Foxtable 2019 - 高级开发指南

Table of Contents

1. 关于文档构成 7
2. 关于移动开发 8
3. 关于视频教程 14
4. 关于功能演示 15
5. 关于网页设计 16
6. HTML 入门 17
6.1 Hello World 17
6.2 服务器 IP 18
6.3 服务器端口 19
6.4 增加入站规则 20
6.5 开启多个服务 23
6.6 显示不同的内容 23
6.7 关于 WriteString 25
6.8 关于网页设计 27
6.9 使用 StringBuilder 29
6.10 用 HTML 生成网页 29
6.11 生成结构完整的网页 30
6.12 我想显示一个图片 32
6.13 文件引用的路径问题 34
6.14 使用 JavaScript 文件 36
6.15 使用 CSS 文件 40
6.16 使用第三方库 41
6.17 使用设计好的网页 43
6.18 网页出现乱码啦 44
6.19 接收表单数据 46
6.20 在表单存储标记数据 49
6.21 另一种数据提交方式 49
6.22 Values、GetValues 和 PostValues 51
6.23 简单的数据统计 53
6.24 可能会出现性能问题 56
6.25 一个页面完成一个任务 57
6.26 简化 HttpRequest 事件代码 60
6.27 通用 HttpRqeust 事件头 62
6.28 一个简单的录入界面 64
6.29 文件的上传与接收 66
6.30 避免覆盖同名文件 68
6.31 手机拍照与上传 69
6.32 带上传功能的录入页面 71
6.33 生成表格 73
6.34 分页显示数据 74
6.35 必须掌握的 Row_Number 76

6.36 分页显示后台数据 80
6.37 一个综合性的例子 81
6.38 使用 Cookie 86
6.39 用户身份验证 87
6.40 特殊内容的显示 91
6.41 Excel 报表与后台数据 93
6.42 提升 Excel 报表效率 97
6.43 适合在手机显示的网页 100
6.44 使用框架生成网页 105
6.45 去掉通用事件头 107
6.46 提高编码效率 108
7. WeUI 框架 109
7.1 使用框架生成网页 109
7.2 请输入正确网址 111
7.3 设置默认网页 112
7.4 用 Chorme 模拟手机 113
7.5 定义表单 114
7.6 定义按钮 115
7.7 文本输入框 117
7.8 日期输入框 121
7.9 数值输入框 124
7.10 列表输入框 126
7.11 使用 InputCell 129
7.12 逻辑开关 132
7.13 多行文本框 133
7.14 单选列表项 135
7.15 复选列表项 137
7.16 接收表单数据 139
7.17 Class 和 Attribute 142
7.18 再谈按钮类型 143
7.19 使用 MsgPage 145
7.20 一个录入界面 147
7.21 引用外部文件 152
7.22 InsertHTML 155
7.23 插入标记数据 156
7.24 动态列表项目之一 159
7.25 动态列表项目之二 161
7.26 动态列表项目之三 163
7.27 区分多个提交按钮 163
7.28 使用 Uploader 166
7.28.1 使用 Uploader 166
7.28.2 改变标题位置 169
7.28.3 在微信浏览器中使用相机 171
7.28.4 接收上传的文件 172

7.28.5 图片浏览窗口 174
7.28.6 我只想浏览图片 175
7.28.7 使用缩略图 177
7.28.8 开启图片删除功能 179
7.28.9 删除后台文件 180
7.29 使用 PageTitle 182
7.30 使用 PageFooter 183
7.31 使用 List 185
7.32 给 List 加上徽章 187
7.33 使用 Grid 188
7.34 使用 Panel 191
7.35 给 Panel 加上徽章 193
7.36 使用 Article 195
7.37 使用 Dialog 198
7.38 showDialog 函数 202
7.39 使用 TabBar 203
7.40 使用 Page 206
7.41 使用 Toast 209
7.42 给用户一个提示 213
7.43 使用 TopTips 214
7.44 使用 ActionSheet 217
7.45 使用 Preview 220
7.46 使用 Gallery 222
7.47 结合 Article 和 Gallery 224
7.48 使用 Progress 226
7.49 动态生成图表 228
7.50 使用 Cookie 230
7.51 设计首页和登录页面 231
7.52 让登录变得更安全 236
7.53 使用 AJAX 242
7.53.1 使用 AJAX 函数 242
7.53.2 setAjaxOptions 242
7.53.3 submitAjaxForm 247
7.53.4 submitAjaxFileds 253
7.53.5 sendAjaxJSON 257
7.53.6 sendAjaxText 259
7.53.7 再谈表单验证 261
7.53.8 接收完整的页面 264
7.53.9 使用回调函数 268
7.54 再探 Uploader 270
7.54.1 增强 Uploader 270
7.54.2 微信内置浏览器问题 272
7.54.3 图片压缩上传 273
7.54.4 图片旋转上传 275

7.54.5 显示上传进度 277
7.55 使用表格 279
7.55.1 生成表格 279
7.55.2 多层表头 282
7.55.3 合并模式 283
7.55.4 显示行号 284
7.55.5 设置列宽 285
7.55.6 高亮显示行列 286
7.55.7 交替行背景颜色 287
7.55.8 逐个单元格添加 288
7.55.9 由 Table 自动生成 290
7.55.10 由 DataTable 自动生成 292
7.55.11 手工编码生成 294
7.55.12 编码合并单元格 297
7.55.13 分页显示数据 299
7.55.14 生成汇总模式 301
7.55.15 数据筛选 303
7.55.16 数据筛选与分页 305
7.55.17 给表格加上菜单 310
7.55.18 传递主键和页码 312
7.55.19 手工编码传递主键 315
7.55.20 菜单综合示例 318
7.55.21 改用按钮操作 324
7.55.22 显示动态菜单 327
7.55.23 又一个综合示例 329
7.56 使用关联表 340
7.56.1 后台表结构 340
7.56.2 如何传递命令 341
7.56.3 list.htm 的设计思路 342
7.56.4 edit.htm 的设计思路 345
7.56.5 准备 JS 文件 353
7.56.6 HttpRequest 事件代码 355
7.56.7 List 函数 355
7.56.8 Edit 函数 363
7.56.9 Save 函数 375
7.56.10 GetOrderID 函数 377
7.56.11 继续扩展功能 378
8. 客户端类 392
8.1 HttpClient 392
8.2 接收数据 392
8.3 发送数据 393
8.4 收发 Cookie 393
8.5 模拟表单 395
8.6 接收文件 396

8.7 上传文件 396
8.8 接收 DataTable 398
8.9 获取内容信息 399
8.10 分两步获取信息和内容 400
8.11 单纯获取内容信息 402
8.12 再谈头部信息 403
8.13 一个经典示例 405
9. JSON 解析 406
9.1 解析 JSON 406
9.2 解析实例 411
9.3 生成 JSON 413
9.4 生成实例 417
9.5 压缩 JSON 419
9.6 关于引号 419
9.7 解析 XML 420
9.8 生成 XML 423
9.9 CDATA 结点 425
10. 微信接口 427
10.1 关于微信接口 427
10.2 WXBizCrypt 类 427
10.3 公众号接口 429
10.3.1 申请测试号 429
10.3.2 接入 HTTP 服务 430
10.3.3 获取 access_token 432
10.3.4 消息接口 433
10.3.5 素材接口 456
10.3.6 用户接口 464
10.3.7 群发接口 477
10.3.8 二维码接口 486
10.3.9 菜单接口 491
10.3.10 网页授权接口 504
10.3.11 在网页实现扫码输入数据 510
10.3.12 中控服务器的设计 513
10.3.13 被动回复还是主动发送？ 515
10.4 企业号接口 516
10.4.1 接入 HTTP 服务 516
10.4.2 获取 AccessToken 517
10.4.3 消息接口 518
10.4.4 菜单接口 535
10.4.5 素材接口 541
10.4.6 通讯录接口 548
10.4.7 网页授权接口 566
10.4.8 在网页实现扫码输入数据 568
10.5 客户上传文件的下载 571

11. 更多接口 572
11.1 阿里云全国物流查询接口 572
12. 类型参考 573
12.1 HttpServer 573
12.2 HttpRequest 574
12.3 HttpClient 577
12.4 WeUI 578
12.5 WXBizCrypt 579
12.6 JObject 580
12.7 XObject 581
12.8 使用缓存 581
12.9 常用函数 582
13. 异步编程 583
13.1 关于异步编程 583
13.2 主线程和子线程 583
13.3 异步函数和同步函数 586
13.4 异步函数示例 587
13.5 同步函数示例 589
13.6 能用的和不能用的 592
13.7 异步函数与 UI 592
13.8 不要滥用异步编程 594
13.9 HttpServer 与异步编程 594
13.9.1 HttpServer 与异步编程 594
13.9.2 用异步函数生成网页 595
13.9.3 异步接收数据和文件 597
13.10 OpenQQ 与异步编程 599
13.10.1 QQServer 与异步编程 599
13.10.2 QQServer 异步登录示例 599
14. Web 数据源 600
14.1 关于三层架构和 Web 数据源 600
14.2 在服务端建立 Web 数据源 601
14.3 在客户端连接 Web 数据源 603

1. 关于文档构成

本文档是 Foxtable 的《高级开发指南》，是 Foxtable 主文档的补充，用于介绍部分 Foxtable 高级开发版特有的功能，例如网页设计入门、手机管理系统的开发，微信接入、三层架构设计和多线程编程等等。

我们必须在已经掌握《使用指南》和《开发指南》的基础上，才能学习《高级开发指南》。
本文档主要由七个部分组成

- **HTML 入门**

  这一章介绍了如何开启一个 HTTP 服务，如果和用 HTML 生成网页，如何接收用户通过网页输入的数据，如何保存用户上传的文件等等。一般用户对于 HTML 稍作了解即可，因为实际开发的时候，我们一般会用内置的 WeUI 框架生成网页。

- **WeUI 框架**

  这一张是重点，通过这一章，你会发现在 Foxtable 生成网页是多么的简单。
  WeUI 框架来自于腾讯，我们对其进行了一些扩展，增加了表格和 AJAX 功能，通过框架，我们可以用非常简单的代码生成专业效果的网页，本框架主要针对移动设备，虽然也能在 PC 上使用，但在手机上的效果更好。

- **客户端类**

  这一章介绍了客户端类 HttpClient，Foxtable 通过 HttpClient 和 HTTP 服务进行交互，包括第三方的 HTTP 服务。我们在进行微信开发的时候，实际上就是通过 HttpClient 调用腾讯提供的 HTTP 服务。

- **JSON 解析**

  这一章介绍了 JSON 和 XML 数据的解析生成，互联网中的数据传递格式主要是 JSON 和 XML，例如进行微信开发的时候，你会发现除了回调部分的数据是 XML 格式，其余都是 JSON 格式。通过这一章，你会发现在 Foxtable 进行 JSON 和 XML 的解析与生成，实在是太轻松了。

- **微信接口**

  这一章介绍了如何在 Foxtable 进行微信开发，分别针对公众号和企业号进行讲述，基本上涵盖了所有微信常用接口，由于我们进行了适度的封装，微信开发变得前所未有的轻松。实际上不仅是微信开发，你可以用 Foxtable 完成任何第三方接口的开发，帮助就提供了一个阿里云快递查询接口的开发示例，只有短短的三行代码而已 。

- **Web 数据源**
  Foxtable 默认是两层结构，客户端直接连接数据库，这一章介绍如何使用 Web 数据源设计三层架构的系统，使得客户端不再直连数据库。

- **异步编程**
  这一节章介绍如何如何用异步编程同时处理多个任务，以提高服务器的负载能力。

2. 关于移动开发

Foxtable 的高级开发版提供了移动开发功能。

Foxtable 的移动开发功能以 B/S 形式实现，内置手机网页自动生成功能，你不需要额外的学习，你可以完全不懂网页设计，也可以快速开发基于手机的企业管理系统，包括服务端的架设和网页的生成，全都可以在 Foxtable 中用最简单的方式完成，不仅简单到人人都能掌握，而且效率极高，相对于其它开发工具，代码节省了 85%左右。

此外高级开发版还提供了微信接入功能，用户可以直接通过微信输入、查询、统计和审核数据，还可以直接通过手机扫描二维码并将扫描结果发送到 Foxtable。

下面是 Foxtable 自动生成的一些手机网页。

一个订单编辑页面：


编辑订单
编号
161203005
客户
柳传志
日期
2016-12-03
总数量
10976
总金额
15664
编辑明细
产品
Foxtable 开发版
数量
4688
单价
2
金额
9376
产品
数量
单价
金额
Foxtable 开发版
4688
2
9376
iphone 6 plus
6288
1
6288
增加明细
保存
取消

图片浏览和上传页面：



输入界面很灵活：



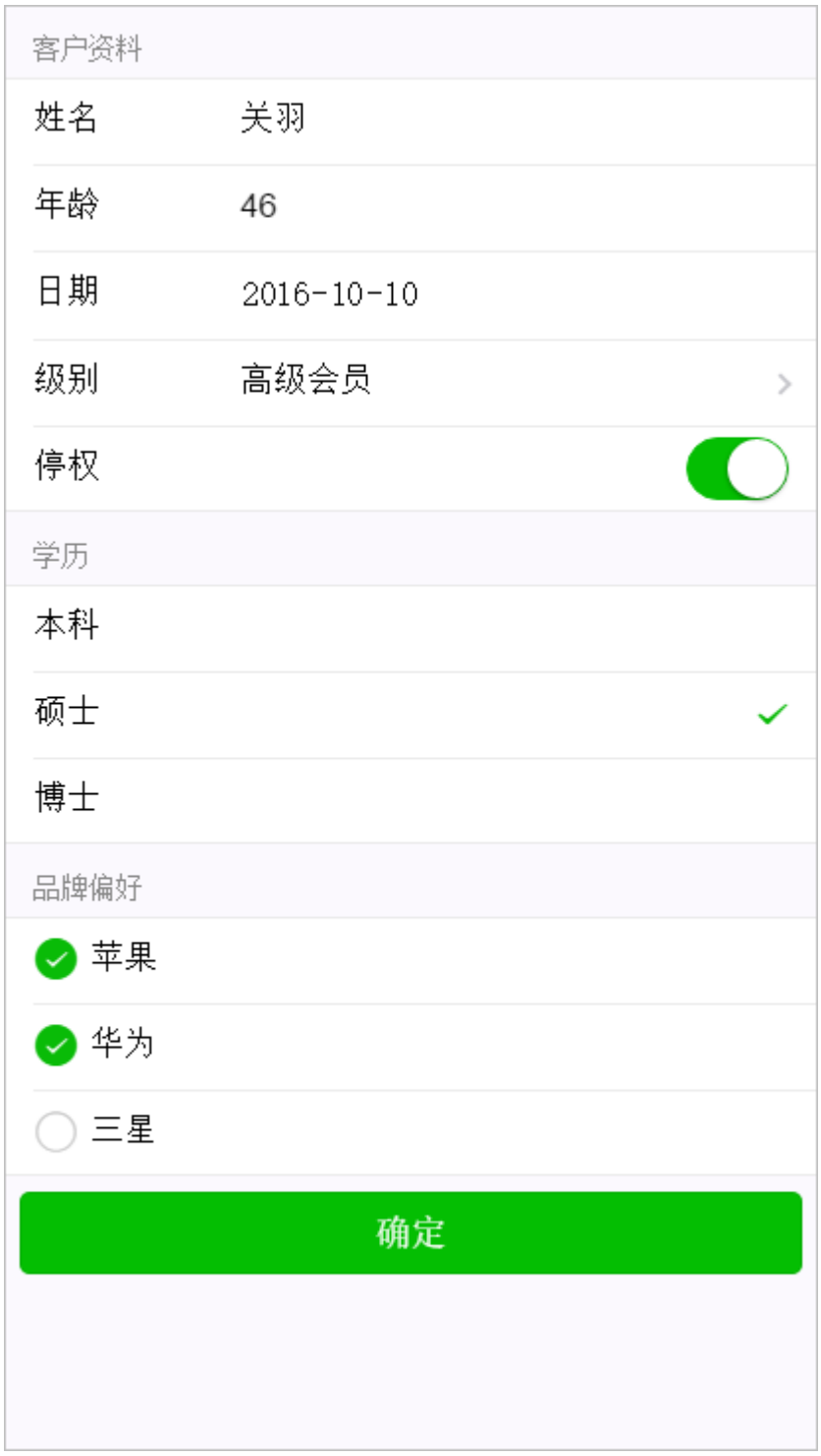
可以直接在手机端生成 Excel 报表：



可以直接扫描条形码或者二维码输入数据：



提供了多种输入组件：



3. 关于视频教程

我们并不建议您看视频来学习 Foxtable，我们在编写帮助的时候，是将用户当作零基础来看待的，一般用户，只需踏踏实实按顺序学习帮助文件都能掌握。

如果你在学习和使用过程遇到问题，可以在我们的论坛发帖提问(建议用 IE 或 Chorme 访问)，我们的工程师会非常乐意为您服务：

http://www.foxtable.com/support/

但确实有部分用户看帮助学习有困难，所以我们也提供了视频教程：

http://www.foxtable.com/video/

必须强调一下，任何开发工具，都没有办法通过视频完全掌握，视频只能带您入门，最好的学习方式就是四个字：啃透帮助。

Foxtable 已经将管理软件开发的门槛降低到人人都能掌握的程度，已经让大量的普通企业管理人员开发出很专业的管理系统，甚至让一些资深程序员也目瞪口呆，但我们依然没有办法让用户看看视频就能完全掌握。

4. 关于功能演示

你可以用微信扫描以下二维码，关注 Foxtable 的公众号：

我们在这个公众号上做了一个简单的示例，进入公众号后，单击菜单“移动测试”可以进入这个示例。

我们很难用一个例子演示完毕 Foxtable 的移动开发功能，只要你认真学习本文档，就能很轻松地用 Foxtable 设计出很专业的手机端管理系统。

5. 关于网页设计

在本文档的开始部分，我们会介绍一些网页设计方面的知识，让大家了解一下网页设计是怎么回事。

实际上你可以完全不会网页设计，Foxtable内置了网页自动生成功能(通过扩展后的WeUI框架实现)，你不需要有任何的网页设计知识，一样可以设计出具备专业效果的网页来。

例如你只需寥寥十来行代码，就可以设计出下面的网页来：



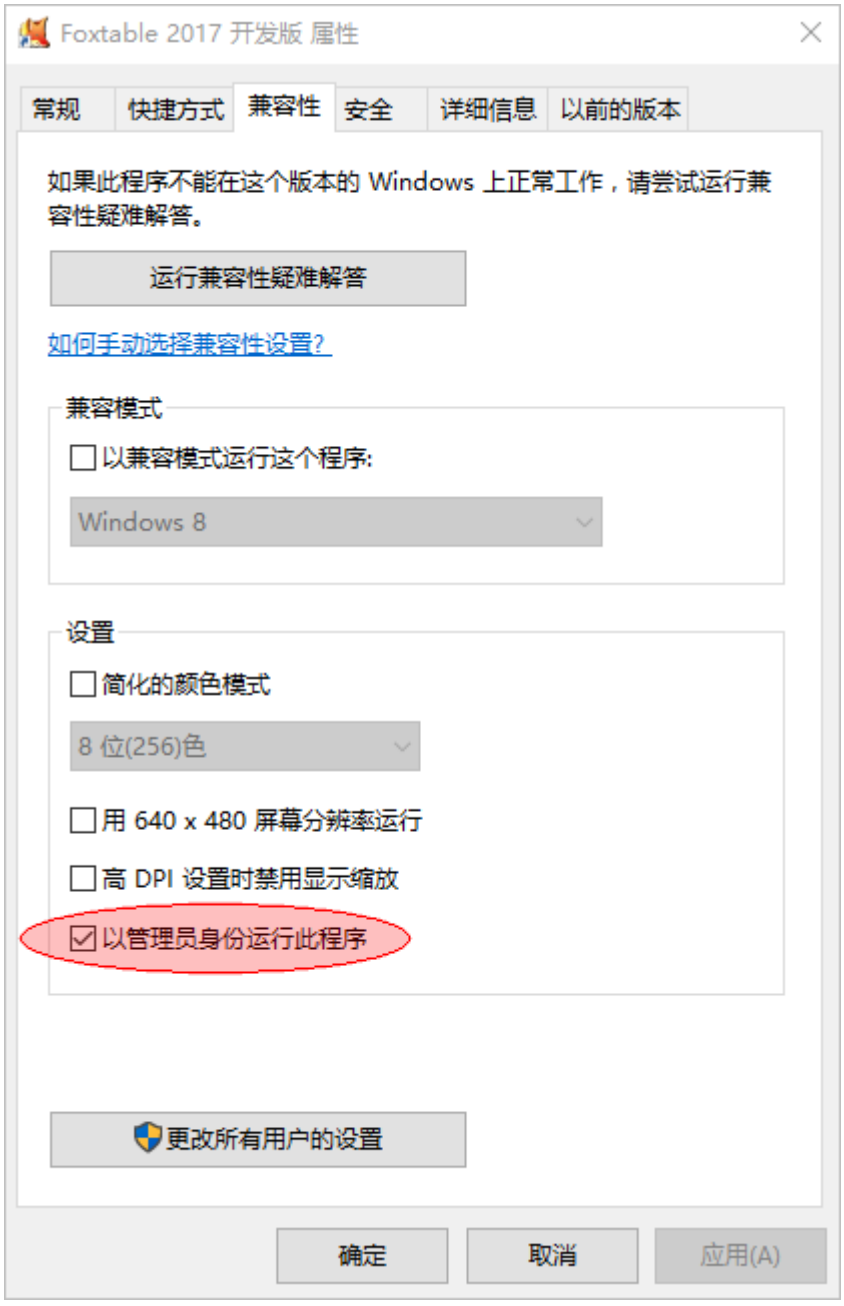
6. HTML 入门

6.1 Hello World

我们先用一个经典的例子，来说明如何使用 Foxtable 实现 http 服务。

首先是一个重要提示：要启动 http 服务，必须以管理员身份运行 Foxtable。

你可以在 Windows 的桌面上右击 Foxtable 的快捷方式，在快捷菜单中的单击属性命令，然后在兼容性页面中勾选“以管理员身份运行此程序”：



启动 Foxtable 后，将下面的代码复制到命令窗口执行：

```
HttpServer.Prefixes.Add("http://127.0.0.1/")
HttpServer.Start()
```

然后在菜单的“管理项目”功能区，单击“网络监视器”，将“HttpRequest”事件的代码设置为：

```
e.WriteString("Hello World")
```

现在打开你的浏览器，输入地址：http://127.0.0.1，会显示如下内容：



是的，我们只用了 3 行代码，就建立了一个简单的 http 服务系统。

通过下面的代码，可以关闭 http 服务：

```
HttpServer.Close()
```

现在为了接下来的测试方便，我们可以设计一个窗口，窗口有两个按钮，分别用于开启和关闭 http 服务：



按钮的代码就是前面介绍的，就不重复了。

注意每次开启新的服务之前，都必须关闭之前的服务，否则会无效或出现错误提示。

6.2 服务器 IP

上一节我们搭建了一个简单的 http 服务，你只能在本机访问这个 http 服务，因为 127.0.0.1 表示的是本机 IP。
我们可以设置多个 IP，假定服务器的局域网 IP 是：192.168.0.100，我们可以将开启服务按钮的代码改为：

```
HttpServer.Prefixes.Add("http://127.0.0.1/")
HttpServer.Prefixes.Add("http://192.168.0.100/")
HttpServer.Start()
```

这样局域网中的其他电脑，就能通过局域网IP来访问我们的http服务了：



如果服务器有公网 IP，一样可以加入，这样外网用户就能通过公网 IP 来访问我们的 http 服务。
如果服务器有多个 IP，可以用下面的代码来开启 http 服务：

```
HttpServer.Prefixes.Add("http://*/")
HttpServer.Start()
```

这样用户就可以通过指向服务器的任何一个IP来访问http服务，包括本机IP、局域网IP和公网IP。

建议今后都采用以上代码来开启http服务，这样就无需因IP的变动而修改代码了。

如果是公网，可以设置域名指向你的服务器，让用户通过域名来访问你的http服务。

提示：为了测试方便，接下来所有例子，都将通过127.0.0.1来访问。

6.3 服务器端口

如果我们不指定端口，那么 http 服务使用的就是默认的端口 80。
下面的代码：

```
HttpServer.Prefixes.Add("http://*/")
HttpServer.Start()
```

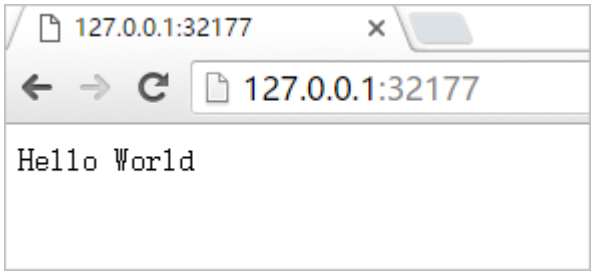
等同于：

```
HttpServer.Prefixes.Add("http://*:80/")
HttpServer.Start()
```

有时，我们需要给自己的 http 服务更换端口。
例如你公司已经有了一台 web 服务器，现在你需要利用这台服务器运行 Foxtable 的 http 服务，那么就不能使用 80 端口了，因为这样会和原来的 web 服务发生冲突。
换个端口开启 http 服务很简单，例如：

```
HttpServer.Prefixes.Add("http://*:32177/")
HttpServer.Start()
```

现在你要访问这个http服务，地址必须加上端口号才行：

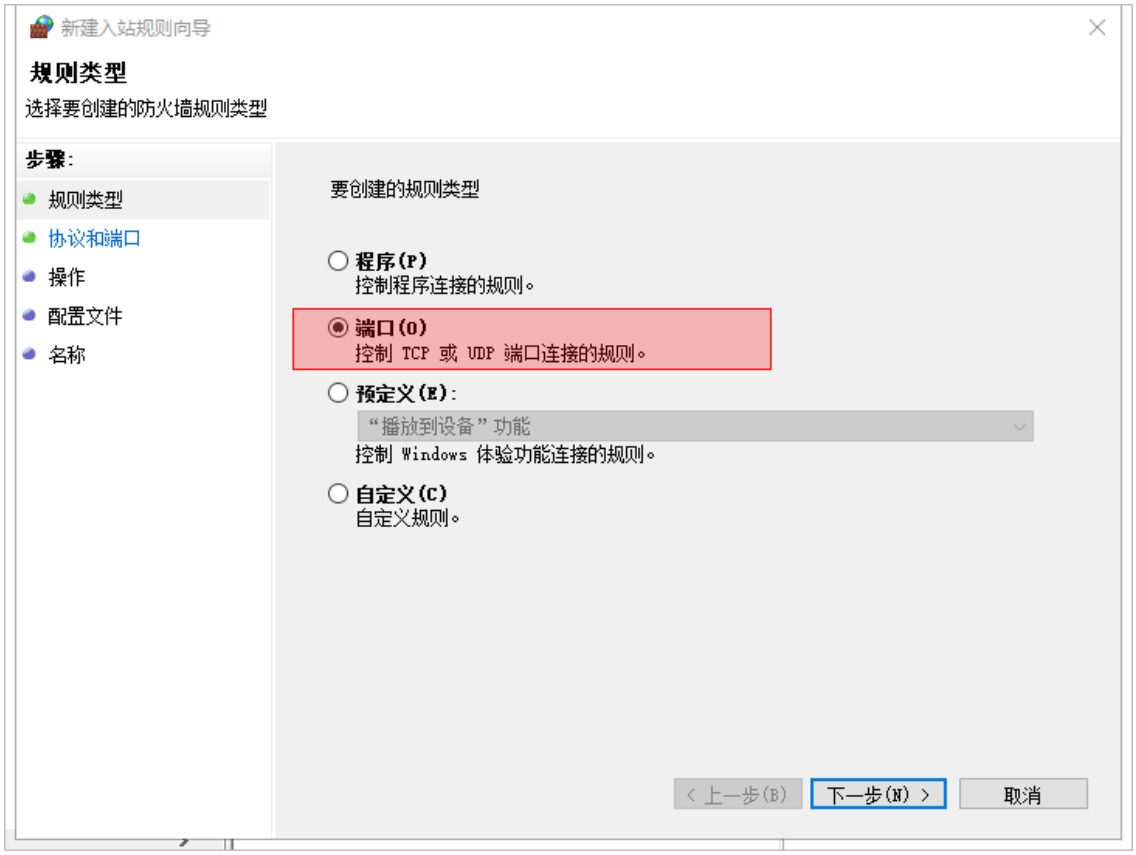


为了方便，就下来的例子还是使用默认的端口80。

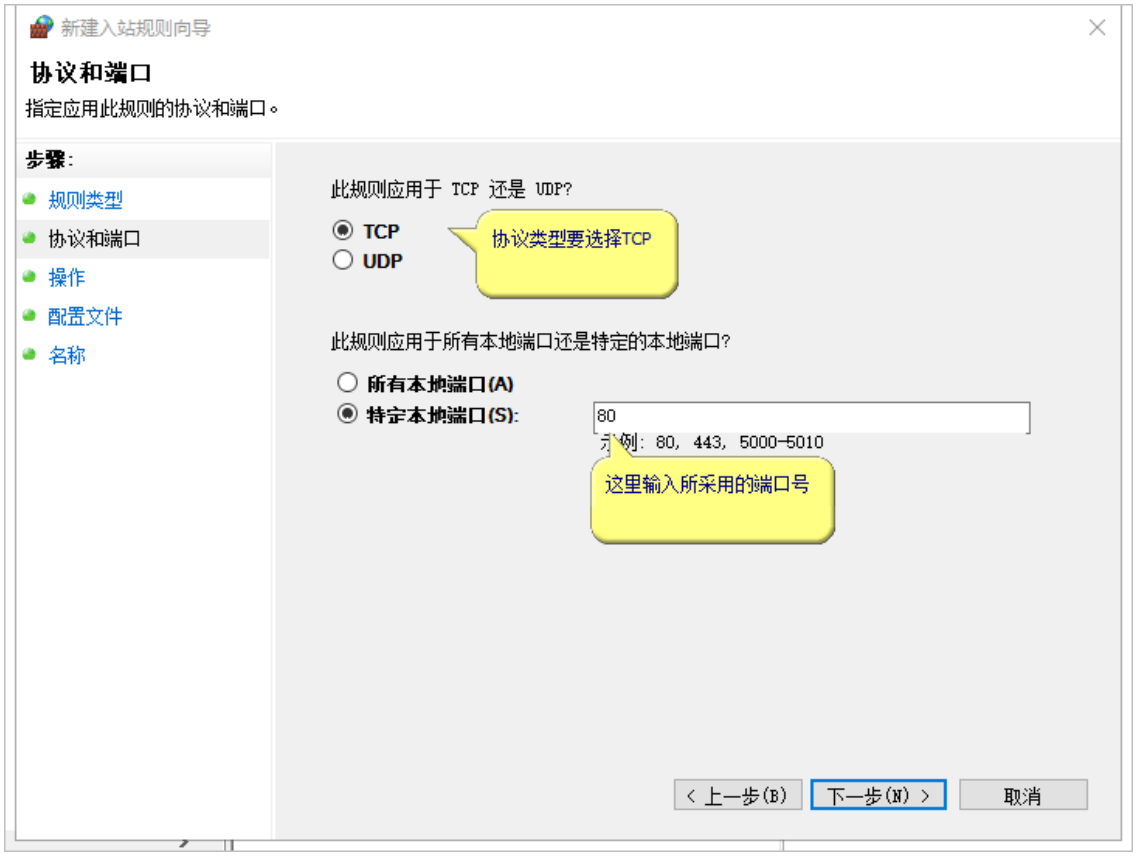
6.4 增加入站规则

如果你在另一台电脑访问我们新建立的 http 服务，很可能无法访问。

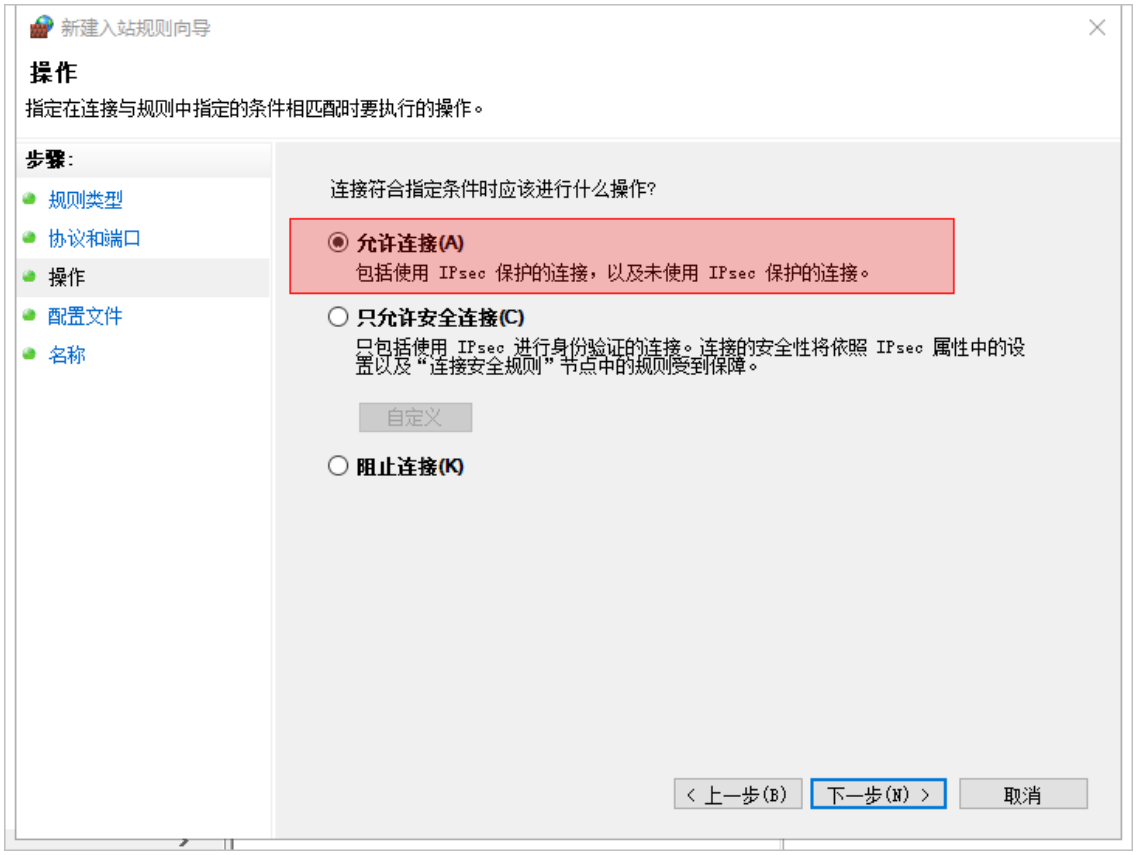
通常这是防火墙造成的，我们可以新建一个入站规则，让防火墙允许客户端访问我们的 http 服务。
以 windows 10 自带的防火墙为例，新建一个入站规则的步骤为：
1、在 Windows 的控制面板，打开“Windows 防火墙”。
2、在“Windows 防火墙”窗口，单击“高级设置”。
3、在“高级设置”窗口，单击“入站规则”，然后在右侧单击“新建规则”。
4、选择规则类型为“端口”：



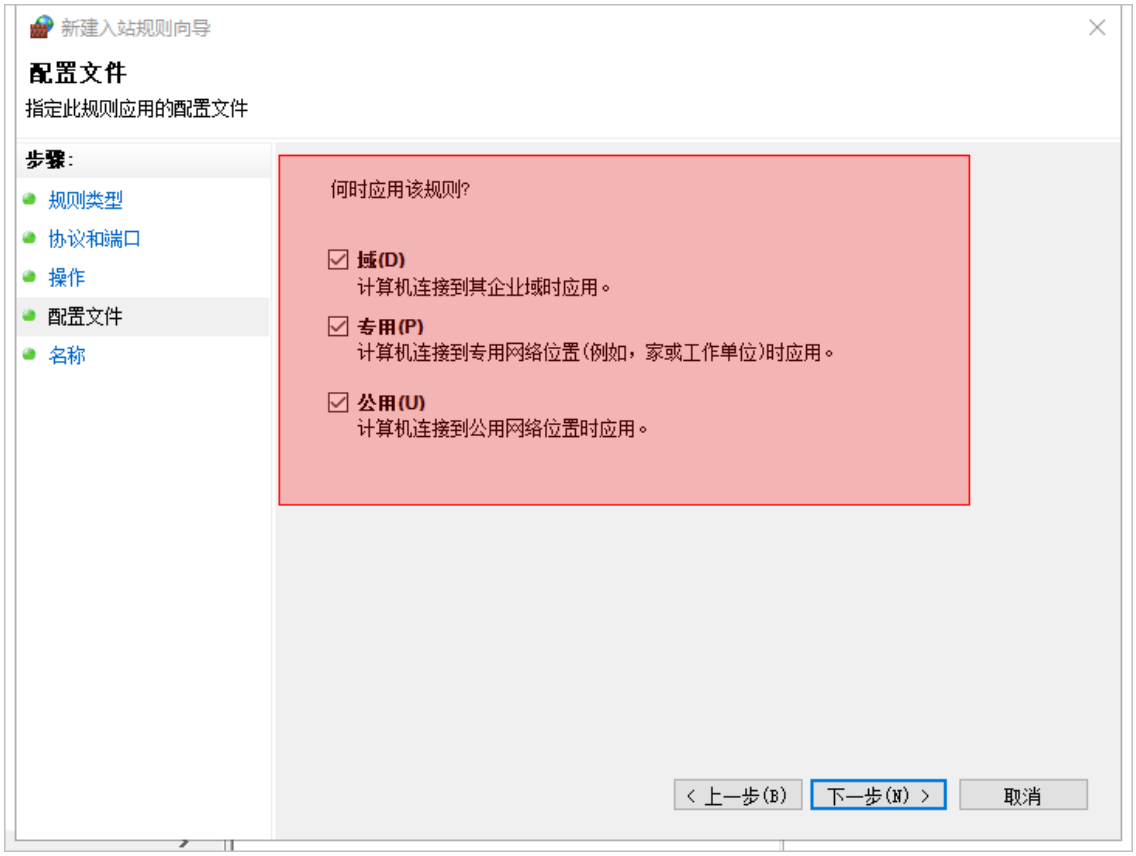
5、协议类型选择 TCP，并输入你的 http 服务所采用的端口号：



5、接下来选择允许接入：



6、应用规则勾选所有：



7、最后输入规则名称即可，名称可以随便取，最好方便识别。

6.5 开启多个服务

你可以一次开启多个 http 服务，例如：

```
HttpServer.Prefixes.Add("http://*/")
HttpServer.Prefixes.Add("http://*:32177/")
HttpServer.Prefixes.Add("http://192.168.0.100:32188/")
HttpServer.Start()
```

上面的代码开启了三个 http 服务，头两个服务可以通过指向服务器的任何 ip 来访问，例如：
http://127.0.0.1/
http://192.168.0.100/
http://127.0.0.1:32177/
http://192.168.0.100:32177/
第三个服务则只能根据固定的 IP 地址访问：
http://192.168.0.100:32188/
重要提示：当你通过多个 IP 开启服务时，只要有一个 IP 错误，所有服务都是不能启动的。

6.6 显示不同的内容

在上一节中，我们开启了三个 http 服务，当我们分别访问这三个服务时，显示的内容都是“Hello Word”。
显然，这不符合实际需要，真正的 http 服务应该能根据用户输入的不同地址，显示不同的内容。

HttpRequest 事件的 e 参数有很多，其中有三个 e 参数，用于获取用户访问路径的各部分。

Host: 返回 IP 地址(或域名)
Port：返回端口号
Path：返回路径，含文件名。

假定用户输入的访问路径是"http://192.168.0.100:32188/fox/",这三个 e 参数的返回值分别是：

Host: 192.168.0.100
Port: 32188
Path: fox

假定用户输入访问路径是"http://192.168.0.100:32188/fox/china.htm"，这三个 e 参数的返回值分别是：

Host: 192.168.0.100
Port: 32188
Path: fox\china.htm

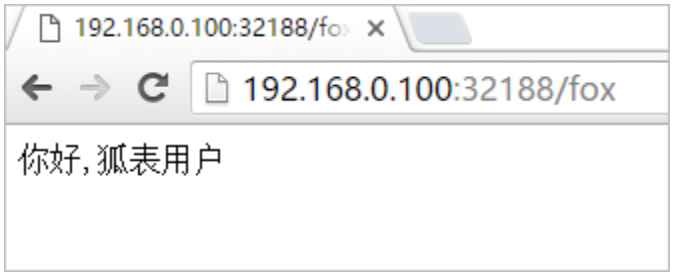
注意：

1、e 参数 path 中的路径分割符号是"\"，不是"/"。
2、尽量通过路径和文件名来区分用户的访问请求，不要占用太多端口，因为端口是有限的。

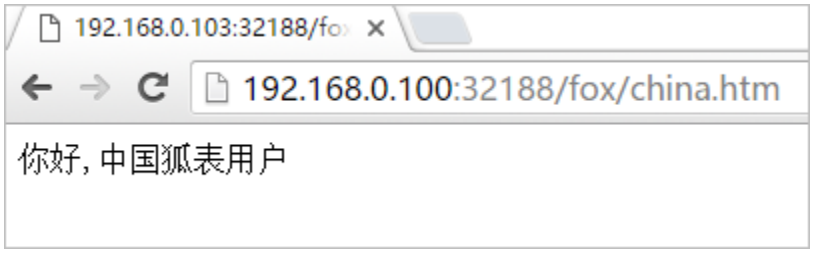
现在就可以根据用户访问的不同端口、路径和和文件，甚至是不同的 IP，来显示不同的内容，例如将 HttpRequest 的代码设置为：

```
If e.host = "127.0.0.1" Then
    e.WriteString("你是本地用户")
Else
    Select Case e.Port
        Case 80
            e.WriteString("Hello World")
        Case 32177
            e.WriteString("你好,世界")
        Case 32188
            If e.path = "fox" Then
                e.WriteString("你好,狐表用户")
            ElseIf e.path = "fox\china.htm" Then '注意这里的路径分隔符是"\"
                e.WriteString("你好,中国狐表用户")
            Else
                e.WriteString("你好,未找到网页")
            End If
        Case Else
            e.WriteString("你好,端口错误")
    End Select
End If
```

如果你通过地址“http://192.168.0.100:32188/fox”访问 http 服务，将显示:



如果你通过地址"http://192.168.0.100:32188/fox/china.htm"访问 http 服务，将显示：

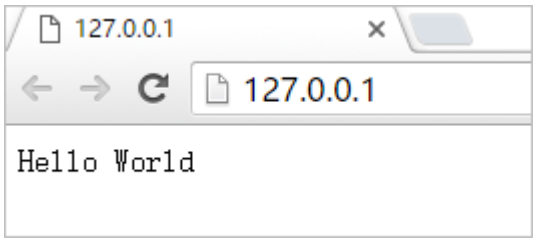


6.7 关于 WriteString

在前面的例子中，在 HttpReuest 用下面的代码：

```
e.WriteString("Hello World")
```

在浏览器中显示了下面的内容：



需要注意的是，每次只能执行一次 WriteString 或 WriteFile(后面会介绍)，例如下面的代码是无法正常运行的：

```
e.WriteString("Hello World")
e.WriteString("Hello China") '这一行会出错
```

需要将两个字符串合并为一个，再用 WriteString 写入浏览器，例如：

```
Dim rs As String = "Hello World"
rs = rs & vbcrlf & "Hello China"
e.WriteString(rs)
```

在浏览器中显示的内容是：



你肯定会奇怪，vbcrlf是回车换行，按理应该分两行显示，怎么上面的内容只显示了一行呢？
这就是涉及到HTML标签的知识，在HTML中，连续的多个空格或换行，都会被显示为1个空格，而换行用
表示。

所以要分开两行显示，正确的HttpReuest事件代码是：

```
Dim rs As String = "Hello World"
rs = rs & "<br/>Hello China"
e.WriteString(rs)
```

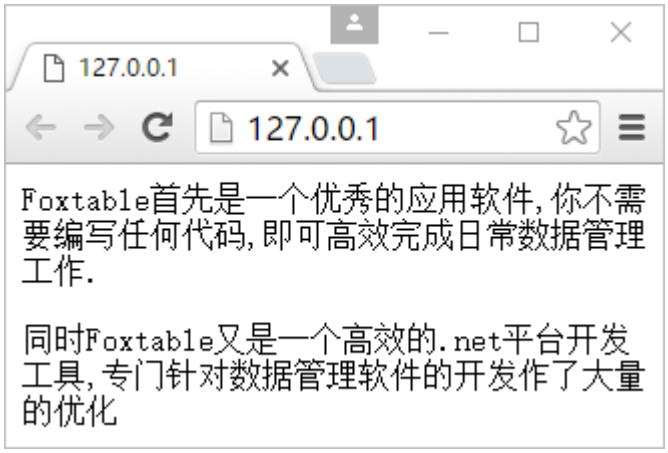
在浏览器显示的内容是：



如果要分段显示文字，可以将段落放在 和 之间，例如将HttpReuest事件代码设置为：

```
Dim rs As String = "<p>Foxtable首先是一个优秀的应用软件,你不需要编写任何代码,即可高效完成日常数据管理工作.</p>"
rs = rs & "<p>同时Foxtable又是一个高效的.net平台开发工具,专门针对数据管理软件的开发作了大量的优化</p>"
e.WriteString(rs)
```

在浏览器显示的结果为：



用a标签可以插入超链接，例如将HttpReuest事件代码设置为：

```
Dim rs As String = "<a href='http://www.foxtable.com/'>单击此处</a>访问Foxtable主页"
e.WriteString(rs)
```

<a>标签的href属性指定链接地址，属性用单引号括起来，<a>和之间为要显示的文本，上述代码在浏览器显示的结果为：



HTML的标签有很多，我们并不打算一一介绍，教会大家HTML不是我们的任务，有人比我们做得更好，有兴趣的话，可以访问：

http://www.w3school.com.cn/html/index.asp

通常不超过半天时间就可以掌握了。

6.8 关于网页设计

上一节我们接触了一些常用的HTML标签，但我们不会有专门的章节来介绍HTML标签。
因为我们并不打算教会大家网页设计，这不是我们的长处，已经有很多比我们更擅长的人在做这项工作。
我只想告诉大家，如果你掌握了网页设计，你之前所有的知识，包括HTML、CSS、JavaScript和JQuery等等，在Foxtable都能继续派上用场。

可以不会网页设计

如果你完全不会网页设计，也没有关系，Foxtable内置了网页自动生成功能，不需要有任何的网页设计知识，一样可以设计出具备专业效果的网页来。

例如你只需寥寥十来行代码，就可以设计出下面的网页来：



如果你想提前了解一下，可以参考：使用框架生成网页

会网页设计更好

自动生成，意味着轻松，也意味着约束，如果你懂一些网页设计，你就可以突破这种约束，甚至可以完全放弃Foxtable的自动网页生成功能。

如果你打算学习网页设计，我给你推荐一个网站：http://www.w3school.com.cn/h.asp

对于一般用户来说，花一两天的时间掌握HTML、HTML5和CSS，就足够使用了，如果想做一些动态的效果，可以再掌握一下JavaScript和CSS3。

6.9 使用 StringBuilder

前面的例子，我们是直接将字符串组合起来，然后写入客户端的浏览器：

```
Dim rs As String = "Hello World"
rs = rs & "<br/>Hello China"
e.WriteString(rs)
```

如果要写入的内容比较多，这种方法是比较低效的，因为.net合并字符串的效率并不好。

我们建议用StringBuilder来合并字符串，例如：

```
Dim sb As New StringBuilder
sb.AppendLine("Hello World")
sb.AppendLine("<br/>Hello China")
e.WriteString(sb.ToString)
```

生成的网页内容越多，使用StringBuilder的效率优势就越明显，今后例子都将采用这种方式。

6.10 用 HTML 生成网页

本节内容适合已经掌握 HTML 的用户，其实不管你是否已经掌握 HTML，建议都学习一下，因为：真的很简单！

现在我们将 HttpRequest 事件代码改为：

```
Dim sb As New StringBuilder
sb.AppendLine("<form enctype='multipart/form-data' method='post' id='form1' name='form1'>")
sb.AppendLine("产品: <input name='cp' id='cp'><br/><br/>")
sb.AppendLine("客户: <input name='kh' id='kh'><br/><br/>")
sb.AppendLine("雇员: <input name='gy' id='gy'><br/><br/>")
sb.AppendLine("单价: <input type='number' name='dj' id='dj'><br/><br/>")
sb.AppendLine("折 扣 : <input type='number' name='zk' id='zk' min='0' max='0.15' step='0.01'><br/><br/>")
sb.AppendLine("数量: <input type='number' name='sl' id='sl'><br/><br/>")
sb.AppendLine("日期: <input type='date' name='rq' id='rq'><br/><br/>")
sb.AppendLine("<input Type='submit' name='Sumbit' id='Sumbit' value='确定'>")
sb.AppendLine("</form>")
e.WriteString(sb.ToString)
```

我们用上面的代码生成了一个表单，在Chorme浏览器中的显示效果为：



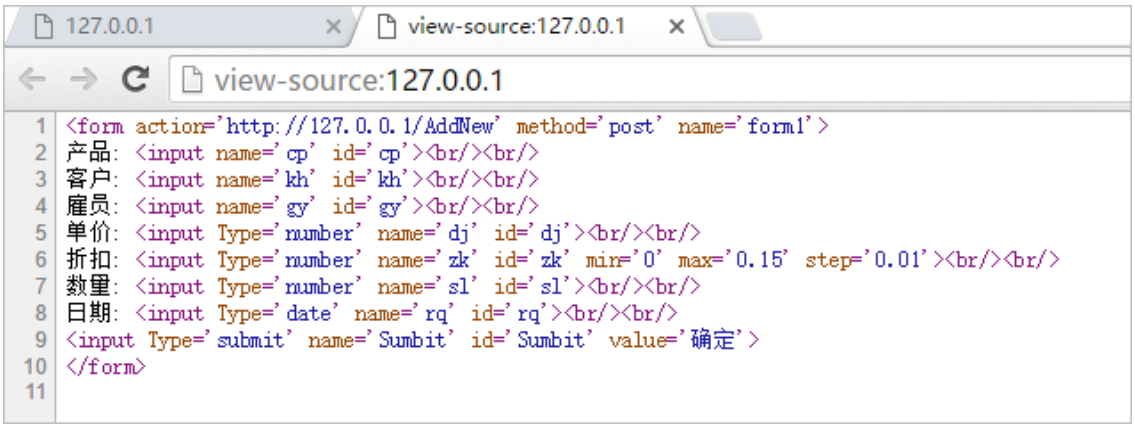
即使你完全不懂HTML，看懂上面的代码也没有问题，以下面这一行代码为例：

```
sb.AppendLine(" 折 扣 ： <input type='number' name='zk' id='zk' min='0' max='0.15' step='0.01'><br/><br/>")
```

表示增加一个输入框，输入类型为数字(type='number')，允许最小值为 0(min='0')，允许最大值为 0.15(max='0.15'),允许输入精度为 0.01(step='0.01')，id 为 zk(id='zk')，name 为 zk(name='zk')。
在不同的浏览器，上述代码生成的页面会有不同的显示效果，例如对于日期输入框，Chorme 可以显示一个下拉 日历供选择，而且你只能输入日期，对于 IE 来说，这个日期输入框和普通的文本输入框没有任何差别。
教会你 HTML 不是我的任务，有人干这个我们擅长很多，如果你有兴趣，可以访问：http://www.w3school.com.cn/html/index.asp，通常不会超过半天时间就能基本掌握。

6.11 生成结构完整的网页

在浏览器打开我们上一节生成的网页，右击，查看源代码，可以看到其代码为：



没有 head？ 没有 body？ 如果你接触过 HTML 网页设计，你会知道这不是一个结构完整的网页。你也不要担心，正如我们所演示的，目前所有的浏览器，都能正确解释上面的代码。

不过遵循规范总是没有错，我们来改进一下 HttpRequst 事件的代码：

```
Dim sb As New StringBuilder
sb.AppendLine("<!doctype html>")
sb.AppendLine("<html>")
sb.AppendLine("<head>")
sb.AppendLine("<meta charset='utf-8'>")
sb.AppendLine("<title>表单</title>")
sb.AppendLine("</head>")
sb.AppendLine("<body>")
sb.AppendLine("<form enctype='multipart/form-data' method='post' id='form1' name='form1'>")
sb.AppendLine("产品: <input name='cp' id='cp'><br/><br/>")
sb.AppendLine("客户: <input name='kh' id='kh'><br/><br/>")
sb.AppendLine("雇员: <input name='gy' id='gy'><br/><br/>")
sb.AppendLine("单价: <input Type='number' name='dj' id='dj'><br/><br/>")
sb.AppendLine("折扣: <input Type='number' name='zk' id='zk' min='0' max='0.15' step='0.01'><br/><br/>")
sb.AppendLine("数量: <input Type='number' name='sl' id='sl'><br/><br/>")
sb.AppendLine("日期: <input Type='date' name='rq' id='rq'><br/><br/>")
sb.AppendLine("<input Type='submit' name='Sumbit' id='Sumbit' value='确定'>")
sb.AppendLine("</form>")
sb.AppendLine("</body>")
sb.AppendLine("</html>")
e.WriteString(sb.ToString)
```

现在生成的就是一个结构完整的网页，在浏览器看到的代码是：

表单 view-source:127.0.0.1

view-source:127.0.0.1

```
<!doctype html>
<html>
<head>
<meta charset='utf-8'>
<title>表单</title>
</head>
<body>
<form action='http://127.0.0.1/AddNew' method='post' name='form1'>
产品: <input name='cp' id='cp'><br/><br/>
客户: <input name='kh' id='kh'><br/><br/>
雇员: <input name='gy' id='gy'><br/><br/>
单价: <input Type='number' name='dj' id='dj'><br/><br/>
折扣: <input Type='number' name='zk' id='zk' min='0' max='0.15' step='0.01'><br/><br/>
数量: <input Type='number' name='sl' id='sl'><br/><br/>
日期: <input Type='date' name='rq' id='rq'><br/><br/>
<input Type='submit' name='Sumbit' id='Sumbit' value='确定'>
</form>
</body>
</html>
```

6.12 我想显示一个图片

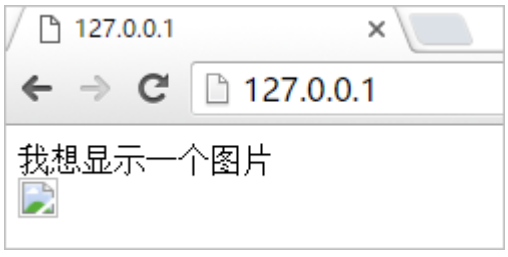
本节内容适合已经掌握 HTML 的用户，其实不管你是否已经掌握 HTML，建议你学习一下，因为：真的很简单！

在 D 盘建立一个名为 web 的目录，复制一个图片文件到这个目录，假定图片文件的名称是 abc.jpg。

我们将 HttpRequest 事件的代码改为：

```
Dim sb As New StringBuilder
sb.Appendline("我想显示一个图片<br/>")
sb.AppendLine("<img src='abc.jpg'>")
e.WriteString(sb.ToString())
```

在浏览器显示的结果为：



img标签用于显示图片，其src属性为要显示的图片文件名称。

不过图片并没有显示，这是意料之中的，因为服务端并没有发送图片文件到浏览器。

浏览器其实是分两步来显示这个网页的：

1、首先获取网页内容：

```
我想显示一个图片<br/>
<img src='abc.jpg'>
```

2、然后浏览器解析网页内容，发现要显示一个图片"abc.jpg"，如是又向服务器发送访问请求：

```
http://127.0.0.1/abc.jpg
```

要显示图片，服务器必须对这个访问请求对出响应，将后台的图片"abc.jpg"发送给客户端的浏览器，浏览器则显示收到的图片。

实际上不单单是图片，网页中引用的任何文件，包括js、css等等，都会单独向服务器发出下载这个文件的请求，服务器必须对此做出响应。

据此，我们修改HttpRequest事件代码为：

```
Dim fl As String = "d:\web\" & e.path '合成含路径的文件名
If filesys.FileExists(fl) '如果是请求一个已经存在的文件
    e.WriteFile(fl) '则发送此文件
Else
    Dim sb As New StringBuilder
    sb.Appendline("我想显示一个图片<br/><br/>")
    sb.AppendLine("<img src='abc.jpg'>")
    e.WriteString(sb.ToString())
End If
```

现在浏览器可以正常显示图片了：



提示：HttpRequest事件的WriteFile方法用于向客户端发送本地文件。

安全问题

上面的代码是不严谨的，因为用户可以下载"d:\web"目录下的所有文件，包括非图片文件。

我们需要进行一些判断，限制用户只能下载图片文件，为此可以将HttpRequest事件代码改为：

```
Dim fl As String = "d:\web\" & e.path
If filesys.FileExists(fl) '如果是请求一个已经存在的文件
    Dim idx As Integer = fl.LastIndexOf(".")
    Dim ext As String   = fl.SubString(idx)
    Select Case ext
      Case ".jpg",".gif",".png",".bmp",".wmf" '只允许请求图片文件
        e.WriteFile(fl)
    End Select
Else
    Dim sb As New StringBuilder
    sb.Appendline("我想显示一个图片<br/><br/>")
    sb.AppendLine("<img src='abc.jpg'>")
    e.WriteString(sb.ToString())
End If
```

6.13 文件引用的路径问题

对于稍具规模的系统，为便于维护，我们很少会将所有文件放在同一个文件夹中。
那么某一文件夹中的网页，如何引用其他文件夹中的文件呢。

使用绝对路径

如果被引用文件的路径是固定的，那么使用绝对路径即可。

假定"d:\web"为网页根目录，图片文件"abc.jpg"位于该目录的"images"子目录下。
那么在任何路径的网页，都可以通过路径"/images/abc.jpg"引用此图片，"/"表示根目录"d:\web"，所以"/images/abc.jpg"就表示"d:\web\images\abc.jpg"。

示例

现在将文件"abc.jpg"复制到目录"d:\web\images"下，然后将 HttpRequest 事件代码改为：

```
Dim fl As String = "d:\web\" & e.path
If filesys.FileExists(fl)
    Dim idx As Integer = fl.LastIndexOf(".")
    Dim ext As String   = fl.SubString(idx)
    Select Case ext
      Case ".jpg",".gif",".png",".bmp",".wmf"
        e.WriteFile(fl)
    End Select
```

```
Else
    Dim sb As New StringBuilder
    sb.Appendline("我想显示一个图片<br/><br/>")
    Select Case e.path
       Case "a.htm"
          sb.AppendLine("<img src='/images/abc.jpg'>")
       Case "sub1\a.htm"
          sb.AppendLine("<img src='/images/abc.jpg'>")
        Case "sub1\sub2\a.htm"
          sb.AppendLine("<img src='/images/abc.jpg'>")
    End Select
    e.WriteString(sb.ToString())
End If
```

现在你通过下面三个网页访问HttpServer：

http://127.0.0.1/a.htm
http://127.0.0.1/sub1/a.htm
http://127.0.0.1/sub1/sub2/a.htm

可以看到，三个不同路径的网页，使用相同的引用路径""\images\abc.jpg""，都可以正常显示图片"d:\web\images\abc.jpg"。

使用相对路径

如果要使用相对路径，需要用到两个重要的符号，一个是表示网页所在的目录的"."，一个是表示父目录的".."

假定网页所在目录为"d:\web"目录，图片文件位于该目录的 images 子目录下，也就是"d:\web\images"目录下，那么使用相对路径的代码为：

```
<img src='./images/abc.jpg'>
```

"."表示网页所在目录"d:\web"，所以"./images"表示"d:\web\images"

假定网页所在目录为"d:\web\sale"目录，图片文件位于"d:\web\images"目录下， 那么使用相对路径的代码为：

```
<img src='../images/abc.jpg'>
```

".."表示"d:\web\sale"的父目录，也就是"d:\web"， 所以"../images"表示"d:\web\images"。

示例

现在将文件"abc.jpg"复制到目录"d:\web\images"下，然后将 HttpRequest 事件代码改为：

```
Dim fl As String = "d:\web\" & e.path
If filesys.FileExists(fl)
    Dim idx As Integer = fl.LastIndexOf(".")
    Dim ext As String   = fl.SubString(idx)
    Select Case ext
       Case ".jpg",".gif",".png",".bmp",".wmf"
```

```
            e.WriteFile(fl)
        End Select
    Else
        Dim sb As New StringBuilder
        sb.Appendline("我想显示一个图片<br/><br/>")
        Select Case e.path
            Case "a.htm"
                sb.AppendLine("<img src='.\images\abc.jpg'>")
            Case "sub1\a.htm"
                sb.AppendLine("<img src='..\images\abc.jpg'>")
            Case "sub1\sub2\a.htm"
                sb.AppendLine("<img src='..\..\images\abc.jpg'>")
        End Select
        e.WriteString(sb.ToString())
    End If
```

现在你通过下面三个网页访问HttpServer：

http://127.0.0.1/a.htm
http://127.0.0.1/sub1/a.htm
http://127.0.0.1/sub1/sub2/a.htm

都可以正常显示"d:\web\images"的图片"abc.jpg".

6.14 使用 JavaScript 文件

本节内容要使用到 JavaScript。

即使您没有掌握 JavaScript，也没有关系，因为真的很简单。

假定在以下网页中，希望输入单价、折扣和数量后，能即时计算出金额：



要完成这个任务，就需要使用 Javascript 来编写脚本了。

知识准备

下面会对本节要使用的 JavaScript 知识做一个简单的介绍。

教会你 JavaScript，并不是我们的任务， 如果你有兴趣进行进一步的了解，可以访问：http://www.w3school.com.cn/b.asp

我个人建议你花上一两天时间，通过上述网页，掌握一下 JavaScript 的基础知识。

首先是一个重要的提示：JavaScript 是区分大小写的，例如 calc 和 Calc 是两个不同的变量或函数，逻辑值是 true 和 false，写成 True 和 False，也是不行的。

1、如何引用表单元素

我们在设计表单的时候，需要给输入框指定 id 属性和 name 属性，例如：

sb.appendLine("单价: <input Type='number' name='dj' id='dj'>")

表示定义了一个输入框用于输入单价，这个输入框的 id 和 name 都是"dj"，id 和 name 属性可以设置为不同的值，除非有特殊需要，一般设置为相同的值即可。

我们可以在 js 代码中直接通过 id 属性引用这个输入框，例如：

dj.value

表示单价输入框的值，这是一种简写方式，使用起来很方便。

还有一种更为标准的写法，例如下面的代码，同样是表示单价输入框的值：

document.getElementById('dj').value

前一种方法更为简洁，但后一种方法更为标准，多数编程书籍使用的都是后一种方法，本帮助文

件两种都用。

2、网页事件

网页中所有元素都是有事件的，例如单击按钮，会触发按钮的 onclick 事件，修改输入框的内容，会触发输入框的 onchange 事件。

我们可以在设计网页的时候，指定事件触发后要执行的代码，例如：

```
sb.appendLine("单价: <input Type='number' name='dj' id='dj' onchange='calc()'>")
```

表示修改单价输入框的内容后，执行 calc 函数，注意函数名是区分大小写的，括号也是不能省略的。

3、在哪里写代码

JavaScript 代码一般放在单独的文本文件中，后缀名为 js，你可以直接用记事本编写 js 代码，我个人喜欢结合 nodepad++和 Dreamweaver 编写，前者可以让我更方便地分析代码结构，后者可以帮我检查出语法错误。

编写好的 js 代码一般放在某个子目录下，例如本帮助文件所有的 js 代码，都放在"d:\web\lib"目录下。

编写好的 js 代码，并不会自动自动生效，我们在设计网页的时候，还需要将编写好的 js 代码文件，引入到网页中，例如：

```
sb.appendline("<script src='./lib/calc.js'></script>")
```

表示将 lib 子目录下的 calc.js 文件引入到网页中。

有了上面的知识，我们就能很轻松地完成自动计算金额的任务了。

设计步骤

1、在"d:\web"目录下，建立一个子目录 lib，在这个子目录中新建一个文本文件，文件名为"calc.js"，文件内容为：

```
function calc(){
    je.value = dj.value * sl.value * (1 - zk.value);
}
```

上述的代码定义了一个名为 calc 的函数，用于根据单价(dj)、数量(sl)和折扣(zk)的值，计算出金额。

2、然后修改 HttpRequest 事件的代码，在有变动的地方，我加上了注释：

```
Dim fl As String = "d:\web\" & e.path
If filesys.FileExists(fl)
    Dim idx As Integer = fl.LastIndexOf(".")
    Dim ext As String   = fl.SubString(idx)
    Select Case ext
      Case ".jpg",".gif",".png",".bmp",".wmf",".js" '这里加上了js扩展名
        e.WriteFile(fl)
    End Select
Else
    Dim sb As New StringBuilder
    sb.appendLine("<!doctype html>")
    sb.appendLine("<html>")
    sb.appendLine("<head>")
```

```
    sb.appendLine("<meta charset='utf-8'>")
    sb.appendLine("<title>表单</title>")
    sb.appendLine("</head>")
    sb.appendLine("<body>")
    sb.appendLine("<form enctype='multipart/form-data' method='post' id='form1' name='form1'>")
    sb.appendLine("产品: <input name='cp' id='cp'><br/><br/>")
    sb.appendLine("客户: <input name='kh' id='kh'><br/><br/>")
    sb.appendLine("雇员: <input name='gy' id='gy'><br/><br/>")
    sb.appendLine("单价: <input Type='number' name='dj' id='dj' onchange='calc()'><br/><br/>") '加上事件触发
        sb.appendLine("折 扣 : <input Type='number' name='zk' id='zk' step='0.01' onchange='calc()'><br/><br/>")
    sb.appendLine("数量: <input Type='number' name='sl' id='sl' onchange='calc()'><br/><br/>")
    sb.appendLine("金额: <input Type='number' name='je' id='je' readonly><br/><br/>")
    sb.appendLine("日期: <input Type='date' name='rq' id='rq'><br/><br/>")
    sb.appendLine("<input Type='submit' name='Sumbit' id='Sumbit' value='确定'>")
    sb.appendLine("</form>")
    sb.appendline("<script src='./lib/calc.js'></script>") '引入脚本文件
    sb.appendLine("</body>")
    sb.appendLine("</html>")
    e.WriteString(sb.ToString)
End If
```

现在输入数量、单价和折扣，金额就会自动计算得出了：



6.15 使用 CSS 文件

本节内容适合已经掌握 CSS 的用户。

同样，教会你 CSS 不是我们的任务，有人干这个我们擅长很多，如果有兴趣，可以访问：http://www.w3school.com.cn/css/index.asp，通常不会超过半天时间就能基本掌握。
使用 CSS 文件和使用 js 文件的方法一样，只是 CSS 文件通常在 head 块引入，js 通常在 body 块的结束位置引入。

一个例子

1、在"d:\web"目录下，建立一个子目录 css，在这个子目录中新建一个文本文件，文件名为文件名为“test.css”，文件内容为：

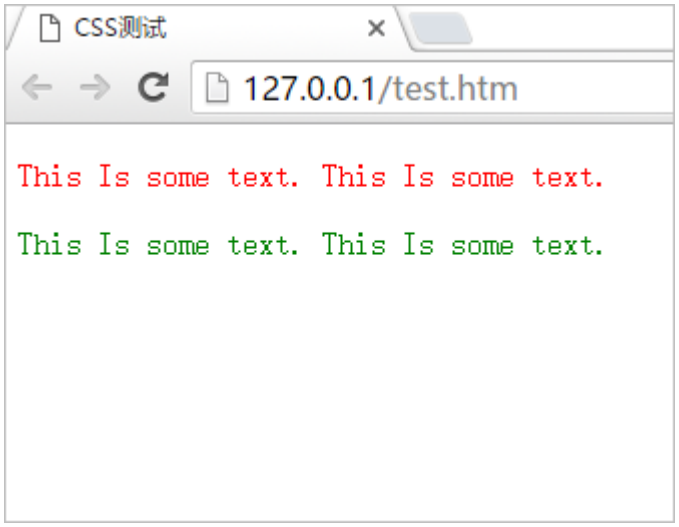
```
.rd{color:red;}
.gr{color:green;}
```

2、然后修改 HttpRequest 事件的代码，在有变化的地方，我加上了注释：

```
Dim fl As String = "d:\web\" & e.path
If filesys.FileExists(fl)
    Dim idx As Integer = fl.LastIndexOf(".")
    Dim ext As String   = fl.SubString(idx)
    Select Case ext
      Case ".jpg",".gif",".png",".bmp",".wmf",".js",".css" '这里加上了css扩展名
```

```
            e.WriteFile(fl)
        End Select
    Else
        Dim sb As New StringBuilder
        sb.AppendLine("<!doctype html>")
        sb.AppendLine("<html>")
        sb.AppendLine("<head>")
        sb.AppendLine("<meta charset='utf-8'>")
        sb.AppendLine("<title>CSS测试</title>")
        sb.AppendLine("<link rel='stylesheet' href='./css/test.css'/>") '引入css文件
        sb.AppendLine("</head>")
        sb.AppendLine("<body>")
        sb.AppendLine("<p class='rd'>This Is some text. This Is some text.</p>") '通过class属性指定所采用的样式
        sb.AppendLine("<p class='gr'>This Is some text. This Is some text.</p>")
        sb.AppendLine("</body>")
        sb.AppendLine("</html>")
        e.WriteString(sb.ToString)
    End If
```

在浏览器的显示结果为：



6.16 使用第三方库

一般用户可忽略本节内容。

本节内容适合已经掌握 JavaScript 和 CSS，希望使用第三方 JavaScript 库的用户。
使用第三方的 Javascript 库和使用本地库并没有差别，下面是一个使用 jQuery 库的例子。
1、在"d:\web"目录下，建立一个子目录 lib，在这个子目录中新建一个文本文件，文件名为"test.js"，文件内容为：

```
$(document).ready(function(){
    $('button').click(function(){
        $('div').animate({left:'250px'});
    });
});
```

2、HttpRequest 事件代码设置为，在有变化的地方，我加上了注释：

```
Dim fl As String = "d:\web\" & e.path
If filesys.FileExists(fl)
    Dim idx As Integer = fl.LastIndexOf(".")
    Dim ext As String   = fl.SubString(idx)
    Select Case ext
       Case ".jpg",".gif",".png",".bmp",".wmf",".js",".css"
          e.WriteFile(fl)
    End Select
Else
    Dim sb As New StringBuilder
    sb.AppendLine("<!doctype html>")
    sb.AppendLine("<html>")
    sb.AppendLine("<head>")
    sb.AppendLine("<meta charset='utf-8'>")
    sb.AppendLine("<title>jQuery测试</title>")
                                                    sb.AppendLine("<script
src='http://cdn.static.runoob.com/libs/jquery/1.10.2/jquery.min.js'></script>") '引入jQuery库
    sb.AppendLine("<script src='./lib/test.js'></script>") '引入本地库
    sb.AppendLine("</head>")
    sb.AppendLine("<body>")
    sb.AppendLine("<button>开始动画</button><br/><br/>")
    sb.AppendLine("<div style='background:#98bf21;height:100px;width:100px;position:absolute;'>")
    sb.AppendLine("</body>")
    sb.AppendLine("</html>")
    e.WriteString(sb.ToString)
End If
```

在浏览器显示的结果如下，单击“开始动画”按钮，绿色方框会移动：



6.17 使用设计好的网页

前面的例子，我们都是用 Foxtable 动态合成网页。

动态合成的好处是：网页是“活”的，同样的页面，根据不同的用户、数据和业务逻辑，显示内容和结构可以完全不同。

我们也可以用第三方网页设计工具事先设计好网页，由 Foxtable 负责将网页发送给用户浏览器，代码很简单。
例如你将设计好的网页，放在"d:\web"目录及其子目录下，包括图片、js 文件、css 文件等等。

然后将 HttpRequest 事件代码设置为：

```
Dim fl As String = "d:\web\" & e.path
If filesys.FileExists(fl)
    Dim idx As Integer = fl.LastIndexOf(".")
    Dim ext As String   = fl.SubString(idx)
    Select Case ext
      Case ".jpg",".gif",".png",".bmp",".wmf",".js",".css" ,".html",".htm"
        e.WriteFile(fl)
    End Select
End If
```

不到 10 行代码，这可能是史上最简单的架设 http 服务的方法。
不，这不是最简单的，很快你会看到真正简单的 http 服务架设方法，只需三行代码。

静态与动态

如果网页需要和动态的数据以及业务逻辑相结合，就用代码动态生成；如果网页基本不变化，就事先设计好网页，也就是静态网页，以提高效率。

同一个系统，可以同时有静态网页和动态网页，除非你明确告诉客户，否则客户并不会感受到这

些页面什么不同。
例如：

```
Select Case e.Path
    Case ""
        e.WriteString("这是根目录下的默认页面,是动态生成的")
    Case "order.htm"
        e.WriteString("这是根目录下的 Order.htm,这个文件并不存在,是自动生成的")
    Case "sale"
        e.WriteString("这是根目录的 sale 子目录下的默认页面,是动态生成的")
    Case "sale\add.htm"
        e.WriteString("这是根目录的 sale 子目录下的 add.htm,这个文件并不存在,是自动生成的")
    Case Else
        Dim fl As String = "d:\web\" & e.Path
        If filesys.FileExists(fl)
            Dim idx As Integer = fl.LastIndexOf(".")
            Dim ext As String   = fl.SubString(idx)
            Select Case ext
                Case ".jpg",".gif",".png",".bmp",".wmf",".js",".css" ,".html",".htm"
                    e.WriteFile(fl)
            End Select
        Else
            e.WriteString("好奇怪,你访问的页面不存在!")
        End If
End Select
```

有个小问题需要注意一下，e.path 的路径分割符是\"，不是"/"。

实际上，很难严格区分动态网页和动态网页，因为就是动态生成的网页，也需要使用不少静态的文件，例如图片、js、css 等等。

6.18 网页出现乱码啦

我用古老的 FrontPage 中文版设计了一个网页，这个网页很简单，就一行文字：
我喜欢用 Foxtable 开发管理软件
我将这个网页复制到"d:\web"目录下，通过 HttpRequest 将这个网页发送到客户端的浏览器上，意外出现了，中文全部变成了乱码：



如果我们在本机双击这个文件，浏览器是能正常显示中文的，说明网页本身是没有问题的：



为了查明原因，我用 Nodepad++（也可以用记事本）打开这个网页，看到的源代码是：

```
1  <html>
2
3  <head>
4  <meta http-equiv="Content-Language" content="zh-cn">
5  <meta http-equiv="Content-Type" content="text/html; charset=gb2312">
6  <title>新建文档</title>
7  </head>
8
9  <body>
10
11 <p>我喜欢用Foxtable开发管理软件</p>
12
13 </body>
14
15 </html>
```

注意这里

现在出现中文乱码的原因清楚了，因为 HttpRequest 并不会检查要发送网页的编码格式，默认一律采用 utf-8 编码格式读取网页内容，然后发送到用户浏览器。

为解决这个问题，我们可以修改 HttpRequest 事件代码：

```
Dim fl As String = "d:\web\" & e.Path
If filesys.FileExists(fl)
    Dim idx As Integer = fl.LastIndexOf(".")
    Dim ext As String   = fl.SubString(idx)
    Select Case ext
                                                                    Case
".jpg",".gif",".png",".bmp",".wmf",".js",".css" ,".html",".htm",".doc",".docx",".xls",".xlsx",".pdf"
        e.ResponseEncoding = "gb2312" '设置网页编码为gb2312
        e.WriteFile(fl)
    End Select
Else
    e.WriteString("好奇怪,你访问的页面不存在!")
End If
```

现在浏览器能正常显示中文了：



HttpRequest的e参数ResponseEncoding，用于设置向客户端发送文本性质的内容时，所采用的编码格式。

修改网页编码为utf-8格式

你也可以不修改HttpRequest事件代码，使用utf-8编码保存网页即可，包括js和css文件， 我们都建议采用utf-8编码。
新的网页设计工具，多数默认编码就是utf-8格式。
如果已经有其他编码格式的网页，需要改为utf-8编码，以上面的文件为例，步骤为：

1、用记事本打开这个网页，找到charset=gb2312，改为charset=utf-8
2、不要直接保存，在文件菜单单击"另存为"，在另存窗口选择编码格式为"utf-8"，然后单击"确定"按钮保存即可。

6.19 接收表单数据

多数时候，用户是通过表单输入数据的，表单类似于 Foxtable 的窗口。

定义表单

表单用<form>标签开始，用</form>标签结束，中间是各种输入元素。
下面是一个表单的定义，表单名为form1，数据提交方式为post，数据接收页面是accept.htm:

```
<form action='accept.htm' enctype='multipart/form-data' method='post' id='form1' name='form1'>

</form>
```

提示：表单的action属性是一个网页地址，用户输入完成后单击提交按钮，会将表单数据提交到这个网页。

Values属性

HttpRequest事件e参数有个Values属性，这是一个字典，包括所有用户提交的数据，键为输入元素的name属性，值就是输入元素的值。

一个例子

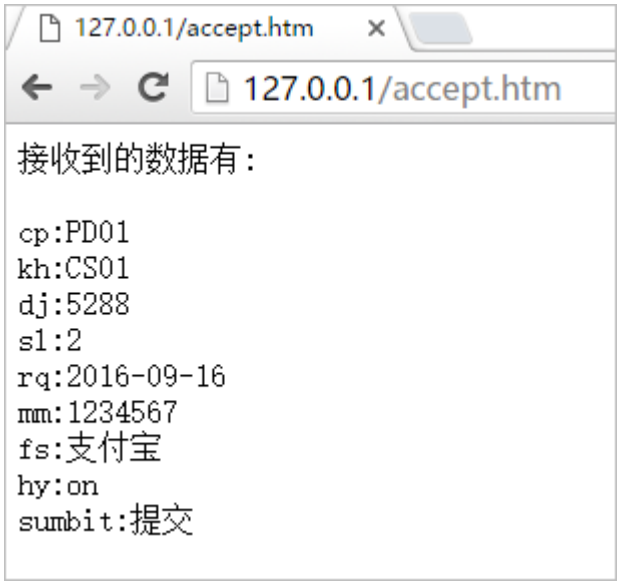
我们下面用一个例子看看 Foxtable 是如何定义表单和接收表单数据的，将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "input.htm"
        Dim sb As New StringBuilder
          sb.AppendLine("<form action='accept.htm' enctype='multipart/form-data' method='post' id='form1' name='form1'>")
        sb.AppendLine("产品: <input name='cp' id='cp'><br/><br/>")
        sb.AppendLine("客户: <input name='kh' id='kh'><br/><br/>")
        sb.AppendLine("单价: <input type='number' name='dj' id='dj'><br/><br/>")
        sb.AppendLine("数量: <input type='number' name='sl' id='sl'><br/><br/>")
        sb.AppendLine("日期: <input type='date' name='rq' id='rq'><br/><br/>")
        sb.AppendLine("密码: <input type='password' name='mm' id='mm'><br/><br/>")
        sb.AppendLine("支付方式: <br/>")
        sb.AppendLine("<input type='radio' name='fs' id='fs1' value = '支付宝' checked>支付宝") '注意同一组的radio，name属性相同，id属性不同
        sb.AppendLine("<input type='radio' name='fs' id='fs2' value = '微信'>微信")
        sb.AppendLine("<input type='radio' name='fs' id='fs3' value = '微信'>网银<br/><br/>")
        sb.AppendLine("会员: <input type='checkbox' name='hy' id='hy'><br/><br/>")
        sb.AppendLine("<input type='submit' name='sumbit' id='sumbit' value='提交'>")
        sb.AppendLine("<input type='reset' name='reset' id='reset' value='重置'>")
         sb.AppendLine("<input type='button' name='foxtble' id='foxtable' value='Foxtable 主 页 ' onclick='location=""http://www.foxtable.com""'>")
        sb.AppendLine("</form>")
        e.WriteString(sb.ToString)
    Case "accept.htm"
        Dim sb As New StringBuilder
        sb.AppendLine("接收到的数据有:<br/><br/>")
        For Each key As String In e.Values.Keys
          sb.AppendLine(key & ":" & e.Values(key) & "<br/>")
        Next
        e.WriteString(sb.ToString)
End Select
```

这个代码生成了两个页面，第一个页面input.htm是一个表单，用于输入数据，这个表单展示了最常用的输入类型：



用户单击提交按钮后，会将输入的数据提交到第二个页面accept.htm，该页面将收到的数据回写到客户端浏览器：



建议你以后开发系统的时候，都先用上面的方法，实际测试一下表单所提交的数据内容和格式。

需要注意的是，只有已经输入数据的输入框，以及已经勾选的radio和checbox，才会被提交到服务器，也就是说：

1、没有数据的空白输入框，不会提交到服务器。
2、没有勾选的radio和checkbox，也不会提交到服务器。

6.20 在表单存储标记数据

我们在生成表单的时候，可能需要在表单插入一些标记数据。

例如你需要根据一个现有订单，来生成一个表单，那么就需要在生成的表单中存储主键值，这样用户编辑完成，将编辑结果提交到后台时，Foxtable 才能知道用户编辑的是哪一个订单。
我们可以增加一个隐藏的文本框，将这个文本框的值设置为主键，例如：

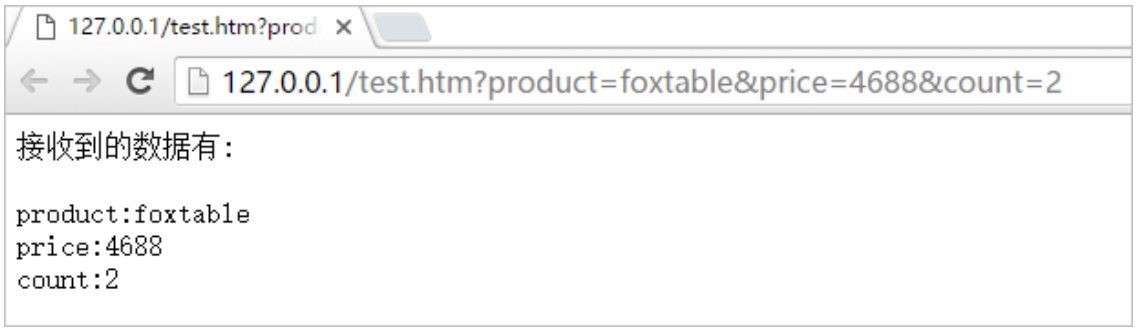
<input name='identify' id='identify' value='1' **hidden**>
value 属性指定了这个文本框的默认值为 1，加上 hidden 属性使得这个文本框在表单中处于隐藏状态，当用户提交表单数据到后台时，这个文本框的值会和其他输入框的值一起提交到后台 ，Foxtable 就知道用户编辑的是主键为 1 的订单。
我们通常不会专门介绍 HTML 的知识，但是这个方法对于我们今后设计应用系统很重要，所以用专门的一小节介绍一下。

6.21 另一种数据提交方式

除了前面介绍的通过表单提交数据，我们还可以通过 URL 地址提交数据，格式为：
http://网页地址/?键 1=值 1&键 2=值 2&键 3=值 3
示例
将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "test.htm"
      Dim sb As New StringBuilder
      sb.AppendLine("接收到的数据有:<br/><br/>")
      For Each key As String In e.Values.Keys
        sb.AppendLine(key & ":" & e.Values(key) & "<br/>")
      Next
      e.WriteString(sb.ToString)
End Select
```

然后在浏览器中输入地址：
http://127.0.0.01/test.htm?product=foxtable&price=4688&count=2
可以看到 Foxtable 已经正确提取出 URL 网址中包括的数据：



这种数据提交方式，通常称为 get 方式，通过表单提交的方式，通常称为 post 方式。

当然你可以将表单的 method 属性设置为 get,这样表单也可以采用 get 方式提交数据，但是没有必要，因为 post 方式更安全、更方便，一次可以提交更多的数据。

但是 get 方式也有自己的优势，就是轻量级，通常用于提交一些事先约定好的、不需要用户输入的数据。

混合 Get 和 Post 方式

其实 Get 和 Post 两种方式是可以混合使用的，例如我们将 HttpRequest 事件代码设置为：

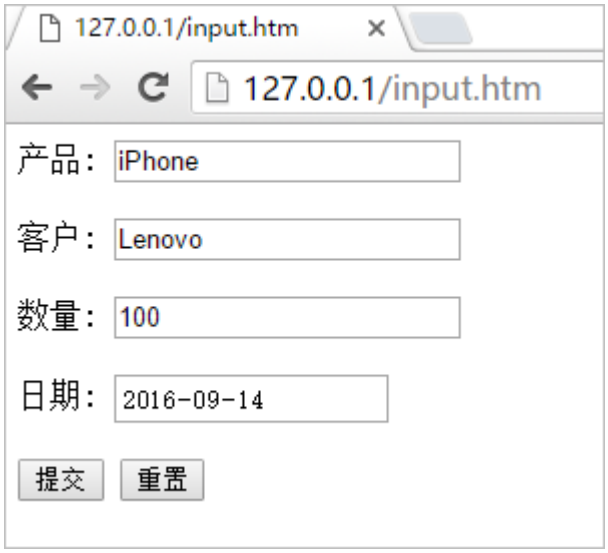
```
Select Case e.Path
    Case "input.htm"
        Dim sb As New StringBuilder
            sb.AppendLine("<form enctype='multipart/form-data' action='accept.htm?id=1&page=10'
method='post' id='form1' name='form1'>")
        sb.AppendLine("产品: <input name='cp' id='cp'><br/><br/>")
        sb.AppendLine("客户: <input name='kh' id='kh'><br/><br/>")
        sb.AppendLine("数量: <input type='number' name='sl' id='sl'><br/><br/>")
        sb.AppendLine("日期: <input type='date' name='rq' id='rq'><br/><br/>")
        sb.AppendLine("<input type='submit' name='sumbit' id='sumbit' value='提交'>")
        sb.AppendLine("<input type='reset' name='reset' id='reset' value='重置'>")
         sb.AppendLine("</form>")
        e.WriteString(sb.ToString)
    Case "accept.htm"
        Dim sb As New StringBuilder
        sb.AppendLine("接收到的数据有:<br/><br/>")
        For Each key As String In e.Values.Keys
            sb.AppendLine(key & ":" & e.Values(key) & "<br/>")
        Next
        e.WriteString(sb.ToString)
End Select
```

上述代码中，我们定义的表单的代码为：

<form action='accept.htm?id=1&page=10' method='post' name='form1'>

用户输入完成单击提交按钮后，会将输入结果提交到地址"accept.htm?id=1&page=10"，这个地址包括 id 和 page 两个值。

所以当用户按下图所示输入数据：



然后单击提交按钮，提交到后台后，显示的内容会包括 url 地址中的内容，以及用户输入的内容：



6.22 Values、GetValues 和 PostValues

前面已经说过，不管是通过 GET 方式提交，还是通过 POST 方式提交，客户端提交的所有数据都包括在 Values 字典中。

HttpReqquest 事件还有两个字典属性，他们是 Values 的子集：

GetValues： 字典，仅包括通过 GET 方式提交的数据。

PostValues：字典，仅包括通过 POST 方式提交的数据。

例如我们将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "input.htm"
        Dim sb As New StringBuilder
```

```
            sb.AppendLine("<form enctype='multipart/form-data' action='accept.htm?id=1&page=10' method='post' id='form1' name='form1'>")
        sb.AppendLine("产品: <input name='cp' id='cp'><br/><br/>")
        sb.AppendLine("客户: <input name='kh' id='kh'><br/><br/>")
        sb.AppendLine("数量: <input type='number' name='sl' id='sl'><br/><br/>")
        sb.AppendLine("日期: <input type='date' name='rq' id='rq'><br/><br/>")
        sb.AppendLine("<input type='submit' name='sumbit' id='sumbit' value='提交'>")
        sb.AppendLine("<input type='reset' name='reset' id='reset' value='重置'>")
         sb.AppendLine("</form>")
        e.WriteString(sb.ToString)
    Case "accept.htm"
        Dim sb As New StringBuilder
        sb.AppendLine("通过 GET 方式提交的数据:<br/><br/>")
        For Each key As String In e.GetValues.Keys
            sb.AppendLine(key & ":" & e.GetValues(key) & "<br/>")
        Next
        sb.AppendLine("<br/><br/>通过 POST 方式提交的数据:<br/><br/>")
        For Each key As String In e.PostValues.Keys
            sb.AppendLine(key & ":" & e.PostValues(key) & "<br/>")
        Next
        e.WriteString(sb.ToString)
End Select
```

上述代码中，我们定义的表单的代码为：

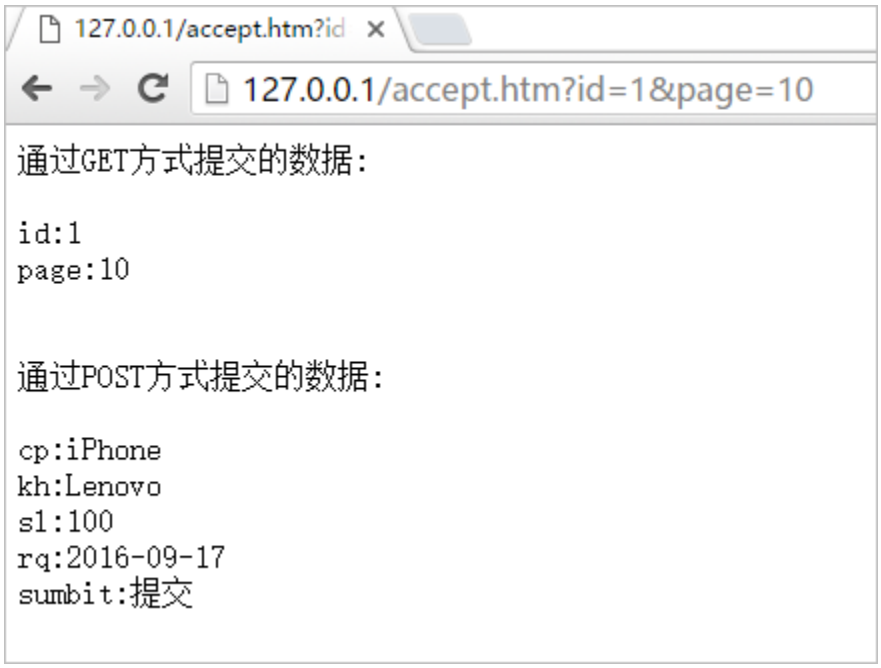
<form enctype='multipart/form-data' action='accept.htm?id=1&page=10' method='post' id='form1' name='form1'>

用户输入完成单击提交按钮后，会将输入结果提交到地址"accept.htm?id=1&page=10"，这个地址包括 id 和 page 两个值。

当用户按下图所示输入数据：



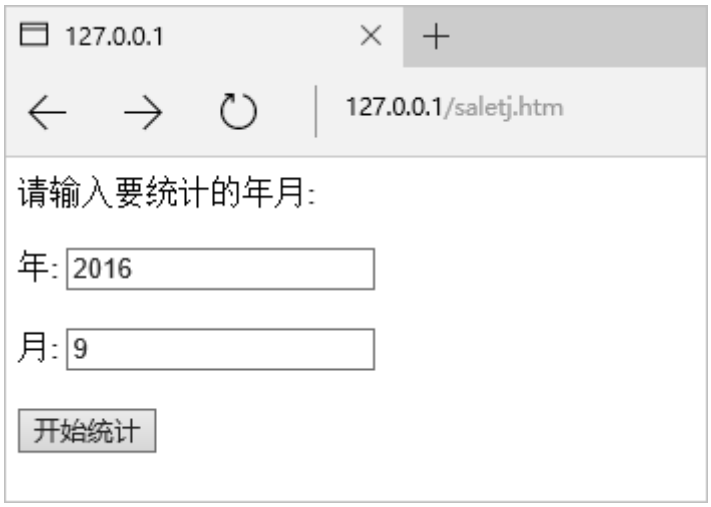
然后单击提交按钮，提交到后台后，可以看到分别显示 GET 和 POST 方式提交的数据：



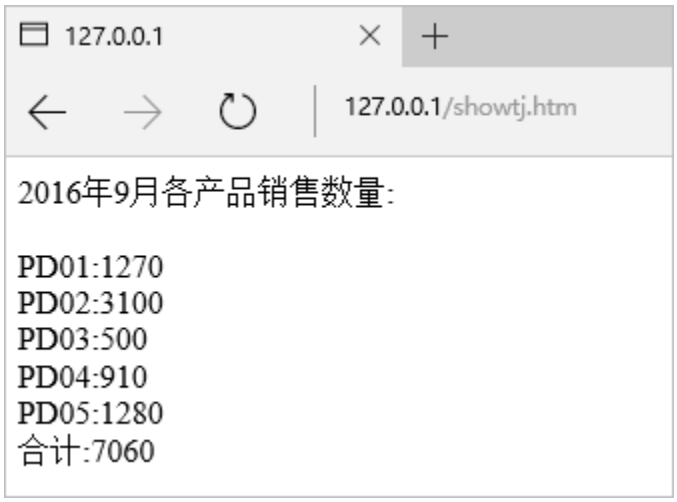
6.23 简单的数据统计

本节任务

我们希望让用户通过下面的网页输入年月：



单击“开始统计”，由 Foxtable 统计出该月各产品的销售数量，并显示在用户的浏览器上：



知识准备

用户是通过表单(form)输入数据的，这是我们本节要用到的一个表单：

```
1 <form action='showtj.htm' enctype='multipart/form-data' method='post' id='form1' name='form1'>
2 请输入要统计的年月:</br></br>
3 年: <input type='number' name='year' id='year' min='1999' max='2018'><br/><br/>
4 月: <input type='number' name='month' id='month' min='1' max='12'><br/><br/>
5 <input Type='submit' name='Sumbit' id='Sumbit' value='开始统计'>
6 </form>
```

第 1 行代码为：

<form action='showtj.htm' enctype='multipart/form-data' method='post' id='form1' name='form1'>

表示用户单击确定按钮后，将结果提交到后台的 showtj.htm 网页处理，这个表单的名称为 form1，数据提交方式为 post

再看第 3 行代码：

年: <input type='number' name='year' id='year' min='2000' max='2018'>

表示这是一个数值输入框，输入值的范围在 2000 到 2018 之间，这个输入框的名称（name）为"year"，我们必须给输入框指定 name 属性，因为后台的 Foxtable 要通过 name 属性取得输入值。

服务端的 HttpRequest 事件通过 e 参数 PostValues 获取用户输入的值，这是一个字典，键为输入框的 name 属性，值就是用户所输入的值，例如在 HttpRequest 事件 获取用户通过上述表单输入的年份的代码为：

```
Dim y As Integer = 0
'将输入的年转换为整数，因为有的浏览器数值输入框也能输入字符!
Integer.TryParse(e.PostValues("year"), y)
If y = 0 Then
    e.WriteString("请输入年!")
End If
```

完整代码

下面是HttpRequest事件的完整代码，代码不长，却完成了生成输入页面、获取用户输入 内容、统计数据，以及呈现统计结果等四大任务：

```
Select Case e.Path
    Case "saletj.htm"
        Dim sb As New StringBuilder
          sb.AppendLine("<form enctype='multipart/form-data' action='showtj.htm' method='post' id='form1' name='form1'>")
        sb.AppendLine("请输入要统计的年月:</br></br>")
          sb.AppendLine("  年  :  <input type='number' name='year' id='year' min='1999' max='2018'><br/><br/>")
          sb.AppendLine("  月  :  <input type='number' name='month' id='month' min='1' max='12'><br/><br/>")
        sb.AppendLine("<input Type='submit' name='Sumbit' id='Sumbit' value='开始统计'>")
        sb.AppendLine("</form>")
        e.WriteString(sb.ToString)
    Case "showtj.htm"
        Dim y As Integer = 0
        Dim m As Integer = 0
        Integer.TryParse(e.PostValues("year"), y) '获取用户输入的 年
        Integer.TryParse(e.PostValues("month"), m) '获取用户输入的月
        If y = 0 OrElse m = 0 Then
          e.WriteString("请输入正确的年月!")
        Else
          '根据输入的年月,统计各产品的销售数量
          Dim g As New SQLGroupTableBuilder("统计表1", "订单") '这里使用后台统计,如果数据已经全部加载可以直接用GroupTableBuilder
          'g.ConnectionName = "数据源" '外部数据表的话要指定数据源名称
          g.Groups.AddDef("产品")
          g.Totals.AddDef("数量")
          g.VerticalTotal = True
          g.Filter = "Year(日期) = " & y & " And Month(日期) = " & m  '后台统计才可以使用这种表达式的哦
          Dim dt As DataTable = g.Build(False)
          '将统计结果输出到用户浏览器
          Dim sb As New StringBuilder
          sb.AppendLine(y & "年" & m & "月各产品销售数量: <br/><br/>")
          For Each dr As DataRow In dt.DataRows
            sb.AppendLine(dr("产品") & ":" & dr("数量") & "<br/>")
          Next
          e.WriteString(sb.ToString)
        End If
End Select
```

6.24 可能会出现性能问题

一般用户可以忽略本节内容。

上一节我们完成了一个简单的数据交互任务： 让用户通过网页输入年月，后台根据用户输入的年月，统计出该月各产品的销售数量，并显示在用户浏览器上。

不管是输入页面"saletj.htm"，还是显示统计结果的页面"showtj.htm"，其实都是不存在的，是后台动态生成的，是虚的。

动态生成网页的优势是灵活，但是也有一个劣势，就是性能。

以上一节的例子为例，每次统计的耗时和数据量是成正比的，如果用户每次访问都重新进行一次统计，且用户量和数据量都比较大，那么就必须考虑服务器的负担问题了。

要减轻服务器的负担，我们可以这样考虑：

1、本月的销售数据在不断变化中，适合每次访问都重新统计，而之前月份的销售数据已经固定，没有必要重复统计。

2、所以我们可以将之前月份的统计结果保存为一个网页，这样以后访问时，无需再次进行统计 ，直接发送已经生成的网页即可。

3、至于本月销售数据，当然每次都重新统计是最好的，但实际上用户统计本月数据的概率是最大的，每次重新统计，也可能导致服务器负担过重。

4、所以对于本月统计，我们一样可以将统计结果保存为一个网页，用户每次访问时，判断这个网页的生成时间，如果没有超过 1 小时，就直接发送这个网页，否则重新进行统计。

5、为方便管理，可以在"d:\web"目录下，建立一个子目录"temp"，用于存放这些临时生成的网页。

这是修改后的 HttpRequest 事件的代码，要正常运行这段代码，请在"d:\web"目录下，建立一个子目录"temp"：

```
Select Case e.Path
    Case "saletj.htm"
        Dim sb As New StringBuilder
        sb.AppendLine("<form enctype='multipart/form-data' action='showtj.htm' method='post' id='form1' name='form1'>")
        sb.AppendLine("请输入要统计的年月:</br></br>")
        sb.AppendLine("年 : <input type='number' name='year' id='year' min='1999' max='2018'><br/><br/>")
        sb.AppendLine("月 : <input type='number' name='month' id='month' min='1' max='12'><br/><br/>")
        sb.AppendLine("<input Type='submit' name='Sumbit' id='Sumbit' value='开始统计'>")
        sb.AppendLine("</form>")
        e.WriteString(sb.ToString)
    Case "showtj.htm"
        Dim y As Integer = 0
        Dim m As Integer = 0
        Integer.TryParse(e.PostValues("year"), y)'获取用户输入的年
        Integer.TryParse(e.PostValues("month"), m) '获取用户输入的月
        If y = 0 OrElse m = 0 Then
            e.WriteString("请输入正确的年月!")
```

```
    Else
        Dim fl As String = "d:\web\temp\" & y & m & ".htm" '合成文件名
        If FileSys.FileExists(fl) Then '如果文件存在
            Dim ifo As new FileInfo(fl)
            '如果不是本月,或者文件最近一次修改时间在1个小时内,则发送文件后返回
            If y <> Date.Today.Year OrElse m <> Date.Today.Month OrElse (Date.Now - Ifo.LastWriteTime).TotalHours < 1
                e.WriteFile(fl)
                Return
            End If
        End If
        '根据输入的年月,统计各产品的销售数量
        Dim g As New SQLGroupTableBuilder("统计表1", "订单") '这里使用后台统计,如果数据已经全部加载可以直接用GroupTableBuilder
        'g.ConnectionName = "数据源" '外部数据表的话要指定数据源名称
        g.Groups.AddDef("产品")
        g.Totals.AddDef("数量")
        g.VerticalTotal = True
        g.Filter = "Year(日期) = " & y & " And Month(日期) = " & m  '后台统计才可以使用这种表达式的哦
        Dim dt As DataTable = g.Build(False)
        '统计数据,将结果保存为网页,然后发送这个网页
        Dim sb As New StringBuilder
        sb.AppendLine(y & "年" & m & "月各产品销售数量: <br/><br/>")
        For Each dr As DataRow In dt.DataRows
            sb.AppendLine(dr("产品") & ":" & dr("数量") & "<br/>")
        Next
        sb.AppendLine("<br/>")
        sb.AppendLine("统计时间:" & Date.Now) '在网页显示统计时间,提醒用户这可能不是最新的结果
        FileSys.WriteAllText(fl, sb.ToString, False, Encoding.UTF8) '记得用UTF8格式保存网页
        e.WriteFile(fl) '发送保存的网页
    End If
End Select
```

如果之前月份的销售数据也有调整的可能，导致之前月份的统计结果也会变化，只是变化的概率和频率不高，那么你可以调整代码，检测之前月份统计文件的创建时间，如果超过1天 （具体时长根据需要调整），就重新统计。

6.25 一个页面完成一个任务

上一节的统计任务，我们使用了两个页面，页面"saletj.htm"用于输入统计年月，页面"showtj.htm"用于显示统计结果。

如果你愿意的话，也可以只用一个页面"saletj.htm"来完成这个统计任务，从代码维护角度而言，我更愿意用一个页面处理一个任务。

知识准备

我们已经知道，HttpRequest 事件有个 e 参数 PostValues，用于获取用户通过表单提交的数据。

我们可以用一个隐藏的输入框，在这个输入框填入约定的数据，HttpRequest 根据这个输入框的值，来执行相应的操作。

例如：

<input name='tj' id='tj' value='按年月统计' hidden>"

这里创建了一个名为"tj"的输入框，默认值是"按年月统计"，这个输入框加上了 hidden 属性，所以在页面中并不会显示，用户不会感觉有什么变化。

在 HttpRequest 通过下面的代码，判断如果提交的页面名为 tj 的输入框，且他的值是"按年月统计"，那么就进行统计并显示统计结果，否则生成年月输入页面：

```
Select Case e.Path
    Case "saletj.htm"
        If e.PostValues.ContainsKey("tj") AndAlso e.PostValues("tj") = "按年月统计" Then
            '统计数据，并显示统计结果
        Else
            '生成年月录入界面
        End If
End Select
```

我们也可以用更简单的方式来区分两个页面：

```
Select Case e.Path
    Case "saletj.htm"
        If e.PostValues.Count > 0 Then
            '统计数据，并显示统计结果
        Else
            '生成年月录入界面
        End If
End Select
```

原理很简单，如果是直接访问 saletj.htm，那么 PostValues 为空，其 Count 属性等于 0，如果是通过表单的“开始统计”按钮访问 saletj.htm，那么 PostValues 就会包括用户输入的值，其 Count 属性就会大于 0。

最后别忘记修改表单定义语句，该语句之前是：

<form action='showtj.htm' enctype='multipart/form-data' method='post' id='form1' name='form1'>

我们需要改为：

<form action='saletj.htm' enctype='multipart/form-data' method='post' id='form1' name='form1'>

完整的代码

下面是HttpRequest事件的完整代码，只使用一个页面完成和上一节完全相同的任务：

```
Select Case e.Path
    Case "saletj.htm"
        If e.PostValues.ContainsKey("tj") AndAlso e.PostValues("tj") = "按年月统计" Then
            Dim y As Integer = 0
            Dim m As Integer = 0
            Integer.TryParse(e.PostValues("year"), y) '获取用户输入的年
            Integer.TryParse(e.PostValues("month"), m) '获取用户输入的月
            If y = 0 OrElse m = 0 Then
                e.WriteString("请输入正确的年月!")
            Else
                Dim fl As String = "d:\web\temp\" & y & m & ".htm" '合成文件名
                If FileSys.FileExists(fl) Then '如果文件存在
                    Dim ifo As new FileInfo(fl)
                    '如果不是本月,或者文件创建时间在1个小时内,则发送文件后返回
                     If y <> Date.Today.Year OrElse m <> Date.Today.Month OrElse (Date.Now -
Ifo.LastWriteTime).TotalHours < 1
                        e.WriteFile(fl)
                        Return
                    End If
                End If
                '根据输入的年月,统计各产品的销售数量
                Dim g As New SQLGroupTableBuilder("统计表1", "订单") '这里使用后台统计,如果数据已经
全部加载可以直接用GroupTableBuilder
                'g.ConnectionName = "数据源" '外部数据表的话要指定数据源名称
                g.Groups.AddDef("产品")
                g.Totals.AddDef("数量")
                g.VerticalTotal = True
                g.Filter = "Year(日期) = " & y & " And Month(日期) = " & m  '后台统计才可以使用这种表达
式的哦
                Dim dt As DataTable = g.Build(False)
                '统计数据,将结果保存为网页,然后发送这个网页
                Dim sb As New StringBuilder
                sb.AppendLine(y & "年" & m & "月各产品销售数量: <br/><br/>")
                For Each dr As DataRow In dt.DataRows
                    sb.AppendLine(dr("产品") & ":" & dr("数量") & "<br/>")
                Next
                sb.AppendLine("<br/>")
                 sb.AppendLine("统计时间:" & Date.Now) '在网页显示统计时间,提醒用户这可能
不是最新的结果
                FileSys.WriteAllText(fl, sb.ToString, False, Encoding.UTF8) '记得用UTF8格式保存网页
                e.WriteFile(fl) '发送保存的网页
            End If
        Else
            Dim sb As New StringBuilder
```

```
            sb.AppendLine("<form enctype='multipart/form-data' action='saletj.htm' method='post' id='form1' name='form1'>")
            sb.AppendLine("请输入要统计的年月:</br></br>")
            sb.AppendLine("<input name='tj' id='tj' value='按年月统计' hidden>") '这个输入框不会显示，仅用于标记
            sb.AppendLine("年 : <input type='number' name='year' id='year' min='1999' max='2018'><br/><br/>")
            sb.AppendLine("月 : <input type='number' name='month' id='month' min='1' max='12'><br/><br/>")
            sb.AppendLine("<input Type='submit' name='Sumbit' id='Sumbit' value='开始统计'>")
            sb.AppendLine("</form>")
            e.WriteString(sb.ToString)
        End If
End Select
```

6.26 简化 HttpRequest 事件代码

到上一节位置，我们的代码已经有点长了，这只是一个页面而已哦。
随着页面的增加，代码会十倍甚至百倍成长，而 Foxtable 的代码编辑器并不适合编辑超长的代码，所以最好使用自定义函数，一个页面做成一个函数，方便管理。
关于自定义函数，大家可以参考：自定义函数

用上一节的任务作为例子，我们采用自定义函数来实现：

1、定义函数

首先定义一个名为 saletj 的自定义函数，代码为：

```
Dim e As RequestEventArgs = args(0)
If e.PostValues.ContainsKey("tj") AndAlso e.PostValues("tj") = "按年月统计" Then
    Dim y As Integer = 0
    Dim m As Integer = 0
    Integer.TryParse(e.PostValues("year"), y) '获取用户输入的年
    Integer.TryParse(e.PostValues("month"), m) '获取用户输入的月
    If y = 0 OrElse m = 0 Then
        e.WriteString("请输入正确的年月!")
    Else
        Dim fl As String = "d:\web\temp\" & y & m & ".htm" '合成文件名
        If FileSys.FileExists(fl) Then '如果文件存在
            Dim ifo As new FileInfo(fl)
            '如果不是本月,或者文件创建时间在1个小时内,则发送文件后返回
            If y <> Date.Today.Year OrElse m <> Date.Today.Month OrElse (Date.Now - Ifo.LastWriteTime).TotalHours < 1
                e.WriteFile(fl)
                Return ""
```

```
        End If
    End If
    '根据输入的年月,统计各产品的销售数量
    Dim g As New SQLGroupTableBuilder("统计表1", "订单") '这里使用后台统计,如果数据已经全部加载可以直接用GroupTableBuilder
    'g.ConnectionName = "数据源" '外部数据表的话要指定数据源名称
    g.Groups.AddDef("产品")
    g.Totals.AddDef("数量")
    g.VerticalTotal = True
    g.Filter = "Year(日期) = " & y & " And Month(日期) = " & m  '后台统计才可以使用这种表达式的哦
    Dim dt As DataTable = g.Build(False)
    '统计数据,将结果保存为网页,然后发送这个网页
    Dim sb As New StringBuilder
    sb.AppendLine(y & "年" & m & "月各产品销售数量: <br/><br/>")
    For Each dr As DataRow In dt.DataRows
      sb.AppendLine(dr("产品") & ":" & dr("数量") & "<br/>")
    Next
    sb.AppendLine("<br/>")
    sb.AppendLine("统计时间:" & Date.Now) '在网页显示统计时间,提醒用户这可能不是最新的结果
    FileSys.WriteAllText(fl, sb.ToString, False, Encoding.UTF8) '记得用UTF8格式保存网页
    e.WriteFile(fl) '发送保存的网页
  End If
Else
  Dim sb As New StringBuilder
  sb.AppendLine("<form enctype='multipart/form-data' action='saletj.htm' method='post' id='form1' name='form1'>")
  sb.AppendLine("请输入要统计的年月:</br></br>")
  sb.AppendLine("<input name='tj' id='tj' value='按年月统计' hidden>")
    sb.AppendLine("年 : <input type='number' name='year' id='year' min='1999' max='2018'><br/><br/>")
    sb.AppendLine("月 : <input type='number' name='month' id='month' min='1' max='12'><br/><br/>")
  sb.AppendLine("<input Type='submit' name='Sumbit' id='Sumbit' value='开始统计'>")
  sb.AppendLine("</form>")
  e.WriteString(sb.ToString)
End If
```

上面的代码和上一节相比， 变化非常小，只有两处，首先在第一行位置插入了一行代码：

```
Dim e As RequestEventArgs = args(0)
```

RequestEventArgs是HttpRequest事件的e参数类型。

其次是将代码中的

```
Return
```

改为了

```
Return ""
```

2、修改HttpRequest事件代码

接下来将HttpRequest事件的代码改为：

```
Select Case e.Path
    Case "saletj.htm"
        Functions.Execute("saletj",e)
End Select
```

由于主要的工作已经转移到自定义函数中，所以HttpRequest变得很简洁。

为便于讲述，接下来的例子并不会采用自定义函数，但是实际开发的时候，建议大一点的系统，都应该采用自定义函数，如果所有代码都写在HttpRequest事件中，你将痛苦不堪。

6.27 通用 HttpRqeust 事件头

建议实际开发的时候，建议在 HttpRqeust 事件的开始位置都加上一段代码，用于发送已经存在的常见文件：

```
'通用事件头,用于发送已经存在的常见文件
Dim fl As String = "d:\web\" & e.path
If filesys.FileExists(fl)
    Dim idx As Integer = fl.LastIndexOf(".")
    Dim ext As String   = fl.SubString(idx)
    Select Case ext
      Case ".jpg",".gif",".png",".bmp",".wmf",".js",".css" ,".html",".htm",".zip",".rar"
        e.WriteFile(fl)
        Return '这里必须返回
    End Select
End If
'以下是动态生成网页的代码
Select Case e.Path
    Case "addnew.htm"
      '生成新增页面
```

```
    Case "tongji.htm"
        '生成统计页面
    Case "order.htm"
        '生成订购页面
    Case Else
      e.WriteString("糟糕,文件未找到！")
End Select
```

为了让大家专注与每一节要讲述的问题，帮助文件中很多例子是不会加上事件头的。

有的时候，文件即使已经存在，也可能需要重新生成，参考：可能出现的性能问题
对于这种情况，我们可以进行一些判断即可，排除这些特殊页面：

```
'通用事件头,用于发送已经存在的常见文件
Select Case e.Path
    Case "saletj.htm","showtj.htm" '排除两个页面,由后面的代码负责处理
    Case Else '其他页面或文件如果已经存在,直接发送
        Dim fl As String = "d:\web\" & e.path
        If filesys.FileExists(fl)
            Dim idx As Integer = fl.LastIndexOf(".")
            Dim ext As String   = fl.SubString(idx)
            Select Case ext
                Case ".jpg",".gif",".png",".bmp",".wmf",".js",".css" ,".html",".htm",".zip",".rar"
                    e.WriteFile(fl)
                    Return '这里必须返回
            End Select
        End If
End Select
'以下是动态生成网页的代码
Select Case e.Path
    Case "addnew.htm"
        '生成新增页面
    Case "saletj.htm"
        '生成统计页面
    Case "showtj.htm"
        '生成统计结果显示页面
    Case "order.htm"
        '生成订购页面
    Case Else
      e.WriteString("糟糕,文件未找到！")
End Select
```

有一个问题需要特别留意：

以上面的代码为例，如果你在"d:\web"目录下保存了一个"addnew.htm"网页文件，那么系统将直接

发送这个文件，而不会执行后面动态生成"addnew.htm"的代码。
如果某次你修改代码后，生成的网页始终没有变化，一般都是这个原因造成的。

6.28 一个简单的录入界面

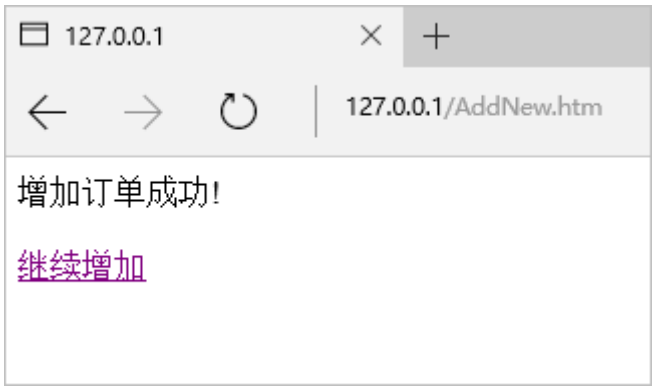
后台有一个订单表，结构如下：

	产品	客户	雇员	单价	折扣	数量	金额	日期
1	PD05	CS03	EP04	17	0.1	650	9945	2015-01-02
2	PD02	CS01	EP01	20	0.1	400	7200	2015-01-03
3	PD05	CS04	EP02	17	0	320	5440	2015-01-03
4	PD01	CS03	EP04	18	0.15	80	1224	2015-01-04
5	PD03	CS03	EP04	10	0	490	4900	2015-01-04
6	PD04	CS04	EP05	17.6	0	240	4224	2015-01-07

现在希望设计一个网页用于输入新的订单：



用户输入数据，单击确定后，能在后台的表新增订单，并显示如下页面，让用户选择是否继续增加订单：



看似复杂，实际上 HttpRequet 事件的代码很简单，望大家仔细体会：

```
Select Case e.Path
    Case "AddNew.htm"
        If e.PostValues.Count = 0 Then
            Dim sb As New StringBuilder
                sb.AppendLine("<form action='AddNew.htm' enctype='multipart/form-data' method='post' id='form1' name='form1'>")
            sb.AppendLine("产品: <input name='cp' id='cp'><br/><br/>")
            sb.AppendLine("客户: <input name='kh' id='kh'><br/><br/>")
            sb.AppendLine("雇员: <input name='gy' id='gy'><br/><br/>")
            sb.AppendLine("单价: <input type='number' name='dj' id='dj'><br/><br/>")
             sb.AppendLine(" 折 扣 : <input type='number' name='zk' id='zk' min='0' max='0.15' step='0.01'><br/><br/>")
            sb.AppendLine("数量: <input type='number' name='sl' id='sl'><br/><br/>")
            sb.AppendLine("日期: <input type='date' name='rq' id='rq'><br/><br/>")
            sb.AppendLine("<input Type='submit' name='Sumbit' id='Sumbit' value='确定'>")
            sb.AppendLine("</form>")
            e.WriteString(sb.ToString)
        Else
            Dim dr As DataRow = DataTables("订单").SQLAddNew()
            Dim inms() As String = {"cp","kh","gy","dj","zk","sl","rq"} '输入框名称数组
            Dim cnms() As String = {"产品","客户","雇员","单价","折扣","数量","日期"} '列名数组,注意列名和输入框必须一一对应,位置不能错乱
            For i As Integer = 0 To inms.Length -1
              dr(cnms(i)) = e.PostValues(Inms(i))
            Next
            dr.Save() '用SQLAddNew增加的行,必须保存一下,否则会被丢弃.

            Dim sb As New StringBuilder
            sb.AppendLine("增加订单成功! <br/><br/>")
            sb.AppendLine("<a href='AddNew.htm'>继续增加</a>")
            e.WriteString(sb.ToString)
        End If
End Select
```

提示：

1、这里用SQLAddNew直接在后台增加行，SQLAddNew是Foxtable 2017新增加的一个方法，适合加载结构但不加载数据的表，如果希望服务端能即时显示用户通过网页增加的行，请改用AddNew，二者效率差不多。
2、如果用户量很大，需要更高的效率，建议采用SQL语句插入行，需要注意的是，只有外部数据源才支持用SQL语句插入行。

6.29 文件的上传与接收

用户可以通过表单上传文件。

表单准备

如果表单要包括文件上传组件，在定义表单的时候，要将 enctype 设置为"multipart/form-data"，例如：

```
<form enctype='multipart/form-data' action='accept.htm' method='post' id='form1' name='form1'>
</form>
```

否则，HttpServer 将无法收到用户上传的文件。

在表单插入一个 type 为 file 的 input 元素，即可实现文件上传，例如：

```
<input type='file' name='up1' id='up1'>
```

如果要允许用户选择多个文件上传，加上 multiple 属性即可，例如

```
<input type='file' name='up2' id='up2' multiple>
```

下面是一个完整的文件上传表单：

```
<form enctype='multipart/form-data' action='accept.htm' method='post' id='form1' name='form1'>
单文件上传: <input type='file' name='up1' id='up1'><br/><br/>
多文件上传: <input type='file' name='up2' id='up2' multiple><br/><br/>
<input Type='submit' name='Sumbit' id='Sumbit' value='确定'>
</form>
```

接收文件

HttpRequest 的 e 参数有个 Files 属性，这是一个字典，键为文件上传组件的 name 属性，值是一个字符串集合，包括用户通过这个上传组件上传的所有文件名。

HttpRequest 的 e 参数有个 SaveFile 方法，用于保存接收到的文件，其语法为：

SaveFile(Key,UploadFile,LocalFile)

Key	文件上传组件的 name 属性
UploadFile	用户上传的文件名称，不含路径。
LocalFile	要保存到本地的文件名称，含路径

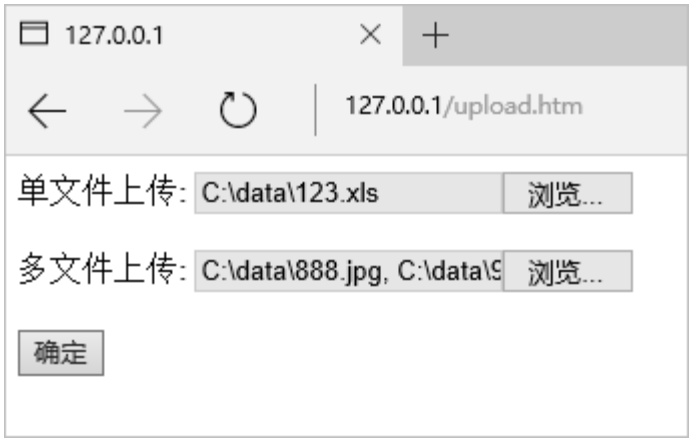
完整示例

将 HttpRequest 设置为以下代码：

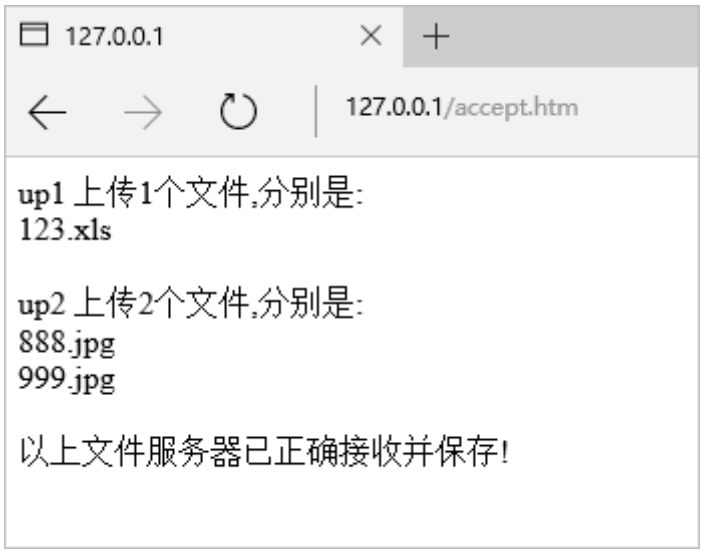
```
Select Case e.Path
```

```
    Case "upload.htm"
        Dim sb As New StringBuilder
          sb.appendLine("<form enctype='multipart/form-data' action='accept.htm' method='post' id='form1' name='form1'>")
        sb.appendLine("单文件上传: <input type='file' name='up1' id='up1'><br/><br/>")
        sb.appendLine("多文件上传: <input type='file' name='up2' id='up2' multiple><br/><br/>")
        sb.appendLine("<input Type='submit' name='Sumbit' id='Sumbit' value='确定'>")
        sb.appendLine("</form>")
        e.WriteString(sb.ToString)
    Case "accept.htm"
        Dim sb As New StringBuilder
        For Each key As String In e.Files.Keys
          sb.AppendLine(key & "  上传" & e.Files(key).Count & "个文件,分别是:</br>")
          For Each fl As String In e.Files(key)
            sb.AppendLine(fl & "<br>")
            e.SaveFile(key,fl,"d:\web\uploadfiles\" & fl) '保存接收到的文件
          Next
          sb.AppendLine("</br>")
        Next
        sb.AppendLine("以上文件服务器已正确接收并保存!")
        e.WriteString(sb.ToString)
End Select
```

上面的代码生成了两个页面，upload 页面用于上传文件：



accept.htm 页面用于显示并保存用户上传的文件：



6.30 避免覆盖同名文件

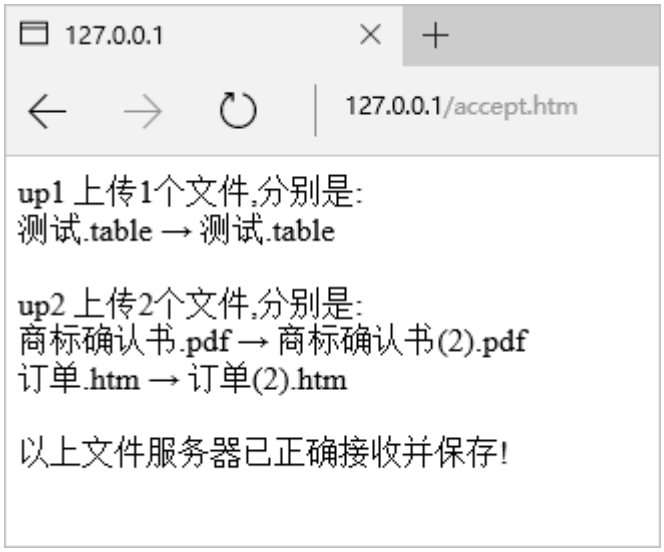
我们在上一节设计的文件上传和接收示例中，如果多个用户上传了同名的文件，会出现覆盖的情况。

为避免同名文件被覆盖，我们可以将 HttpRequest 事件代码改为：

```
Select Case e.Path
    Case "upload.htm"
        Dim sb As New StringBuilder
            sb.appendLine("<form enctype='multipart/form-data' action='accept.htm' method='post' id='form1' name='form1'>")
        sb.appendLine("单文件上传: <input type='file' name='up1' id='up1'><br/><br/>")
        sb.appendLine("多文件上传: <input type='file' name='up2' id='up2' multiple><br/><br/>")
        sb.appendLine("<input Type='submit' name='Sumbit' id='Sumbit' value='确定'>")
        sb.appendLine("</form>")
        e.WriteString(sb.ToString)
    Case "accept.htm"
        Dim sb As New StringBuilder
        For Each key As String In e.Files.Keys
            sb.AppendLine(key & "  上传" & e.Files(key).Count & "个文件,分别是:</br>")
            For Each fl As String In e.Files(key)
                Dim NewName As String = fl
                Dim idx As Integer = fl.LastIndexOf(".")
                Dim cnt As Integer = 1
                Do While FileSys.FileExists("d:\web\uploadfiles\" & NewName) '判断文件夹是否存在同名文件
                    NewName = fl.Insert(idx,"(" & cnt & ")") '如果存在同名文件,在原文件名加上序号
```

```
            cnt = cnt + 1 '递增序号
          Loop
          sb.AppendLine(fl & "  →  " & NewName & "<br>") '
          e.SaveFile(key,fl,"d:\web\uploadfiles\" & NewName) '保存接收到的文件
        Next
        sb.AppendLine("</br>")
      Next
      sb.AppendLine("以上文件服务器已正确接收并保存!")
      e.WriteString(sb.ToString)
End Select
```

这样 HttpServer 接收到同名文件后，会自动给同名文件加上编号，而不是直接覆盖：



6.31 手机拍照与上传

如果在手机上访问上一节生成的网页，效果和 pc 上是不太一样的。

例如在 iPhone 上访问上面的网页，可以直接拍照并上传：


中国联通
10:38
192.168.0.103/upload.htm
单文件上传:
选取文件
未选择文件
多文件上传:
选取文件
未选择文件
确定
拍照或录像
照片图库
取消



实际上你通过 iPhone 访问上述页面时，字体是很小的，上面是放大后截图的，后面的章节会告诉大家解决办法，现在暂时放过这个问题。

6.32 带上传功能的录入页面

假定有一个员工表，结构如下：

	姓名	部门	职务	学历	照片
1	李云龙	开发部	经理	本科	EP1.BMP
2	赵刚	行政部	经理	硕士	EP2.BMP

希望生成一个下图所示的页面，让用户此页面增加员工：



HttpRequest 事件代码很简单：

```
Select Case e.Path
    Case "AddNew.htm"
        If e.PostValues.Count = 0 AndAlso e.Files.Count = 0 Then
            Dim sb As New StringBuilder
             sb.AppendLine("<form  enctype='multipart/form-data'  action='AddNew.htm'  method='post' id='form1' name='form1'>")
            sb.AppendLine("姓名: <input name='xm' id='xm'><br/><br/>")
            sb.AppendLine("部门: <input name='bm' id='bm'><br/><br/>")
            sb.AppendLine("职务: <input name='zw' id='zw'><br/><br/>")
            sb.AppendLine("学历: <input name='xl' id='xl'><br/><br/>")
            sb.AppendLine("照片: <input type='file' name='zp' id='zp' multiple><br/><br/>")
            sb.AppendLine("<input Type='submit' name='Sumbit' id='Sumbit' value='确定'>")
            sb.AppendLine("</form>")
            e.WriteString(sb.ToString)
        Else
            Dim dr As DataRow = DataTables("员工").AddNew()
            Dim inms() As String = {"xm","bm","zw","xl"} '输入框名称数组
            Dim cnms() As String = {"姓名","部门","职务","学历"} '列名数组,注意列名和输入框必须一一对应,位置不能错乱
            For i As Integer = 0 To inms.Length -1
                dr(cnms(i)) = e.PostValues(Inms(i))
            Next
            If  e.Files.ContainsKey("zp") Then
                dr("照片") = e.Files("zp")(0)
                e.SaveFile("zp", e.Files("zp")(0), ProjectPath & "Attachments\" & e.Files("zp")(0))
            End If
            dr.Save()
```

```
        Dim sb As New StringBuilder
        sb.AppendLine("增加记录成功! <br/><br/>")
        sb.AppendLine("<a href='AddNew.htm'>继续增加</a>")
        e.WriteString(sb.ToString)
    End If
End Select
```

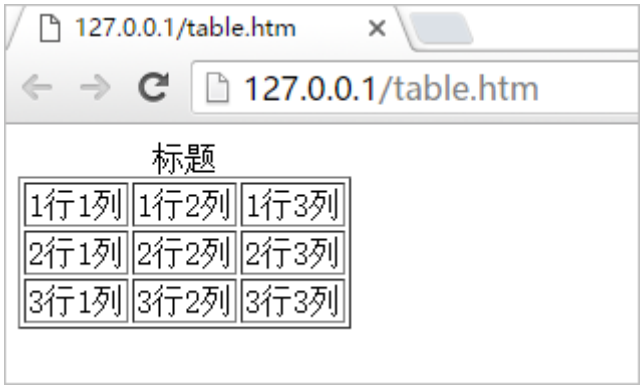
上面的代码，用一个页面"AddNew.htm"完成了录入网页的生成，数据和文件的接收与 保存，以及记录增加成功后反馈页面的生成。

6.33 生成表格

HTML 中的表、行、单元格的开始标签分别为<table>、<tr>、<td>，对应的结束标签分别为</table>、</tr>、</td>。
我们将 HttpRequest 事件代码代码设置为：

```
Select Case e.Path
    Case "table.htm"
        Dim sb As New StringBuilder
        sb.AppendLine("<table border='1'>")
        sb.AppendLine("<caption>标题</caption>")
        sb.AppendLine("<tr><td>1行1列</td><td>1行2列</td><td>1行3列</td></tr>")
        sb.AppendLine("<tr><td>2行1列</td><td>2行2列</td><td>2行3列</td></tr>")
        sb.AppendLine("<tr><td>3行1列</td><td>3行2列</td><td>3行3列</td></tr>")
        sb.AppendLine("</table>")
        e.WriteString(sb.ToString)
End Select
```

在浏览器中显示的结果为：



有关 HTML 表格的更多知识，请访问：
http://www.w3school.com.cn/html/html_tables.asp
http://www.w3school.com.cn/css/css_table.asp

6.34 分页显示数据

本机希望设计一个分页显示订单表的页面，用户可以通过单击“上一页”、“下一页”来切换页面：



产品	客户	数量	单价	折扣	金额	日期
PD01	CS05	60	18	0	1080	2015-06-26
PD01	CS02	300	18	0	5400	2015-06-25
PD02	CS03	30	19	0	570	2015-06-24
PD02	CS02	400	15.2	0	6080	2015-06-23
PD05	CS01	20	17	0	340	2015-06-22
PD05	CS03	300	17	0	5100	2015-06-21
PD05	CS02	120	17	0.2	1632	2015-06-20
PD01	CS05	500	18	0	9000	2015-06-19
PD04	CS02	160	17.6	0.2	2252.8	2015-06-18
PD01	CS01	10	14.4	0.02	141.12	2015-06-17



HttpRequst 事件的代码并不复杂，其中一些技巧请仔细体会：

```
Select Case e.Path
    Case "list.htm"
        Dim page As Integer = 0 '默认page为0，显示第一页
        Dim pageRows As Integer = 10 '每页10行
        If e.GetValues.ContainsKey("page") Then  '如果地址中有page参数
            Integer.TryParse(e.GetValues("page"), page) '提取page参数
        End If
        Dim StartRow As Integer = page * pageRows '此页第一行
        Dim EndRow As Integer = (page + 1) * pageRows - 1 '此页最后一行
        Dim lst As List(of DataRow) = DataTables("订单").Select("","日期  Desc") '按日期顺序显示
        If StartRow > lst.Count -1 Then
            e.WriteString("已经是最后一页!")
            Return
        End If
        EndRow = Math.Min(EndRow, lst.Count - 1) '这是必须的
        Dim sb As New StringBuilder
        Dim nms() As String = {"产品","客户","数量","单价","折扣","金额","日期"}
        sb.AppendLine("<Table border='1'>")
        sb.AppendLine("<caption>订单浏览<caption></br>")
        sb.Append("<tr>")
        For Each nm As String In nms
```

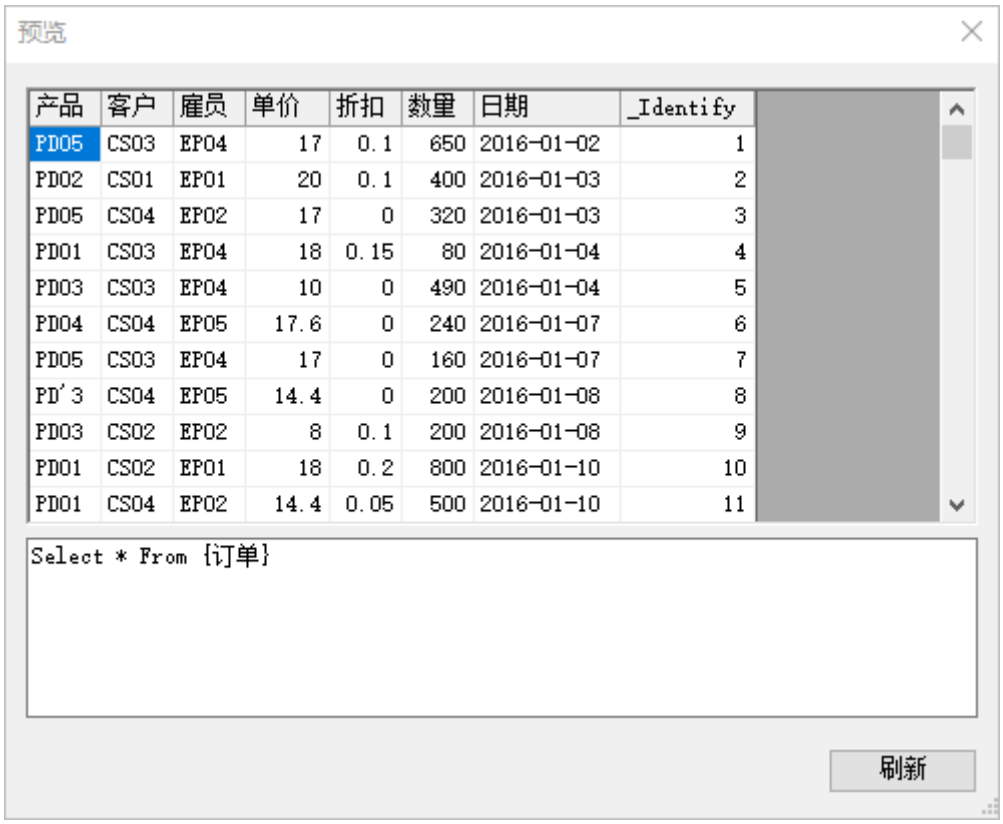
```
        sb.Append("<td>" & nm & "</td>")
    Next
    sb.AppendLine("</tr>")
    For r As Integer = StartRow To EndRow
        sb.Append("<tr>")
        For Each nm As String In nms
            sb.Append("<td>" & lst(r)(nm) & "</td>")
        Next
        sb.AppendLine("</tr>")
    Next
    sb.AppendLine("</Table></br></br>")
    If page > 0 Then
        sb.Append("<a href='list.htm?page=" & page - 1 & "'>上一页</a> ")
    End If
    If EndRow < lst.Count -1 Then
        sb.Append("<a href='list.htm?page=" & page + 1 & "'>下一页  </a>" )
    End If
    e.WriteString(sb.ToString)
End Select
```

上述代码合成的上一页、下一页链接地址包括了要显示的页面参数，例如：
http://127.0.0.1/list.htm?page=1
表示请求显示第二页（0 表示第一页），HttpRequst 事件提取出这个页面参数后，以表格形式将对应页面数据显示在用户浏览器上。
这种数据传递方式，之前我们已经介绍过，请参考：另一种数据提交方式，只是本节是通过 GetValues 属性获取 page 参数的。

6.35 必须掌握的 Row_Number

接下来我们将给大家介绍如何分页显示后台数据，但是在这之前我需要先给大家介绍一下 SQL Server 的 Row_Number 函数。

Row_Number 函数让我们分页显示后台变得非常方便。
一般用户对于 Row_Number 函数的使用会比较迷糊，不过通过本节的学习，你会发现 Row_Number 其实很简单。
一个例子
假定我们需要按照日期降序加载订单表数据，每页 10 行，加载第 8 页，我们可以在 Foxtable 的 SQL 窗口测试以下各步骤的 Select 语句。
1、按照常规写出最简单的 Select 语句：
Select * From {订单}
这是显示的结果是：



2、RowNumber 函数用于按指定的列排序，来生成一个序号列，语法为：

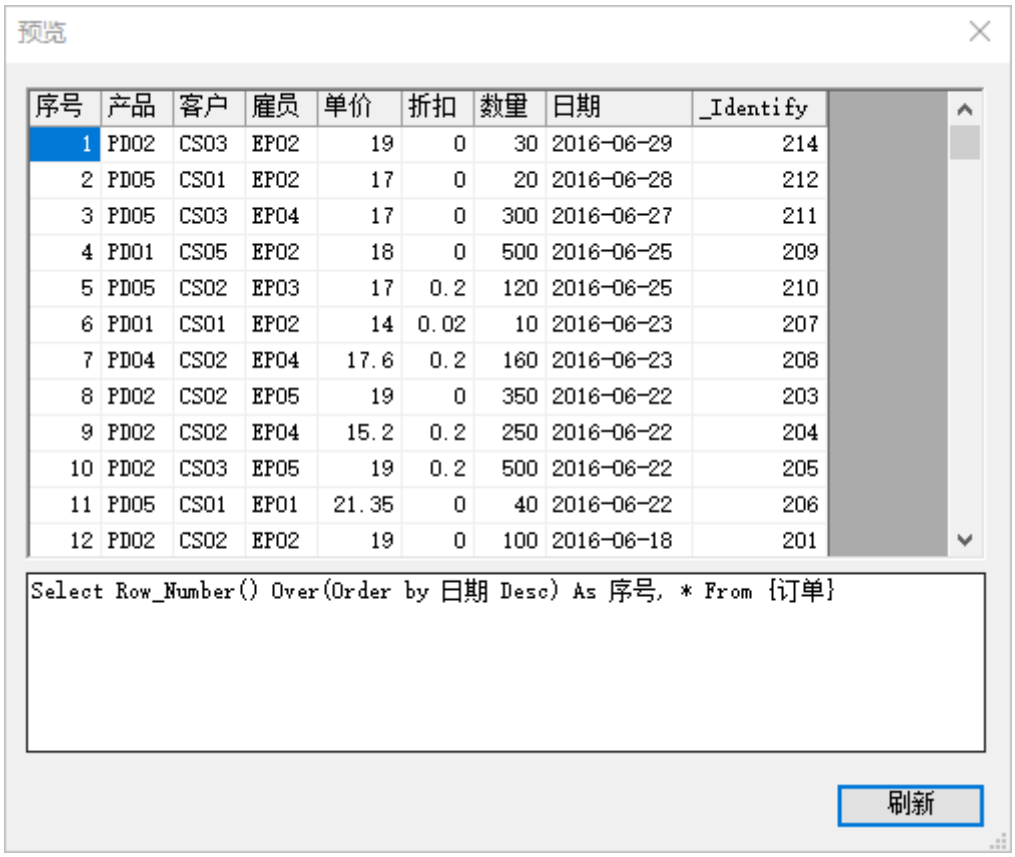
Row_Number() Over(Order by 排序列) As 序号列

可变的只有排序列和序号列名，默认是升序，如果需要降序，在排序列后面加上 DESC 参数即可。

现在将我们的 Select 语句改写为：

Select Row_Number() Over(Order by 日期 Desc) As 序号, * From {订单}

显示的结果如下图，可以看到增加了一个序号列，序号是根据日期按照降序编排的。

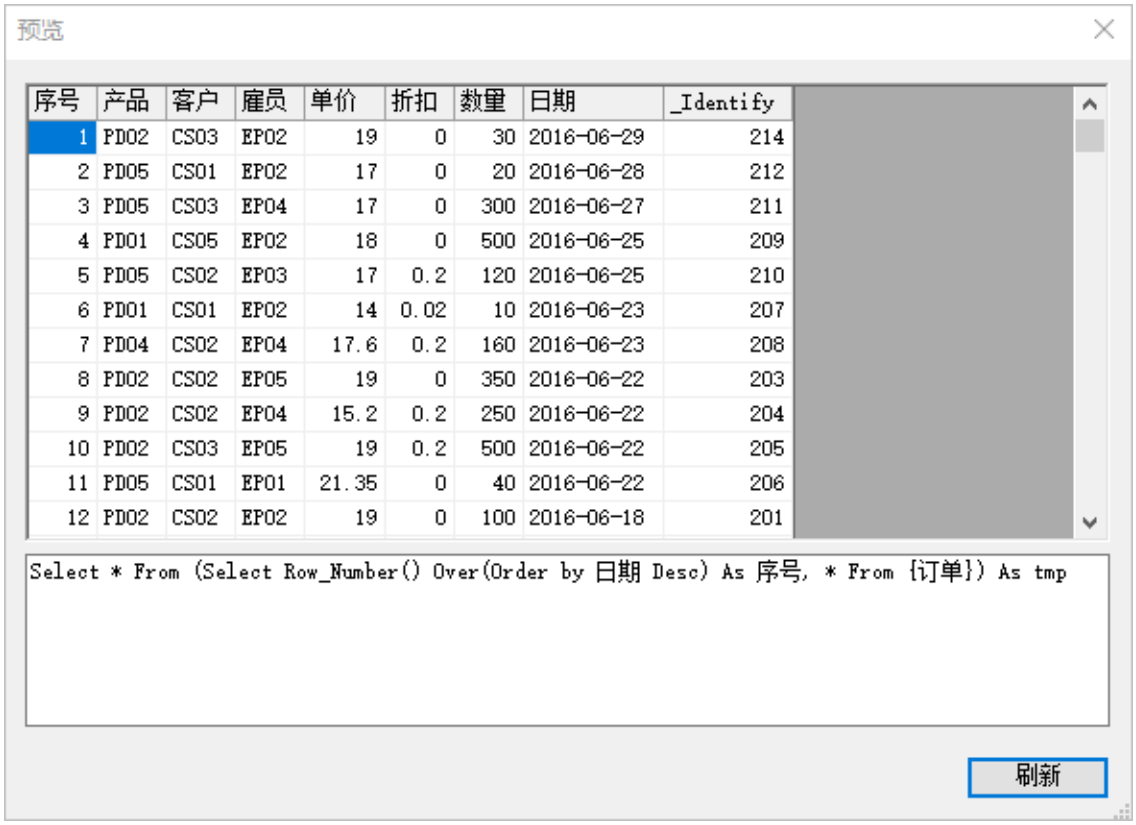


3、现在我们可以将上面的 Select 语句当作一个表来看待，我们从这个表提取数据:

```
Select * From (Select Row_Number() Over(Order by 日期 Desc) As 序号, * From {订单}) As tmp
```

上述语句分三部分，中间括起来的蓝色部分可以理解为一个临时表，我们需要给这个临时表取个名字，后面这段红色部分将这个临时表取名为"tmp"

现在的显示结果没有变化：

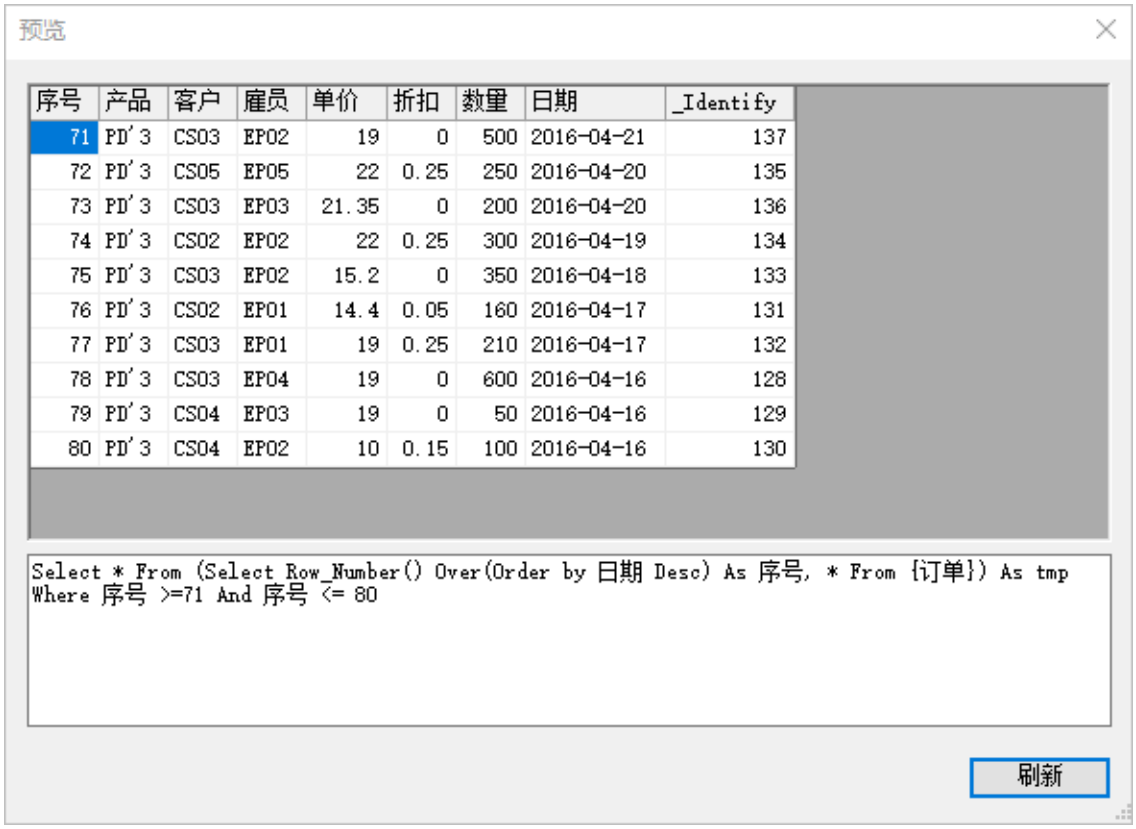


序号	产品	客户	雇员	单价	折扣	数量	日期	_Identify
1	PD02	CS03	EP02	19	0	30	2016-06-29	214
2	PD05	CS01	EP02	17	0	20	2016-06-28	212
3	PD05	CS03	EP04	17	0	300	2016-06-27	211
4	PD01	CS05	EP02	18	0	500	2016-06-25	209
5	PD05	CS02	EP03	17	0.2	120	2016-06-25	210
6	PD01	CS01	EP02	14	0.02	10	2016-06-23	207
7	PD04	CS02	EP04	17.6	0.2	160	2016-06-23	208
8	PD02	CS02	EP05	19	0	350	2016-06-22	203
9	PD02	CS02	EP04	15.2	0.2	250	2016-06-22	204
10	PD02	CS03	EP05	19	0.2	500	2016-06-22	205
11	PD05	CS01	EP01	21.35	0	40	2016-06-22	206
12	PD02	CS02	EP02	19	0	100	2016-06-18	201

4、现在是最后一步，每页 10 行，第 8 页的数据的序号列范围为 70 到 80，我们将 Select 语句改为：

Select * From (Select Row_Number() Over(Order by 日期 Desc) As 序号, * From {订单}) As tmp Where 序号 >=71 And 序号 <= 80

显示结果为：



序号	产品	客户	雇员	单价	折扣	数量	日期	_Identify
71	PD'3	CS03	EP02	19	0	500	2016-04-21	137
72	PD'3	CS05	EP05	22	0.25	250	2016-04-20	135
73	PD'3	CS03	EP03	21.35	0	200	2016-04-20	136
74	PD'3	CS02	EP02	22	0.25	300	2016-04-19	134
75	PD'3	CS03	EP02	15.2	0	350	2016-04-18	133
76	PD'3	CS02	EP01	14.4	0.05	160	2016-04-17	131
77	PD'3	CS03	EP01	19	0.25	210	2016-04-17	132
78	PD'3	CS03	EP04	19	0	600	2016-04-16	128
79	PD'3	CS04	EP03	19	0	50	2016-04-16	129
80	PD'3	CS04	EP02	10	0.15	100	2016-04-16	130

上图就是按照日期降序加载订单表，每页 10 行，加载第 8 页的结果。
最终的 Select 语句看起来有点复杂：

```
Select * From (Select Row_Number() Over(Order by 日期 Desc) As 序号, * From {订单}) As tmp Where 序号 >=71 And 序号 <= 80
```

但是你可以理解为：

```
Select * From tmp Where 序号 >=71 And 序号 <= 80
```

而 tmp 是个临时表，是通过以下语句生成的：

```
(Select Row_Number() Over(Order by 日期 Desc) As 序号, * From {订单}) As tmp
```

注意临时表的 Select 语句必须用圆括号括起来。

6.36 分页显示后台数据

上一节我们实现分页显示已经加载到 DataTable 中的数据。
这一节介绍如何分页显示未加载的数据，在给出具体代码之前，我先给大家一些建议：
1、如果服务端的数据量比较小，将数据加载进来，可以提高处理速度。
2、如果服务端数据量比较大，加载全部数据，可能会耗尽内存，甚至导致系统崩溃。
3、怎样才算小，怎样才算大，这没有标准，请以实际运行效果为准，一般不建议服务端加载超过 5 万行的数据。
4、如果某个表虽然不加载数据，但是需要对这个表进行增加、删除和修改等操作，那么建议加载这个表的结构，这样就可以利用后台数据处理函数简化代码。
5、建议使用 SQL Server 作为数据源，容量更大，性能更好，而且 SQL Server 有 Row_Number 函数，分页加载方便很多。

完整示例

下面的 HttpRequest 事件，用于分页显示后台数据，这里假定你使用的是 SQL Server 作为数据源：

```
Select Case e.Path
    Case "list.htm"
        Dim page As Integer = 0 '默认page为0，显示第一页
        Dim pageRows As Integer = 10 '每页10行
        If e.GetValues.ContainsKey("page") Then  '如果地址中有page参数
            Integer.TryParse(e.GetValues("page"), page) '提取page参数
        End If
        Dim StartRow As Integer = page * pageRows '此页第一行
        Dim EndRow As Integer = (page + 1) * pageRows - 1 '此页最后一行
        Dim cmd As New SQLCommand
        cmd.ConnectionName = "orders" '记得设置数据源名称
        cmd.CommandText = "Select Count(*) From {订单}"
```

```
    Dim Count As Integer = cmd.ExecuteScalar() '获取总的行数
    cmd.CommandText = "Select *, [数量]*[单价]*(1-[折扣]) As 金额 From (Select Row_Number() Over(Order by [日期]) As RowNum, * From 订单) As a "
    cmd.CommandText = cmd.CommandText & "  Where RowNum >= " & StartRow & " And RowNum <= " & EndRow
    Dim dt As DataTable = cmd.ExecuteReader '获取该页数据
    Dim sb As New StringBuilder
    Dim nms() As String = {"产品","客户","数量","单价","折扣","金额","日期"}
    sb.AppendLine("<Table border='1'>")
    sb.AppendLine("<caption>订单浏览<caption></br>")
    sb.Append("<tr>")
    For Each nm As String In nms
      sb.Append("<td>" & nm & "</td>")
    Next
    sb.AppendLine("</tr>")
    For Each r As DataRow In dt.DataRows
      sb.Append("<tr>")
      For Each nm As String In nms
        sb.Append("<td>" & r(nm) & "</td>")
      Next
      sb.AppendLine("</tr>")
    Next
    sb.AppendLine("</Table></br></br>")
    If page > 0 Then
      sb.Append("<a href='list.htm?page=" & page - 1 & "'>上一页</a> ")
    End If
    If EndRow < Count -1 Then
      sb.Append("<a href='list.htm?page=" & page + 1 & "'>下一页 </a>" )
    End If
    e.WriteString(sb.ToString)
End Select
```

6.37 一个综合性的例子

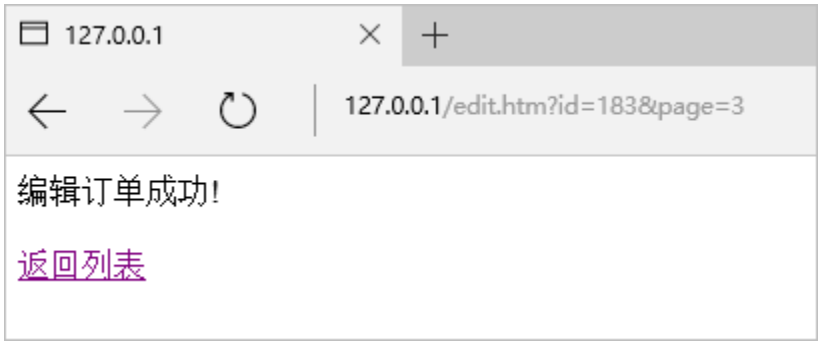
本节的任务是在上一节分页显示后台数据的基础上，加上编辑和删除订单的功能：



单击删除链接，可以删除对应的订单，单击编辑链接，可以编辑对应的订单：



编辑或删除订单后，可以返回原来的分页：



知识准备

用户单击“编辑”链接，后台如何知道要编辑哪一个订单呢？ 编辑完成后，又怎么知道返回哪一个页面呢？

我们可以在表单中插入隐藏字段，来传递主键和页码，参考：

在表单中插入标记数据

不过本节我们不采用这个方法，我们采用 get 方式来传递主键和页码，参考：

另一种数据提交方式

Values、PostValues 和 GetValues

完整代码

这是 HttpRequest 事件的完整代码，由于包括分页、编辑、删除三部分的代码，所以代码有点长，后面会对关键代码进行解释：

```
Dim sb As New StringBuilder
Select Case e.Path
    Case "list.htm" '分页显示
        Dim page As Integer = 0 '默认page为0，显示第一页
        Dim pageRows As Integer = 10 '每页10行
        If e.GetValues.ContainsKey("page") Then  '如果地址中有page参数
            Integer.TryParse(e.GetValues("page"), page) '提取page参数
        End If
        Dim StartRow As Integer = page * pageRows '此页第一行
        Dim EndRow As Integer = (page + 1) * pageRows - 1 '此页最后一行
        Dim cmd As New SQLCommand
        cmd.ConnectionName = "orders" '记得设置数据源名称
        cmd.CommandText = "Select Count(*) From {订单}"
        Dim Count As Integer = cmd.ExecuteScalar() '获取总的行数
        cmd.CommandText = "Select *, [数量]*[单价]*(1-[折扣]) As 金额 From (Select Row_Number() Over(Order by [日期] desc) As RowNum, * From 订单) As a "
        cmd.CommandText = cmd.CommandText & "  Where RowNum >= " & StartRow & " And RowNum <= " & EndRow
        Dim dt As DataTable = cmd.ExecuteReader '获取该页数据
        Dim nms() As String = {"产品","客户","数量","单价","折扣","金额","日期"}
        sb.AppendLine("<Table border='1'>")
        sb.AppendLine("<caption>订单浏览<caption></br>")
        sb.Append("<tr>")
```

```
        For Each nm As String In nms
            sb.Append("<td>" & nm & "</td>")
        Next
        sb.Append("<td></td>")
        sb.Append("<td></td>")
        sb.AppendLine("</tr>")
        For Each r As DataRow In dt.DataRows
            sb.Append("<tr>")
            For Each nm As String In nms
                sb.Append("<td>" & r(nm) & "</td>")
            Next
            sb.Append("<td><a  href='edit.htm?id=" & r("_Identify") & "&page=" & page & "'>编辑</a></td>")
            sb.Append("<td><a  href='delete.htm?id=" & r("_Identify") & "&page=" & page & "'>删除</a></td>")
            sb.AppendLine("</tr>")
        Next
        sb.AppendLine("</Table></br></br>")
        If page > 0 Then
            sb.Append("<a href='list.htm?page=" & page - 1 & "'>上一页</a> ")
        End If
        If EndRow < Count -1 Then
            sb.Append("<a href='list.htm?page=" & page + 1 & "'>下一页  </a>" )
        End If
        e.WriteString(sb.ToString)
    Case "delete.htm" '删除订单
        If e.GetValues.ContainsKey("id") AndAlso e.GetValues.ContainsKey("page") Then
            Dim cnt As Integer
            cnt = DataTables("订单").SQLDeleteFor("[_Identify] = " & e.GetValues("id"))
            If cnt > 0 Then
                sb.AppendLine("删除成功!</br></br>")
            Else
                sb.AppendLine("删除失败!</br></br>")
            End If
            sb.Append("<a href='list.htm?page=" & e.GetValues("page") & "'>返回列表</a>" )
            e.WriteString(sb.ToString)
        Else
            e.WriteString("糟糕,可能出错了")
        End If
    Case "edit.htm" '编辑订单
        If e.GetValues.ContainsKey("id") AndAlso e.GetValues.ContainsKey("page") Then
            Dim dr As DataRow = DataTables("订单").SQLFind("[_Identify] = " & e.GetValues("id"))
            If dr Is Nothing Then
                e.WriteString("此订单不存在,可能已经被其他用户删除!")
```

```
            Return
        End If
        If e.PostValues.Count = 0 Then '生成编辑页面
            sb.appendLine("<form action='edit.htm?id=" & e.GetValues("id") & "&page=" & e.GetValues("page") & "' method='post' id='form1' name='form1'>")
            sb.appendLine("产品：<input name='cp' id='cp' value='" & dr("产品") & "'><br/><br/>")
            sb.appendLine("客户：<input name='kh' id='kh' value='" & dr("客户") & "'><br/><br/>")
            sb.appendLine("雇员：<input name='gy' id='gy' value='" & dr("雇员") & "'><br/><br/>")
            sb.appendLine("单价：<input type='number' name='dj' id='dj' step='0.1' value='" & dr("单价") & "'><br/><br/>")
            sb.appendLine("折扣: <input type='number' name='zk' id='zk' step='0.01' value='" & dr("折扣") & "'><br/><br/>")
            sb.appendLine("数量：<input type='number' name='sl' id='sl' value='" & dr("数量") & "'><br/><br/>")
            sb.appendLine("日期：<input type='date' name='rq' id='rq' value='" & dr("日期") & "'><br/><br/>")
            sb.appendLine("<input Type='submit' name='Sumbit' id='Sumbit' value='确定'>")
            sb.appendLine("</form>")
            e.WriteString(sb.ToString)
        Else '获取用户提交的数据
            Dim inms() As String = {"cp","kh","gy","dj","zk","sl","rq"} '输入框名称数组
            Dim cnms() As String = {"产品","客户","雇员","单价","折扣","数量","日期"} '列名数组,注意列名和输入框必须一一对应,位置不能错乱
            For i As Integer = 0 To inms.Length -1
                If e.PostValues.ContainsKey(inms(i)) Then '必须判断,因为PostValues集合只包括已经输入的值
                    dr(cnms(i)) = e.PostValues(Inms(i))
                End If
            Next
            dr.Save() '用SQLAddNew增加的行,必须保存一下,否则会被丢弃.
            sb.AppendLine("编辑订单成功! <br/><br/>")
            sb.AppendLine("<a href='list.htm?page=" & e.GetValues("page") & "'>返回列表</a>")
            e.WriteString(sb.ToString)
        End If
    Else
        e.WriteString("糟糕,可能出错了")
    End If
End Select
```

分页显示的代码和上一节基本一样，只是为每一行增加了编辑和删除链接，合成这两个链接的代码为：

```
sb.Append("<td><a href='edit.htm?id=" & r("_Identify") & "&page=" & page & "'>编辑</a></td>")
sb.Append("<td><a href='delete.htm?id=" & r("_Identify") & "&page=" & page & "'>删除</a></td>")
```

下面是第 5 页(页面编号 4)中主键为 176 的行对应的链接：

```
<a href='edit.htm?id=176&page=4'>编辑</a>
<a href='delete.htm?id=176&page=4'>删除</a>
```

表示要编辑或删除的订单的主键是 176，当前页面编号是 4。

例如 HttpRequset 事件收到访问请求“delete.htm?id=176&page=4”，从中提取出主键 176 和页面编号 4，删除主键为 176 的订单后，再合成一个链接：

```
sb.Append("<a href='list.htm?page=" & e.GetValues("page") & "'>返回列表</a>" )
```

这样用户单击这个连接，即可返回原来的分页。

再例如 HttpRequset 事件收到访问请求“edit.htm?id=176&page=4”后，从中提取出主键 176 和页面编号 4，然后从后台找出主键为 176 的行，根据 该行的内容生成编辑表单，注意我们定义表单的代码：

```
sb.appendLine("<form action='edit.htm?id=" & e.GetValues("id") & "&page=" & e.GetValues("page") &
"' method='post' id='form1' name='form1'>")
```

这段代码合成的内容类似：

```
<form action='edit.htm?id=176&page=4' method='post' id='form1' name='form1'>
```

当用户编辑完成，单击确定按钮时，将输入的值发送到到"edit.htm?id=176&page=4"。

HttpRequset 收到这个请求后，e.PostValues.Count 属性会大于 0，系统知道用户本次访问 目的是提交编辑结果。

HttpRequset 事件找出主键为 176 的行，将接收到的值存储到对应的列，并生成一个链接，让用户可以返回原来的页面：

```
sb.AppendLine("<a href='list.htm?page=" & e.GetValues("page") & "'>返回列表</a>")
```

重要提示：所有的浏览器都可以查看网页源码，通过查看网页源码，有助于我们学习理解和排查错误。

增加订单功能

为了让代码不至于太长，上面的例子，只有编辑和删除订单的功能，没有增加订单的功能。

实际上，增加订单功能实现起来更简单，而且我们之前已经讲述过了，大家自己将这部分的代码整合进来即可，参考：

一个简单的录入功能

简化 HttpRequest 事件代码

这是到目前为止，我们写的最长的一段 HttpRequest 事件代码，用了三个页面，实现了分页显示、增加订单、删除订单的功能。

建议大家一个页面做成一个函数，在 HttpRequest 不要涉及具体功能，只负责调用相关函数，参考：

简化 HttpRequest 事件代码

6.38 使用 Cookie

通过 Cookie 可以在本机临时存储数据，每次访问服务器网页时，都会自动将 Cookie 中的值，传递给服务器。

HttpRequest 事件有个 AppendCookie 方法，用于添加 Cookie，语法为：
AppendCookie(Name, Value)
Name：Cookie 名称
Value：Cookie 值
HttpRequest 事件有个 Cookies 字典，包括所有的 Cookie，例如要列出所有 Cookie 的名称和值：

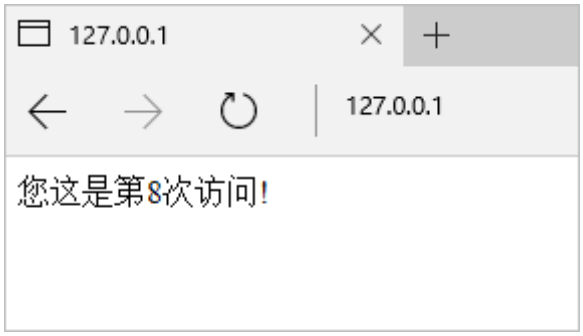
```
Dim sb As New StringBuilder
For Each key As String In e.Cookies.Keys
    sb.AppendLine(key & ":" & e.Cookies(key))
Next
e.WriteString(sb.Tostring)
```

示例

将 HttpRequest 事件代码设置为：

```
Dim sb As New StringBuilder
Dim cnt As   Integer = 1
Integer.TryParse(e.Cookies("count"),cnt) '提取 cookie 的值，并转换为整数
cnt = cnt + 1
e.AppendCookie("count",cnt) '在客户端存储 Cookie
e.WriteString("您这是第" & cnt & "次访问!")
```

现在每次刷新页面，访问次数都会递增 1：



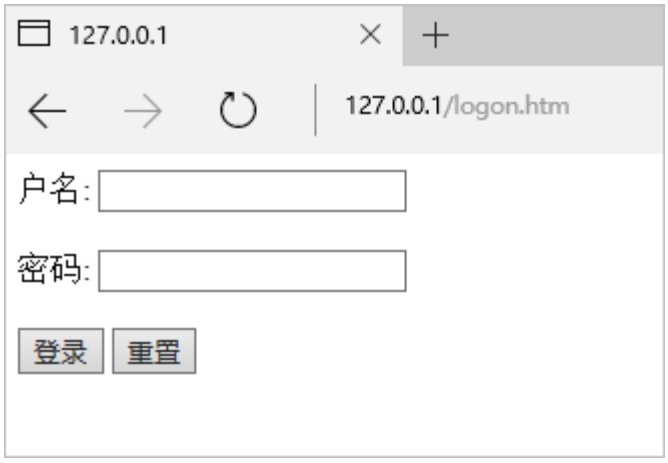
需要注意的是：Cookie 名是区分大小写的。

6.39 用户身份验证

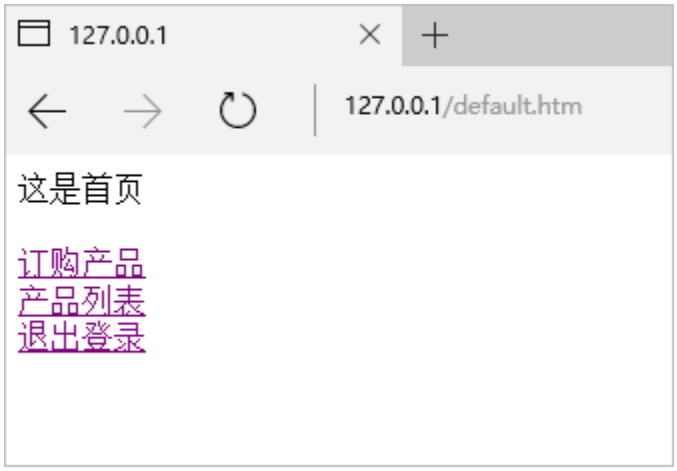
本节的任务是利用上节介绍的Cookie知识，设计一个身份验证功能。

这个系统有五个页面，分别是登录页面logon.htm，退出登录页面exit.htm，以及首页default.htm，还有两个普通页面order.htm和product.htm.

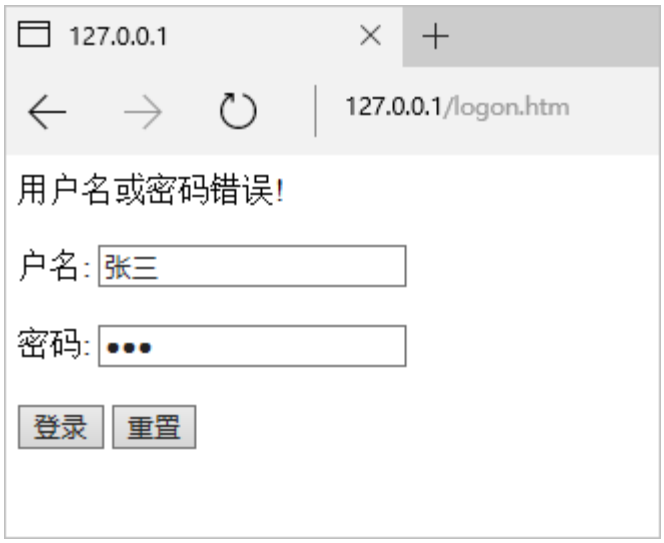
这是登录页面logon.htm：



这是登录成功后的首页default.htm：



如果用户身份验证失败，会重新进入登录页面，并有错误提示：



设计要求:

1、当用户第一次访问时，不管访问任何页面，都会自动跳转到登录页面logon.htm，要求输入账户名和密码 。
2、等用户身份验证通过后，自动跳转到首页default.htm，并可访问任何其他页面。
3、如果用户身份验证失败，会重新进入登录页面logon.htm，并有错误提示。
4、在首页单击"退出登录"，可以清除当前的登录状态，重新进入登录页面。

HttpRequest 事件代码:

```
Dim sb As New StringBuilder
Dim Verified As Boolean
Dim UserName As String = e.Cookies("username") '从 cookie 中获取用户名
Dim Password As String = e.Cookies("password") '从 cookie 中获取用户密码
'如果在登录页面输入了用户名和密码后单击确定按钮
If  e.Path  =  "logon.htm"  AndAlso  e.PostValues.ContainsKey("username")  AndAlso
e.PostValues.ContainsKey("password")  Then
        UserName = e.PostValues("username")
        Password = e.PostValues("password")
End If
'验证用户身份
If UserName = "张三" AndAlso Password = "888" Then
        Verified  = True
ElseIf Username = "李四" AndAlso Password="999" Then
        Verified  = True
End If
If Verified AndAlso e.Path = "logon.htm"  Then '如果用户访问的是登录页,且身份验证成功
        e.Appendcookie("username",UserName) '将用户名和密码写入 cookie
```

```
        e.Appendcookie("password",Password)
        e.WriteString("<meta http-equiv='Refresh' content='0;url=/default.htm'>")'直接跳转到首页
        Return '必须的
ElseIf Verified = False AndAlso e.Path <> "logon.htm" Then '如果用户身份验证失败,且访问的不是登录页面
        e.WriteString("<meta http-equiv='Refresh' content='0; url=/logon.htm'>") '那么直接跳转到登录页面
        Return '必须的
End If
Select Case e.path
        Case "logon.htm"
                sb.AppendLine("<form action='logon.htm' enctype='multipart/form-data' method='post' id='form1'  name='form1'>")
                        If e.PostValues.ContainsKey("username") AndAlso e.PostValues.ContainsKey("password") Then '判断是否是验证失败后的重新登录
                        sb.AppendLine("用户名或密码错误!</br></br>")
                        sb.AppendLine("户名:<input name='username' id='username' value='" & UserName & "''><br/><br/>")
                        sb.AppendLine("密码:<input type='password' name='password' id='password' value ='" & Password & "'><br/><br/>")
                Else
                        sb.AppendLine("  户  名  :  <input  name='username' id='username'><br/><br/>")
                        sb.AppendLine("密码: <input type='password' name='password' id='password'><br/><br/>")
                End If
                sb.AppendLine("<input type='submit' name='sumbit' id='sumbit' value='登录'>")
                sb.AppendLine("<input type='reset' name='reset' id='reset' value='重置'>")
                sb.AppendLine("</form>")
                e.WriteString(sb.ToString)
        Case "exit.htm"
                e.Appendcookie("username", "") '清除 cookie 中原来的用户名和密码
                e.Appendcookie("password", "")
                e.WriteString("<meta http-equiv='refresh' content='0;url=/logon.htm'>")'跳转到登录页
        Case "", "default.htm"
                sb.AppendLine("这是首页<br/><br/>")
                sb.AppendLine("<a href='order.htm'>订购产品<a><br/>")
                sb.AppendLine("<a href='product.htm'>产品列表<a><br/>")
                sb.AppendLine("<a href='exit.htm'>退出登录<a><br/>")
                e.WriteString(sb.Tostring)
        Case "order.htm"
```

```
            e.WriteString("这是订购页")
        Case "product.htm"
            e.WriteString("这是产品页")
End Select
```

代码逻辑并不复杂，所有知识之前都已经讲述过，唯一没有接触过的是自动跳转网页的代码：

```
<meta http-equiv='refresh' content='2; url=/logon.htm'>
```

表示 2 秒后跳转到"/logon.htm"页面，如果你要立即跳转，将 2 改为 0 即可。

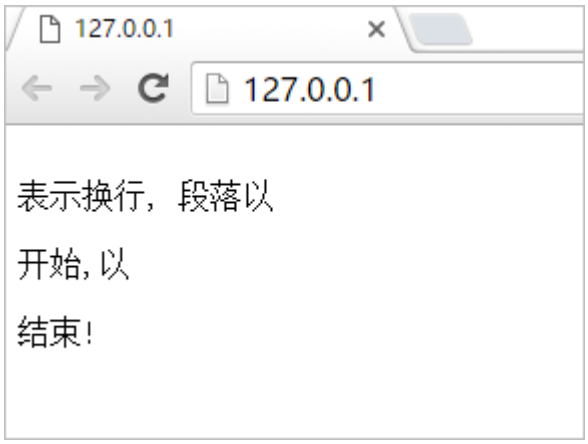
6.40 特殊内容的显示

显示 HTML 标签

如果我们在 HttpRequest 设置以下代码：

```
Dim sb As new StringBuilder
sb.AppendLine("</br>表示换行，段落以<p>开始,以</p>结束!")
e.WriteString(sb.Tostring)
```

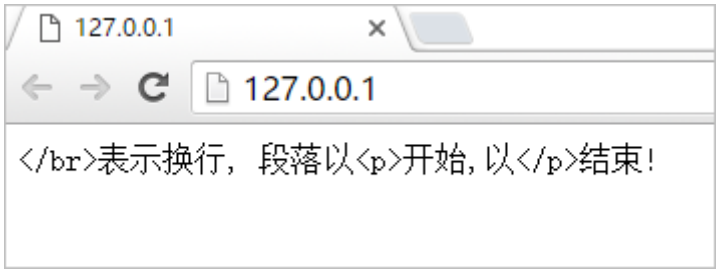
意料之中，浏览器没有按照我们预期的方式显示：



如果要正常显示</br>等 HTML 标签，可以将代码改为：

```
Dim sb As new StringBuilder
sb.AppendLine(HTMLEncode("</br>表示换行，段落以<p>开始,以</p>结束!"))
e.WriteString(sb.Tostring)
```

现在浏览器就能正常显示了：



保持格式显示

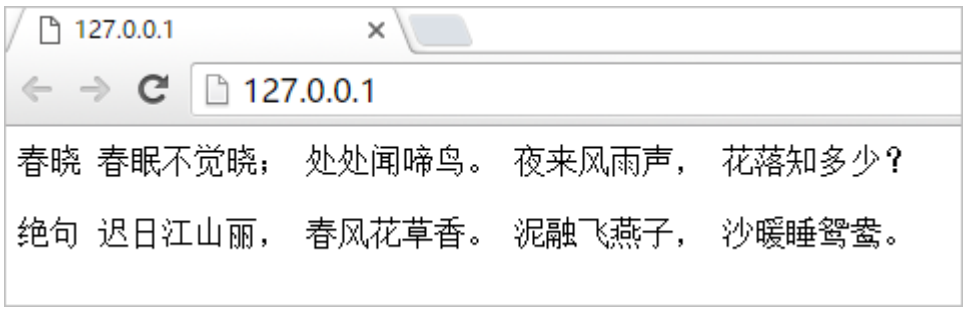
假定有下图所示的一个表：

	作者	内容
1	孟浩然	春晓 春眠不觉晓； 处处闻啼鸟。 夜来风雨声， 花落知多少？
2	杜甫	绝句 迟日江山丽， 春风花草香。 泥融飞燕子， 沙暖睡鸳鸯。

如果我们在HttpRequest设置以下代码：

```
Dim sb As new StringBuilder
For Each dr As DataRow In DataTables("唐诗").DataRows
    sb.Appendline(dr("内容"))
    sb.AppendLine("</br></br>")
Next
e.WriteString(sb.Tostring)
```

不出意料，浏览器的没有按照我们希望的格式显示：

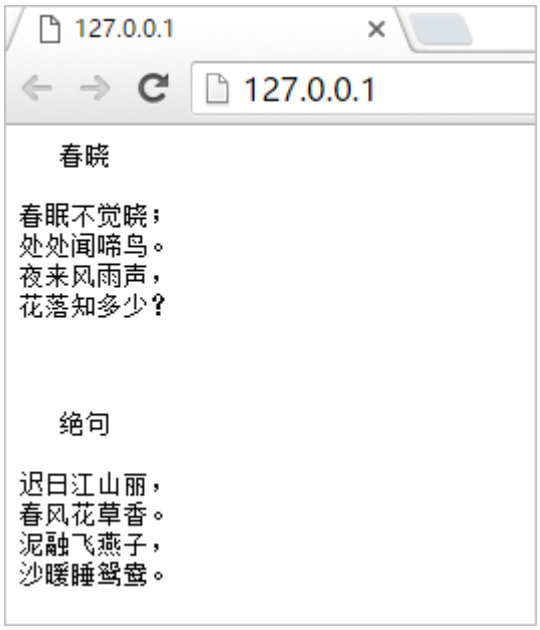


这是因为浏览器在解析内容时，会将多个连续的空格或换行，转换为一个空格。

如果希望按照原来的格式显示，可以将代码改为：

```
Dim sb As new StringBuilder
For Each dr As DataRow In DataTables("唐诗").DataRows
    sb.Appendline("<pre>" & dr("内容") & "</pre>")
    sb.AppendLine("</br>")
Next
e.WriteString(sb.Tostring)
```

现在就可以按照原格式显示了：



提示：<pre>和</pre>之间的内容，会按照原格式显示。

6.41 Excel 报表与后台数据

如果你是一个网页设计高手，那么网页的数据展示能力是非常优秀的。

但绝大多数用户，包括我自己，都无法设计出高水准的网页，所以需要一个简单的替代工具，实现复杂数据的展示，特别是手机端的复杂数据展示。
经过综合考虑，我们选择了 Excel 报表来完成这项任务，但是在 Foxtable 2017 之前，Excel 报表只能根据已经加载的数据生成，而作为服务端程序，通常是没有办法将所有数据加载的。
所以，我们对 Excel 报表进行了改进，使得其可以直接基于后台数据生成报表。
我首先要提醒大家一下：

服务端的动态报表功能要尽量采用 Excel 报表实现，不要尝试在服务端动态生成 Word 报表，也不要尝试将报表保存为 PDF 文件，这些操作的时间和资源的消耗都非常大，是 Excel 报表的百倍以上，不适合在服务端使用。

一个例子

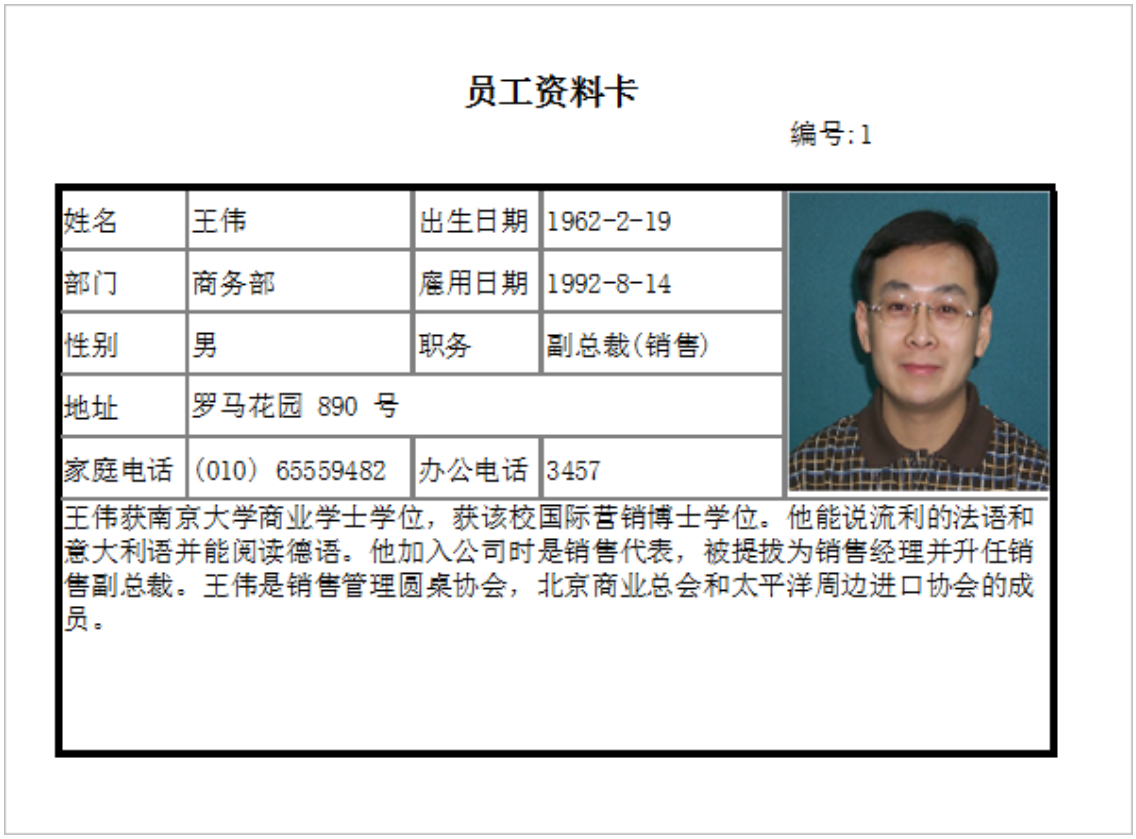
如果你设计了一个下图所示的 Excel 报表模板：

	A	B	C	D	E	F	G
1							<End>
2	<员工, 2>	员工资料卡					
3						<"编号:" & [编号]>	
4							
5		姓名	[姓名]	出生日期	[出生日期]	[&照片, 3, 3, 150, 159]	
6		部门	[部门]	雇用日期	[雇佣日期]		
7		性别	[性别]	职务	[职务]		
8		地址	[地址]				
9		家庭电话	[家庭电话]	办公电话	[办公电话]		
10		[备注]					
11							
12	<员工>						
13	<End>						

假定系统并没有加载员工表，或者员工表只加载了部分数据，我们用下面的代码，一样可以从后台提取未加载的数据来生成报表，而且非常简单：

```
Dim Book As New XLS.Book(ProjectPath & "Attachments\资料卡.xls")
Dim fl As String = ProjectPath & "Reports\资料卡.xls"
book.AddDataTable("员工","数据源名称","Select * from {员工} where  姓名  = '王伟'")
Book.Build()
Book.Save(fl)
Dim Proc As New Process
Proc.File = fl
Proc.Start()
```

这是生成的报表：



员工资料卡

编号:1

姓名	王伟	出生日期	1962-2-19	
部门	商务部	雇用日期	1992-8-14	
性别	男	职务	副总裁(销售)	
地址	罗马花园 890 号			
家庭电话	(010) 65559482	办公电话	3457	

王伟获南京大学商业学士学位，获该校国际营销博士学位。他能说流利的法语和意大利语并能阅读德语。他加入公司时是销售代表，被提拔为销售经理并升任销售副总裁。王伟是销售管理圆桌协会，北京商业总会和太平洋周边进口协会的成员。

提示：

AddDataTable方法用于从后台提取数据生成一个临时表，由Excel报表模板根据此临时表生成报表。例如上面的代码中，AddDataTable利用Select语句生成了一个临时表，表名为"员工"。
系统在生成Excel报表的过程中，优先调用AddDataTable生成的临时表，无需担心和现有同名表产生冲突。

AddDataTable的语法为：

AddDataTable(Name,DataSouce,SelectString)
AddDataTable(Name,DataTable)

Name	字符型，临时表的名称，必须和报表模板中的表名保持一致。
DataSouce	字符型，用于指定数据源名称。
SelectString	字符型，用于指定 Select 语句
DataTable	DataTable 型，用于直接指定一个临时 DataTable 作为报表的数据来源，此 DataTable 不能是 DataTables 中的一员。 GroupTableBuilder 、 CrossTableBuilder 、 SQLGroupTableBuilder 、 SQLCrossTableBuilder、SQLJoinTableBuilder 和 DataTableBuilder 的 Build 方法，都有一个逻辑参数，将此参数设置为 True，将生成一个 DataTable，此 DataTable 将不会包括在 DataTables 中。 例如：

```
Dim Book As New XLS.Book(ProjectPath & "Attachments\test.xls")
Dim fl As String = ProjectPath & "Reports\test.xls"
Dim g As New SQLGroupTableBuilder("统计表 1", "订单")
g.ConnectionName = "数据源名称"
g.Groups.AddDef("产品")
g.Totals.AddDef("数量")
book.AddDataTable("统计表 1", g.Build(True)) '添加临时表
Book.Build()
Book.Save(fl)
Dim Proc As New Process
Proc.File = fl
Proc.Start()
```

多个数据表和关联

我们可以添加多个临时表，而且可以在临时表之间建立关联。

假定设计了下图所示的报表模版，用于生成出库单。

	A	B	C	D	E	F	G	H
1								<END>
2	<出库,1>	湛江佳信公司出库单						
3								
4		<"出库单号:" & [出库单编号]>						
5		<"领用部门:" & [领用部门] & " 出库日期:" & [出库日期]>					<"领料人:" & [领料人]>	
6	<HeaderRow>	商品代码	商品名称	规格	单位	数量	备注	
7	<出库明细,-9>	[商品代码]	[商品名称]	[规格]	[单位]	[数量]	[备注]	
8	<出库>	<"库管员:" & [库管员]>			合计:	[%出库明细,Sum(数量)]		
9	<END>							
10								

此模版涉及到出库和出库明细两个表，这两个表通过出库单号建立关联。

不管系统是否已经加载了这两个表，我们都可以通过下面的代码生成一个出库单：

```
Dim Book As New XLS.Book(ProjectPath & "Attachments\出库单.xls")
Dim fl As String = ProjectPath & "Reports\出库单.xls"
book.AddDataTable("出库","数据源名称","Select * from {出库} where 出库单编号='CK-20030726001'") '添加父表
book.AddDataTable("出库明细","数据源名称","Select * from {出库明细} where 出库单编号='CK-20030726001'") '添加子表
book.AddRelation("出库","出库单编号","出库明细","出库单编号") '建立关联
Book.Build() '生成细节区
Book.Save(fl) '保存工作簿
Dim Proc As New Process '打开工作簿
Proc.File = fl
Proc.Start()
```

这是生成的报表：



湛江佳信公司出库单

出库单号:CK-20030726001

领用部门:销售部　　出库日期:2008-07-26　　领料人:林海

商品代码	商品名称	规格	单位	数量	备注
HC00101	打印纸	80列	箱	50	
HC00102	打印纸	40列	箱	20	

库管员:陶正东　　合计：70

提示： AddRelation 方法用于在临时表之间建立关联。

AddRelation 的语法为：

AddRelation(ParentTable,ParentCol,ChildTable,ChildCol)

ParentTable	字符型，父表名称。
ParentCol	字符型，父表关联列名称，如果有多个关联列，可以用一个字符型数组表示。
ChildTable	字符型，子表名称。
ChildCol	字符型，子表关联类名称，如果有多个关联列，可以用一个字符型数组表示。

6.42 提升 Excel 报表效率

我们通常会采用下面的 HttpRequest 事件代码来响应用户访问 Excel 报表的请求：

```
Select Case e.path
    Case "ckd.xls"
        Dim Book As New XLS.Book(ProjectPath & "Attachments\出库单.xls")
        Dim fl As String = ProjectPath & "Reports\出库单.xls"
        book.AddDataTable("出库","数据源名称","Select * from {出库} where 出库单编号='CK-20030726001'") '添加父表
        book.AddDataTable("出库明细","数据源名称","Select * from {出库明细} where 出库单编号='CK-20030726001'") '添加子表
        book.AddRelation("出库","出库单编号","出库明细","出库单编号") '建立关联
        Book.Build()
        Book.Save(fl)
        e.WriteFile(fl)
End Select
```

过程很简单：基于模版生成 Excel 报表，将生成的报表保存为一个文件，然后用 WriteFile 方法将文件发送到客户端的浏览器。

这样的设计有以下不足：

1、虽然 WriteFile 是异步执行的，但是 Excel 报表的生成和保存却是和主线程同步的，且 Excel 报表的生成并非很高效，所以用户量比较大的时候，会影响效率。

2、整个过程要分别保存和读取一次文件，我们知道计算机性能的瓶颈就是硬盘的读写，所以这同样会影响效率。

3、由于要保存为实际的文件，所以容易出现文件名冲突，假定有多个用户同时访问，上面的代码肯定会出错，虽然可以采用一些手段避免同名冲突，但毕竟增加了工作量。

WriteBook

为彻底解决上述问题，我们为 HttpRequest 事件增加了一个 WriteBook 方法，此方法专门用于发送 Excel 报表，其语法为：

WriteBook(Book,FileName，InLine)

Book	要发送的 Excel 报表
FileName	客户端浏览器下载此报表时使用的文件名
InLine	可选参数，逻辑型，是否直接在浏览器显示报表，默认为 True，设为 False 将下载报表。 实际上除了 iOS 设备，其他设备不管如何设置，都会下载报表。

WriteBook 是异步执行的，而且在发送报表之前，还是会异步执行 Build 方法生成报表，所以如果不是 Excel 报表模板，请将 Book 的 PreBuild 属性设置为 False，避免 WriteBook 执行 Build 方法。

一个例子

HttpRequest 事件代码：

```
Select Case e.path
    Case "ckd.xls" '直接在浏览器显示
        Dim Book As New XLS.Book(ProjectPath & "Attachments\出库单.xls")
        book.AddDataTable("出库","数据源名称","Select * from {出库} where 出库单编号='CK-20030726001'") '添加父表
        book.AddDataTable("出库明细","数据源名称","Select * from {出库明细} where 出库单编号='CK-20030726001'") '添加子表
        book.AddRelation("出库","出库单编号","出库明细","出库单编号") '建立关联
        e.WriteBook(book,"出库单.xls",True)
    Case "emp.xls" '下载为文件
        Dim Book As New XLS.Book(ProjectPath & "Attachments\资料卡.xls")
        book.AddDataTable("员工","数据源名称","Select * from {员工} where 姓名 ='王伟'")
        e.WriteBook(book,"emp.xls",False)
    Case "orders.htm" '请求的是网页，返回的是Excel文件
        Dim nms() As String = {"产品","客户","数量","单价","金额","日期"}
        Dim cmd As New SQLCommand
        cmd.ConnectionName = "数据源名称"
        cmd.CommandText = "Select 产品, 客户, 数量, 单价, 数量 * 单价 As 金额, 日期 From {订单} Where 日期 =#" & Date.Today & "#"
        Dim dt As DataTable = cmd.ExecuteReader
        Dim Book As New XLS.Book
```

```
    Dim Sheet As XLS.Sheet = Book.Sheets(0)
    For c As Integer = 0 To nms.Length - 1
      Sheet(0, c).Value = nms(c)
    Next
    For r As Integer = 0 To dt.DataRows.Count - 1
      For c As Integer = 0 To nms.Length - 1
        Sheet(r + 1, c).Value = dt.Datarows(r)(nms(c))
      Next
    Next
    book.PreBuild = False '非报表模请将PreBuild 属性设置为False
    e.WriteBook(book,"订单.xls",True)
End Select
```

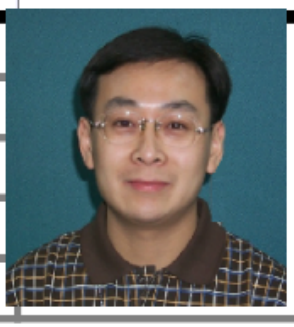
提示：

1、InLine参数仅对iOS设备有效。
2、非报表模板请将Book的PreBuild参数设置为False。
3、FileName参数可以和客户请求的文件名不同，甚至可以在客户请求一个网页时，返回一个excel文件。

尽管iOS设备可以在浏览器直接查看Office文档，但是兼容性却不好，例如上面的第二个报表，分别在iOS上的浏览器和iOS上的WPS显示，存在很大的差异：

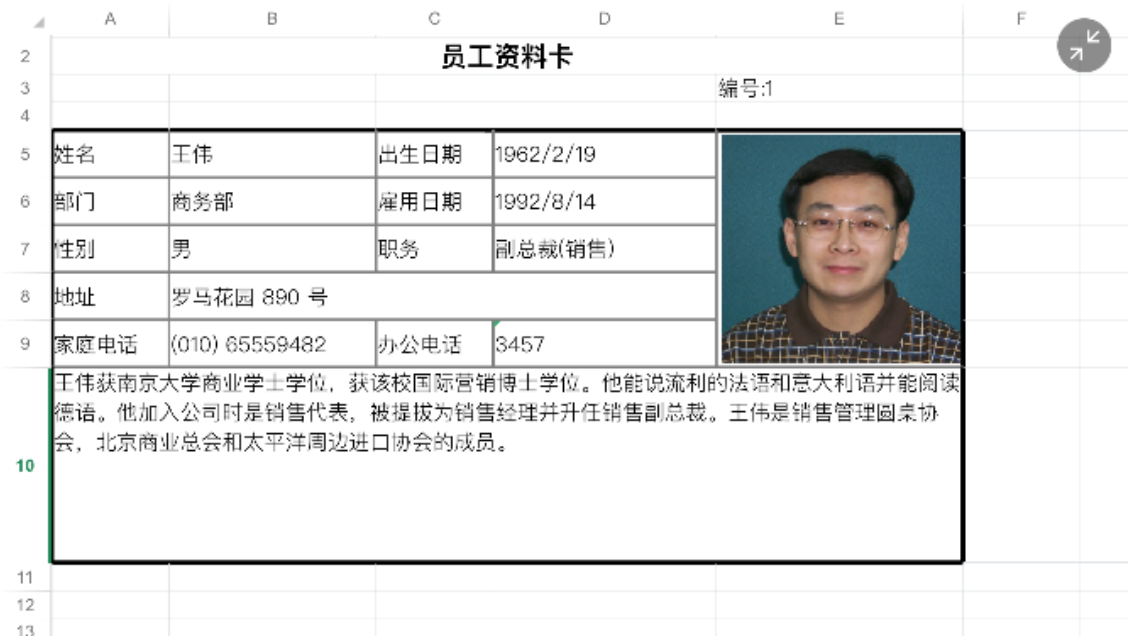
iOS上的浏览器显示为：

姓名	王伟	出生日期	2/19/62
部门	商务部	雇用日期	8/14/92
性别	男	职务	副总裁(销售)
地址	罗马花园 890 号		
家庭电话	(010) 65559482	办公电话	3457



王伟获南京大学商业学士学位，获该校国际营销博士学位。他能说流利的法语和意大利语并能阅读德语。他加入公司时是销售代表，被提拔为销售经理并升任销售副总裁。王伟是销售管理圆桌协会，北京商业总会和太平洋周边进口协会的成员。

iOS上的WPS显示为：



员工资料卡

编号:1

姓名	王伟	出生日期	1962/2/19
部门	商务部	雇用日期	1992/8/14
性别	男	职务	副总裁(销售)
地址	罗马花园 890 号		
家庭电话	(010) 65559482	办公电话	3457

王伟获南京大学商业学士学位，获该校国际营销博士学位。他能说流利的法语和意大利语并能阅读德语。他加入公司时是销售代表，被提拔为销售经理并升任销售副总裁。王伟是销售管理圆桌协会，北京商业总会和太平洋周边进口协会的成员。

6.43 适合在手机显示的网页

如果将 HttpRequest 事件的代码设置为：

```
Dim sb As New StringBuilder
sb.appendLine("<form action='showtj.htm' enctype='multipart/form-data' method='post' id='form1' name='form1'>")
sb.appendLine("请输入要统计的年月:</br></br>")
sb.appendLine("年: <input type='number' name='year' id='year' min='1999' max='2018'><br/><br/>")
sb.appendLine("月: <input type='number' name='month' id='month' min='1' max='12'><br/><br/>")
sb.appendLine("<input Type='submit' name='Sumbit' id='Sumbit' value='开始统计'>")
sb.appendLine("</form>")
e.WriteString(sb.ToString)
```

在iPhone上用QQ浏览器访问，显示效果为：



文字很小，你要放大才能看清楚内容。

为解决这个问题，可以将代码改为：

```
Dim sb As New StringBuilder
sb.AppendLine("<meta  name='viewport'  content='width=device-width,initial-scale=1,user-scalable=0'>")
sb.appendLine("<form action='showtj.htm' method='post' name='form1'>")
sb.appendLine("请输入要统计的年月:</br></br>")
sb.appendLine("年: <input type='number' name='year' id='year' min='1999' max='2018'><br/><br/>")
sb.appendLine("月: <input type='number' name='month' id='month' min='1' max='12'><br/><br/>")
sb.appendLine("<input Type='submit' name='Sumbit' id='Sumbit' value='开始统计'>")
```

```
sb.appendLine("</form>")
e.WriteString(sb.ToString)
```

现在在iPhone上用QQ浏览器访问，可以正常显示了：





这里给网页加上了一行代码：

```
<meta name='viewport' content='width=device-width,initial-scale=1,user-scalable=0'>
```

viewport的content属性中可设置的值有：

initial-scal: 默认缩放比例。
maximum-scale: 允许用户缩放到的最大比例。
minimum-scale: 允许用户缩放到的最小比例。
user-scalable: 用户是否可以手动缩放，0禁止，1允许。
width: 视区宽度,以像素为单位的数字，设置为device-width表示整个设备的宽度。
height: 视区高度,以像雾为单位的数字，设置为device-height表示整个设备的高度。

这里要说明一下，我为什么要在手机上用QQ浏览器进行测试，这是因为微信已经是很多网页的入口，而QQ浏览其和微信内置的浏览器出自同门，显示效果是一样的。

规范结构

上述的代码虽然能在浏览器中正常显示，但实际上是不规范的，因为meta标签应该放在网页的head标签中：

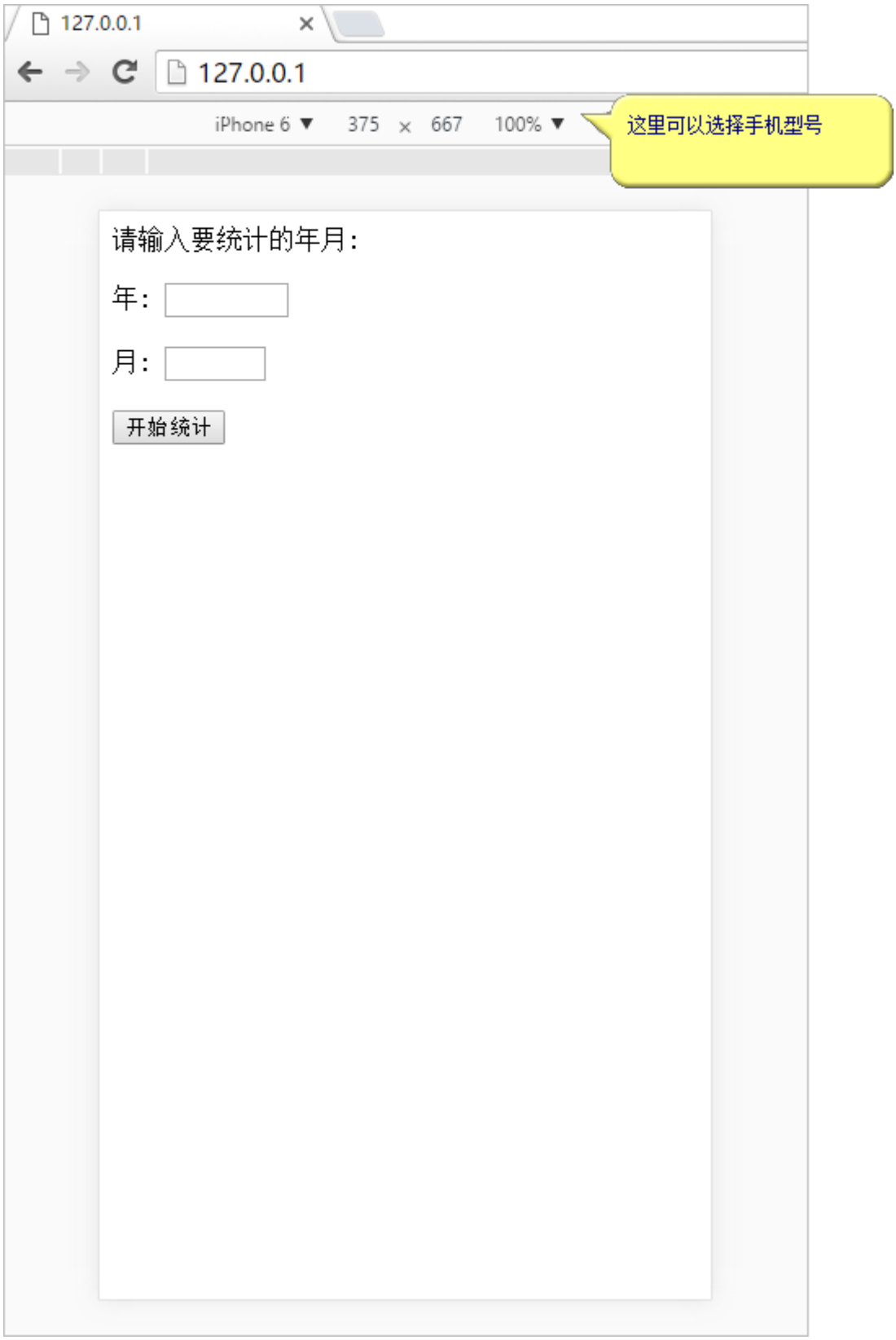
```
<head>
<meta name='viewport' content='width=device-width,initial-scale=1,user-scalable=1'>
</head
```

而网页内容应该放在<body>标签中，所以我们修改一下代码：

```
Dim sb As New StringBuilder
sb.appendLine("<!doctype html>")
sb.appendLine("<html>")
sb.appendLine("<head>")
sb.AppendLine("<meta  name='viewport'  content='width=device-width,initial-scale=1,user-scalable=1'>")
sb.appendLine("</head>")
sb.appendLine("<body>")
sb.appendLine("<form action='showtj.htm' method='post' name='form1'>")
sb.appendLine("请输入要统计的年月:</br></br>")
sb.appendLine("年: <input type='number' name='year' id='year' min='1999' max='2018'><br/><br/>")
sb.appendLine("月: <input type='number' name='month' id='month' min='1' max='12'><br/><br/>")
sb.appendLine("<input Type='submit' name='Sumbit' id='Sumbit' value='开始统计'>")
sb.appendLine("</form>")
sb.appendLine("</body>")
sb.appendLine("</html>")
e.WriteString(sb.ToString)
```

用Chorme模拟手机浏览

建议在开发电脑上，安装谷歌的Chorme浏览器。
在新版的Chorme中，按“Ctrl+Shift+I”，会进入开发者工具界面，在这个界面中，可以模拟网页在各种移动设备下的显示效果：


127.0.0.1
127.0.0.1
iPhone 6
375
667
100%
这里可以选择手机型号
请输入要统计的年月:
年:
月:
开始统计

6.44 使用框架生成网页

其实即使是网页设计高手，想要好的设计效果，多数也是要使用框架的，至于菜鸟，更是离不开框架。

你可以使用第三方的框架，目前用得比较多的是 jQuery，在这个基础上还有两个官方维护的界面框架，分别是 jQueryUI 和 jQuery Mobile，前者适合常规网页设计，后者适合手机端网页设计。

第三方框架的一般都是有 js 和 css 文件组成，如何使用，我们之前已经介绍，这里就不重复了，参考：使用 JavaScript 文件　使用 CSS 文件

FoxTable 内置了一个框架，这个框架基于腾讯的微信网页开发样式库(WeUI)，选择这个框架，不是因为这个框架有多么的优秀，是为了让你开发的网页未来能更好地和微信整合，毕竟都是腾讯的产品。

我们用一个简单的例子来说明使用这个框架的步骤：

1、准备 WeUI 框架文件

你可以在以下地址下载到 WeUI 框架文件：

http://www.foxtable.com/download/mobile/weui.zip

解压会得到三个文件，分别是：

weui.me.js

weui.me.css

weui.min.css

解压后将上述三个文件复制到"d:\web\weui"目录下

2、用以下代码开启 http 服务网：

```
HttpServer.Prefixes.Add("http://*/")
HttpServer.WebPath = "d:\web"  '指定静态文件存储位置
HttpServer.Start()
```

重要提示，必须将 HttpServer 的 WebPath 设置未正确的路径，否则框架将不会生效。

3、将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "addnew.htm"
        Dim wb As New WeUI '定义一个基于 weui 框架的网页生成器
        wb.AddForm("","form1","addnew.htm")
        With wb.AddInputGroup("form1","ipg1","新增订单")
            .AddSelect("cp","产品","PD01|PD02|PD03|PD04|PD05")
            .AddInput("gy","雇员","text")
            .AddInput("kh","客户","text")
            .AddInput("dj","单价","number")
            .AddInput("zk","折扣","number")
            .AddInput("sl","数量","number")
            .AddInput("rq","日期","date")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
```

```
            .Add("btnok","确定")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

3、现在通过手机访问，可以得到下图所示的网页：



你可以通过网页浏览器查看上述网页的源代码，合计有 73 行，而 Foxtable 的 Httprequest 事件只用了 14 行代码就生成了这个网页。

节省的代码量超过了 80%，更重要的是，上述代码简洁易懂，不涉及任何网页设计知识，完全是傻瓜式的，而且网页效果很专业，一般用户完全靠自己编码，可能很难做出这样的效果。

6.45 去掉通用事件头

我们知道，很难严格区分静态网页和动态网页，因为动态生成的网页，也需要使用不少静态的文件，例如图片、js、css 等等。

所以我们总是在 HtttpRequest 事件的开始位置加上代码：

```
'通用事件头,用于发送已经存在的常见文件
Dim fl As String = "d:\web\" & e.path
If filesys.FileExists(fl)
    Dim idx As Integer = fl.LastIndexOf(".")
    Dim ext As String   = fl.SubString(idx)
    Select Case ext
      Case ".jpg",".gif",".png",".bmp",".wmf",".js",".css" ,".html",".htm",".zip",".rar"
        e.WriteFile(fl)
        Return '这里必须返回
    End Select
End If
'以下是动态生成网页的代码
'...
```

这有点繁琐，实际上这个事件头可以去掉的，只需将 Httprequest 事件代码改为：

```
HttpServer.Prefixes.Add("http://*/")
HttpServer.WebPath = "d:\web"  '指定静态文件存储位置
HttpServer.Start()
```

现在 HtttpRequest 不需要任何代码， 当用户访问某个文件时，系统会自动判断"d:\web"目录是否存在此文件，如果存在，则自动发送此文件，否则触发 HttpRequet 事件。

我们只用了 3 行代码，这应该是史上最简单的 web 服务搭建方法了。

从现在开始，我们的 HttpRequest 事件代码将不再包括通用事件头了，我们假定你已经使用上述代码启动了 Web 服务。

关于后缀名

如果给 HttpServer 设置了 WebPath 属性，在默认情况下，系统会发送该目录下以下类型的文件：

.jpg
.gif
.png
.bmp
.wmf
.js
.css
.html
.htm
.zip
.rar
.txt
.json

.svg
.ttf
.woff
.woff2
.eot
.ico
.map
.doc
.docx
.xls
.xlsx

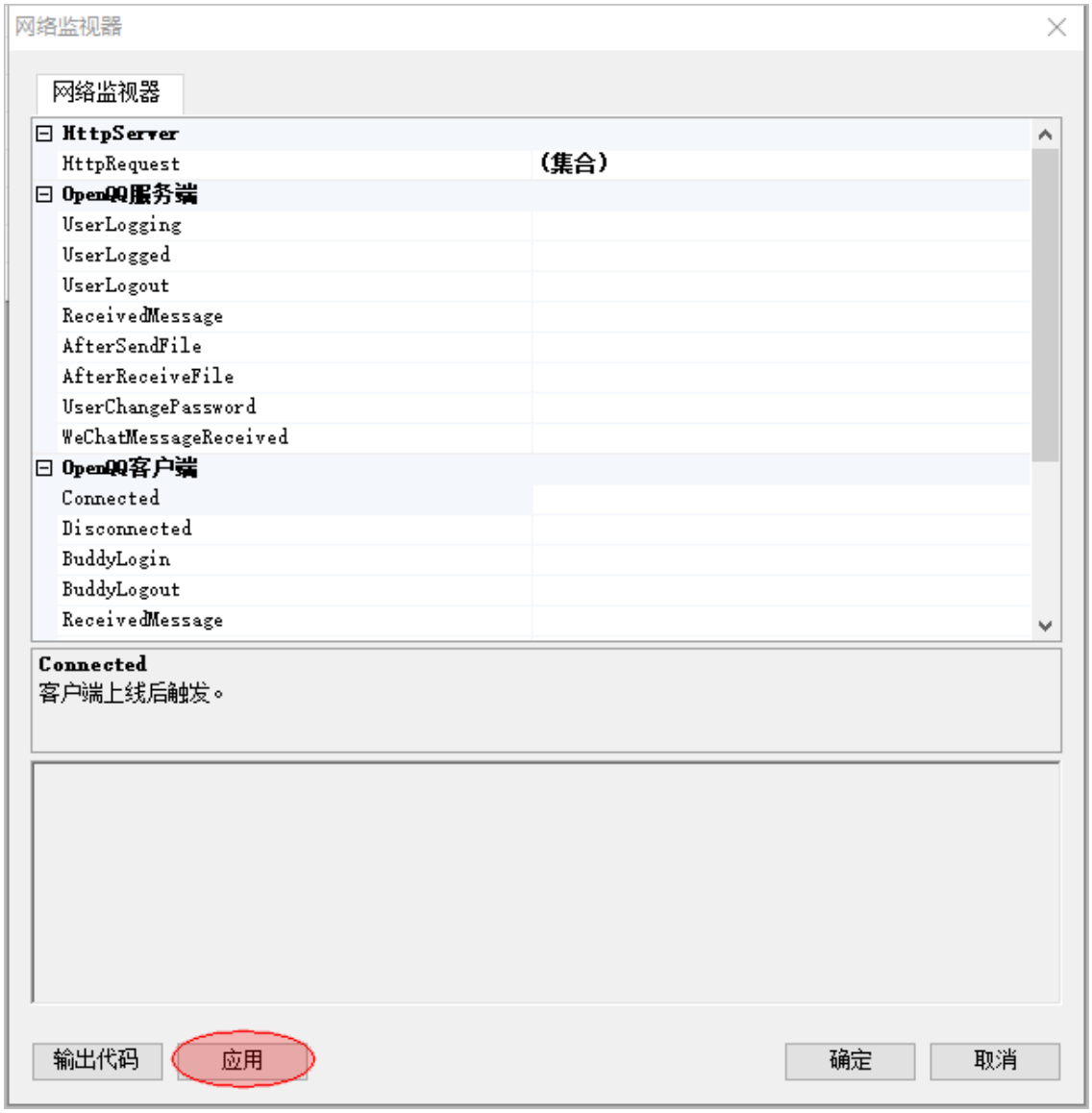
HttpServer 还有一个 Extensions 属性，这是一个集合，用于管理可发送文件的后缀名，我们可以根据需要添加删除后缀名，例如：

```
HttpServer.Prefixes.Add("http://*/")
HttpServer.WebPath = "d:\web"
HttpServer.Extensions.Remove(".doc")
HttpServer.Extensions.Remove(".docx")
HttpServer.Extensions.add(".table")
HttpServer.Start()
```

提示：代码中的后缀名必须是小写，且必须以符号"."开头。

6.46 提高编码效率

在开发 B/S 系统时，需要频繁通过菜单打开网络监视器，编写 HttpRequest 事件代码。
因为架构问题，打开网络监视器会有几秒的延时，对于开发效率会有一些影响，为解决这个问题，我们特意在网络监视器窗口加上了应用按钮：



单击此按钮，相关事件代码即时生效，无需关闭网络监视器，这样再次修改代码，就节省了打开网络监视器的时间。

此外，我们在自定义函数窗口也加上了应用按钮。

7. WeUI 框架

7.1 使用框架生成网页

其实即使是网页设计高手，想要好的设计效果，多数也是要使用框架的，至于菜鸟，更是离不开框架。

你可以使用第三方的框架，目前用得比较多的是 jQuery，在这个基础上还有两个官方维护的界面框架，分别是 jQueryUI 和 jQuery Mobile，前者适合常规网页设计，后者适合手机端网页设计。

第三方框架的一般都是有 js 和 css 文件组成，如何使用，我们之前已经介绍，这里就不重复了，参考：使用 JavaScript 文件 使用 CSS 文件

FoxTable 内置了一个框架，这个框架基于腾讯的微信网页开发样式库(WeUI)，选择这个框架，不是因为这个框架有多么的优秀，是为了让你开发的网页未来能更好地和微信整合，毕竟都是腾讯的产品。

我们用一个简单的例子来说明使用这个框架的步骤：

1、准备 WeUI 框架文件

你可以在以下地址下载到 WeUI 框架文件：

http://www.foxtable.com/download/mobile/weui.zip

解压会得到三个文件，分别是：

weui.me.js

weui.me.css

weui.min.css

解压后将上述三个文件复制到"d:\web\weui"目录下

2、用以下代码开启 http 服务网：

```
HttpServer.Prefixes.Add("http://*/")
HttpServer.WebPath = "d:\web"  '指定静态文件存储位置
HttpServer.Start()
```

重要提示，必须将 HttpServer 的 WebPath 设置未正确的路径，否则框架将不会生效。

3、将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "addnew.htm"
        Dim wb As New WeUI '定义一个基于 weui 框架的网页生成器
        wb.AddForm("","form1","addnew.htm")
        With wb.AddInputGroup("form1","ipg1","新增订单")
            .AddSelect("cp","产品","PD01|PD02|PD03|PD04|PD05")
            .AddInput("gy","雇员","text")
            .AddInput("kh","客户","text")
            .AddInput("dj","单价","number")
            .AddInput("zk","折扣","number")
            .AddInput("sl","数量","number")
            .AddInput("rq","日期","date")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btnok","确定")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

3、现在通过手机访问，可以得到下图所示的网页：



你可以通过网页浏览器查看上述网页的源代码，合计有 73 行，而 Foxtable 的 Httprequest 事件只用了 14 行代码就生成了这个网页。
节省的代码量超过了 80%，更重要的是，上述代码简洁易懂，不涉及任何网页设计知识，完全是傻瓜式的，而且网页效果很专业，一般用户完全靠自己编码，可能很难做出这样的效果。

7.2 请输入正确网址

在很大程度上而言，本节内容真的是多余，输入正确网址难道也需要专门提醒吗？

可就是有用户犯这样的错误，而且还不少，^_^

以我们上一节的例子而言，HttpRequest 事件代码为：

```
Select Case e.Path
    Case "addnew.htm"
        Dim wb As New WeUI '定义一个基于 weui 框架的网页生成器
        wb.AddForm("","form1","addnew.htm")
        With wb.AddInputGroup("form1","ipg1","新增订单")
            .AddSelect("cp","产品","PD01|PD02|PD03|PD04|PD05")
            .AddInput("gy","雇员","text")
            .AddInput("kh","客户","text")
            .AddInput("dj","单价","number")
            .AddInput("zk","折扣","number")
            .AddInput("sl","数量","number")
            .AddInput("rq","日期","date")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btnok","确定")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

那么测试的时候，应该输入：

http://ip 地址/addnew.htm

如果是本机测试，就是：

http://127.0.0.1/addnew.htm

相当多的用户只是简单的输入 ip 地址，不输入后面的网页名称，导致网页内容不能显示，然后很肯定地认为框架有问题。
如果你想直接输入 ip 或者域名就能访问某个页面，请参考下一节设置默认页面

7.3 设置默认网页

一个网页管理系统会有多个网页，其中一个只需输入 IP 或者域名就能访问的网页，称为默认网页。

设置默认网页很简单，e.path 为空的时候，表示要访问默认网页。

例如 HttpRequest 事件代码：

```
Dim wb As New WeUI '定义一个基于 weui 框架的网页生成器
Select Case e.Path
```

```
    Case "addnew.htm",""
        wb.InsertHTML("这是默认页面 addnew.htm")
    Case "order.htm"
        wb.InsertHTML("这是普通页面 order.htm")
End Select
e.WriteString(wb.Build) '生成网页
```

上面的代码生成了两个网页，其中 addnew.htm 是默认页面，可以直接通过 ip 访问，也可以输入完整路径：

http://127.0.0.1/
http://127.0.0.1/addnew.htm

另一个页面 order.htm，是普通页面，必须输入完整路径才能访问：

http://127.0.0.1/order.htm

7.4 用 Chorme 模拟手机

WeUI 框架主要是针对手机的，如果每次修改 HttpRequest 代码后，都要通过手机访问来预览效果，可能会影响开发效率。

建议在开发电脑上，安装谷歌的 Chorme 浏览器。

在新版的 Chorme 中，按“Ctrl+Shift+I”，会进入开发者工具界面，在这个界面中，可以模拟网页在各种移动设备下的显示效果：



当然电脑上模拟的和手机上实际显示的效果，还是会有一些细节差异的。

7.5 定义表单

使用 WeUI 定义表单的语法很简单：

AddForm(ParentID, ID, Action)

ParentID　　父容器的 ID,如果是顶层对象，设置为""即可

ID	表单 ID
Action	接收表单数据的网页。

示例

将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui '定义一个基于 weui 框架的网页生成器
        wb.AddForm("","form1","accept.htm")
        e.WriteString(wb.Build) '生成网页
End Select
```

表示新增一个表单，表单的名称为 form1，接收数据的网页为"accpet.htm"

7.6 定义按钮

在 WeUI 定义按钮之前，首先要定义一个按钮分组，定义按钮分组的语法为：

AddButtonGroup(ParentID, ID)

AddButtonGroup(ParentID, ID, Vertical)

ParentID	父容器的 ID,如果是顶层对象，设置为""即可
ID	按钮分组 ID
Vertical	可选参数，逻辑型，按钮是否垂直排列，默认为 True。

添加按钮的语法为：

Add(ID, Text)

Add(ID, Text, Type)

Add(ID, Text, Type, Href)

ID	按钮 ID
Text	按钮标题
Type	可选参数，字符型，用于指定按钮类型，可用值有"submit"、"reset"、"button"
Href	可选参数，字符型，指定单击按钮之后跳转的目标网页的 URL

按钮有两个属性，分别为：

Kind	用于设置按钮的颜色，默认为 0，表示绿色，1 表示灰色，2 表示红色
Value	用于设置按钮的值，

示例

将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "test.htm"
```

```
    Dim wb As New weui
    wb.AddForm("","form1","test.htm")
    With wb.AddButtonGroup("form1","btg1",True)  '垂直排列
      .Add("btn1", "按钮")
      .Add("btn4", "按钮", "reset")
      .Add("btn5", "按钮", "", "http://www.foxtable.com") '单击这个按钮可以打开 foxtable 主
页
    End With
    With wb.AddButtonGroup("form1","btg2", False)  '水平排列
      .Add("btn6", "按钮")
      .Add("btn7", "按钮").Kind = 2
    End With
    With wb.AddButtonGroup("form1","btg3", False)
      .Add("btn8", "按钮").Kind = 0
      .Add("btn9", "按钮").Kind = 1
    End With
    e.WriteString(wb.Build) '生成网页
End Select
```

现在通过手机访问，可以看到下图所示的网页：



7.7 文本输入框

在定义输入框之前，首先你得定义一个输入框组(InputGroup)，定义输入框组的语法为：

AddInputGroup(ParentID, ID)

AddInputGroup(ParentID, ID, Text)

ParentID	父容器的 ID,如果是顶层对象，设置为""即可。
ID	分组 ID。
Text	可选参数，用于指定分组标题。

增加文本输入框的语法为：

AddInput(ID, Label, type)

ID	输入框 ID,注意生成网页时，WeUI 会自动将所有 ID 转换为小写，例如"Table1"或转换为"table1"，今后不再重复提示，大家请自行留意。
Label	在输入框左侧显示的标签内容

Type　　输入框类型，普通文本输入框设置为"text"，密码输入框设置为"password"

通过 AddInput，还可以增加日期输入框和数值输入框。

示例

将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1","登录")
            .AddInput("xm","户名","text")
            .AddInput("pw","密码","password")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

现在通过手机访问，可以看到下图所示的网页：



属性

Foxtable为输入框提供了以下属性：

Value	字符型，输入框的初始值
Readonly	逻辑型，输入框是否只读
Post	逻辑型，提示表单数据时是否包括此输入框的值，默认为 True。
Required	逻辑型，是否必须输入内容，目前基于 iOS 的多数浏览器暂不支持这个属性。
Placeholder	字符型，对输入框预期值的提示

将HttpRequest事件代码设置为：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
```

```
    wb.AddForm("","form1","test.htm")
    With wb.AddInputGroup("form1","ipg1","登录")
      With .AddInput("xm","户名","text")
        .Value = "张三"
        .Readonly= True
      End With
      .AddInput("xm","密码","password").Placeholder = "请输入6位数密码"
    End With
    With wb.AddButtonGroup("form1","btg1",True)
      .Add("btn1", "确定", "submit")
    End With
    e.WriteString(wb.Build) '生成网页
End Select
```

现在通过手机访问，可以看到下图所示的网页 ，其中户名输入框是只读的：



7.8 日期输入框

和文本输入框一样，日期输入框也必须添加在输入框组中。
在输入框组增加一个日期输入框的语法为：
AddInput(ID, Label, type)

ID	输入框 ID。
Label	在输入框左侧显示的标签内容
Type	输入框类型，普通日期设置为"date"， 只有时间设置为"time"，同时有日期时间设置为"datetime-local"

示例
将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1","日期输入")
            .AddInput("rq","日期","date")
            .AddInput("sh","时间","time")
            .AddInput("rs","日期时间","datetime-local")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

下图是通过 iPhone 访问的截屏：


●●●●● 中国联通
07:55
192.168.0.103/test.htm
时间输入
日期
2016年10月9日
时间
07:43
日期时间
2016年10月9日 07:43
确定
清除
完成
2013年 7月 6日
2014年 8月 7日
2015年 9月 8日
2016年 10月 9日
2017年 11月 10日
2018年 12月 11日
2019年 1月 12日

属性

日期输入框有以下属性：

Value	字符型，输入框的初始值。
Readonly	逻辑型，输入框是否只读。
Required	逻辑型，是否必须输入内容，目前基于 iOS 的多数浏览器暂不支持这个属性。
Post	逻辑型，提示表单数据时是否包括此输入框的值，默认为 True。
Min	字符型，设置允许输入的最小值，目前基于 iOS 的多数浏览器暂不支持这个属性。
Max	字符型，设置允许输入的最大值，目前基于 iOS 的多数浏览器暂不支持这个属性。

例如下面的 HttpRequest 事件代码设置了日期输入框的一些属性：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1","时间输入")
            .AddInput("xm","日期","date").Value = Format(Date.Today,"yyyy-MM-dd")
            With .AddInput("xm","时间","time")
                .value = Format(Date.Now,"HH:mm")
                .Min = "08:00" '输入的事件必须在8:00到12:00之间
                .Max = "12:00"
            End With
            With .AddInput("xm","日期时间","datetime-local")
                .Value = Format(Date.Now,"yyyy-MM-ddTHH:mm") '留意一下这个格式,日期和时间之间用字母T隔开
                .Readonly = True
            End With
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

Max和Min属性在iPhone上无效，在安卓和Windows系统下有效，这是我在Windows中用Chorme模拟手机访问的效果：



7.9 数值输入框

和文本输入框一样，数值输入框也必须添加在输入框组中。
在输入框组增加一个数值输入框的语法为：
AddInput(ID, Label, type)

ID	输入框 ID。
Label	在输入框左侧显示的标签内容
Type	输入框类型，对于数值输入框，这个参数必须设置为"number"

数值输入框有以下属性：

Value	字符型，输入框的初始值。
Step	字符型，指定输入精度，数值输入框默认只能输入整数，Step 属性设置为"0.1"，可以输入一位小数，设置为"0.01"可以输入两位小数，依次类推。
Post	逻辑型，提示表单数据时是否包括此输入框的值，默认为 True。
Min	字符型，设置允许输入的最小值，目前基于 iOS 的多数浏览器暂不支持这个属性。
Max	字符型，设置允许输入的最大值，目前基于 iOS 的多数浏览器暂不支持这个属性。
Required	逻辑型，是否必须输入内容，目前基于 iOS 的多数浏览器暂不支持这个属性。
Readonly	逻辑型，输入框是否只读。
Placeholder	字符型，对输入框预期值的提示。

示例

将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1","数值输入")
            .AddInput("xm","姓名","text")
            With .AddInput("sl","年龄","number")
                .Min = "18"
                .Max = "60"
                .Placeholder = "年龄范围为18到60"
            End With
            .AddInput("dj","工资","number").Step = "0.01"
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

现在通过手机访问，可以看到下图所示的网页 ，其中年龄只能输入整数，工资可以输入两位小数：



7.10 列表输入框

和文本输入框一样，列表输入框也必须添加在输入框组中。
在输入框组增加一个列表输入框的语法为：
AddSelect(ID, Label, Values)

ID	输入框 ID。
Label	在输入框左侧显示的标签内容
Values	列表项目，用符号"|"隔开，例如"大专|本科|硕士|博士"。 打开网页后，默认会选择第一个值，如果要将其他位置的值作为默认值，可以将其用方括号括起来，例如"大专|[本科]|硕士|博士"，打开网页后，会自动选择本科。

列表输入框有以下属性：

Post　　逻辑型，提示表单数据时是否包括此输入框的值，默认为 True。

Required　　逻辑型，是否必须输入内容，目前基于 iOS 的多数浏览器暂不支持这个属性。

示例

将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1","列表项目")
            .AddSelect("os","操作系统","iOS|Windows|Andriod")
            .AddSelect("bw","浏览器","Chorme|[Edge]|Firefox|Internet Explorer")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

下图是通过 iPhone 访问时的截屏:


●●○○○ 中国联通
11:34
http://192.168.0.103/test.htm
列表项目
操作系统
iOS
浏览器
Edge
确定
语音输入
完成
Chorme
Edge
Firefox
Internet Explorer

7.11 使用 InputCell

我们之前定义的各种输入框，其实都是放在输入格中的，每个输入格被分成两部分，左边显示标签，右边显示输入框。

通过 InputCell，我们可以对格子进行更多的控制。

一个例子

我们先用一个例子说明如何使用 InputCell。

在运行这个例子之前，先下载下面的图片到"d:\web\images"目录中，名称为"vcode.jpg"：



HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1","基本资料")
            .AddInput("xm","姓名","text") '常规语法增加输入框
            With .AddInputCell("ic1") '通过 InputCell 增加输入框
                .AddLabel("lnl","年龄",0) '增加标签,0 显示在左边
                .AddInput("nl","number",1) '增加输入框,1 表示显示在中间
            End With
            With .AddInputCell("ic2",1) '通过 InputCell 增加输入框,1 表示突出显示
                .AddLabel("lkh","卡号",0) '增加标签,0 显示在左边
                .AddInput("kh","number",1).PlaceHolder= "请输入卡号" '增加输入框,1 表示显示在中间
            End With
            With .AddInputCell("ic3",2) '通过 InputCell 增加输入框, 2 表示突出显示(含图标)
                .AddLabel("lmm","密码",0) '增加标签,0 显示在左边
                .AddInput("mm","text",1).PlaceHolder= "请输入密码" '增加输入框,1 表示显示在中间
            End With
            With .AddInputCell("ic4") '通过 InputCell 增加输入框
                .AddLabel("lsj","手机",0) '增加标签,0 显示在左边
                .AddInput("sj","text",1) '增加输入框,1 表示显示在中间
                .AddVcodeButton("hym","获取验证码",2) '增加获取验证码按钮,2 表示显示在右边
            End With
            With .AddInputCell("ic5") '通过 InputCell 增加输入框
                .AddLabel("lyzm","验证码",0) '增加标签,0 显示在左边
                .AddInput("yzm","text",1) '增加输入框,1 表示显示在中间
                .AddImage("pim",".\images\vcode.jpg",2) '增加一个图片,2 比表示显示在右边
            End With
            With .AddInputCell("ic6") '通过 InputCell 增加输入框
                .AddSelect("zn","+86|+87|+88|+89",0) '增加下拉列表,0 表示显示在左边
```

```
            .AddInput("dh","text",1).PlaceHolder = "请输入联系电话" '增加输入框,1 表示显示在中间
        End With
    End With
    e.WriteString(wb.Build) '生成网页
End Select
```

下图是是通过手机访问的显示效果：



上面的例子基本涵盖了 InputCell 的全部用法。

增加 InputCell 的语法很简单：

AddInputCell(ID)

AddInputCell(ID，Warn)

ID　　InputCell 的 ID

Warn　　可选参数，设为 1 左边标签会套红显示，设为 2 右边还会显示一个红色警示图标。

AddLabel

InputCell 中通过 AddLabel 方法增加标签，语法为：

AddLabel(ID, Text, Position)

ID　　标签 ID

Text　　标签内容

Position　　标签位置，0 靠左显示，1 居中显示，2 靠右显示

AddInput

InputCell 通过 AddInput 方法增加输入框，语法：

AddInput(ID, Type, Position)

ID　　输入框 ID

Type　　输入框类型，可选值有：

text	文本输入框，参考：文本输入框
password	密码输入框，参考：文本输入框
date	日期输入框，参考：日期时间输入框
time	时间输入框，参考：日期时间输入框
datetime-local	日期时间输入框，参考：日期时间输入框
number	数值输入框，参考：数值输入框

Position　　输入框位置，0 靠左显示，1 居中显示，2 靠右显示

AddSelect

InputCell 通过 AddSelect 方法增加列表输入框，语法：

AddSelect(ID, Values, Position)

ID　　列表输入框 ID。

Values　　列表项目，用符号"|"隔开，例如"大专|本科|硕士|博士"。

Position　　标签位置，0 靠左显示，1 居中显示，2 靠右显示

AddVcodeButton

InputCell 通过 AddVcodeButton 方法增加类似"获取验证码"的按钮，语法：

AddVcodeButton(ID, Text, Position)

ID　　按钮 ID

Text　　按钮标题。

Position　　按钮位置，0 靠左显示，1 居中显示，2 靠右显示

AddImage

InputCell 通过 AddImage 方法增加图片，语法：

AddImage(ID, File, Postion)

ID　　图片 ID

File　　图片文件，包括路径。

Position　　图片位置，0 靠左显示，1 居中显示，2 靠右显示

7.12 逻辑开关

逻辑开关（Switch）类似复选框(CheckBox)，但视觉效果更好。
逻辑开关必须添加在输入框组中。
在输入框组中添加逻辑开关的语法是：
AddSwtich(ID, Label)
AddSwtich(ID, Label, Checked)

ID	开关 ID。
Label	在开关左侧显示的标签内容
Checked	逻辑型，可选参数，开关默认是否处于开启状态。

开关的属性有：

Value	字符型，开关开启后提交端到服务端的值，如果不设置，将传递"on"值给服务端。
Post	逻辑型，提示表单数据时是否包括此开发的值，默认为 True。
Enabld	逻辑型，设置为 False，将无法改变开关状态。

示例
将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1","增加客户")
            .AddInput("xm","姓名","text")
            .AddInput("nl","年龄","number")
            .AddSwitch("hy","会员",True)
            .AddSwitch("vip","VIP 客户")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

现在通过手机访问，可以看到下图所示的网页：



7.13 多行文本框

和文本输入框一样，多行文本框也必须添加在输入框组中。
在输入框组增加一个多行文本框的语法为：
AddTextArea(ID)
AddTextArea(ID，Rows)

ID	输入框 ID。
Rows	整数型，可选参数，用于设置行数，默认为 3 行。

Foxtable为多行输入框提供了以下属性：

Value	字符型，输入框的初始值。
Readonly	逻辑型，输入框是否只读。

Required	逻辑型，是否必须输入内容，目前基于 iOS 的多数浏览器暂不支持这个属性。
Post	逻辑型，提示表单数据时是否包括此输入框的值，默认为 True。
Placeholder	字符型，对输入框预期值的提示。

示例

将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1","基本资料")
            .AddInput("xm","姓名","text")
            .AddInput("nl","年龄","number")
            .AddInput("rq","日期","date")
            .AddSwitch("vip","VIP客户")
        End With
        With wb.AddInputGroup("form1","ipg2","备注")
            .AddTextArea("bz",5).Placeholder = "请输入200字以内的备注"
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

下图是在手机上的显示效果：



7.14 单选列表项

要使用单选列表项，首先得定义一个单选列表组，定义单选列表组的语法是：

AddRadioGroup(ParentID, ID)

AddRadioGroup(ParentID, ID, Text)

ParentID	父容器的 ID,如果是顶层对象，设置为""即可。
ID	分组 ID。
Text	可选参数，用于指定分组标题。

增加单选列表项的语法是：

Add(ID, Text)

Add(ID, Text, Checked)

ID	列表项 ID。
Text	单选列表项的文本内容
Checked	逻辑型，可选参数，单选列表项默认是否勾选。

单选列表项的属性有：

Value	字符型，勾选后传递给服务端的值，如果不设置，将传递 ID 值给服务端。
Enabld	逻辑型，设置为 False，将无法勾选此列表项。

示例

HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddRadioGroup("form1","rdg1","浏览器")
            .Add("bw1","Intenet Explorer")
            .Add("bw2","Google Chorme", True) '默认勾选
            .Add("bm3","FireFox")
            .Add("bm4","Safari").Enabled = False '此项不可选
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

下图是在手机上的显示效果：



7.15 复选列表项

要使用复选列表项，首先得定义一个复选列表组，定义复选列表组的语法是：

AddCheckGroup(ParentID, ID)

AddCheckGroup(ParentID, ID, Text)

ParentID　　父容器的 ID,如果是顶层对象，设置为""即可。

ID　　分组 ID。

Text　　可选参数，用于指定分组标题。

增加复选列表项的语法是：

Add(ID, Text)

Add(ID, Text, Checked)

ID　　列表项 ID。

Text　　复选列表项的文本内容。

Checked　　逻辑型，可选参数，复选列表项默认是否勾选。

复选列表项的属性有：

Value　　字符型，勾选后传递给服务端的值，如果不设置，将传递"on"值给服务端。

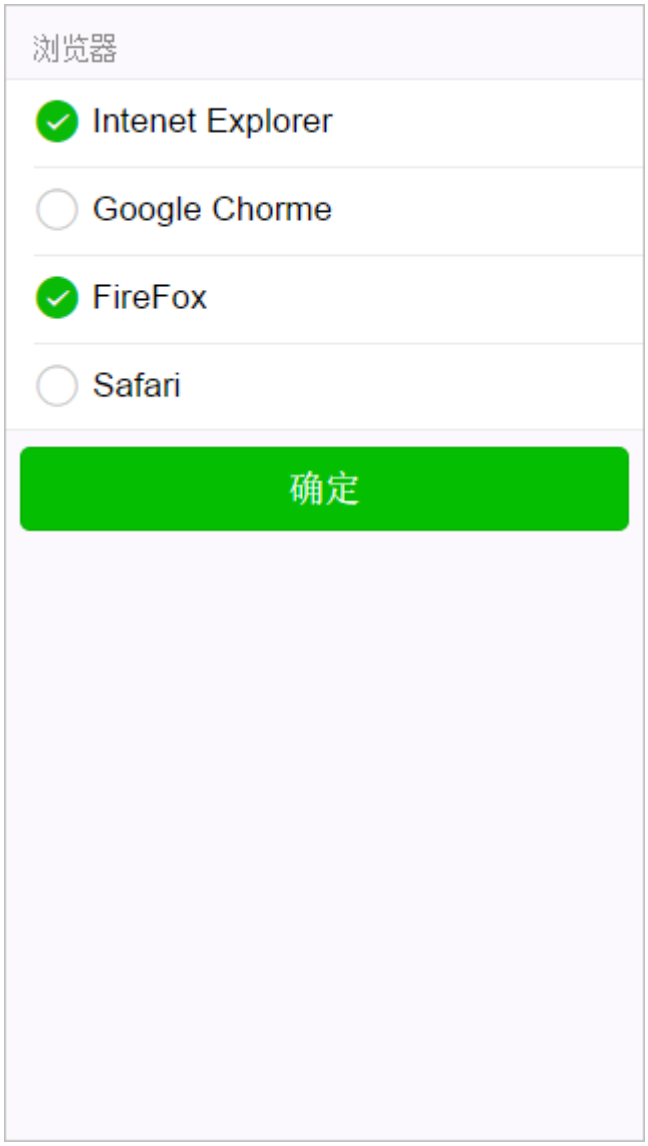
Enabld　　逻辑型，设置为 False，将无法勾选此列表项。

示例

HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddCheckGroup("form1","rdg1","浏览器")
            .Add("bw1","Intenet Explorer", True) '默认勾选
            .Add("bw2","Google Chorme")
            .Add("bm3","FireFox")
            .Add("bm4","Safari").Enabled = False '此项不可选
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

下图是在手机上的显示效果：



7.16 接收表单数据

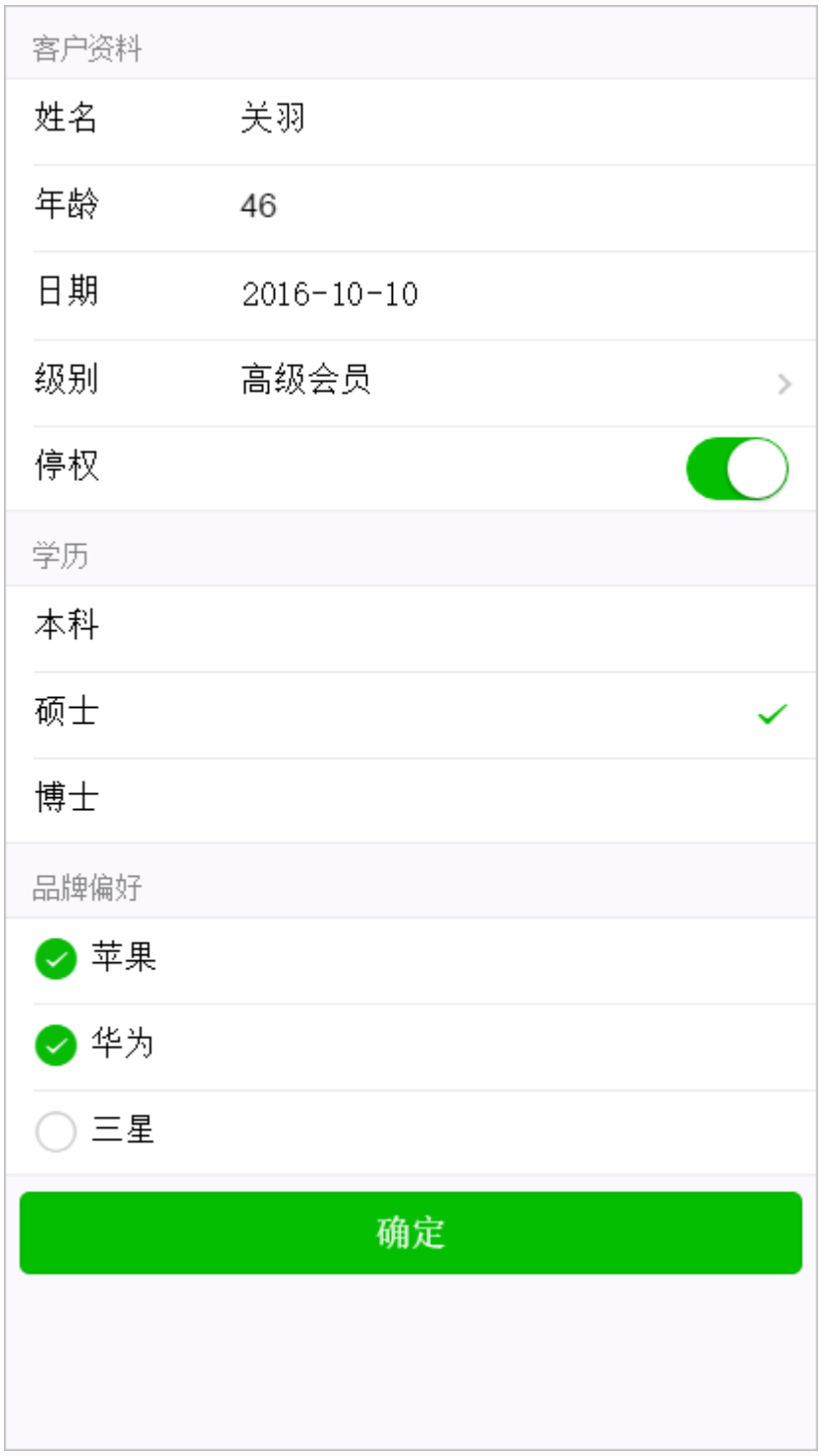
我们已经介绍完毕 WeUI 框架中基本的表单输入元素。

现在我们用一个例子来演示如何在服务端接收用户通过表单输入的数据。
以下是 HttpRequest 事件代码，为了方便，我们首次采用中文作为输入元素的 ID:

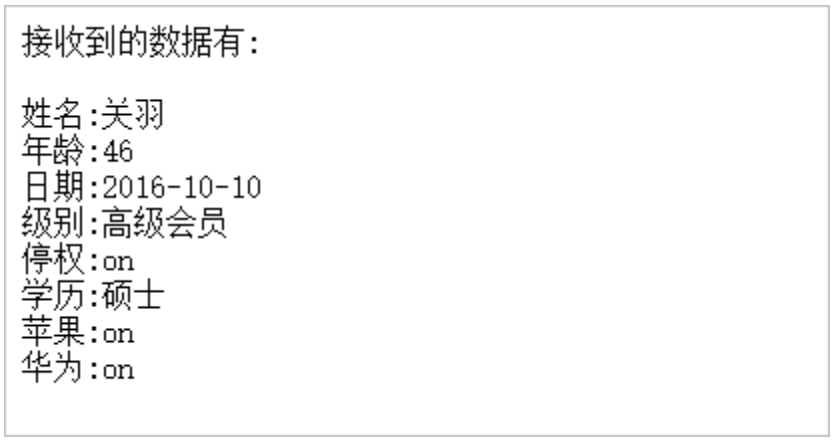
```
Select Case e.Path
    Case "test.htm"
        If e.PostValues.count = 0 Then
            Dim wb As New weui
            wb.AddForm("","form1","test.htm")
            With wb.AddInputGroup("form1","ipg1","客户资料")
                .AddInput("姓名","姓名","Text") '前一个"姓名"是ID,后一个"姓名"是标题
```

```
            .AddInput("年龄","年龄","number")
            .AddInput("日期","日期","date")
            .AddSelect("级别","级别","普通会员|高级会员|VIP会员")
            .AddSwitch("停权","停权")
        End With
        With wb.AddRadioGroup("form1","学历","学历")
            .Add("本科","本科")
            .Add("硕士","硕士")
            .Add("博士","博士")
        End With
        With wb.AddCheckGroup("form1","偏好","品牌偏好")
            .Add("苹果","苹果")
            .Add("华为","华为")
            .Add("三星","三星")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build) '生成网页
    Else
        Dim sb As New StringBuilder
                                    sb.AppendLine("<meta                                    name='viewport'
content='width=device-width,initial-scale=1,user-scalable=0'>")
        sb.AppendLine("接收到的数据有:<br/><br/>")
        For Each key As String In e.PostValues.Keys
            sb.AppendLine(key & ":" & e.PostValues(key) & "<br/>")
        Next
        e.WriteString(sb.ToString)
    End If
End Select
```

在浏览器中按下图所示输入数据：



单击确定按钮，浏览器会显示服务端接收到的数据：



框架表单和我们在快速入门中介绍的常规表单，基本上是一样的，只需留意一下：
1、单选列表项在勾选状态下，传递到服务器的值是其 ID，你可以通过 Value 属性设置你希望传递给服务器的值。

2、每个单选列表项组只能勾选一项，服务端收到的键是组的 ID 值，值为勾选项的 ID 值或其 Value 属性值。

3、复选列表项组中的每一项，只要被勾选了，都会独立向服务端传递值，键为勾选项的 ID 值，值为"on"。

3、开关组件本质上就是复选列表项，所以开关在开启后，提交给服务端的值也是"on"。

4、开关组件和复选列表项都有 Value 属性，用于设置在勾选(开启)后，提交给服务器的值。

5、没有输入内容的输入框，没有勾选项目的单选或复选列表项组，都不会向服务端传递值。

7.17 Class 和 Attribute

所谓通过 WeUI 自动生成的网页，其实就是自动生成各种 HTML 元素。
各种 HTML 元素可定义的属性是很多的，但是 Foxtable 通过 WeUI 组件提供的可直接设置的属性却不多。
不过 Foxtable 为所有 WeUI 组件都提供了 Class 和 Attribute 属性，前者用于设置 HTML 元素的 Class 属性，后者用于设置 HTML 元素除 Class 之外的所有属性。
示例
将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddButtonGroup("form1","btngrp1")
            With .Add("btn1","单击我","button")
```

```
            .Class = "btnClass"
            .Attribute = "style='font-style:italic;font-weight:900;' onclick='location=""http://www.foxtable.com""'"
        End With
    End With
    e.WriteString(wb.Build)
End Select
```

上面的代码通过Class属性给按钮添加了一个新类"btnClass"。

并通过Attribute属性完成了两项设置：

1、按钮的字体为斜体加粗：



2、单击按钮跳转到Foxtable主页。

如果通过浏览器查看生成的HTML源代码，可以看到生成的按钮的代码是:

<button id='btn1' name='btn1' type='button' class='weui_btn weui_btn_primary btnClass' style='font-style:italic;font-weight:900;' onclick='location="http://www.foxtable.com"'> 单 击 我 </button>

7.18 再谈按钮类型

现在我们有必要回头看看按钮的类型(Type)，我们知道添加按钮的语法是：

Add(ID, Text)
Add(ID, Text, Type)
Add(ID, Text, Type, Href)

ID	按钮 ID
Text	按钮标题
Type	可选参数，字符型，用于指定按钮类型，可用值有"submit"、"reset"、"button"
Href	可选参数，字符型，指定单击按钮之后跳转的目标网页的 URL

按钮的类型由第三个参数决定，默认就是"submit"。

一般来说提交按钮的type参数设置为"submit"或者不设置；重置按钮的type参数设置为"reset"；超链接按钮不要设置type参数，设置href参数即可；自定义按钮(单击执行代码)的type参数设置为"button"。

例如：

```
Select Case e.Path
    Case "test.htm"
        If e.PostValues.Count = 0 Then
            Dim wb As New weui
            wb.AddForm("","form1","test.htm")
            With wb.AddInputGroup("form1","ipg1","登录")
                .AddInput("姓名","户名","text")
                .AddInput("密码","密码","password")
            End With
            With wb.AddButtonGroup("form1","btngrp1",False)
                .Add("btn1", "确定", "submit") '提交
                .Add("btn2", "重置", "reset") '重置
                With .Add("btn3", "自定义", "button") '自定义按钮
                    .Attribute = "onclick=""confirm('你喜欢foxtable吗?')"""
                End With
                .Add("btn4", "主页", "", "http://www.foxtable.com") '超链接
            End With
            e.WriteString(wb.Build)
        Else
            Dim sb As New StringBuilder
                                        sb.AppendLine("<meta                                    name='viewport'
content='width=device-width,initial-scale=1,user-scalable=0'>")
            sb.AppendLine("接收到的数据有:<br/><br/>")
            For Each key As String In e.PostValues.Keys
                sb.AppendLine(key & ":" & e.PostValues(key) & "<br/>")
            Next
            e.WriteString(sb.ToString)
        End If
End Select
```

这是在手机上的访问效果，您可以分别单击这些按钮，看看有什么不同：



7.19 使用 MsgPage

结果页(MsgPage)通常在一系列操作后显示，用于告知用户操作结果以及一些必要的细节。

添加 MsgPage 的语法是：

AddMsgPage(ParentID, ID, Title, Content)

ParentID	父容器的 ID,如果是顶层对象，设置为""即可。
ID	MsgPage 的 ID。
Text	内容标题

Content　　内容详情

MsgPage 的方法有：

AddButton

用于添加按钮，语法：

AddButton(ID, Text)

AddButton(ID, Text, Href)

ID　　按钮 ID

Text　　按钮标题

Href　　可选参数，字符型，指定单击按钮之后跳转的目标网页的 URL

AddExtra

用于添加要在页面底部显示的内容，语法：

AddExtra(text)

AddExtra(text, href)

Text　　内容

href　　可选参数，指定单击底部内容后跳转的目标网页地址。

MsgPage 的属性有：

Icon　　字符型：指定显示的图标，可选值有"success","info","warn",对应的图标分别是：



默认为"success"

示例

HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As new WeUI
        With wb.AddMsgPage("","msgpage","操作完成","内容详情,可根据实际需要安排")
            .AddButton("btn1","确定")
            .AddButton("btn2","取消").kind = 1
            .AddExtra("详细信息","http://www.foxtable.com/")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

下图是通过iPhone访问该页面的显示效果：



7.20 一个录入界面

假定有下图所示的一个表：

	姓名	学历	偏好	年龄	级别	日期	停权
1	杨刚	硕士	华为	123	普通会员	2002-12-03	☐
2	周明	博士	苹果,三星	123	普通会员	2002-12-03	☐

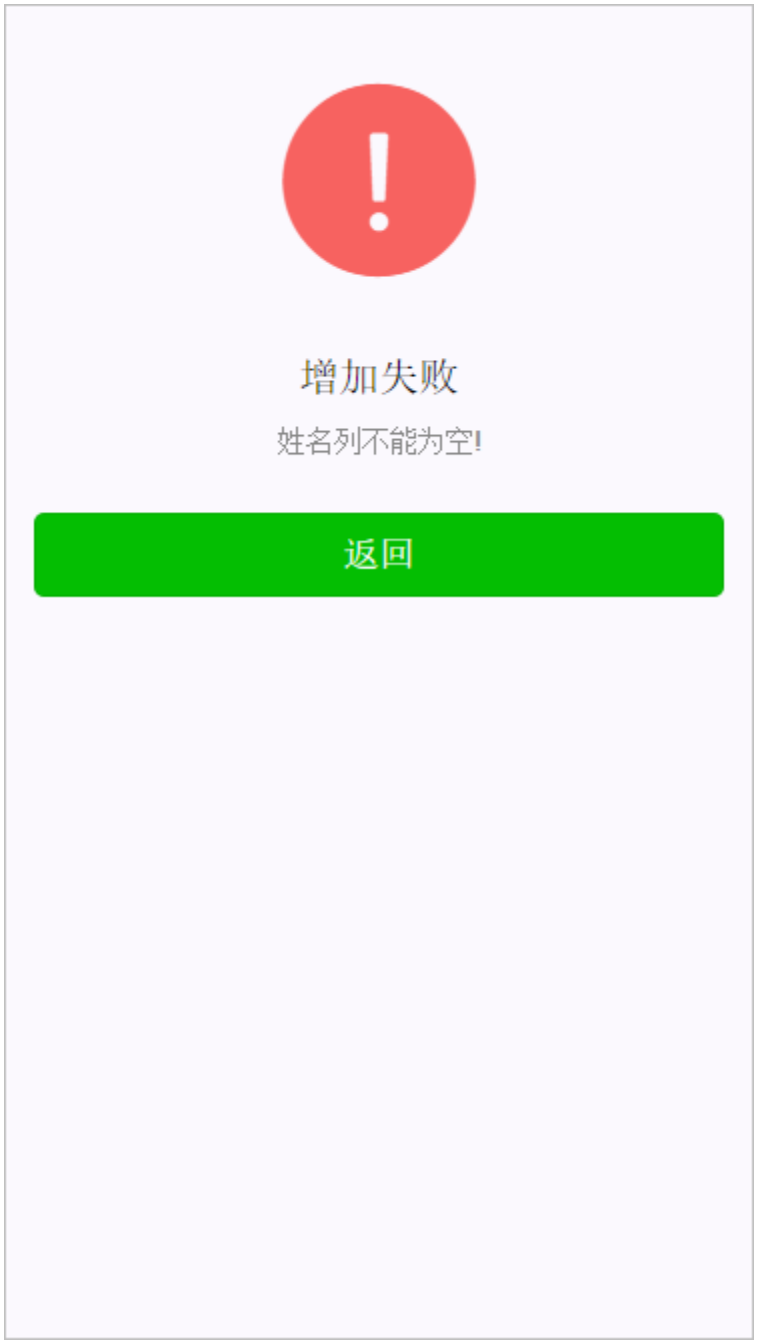
希望设计一个手机录入界面：



在输入数据，单击确定按钮，能提示下图所示的页面：



如果有些关键列内容没有输入，则提示下图所示页面：



下面是 HttpRequest 事件代码，这段代码完成了 HTTP 服务的建立，3 个页面的生成，以及数据接收等任务：

```
Dim wb As New weui
Select Case e.Path
    Case "addnew.htm"
        If e.PostValues.Count = 0 Then
            wb.AddForm("","form1","addnew.htm")
            With wb.AddInputGroup("form1","ipg1","客户资料")
                .AddInput("姓名","姓名","Text") '前一个"姓名"是 ID,后一个"姓名"是标题
                .AddInput("年龄","年龄","number")
```

```
            .AddInput("日期","日期","date")
            .AddSelect("级别","级别","普通会员|高级会员|VIP 会员")
            .AddSwitch("停权","停权").Value = "True"
        End With
        With wb.AddRadioGroup("form1","学历","最高学历")
            .Add("本科","本科")
            .Add("硕士","硕士")
            .Add("博士","博士")
        End With
        With wb.AddCheckGroup("form1","偏好","品牌偏好")
            .Add("苹果","苹果")
            .Add("华为","华为")
            .Add("三星","三星")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build)
    Else
        Dim nms() As String = {"姓名","年龄","日期","级别"} '不能为空的列名数组
        For Each nm As String In nms
            If e.PostValues.ContainsKey(nm) = False Then '生成错误提示页
                With wb.AddMsgPage("","msgpage","增加失败", nm & "列不能为空!")
                    .icon = "Warn" '改变图标
                    .AddButton("btn1","返回").Attribute = "onclick='history.back()'"
                End With
                e.WriteString(wb.Build)
                Return '必须返回
            End If
        Next
        nms = New String() {"姓名","年龄","日期","级别","停权","学历"}  '重新定义了 nms 数组,增加了两列.
        Dim dr As DataRow = DataTables("员工").AddNew()
        For Each nm As String In nms
            If e.PostValues.ContainsKey(nm) Then
                dr(nm) = e.PostValues(nm)
            End If
        Next
        '以下代码处理品牌复选列表项
        Dim pp As String
        nms = New String() {"苹果","华为","三星"} '将 nms 重新定义为品牌数组
        For Each nm As String In nms
            If e.PostValues.ContainsKey(nm) AndAlso e.PostValues(nm).Trim() = "on" Then '不能省略 Trim
                pp = pp & nm & ","
```

```
            End If
        Next
        If pp > "" Then
            dr("偏好") = pp.Trim(",")
        End If
        '保存并生成增加成功提示页面
        dr.save()
        With  wb.AddMsgPage("","msgpage","增加成功", "好好学习,天天向上") '生成成功提示
页
            .AddButton("btn1","继续增加","addnew.htm")
        End With
        e.WriteString(wb.Build)
    End If
End Select
```

代码很好理解，以下几点注意一下：

1、输入组件的 ID 尽量用列名，方便简化编码。
2、对于开关组件，默认返回值是"on"，可以将其 Value 属性设置为"True"，方便统一编码。
3、开关组件和复选列表项的返回值可能有空格，例如"on "，所以比较值的时候要用 Trim 方法去掉空格。

7.21 引用外部文件

Foxtable 为 WeUI 提供了一个 AppendHTML 方法，用于添加原生的 HTML 代码。
语法：
AppendHTML(HTML，Head)

HTML	字符型，HTML 代码。
Head	字符型，设置为 True，代码添加在网页的 Head 区，否则添加在网页的 Body 区，默认为 False。

很少用这个方法生成可视的 HTML 元素，通常用来添加第三方库文件，如 CSS 或 JavaScript 文件，或动态合成 JavaScript 代码。

引用外部文件

假定有下图所示的一个手机录入界面，希望录入单价、折扣和数量后，能自动计算出金额 ，而且当金额超过 30000 时，字体颜色自动变为红色：



设计步骤

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"mark.css"，文件内容为：

```
input.mark{color:red;}
```

再另外建立一个文本文件，文件名为"calc.js"，文件内容为：

```
function markCalc(){
    je.value=dj.value * sl.value * (1 - zk.value);
    if(je.value >= 30000){
        if(!je.classList.contains('mark')){
            je.classList.add('mark');
        }
    }
    else{
        if(je.classList.contains('mark')){
```

```
            je.classList.remove('mark');
        }
    }
}
```

提示：

dj.value表示id为"dj"的输入框框的值，需要注意的时候，WeUI在生成网页时，会自动将所有id转换为小写，所以即使你在HttpRequest事件中设置的id是大写的"DJ"，在js代码中依然应用用小写的"dj"。

2、将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "addnew.htm"
        Dim wb As New weui
        wb.AppendHTML("<link rel='stylesheet' href='./lib/mark.css'/>",True) '引入样式文件,参数True表示添加到head区
        wb.AddForm("","form1","addnew.htm")
        With wb.AddInputGroup("form1","ipg1","新增订单")
            .AddInput("cp","产品","text")
            .AddInput("gy","雇员","text")
            .AddInput("kh","客户","text")
            .AddInput("dj","单价","number").Attribute = "step='0.1' onchange='markCalc()'" '事件调用
            .AddInput("zk","折扣","number").Attribute = "step='0.01' onchange='markCalc()'"
            .AddInput("sl","数量","number").Attribute = "onchange='markCalc()'"
            .AddInput("je","金额","number")
            .AddInput("rq","日期","date")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btnok","确定")
        End With
        wb.AppendHTML("<script src='./lib/calc.js'></script>") '引入脚本文件
        e.WriteString(wb.Build)
End Select
```

再谈Attribute属性

这里特意通过Attribute来设置step属性。

Attribute可以用一行代码集中设置各种属性，例如下面的代码设置折扣的输入精度为0.01，最小值为0，最大值为0.15，值发生变化后调用calc函数。

```
.AddInput("zk","折扣","number").Attribute = "step='0.01' min='0' max='0.15' onchange='calc()'"
```

遗憾的是，精度、最大值和最小值属性在iOS平台的设备上是无效的。

7.22 InsertHTML

之前介绍的 AppendHTML，用于在 Head 区或 Body 区的结束位置追加代 HTML 代码。

如果需要在当前位置插入 HTML 代码，请使用 InsertHTML，语法：
InsertHTML(Content)
InsertHTML(ParentId,Content)
Content: 要插入的内容。
ParentId: 父容器 ID，如果是顶层对象，可以不设置。
一个例子
HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New WeUI
        wb.InsertHTML("<h3 align='center' style='margin-top:5px'>用户登录</h3>")
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1")
            .Attribute = "style='margin-top:5px'"
            .AddInput("xm","户名","text")
            .AddInput("pw","密码","password")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build)
End Select
```

下图是通过手机访问的效果，标题“用户登录”四个字是通过 InsertHTML 方法插入的：



7.23 插入标记数据

我们在生成表单的时候，可能需要在表单插入一些标记数据。

例如你需要根据现有订单来生成一个表单，那么就需要在生成的表单中存储主键值，这样用户编辑完成，将编辑结果提交到后台时，Foxtable 才能知道用户编辑的是哪一个订单。
我们可使用 AddHiddenValue 方法，在输入框组插入隐藏的标记数据，语法为：
AddHiddenValue(ID,Value)
ID： ID 值
Value： 值
用户提交表单时，隐藏数据将随表单数据一起被提交到服务器。
一个例子
HttpRequest 事件代码：

```
Dim wb As New weui
Select Case e.Path
    Case "test.htm"
        If e.PostValues.count = 0 Then
            wb.AddForm("","form1","test.htm")
            With wb.AddInputGroup("form1","ipg1","订单编辑")
                .AddHiddenValue("订单编号","123")
                .AddInput("客户","客户","text")
                .AddInput("产品","产品","text")
                .AddInput("数量","数量","number")
                .AddInput("单价","单价","number").Step= "0.01"
            End With
            With wb.AddButtonGroup("form1","btg1",True)
                .Add("btn1", "确定", "submit")
            End With
        Else
            wb.InsertHTML("接收到的数据有:<br/>")
            For Each key As String In e.PostValues.Keys
                wb.InsertHTML(key & ":" & e.PostValues(key) & "<br/>")
            Next
        End If
End Select
e.WriteString(wb.Build) '生成网页
```

用户在表单中输入数据：



单击提交按钮，可以看到，提交的数据包括隐藏的订单编号：

```
接收到的数据有:
订单编号:123
客户:苏东坡
产品:iPhone 7 plus
数量:2
单价:6288
```

7.24 动态列表项目之一

掌握了 JavaScript，可以给网页增加很多动态效果，例如最常见的动态列表项目。

一个例子

如下图所示的录入界面，希望选择不同国家后，能自动列出该国家的汽车品牌供选择：



设计步骤：

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"brands.js"，文件内容为：

```
function setBrands(){
var gj=document.getElementById("国家").value;
if(gj=="中国")setOptions("品牌","比亚迪|奇瑞|长城|荣威|吉利");
else if(gj=="德国")setOptions("品牌","奔驰|宝马|奥迪|大众|欧宝");
else if(gj=="日本")setOptions("品牌","丰田|本田|日产|讴歌|雷克萨斯");
}
```

需要特别提醒的是:

a、因为有中文，请将这个文件保存为 utf-8 格式，如果你是用 Windows 自带的记事本编辑，第一次请不要直接保存，请单击文件菜单中的另存为命令，在对话框中将编码格式选择为"UTF-8"

b、setOptions 不是原生的 JavaScript 函数，是为了方便大家，我们在 weui.me.js 中扩展的一个函数，用于设置列表项目，第一个参数为列表输入框的 ID，第二个参数为列表项目。

2、HttpRequet 事件代码：

```
Select Case e.Path
```

```
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1","动态列表")
            .AddSelect("国家","国家","中国|德国|日本").Attribute = "onchange='setBrands()'"
            .AddSelect("品牌","品牌","比亚迪|奇瑞|长城|荣威|吉利")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        wb.AppendHTML("<script src='./lib/brands.js'></script>") '引入脚本文件
        e.WriteString(wb.Build) '生成网页
End Select
```

7.25 动态列表项目之二

上一节介绍的动态列表项目，其实是不够动态的，因为列表内容是固化在 js 代码文件中的，如果列表项目要根据后台数据库内容动态生成，显然这种方式是行不通的。

假定后台有个名为"汽车"的数据表：

	国家	品牌
1	德国	奔驰
2	德国	宝马
3	德国	奥迪
4	德国	大众
5	德国	欧宝
6	中国	比亚迪
7	中国	奇瑞
8	中国	长城
9	中国	荣威
10	中国	吉利
11	日本	丰田
12	日本	本田
13	日本	日产
14	日本	讴歌
15	日本	雷克萨斯

前台的录入界面如下图，希望选择不同的国家后，能自动根据后台数据表的内容，列出该国的汽车品牌供选择：



设计步骤

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"brands.js"，文件内容为：

```
function getBrands(){
    var gj=document.getElementById("国家").value;
    setOptions("品牌",document.getElementById(gj).innerHTML);
}
```

2、HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        Dim gjs As List(of String) = DataTables("汽车").GetValues("国家")
        With wb.AddInputGroup("form1","ipg1","动态列表")
            .AddSelect("国家","国家","|" & String.Join("|",gjs.ToArray)).Attribute = "onchange='getBrands()'" '调用js函数。
```

```
        .AddSelect("品牌","品牌","")
    End With
    With wb.AddButtonGroup("form1","btg1",True)
        .Add("btn1", "确定", "submit")
    End With
    For Each gj As String In gjs
        '插入一些隐藏段落,用于存储各个国家的汽车品牌
        wb.InsertHTML("<p hidden id='" & gj & "'>" & DataTables("汽车").GetComboListString("品牌","
国家='" & gj & "'") & "</p>")
    Next
    wb.AppendHTML("<script src='./lib/brands.js'></script>") '引入脚本文件
    e.WriteString(wb.Build) '生成网页
End Select
```

设计的思路很简单，就是生成的网页中，插入一些隐藏的段落，段落 id 为国家名称，内容为该国家的汽车品牌：

<p hidden id='德国'>奥迪|宝马|奔驰|大众|欧宝</p>

<p hidden id='日本'>本田|丰田|雷克萨斯|讴歌|日产</p>

<p hidden id='中国'>比亚迪|长城|吉利|奇瑞|荣威</p>

当用户选择不同国家后，触发 onchange 事件，执行 getBrand 函数，自动找出对应的段落，从中提取出列表项目。

7.26 动态列表项目之三

前面介绍的动态列表项目方法，优势是响应快，因为运行过程不需要连接服务器。

但是如果数据量比较大，用这种方法是不合适的，所以我们还提供了在运行过程中连接服务器来动态生成列表项目的方法。

参考：setAjaxOptions

三种方法各有优势，根据需要选用即可。

7.27 区分多个提交按钮

同一个表单，可能需要多个提交按钮，服务端在接收表达数据后，根据用户单击不同的提交按钮，进行不同的后续操作。

服务端是如何区分不同的提交按钮呢？

方法一

最简单的方式是设置提交按钮的 value 属性，单击某个提交按钮，此按钮的 Value 属性值将作为表单数据的一部分提交到服务器。

方法二

另一中方法是设置提交按钮的 FormAction 属性，单击提交按钮后，数据将被提交到 FormAction 所制定的网页链接，这种方法的优势是可以通过 get 方式附加更多的数据。

一个例子

HttpRequest 事件代码：

```
Dim wb As New weui
Select Case e.Path
    Case "input.htm"
        If e.PostValues.Count = 0 Then
            wb.AddForm("","form1","input.htm")
            With wb.AddInputGroup("form1","ipg1","数据输入")
                .AddInput("客户","客户","text")
                .AddInput("日期","日期","date")
                .Addinput("产品","产品","text")
                .Addinput("数量","数量","number")
            End With
            With wb.AddButtonGroup("form1","btngrp1",False)
                .Add("btn1", "按钮1", "submit").Value = "btn1"
                .Add("btn2", "按钮2", "submit").Value = "btn2"
                .Add("btn3", "按钮3", "submit").FormAction = "accept.htm"
                .Add("btn4", "按钮4", "submit").FormAction = "accept.htm?type=1&model=ha"
            End With
        Else
            wb.InsertHTML("接收到的数据有:<br/>")
            For Each key As String In e.PostValues.Keys
                wb.InsertHTML(key & ":" & e.PostValues(key) & "<br/>")
            Next
        End If
    Case "accept.htm"
        wb.InsertHTML("通过accept.htm接收到的数据有:<br/>")
        For Each key As String In e.PostValues.Keys
            wb.InsertHTML(key & ":" & e.PostValues(key) & "<br/>")
        Next
End Select
e.WriteString(wb.Build)
```

通过浏览器访问，可以看到四个提交按钮：



定义表单的代码为：

```
wb.AddForm("","form1","input.htm")
```

表示默认接收表单的目标网页为input.htm，单击按钮1和按钮2会将数据提交到这个页面。

由于我们给按钮1和按钮2设置了Value属性，服务端通过接收到的值，可以判断出用于单击的是那个按钮。

按钮3和按钮4通过FormAction属性，将数据接收页面改为accept.htm，不同的是按钮4还会通过get方式 附加了两个数据，这两个数据会随表单数据一并提交到服务器。。

下表分别单击四个按钮后的显示结果：

按钮 1	接收到的数据有: 客户:CS01 日期:2016-11-30 产品:PD01 数量:100 btn1:btn1
按钮 2	接收到的数据有: 客户:CS01 日期:2016-11-30 产品:PD01 数量:100 btn2:btn2
按钮 3	通过 accept.htm 接收到的数据有: 客户:CS01 日期:2016-11-30 产品:PD01 数量:100
按钮 4	通过 accept.htm 接收到的数据有: type:1 model:ha 客户:CS01 日期:2016-11-30 产品:PD01 数量:100 提示：type 和 model 两个值是通过 get 方式传递的。

7.28 使用 Uploader

7.28.1 使用 Uploader

Uploader 用于上传文件，和其他数据输入组件一样，Uploader 需要放在输入框分组中。
定义一个 Uploader 的语法是：
AddUploader(ID, Label, Multiple)

ID	组件 ID
Text	组件标题
Multiple	逻辑型，可选参数，手否允许选择多个文件上传。

上传组件可以浏览图片，通过 AddImage 添加要浏览的图片，语法：

AddImage(Image)

AddImage(Thumbnail, Image)

Thumbnail　缩略图文件。

Image　图片文件。

示例

在运行这个例子之前，请先复制几个图片到"d:\web\images 目录"，文件名分别是"001.jpg"、"002.jpg"和"003.jpg"。

HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1","文件上传") '文件上传
            .AddUploader("up1","")
        End With
        With wb.AddInputGroup("form1","ipg22","文件上传") '带图片浏览的文件上传
            With .AddUploader("up2","图片",True) 'True表示允许一次上传多个文件
                .AddImage("./images/001.jpg")
                .AddImage("./images/002.jpg")
                .AddImage("./images/003.jpg")
            End With
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

下图是在 iPhone 上的显示效果：



单击上图中的“+”按钮，可以拍照上传，或选择现有的照片上传：



7.28.2 改变标题位置

Uploader 组件有个 TextPosition 属性，用于设置标题位置，0 表示标题靠左，1 表示标题靠上，默认为 1。

HttpRequet 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
```

```
    With wb.AddInputGroup("form1","ipg1","客户资料")
      .AddInput("姓名","姓名","Text").value = "舒淇"
      .AddInput("年龄","年龄","number").Value = "28"
      .AddInput("日期","日期","date").value = #10/12/2012#
      With .AddUploader("up1","照片",True)
        .TextPosition = 0 '标题靠左
        .AddImage("./images/shuqi1.jpg")
        .AddImage("./images/shuqi2.jpg")
      End With
    End With
    With wb.AddButtonGroup("form1","btg1",True)
      .Add("btn1", "确定", "submit")
    End With
    e.WriteString(wb.Build) '生成网页
End Select
```

下图是通过手机访问的效果：



7.28.3 在微信浏览器中使用相机

如果你的手机是安卓系统，并且使用的是微信内置的浏览器浏览 WeUI 生成的网页，你会发现 Uploader 只能从相册中选择图片上传，无法使用相机拍照上传。

要解决这个问题很简单， 只需将 Uploader 的 Accept 属性设置为"image/*"，例如：

```
Select Case e.Path

    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg22","文件上传")
            With .AddUploader("up2","图片",True)
                .Accept = "image/*"    '允许使用相册和相机
            End With
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build)
End Select
```

如果希望不能从相册选择，而是直接拍照上传，可以再将 Capture 属性设置为"camera"，例如：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg22","文件上传")
            With .AddUploader("up2","图片",True)
                .Accept = "image/*"
                .Capture = "camera" '只能拍照上传
            End With
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build)
End Select
```

7.28.4 接收上传的文件

假定有下图所示的一个表：

	姓名	学历	年龄	照片
1	陈夏燕	大专	26	EP9.BMP
2	郑卫国	大专	46	EP8.BMP

希望设计一个下图所示的录入界面，能拍照并上传照片：



HttpRequest 事件代码：

```
Dim wb As New weui
Select Case e.Path
    Case "addnew.htm"
        If e.PostValues.Count = 0 Then
            wb.AddForm("","form1","addnew.htm")
            With wb.AddInputGroup("form1","ipg1","增加员工")
                .AddInput("姓名","姓名","Text") '前一个"姓名"是ID,后一个"姓名"是标题
                .AddInput("年龄","年龄","number")
                .AddSelect("学历","学历","大专|本科|硕士|博士")
                .AddUploader("up1","照片",True) 'True表示允许上传多个文件
```

```
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build)
    Else
        Dim nms() As String = {"姓名","年龄","学历"}
        Dim dr As DataRow = DataTables("员工").AddNew()
        For Each nm As String In nms
            dr(nm) = e.PostValues(nm)
        Next
        For Each key As String In e.Files.Keys
            If key = "up1" Then
                For Each fln As String In e.Files(key)
                    e.SaveFile(key, fln, ProjectPath & "Attachments\" & fln)
                Next
                dr.Lines("照片") = e.Files(key)
            End If
        Next
        '保存并生成增加成功提示页面
        dr.save()
        With wb.AddMsgPage("","msgpage","增加成功", "好好学习,天天向上") '生成成功提示页
            .AddButton("btn1","继续增加","addnew.htm")
        End With
        e.WriteString(wb.Build)
    End If
End Select
```

关于 Foxtable 是如何接收用户上传的文件，我们在快速入门这一章已经讲述，参考：文件的上传与接收

关于如何处理包括多行内容的单元格，参考：单元格多行内容的处理

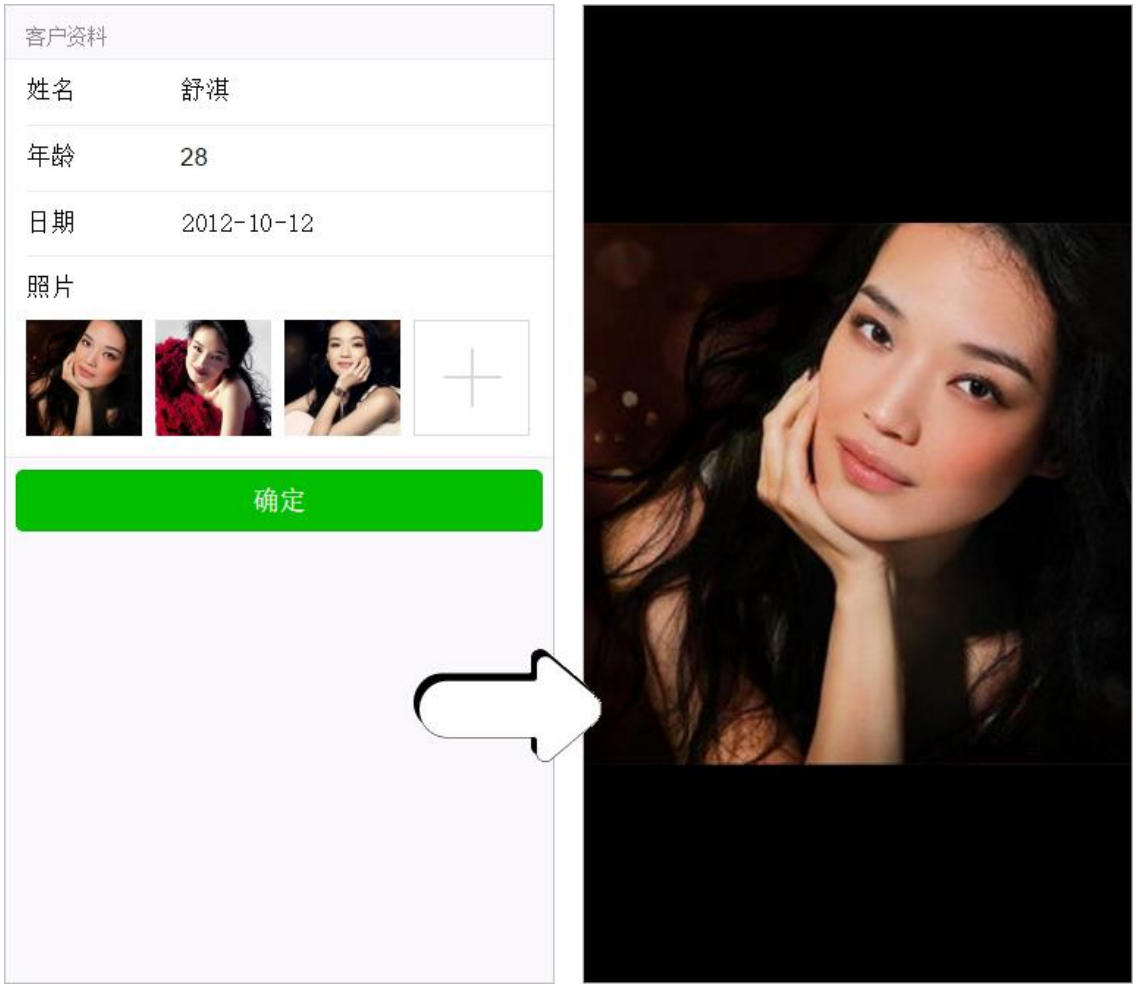
7.28.5 图片浏览窗口

在 Uploader 中，单击图片，会出现一个图片浏览窗口，在这个窗口中，你可以：

1、点击窗口左侧显示上一副图片。

2、点击窗口右侧显示下一幅图片。

3、点击窗口中央位置关闭图片浏览窗口。



7.28.6 我只想浏览图片

如果将 Uploader 的 AllowAdd 属性设置为 False，Uploader 的文件上传功能将被关闭，成为一个单纯的图片浏览器。

HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1","客户资料")
            .AddInput("姓名","姓名","text").value = "舒淇"
            .AddInput("地点","地点","text").Value = "蒙古草原"
            .AddInput("日期","日期","date").value = #10/12/2012#
            With .AddUploader("up1","",True)
                .AllowAdd = False '关闭文件上传功能
                .AddImage("./images/001.jpg")
                .AddImage("./images/002.jpg")
```

```
            .AddImage("./images/003.jpg")
            .AddImage("./images/004.jpg")
            .AddImage("./images/005.jpg")
            .AddImage("./images/006.jpg")
            .AddImage("./images/007.jpg")
            .AddImage("./images/008.jpg")
            .AddImage("./images/009.jpg")
            .AddImage("./images/010.jpg")
            .AddImage("./images/011.jpg")
            .AddImage("./images/012.jpg")
        End With
    End With
    e.WriteString(wb.Build) '生成网页
End Select
```

下图是通过手机访问的效果：



7.28.7 使用缩略图

为了提高网页打开速度，我们可以在 Uploader 中显示低分辨率的缩略图，在图片浏览器中显示高分辨率的原图。

一个例子

HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
```

```
    wb.AddForm("","form1","test.htm")
    With wb.AddInputGroup("form1","ipg1","客户资料")
      .AddInput("姓名","姓名","text").value = "舒淇"
      .AddInput("地点","地点","text").Value = "蒙古草原"
      .AddInput("日期","日期","date").value = #10/12/2012#
      With .AddUploader("up1","",True)
        .AllowAdd = False '关闭文件上传功能
        'AddImage的第一个参数为缩略图，第二个参数为原图
        .AddImage("./images/001s.jpg","./images/001.jpg")
        .AddImage("./images/002s.jpg","./images/002.jpg")
        .AddImage("./images/003s.jpg","./images/003.jpg")
        .AddImage("./images/004s.jpg","./images/004.jpg")
        .AddImage("./images/005s.jpg","./images/005.jpg")
        .AddImage("./images/006s.jpg","./images/006.jpg")
        .AddImage("./images/007s.jpg","./images/007.jpg")
        .AddImage("./images/008s.jpg","./images/008.jpg")
        .AddImage("./images/009s.jpg","./images/009.jpg")
        .AddImage("./images/010s.jpg","./images/010.jpg")
        .AddImage("./images/011s.jpg","./images/011.jpg")
        .AddImage("./images/012s.jpg","./images/012.jpg")
      End With
    End With
    e.WriteString(wb.Build) '生成网页
End Select
```

这是通过手机访问的显示效果：



7.28.8 开启图片删除功能

如果将 Uploader 的 AllowDelete 属性设置为 True,在图片浏览窗口的底部会出现一个删除按钮，单击这个按钮，可以删除正在浏览的图片。

HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1","客户资料")
            .AddInput("姓名","姓名","Text").value = "舒淇"
            .AddInput("年龄","年龄","number").Value = "28"
            .AddInput("日期","日期","date").value = #10/12/2012#
            With .AddUploader("up1","照片",True)
                .AllowDelete = True '允许用户删除图片
                .AddImage("./images/shuqi1.jpg")
                .AddImage("./images/shuqi2.jpg")
                .AddImage("./images/shuqi3.jpg")
```

```
        End With
    End With
    With wb.AddButtonGroup("form1","btg1",True)
        .Add("btn1", "确定", "submit")
    End With
    e.WriteString(wb.Build) '生成网页
End Select
```

这是通过手机访问的效果，当所有图片都被删除完毕时，图片浏览窗口会被自动关闭：





7.28.9 删除后台文件

在默认情况下，用户删除 Uploader 中显示的图片，并不会影响服务器中对应的的图片文件。
如果要同步删除服务器中的图片文件，我们还需要做响应的处理。

原理：

将 Uploader 的 AllowEdit 属性设置为 True 之后，系统会自动生成一个隐藏的输入框，这个输入框的 ID 是 Uploader 的 ID 加上“_deleted”。
该隐藏输入框会自动记录用户删除的图片文件，如果删除了多个图片文件，会用符号"|"分割文件名。

用户向服务端提交数据时，这个隐藏输入框的值会一并提交，服务端可以从这个值中提取出用户已经删除的文件，然后删除后台对应的文件。

一个例子：

HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        If e.PostValues.Count = 0 Then
            Dim wb As New weui
            wb.AddForm("","form1","test.htm")
            With wb.AddInputGroup("form1","ipg1","客户资料")
                .AddInput("姓名","姓名","Text").value = "舒淇"
                .AddInput("年龄","年龄","number").Value = "28"
                .AddInput("日期","日期","date").value = #10/12/2012#
                With .AddUploader("up1","照片",True)
                    .AllowDelete = True '允许用户删除图片
                    .AddImage("./images/shuqi1.jpg")
                    .AddImage("./images/shuqi2.jpg")
                    .AddImage("./images/shuqi3.jpg")
                End With
            End With
            With wb.AddButtonGroup("form1","btg1",True)
                .Add("btn1", "确定", "submit")
            End With
            e.WriteString(wb.Build) '生成网页
        Else
            Dim sb As New StringBuilder
            sb.AppendLine("<meta name='viewport' content='width=device-width,initial-scale=1,user-scalable=0'>")
            sb.AppendLine("已经删除了以下文件:<br/>")
            If e.PostValues.ContainsKey("up1_deleted") Then
                Dim Files = e.PostValues("up1_deleted").Split("|")
                For Each File As String In Files
                    sb.AppendLine(file & "<br/>")
                    file = "d:\web\" & File.Trim(".")
                    If Filesys.FileExists(file) Then
                        Filesys.DeleteFile(file)
                    End If
                Next
            End If
            e.WriteString(sb.ToString)
        End If
End Select
```

7.29 使用 PageTitle

PageTile 用于给页面加上标题。
添加 PageTitle 的语法是：
AddPageTitle(ParentID, ID, Title, SubTitle)

ParentID	父容器的 ID,如果是顶层对象，设置为""即可。
ID	PageTitle 的 ID。
Title	标题。
SubTitle	副标题。

一个例子
HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New WeUI
        wb.AddPageTitle("","ph1","FoxUI","为Foxtable用户量身设计")
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1")
            .AddInput("xm","户名","text")
            .AddInput("pw","密码","password")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build)
End Select
```

下图是通过手机浏览显示的效果：



7.30 使用 PageFooter

PageFooter 用于在页面底部显示文字信息和链接，例如版权信息。

一个例子

HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New WeUI
        wb.AddPageTitle("","ph1","FoxUI","为Foxtable用户量身设计")
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1")
            .AddInput("xm","户名","text")
            .AddInput("pw","密码","password")
```

```
        End With
        With wb.AddButtonGroup("form1","btg1",True)
          .Add("btn1", "确定", "submit")
        End With
        With wb.AddPageFooter("","pf1","Copyright &copy; 2008-2016 foxtable.com")
          .AddLink("底部链接","http://www.foxtable.com")
        End With
        e.WriteString(wb.Build)
End Select
```

下图是通过手机访问的效果，可以看到页面底部有版权说明和链接：



AddPageFooter

AddPageFooter 用于增加 PageFooter，语法：

AddPageFooter(ParentId, ID, Text)

ParentID　　父容器的 ID,如果是顶层对象，设置为""即可。

ID	PageFooter 的 ID。
Text	PageFooter 的文本内容。

AddLink

AddLink 用于给 PageFooter 增加链接，语法：
AddLink(Text, Href, Attribute)

Text	链接的文本内容。
Href	目标 URL。
Attribute	可选参数，用于设置链接属性。

7.31 使用 List

List(列表)用于列表内容，列表项可以附带说明、超链接和图标。
一个例子
HttpRequest 事件代码

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New WeUI
        With wb.AddListGroup("", "lsg1","简单列表")
            .Add("ls1", "新浪主页")
            .Add("ls2", "网易主页")
        End With
        With wb.AddListGroup("", "lsg2", "增加说明的列表")
            .Add("ls3","新浪主页", "sina.com")
            .Add("ls4","网易主页", "163.com")
        End With
        With wb.AddListGroup("", "lsg3", "增加跳转的列表")
            .Add("ls5","新浪主页", "sina.com", "http://www.sina.com.cn")
            .Add("ls6","网易主页", "163.com", "http://www.163.com")
        End With
        With wb.AddListGroup("", "lsg4","增加图标的列表")
            .Add("ls7","新浪主页", "sina.com", "http://www.sina.com.cn", "./images/sina.png")
            .Add("ls8","网易主页", "163.com", "http://www.163.com", "./images/163.png")
        End With
        e.WriteString(wb.Build)
End Select
```

图标大小建议为 16*16 像素，下图是通过手机访问的效果：



AddListGroup

Add 用于增加列表项组，其语法为：

AddListGroup(ParentID, ID, Text)

ParentID	父容器的 ID,如果是顶层对象，设置为""即可。
ID	分组 ID。
Text	可选参数，用于指定分组标题。

Add

Add 用于增加列表项，其语法为：

Add(ID, Text)

Add(ID, Text, Description)

Add(ID, Text, Description, Href)

Add(ID, Text, Description, Href, Image)

ID	ID。
Text	列表内容。
Description	列表说明
Href	超链接。
Image	图标。

7.32 给 List 加上徽章

List 的列表项有个字符型属性 Badge，可以用来给列表项设置徽章。

一个例子

HtppRequest 事件代码

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New WeUI
        With wb.AddListGroup("", "lsg1","简单列表")
            .Add("ls1", "新浪主页").Badge = "8"
            .Add("ls2", "网易主页").Badge= " "
        End With
        With wb.AddListGroup("", "lsg2", "增加说明的列表")
            .Add("ls3","新浪主页", "sina.com").Badge = "New"
            .Add("ls4","网易主页", "163.com")
        End With
        With wb.AddListGroup("", "lsg3", "增加跳转的列表")
            .Add("ls5","新浪主页", "sina.com", "http://www.sina.com.cn").Badge = " "
            .Add("ls6","网易主页", "163.com", "http://www.163.com")
        End With
        With wb.AddListGroup("", "lsg4","增加图标的列表")
            .Add("ls7","新浪主页", "sina.com", "http://www.sina.com.cn", "./images/sina.png")
             .Add("ls8"," 网 易 主 页 ",  "163.com",  "http://www.163.com",  "./images/163.png").Badge="
新"
        End With
        e.WriteString(wb.Build)
End Select
```

这是通过手机访问的显示效果：



提示：如果Badge被设置为空格，徽章会显示为一个红色小圆。

7.33 使用 Grid

Grid(网格)通常用于设计首页的功能导航。

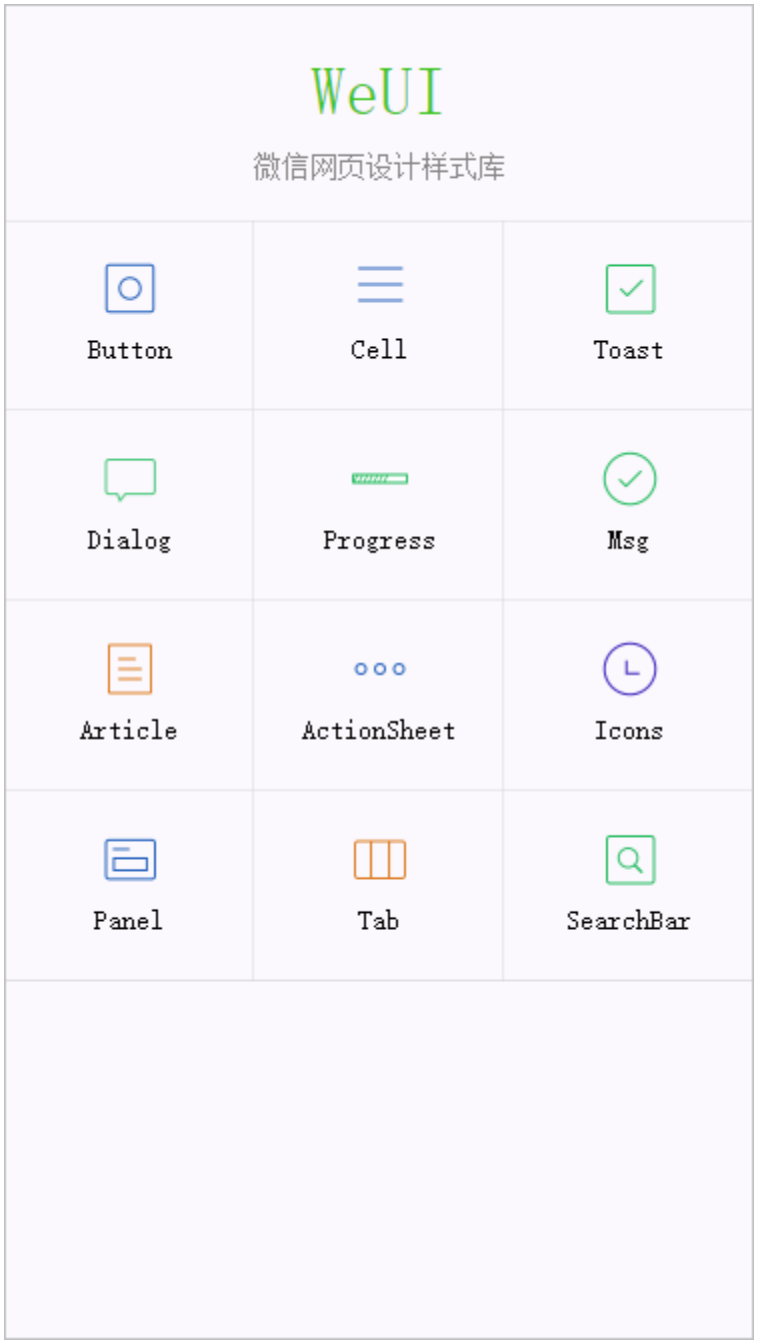
一个例子

HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New WeUI
        wb.AddPageTitle("","pageheader","WeUI","微信网页设计样式库")
        With wb.AddGrid("","g1")
            .Add("c1","Button",  "./images/button.png").Attribute  =  "onclick='javascript:alert("" 你 单
```

```
击了我!"")'"
        .Add("c2","Cell", "./images/cell.png", "http://www.foxtable.com")
        .Add("c3","Toast", "./images/toast.png", "http://www.foxtable.com")
        .Add("c4","Dialog", "./images/dialog.png", "http://www.foxtable.com")
        .Add("c5","Progress", "./images/progress.png", "http://www.foxtable.com")
        .Add("c6","Msg", "./images/msg.png", "http://www.foxtable.com")
        .Add("c7","Article", "./images/article.png", "http://www.foxtable.com")
        .Add("c8","ActionSheet", "./images/actionSheet.png", "http://www.foxtable.com")
        .Add("c9","Icons", "./images/icons.png", "http://www.foxtable.com")
        .Add("c10","Panel", "./images/panel.png", "http://www.foxtable.com")
        .Add("c11","Tab", "./images/tab.png", "http://www.foxtable.com")
        .Add("c12","SearchBar", "./images/search.png", "http://www.foxtable.com")
    End With
    e.WriteString(wb.Build)
End Select
```

这是通过手机访问的显示效果：



AddGrid

AddGrid方法用于增加网格，语法：

AddGrid(ParentID,ID)

ParentID　　父容器的 ID,如果是顶层对象，设置为""即可。

ID　　分组 ID。

Add

Add 方法用于在网格中添加单元格，语法：

Add(ID, Text, Image)
Add(ID, Text, Image, Href)

ID	单元格 ID。
Text	单元格文本
Image	单元格图片
Href	可选参数，单元格图片。

7.34 使用 Panel

Panel 主要用于图文组合列表显示。
一个例子

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New WeUI
        Dim txt As String = "由各种物质组成的巨型球状天体,叫做星球.星球有一定的形状,有自己的运行轨道."
        With wb.AddPanelGroup("","pg1","图文组合列表")
            .Add("pn1","标题一",txt,"./images/button.png","http://www.foxtable.com") '超链接
            With .Add("pn2","标题二",txt,"./images/search.png") '带子链接
                .AddFoot("文字来源")
                .AddFoot("时间")
                .AddFoot("|其他信息","http://www.foxtable.com")
            End With
        End With
        With wb.AddPanelGroup("","pg2","文字组合列表")
            .Add("pn1","标题一",txt) '普通列表
            .Add("pn1","标题  二",txt,"","http://www.foxtable.com") '超链接
            With .Add("pn2","标题三",txt) '带有子链接
                .AddFoot("文字来源")
                .AddFoot("时间")
                .AddFoot("|其他信息","http://www.foxtable.com")
            End With
            .GroupFoot = "查看更多" '底部链接
            .GroupHref = "http://www.foxtable.com/"
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

这是在手机上的显示效果：



AddPanelGroup

AddPanelGroup用于增加组合列表组，语法：

AddPanelGroup(ParentID,ID,Text)

ParentID　　父容器的 ID,如果是顶层对象，设置为""即可。

ID	分组 ID。
Text	可选参数，用于指定分组标题。

GroupFoot和GroupHref

GroupFoot和GroupHref属性用于在列表组的结束位置显示一个超链接，上图中的"查看更多"链接就是通过这两个属性实现的。

Add

Add方法用于增加组合列表项，语法：

Add(ID, Text, Content)
Add(ID, Text, Content, Image)
Add(ID, Text, Content, Image, Href)

ID	列表 ID。
Text	列表标题。
Content	列表内容
Image	可选参数，列表图片。
Href	可选参数，单击列表要跳转到的目标 URL。

AddFoot

AddFoot 方法用于在列表项的下方添加脚注，语法：

AddFoot(Text)
AddFoot(Text，Href)

Text	脚注内容。
Href	可选参数，单击脚注要跳转到的目标 URL。

7.35 给 Panel 加上徽章

Panel 有个字符型属性 Badge，可以用来设置徽章。

一个例子

HtppRequest 事件代码:

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New WeUI
        Dim txt As String = "由各种物质组成的巨型球状天体,叫做星球.星球有一定的形状,有自己的运行轨道."
        With wb.AddPanelGroup("","pg1","图文组合列表")
```

```
        .Add("pn1","标题一",txt,"./images/button.png","http://www.foxtable.com").Badge = "8"
        With .Add("pn2","标题二",txt,"./images/search.png")
          .Badge= " "
          .AddFoot("文字来源")
          .AddFoot("时间")
          .AddFoot("|其他信息","http://www.foxtable.com")
        End With
      End With
      With wb.AddPanelGroup("","pg2","文字组合列表")
        .Add("pn1","标题一",txt)
        .Add("pn1","标题二",txt,"","http://www.foxtable.com").Badge="新"
        With .Add("pn2","标题三",txt)
          .Badge = "New"
          .AddFoot("文字来源")
          .AddFoot("时间")
          .AddFoot("|其他信息","http://www.foxtable.com")
        End With
      End With
      e.WriteString(wb.Build) '生成网页
End Select
```

下图是通过手机访问的显示效果：



提示：如果 Badge 被设置为空格，徽章会显示为一个红色小圆。

7.36 使用 Article

Article 用于显示大段的文字，支持分段、多层标题、引用和内嵌图片。

一个例子

HttpRequest 事件代码

```
'''
Select Case e.Path
```

```
    Case "test.htm"
        Dim wb As New WeUI
        With wb.AddArticle("","ar1")
            .AddTitle("h1","大标题")
            .AddTitle("h2","章标题")
            .AddTitle("h3","1.1节标题")
            .AddContent("Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute")
            .AddImage("./images/001.jpg")
            .AddTitle("h3","1.2节标题")
            .AddContent("Foxtable将Excel、Access、Foxpro、VB以及易表的优势融合在一起，无论是数据录入、查询、统计，还是报表生成，都前所未有的强大和易用，普通用户无需编写任何代码，即可轻松完成复杂的数据管理工作，真正做到拿来即用。")
            .AddImage("./images/002.jpg")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

这是通过手机访问的显示效果：



AddArticle

AddArticle 用于增加 Article，语法为：
AddArticle(ParentID, ID)
ParentID　　父容器的 ID,如果是顶层对象，设置为""即可。
ID　　分组 ID。

AddTitle

AddTitle 用于增加标题，语法为：

AddTitle(Level, Text)

AddTitle(Level, Text, Attribute)

Level	标题层级，有 6 个可选值，分别为：h1、h2、h3、h4、h5、h6。
Text	标题内容。
Attribute	可选值，标题属性。

AddContent

AddContent 用于增加文本段落，语法为：

AddContent(Text)

AddContent(Text, Attribute)

Text	段落内容。
Attribute	可选值，段落属性。

AddImage

AddImage 用于增加图片，语法为：

AddImage(Image, Attribute)

Image	图片文件。
Attribute	可选值，图片属性。

7.37 使用 Dialog

Dialog 用于显示一个对话框。

一个例子

HttpRequest 事件代码：

```
'''
Select Case e.Path
    Case "test.htm"
        Dim wb As New WeUI
        With wb.AddButtonGroup("","bng2",True)
            .Add("btn1","Dialog1").Attribute = "onclick=""show('dlg1')"""
            .Add("btn2","Dialog2").Attribute = "onclick=""show('dlg2')"""
        End With
        With wb.AddDialog("","dlg1", "提示","您的订单正在派送,请注意查收!")
            .AddButton("btnOK","确定").Attribute = "onclick=""alert('谢谢支持!')"""
        End With
```

```
    With wb.AddDialog("","dlg2", "删除确认","您确定要删除当前记录吗?")
      .AddButton("btnCancel","取消").Kind = 1
      .AddButton("btnOK","确定","./delete.htm?id=12")
    End With
    e.WriteString(wb.Build) '生成网页
  Case "delete.htm"
    Dim sb As New StringBuilder
                                    sb.AppendLine("<meta                                    name='viewport'
content='width=device-width,initial-scale=1,user-scalable=0'>")
    sb.AppendLine("呜呜，我被删除了！ ")
    e.WriteString(sb.ToString)
End Select
```

提示：

1、上述代码中的 show，是我们在文件"weui.me.js"定义的一个函数，分别用于显示和隐藏指定 ID 的对象 ，对应地还有一个 hide 函数，用于隐藏指定 ID 的对象。
2、注意所有 js 函数都是区分大小写的，例如你用 Show 或者 Hide，是不会执行的，今后不会再重复提示，请大家注意了。

通过手机访问，会显示两个按钮，分别单击两个按钮，会显示两个对话框：


Dialog1
Dialog2
提示
您的订单正在派送, 请注意查收!
确定



AddDialog

AddDialog 用于添加对话框,语法：

AddDialog(ParentId, ID, Title, Content)

ParentID	父容器的 ID,如果是顶层对象，设置为""即可。
ID	对话框 ID。
Title	对话框标题。
Content	对话框内容。

AddButton

AddButton 用于添加对话框按钮，语法：

AddButton(ID, Text)

AddButton(ID, Text, Href)

ID	按钮 ID
Text	按钮标题

Href　　可选参数，字符型，指定单击按钮之后跳转的目标网页的 URL

Dialog 的属性有：

Visible：逻辑型，打开网页后是否显示 Dialog，默认为 False。

7.38 showDialog 函数

showDialog 是是我们在文件"weui.me.js"定义的一个函数，专门用于显示对话框，而且可以修改显示的标题和内容，语法：

showDialog(id,Title,Content)

ID　　对话框 ID。

Title　　对话框标题。

Content　　对话框内容。

有了 showDialog，我们可以用同一个对话框，显示不同的内容。

一个例子

HttpRequet 事件代码：

```
Select Case e.Path
   Case "test.htm"
      Dim wb As New WeUI
      With wb.AddButtonGroup("","bng2",True)
         .Add("btn1","内容1").Attribute = "onclick=""showDialog('dlg1','恭喜','恭喜您抽中大奖,记得及时领奖哦!')"""
         .Add("btn2","内容2").Attribute = "onclick=""showDialog('dlg1','成功','增加记录成功,单击确定继续增加!')"""
      End With
      With wb.AddDialog("","dlg1", "","")
         .AddButton("btnOK","确定")
      End With
      e.WriteString(wb.Build) '生成网页
End Select
```

下图是通过手机访问的效果，我们只定义了一个对话框，分别单击两个按钮，会在这个对话框显示不同的内容：





7.39 使用 TabBar

TabBar 可以包括多个页面，可以在顶端或底端显示页面切换按钮：



一个例子

上图的效果是通过下面的HttpRequest事件代码生成的，因为包括三个页面的内容，所以代码比较长：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New WeUI
        Dim txt As String = "由各种物质组成的巨型球状天体,叫做星球.星球有一定的形状,有自己的运行轨道."
        '增加三个页面,一个按钮
        With wb.AddTabBar("", "tb1", 1)
```

```
        .AddPage("page1","微信","./images/button.png")
        .AddPage("page2","通讯录","./images/msg.png")
        .AddPage("page3","发现","./images/article.png")
        .AddButton("bt1","我","./images/cell.png","http://www.foxtable.com")
    End With
    '为第一个页面增加内容
    wb.AddForm("page1","form1","test.htm")
    With wb.AddInputGroup("form1","ipg1")
        .AddInput("xm","户名","text")
        .AddInput("mm","密码","password")
    End With
    With wb.AddButtonGroup("form1","btg1",True)
        .Add("btn1", "确定", "submit")
    End With
    '为第二个页面增加内容
    With wb.AddPanelGroup("page2","pg1","图文组合列表")
        .Add("pn1","标题一",txt,"./images/button.png","http://www.foxtable.com")
        With .Add("pn2","标题二",txt,"./images/search.png")
            .AddFoot("文字来源")
            .AddFoot("时间")
            .AddFoot("|其他信息","http://www.foxtable.com")
        End With
        .GroupFoot = "查看更多"
        .GroupHref = "http://www.foxtable.com/"
    End With
    '为第三个页面增加内容
    With wb.AddArticle("page3","ar1")
        .AddTitle("h1","发现")
        .AddTitle("h2","章标题")
        .AddTitle("h3","1.1节标题")
        .AddContent(txt)
        .AddImage("./images/001.jpg")
        .AddTitle("h3","1.2节标题")
        .AddContent(txt)
        .AddImage("./images/002.jpg")
    End With
    e.WriteString(wb.Build)
End Select
```

AddTabBar

AddTabBar用于增加TabBar，语法：

AddTabBar(ParentID, ID, Positon)

ParentID　　父容器的 ID,如果是顶层对象，设置为""即可。

ID　　TabBar 的 ID

Position　　按钮位置，0 显示在底端，1 显示在顶端。

AddPage

AddPage用于增加页面，语法：

AddPage(ID, Text)
AddPage(ID, Text, Image)

ID　　页面 ID

Text　　页面按钮标题

Image　　可选参数，页面按钮图片

AddButton

AddButton 用于添加按钮，语法：

AddButton(ID, Text)
AddButton(ID, Text, Image)
AddButton(ID, Text, Image, Href)

ID　　按钮 ID

Text　　按钮标题

Image　　可选参数，

Href　　可选参数，字符型，指定单击按钮之后跳转的目标网页的 URL

7.40 使用 Page

Page 表示一个页面，一个网页可以有多个 Page，用于显示不同的内容。
Page 需要自行编写代码切换，如果需要自带切换按钮，可以使用 TabBar。

添加 Page 的语法为：

AddPage(ParentID,ID,Visible)

ParentID　　父容器 ID，如果是顶层对象，设为""即可。

ID　　页面 ID。

Visible　　可选参数，逻辑型，用于设置页面默认是否显示。默认为 True。

一个例子

假定要生成一个网页，这个网页包括两个 Page，可以相互切换：

关于Foxtable

Foxtable将Excel、Access、Foxpro、VB以及易表的优势融合在一起,无论是数据录入、查询、统计,还是报表生成,都前所未有的强大和易用,普通用户无需编写任何代码,即可轻松完成复杂的数据管理工作,真正做到拿来即用.



下一页



HttpRequest 事件代码：

```
'''
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddPage("","page1") '增加两个page
        wb.AddPage("","page2",False) '第二个
        With wb.AddArticle("page1","ar1")
            .AddTitle("h1","关于Foxtable")
            .AddContent("Foxtable将Excel、Access、Foxpro、VB以及易表的优势融合在一起,无论是数据录入、查询、统计,还是报表生成,都前所未有的强大和易用,普通用户无需编写任何代码,即可轻松完成复杂的数据管理工作,真正做到拿来即用.")
            .AddImage("./images/001.jpg")
        End With
        With wb.AddButtonGroup("page1","btg1")
            .Add("btn1", "下一页", "button").Attribute="onclick=""hide('page1');show('page2')"""
```

```
    End With
    With wb.AddArticle("page2","ar2")
      .AddTitle("h1","关于易表")
      .AddContent("易表.net介于电子表格和数据库软件之间，它有类似电子表格的界面，同时又有很多数据库软件特有的功能和灵活性，它能将复杂的操作简单化，让普通用户轻松完成复杂的数据管理和统计分析工作.")
      .AddImage("./images/002.jpg")
    End With
    With wb.AddButtonGroup("page2","btg2")
      .Add("btn2", "上一页", "button").Attribute="onclick=""hide('page2');show('page1')"""
    End With
    e.WriteString(wb.Build) '生成网页
End Select
```

7.41 使用 Toast

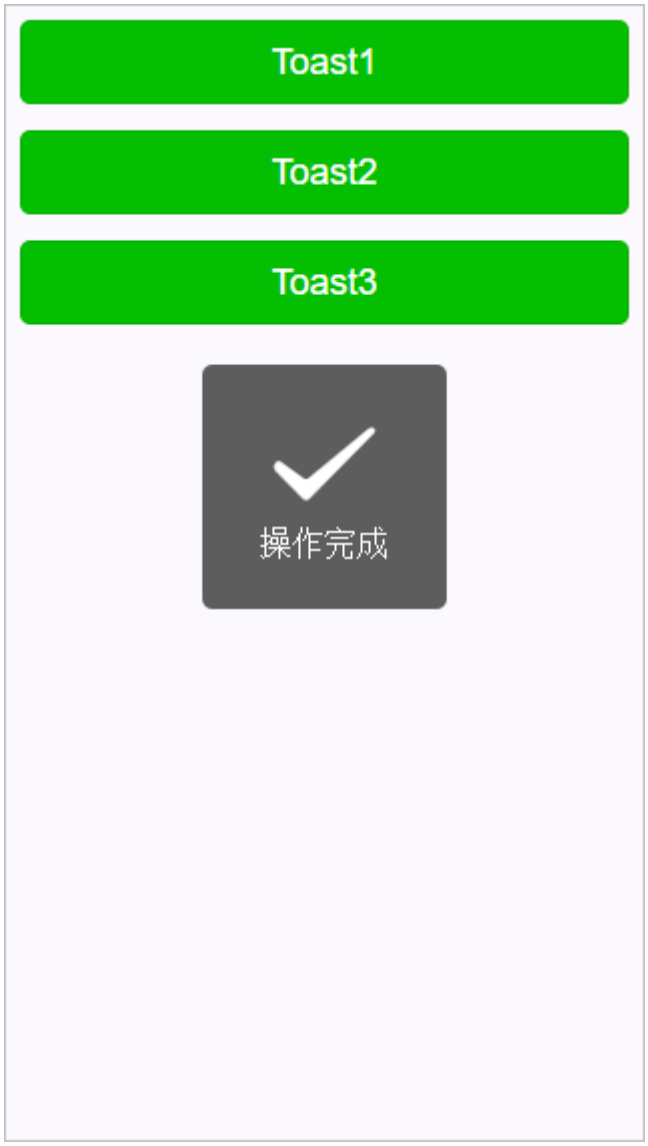
Toast 用于临时显示某些信息，并且会在数秒后自动消失。

一个例子

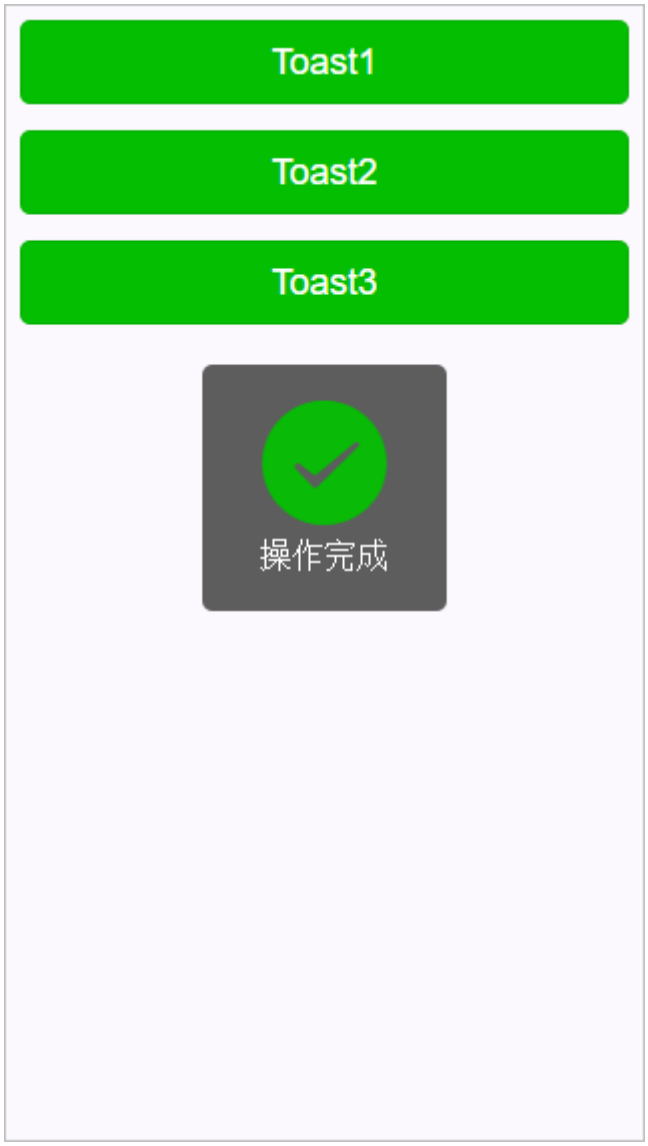
HttpRequest 事件代码:

```
Select Case e.Path
  Case "test.htm"
    Dim wb As New WeUI
    With wb.AddButtonGroup("","bng2",True)
      .Add("btn1","Toast1").Attribute  =  "onclick=""show('t1',2000)"""  '参数2000表示2秒后隐藏
      .Add("btn2","Toast2").Attribute = "onclick=""show('t2',2000)"""
      .Add("btn3","Toast3").Attribute = "onclick=""show('t3',2000)"""
    End With
    wb.AddToast("","t1", "操作完成",0)
    wb.AddToast("","t2", "正在加载",1)
    wb.AddToast("","t3", "操作完成",0).Icon= "success"
    e.WriteString(wb.Build)
End Select
```

下图是在手机中访问的效果，单击三个按钮，会分别显示三个 Toast，在 2 秒后会自动消失：


Toast1
Toast2
Toast3
操作完成


Toast1
Toast2
Toast3
正在加载



AddToast

AddToast 方法用于增加 Toast，语法：

AddToast(ParentId, ID, Text, Type)

ParentID	父容器的 ID,如果是顶层对象，设置为""即可。
ID	ToastID。
Text	Toast 文本内容。
Type	Toast 类型，默认为 0，如果设置为 1，图标将显示为一个表示正在运行的动画。

Toast 的属性有：

Visible	逻辑型，打开网页后是否显示 Toast，默认为 False。
msec	整数型，默认为 0，用于设置 Toast 初始显示的毫秒数。
Icon	字符型，指定 Toast 显示的图标，默认为"default"。
	其他可选值有"success","info","warn",对应的图标分别是：



7.42 给用户一个提示

如果表单包括文件上传组件 Uploader，那么有必要在提交表单的过程中，给用户一个提示，避免用户失去耐心：



知识准备

用户单击确定按钮提交表单的时候，会触发表单的 onsubmit 事件，我们可以在这个事件中显示一个 toast，提示用户正在上传文件，例如：

wb.AddForm("","form1","test.htm").attribute= "onsubmit=""show('t1')"""

表示定义了一个表单，提交这个表单的时候，会显示名为 t1 的 toast，不仅可以给用户一个提示，还可以避免用户再次单击确定按钮，重复提交数据。

HttpRequest 事件代码:

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddToast("","t1", "正在上传",1) '定义提示
        wb.AddForm("","form1","test.htm").attribute= "onsubmit=""show('t1')"""
        With wb.AddInputGroup("form1","ipg1","客户资料")
            .AddInput("姓名","姓名","Text").value = "舒淇"
            .AddInput("年龄","年龄","number").Value = "28"
            .AddInput("日期","日期","date").value = #10/12/2012#
            With .AddUploader("up1","照片",True)
                .AddImage("./images/shuqi1.jpg")
                .AddImage("./images/shuqi2.jpg")
            End With
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

7.43 使用 TopTips

TopTips 用于在页面顶端临时显示信息，并且会在数秒后自动消失。

一个例子

假定你设计了一个用户登录页面，如果用户单击确定按钮之前，没有输入用户名和密码，就在顶部显示一个提示信息，2 秒后自动消失：



这个例子对于 iOS 用户比较有意义，因为 required 属性对于 iOS 设备无效。

知识准备

用户单击确定按钮提交表单的时候，会触发表单的 onsubmit 事件，这个事件如果返回 false，将终止提交表单。

例如下面的定义表单的代码，表示提交表单前，先执行 valid 函数:

```
wb.AddForm("","form1","logon.htm").Attribute = "onsubmit='return valid()'"
```

如果 valid 函数返回 false，将终止提交表单。

设计步骤：

1、在"d:\web\lib"新建一个文本文件，文件名为"valid.js"，文件内容为：

```
function valid(){
   var v1 = document.getElementById("xm").value;
   var v2 = document.getElementById("pw").value;
   if (v1 && v2){return true}
   show("toptip1",2000);
```

```
    return false;
}
```

2、HttpRequest 事件代码：

```
Select Case e.Path
    Case "logon.htm"
        If e.PostValues.Count = 0 Then
            Dim wb As New WeUI
            wb.AddTopTips("","toptip1","请输入姓名和密码!")
            wb.AddForm("","form1","logon.htm").Attribute = "onsubmit='return valid()'" '调用函数
            With wb.AddInputGroup("form1","ipg1","用户登录")
                .AddInput("xm","户名","text")
                .AddInput("pw","密码","password")
            End With
            With wb.AddButtonGroup("form1","btg1",True)
                .Add("btn1", "确定", "submit")
            End With
            wb.AppendHTML("<script src='./lib/valid.js'></script>") '引入脚本文件
            e.WriteString(wb.Build)
        Else
            Dim sb As New StringBuilder
                                        sb.AppendLine("<meta                                    name='viewport'
content='width=device-width,initial-scale=1,user-scalable=0'>")
            For Each key As String In e.PostValues.Keys
                sb.AppendLine(key & ":" & e.PostValues(key) & "</br>")
            Next
            e.WriteString(sb.ToString)
        End If
End Select
```

AddTopTips

AddTopTips 用于增加 TopTips，语法：

AddTopTips(ParentId, ID, Text)

ParentID	父容器的 ID,如果是顶层对象，设置为""即可。
ID	TopTips 的 ID。
Text	TopTips 文本内容。

showTopTips

前面的例子，我们用 show 方法显示 TopTips，如果显示内容是变化的，可以改用 showTopTips 方法。

showTopTips 是是我们在文件"weui.me.js"定义的一个函数，专门用于显示 Toptips，而且可以动态指定显示内容，语法：

showTopTips(id,text,msec)

id	Toptips 的 ID
text	指定要显示的内容
msec	指定显示时长，整数型，范围为毫秒

TopTips 的属性有：

msec	整数型，默认为 0，用于设置 TopTips 初始显示的毫秒数。

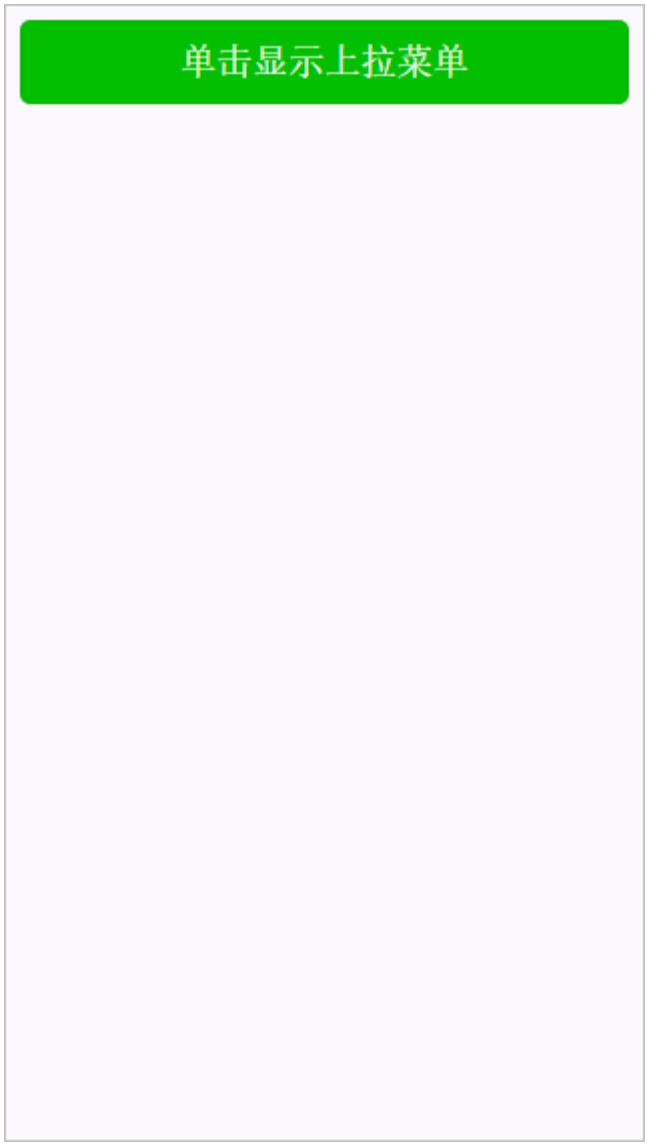
7.44 使用 ActionSheet

ActionSheet 用于显示一个从底部弹出的菜单，一般用于响应用户单击页面的动作。

一个例子

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New WeUI
        With wb.AddButtonGroup("","btg",True)
            .Add("btn1","单击显示上拉菜单").Attribute = "onclick=""show('s1')"""
        End With
        With wb.AddActionSheet("","s1")
            .Add("menu1", "菜单项目1", "http://www.foxtable.com/")
            .Add("menu2", "菜单项目2")
            .Add("menu3", "菜单项目3").Attribute = "onclick='alert(""你单击了我"")'"
            .Add("menu4","取消","",True)
        End With
        e.WriteString(wb.Build)
End Select
```

通过手机访问，会显示一个按钮，点击这个按钮，底部会弹出一个菜单：


单击显示上拉菜单



AddActionSheet

AddActionSheet 用于增加 ActionSheet，语法：

ParentID　父容器的 ID,如果是顶层对象，设置为""即可。

ID　ActionSheet 的 ID。

Add

Add 用于在 ActionSheet 中增加菜单项，语法：

Add(ID, Text)

Add(ID, Text, Href)

Add(ID, Text, Href, Separator)

ID　菜单项的 ID。

Text　菜单项的文本内容。

Href　可选参数，单击菜单项后要跳转到的目标 URL。

Separator　可选参数，逻辑型，是否在此菜单项之前显示一个分割条。

7.45 使用 Preview

Preview 通常用于表单内容预览。

一个例子

HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        With wb.AddPreview("","pv1","付款金额","￥2400")
            .AddItem("项目1","内容1")
            .AddItem("项目2","内容2")
            .AddItem("项目3", "电动打蛋机")
            .Addbutton("操作", "", 1, "onclick='alert(""你单击了我"")'")
        End With
        wb.AppendHTML("<br/>")
        With wb.AddPreview("","pv2","付款金额","￥3400")
            .AddItem("项目1","内容1")
            .AddItem("项目2","内容2")
            .AddItem("项目3", "电动打蛋机")
            .Addbutton("辅助操作", "http://www.foxtable.com", 0)
            .Addbutton("操作", "", 1, "onclick='alert(""你单击了我"")'")
        End With
        e.WriteString(wb.Build) '生成网页
End Select
```

下面是通过手机访问的显示效果：



AddPreview

AddPreview 用于增加 Preview，语法：

AddPreview(ParentID, ID, HeadText, HeadValue)

ParentID	父容器的 ID,如果是顶层对象，设置为""即可。
ID	Preview 的 ID。
HeadText	标题文本
HeadValue	标题的值

AddItem

AddItem 用于在 Preview 中增加项目，语法：

AddItem(Text, Value)
AddItem(Text, Value, Attribute)

Text	项目标题
Value	项目值
Attribute	可选参数，项目属性。

Addbutton

Addbutton 用于在 Preview 底部增加操作按钮，语法：

Addbutton(Text, Href)
Addbutton(Text, Href, Type)
Addbutton(Text, Href, Type, Attribute)

Text	按钮标题
Href	单击按钮后要跳转到的目标 URL。
Type	可选参数，设置为 0 按钮文字为灰色，设为 1 按钮文字为绿色
Attribute	可选参数，按钮属性。

7.46 使用 Gallery

Gallery 用于实现图片的展示或幻灯片播放。
我们之前介绍的 Uploader，其图片浏览窗口就是一个 Gallery。
Gallery 也可以单独创建和使用。
一个例子
HttpRequest 事件代码：

```
Select Case e.Path
   Case "test.htm"
      Dim wb As new WeUI
      With wb.AddGallery("","gla1")
         .AddImage("./images/001.jpg","./images/002.jpg","./images/003.jpg","./images/004.jpg")
      End With
      e.WriteString(wb.Build) '生成网页
End Select
```

下图是通过手机访问的效果，点击图片的右侧切换到下一副图片，单击图片左侧切换到上一副图片：



AddGallery

AddGallery 用于添加 AddGallery，语法：
AddGallery(ParentID, ID)
AddGallery(ParentID, ID, Visible)
AddGallery(ParentID, ID, Visible, AutoHide)

ParentID	父容器的 ID,如果是顶层对象，设置为""即可。
ID	Gallery 的 ID。
Visible	逻辑型，可选参数，Gallery 初始是否可见，默认为 True
AutoHide	逻辑型，可选参数，默认为 False，当需要和其他组件配合使用时，可将此属性设置为 True，这样点击图片中央位置，会自动隐藏 Gallery。

AddImage

AddImage 用于向 Gallery 添加图片，语法：
AddImage(Images)
AddImage(Image1, Image2, Image3...)

Images	图片文件的集合或数组。
Image1, Image2, Image3	要添加的多个图片文件。

7.47 结合 Article 和 Gallery

Article可能包括很多图片，有的时候，可能希望点击Article中的某个图片，能自动显示一个Gallery，集中显示所有图片：



自动实现

实现这个功能很简单，只需将Article的UserGallery属性设置为True即可，参考下面的HttpRequest事件代码：

```
Select Case e.Path
       Case "test.htm"
```

```
            Dim wb As new WeUI
            With wb.AddArticle("","ar1")
                    .UseGallery = True '启用Gallery,必须放在第一行
                    .AddTitle("h1","大标题")
                    .AddTitle("h2","章标题")
                    .AddTitle("h3","1.1节标题")
                   .AddContent("Write your Sad Times in Sand, Write your Good Times in Stone.—
George Bernard Shaw")
                    .AddImage("./images/001.jpg")
                    .AddTitle("h3","1.2节标题")
                   .AddContent("Write your Sad Times in Sand, Write your Good Times in Stone.—
George Bernard Shaw")
                    .AddImage("./images/002.jpg")
                    .AddTitle("h2","章标题")
                    .AddTitle("h3","2.1节标题")
                   .AddContent("Write your Sad Times in Sand, Write your Good Times in Stone.—
George Bernard Shaw")
                    .AddImage("./images/003.jpg")
                    .AddTitle("h3","2.2节标题")
                   .AddContent("Write your Sad Times in Sand, Write your Good Times in Stone.—
George Bernard Shaw")
                    .AddImage("./images/004.jpg")
            End With
            e.WriteString(wb.Build)
End Select
```

手工实现

你也可以自己编码完成同样的任务，自己编码的好处是可以进行更多的控制，例如排除部分图片，或为提高网页打开速度，在 Article 显示低分辨率的图，在 Gallery 中显示高分辨率的原图。你还可以用这个方法，将 Gallery 和其他元素组合使用。

HttpRequest事件代码：

```
Select Case e.Path
       Case "test.htm"
            Dim wb As new WeUI
            With wb.AddArticle("","ar1")
                    .AddTitle("h1","大标题")
                    .AddTitle("h2","章标题")
                    .AddTitle("h3","1.1节标题")
                   .AddContent("Write your Sad Times in Sand, Write your Good Times in Stone.—
George Bernard Shaw")
                    .AddImage("./images/001.jpg","onclick=""showGallery('gla1','./images/001
```

```
.jpg')""")
                        .AddTitle("h3","1.2节标题")
                       .AddContent("Write your Sad Times in Sand, Write your Good Times in Stone.—
George Bernard Shaw")
                        .AddImage("./images/002.jpg","onclick=""showGallery('gla1','./images/002
.jpg')""")
                        .AddTitle("h2","章标题")
                        .AddTitle("h3","2.1节标题")
                       .AddContent("Write your Sad Times in Sand, Write your Good Times in Stone.—
George Bernard Shaw")
                        .AddImage("./images/003.jpg","onclick=""showGallery('gla1','./images/003
.jpg')""")
                        .AddTitle("h3","2.2节标题")
                       .AddContent("Write your Sad Times in Sand, Write your Good Times in Stone.—
George Bernard Shaw")
                        .AddImage("./images/004.jpg","onclick=""showGallery('gla1','./images/004
.jpg')""")
                End With
                '增加Gallery,第三个参数False表示初始隐藏,第四个参数True表示点击图片自
动隐藏
                With wb.AddGallery("","gla1",False,True)
                        .AddImage("./images/001.jpg","./images/002.jpg","./images/003.jpg","./im
ages/004.jpg")
                End With
                e.WriteString(wb.Build) '生成网页
End Select
```

showGallery

showGallery 是在框架文件"weui.me.js"中定义的一个函数(注意 JavaScript 是区分大小写的)，用于显示 Gallery，语法为：

```
showGallery(ID, Image)
```

ID	Gallery 的 ID
Image	要在 Gallery 中显示的图片，必须事先添加到 Gallery 中。

7.48 使用 Progress

Progress 表示进度条。

一个例子

HttpRequest事件代码

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New WeUI
        wb.AddPageTitle("","ph1","Progress")
        wb.AddProgress("","pgb1",True,20).CancelCommand = "onclick=""alert('你取消了')"""
        With wb.AddButtonGroup("","btg1",)
            .Add("btn1", "设置进度").Attribute= "onclick=""setProgressValue('pgb1',80)"""
        End With
        e.WriteString(wb.Build)
End Select
```

这是在手机上的显示效果。



AddProgress

AddProgress用于增加Progress，语法：

AddProgress(ParentID, ID)
AddProgress(ParentID, ID, Visible)
AddProgress(ParentID, ID, Visible, Value)

ParentID	父容器的 ID,如果是顶层对象，设置为""即可。
ID	Progress 的 ID。
Visible	逻辑型，可选参数，Progress 初始是否可见，默认为 True
Value	整数型，可选参数，设置 Progress 初始值，范围为 0-100。

setProgressValue

setProgressValue 是我们在文件 weui.me.js 中扩展的一个函数，用于设置 Progress 的值，语法：
setProgressValue(ID,Value)

ID	Progress 的 ID。
Value	要设置的值，范围为 0-100。

getProgressValue
我们在为你教案 weui.me.js 还扩展了一个 getProgressValue 函数，用于获取 Progress 的进度值，语法：
getProgressValue(ID)

ID	Progress 的 ID。

7.49 动态生成图表

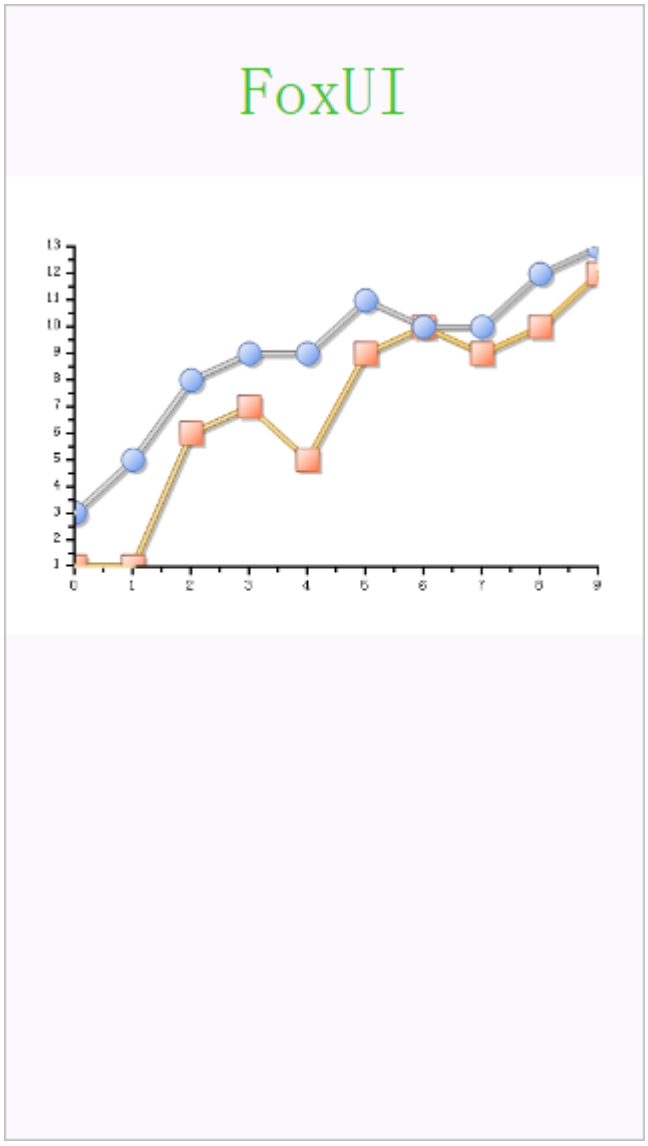
关于 Foxtable 是如何动态生成图表的，请参考：ChartBuilder

我们可以利用 Foxtable 的 ChartBuilder 动态生成图表，然后保存为图片文件，发送给客户端。
但是这样会产生大量的临时文件，为此 WeUI 提供了一个 ImageToBase64 方法，可以将 ChartBuilder 生成的图片直接转换为 base64 字符串发送给客户端，无需先保存为文件。
一个例子
HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New WeUI
        Dim Chart As New ChartBuilder
        Dim Series As WinForm.ChartSeries
        Chart.PrintWidth = 140
        Chart.PrintHeight = 100
```

```
    Chart.VisualEffect = True
    For n As Integer = 0 To 1
      Series = Chart.SeriesList.Add()
      Series.Length = 10
      For i As Integer = 0 To 9
        Series.X(i) = i
        Series.Y(i) = i + n * 2 + Rand.Next(5)
      Next
    Next
    wb.AddPageTitle("","ph1","FoxUI")
    wb.InsertHTML("<img width='100%' src='" & wb.ImageToBase64(Chart.Image) &  "'/>")
    e.WriteString(wb.Build)
End Select
```

这是显示效果：



7.50 使用 Cookie

通过 Cookie 可以在本机临时存储数据，每次访问服务器网页时，都会自动将 Cookie 中的值，传递给服务器。
HttpRequest 事件有个 Cookies 字典，包括所有的 Cookie，例如要列出所有 Cookie 的名称和值：

```
Dim sb As New StringBuilder
For Each key As String In e.Cookies.Keys
    sb.AppendLine(key & ":" & e.Cookies(key))
Next
e.WriteString(sb.Tostring)
```

HttpRequest只有一个AppendCookie方法，用于增加Cookie，使用起来不是很方便，参考使用Cookie。

WeUI 扩展了三个和 Cookie 相关的方法，分别是：

- **AppendCookie**

  用于添加 Cookie，语法为：

  AppendCookie(Name, Value, Expires)

  Name： Cookie 名称
  Value： Cookie 值
  Expires：可选参数，整数型，用于设置 Cookie 的有效时间，单位是分钟，如果不设置，关闭浏览器后 Cookie 将失效。

- **DeleteCookie**

  用于删除指定名称的 Cookie，语法：

  DeleteCookie(Name)

  Name：Cookie 名称

- **ClearCookie**

  清除全部 Cookie。

我的建议是：
尽量用 WeUI 处理 Cookie，但是也有一些特殊情况无法使用 WeUI，例如 HttpClient，此时只能使

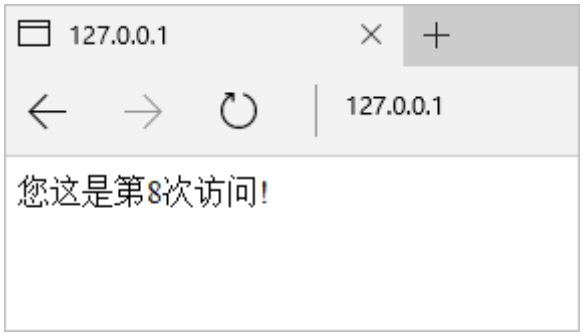
用 HttpRequest 内置的 AppendCookie 方法。

一个例子

将HttpRequest事件代码设置为：

```
Dim wb As New WeUI
Dim cnt As  Integer = 1
If e.Cookies.ContainsKey("count") '如果存在名为count的Cookie
    Integer.TryParse(e.Cookies("count"),cnt) '提取cookie的值，并转换为整数
    cnt = cnt + 1
End If
wb.AppendCookie("count",cnt) '在客户端存储Cookie
wb.InsertHTML("您这是第" & cnt & "次访问!")
e.WriteString(wb.Build)
```

现在每次刷新页面，访问次数都会递增1：



7.51 设计首页和登录页面

本节的任务是设计一个登录页面：



如果用户名和密码错误，会显示 2 秒的错误提示：



如果用户名和密码正确，进入首页：



如果在首页点击退出，会回到登录页面。

HttpRequest 事件代码：

```
Dim wb As New weui
'身份验证
Dim Verified As Boolean '用于标记用户是否通过了身份验证
Dim UserName As String = e.Cookies("username") '从 cookie 中获取用户名
Dim Password As String = e.Cookies("password") '从 cookie 中获取用户密码
If e.Path = "logon.htm" '如果是通过登录页面访问,从 PostValues 即可中提取用户名和密码
    If e.PostValues.ContainsKey("username") AndAlso e.PostValues.ContainsKey("password")  Then
        UserName = e.PostValues("username")
        Password = e.PostValues("password")
    End If
End If
If UserName = "张三" AndAlso Password = "888" Then  '实际使用的时候,请改为从数据库读取用户名和密码进行比较
    Verified  = True
```

```
ElseIf Username = "李四" AndAlso Password="999" Then
    Verified   = True
End If
If Verified AndAlso e.Path = "logon.htm"   Then '如果用户访问的是登录页,且身份验证成功
    wb.AppendCookie("username",UserName) '将用户名和密码写入 cookie
    wb.AppendCookie("password",Password)
    wb.InsertHTML("<meta http-equiv='Refresh' content='0; url=/default.htm'>") '直接跳转到首页
    e.WriteString(wb.Build) '生成网页
    Return '必须的
ElseIf Verified = False AndAlso e.Path <> "logon.htm" Then '如果用户身份验证失败,且访问的不是登录页面
    wb.InsertHTML("<meta http-equiv='Refresh' content='0; url=/logon.htm'>") '那么直接跳转到登录页面
    e.WriteString(wb.Build) '生成网页
    Return '必须的
End If
'开始生成网页
Select Case e.path
    Case "logon.htm" '登录页面
        wb.AddPageTitle("","pageheader","销售系统","由湛江辉迅基于 Foxtable 开发")
                        If                  e.PostValues.ContainsKey("username")                AndAlso
e.PostValues.ContainsKey("password")   Then '判断是否是验证失败后的重新登录
            wb.AddTopTips("","toptip1","用户名或密码错误!").msec = 2000 '如果用户通过登录按钮访问,则给用户一个 2 秒的提示.
        End If
        wb.AddForm("","form1","logon.htm")
        With wb.AddInputGroup("form1","ipg1")
            .AddInput("username","户名","text")
            .AddInput("password","密码","password")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "登录", "submit")
        End With
    Case "exit.htm" '退出登录
        wb.DeleteCookie("username") '清除 cookie 中原来的用户名和密码
        wb.DeleteCookie("password")
        wb.InsertHTML("<meta  http-equiv='Refresh'  content='0;  url=/logon.htm'>")  '那么直接跳转到登录页面
    Case "", "default.htm" '首页
        wb.AddPageTitle("","pageheader","销售系统","由湛江辉迅基于 Foxtable 开发")
        With wb.AddGrid("","g1")
            .Add("c1","增加订单",  "./images/button.png").Attribute  =  "onclick='javascript:alert(""你单击了我!"")'"
            .Add("c2","客户管理", "./images/cell.png", "http://www.foxtable.com")
```

```
        .Add("c3","销售统计", "./images/toast.png", "http://www.foxtable.com")
        .Add("c4","Dialog", "./images/dialog.png", "http://www.foxtable.com")
        .Add("c5","Progress", "./images/progress.png", "http://www.foxtable.com")
        .Add("c6","Msg", "./images/msg.png", "http://www.foxtable.com")
        .Add("c7","Article", "./images/article.png", "http://www.foxtable.com")
        .Add("c8","ActionSheet", "./images/actionSheet.png", "http://www.foxtable.com")
        .Add("c9","Icons", "./images/icons.png", "http://www.foxtable.com")
        .Add("c10","Panel", "./images/panel.png", "http://www.foxtable.com")
        .Add("c11","Tab", "./images/tab.png", "http://www.foxtable.com")
        .Add("c12","退出", "./images/exit.png", "exit.htm") '退出登录
    End With
End Select
e.WriteString(wb.Build) '生成网页
```

代码逻辑并不复杂，所有知识之前都已经讲述过，唯一没有接触过的是自动跳转网页的代码：

```
<meta http-equiv='refresh' content='2; url=/logon.htm'>
```

表示 2 秒后跳转到"/logon.htm"页面，如果你要立即跳转，将 2 改为 0 即可。

7.52 让登录变得更安全

我们上一节介绍的用户登录，无法避免同名用户重复登录，而且直接通过 Cookie 存储用户密码 在安全方面也是有欠缺的。

在安全方面，我的建议是：

1、Cookie 名称不要用 UserName 和 PassWord 等指示性很强的名称。

2、内容加密存储，关于字符的加密和机密，参考：加密和解密函数。

3、不要设置 Cookie 的有效时间，这样 Cookie 会随浏览器的关闭而消失。

如果你觉得以上措施还不足以让你在安全方面放心，而且需要禁止重复登录，可以考虑下面的方法。

设计任务

1、不能在 Cookie 存储密码，即使时加密后的密码，也不能出现在 Cookie 中。

2、同名用户在其他电脑登录后(实际开发时，可以用不同的浏览器来模拟不同电脑)，前一次的登录自动失效。

3、如果用户 30 分钟内没有操作，或关闭浏览器，登录失效。

设计思路

1、服务端新建一个临时表，包括三列，分别是：UserName，UserID 和 ActiveTime。

2、服务端在验证用户身份后，在临时表中新增一行，生成一个 16 位的随机字符串存储在 UserID 列，用户名和当前时间则分别存储在 UserName 和 ActiveTime 列。

3、完成上述操作后，再将 UserName 和 UserID 保存在 Cookie 中，其中 UserName 存储在 Cookie 中的是加密后的字符串。

4、为方便理解，本节的示例代码直接使用 username 和 userid 作为 Cookie 名称，实际开发的时候，请采用没有任何指示性的名称，注意 Cookie 名是区分大小写的，这里用的是小写。

5、用户在登录成功，继续访问其他页面时，Cookie 中会包括 username 和 userid，服务端根据 username 和 userid 从临时表中查找对应的行，此时会有三种可能：

- 如果找到，且 ActiveTime 和当前时间相比，相差不超过 30 分钟，允许访问，并将 ActiveTime 列更新为当前时间。
- 如果找到，但 ActiveTime 和当前时间相比，相差已超过 30 分钟，拒绝访问，并删除此行，然后跳转到登录页面。
- 如果没有找到，则拒绝访问，直接跳转到登录页面。

6、如果用户在其他位置登录成功，服务端根据加密后的 UserName 在临时表中查找对应的行，如果找到，说明是重复登录，删除此行，使得前一次的登录无效 ；然后重复 2、3 操作，在临时表和 Cookie 中存储加密后的 UserName 和随机生成的 UserID，并在 ActiveTime 列记录当前时间。
7、显然临时表的记录会越来越多，我们可以定期清除 ActiveTime 列和当前时间相比超过 30 分钟的行，也可以在表在超过约定行数后执行一次清理动作，本示例选择的是前者，每 30 分钟清理一次。

HttpRequest 事件代码

以下是根据上述设计思路整理出来的 HttpRequest 事件代码，由于 Cookie 中的用户名是加密存储的，而且 Cookie 不会出现密码，出现的是随机生成 UserID， 且每次登录 UserID 都不同，所以安全性会大大提高。
看起来代码有点长，这是因为完成的任务比较多，如果你分开任务看，每一个任务的代码都很精简，也很好理解：

第 1 到 14 行代码用于生成临时表和清除过期登录信息；
第 21 到 47 行代码用于根据用户输入的用户名和密码进行身份验证，并在在临时表和 Cookie 中保存登录信息；
第 49 到第 58 行代码用于从 Cookie 中提取登录信息进行身份验证，并更新最后一次活动时间；
第 63 到第 74 行代码用于生成登录页面；
第 76 到第 78 行代码用于退出登录；
第 80 到第 94 行代码用于生成首页。

```
1  Static UserTable As DataTable '定义一个变量,用于存储用户随机身份ID,以及最后一次活动时间.
2  Static ClearTime As Date
3  If UserTable Is Nothing Then '创建用于记录登录信息的临时表
4      ClearTime = Date.Now()
5      Dim dtb As New DataTableBuilder("UserInfos")
6      dtb.AddDef("UserName", Gettype(String), 16)
7      dtb.AddDef("UserID",Gettype(String),16)
8      dtb.AddDef("ActiveTime",Gettype(Date))
9      UserTable = dtb.Build(True)
1  End If
0  If (Date.Now - ClearTime).TotalMinutes >= 30 Then '清除超过30分钟没有操作的登录信息
1      UserTable.DeleteFor("ActiveTime < #" & Date.Now.AddMinutes(-30) & "#")
1      ClearTime = Date.Now()
1  End If
2  Dim wb As New weui
```

```
'身份验证
Dim UserName As String
Dim Password As String
Dim UserID As String
If e.Path = "logon.htm" '验证用户名和密码
    If e.PostValues.ContainsKey("username") AndAlso e.PostValues.ContainsKey("password")  Then
        Dim Verified As Boolean '用于标记用户是否通过了身份验证
        UserName = e.PostValues("username")
        Password = e.PostValues("password")
        If UserName = "张三" AndAlso Password = "888" Then  '实际使用的时候,请改为从数据库读取用户名和密码进行比较
            Verified  = True
        ElseIf Username = "李四" AndAlso Password="999" Then
            Verified  = True
        End If
        If Verified Then
            UserID = Rand.NextString(16) '生成随机用户ID
            UserName = EncryptText(UserName,"123","123") '将用户名加密.
            Dim dr As DataRow = UserTable.Find("UserName = '" & UserName & "'")
            If  dr IsNot Nothing Then '如果是重复登录,删除以前的登录信息
                dr.Delete()
            End If
            dr = UserTable.AddNew()
            dr("UserName") = UserName
            dr("UserID") = UserId
            dr("ActiveTime") = Date.Now '记录登录时间
            wb.AppendCookie("username",UserName) '将用户名和密码写入cookie
            wb.AppendCookie("userid",UserID)
             wb.InsertHTML("<meta  http-equiv='Refresh'  content='0;  url=/default.htm'>")  '直接跳转到首页
            e.WriteString(wb.Build) '生成网页
            Return '必须的
        End If
    End If
Else '其它页面从Cookie提取登录信息进行验证
    UserName = e.Cookies("username")  '从cookie中获取用户名
    UserID = e.Cookies("userid")  '从cookie中获取  随机ID
    Dim dr As DataRow = UserTable.Find("UserName = '" & UserName & "'")
    If dr IsNot Nothing AndAlso dr("UserID") = UserID Then  '如果通过验证,更新活动时候,继续访问其它页面.
        dr("ActiveTime") = Date.Now '更新活动时间
    Else '如果验证失败
        wb.InsertHTML("<meta http-equiv='Refresh' content='0; url=/logon.htm'>") '那么直接跳转到登录页面
```

```
        e.WriteString(wb.Build) '生成网页
        Return '必须的
    End If
End If
'开始生成网页
Select Case e.path
    Case "logon.htm" '登录页面
        wb.AddPageTitle("","pageheader","销售系统","由湛江辉迅基于Foxtable开发")
                    If          e.PostValues.ContainsKey("username")          AndAlso
e.PostValues.ContainsKey("password")  Then '判断是否是验证失败后的重新登录
            wb.AddTopTips("","toptip1","用户名或密码错误!").msec = 2000 '如果用户通过登录按钮访
问,则给用户一个2秒的提示.
        End If
        wb.AddForm("","form1","logon.htm")
        With wb.AddInputGroup("form1","ipg1")
            .AddInput("username","户名","text")
            .AddInput("password","密码","password")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "登录", "submit")
        End With
    Case "exit.htm" '退出登录
        wb.DeleteCookie("username") '清除cookie中原来的用户名和密码
        wb.DeleteCookie("password")
        wb.InsertHTML("<meta  http-equiv='Refresh'  content='0;  url=/logon.htm'>")  '然后直接跳转
到登录页面
    Case "", "default.htm" '首页
        wb.AddPageTitle("","pageheader","销售系统","由湛江辉迅基于Foxtable开发")
        With wb.AddGrid("","g1")
            .Add("c1","增加订单",  "./images/button.png").Attribute  =  "onclick='javascript:alert(""
你单击了我!"")'"
            .Add("c2","客户管理", "./images/cell.png", "http://www.foxtable.com")
            .Add("c3","销售统计", "./images/toast.png", "http://www.foxtable.com")
            .Add("c4","Dialog", "./images/dialog.png", "http://www.foxtable.com")
            .Add("c5","Progress", "./images/progress.png", "http://www.foxtable.com")
            .Add("c6","Msg", "./images/msg.png", "http://www.foxtable.com")
            .Add("c7","Article", "./images/article.png", "http://www.foxtable.com")
            .Add("c8","ActionSheet", "./images/actionSheet.png", "http://www.foxtable.com")
            .Add("c9","Icons", "./images/icons.png", "http://www.foxtable.com")
            .Add("c10","Panel", "./images/panel.png", "http://www.foxtable.com")
            .Add("c11","Tab", "./images/tab.png", "http://www.foxtable.com")
            .Add("c12","退出", "./images/exit.png", "exit.htm") '退出登录
        End With
End Select
```

```
e.WriteString(wb.Build) '生成网页
```

```
7
9
8
0
8
1
8
2
8
3
8
4
8
5
8
6
8
7
8
8
8
9
9
0
9
1
9
2
9
3
9
4
9
5
9
6
```

提示：用户表以及最近一次清理过期信息的时间，用静态变量保存，关于静态变量，参考：使用静态变量

7.53 使用 AJAX

7.53.1 使用 AJAX 函数

AJAX 可以实现网页和后台服务器的即时交互。
一般用户使用 AJAX 可能会有一些困难，所以我们提供了一些简单的 AJAX 函数，使得大家可以快速上手。
例如通过 setAjaxOptions 函数，我们只需一行代码，就 能实现动态列表项目的功能。
当然，如果你熟悉 JavaScript 和 AJAX，完全可以自行编码实现相关函数，以获得更好的灵活性。

7.53.2 setAjaxOptions

setAjaxOptions 是我们在 weui.me.js 中扩展的一个函数，用于根据后台数据动态生成列表项目，语法：

setAjaxOptions(SelectID, URL, ID1, ID2, ..., asyn)

SelectID	列表输入框的 ID。
URL	后台用于生成列表项目的网页。
ID1, ID2, ...	输入框 ID，个数不限，这些输入框的内容将以 POST 方式传递给后台，后台据此生成列表项目。
asyn	逻辑型，可选参数，是否异步执行，默认为 true，如果要同步执行，请设置为 false。 注意是 true 和 false，不是 True 和 False。

例如：
setAjaxOptions('品牌','getBrands.htm','国家',false)
表示将"国家"输入框的值，传递给后台的网页 getBrands.htm，为"品牌"输入框生成列表项目 ，注意这里的"国家"和"品牌"，都是输入框的 ID。
后台的处理代码可以非常的简单，例如根据收到的国家名称，从数据表提取品牌，回传给客户端即可：

```
Dim pps As String = DataTables("汽车").GetComboListString("品牌","国家='" & e.PostValues("国家") & "'")
e.WriteString(pps)
```

示例一
假定后台有个名为"汽车"的数据表：

	国家	品牌
1	德国	奔驰
2	德国	宝马
3	德国	奥迪
4	德国	大众
5	德国	欧宝
6	中国	比亚迪
7	中国	奇瑞
8	中国	长城
9	中国	荣威
10	中国	吉利
11	日本	丰田
12	日本	本田
13	日本	日产
14	日本	讴歌
15	日本	雷克萨斯

前台的录入界面如下图，希望选择不同的国家后，能自动根据后台数据表的内容，列出该国的汽车品牌供选择：



看起来比较复杂，其实非常简单，HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1","动态列表")
            With .AddSelect("国家","国家","|中国|德国|日本")
                .Attribute = "onchange=""setAjaxOptions('品牌','getBrands.htm','国家',false)"""
            End With
            .AddSelect("品牌","品牌","")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
        e.WriteString(wb.Build) '生成网页
    Case   "getBrands.htm"
```

```
    Dim pps As String = DataTables("汽车").GetComboListString("品牌","国家='" & e.PostValues("国家") & "'")
    e.WriteString(pps)
End Select
```

示例二

根据 setAjaxOptions 可以做出级联列表的效果，假定后台有个表 A：

	型号	规格	颜色
1	VV	4	红
2	VV	6	红
3	YJV	10	黄
4	YJV	16	绿
5	VV	10	黄
6	VV	10	蓝
7	VV	10	黑
8	YJV	16	白

希望前台的输入页面，选择不同的型号，能自动列出对应的规格供选择，选择某个规格后，还能自动列出对应的颜色供选择：



HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddInputGroup("form1","ipg1","动态列表")
            With .AddSelect("xh","型号","|" & DataTables("表A").GetComboListString("型号"))
                .Attribute = "onchange=""setAjaxOptions('gg','getProducts.htm','xh',false)"""
            End With
            With .AddSelect("gg","规格","")
                .Attribute = "onchange=""setAjaxOptions('ys','getProducts.htm','xh','gg',false)"""
            End With
            .AddSelect("ys","颜色","")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
```

```
        End With
        e.WriteString(wb.Build) '生成网页
    Case  "getProducts.htm"
        Dim vals As String
        If e.PostValues.Count = 1 Then
            vals  = "|" & DataTables("表A").GetComboListString("规格","型号='" & e.PostValues("xh") & "'")
        ElseIf e.PostValues.Count = 2 Then
            Dim Filter As String = "型号='" & e.PostValues("xh") & "' And  规格='" & e.PostValues("gg") & "'"
            vals = "|" & DataTables("表A").GetComboListString("颜色", Filter)
        End If
        e.WriteString(vals)
End Select
```

注意这里型号输入框的 ID 是"xh"，规格输入框的 id 是"gg"，setAjaxOptions 通过 ID 来提交输入框的值，例如：

setAjaxOptions('ys','getProducts.htm','xh','gg',false)

服务端也是通过 ID 来获取对应输入框的值：e.PostValues("xh")

7.53.3 submitAjaxForm

submitAjaxForm 用于不刷新当前页面的情况下提交表单数据。

submitAjaxForm(id,func,asyn)

ID	表单 ID。
func	回调函数名，收到服务器返回信息后，会调用此函数进行处理。
asyn	逻辑型，可选参数，是否异步执行，默认为 true，如果要同步执行，请设置为 false。注意是 true 和 false，不是 True 和 False。
toast	可选参数，一个 toast 的 ID，此 toast 用于显示表单数据和文件的上传进度。

例如：

submitAjaxForm('form1','myfunction');

表示异步提交表单 form1，由 myfunction 函数负责接收处理服务器返回的数据。

当 asyn 参数为 False，也可以不指定 func 参数，此时 submitAjaxForm 函数将返回一个值，此值就是服务器返回的数据，例如：

var result=submitAjaxForm('form1','',false);

表示同步提交表单 form1，并将服务器返回的数据保存在变量 result 中。

注意，这里的 func 参数设置为""，并不能直接省略。

一个例子

这次我们要设计一个录入页面：



录入完成后，单击确定按钮，在服务器成功添加订单后，会提示：



如果单击按钮“否”，可以跳转到其他页面(通常是首页)，如果单击按钮"是"，会自动清除之前已经输入的内容，开始输入下一个订单：



如果输入数据录入不完整，就单击确定按钮，会提示错误：



在错误窗口单击"确定"后，可以继续输入内容再提交：



看起来有点复杂，不过设计起来却一点不复杂。

设计步骤

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"ajaxform.js"，文件内容为：

```
function myfunction(){
    var result = submitAjaxForm('form1','',false);
    if (result =='OK') {show('dlg1')}
    else {showDialog('dlg2','错误',result)}
}
```

这个文件定义了一个 JS 函数，用于接收服务器返回的值，如果值为"OK"，显示对话框"dlg1"，否则显示对话框"dlg2",通过"dlg2"显示服务器返回的错误信息。

注意

a、JavaScript(JS)中的函数名、变量名、字符比较等等，都是区分大小写的，所以"ok"不等于"OK"，"returnval"不等于"returnVal"。

b、记得用 utf-8 格式存盘，否则中文乱码。

2、现在编写 HttpRequest 事件代码， 用页面"addnew.htm"用于输入数据，用页面"handle.htm"用于

接收用户提交的数据：

```
Select Case e.Path
    Case "addnew.htm"
        Dim wb As New weui
        wb.AddForm("","form1","handle.htm") '指定接收表单数据的的页面为handle.htm
        With wb.AddInputGroup("form1","ipg1","新增订单")
            .AddInput("客户","客户","text")
            .AddInput("日期","日期","date")
            .AddInput("产品","产品","text")
            .AddInput("数量","数量","number")
            .AddInput("单价","单价","number").Step = 0.1
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "button").Attribute= "onclick='myfunction()'"
        End With
        With wb.AddDialog("","dlg1", "提示","增加订单成功,是否继续增加?") '增加订单成功提示框
            .AddButton("btnYes","是").Attribute = "onclick='form1.reset()'"
            .AddButton("btnNo","否","http://www.foxtable.com").Kind = 1
        End With
        With wb.AddDialog("","dlg2", "错误","") '增加订单失败提示框
            .AddButton("btnOK","确定")
        End With
        wb.AppendHTML("<script src='./lib/ajaxform.js'></script>") '引入脚本文件
        e.WriteString(wb.Build)
    Case "handle.htm"
        Dim nms() As String = {"客户","日期","产品","数量","单价"}
        For Each nm As String In nms
            If e.PostValues.ContainsKey(nm) = False Then
                e.WriteString("请输入" & nm & "!") '返回错误消息
                Return '必须返回
            End If
        Next
        Dim dr As DataRow = DataTables("订单").AddNew()
        For Each nm As String In nms
            dr(nm) = e.PostValues(nm)
        Next
        dr.Save()
        e.WriteString("OK") '返回增加成功消息
End Select
```

7.53.4 submitAjaxFileds

submitAjaxFiled 用于发送表单中部分输入框的值，语法：

submitAjaxFileds(url,func,id1,id2...,asyn)

url	接收数据的目标网页
func	回调函数名，收到服务器返回信息后，会调用此函数进行处理。
id1,id2...	输入框 ID，个数不限。
asyn	逻辑型，可选参数，是否异步执行，默认为 true，如果要同步执行，请设置为 false。注意是 true 和 false，不是 True 和 False。

例如：

submitAjaxFileds('getCodes.htm','setCodes','province','county');

表示将 ID 为"province"和"county"两个输入框的值 ，异步提交到后台的"getCodes.htm"页面处理， 服务器返回的数据由函数 setCodes 函数负责接收和处理。

当 asyn 参数为 False，也可以不指定 func 参数，此时 submitAjaxFileds 函数将返回一个值，此值就是服务器返回的数据，例如：

var result=submitAjaxFileds("valid.htm","","产品","数量","折扣",false);

表示将产品、数量和折扣三个输入框的值提交到后台的"valid.htm"页面处理，并将服务器返回的结果存储在变量 result 中。

注意，这里的 func 参数设置为""，并不能直接省略。

一个例子

假定要求设计一个下图所示的输入页面，选择省市后，自动列出所有该省所有的县供选择，选择县之后，自动输入该县的区号和邮编：



当然，要完成这个设计，首先得在后台准备一个"行政区域"数据表：

	省市	县市	区号	邮编
1	北京市	北京市	10	100000
2	北京市	东城区	10	100010
3	北京市	西城区	10	100030
4	北京市	崇文区	10	100060
5	北京市	宣武区	10	100050
6	北京市	朝阳区	10	100020
7	北京市	丰台区	10	100070
8	北京市	石景山区	10	100040

设计步骤

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"ajaxform.js"，文件内容为：

```
function setCodes(){
    var result= submitAjaxFileds('getCodes.htm','','province','county',false);
    if(result){
```

```
        var vals=result.split("|");
        if(vals.length==2){
           document.getElementById("areacode").value=vals[0];
           document.getElementById("postcode").value=vals[1];
        }
    }
}
```

这个文件定义了一个 JS 函数，将 province(省)和 county(县)输入框的值同步发送到服务器，服务区返回的数据的格式为"区号|邮编"，将收到的 数据拆分之后，分别写入区号(areacode)和邮编(postcode)输入框。

2、HttpRequest 事件代码：

```
Select Case e.Path
   Case "test.htm"
      Dim wb As New weui
      wb.AddForm("","form1","test.htm")
      With wb.AddInputGroup("form1","ipg1","自动输入")
         With .AddSelect("province","省市","|"+DataTables("行政区域").GetComboListString("省市"))
            .Attribute = "onchange=""setAjaxOptions('county','getCounties.htm','province',false)"""
         End With
         With .AddSelect("county","县","")
            .Attribute = "onchange='setCodes()'" '调用js函数
         End With
         .AddInput("areacode","区号","text")
         .AddInput("postcode","邮编","text")
      End With
      With wb.AddButtonGroup("form1","btg1",True)
         .Add("btn1", "确定", "submit")
      End With
      wb.AppendHTML("<script src='./lib/ajaxform.js'></script>") '引入脚本文件
      e.WriteString(wb.Build) '生成网页
   Case  "getCounties.htm" '根据输入的省,获取县市列表
      Dim pps As String = DataTables("行政区域").GetComboListString("县市","省市='" & e.Values("province") & "'")
      e.WriteString("|" & pps)
   Case "getCodes.htm" '根据输入的省和县,获取区号有邮编
      Dim dr As DataRow = DataTables("行政区域").Find("省市='" & e.Values("province") & "' and 县市='" &
e.values("county") & "'")
      If dr IsNot Nothing Then
        e.WriteString(dr("区号") & "|" & dr("邮编"))
      End If
End Select
```

7.53.5 sendAjaxJSON

一般用户可以忽略本节内容。

Foxtable 为 WeUI 扩展了一个名为"sendAjaxJSON"的函数，用于向服务器发送 JSON 数据，语法：

sendAjaxJSON(data,url,func,asyn)

data	要发送的数据，可以是一个对象，也可以是一个 JSON 格式的字符串。
url	接收数据的目标网页。
func	回调函数名，收到服务器返回信息后，会调用此函数进行处理。
asyn	逻辑型，可选参数，是否异步执行，默认为 true，如果要同步执行，请设置为 false。注意是 true 和 false，不是 True 和 False。

例如：

```
sendAjaxJSON(val,"json.htm","afterSendJson")
```

表示将 val(JSON 数据或对象)异步发送到页面 json.htm，由 afterSendJson 函数负责处理接收到的数据。

当 asyn 参数为 False，也可以不指定 func 参数，此时 sendAjaxJSON 函数将返回一个值，此值就是服务器返回的数据，例如：

```
var result = sendAjaxJSON(val,"json.htm","",false)
```

表示将将 val(JSON 数据或对象）同步发送到页面 json.htm，将返回值保存在变量 result 中。

一个例子

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"ajaxform.js"，文件内容为：

```
//发送 JSON 对象
function sendJsonObject(){
    var obj=new Object();
    obj.name = "舒淇";
    obj.age=39;
    var result = sendAjaxJSON(obj,"json.htm","",false);
    document.getElementById("p1").innerHTML=result;
}

//发送 JSON 字符串
function sendJsonString(){
    var val= '{"name":"李云龙","age":"36"}'
    var result = sendAjaxJSON(val,"json.htm","",false);
    document.getElementById("p1").innerHTML=result;
}
```

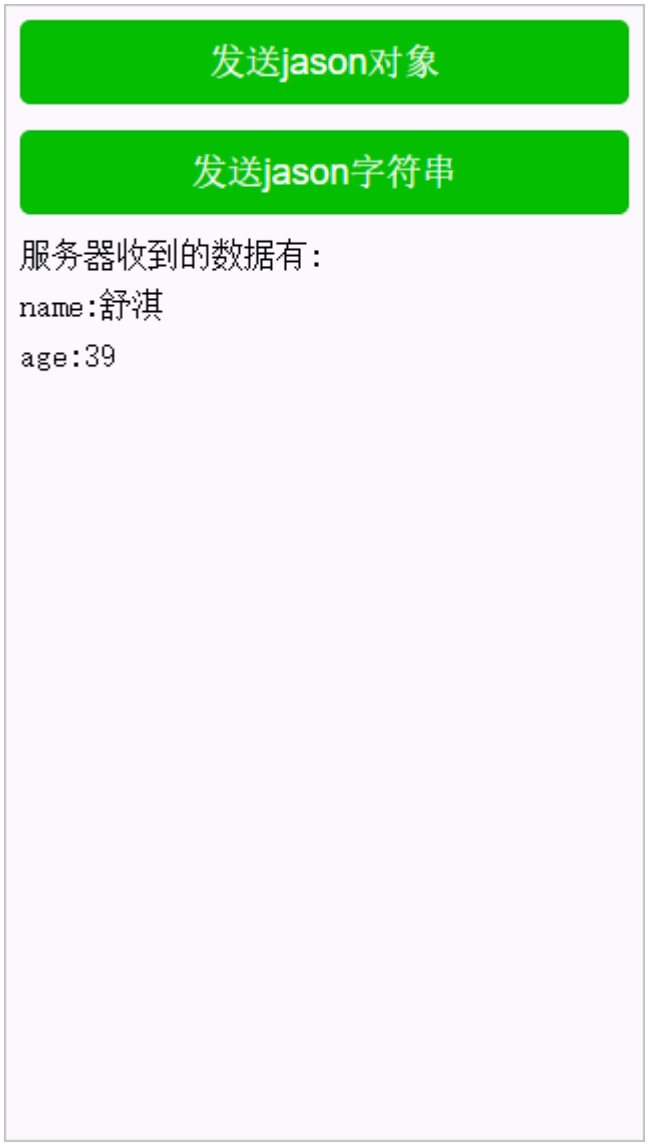
2、HttpRequest 事件代码：

```
Select Case e.Path
```

```
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddButtonGroup("form1","btg1",True)  '垂直排列
            .Add("btn1", "发送jason对象","button").Attribute="onclick='sendJsonObject()'"
            .Add("btn2", "发送jason字符串","button").Attribute="onclick='sendJsonString()'"
        End With
        wb.AppendHTML("<div id='p1' style='margin:0.5em'></div>") '插入一个div,用于显示服务器返回的数据
        wb.AppendHTML("<script src='./lib/ajaxform.js'></script>") '引入脚本文件
        e.WriteString(wb.Build)
    Case "json.htm"
        Dim sb As New StringBuilder
        Dim jo As JObject = JObject.Parse(e.PlainText) '解析JSON数据
        sb.AppendLine("服务器收到的数据有:<br/>")
        sb.AppendLine("name:" & jo("name").ToString()  & "<br/>")
        sb.AppendLine("age:" & jo("age").ToString())
        e.WriteString(sb.ToString())
End Select
```

HttpRequest将收到JSON数据以纯文本形式保存在e参数PlainText中，我们需要对其进行解析，参考：解析JSON。

此外我们这个例子，直接将服务器返回的数据显示在当前页面中，这个技巧是很实用的，望大家掌握：



7.53.6 sendAjaxText

Foxtable 为 WeUI 扩展了一个名为"sendAjaxText"的函数，用于向服务器发送纯文本数据，语法：
sendAjaxText(data,url,func,asyn)

data	要发送的文本数据，
url	接收数据的目标网页。
func	回调函数名，收到服务器返回信息后，会调用此函数进行处理。
asyn	逻辑型，可选参数，是否异步执行，默认为 true，如果要同步执行，请设置为 false。注意是 true 和 false，不是 True 和 False。

例如：

```
sendAjaxText("getUser","accept.htm","afterGetUser")
```

表示将字符串"getUser"异步发送到页面 accept.htm，由 afterGetUser 函数负责处理接收到的数据。
当 asyn 参数为 False，也可以不指定 func 参数，此时 sendAjaxText 函数将返回一个值，此值就是服务器返回的数据，例如：

```
var result = sendAjaxText("getUser","accept.htm","",false)
```

表示将字符串"getUser"同步发送到页面 accept.htm，将返回值保存在变量 result 中。
提示：
HttpRequest 的 e 参数 PlainText，用于返回接收到的文本数据。

一个例子

这个例子用于演示了如何向服务器发送文本命令，服务器根据收到的文本命令，返回相应的 JSON 数据或文本数据：



设计步骤

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"ajaxform.js"，文件内容为：

```
function getDate(){
    var result = sendAjaxText("getDate","accept.htm","",false);
    document.getElementById("p1").innerHTML="今天日期:<br/>" + result;
}

function getUser(){
    var result = sendAjaxText("getUser","accept.htm","",false);
    var user=JSON.parse(result);
    result="姓名:" + user.name + "<br/>" + "年龄:" + user.age
    document.getElementById("p1").innerHTML=result;
}
```

2、HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","test.htm")
        With wb.AddButtonGroup("form1","btg1",True)  '垂直排列
            .Add("btn1", "getUser","button").Attribute="onclick='getUser()'"
            .Add("btn2", "getDate","button").Attribute="onclick='getDate()'"
        End With
        wb.AppendHTML("<div id='p1' style='margin:0.5em'></div>") '插入一个div,用于显示服务器返回的数据
        wb.AppendHTML("<script src='./lib/ajaxform.js'></script>") '引入脚本文件
        e.WriteString(wb.Build) '生成网页
    Case "accept.htm"
        Select e.PlainText
            Case "getUser" '发送jason数据
                Dim v As String = "{""name"":""李云龙"" , ""age"":""38""}"
                e.WriteString(v)
            Case "getDate" '发送纯文本
                e.Writestring(CUDate(Date.Today))
        End Select
End Select
```

7.53.7 再谈表单验证

多数时候，我们可以用 submitAjaxForm 将表单数据发送给服务器，由服务器对输入结果进行验证处理，参考:submitAjaxForm
不过如果表单中包括文件上传组件，那么需要比较长的时间，才能将所有数据和文件上传到服务

器，验证通过也罢，如果验证失败，时间的浪费是很可惜的。

其实验证通常只是针对数据，而不是针对文件，我们可以用 submitAjaxFileds 将需要验证的数据发送到服务器，验证成功，才提交整个表单数据。

一个例子

希望设计一个下图所示的输入界面，当用户单击提交按钮的时候，可以先将产品、数量和折扣三列的数据提交到服务器，由服务器进行验证，验证通过后，再提交整个表单的数据和文件：



知识准备

用户单击确定按钮提交表单的时候，会触发表单的 onsubmit 事件，这个事件如果返回 false，将终止提交表单。

例如下面的定义表单的代码，表示提交表单前，先执行 validit 函数:

```
wb.AddForm("","form1","addnew.htm").Attribute="onsubmit='return validit()'"
```

我们在 validit 函数中将需要验证的数据，用 submitAjaxFileds 将要验证的数据提交给服务器，根据服务器返回的结果，决定是否正常提交表单。

设计步骤

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"ajaxform.js"，文件内容为：

```
function validit(){
   var result=submitAjaxFileds("valid.htm","","产品","数量","折扣",false);
   if(result=="OK"){
      return true;
   }
   else{
      showTopTips("toptip1",result,2500);
      return false;
   }
}
```

这段 js 代码在提交表单之前执行，将产品、数量和折扣提交到服务器的 valid.htm 页面进行验证，如果服务器返回 OK，则允许提交表单，否则显示一个错误提示，并禁止提交表单。

2、HttpRequest 事件代码：

```
Select Case e.Path
   Case "addnew.htm"
      If e.PostValues.Count = 0 Then
         Dim wb As New weui
         wb.AddTopTips("","toptip1","") '用于显示动态错误提示
         wb.AddForm("","form1","addnew.htm").Attribute="onsubmit='return validit()'"
         With wb.AddInputGroup("form1","ipg1","新增订单")
            .AddSelect("产品","产品","|PD01|PD02|PD03")
            .AddInput("数量","数量","number")
            .AddInput("单价","单价","number").Step = 0.1
            .AddInput("折扣","折扣","number").Step = 0.01
            With .AddUploader("up1","图片",True)
               .TextPosition = 0 '标题靠左
               .AddImage("./images/shuqi2.jpg")
            End With
         End With
         With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定","submit")
         End With
         wb.AppendHTML("<script src='./lib/ajaxform.js'></script>") '引入脚本文件
         e.WriteString(wb.Build)
      Else
         Dim sb As New StringBuilder
                                    sb.AppendLine("<meta                                    name='viewport'
content='width=device-width,initial-scale=1,user-scalable=1'>")
         sb.AppendLine("我已经收到您提交的数据和文件")
```

```
            e.WriteString(sb.Tostring)
        End If
    Case "valid.htm"
        If  e.PostValues.ContainsKey(" 产 品 ")  AndAlso  e.PostValues.ContainsKey(" 数 量 ")  AndAlso
e.PostValues.ContainsKey("折扣") Then
            If e.PostValues("产品") = "PD01" AndAlso e.PostValues("数量") > 1000 Then
                e.WriteString("PD01库存只剩1000啦!")
            ElseIf e.PostValues("产品") = "PD01" AndAlso e.PostValues("折扣") > 0.1 Then
                e.WriteString("PD01的最大允许折扣为0.1哦")
            Else
                e.WriteString("OK")
            End If
        Else
            e.WriteString("请完整输入订单内容!")
        End If
End Select
```

7.53.8 接收完整的页面

使用 Ajax 可以从服务器接收完整的页面，并将接收到的页面显示在当前页面中。
例如我们希望设计一个下图所示的统计页面，用户单击统计按钮后，能直接在当前页面显示统计结果：



设计步骤：

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"ajaxform.js"，文件内容为：

```
function tongji(){
    var result = submitAjaxForm('form1','',false);
    document.getElementById("p1").innerHTML=result;
}
```

2、HttpReqquest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddForm("","form1","tongji.htm")
        With wb.AddInputGroup("form1","ipg1","销售统计")
            .AddSelect("水平分组","水 平分组","产品|客户|雇员")
            .AddSelect("垂直分组","垂直分组","产品|[客户]|雇员")
            .AddSelect("统计列","统计列","数量|金额")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "统计", "button").Attribute= "onclick= 'tongji()'"
```

```
        End With
        wb.AppendHTML("<div id='p1' style='margin:0.5em'></div>") '插入一个div,用于显示服务器返回的 页面
        wb.AppendHTML("<script src='./lib/ajaxform.js'></script>") '引入脚本文件
        e.WriteString(wb.Build)
    Case "tongji.htm"
        Dim wb As New weui
        If e.PostValues.ContainsKey("垂直分组") AndAlso e.PostValues.ContainsKey("水平分组") AndAlso e.PostValues.ContainsKey("统计列") Then
            Dim b As New CrossTableBuilder("统计表1",DataTables("订单"))
            b.HGroups.AddDef(e.PostValues("水平分组"))
            b.VGroups.AddDef(e.PostValues("垂直分组"))
            b.Totals.AddDef(e.PostValues("统计列"))
            wb.AddTable("","Table1").CreateFromDataTable(b.Build(True))
        Else
            wb.InsertHTML("请按输入分组列和统计列!")
        End If
        e.WriteString(wb.Build)
End Select
```

分页显示

我们也可以分开两个 Page，一个 Page 用于统计设置，一个 Page 用于显示统计结果，两个 Page 可以来切换：



设计起来也很简单：

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"ajaxform.js"，文件内容为：

```
function tongji2(){
    var result = submitAjaxForm('form1','',false);
    document.getElementById("page2").innerHTML=result;
    hide('page1');
    show('page2');
}
```

2、HttpRequest 事件代码：

```
Select Case e.Path
    Case "test.htm"
        Dim wb As New weui
        wb.AddPage("","page1") '增加两个 page
        wb.AddPage("","page2")
        wb.AddForm("page1","form1","tongji.htm") '表单显示在第一个页面
        With wb.AddInputGroup("form1","ipg1","销售统计")
            .AddSelect("水平分组","水品分组","产品|客户|雇员")
            .AddSelect("垂直分组","垂直分组","产品|[客户]|雇员")
            .AddSelect("统计列","统计列","数量|金额")
        End With
```

```
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "统计", "button").Attribute= "onclick= 'tongji2()'"
        End With
        wb.AppendHTML("<script src='./lib/ajaxform.js'></script>") '引入脚本文件
        e.WriteString(wb.Build)
    Case "tongji.htm"
        Dim wb As New weui
        If  e.PostValues.ContainsKey("垂直分组") AndAlso e.PostValues.ContainsKey("水平分组")
AndAlso e.PostValues.ContainsKey("统计列") Then
            Dim b As New CrossTableBuilder("统计表 1",DataTables("订单"))
            b.HGroups.AddDef(e.PostValues("水平分组"))
            b.VGroups.AddDef(e.PostValues("垂直分组"))
            b.Totals.AddDef(e.PostValues("统计列"))
            wb.AddTable("","Table1").CreateFromDataTable(b.Build(True))
        Else
            wb.InsertHTML("请按输入分组列和统计列!")
        End If
        With wb.AddButtonGroup("","btg1",True)
            .Add("btn2", "重新统计", "button").Attribute= "onclick =""hide('page2');show('page1')"""
        End With
        e.WriteString(wb.Build)
End Select
```

7.53.9 使用回调函数

一般用户可以忽略本节的内容。

我们前面关于 AJAX 函数的示例，采用的都是同步执行方式，只有获得服务器返回的数据，才能进行其他操作。

所有的 AJAX 函数都可以异步执行，异步执行的好处是在等待服务器响应的过程中，可以进行其他操作，坏处是在操作过程中，你看到的数据，可能并不是最新的。

要异步执行 AJAX 函数，省略 asyn 参数即可。

除了 setAjaxOptions 参数，其他 AJAX 函数在异步执行的时候，都必须定义一个回调函数，客户端在收到服务器返回的数据后，将调用此函数。

所有的回掉函数都必须定义一个参数，这个参数表示服务器返回的值，例如：

```
function afterSendJson(result){
    document.getElementById("p1").innerHTML=result;
}
```

唯一的参数 result 表示服务器返回的数据，上面代码的意思就是：将服务器返回的数据显示在 id 为"p1"的元素中。

我们以 sendAjaxJSON 为例，介绍一下如何使用回掉函数，该函数的语法为：

```
sendAjaxJSON(data,url,func,asyn)
```

data	要发送的数据，可以是一个对象，也可以是一个 JSON 格式的字符串。
url	接收数据的目标网页。
func	回调函数名，收到服务器返回信息后，会调用此函数进行处理。
asyn	逻辑型，可选参数，是否异步执行，默认为 true，如果要同步执行，请设置为 false。注意是 true 和 false，不是 True 和 False。

例如：

```
sendAjaxJSON(val,"json.htm","afterSendJson")
```

表示将 val(JSON 数据或对象)异步发送到页面 json.htm，由 afterSendJson 函数负责处理接收到的数据。

一个例子

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"ajaxform.js"，文件内容为：

```
//异步发送对象
function sendJsonObjectAsyn(){
    var obj=new Object();
    obj.name = "hehui";
    obj.age=39;
    sendAjaxJSON(obj,"json.htm","afterSendJson"); //afterSendJson 为发送完成后要执行的对象
}

//异步发送 JSON 字符串
function sendJsonStringAsyn(){
    var val= '{"name":"李云龙","age":"36"}';
    sendAjaxJSON(val,"json.htm","afterSendJson");//afterSendJson 为发送完成后要执行的对象
}

//显示服务器返回的数据
function afterSendJson(result){
    document.getElementById("p1").innerHTML=result;
}
```

2、HttpRequest 事件代码：

```
Select Case e.Path
   Case "test.htm"
      Dim wb As New weui
      wb.AddForm("","form1","test.htm")
      With wb.AddButtonGroup("form1","btg1",True)  '垂直排列
         .Add("btn1", "发送 jason 对象","button").Attribute="onclick='sendJsonObjectAsyn()'"
         .Add("btn2", "发送 jason 字符串","button").Attribute="onclick='sendJsonStringAsyn()'"
      End With
      wb.AppendHTML("<div id='p1' style='margin:0.5em'></div>") '插入一个 div,用于显示服务器返回的数据
      wb.AppendHTML("<script src='./lib/ajaxform.js'></script>") '引入脚本文件
      e.WriteString(wb.Build) '生成网页
```

```
    Case "json.htm"
        Dim sb As New StringBuilder
        Dim jo As JObject = JObject.Parse(e.PlainText)
        sb.AppendLine("服务器收到的数据有:<br/>")
        sb.AppendLine("name:" & jo("name").ToString & "<br/>")
        sb.AppendLine("age:" & jo("age").ToString)
        e.WriteString(sb.ToString)
End Select
```

7.54 再探 Uploader

7.54.1 增强 Uploader

从本节开始，我们利用前面介绍的 AJAX 函数，来扩展一下 Uploader 组件的功能。

Uploader 在默认情况下：
1、再次选择上传文件，会覆盖上次选择好的文件。
2、如果开启了图片删除功能，删除其中一个选择好的图片，那么所有已经选择好的图片都会被删除，需要全部重选。

多数时候，这关系不大，但是如果你需要连续拍摄多张照片上传，是没有办法的，因为每拍摄一张照片，就会覆盖之前拍摄好的照片，当然你也可以事先用手机自带的相机拍摄好多张照片，然后用 Uploader 选择上传， 不过多少有点不方便。
如果我们将 Uploader 的 Incremental 属性设置为 True，可以实现：
1、再次选择上传文件，不会覆盖之前上传的文件。
2、如果开启了图片删除功能，可以逐个删除选择好的图片。
但是，在这种情况下，表单默认的提交功能，将不会上传选择好的文件，我们只能用 submitAjaxForm 函数提交表单，不过代码依旧很简单。
一个例子
设计一个下图所示的图片上传窗口，要求:
1、可以如前所述，能重复选择文件，或连续拍摄照片。
2、能删除单个的图片，而不影响其他图片。
3、能在上传过程中，显示"正在上传"。



设计过程：

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"ajaxform.js"，文件内容为：

```
function submitForm(){
    show("tst1",2000);
    var result = submitAjaxForm('form1','afterSubmit');
}

function afterSubmit(result){
    hide("tst1");
    if (result=='OK') {
        show("tst2");
        location="upload.htm";
    }
    else{
        show("tst3",2000);
```

```
    }
}
```

提示: 这里的 submitAjaxForm 是通过异步方式运行的，用回调函数接收服务器返回的结果，因为如果用同步方式运行，将无法显示“正在上传”的提示。

2、HttpRequest 事件代码：

```
Select Case e.Path
    Case "upload.htm"
        Dim wb As New weui
        wb.AddForm("","form1","receive.htm")
        With wb.AddInputGroup("form1","ipg1","文件上传")
            With .AddUploader("up128","照片",True)
                .AllowDelete = True '允许删除
                .Incremental = True '允许  重复选择文件或连续拍照
            End With
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "button").Attribute= "onclick='submitForm()'" '调用js函数上传
        End With
        wb.AddToast("","tst1", "正在上传",1)
        wb.AddToast("","tst2", "上传成功",0)
        wb.AddToast("","tst3", "上传失败",0).Icon= "warn"
        wb.AppendHTML("<script src='./lib/ajaxform.js'></script>") '引入脚本文件
        e.WriteString(wb.Build) '生成网页
    Case "receive.htm"
        For Each key As String In e.Files.Keys
            For Each fln As String In e.Files(key)
                e.SaveFile(key,fln,"d:\web\uploadfiles\" & fln) '保存接收到的文件
            Next
        Next
        e.WriteString("OK")
End Select
```

7.54.2 微信内置浏览器问题

微信内置浏览器在页面跳转的时候，似乎存在一些问题。

以上一节的例子为例，在所有独立的浏览器测试都没有问题，但是在微信内置浏览器测试时，发现上传结束后一直显示"上传成功"，并不会重新打开"upload.htm"。

要解决这个问题，可以将代码改为：

```
function submitForm(){
    show("tst1",2000);
    var result = submitAjaxForm('form1','afterSubmit');
```

```
}

function afterSubmit(result){
    hide("tst1");
    if (result=='OK') {
        show("tst2");
        location="upload.htm?v="+Math.random();
    }
    else{
        show("tst3",2000);
    }
}
```

通过在网址后面附加一个随机的 get 参数，确保浏览器重新加载页面，而不会使用缓存。

7.54.3 图片压缩上传

如今手机拍照的分辨率非常高，多数都是千万级别的像素，照片文件动辄好几 M。

有时我们并不需要这么高的分辨率，这种高分辨率的照片不仅耽误了上传时间，也增加了服务器的处理负担。

UpLoader 提供了两个属性，用于对要上传的照片进行自动压缩处理，这两个属性分别为：

- ScaleWidth
  整数型，用于设置图片压缩后的宽度，单位为像素。

- ScaleHeight
  整数型，用于设置图片压缩后的高度，单位为像素。

如果压缩后图片的长宽比例要厚原图保持一致，那么 ScaleWidth 和 ScaleHeight 只能设置一个，另一个由系统按原图长宽比例自动计算得出。
要实现图片的压缩上传，单单设置 ScaleWidth 或 ScaleHeight 是不够的，我们还需要将 Uploader 的 Incremental 属性设置为 True，所以和上一节一样，我们只能用 submitAjaxForm 函数提交表单，不过代码依旧很简单。
一个例子
设计一个下图所示的图片上传窗口，和上一节相比，新增第 4 点的要求：
1、能重复选择文件，或连续拍摄照片。
2、能删除单个的图片，而不影响其他图片。
3、能在上传过程中，显示"正在上传"。
4、所有图片在上传前，宽度统一压缩到 400 个像素，高度则按比例压缩。



设计过程：

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"ajaxform.js"，文件内容和上一节完全相同：

```
function submitForm(){
    show("tst1",2000);
    var result = submitAjaxForm('form1','afterSubmit');
}

function afterSubmit(result){
    hide("tst1");
    if (result=='OK') {
        show("tst2");
        location="upload.htm";
    }
    else{
        show("tst3",2000);
```

```
    }
}
```

提示: 这里的 submitAjaxForm 是通过异步方式运行的，用回调函数接收服务器返回的结果，因为如果用同步方式运行，将无法显示“正在上传”的提示。

2、的 HttpRequest 事件代码如下，和上一节相比，只是增加了一行代码（加粗显示的这行）而已：

```
Select Case e.Path
    Case "upload.htm"
        Dim wb As New weui
        wb.AddForm("","form1","receive.htm")
        With wb.AddInputGroup("form1","ipg1","文件上传")
            With .AddUploader("up128","照片",True)
                .AllowDelete = True '允许删除
                .Incremental = True '允许重复选择文件或连续拍照
                .ScaleWidth = 400 '自动压缩图片宽度为 400 个像素,高度等比例压缩
            End With
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "button").Attribute= "onclick='submitForm()'" '调用 js 函数上传
        End With
        wb.AddToast("","tst1", "正在上传",1)
        wb.AddToast("","tst2", "上传成功",0)
        wb.AddToast("","tst3", "上传失败",0).Icon= "warn"
        wb.AppendHTML("<script src='./lib/ajaxform.js'></script>") '引入脚本文件
        e.WriteString(wb.Build) '生成网页
    Case "receive.htm"
        For Each key As String In e.Files.Keys
            For Each fln As String In e.Files(key)
                e.SaveFile(key,fln,"d:\web\uploadfiles\" & fln) '保存接收到的文件
            Next
        Next
        e.WriteString("OK")
End Select
```

7.54.4 图片旋转上传

手机可以横着拍照，也可以竖着拍照，但通过手机拍照上传的时候，Uploader 并不知道这个照片的拍摄方向。

典型的问题是，在 iPhone 中竖着拍照上传，Uploader 会横过来显示照片，服务端接收到的照片也是横过来的。

为解决这个问题，我们为 Uploader 增加了一个整数型属性 Rotate，用于设置图片的旋转角度，可选值有：

1　　顺时钟旋转 90°

2　　顺时钟旋转 180°

3　　顺时钟旋转 270°

通常在 iPhone 上竖方向拍照上传的时候，需要将 Rotate 属性设置为 1。

要实现图片的旋转上传，单单设置 Rotate 属性是不够的，我们还需要将 Uploader 的 Incremental 属性设置为 True，所以和上一节一样，我们只能用 submitAjaxForm 函数提交表单，不过代码依旧很简单。

一个例子

希望图片顺时钟旋转 90°上传，并将宽度压缩为 400 个像素，高度则等比例压缩，设计步骤：

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"ajaxform.js"，文件内容和上一节完全相同：

```
function submitForm(){
    show("tst1",2000);
    var result = submitAjaxForm('form1','afterSubmit');
}

function afterSubmit(result){
    hide("tst1");
    if (result=='OK') {
        show("tst2");
        location="upload.htm";
    }
    else{
        show("tst3",2000);
    }
}
```

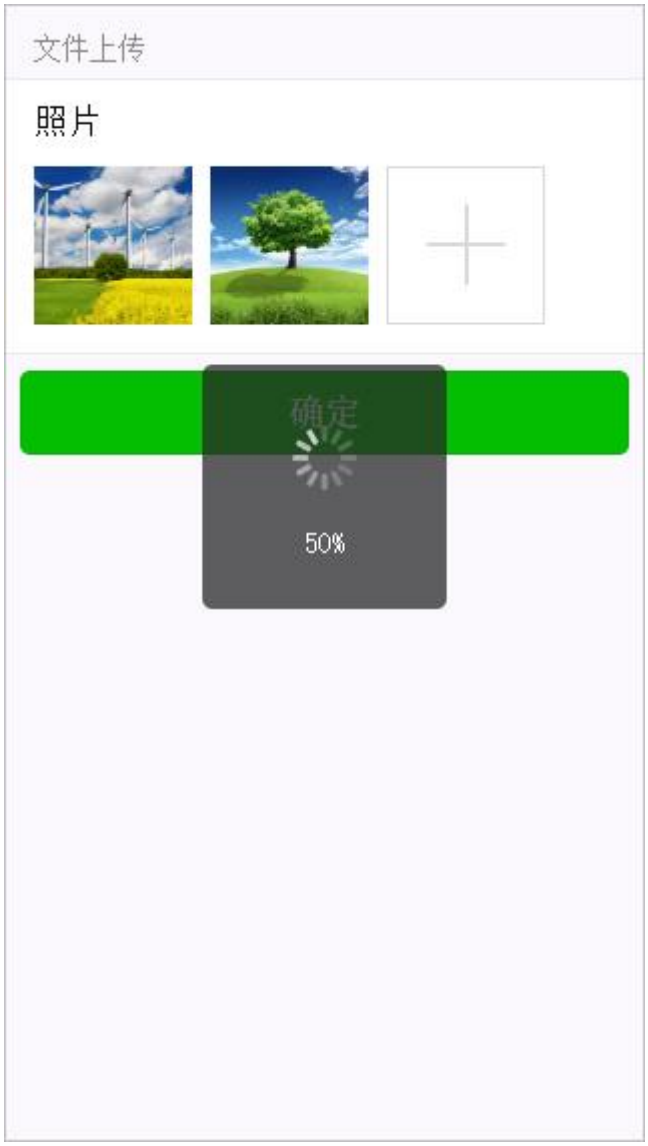
2、的 HttpRequest 事件代码如下，和上一节相比，只是增加了一行代码（加粗显示的这行）而已：

```
Select Case e.Path
    Case "addnew.htm"
        Dim wb As New weui
        wb.AddForm("","form1","receive.htm")
        With wb.AddInputGroup("form1","ipg1","文件上传")
            With .AddUploader("up128","照片",True)
                .AllowDelete = True '允许删除
                .Incremental = True '允许重复选择文件或连续拍照
                .ScaleWidth = 400 '自动压缩图片宽度为400个像素,高度等比例压缩
                .Rotate = 1 '顺时钟旋转90°
            End With
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "button").Attribute= "onclick='submitForm()'" '调用js函数上传
```

```
        End With
        wb.AddToast("","tst1", "正在上传",1)
        wb.AddToast("","tst2", "上传成功",0)
        wb.AddToast("","tst3", "上传失败",0).Icon= "warn"
        wb.AppendHTML("<script src='./lib/ajaxform.js'></script>") '引入脚本文件
        e.WriteString(wb.Build) '生成网页
    Case "receive.htm"
        For Each key As String In e.Files.Keys
            For Each fln As String In e.Files(key)
                e.SaveFile(key,fln,"d:\web\uploadfiles\" & fln) '保存接收到的文件
            Next
        Next
        e.WriteString("OK")
End Select
```

7.54.5 显示上传进度

本节的任务是在上一节的基础上，加上上传进度显示：



实现起来非常简单：

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"ajaxform.js"，文件内容为：

```
function submitWithProgress(){
    var result = submitAjaxForm('form1','afterSubmit',true,'tst1');
}



function afterSubmit(result){
    hide("tst1");
    if (result=='OK') {
        show("tst2");
        location="upload.htm";
    }
    else{
        show("tst3",2000);
```

```
    }
}
```

提示：

a、submitAjaxForm 函数的第 4 个参数，用于指定一个 Toast 的 ID，系统将通过这个 Toast 显示上传进度。

b、这里同样是采用异步上传，所以第三个参数要设置为 true(不是 True)。

2、HttpRequest 事件代码则完全不变(注意 JS 函数名改了)：

```
Select Case e.Path
    Case "upload.htm"
        Dim wb As New weui
        wb.AddForm("","form1","receive.htm")
        With wb.AddInputGroup("form1","ipg1","文件上传")
            With .AddUploader("up128","照片",True)
                .AllowDelete = True '允许删除
                .Incremental = True '允许重复选择文件或连续拍照
            End With
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "button").Attribute= "onclick='submitWithProgress()'" '调用js函数上传
        End With
        wb.AddToast("","tst1", "正在上传",1)
        wb.AddToast("","tst2", "上传成功",0)
        wb.AddToast("","tst3", "上传失败",0).Icon= "warn"
        wb.AppendHTML("<script src='./lib/ajaxform.js'></script>") '引入脚本文件
        e.WriteString(wb.Build) '生成网页
    Case "receive.htm"
        For Each key As String In e.Files.Keys
            For Each fln As String In e.Files(key)
                e.SaveFile(key,fln,"d:\web\uploadfiles\" & fln) '保存接收到的文件
            Next
        Next
        e.WriteString("OK")
End Select
```

7.55 使用表格

7.55.1 生成表格

腾讯的 WeUI 样式库并不包括表格，为方便普通用户，Foxtable 对此进行了扩展。

一个例子

我们先用一个简单的例子，看看如何增加表格，HttpRequest 事件代码：

```
Select Case e.Path
    Case "table.htm"
        Dim wb As New WeUI
        With wb.AddTable("","Table1")
            .head.AddRow("部门","姓名","年龄","电话","地址") '表头
            .body.AddRow("技术部","张三","36","110","中国北京") '数据
            .body.AddRow("技术部","李四","38","110","中国上海")
            .body.AddRow("技术部","王五","39","110","中国深圳")
            .body.AddRow("生产部","赵六","39","110","中国深圳")
            .body.AddRow("生产部","刘七","39","110","中国深圳")
        End With
        e.WriteString(wb.Build)
End Select
```

下图是通过手机访问的效果：

部门	姓名	年龄	电话	地址
技术部	张三	36	110	中国北京
技术部	李四	38	110	中国上海
技术部	王五	39	110	中国深圳
生产部	赵六	39	110	中国深圳
生产部	刘七	39	110	中国深圳

AddTable

增加表格的语法是：

AddTable(ParentID, ID)

ParentID　　父容器的 ID,如果是顶层对象，设置为""即可

ID　　表 ID

AddRow

表格有三个对象，分别为 Head(表头)、Body（表体）、Foot（表尾），这三个对象都有 AddRow 方法，用于增加行。

AddRow 方法的参数就是新增行各列的值。

AddRow 还可以直接用数组或集合作为参数，非常方便：

```
Dim vals() As String = {"生产部","刘七","39","110","中国深圳"}
.body.AddRow(vals)
```

Class 和 Attribute

表有 Class 和 Attribute 属性。

行只有 Attribute 属性，可通过 Attribute 属性设置其 Class。

例如将 HttpRequet 事件代码设置为：

```
Select Case e.Path
    Case "table.htm"
        Dim wb As New WeUI
        wb.AppendHTML("<style>.mark{background-color:red; color:white;}</style>", True) '添加样式
        With wb.AddTable("","Table1")
            .Attribute= "border='2'"
            .head.AddRow("部门","姓名","年龄","电话","地址")
            .body.AddRow("技术部","张三","36","110","中国北京")
            .body.AddRow("技术部","李四","38","110","中国上海")
            .body.AddRow("技术部","王五","39","110","中国深圳").Attribute = "class='mark'"
            .body.AddRow("生产部","赵六","39","110","中国深圳")
            .body.AddRow("生产部","刘七","39","110","中国深圳")
        End With
        e.WriteString(wb.Build)
End Select
```

下图是通过手机访问的效果，表的边框加厚了，第三行变成了红底白字：

部门	姓名	年龄	电话	地址
技术部	张三	36	110	中国北京
技术部	李四	38	110	中国上海
技术部	王五	39	110	中国深圳
生产部	赵六	39	110	中国深圳
生产部	刘七	39	110	中国深圳

7.55.2 多层表头

使用 WeUI 框架，不需要任何技巧，即可实现多层表头。

一个例子

HttpRequest 事件代码：

```
Select Case e.Path
    Case "table.htm"
        Dim wb As New WeUI
        With wb.AddTable("","Table1")
            .head.Addrow("部门","姓名","信息","信息","信息")
            .head.AddRow("部门","姓名","年龄","电话","地址")
            .body.AddRow("技术部","张三","36","110","中国北京")
            .body.AddRow("技术部","李四","38","110","中国上海")
            .body.AddRow("技术部","王五","39","110","中国深圳")
```

```
        .body.AddRow("生产部","赵六","39","110","中国深圳")
        .body.AddRow("生产部","刘七","39","110","中国深圳")
    End With
    e.WriteString(wb.Build)
End Select
```

这是在手机上的显示效果：

部门	姓名	信息		
		年龄	电话	地址
技术部	张三	36	110	中国北京
技术部	李四	38	110	中国上海
技术部	王五	39	110	中国深圳
生产部	赵六	39	110	中国深圳
生产部	刘七	39	110	中国深圳

原理很简单，正常填入表头各单元格的值，Foxtable会自动合并表头中内容相同的单元格。

7.55.3 合并模式

使用 WeUI 框架，实现合并模式是很简单的。

一个例子

HttpRequest 事件代码：

```
Select Case e.Path
  Case "table.htm"
    Dim wb As New WeUI
    With wb.AddTable("","Table1")
      .MergeCols = 2 '合并左边2列
      .head.Addrow("国家","金属","类型","数量")
      .body.AddRow("美国","白银","出口","70")
      .body.AddRow("美国","白银","进口","321")
      .body.AddRow("美国","黄金","出口","78")
      .body.AddRow("美国","黄金","进口","789")
      .body.AddRow("中国","白银","出口","380")
      .body.AddRow("中国","白银","进口","809")
      .body.AddRow("中国","黄金","出口","289")
      .body.AddRow("中国","黄金","进口","668")
    End With
    e.WriteString(wb.Build)
```

```
End Select
```

这是通过手机访问的效果：

国家	金属	类型	数量
美国	白银	出口	70
		进口	321
	黄金	出口	78
		进口	789
中国	白银	出口	380
		进口	809
	黄金	出口	289
		进口	668

提示：MergeCols 用于指定要合并的列数，从左边第一列开始计算。

7.55.4 显示行号

通过表格的 RowHead 属性，可以左边多少列作为行头，行头通常用于显示行号。
一个例子：

```
Select Case e.Path
    Case "table.htm"
        Dim wb As New WeUI
        With wb.AddTable("","Table1")
            .RowHead = 1 '左边第一列作为行头
            .head.AddRow("","部门","姓名","年龄","电话","地址") '表头,行号列标题为空
            .body.AddRow("1","技术部","张三","36","110","中国北京") '数据
            .body.AddRow("2","技术部","李四","38","110","中国上海")
            .body.AddRow("3","技术部","王五","39","110","中国深圳")
            .body.AddRow("4","生产部","赵六","39","110","中国深圳")
            .body.AddRow("5","生产部","刘七","39","110","中国深圳")
        End With
        e.WriteString(wb.Build)
End Select
```

下图是再手机上显示的效果：

	部门	姓名	年龄	电话	地址
1	技术部	张三	36	110	中国北京
2	技术部	李四	38	110	中国上海
3	技术部	王五	39	110	中国深圳
4	生产部	赵六	39	110	中国深圳
5	生产部	刘七	39	110	中国深圳

7.55.5 设置列宽

默认情况下，系统会自动分配列宽。

通过 ColWidth 属性，可以指定列宽，这是一个字符型的属性，可以一次设置多列的宽度。

一个例子

下面的 HttpRequest 事件代码，将第一列(行号)的宽度设置为 12px，第三列（姓名）的宽度设置为 120px：

```
Select Case e.Path
    Case "table.htm"
        Dim wb As New WeUI
        With wb.AddTable("","Table1")
            .RowHead = 1
            .ColWidth = "12px,,120px" '设置列宽
            .head.AddRow("","部门","姓名","年龄","电话","地址")
            .body.AddRow("1","技术部","张三","36","110","中国北京")
            .body.AddRow("2","技术部","李四","38","110","中国上海")
            .body.AddRow("3","技术部","王五","39","110","中国深圳")
            .body.AddRow("4","生产部","赵六","39","110","中国深圳")
            .body.AddRow("5","生产部","刘七","39","110","中国深圳")
        End With
        e.WriteString(wb.Build)
End Select
```

这是在手机上的显示效果：

	部门	姓名	年龄	电话	地址
1	技术部	张三	36	110	中国北京
2	技术部	李四	38	110	中国上海
3	技术部	王五	39	110	中国深圳
4	生产部	赵六	39	110	中国深圳
5	生产部	刘七	39	110	中国深圳

7.55.6 高亮显示行列

在手机中显示查阅一个大表格是很痛苦的，为解决这个问题，Foxtable 提供了高亮显示行列的功能。点击行号，会高亮显示对应的行：

	部门	姓名	年龄	电话	地址
1	技术部	张三	36	110	中国北京
2	技术部	李四	38	110	中国上海
3	技术部	王五	39	110	中国深圳
4	生产部	赵六	39	110	中国深圳
5	生产部	刘七	39	110	中国深圳

点击列标题，会高亮显示对应的列：

	部门	姓名	年龄	电话	地址
1	技术部	张三	36	110	中国北京
2	技术部	李四	38	110	中国上海
3	技术部	王五	39	110	中国深圳
4	生产部	赵六	39	110	中国深圳
5	生产部	刘七	39	110	中国深圳

点击某单元格，会高两显示对应的行列：

	部门	姓名	年龄	电话	地址
1	技术部	张三	36	110	中国北京
2	技术部	李四	38	110	中国上海
3	技术部	王五	39	110	中国深圳
4	生产部	赵六	39	110	中国深圳
5	生产部	刘七	39	110	中国深圳

Foxtable 为表提供了一个整数型的 Highlight 属性，可以用来关闭高亮显示功能：

值	说明
0	高亮显示行列
1	仅高亮显示行
2	仅高亮显示列
-1	关闭高亮显显示

一个例子

下面的 HttpRequest 事件代码关闭了表格的高亮显示功能：

```
Select Case e.Path
    Case "table.htm"
        Dim wb As New WeUI
        With wb.AddTable("","Table1")
            .Highlight = -1  '关闭高亮显示功能
            .RowHead = 1
            .head.AddRow("","部门","姓名","年龄","电话","地址")
            .body.AddRow("1","技术部","张三","36","110","中国北京")
            .body.AddRow("2","技术部","李四","38","110","中国上海")
            .body.AddRow("3","技术部","王五","39","110","中国深圳")
            .body.AddRow("4","生产部","赵六","39","110","中国深圳")
            .body.AddRow("5","生产部","刘七","39","110","中国深圳")
        End With
        e.WriteString(wb.Build)
End Select
```

7.55.7 交替行背景颜色

除了上一节介绍的高亮显示行列功能，我们还可以设置交替行背景颜色，只是此功能默认是关闭的。

表有一个整数型的 Alternate 属性，用于设置每多少行显示一个不同背景颜色的行。

一个例子

```
Select Case e.Path
    Case "table.htm"
        Dim wb As New WeUI
        With wb.AddTable("","Table1")
            .Highlight = -1 '关闭高亮显示
            .Alternate = 2 '每两行显示一个不同背景颜色的行
            .RowHead = 1
            .head.AddRow("","部门","姓名","年龄","电话","地址")
            .body.AddRow("1","技术部","张三","36","110","中国北京")
            .body.AddRow("2","技术部","李四","38","110","中国上海")
            .body.AddRow("3","技术部","王五","39","110","中国深圳")
            .body.AddRow("4","生产部","赵六","39","110","中国深圳")
            .body.AddRow("5","生产部","刘七","39","110","中国深圳")
        End With
        e.WriteString(wb.Build)
End Select
```

这是在手机上显示的效果：

	部门	姓名	年龄	电话	地址
1	技术部	张三	36	110	中国北京
2	技术部	李四	38	110	中国上海
3	技术部	王五	39	110	中国深圳
4	生产部	赵六	39	110	中国深圳
5	生产部	刘七	39	110	中国深圳

7.55.8 逐个单元格添加

前面的例子，都是整行整行地向表中增加行，简单方便。
如果需要更多的控制，可以逐个单元格添加。

一个例子

HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "table.htm"
        Dim wb As New WeUI
        wb.AppendHTML("<style>.mark{background-color:red; color:white;}</style>", True) '添加样式
        With wb.AddTable("","Table1")
            .head.AddRow("部门","姓名","年龄","电话","地址")
            .body.AddRow("技术部","张三","36","110","中国北京") '整行增加
```

```
        With .body.AddRow()
          .AddCell("技术部") '逐个单元格增加
          .AddCell("李四","class='mark'") '第二个参数用于设置单元格的Attribute属性
          .AddCells("38","110") '用AddCells可以一次添加多个单元格
          .AddCell("中国上海")
        End With
        .body.AddRow("技术部","王五","39","110","中国深圳") '整行增加
    End With
    e.WriteString(wb.Build)
End Select
```

下图是通过手机访问的显示效果，一个单元格变成了红底白字：

部门	姓名	年龄	电话	地址
技术部	张三	36	110	中国北京
技术部	李四	38	110	中国上海
技术部	王五	39	110	中国深圳

AddCell

AddCell 用于向行中添加单元格，语法：

AddCell(Value)

AddCell(Value, Attribute)

Value	要在单元格显示的内容。
Attribute	可选参数，用于设置单元格元素的属性

提示：表有 Class 和 Attribute 属性；行只有 Attribute 属性；单元格没有任何属性，但是通过 AddCell 的 Attribute 参数可以设置 Attribute 属性。

AddCells

AddCells 用于向行中一次添加多个单元格，语法：

AddCells(Values)

AddCells(Value1,Value2,Value3...)

Values	一个包括各单元格值的集合或数组。
Value1,Value2,Value3	各单元格的值。

7.55.9 由 Table 自动生成

WeUI 可以根据 Foxtable 中的 Table 自动生成网页。

一个例子

```
Select Case e.Path
    Case "table.htm"
        Dim wb As New WeUI
        With wb.AddTable("","Table1")
            .CreateFromTable(Tables("表名"),True)
        End With
        e.WriteString(wb.Build)
End Select
```

我用一个有多层表头的Table测试，用上面的代码生成了下图所示的网页：

	产品	东部				南部			
		一季度	二季度	三季度	四季度	一季度	二季度	三季度	四季度
1	PD01	7177	6378	6390	6849	7870	3035	3959	5488
2	PD02	5181	4730	6159	6299	5152	4128	4314	3601
3	PD03	4473	4497	6976	6536	4022	7582	5293	5042
4	PD04	6801	4298	5696	7039	7342	6809	5253	7409
5	PD05	3441	5016	6861	4359	4090	4394	6748	7250
6	PD06	3660	3779	6240	7613	3000	4018	3078	4237
7	PD07	7044	6240	7701	6291	6633	6204	6194	5136
8	PD08	6526	7981	5729	5638	6279	6777	5457	7634

生成上面的表其实只用了一行代码，这就是使用框架的优势：

```
.CreateFromTable(Tables("表名"),True)
```

提示：

自动生成的网页，逻辑列中的True倍符号●代替，False被符号o代替，你可以自定义符号，你可以用BooleanSymbol属性自定义符号，例如：

```
With wb.AddTable("","Table1")
    .BooleanSymbol= "√×"
  .CreateFromTable(Tables("订单"))
End With
```

CreateFromTable

CreateFromTable 用于根据 Foxtable 中的 Table 自动生成网页，语法：

CreateFromTable(Table)

CreateFromTable(Table,RowNum)

CreateFromTable(Table,RowNum,OnlyVisible)

CreateFromTable(Table,RowNum,Cols)

CreateFromTable(Table,RowNum,Col1,Col2,Col3...)

参数	说明
Table	Foxtable 中的 Table。
RowNum	逻辑型，是否显示行号。
OnlyVisible	逻辑型，是否只包括可见列。
Cols	一个包括所有要显示列的列名的集合或数组，例如： Dim Cols() As String = "产品,客户,日期,数量".Split(",") .CreateFromTable(Tables("订单"),True,Cols)

Col1, Col2, Col3	字符型，用于指定要显示的列，例如： .CreateFromTable(Tables("订单"),True,"产品","客户","日期","数量")

7.55.10 由 DataTable 自动生成

WeUI 可以根据 Foxtable 中的 DataTable 自动生成网页。

一个例子

```
Select Case e.Path
    Case "table.htm"
        Dim wb As New WeUI
        With wb.AddTable("","Table1")
            '按日期顺序列出客户CS01的订单,仅显示产品/数量/单价/日期/审核等五列
            .CreateFromDataTable(DataTables("订单"),False,"客户='CS01'","日期  desc","产品","单价","数量","日期","审核")
        End With
        e.WriteString(wb.Build)
End Select
```

这是通过手机访问的效果：

产品	单价	数量	日期	审核
PD05	17.00	20	2015-06-22	●
PD01	14.40	10	2015-06-17	○
PD05	21.35	40	2015-06-16	●
PD05	21.35	360	2015-06-07	○
PD04	17.60	180	2015-05-25	○
PD02	15.20	50	2015-05-21	○
PD02	19.00	300	2015-05-19	●
PD02	19.00	200	2015-05-17	○
PD02	15.20	120	2015-05-14	○
PD02	19.00	300	2015-05-12	○
PD03	8.00	700	2015-04-25	●
PD01	14.40	150	2015-04-12	○
PD03	10.00	140	2015-04-12	○
PD02	15.20	400	2015-04-09	○
PD04	22.00	50	2015-04-05	○
PD05	21.35	180	2015-03-16	○
PD03	8.00	500	2015-03-15	○

我们只用了一行代码，就完成了一看起来有点复杂的任务:

```
.CreateFromDataTable(DataTables("订单"),False,"客户='CS01'","日期 desc","产品","单价","数量","日期","审核")
```

这就是使用框架的好处。

提示：

自动生成的网页，逻辑列中的True倍符号●代替，False被符号o代替，你可以自定义符号，你可以用BooleanSymbol属性自定义符号，例如：

```
With wb.AddTable("","Table1")
    .BooleanSymbol= "√×"
  .CreateFromDataTable(DataTables("订单"))
End With
```

CreateFromDataTable

CreateFromDataTable用于根据Foxtable的DataTable自动生成网页，语法：

CreateFromDataTable(DataTable)
CreateFromDataTable(DataTable, RowNum)
CreateFromDataTable(DataTable, RowNum, Filter)
CreateFromDataTable(DataTable, RowNum, Filter)
CreateFromDataTable(DataTable, RowNum, Filter, Sort)
CreateFromDataTable(DataTable, RowNum, Filter, Sort, DataCols)
CreateFromDataTable(DataTable, RowNum, Filter, Sort, DataCol1, DataCol2, DataCol3...)

参数	说明
DataTable	Foxtable 中的 DataTable。
RowNum	逻辑型，是否显示行号。
Filter	筛选条件
Sort	排序列
DataCols	一个包括所有要显示列的列名的集合或数组，例如： Dim nms() As String = "产品,客户,日期,数量".Split(",") .CreateFromDataTable(DataTables("订单"),True,"","",nms)
DataCol1, DataCol2, DataCol3	字符型，用于指定要显示的列，例如： .CreateFromDataTable(DataTables("订单"),False,"","","产品","单价","数量","日期","审核")

7.55.11 手工编码生成

前面我们介绍了根据DataTable和Table自动生成网页的方法，简单到只需一行代码。

如果需要对细节做更多的控制，则需要手工编码，不过也不要担心，代码依然简单，因为这是Foxtable嘛。

一个例子

下面的HttpRequest事件代码，从订单表中提取数量超过100的订单，按日期顺序生成网页表：

```
Select Case e.Path
       Case "table.htm"
              Dim wb As New WeUI
```

```
                With wb.AddTable("","Table1")
                        .Alternate = 3
                        Dim nms() As String = {"产品","客户","数量","单价","日期"}
                        .Head.AddRow(nms)
                        For Each r As DataRow In DataTables("订单").Select("数量 > 100",
"日期 Desc")
                                With .Body.AddRow(r("产品"),r("客户"),r("数量"))
                                        .AddCell(Format(r("单价"),"#0.00"))
                                        .AddCell(Format(r("日期"),"MM月dd日"))
                                End With
                        Next
                End With
                e.WriteString(wb.Build)
End Select
```

这是在手机上访问的效果：

产品	客户	数量	单价	日期
PD01	CS02	300	18.00	06月25日
PD02	CS02	400	15.20	06月23日
PD05	CS03	300	17.00	06月21日
PD05	CS02	120	17.00	06月20日
PD01	CS05	500	18.00	06月19日
PD04	CS02	160	17.60	06月18日
PD02	CS03	500	19.00	06月15日
PD02	CS02	250	15.20	06月14日
PD02	CS02	350	19.00	06月13日
PD05	CS04	300	21.35	06月10日
PD04	CS03	250	22.00	06月09日
PD05	CS01	360	21.35	06月07日
PD03	CS04	300	20.00	06月06日
PD02	CS04	200	19.00	06月05日
PD02	CS04	120	19.00	06月04日
PD05	CS04	140	17.00	06月03日
PD02	CS03	200	19.00	06月02日

加上行号

如果希望显示行号，可以参考下面的HttpRequest事件代码：

```
Select Case e.Path
       Case "table.htm"
              Dim wb As New WeUI
              With wb.AddTable("","Table1")
                     .Alternate = 3
                     .RowHead = 1
                     .ColWidth = "12px"
                     Dim nms() As String = {"","产品","客户","数量","单价","日期"}
                     .Head.AddRow(nms)
                     Dim cnt As Integer
                     For Each r As DataRow In DataTables("订单").Select("数量 > 100",
"日期 Desc")
                            cnt = cnt + 1
                            With .Body.AddRow(cnt, r("产品"), r("客户"), r("数量"))
                                   .AddCell(Format(r("单价"), "#0.00"))
                                   .AddCell(Format(r("日期"), "MM月dd日"))
                            End With
                     Next
              End With
              e.WriteString(wb.Build)
End Select
```

多层表头

假定有下图所示的一个已知结构的有多层表头的数据表：

	产品	东部				南部			
		一季度	二季度	三季度	四季度	一季度	二季度	三季度	四季度
1	PD01	7177	6378	6390	6849	7870	3035	3959	5488
2	PD02	5181	4730	6159	6299	5152	4128	4314	3601
3	PD03	4473	4497	6976	6536	4022	7582	5293	5042
4	PD04	6801	4298	5696	7039	7342	6809	5253	7409
5	PD05	3441	5016	6861	4359	4090	4394	6748	7250
6	PD06	3660	3779	6240	7613	3000	4018	3078	4237
7	PD07	7044	6240	7701	6291	6633	6204	6194	5136
8	PD08	6526	7981	5729	5638	6279	6777	5457	7634
9	PD09	3000	4105	4232	5939	6218	7940	7939	4605
10	PD10	5597	7121	5747	4695	6093	5200	5402	6860

如果要根据这个表生成网页，可以参考下面的HttpRequest事件代码：

```
Select Case e.Path
       Case "table.htm"
              Dim wb As New WeUI
              With wb.AddTable("","Table1")
                     '下面这个数组,用实际的列名.
                     Dim nms() As String = {"产品","东部_一季度","东部_二季度","东部_三季度","东部_四季度","南部_一季度","南部_二季度","南部_三季度","南部_四季度"}
                     .Head.AddRow("产品","东部","东部","东部","东部","西部","西部","西部","西部") '第一层标题
                     .Head.AddRow("产品","一季度","二季度","三季度","四季度","一季度","二季度","三季度","四季度") '第二层标题
                     For Each r As DataRow In DataTables("表名").DataRows
                            With .Body.AddRow()
                                   For Each nm As String In nms
                                          .AddCell(r(nm))
                                   Next
                            End With
                     Next
              End With
              e.WriteString(wb.Build)
End Select
```

7.55.12 编码合并单元格

通过 AddCell 方法的第二个参数，可以给单元格设置属性。
单元格有两个属性用于实现单元格合并，分别为：

- rowspan
  指定单元格在纵向跨越的行数。
- colspan
  指定单元格在横向跨越的列数。

一个例子
HttpRequest 事件代码:

```
Select Case e.Path
   Case "list.htm"
      Dim wb As New WeUI
      wb.InsertHTML("合并单元格<br/><br/>")
      With wb.AddTable("","Table1")
```

```
        With .body.AddRow
          .AddCell("1")
          .AddCell("2","rowspan=3")
          .Addcell("3")
          .Addcell("4")
        End With
        .body.AddRow("5","7","8")
        .body.AddRow("9","11","12")
        With .body.AddRow("a")
          .AddCell("b","colspan='2'")
          .Addcell("d")
        End With
      End With
      e.WriteString(wb.Build)
End Select
```

显示效果：

合并单元格

1	2	3	4
5		7	8
9		11	12
a	b		d

在添加单元格的时候，如果遇到合并单元格，必须省略此单元格的值，按顺序添加其他单元格的值。
例如第二行，我们只添加了三个值，分别为 5、7、8，由于第二个单元格被合并，所以 7、8 顺被移到第三个和第四个单元格显示。
这些都属于 HTML 的知识，有兴趣的话，可以访问以下网址学习：
http://www.w3school.com.cn/html/index.asp

7.55.13 分页显示数据

本节以 SQL Server 为例，介绍一下如何分页显示后台数据，原理和我们在快速入门这一章介绍的相同，在合成的链接地址中包括要现实的页码：
http://127.0.0.1/list.htm?page=1
下面是 HttpRequest 事件代码：

```
Select Case e.Path
    Case "list.htm"
        '获取要显示的页
        Dim page As Integer = 0 '默认page为0,显示第一页
        Dim pageRows As Integer = 15 '每页15行
        If e.GetValues.ContainsKey("page") Then   '如果地址中有page参数
            Integer.TryParse(e.GetValues("page"), page) '提取page参数
        End If
        Dim StartRow As Integer = page * pageRows + 1 '此页第一行
        Dim EndRow As Integer = (page + 1) * pageRows '此页最后一行

        '获取该页数据
        Dim cmd As New SQLCommand
        cmd.ConnectionName = "orders" '记得设置数据源名称
        cmd.CommandText = "Select Count(*) From {订单}"
        Dim Count As Integer = cmd.ExecuteScalar() '获取总的行数
        cmd.CommandText = "Select * From (Select Row_Number() Over(Order by 日期) As [NO.], 产品,客户,数量,单价,日期 From 订单) As a "
        cmd.CommandText = cmd.CommandText & "  Where [NO.]>= " & StartRow & " And [NO.] <= " & EndRow
        Dim dt As DataTable = cmd.ExecuteReader

        '根据此页数据生成网页
        Dim wb As New WeUI
        With wb.AddTable("","Table1")
            .CreateFromDataTable(dt)
        End With
        With wb.AddButtonGroup("","btg2", False)
            If page > 0 Then
```

```
            .Add("btnPrev", "上一页","","List.htm?page=" & page - 1)
        End If
        If Endrow < count Then
            .Add("btnNext", "下一页","","List.htm?page=" & page + 1)
        End If
    End With
    e.WriteString(wb.Build)
End Select
```

这是通过手机访问显示的效果：

NO.	产品	客户	数量	单价	日期
16	PD01	CS03	200.00	14.40	1999-01-14
17	PD02	CS01	100.00	15.20	1999-01-14
18	PD02	CS05	400.00	15.20	1999-01-14
19	PD05	CS04	50.00	21.35	1999-01-16
20	PD02	CS03	200.00	15.20	1999-01-17
21	PD03	CS04	40.00	10.00	1999-01-17
22	PD01	CS02	100.00	14.40	1999-01-20
23	PD01	CS03	240.00	14.40	1999-01-21
24	PD05	CS05	60.00	21.35	1999-01-22
25	PD03	CS02	700.00	8.00	1999-01-26
26	PD03	CS02	200.00	10.00	1999-01-28
27	PD04	CS03	200.00	22.00	1999-01-30
28	PD01	CS05	200.00	18.00	1999-02-01
29	PD05	CS02	150.00	17.00	1999-02-01
30	PD05	CS04	700.00	21.35	1999-02-01

上一页 下一页

7.55.14 生成汇总模式

有了前面的知识，生成汇总模式是很轻松的。
下面是按产品和客户分组，对数量和金额进行统计的 HttpRequest 事件代码：

```
Select Case e.Path
    Case "list.htm"
        '获取要显示的页
        Dim page As Integer = 0 '默认page为0,显示第一页
        Dim pageRows As Integer = 13 '每页13行
        If e.GetValues.ContainsKey("page") Then   '如果地址中有page参数
            Integer.TryParse(e.GetValues("page"), page) '提取page参数
        End If
        Dim StartRow As Integer = page * pageRows + 1 '此页第一行
        Dim EndRow As Integer = (page + 1) * pageRows '此页最后一行
        '获取该页数据
        Dim cmd As New SQLCommand
        cmd.ConnectionName = "orders" '记得设置数据源名称
        cmd.CommandText = "Select Count(*) From {订单}"
        Dim Count As Integer = cmd.ExecuteScalar() '获取总的行数
        cmd.CommandText = "Select * From (Select Row_Number() Over(Order by 产品,客户,日期) As [NO.], 产品, 客户, 数量, 单价, 数量 * 单价 as 金额,日期 From 订单) As a "
        cmd.CommandText = cmd.CommandText & "  Where [NO.]>= " & StartRow & " And [NO.] <= " & EndRow
        Dim dt As DataTable = cmd.ExecuteReader
        '根据此页数据生成网页
        Dim wb As New WeUI
        With wb.AddTable("","Table1")
            Dim nms() As String
            Dim qty As Integer
            Dim amt As Double
            .Head.AddRow("No.","产品","客户","数量","单价","金额","日期")
            For i As Integer = 0 To dt.DataRows.count - 1
                Dim r As DataRow = dt.DataRows(i)
                If i > 0 Then
                    Dim lr As DataRow = dt.DataRows(i-1)
                    If r("客户") <> lr("客户") Then
                        qty = dt.compute("sum(数量)","产品='" & lr("产品") & "' And 客户= '" & lr("客户") & "'")
                        amt = dt.compute("sum(金额)","产品='" & lr("产品") & "' And 客户= '" & lr("客户") & "'")
                        With .Body.AddRow()
                            .Attribute = "style='background-color:#F0FFFF'"
                            .AddCell("小计 " & lr("客户"),"colspan='3'")
                            .AddCells(qty,"",amt,"")
```

```
                End With
            End If
            If r("产品") <> lr("产品") Then
                qty = dt.compute("sum(数量)","产品='" & lr("产品") &  "'")
                amt = dt.compute("sum(金额)","产品='" & lr("产品") & "'")
                With .Body.AddRow()
                    .Attribute = "style='background-color:#FFFFE0'"
                    .AddCell("小计  " & lr("产品"),"colspan='3'")
                    .AddCells(qty,"",amt,"")
                End With
            End If
        End If
        .Body.AddRow(r("NO."),r("产品"),r("客户"),r("数量"),r("单价"),r("金额"),r("日期"))
    Next
    qty = dt.compute("sum(数量)")
    amt = dt.compute("sum(金额)")
    If EndRow >= Count Then
        .Body.AddRow("总计","","",qty,"",amt,"").Attribute = "style='background-color:#98FB98'"
    End If
End With
With wb.AddButtonGroup("","btg2", False)
    If page > 0 Then
        .Add("btnPrev", "上一页","","List.htm?page=" & page - 1)
    End If
    If Endrow < count   Then
        .Add("btnNext", "下一页","","List.htm?page=" & page + 1)
    End If
End With
e.WriteString(wb.Build)
End Select
```

代码逻辑很简单，按产品和客户顺序，分页显示数据，每当产品或客户发生变化时，就插入一个分组行显示该产品或客户的累计数量和金额。
需要注意的是，分组行的背景没有采用外部 CSS 文件设置，直接使用了内嵌样式。

这是在手机上的显示效果：

No.	产品	客户	数量	单价	金额	日期
131	PD03	CS03	300	10	3000	2016-02-26
132	PD03	CS03	160	10	1600	2016-05-11
133	PD03	CS03	20	10	200	2016-05-19
小计 CS03			480		4800	
134	PD03	CS04	40	10	400	2016-01-17
135	PD03	CS04	60	10	600	2016-04-10
136	PD03	CS04	100	10	1000	2016-04-16
137	PD03	CS04	300	10	3000	2016-06-13
小计 CS04			500		5000	
138	PD03	CS05	200	10	2000	2016-02-10
139	PD03	CS05	300	10	3000	2016-02-21
140	PD03	CS05	400	9	3600	2016-04-11
小计 CS05			900		8600	
小计 PD03			1880		18400	
141	PD04	CS01	50	22	1100	2016-04-05
142	PD04	CS01	180	17.6	3168	2016-05-25
小计 CS01			230		4268	
143	PD04	CS02	210	22	4620	2016-03-01

上一页 下一页

7.55.15 数据筛选

本节的任务使设计一个筛选页面，输入输入条件，单击确定后，能从后台筛选出符合条件的行显示：





产品	客户	数量	单价	日期
PD02	CS02	160.00	15.20	2016-05-11
PD02	CS01	300.00	19.00	2016-05-12
PD02	CS04	60.00	19.00	2016-05-12
PD02	CS01	120.00	15.20	2016-05-14
PD02	CS01	200.00	19.00	2016-05-17
PD02	CS01	300.00	19.00	2016-05-19
PD02	CS05	600.00	19.00	2016-05-19

HttpRequest 事件代码：

```
Dim wb As New WeUI
Select Case e.Path
    Case "filter.htm"
        wb.AddForm("","form1","list.htm")
        With wb.AddInputGroup("form1","ipg1","数据筛选")
            .AddSelect("product","产品","PD01|PD02|PD03|PD04|PD05")
            .AddInput("startdate","开始日期","date")
            .AddInput("enddate","结束时间","date")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
    Case "list.htm"
        Dim flt As String
        If e.PostValues.ContainsKey("product") Then
            flt = "产品  = '" & e.PostValues("product") & "'"
        End If
        If e.PostValues.ContainsKey("startdate") Then
            If flt > "" Then
```

```
            flt = flt & " and "
        End If
        flt = flt & "日期  >= '" & e.PostValues("startdate") & "'"
    End If
    If e.PostValues.ContainsKey("enddate") Then
        If flt > "" Then
            flt = flt & " and "
        End If
        flt = flt & "日期  <= '" & e.PostValues("enddate") & "'"
    End If
    Dim cmd As new SQLCommand
    cmd.ConnectionName = "orders"
    cmd.CommandText= "select  产品,客户,数量,单价,[日期] From  订单"
    If flt > "" Then
        cmd.CommandText = cmd.CommandText & " where " & flt
    End If
    With wb.AddTable("","Table1")
        .CreateFromDataTable(cmd.ExecuteReader)
    End With
    With wb.AddButtonGroup("","btg1",True)
        .Add("btn1", "重新筛选", "","filter.htm")
    End With
End Select
e.WriteString(wb.Build)
```

7.55.16 数据筛选与分页

筛选后的数据，可能还是太多，依然需要分页处理：


数据筛选
产品
PD01
开始日期
2016-01-01
结束时间
2016-12-31
确定
NO.
产品
客户
数量
单价
日期
11 PD01 CS03 40.00 18.00 2016-02-07
12 PD01 CS04 30.00 18.00 2016-02-17
13 PD01 CS02 150.00 14.40 2016-02-22
14 PD01 CS04 400.00 14.40 2016-02-23
15 PD01 CS05 180.00 14.40 2016-02-23
16 PD01 CS02 200.00 14.40 2016-02-26
17 PD01 CS05 300.00 18.00 2016-02-27
18 PD01 CS04 250.00 18.00 2016-03-02
19 PD01 CS03 400.00 18.00 2016-03-03
20 PD01 CS02 100.00 18.00 2016-03-04
上一页
下一页
筛选

NO.	产品	客户	数量	单价	日期
11	PD01	CS03	40.00	18.00	2016-02-07
12	PD01	CS04	30.00	18.00	2016-02-17
13	PD01	CS02	150.00	14.40	2016-02-22
14	PD01	CS04	400.00	14.40	2016-02-23
15	PD01	CS05	180.00	14.40	2016-02-23
16	PD01	CS02	200.00	14.40	2016-02-26
17	PD01	CS05	300.00	18.00	2016-02-27
18	PD01	CS04	250.00	18.00	2016-03-02
19	PD01	CS03	400.00	18.00	2016-03-03
20	PD01	CS02	100.00	18.00	2016-03-04



如何筛选，如何分页，前面都有讲述，并不复杂。

将二者组合起来，首先必须解决的问题是如何保存和传递筛选条件，当然我们可以像页码一样，将条件写在 url 地址中，用 get 方式传递。

但是这样的设计，会增加编码难度，我们改用 cookie 来保存和传递筛选条件，会方便很多，关于 cookie，参考：使用 cookie

下面是一小段示例代码，清晰地说明了如何通过 cookie 来保存和传递筛选条件：

```
Dim flt As String
If e.GetValues.ContainsKey("unfilter") Then
    wb.ClearCookie()
ElseIf e.PostValues.ContainsKey("product") Then
    flt = "产品 = '" & e.PostValues("product") & "'"
    wb.AppendCookie("product", e.PostValues("product"))
ElseIf e.Cookies.ContainsKey("product") Then
    flt = "产品 = '" & e.Cookies("product") & "'"
End If
```

代码流程如下：

1、首先判断访问请求中是否包括get参数"unfilter"，如果包括，则清除所有Cookie。
2、然后判断PostValues中是否包括product，如果有则从PostValues中提起此值合成筛选条件，然后将product的值保存在cookie中。
3、如果PostValues中是不包括product，就判断cookie中是否包括product，如果有，则提取此值合成筛选条件。

使用流程如下：

1、用户通过筛选页面输入筛选条件，单击确定按钮提交到服务器，此时product的值是保存在PostValues中的，服务端从PostValues中提取出product合成筛选条件，然后将product的值存入客户端的cookie中。
2、当用户单击上一页、下一页按钮时，保存在cookie中的product值，会自动发送到服务器，服务端从cookie中提取出product值合成筛选条件。
3、如果用户单击"取消筛选"按钮，向服务器发出访问请求"list.htm?unfilter=true"，服务段收到请求后，判断请求中包括get参数"unfilter"，如是清除cookie，系统回到非筛选状态。

完整的HttpRequest事件代码：

```
Dim wb As New WeUI
Select Case e.Path
    Case "filter.htm"
        wb.AddForm("","form1","list.htm")
        With wb.AddInputGroup("form1","ipg1","数据筛选")
            .AddSelect("product","产品","|PD01|PD02|PD03|PD04|PD05")
            .AddInput("startdate","开始日期","date")
            .AddInput("enddate","结束时间","date")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
            .Add("btn1", "确定", "submit")
        End With
    Case "list.htm"
        '合成条件
        Dim flt As String
        If e.GetValues.ContainsKey("unfilter") Then '如果有unfilter参数,则清除cookie
            wb.ClearCookie()
        ElseIf e.PostValues.Count > 0 Then '如果是filter.htm访问,则根据用户输入合成条件表达式
            If e.PostValues.ContainsKey("product") Then
                flt = "产品  = '" & e.PostValues("product") & "'" '合成条件
                wb.AppendCookie("product", e.PostValues("product")) '将值写入cookie中
            Else
                wb.DeleteCookie("product") '删除cookie
            End If
            If e.PostValues.ContainsKey("startdate") Then
```

```
        If flt > "" Then
            flt = flt & " and "
        End If
        flt = flt & "日期 >= '" & e.PostValues("startdate") & "'"
        wb.AppendCookie("startdate", e.PostValues("startdate"))
    Else
        wb.DeleteCookie("startdate")
    End If
    If e.PostValues.ContainsKey("enddate") Then
        If flt > "" Then
            flt = flt & " and "
        End If
        flt = flt & "日期 <= '" & e.PostValues("enddate") & "'"
        wb.AppendCookie("enddate", e.PostValues("enddate"))
    Else
        wb.DeleteCookie("enddate")
    End If
Else '否则根据Cookie合成条件表达式
    If e.Cookies.ContainsKey("product") Then
        flt = "产品 = '" & e.Cookies("product") & "'"
    End If
    If e.Cookies.ContainsKey("startdate") Then
        If flt > "" Then
            flt = flt & " and "
        End If
        flt = flt & "日期 >= '" & e.Cookies("startdate") & "'"
    End If
    If e.Cookies.ContainsKey("enddate") Then
        If flt > "" Then
            flt = flt & " and "
        End If
        flt = flt & "日期 <= '" & e.Cookies("enddate") & "'"
    End If
End If
'获取要显示的页码
Dim page As Integer = 0 '默认page为0,显示第一页
Dim pageRows As Integer = 10 '每页10行
If e.GetValues.ContainsKey("page") Then  '如果地址中有page参数
    Integer.TryParse(e.GetValues("page"), page) '提取page参数
End If
Dim StartRow As Integer = page * pageRows + 1 '此页第一行
Dim EndRow As Integer = (page + 1) * pageRows '此页最后一行
'提取此页数据
Dim cmd As New SQLCommand
```

```
    cmd.ConnectionName = "orders" '记得设置数据源名称
    cmd.CommandText = "Select Count(*) From {订单}"
    If flt > "" Then
      cmd.CommandText = cmd.CommandText & " where " & flt
    End If
    Dim Count As Integer = cmd.ExecuteScalar() '获取总的行数
    cmd.CommandText = "Select * From (Select Row_Number() Over(Order by 日期) As [NO.], 产品,客户,数量,
单价,日期 From 订单 "
    If flt > "" Then
      cmd.CommandText = cmd.CommandText & " where " & flt
    End If
    cmd.CommandText = cmd.CommandText & ") As a "
    cmd.CommandText = cmd.CommandText & "  Where [NO.]>= " & StartRow & " And [NO.] <= " & EndRow
    '合成网页
    With wb.AddTable("","Table1")
      .CreateFromDataTable(cmd.ExecuteReader)
    End With
    With wb.AddButtonGroup("","btg2", False)
      If page > 0 Then
        .Add("btnPrev", "上一页","","List.htm?page=" & page - 1)
      End If
      If Endrow < count Then
        .Add("btnNext", "下一页","","List.htm?page=" & page + 1)
      End If
      If flt  = "" Then
        .Add("btn1", "筛选", "","filter.htm").kind = 1
      Else
        .Add("btn1", "取消筛选", "","list.htm?unfilter=true").kind = 1
      End If
    End With
End Select
e.WriteString(wb.Build)
```

7.55.17 给表格加上菜单

给表加上菜单的代码很简单，我们首先要设计一个 ActionSheet，然后将此 ActionSheet 的 ID 设置为表的 ActiveSheet 属性即可。
例如 HttpRequest 事件代码：

```
Select Case e.Path
```

```
    Case "table.htm"
        Dim wb As New WeUI
        With wb.AddTable("","Table1")
            .head.AddRow("部门","姓名","年龄","电话","地址") '表头
            .body.AddRow("技术部","张三","36","110","中国北京") '数据
            .body.AddRow("技术部","李四","38","110","中国上海")
            .body.AddRow("技术部","王五","39","110","中国深圳")
            .body.AddRow("生产部","赵六","39","110","中国深圳")
            .body.AddRow("生产部","刘七","39","110","中国深圳")
            .ActiveSheet = "menu" '指定菜单
        End With
        With wb.AddActionSheet("","menu") '设计菜单
            .Add("mnudAdd", "增加订单")
            .Add("mnuEdit", "编辑订单")
            .Add("mnuDelete", "删除订单")
            .Add("mnuCancel","取消","",True)
        End With
        e.WriteString(wb.Build)
End Select
```

打开网页后，先点击选择某个单元格，然后再次点击此单元格，即可出现菜单。

也就是说，第一次点击是选择，第二次点击是显示菜单：



单单显示菜单是没有意义的，目的是通过菜单增加行，或编辑和删除选定的行，新增行好办，但是要实现编辑和删除行，就必须给服务器传递当前行的主键，这样服务器才能知道客户端需要编辑和删除的是哪一行，最好也将当前的页码一并传递给服务器，这样编辑和删除行之后，还能回到当前页，这是我们下一节要解决的问题。

7.55.18 传递主键和页码

上一节说到，要实现通过菜单编辑和删除行，必须给客户端传递行的主键和页码。

传递页码很简单，生成表格的时候，直接将其 PageNumber 属性设置为当前页码即可。

传递主键要分两种情况，如果是 CreateFromDataTable 和 CreateFromTable 自动生成表格，那么设置表格的 Primarykey 为主键列的列名即可；如果是手工编码生成，则需要逐行设置主键值。

本节先介绍第一种情况，第二种情况留待下一节讲述。

在编写 JavaScript 代码的时候，表格同样有 pagenumber 和 primarykey(注意是小写)，分别用于返回当前页码和选定行的主键值。

一个例子

HttpRequest 事件代码：

```
Select Case e.Path
    Case "list.htm"
        '获取要显示的页
        Dim page As Integer = 0 '默认 page 为 0,显示第一页
        Dim pageRows As Integer = 15 '每页 15 行
        If e.GetValues.ContainsKey("page") Then  '如果地址中有 page 参数
            Integer.TryParse(e.GetValues("page"), page) '提取 page 参数
        End If
        Dim StartRow As Integer = page * pageRows + 1 '此页第一行
        Dim EndRow As Integer = (page + 1) * pageRows '此页最后一行
        '获取该页数据
        Dim cmd As New SQLCommand
        cmd.ConnectionName = "orders" '记得设置数据源名称
        cmd.CommandText = "Select Count(*) From {订单}"
        Dim Count As Integer = cmd.ExecuteScalar() '获取总的行数
        cmd.CommandText = "Select * From (Select Row_Number() Over(Order by 日期) As [NO.],[_Identify],产品,客户,数量,单价,日期 From 订单) As a "
        cmd.CommandText = cmd.CommandText & "  Where [NO.]>= " & StartRow & " And [NO.] <= " & EndRow
        Dim dt As DataTable = cmd.ExecuteReader
        '生成菜单
        Dim wb As New WeUI
        With wb.AddActionSheet("","menu") '设计菜单
            .Add("mnudAdd", "增加订单").Attribute="onclick=""alert('增加订单')"""
            .Add("mnuEdit", "编辑订单").Attribute ="onclick=""alert('编辑订单 页码:'+table1.pagenumber+ '主键:'+ table1.primarykey)"""
            .Add("mnuDelete", "删除订单").Attribute ="onclick=""alert('删除订单 页码:'+table1.pagenumber+ '主键:'+ table1.primarykey)"""
            .Add("mnuCancel","取消","",True)
        End With
        '根据此页数据生成网页
        With wb.AddTable("","Table1")
            .PageNumber = page '设置页码
            .Primarykey = "_Identify" '设置主键
            .ActiveSheet = "menu" '设置菜单
            .CreateFromDataTable(dt)
        End With
        With wb.AddButtonGroup("","btg2", False)
            If page > 0 Then
                .Add("btnPrev", "上一页","","List.htm?page=" & page - 1)
            End If
            If Endrow < count Then
```

```
            .Add("btnNext", "下一页","","List.htm?page=" & page + 1)
        End If
    End With
    e.WriteString(wb.Build)
End Select
```

因为要传递主键给客户端，所以 select 语句必须将主键列包括进来，接下来你要做的知识指定页码和主键：

```
With wb.AddTable("","Table1")
    .PageNumber = page '设置页码
    .Primarykey = "_Identify" '设置主键
    .ActiveSheet = "menu" '设置菜单
    .CreateFromDataTable(dt)
End With
```

需要注意的是：页码、主键和菜单名，都需要在执行 CreateFromDataTable 之前设置好，否则无效。

现在你在客户端浏览器连续点击某行，会出现菜单，在点击菜单中的命令，会显示当前页码和当前行的主键。

下图是我在 iPhone 按着上述操作后截图：



7.55.19 手工编码传递主键

如果因为特殊原因，不能采用 CreateFromDataTable 或 CreateFromTable 自动生成表格，那就需要逐行设置 Primarykey 属性了 ，例如：

```
Dim wb As New WeUI
With wb.AddTable("","Table1")
    Dim nms() As String = {"产品","客户","数量","单价","日期"}
    .Head.AddRow(nms)
    For Each r As DataRow In DataTables("订单").DataRows
```

```
        With .Body.AddRow(r("产品"),r("客户"),r("数量"))
          .Primarykey = r("_Identify") '设置主键
          .AddCell(Format(r("单价"),"#0.00"))
          .AddCell(Format(r("日期"),"MM月dd日"))
        End With
      Next
End With
```

上述代码假定订单表的主键列是"_Identity"，表格每增加一行，就将其 Primarykey 属性设置为对应 DataRow 的"_Identify"列的值。

完整示例

HttpRequest事件代码：

```
Select Case e.Path
    Case "list.htm"
        '获取要显示的页
        Dim page As Integer = 0 '默认page为0,显示第一页
        Dim pageRows As Integer = 15 '每页15行
        If e.GetValues.ContainsKey("page") Then  '如果地址中有page参数
            Integer.TryParse(e.GetValues("page"), page) '提取page参数
        End If
        Dim StartRow As Integer = page * pageRows + 1 '此页第一行
        Dim EndRow As Integer = (page + 1) * pageRows '此页最后一行
        '获取该页数据
        Dim cmd As New SQLCommand
        cmd.ConnectionName = "orders" '记得设置数据源名称
        cmd.CommandText = "Select Count(*) From {订单}"
        Dim Count As Integer = cmd.ExecuteScalar() '获取总的行数
        cmd.CommandText = "Select * From (Select Row_Number() Over(Order by 日期) As [NO.],[_Identify],产品,客户,数量,单价,日期 From 订单) As a "
        cmd.CommandText = cmd.CommandText & "  Where [NO.]>= " & StartRow & " And [NO.] <= " & EndRow
        Dim dt As DataTable = cmd.ExecuteReader
        '生成菜单
        Dim wb As New WeUI
        With wb.AddActionSheet("","menu") '设计菜单
            .Add("mnudAdd", "增加订单").Attribute="onclick=""alert('增加订单')"""
            .Add("mnuEdit", "编辑订单").Attribute ="onclick=""alert('编辑订单页码:'+table1.pagenumber+'主键:'+ table1.primarykey)"""
            .Add("mnuDelete", "删除订单").Attribute ="onclick=""alert('删除订单页码:'+table1.pagenumber+ '主键:'+ table1.primarykey)"""
            .Add("mnuCancel","取消","",True)
        End With
        '根据此页数据生成网页
```

```
    With wb.AddTable("","Table1")
      .PageNumber = page '设置页码
      .ActiveSheet = "menu"
       Dim nms() As String = {"NO.","产品","客户","数量","单价","日期"}
      .Head.AddRow(nms)
       For Each r As DataRow In dt.DataRows
        With .Body.AddRow(r("NO."),r("产品"),r("客户"),r("数量"))
          .Primarykey = r("_Identify") '设置主键
          .AddCell(Format(r("单价"),"#0.00"))
          .AddCell(Format(r("日期"),"MM月dd日"))
        End With
      Next
    End With
    With wb.AddButtonGroup("","btg2", False)
      If page > 0 Then
        .Add("btnPrev", "上一页","","List.htm?page=" & page - 1)
      End If
      If Endrow < count Then
        .Add("btnNext", "下一页","","List.htm?page=" & page + 1)
      End If
    End With
    e.WriteString(wb.Build)
End Select
```

同样Select语句必须将主键列包括进来。

现在你在客户端浏览器连续点击某行，会出现菜单，在点击菜单中的命令，会显示当前页码和当前行的主键。

下图是我在iPhone按着上述操作后截图：



7.55.20 菜单综合示例

本节的任务是分页浏览后台数据，连续点击某个单元格，能弹出一个菜单，通过这个菜单实现订单的编辑、增加和删除：



NO.	产品	客户	数量	单价	日期
81	PD02	CS03	210.00	19.00	2016-04-17
82	PD02	CS03	600.00	19.00	2016-04-16
83	PD02	CS04	50.00	19.00	2016-04-16
84	PD03	CS04	100.00	10.00	2016-04-16
85	PD01	CS02	600.00	18.00	2016-04-15
86	PD01	CS04	200.00	18.00	2016-04-15
87	PD02	CS03	200.00	19.00	2016-04-15
88	PD04	CS04	120.00	22.00	2016-04-15
89	PD01	CS04	60.00	18.00	2016-04-13
90	PD01	CS04	80.00	18.00	2016-04-13

增加、编辑或删除订单后，都能出现一个提示，例如下图是增加订单成功后的提示：





设计步骤：

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"table.js"，文件内容为：

```
function edit(){
    location="edit.htm?page=" + table1.pagenumber + "&key=" + table1.primarykey;
}
function del(){
    location="delete.htm?page=" + table1.pagenumber + "&key=" + table1.primarykey;
}
function addnew(){
    location="addnew.htm?page=" + table1.pagenumber;
}
```

上述 js 代码定义了三个函数，代码都很简单，将表格的 pagenumber(当前页码)和 primarykey(选定行主键)用 get 方式传递给服务器

假定当前页码是 3，主键为 176，那么执行 edit 函数，传递给服务器的链接地址就是：

http://127.0.0.1/edit.htm?page=3&key=176

服务器就知道：用户要编辑的订单的主键值是 176，编辑结束后，返回第 4 页。
提示：delete 是 JavaScript 的关键词，不能用作函数名。
2、为例避免 HttpRequest 事件代码过长，也为了更便于维护，我们将主要功能用自定义函数实现。本示例包括四个自定义函数，其中 List 函数用于数据的分页显示，以及菜单的生成和显示，另外三个函数为 AddNew、Delete 和 Edit，分别用于增加、删除和编辑订单。

函数的代码为：

函数名	代码

List	Dim e As RequestEventArgs = args(0)
	'获取要显示的页
	Dim page As Integer = 0 '默认page为0,显示第一页
	Dim pageRows As Integer = 10 '每页10行
	If e.GetValues.ContainsKey("page") Then '如果地址中有page参数
	Integer.TryParse(e.GetValues("page"), page) '提取page参数
	End If
	Dim StartRow As Integer = page * pageRows + 1 '此页第一行
	Dim EndRow As Integer = (page + 1) * pageRows '此页最后一行
	'获取该页数据
	Dim cmd As New SQLCommand
	cmd.ConnectionName = "orders" '记得设置数据源名称
	cmd.CommandText = "Select Count(*) From {订单}"
	Dim Count As Integer = cmd.ExecuteScalar() '获取总的行数
	cmd.CommandText = "Select * From (Select Row_Number() Over(Order by 日期 desc) As [NO.],[_Identify],产品,客户,数量,单价,日期 From 订单) As a "
	cmd.CommandText = cmd.CommandText & " Where [NO.]>= " & StartRow & " And [NO.] <= " & EndRow
	Dim dt As DataTable = cmd.ExecuteReader
	'生成菜单
	Dim wb As New WeUI
	With wb.AddActionSheet("","menu") '设计菜单
	.Add("mnudAdd", "增加订单").Attribute="onclick='addnew()'" '调用js函数
	.Add("mnuEdit", "编辑订单").Attribute ="onclick='edit()'"
	.Add("mnuDelete", "删除订单").Attribute ="onclick='del()'"
	.Add("mnuCancel","取消","",True)
	End With
	'根据此页数据生成网页
	With wb.AddTable("","Table1")
	.PageNumber = page '设置页码
	.Primarykey = "_Identify" '设置主键
	.ActiveSheet = "menu" '设置菜单
	.CreateFromDataTable(dt)
	End With
	With wb.AddButtonGroup("","btg2", False) '生成上一页和下一页按钮
	If page > 0 Then

```
        .Add("btnPrev", "上一页","","List.htm?page=" & page - 1)
    End If
    If Endrow < count Then
        .Add("btnNext", "下一页","","List.htm?page=" & page + 1)
    End If
End With
wb.AppendHTML("<script src='./lib/table.js'></script>") '引入脚本文件
e.WriteString(wb.Build)
```

AddNew

```
Dim e As RequestEventArgs = args(0)
Dim wb As New weui
If e.PostValues.Count = 0 Then '生成增加订单网页
    wb.AddForm("","form1","addnew.htm")
    With wb.AddInputGroup("form1","ipg1","增加订单")
        .AddInput("产品","产品","text")
        .AddInput("客户","客户","text")
        .AddInput("数量","数量","number")
        .AddInput("单价","单价","number").Step="0.01"
        .AddInput("日期","日期","date")
    End With
    With wb.AddButtonGroup("form1","btg1",True)
        .Add("btn1", "确定", "submit")
    End With
Else '保存新增的订单
    Dim dr As DataRow = DataTables("订单").AddNew()
    Dim nms() As String = {"产品","客户","数量","单价","日期"}
    For Each nm As String In nms
        dr(nm) = e.PostValues(nm)
    Next
    dr.Save()
    With  wb.AddMsgPage("","msgpage","增加成功",  "好好学习,天天向上")  '增加订单成功提示信息
        .AddButton("btn1","继续增加","addnew.htm")
        .AddButton("btn1","返回列表","list.htm")
    End With
End If
e.WriteString(wb.Build) '生成网页
```

Edit

```
Dim e As RequestEventArgs = args(0)
Dim wb As New weui
Dim PageNumber As Integer = e.GetValues("page")
Dim PrimaryKey As Integer = e.GetValues("key")
Dim PageURL = "List.htm?page=" & PageNumber
If e.PostValues.Count = 0 Then '生成编辑页面
    Dim dr As DataRow = DataTables("订单").SQLFind("[_Identify]=" & PrimaryKey)
```

```
    If dr IsNot Nothing Then
        wb.AddForm("","form1","edit.htm?key=" & PrimaryKey & "&page=" & PageNumber)
        With wb.AddInputGroup("form1","ipg1","编辑订单")
          .AddInput("产品","产品","text").Value = dr("产品")
          .AddInput("客户","客户","text").Value = dr("客户")
          .AddInput("数量","数量","number").Value = dr("数量")
          With .AddInput("单价","单价","number")
            .Step="0.01"
            .Value = dr("单价")
          End With
          .AddInput("日期","日期","date").Value = dr("日期")
        End With
        With wb.AddButtonGroup("form1","btg1",True)
          .Add("btn1", "确定", "submit")
        End With
    Else
        With  wb.AddMsgPage("","msgpage","编辑失败", "此订单可能已经被删除!") '提示用户此订单不存在.
          .icon= "Warn"
          .AddButton("btn1","返回",PageURL) '生成返回原来页面的按钮
        End With
    End If
Else '保存编辑结果
    Dim dr As DataRow = DataTables("订单").SQLFind("[_Identify]=" & PrimaryKey)
    If dr IsNot Nothing Then
        Dim nms() As String = {"产品","客户","数量","单价","日期"}
        For Each nm As String In nms
          dr(nm) = e.PostValues(nm)
        Next
        dr.Save()
        '显示完成提示,2妙手自动返回原来的页面
        wb.AppendHtml("<meta http-equiv='refresh' content='2; url=/" & PageURL & "'>",True)
        wb.AddToast("","t1", "编辑完成",0).Visible = True
    Else
        With  wb.AddMsgPage("","msgpage","保存失败", "此订单可能已经被删除!") '提示用户此订单不存在.
          .icon= "Warn"
          .AddButton("btn1","返回",PageURL) '生成返回原来页面的按钮
        End With
    End If
End If
e.WriteString(wb.Build) '生成网页
```

Delete	Dim e As RequestEventArgs = args(0) Dim wb As New weui Dim PageURL = "List.htm?page=" & e.GetValues("page") '合成当前页面链接 DataTables("订单").SQLDeleteFor("[_Identify] =" & e.GetValues("key")) '根据主键删除行 '显示删除完成提示,2秒后返回原来的页面 wb.AppendHtml("<meta http-equiv='refresh' content='2; url=/" & PageURL & "'>",True) wb.AddToast("","t1", "订单已删除",0).Visible= True e.WriteString(wb.Build)

3、将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "list.htm"
        Functions.Execute("List",e)
    Case "addnew.htm"
        Functions.Execute("AddNew",e)
    Case "edit.htm"
        Functions.Execute("Edit",e)
    Case "delete.htm"
        Functions.Execute("Delete",e)
End Select
```

7.55.21 改用按钮操作

本节任务和上一节相同，但全部改用按钮操作，并加上首页和末页按钮。

此外根据页面的位置，按钮颜色会改变，例如到了最后一页，下一页和末页按钮颜色变灰：



需要改动的地方有两处。

1、首先如果用户没有选择任何行，应该禁止编辑和删除订单，以免出错，所以将 table.js 的代码改为：

```
function edit(){
    if (table1.primarykey){
        location="edit.htm?page=" + table1.pagenumber + "&key=" + table1.primarykey;
    }
}
function del(){
    if(table1.primarykey){
        location="delete.htm?page=" + table1.pagenumber + "&key=" + table1.primarykey;
    }
```

```
}
function addnew(){
    location="addnew.htm?page=" + table1.pagenumber;
}
```

2、自定义函数 list 的代码改为：

```
Dim e As RequestEventArgs = args(0)
'获取要显示的页
Dim page As Integer = 0 '默认page为0,显示第一页
Dim pageRows As Integer = 10 '每页10行
If e.GetValues.ContainsKey("page") Then   '如果地址中有page参数
    Integer.TryParse(e.GetValues("page"), page) '提取page参数
End If
Dim StartRow As Integer = page * pageRows + 1 '此页第一行
Dim EndRow As Integer = (page + 1) * pageRows '此页最后一行
'获取该页数据
Dim cmd As New SQLCommand
cmd.ConnectionName = "orders" '记得设置数据源名称
cmd.CommandText = "Select Count(*) From {订单}"
Dim Count As Integer = cmd.ExecuteScalar() '获取总的行数
Dim Pages As Integer = Math.Ceiling(Count/PageRows) '计算出总页数
cmd.CommandText = "Select * From (Select Row_Number() Over(Order by  日期  desc) As [NO.],[_Identify],产品,客户,数量,单价,日期  From  订单) As a "
cmd.CommandText = cmd.CommandText & "   Where [NO.]>= " & StartRow & " And [NO.] <= " & EndRow
Dim dt As DataTable = cmd.ExecuteReader
Dim wb As New WeUI
'根据此页数据生成网页
With wb.AddTable("","Table1")
    .PageNumber = page '设置页码
    .Primarykey = "_Identify" '设置主键
    .CreateFromDataTable(dt)
End With
With wb.AddButtonGroup("","btg1", False) '生成换页按钮
    If page > 0 Then
        .Add("btnFirst", "首页","","List.htm?page=0")
        .Add("btnPrev", "上一页","","List.htm?page=" & page - 1)
    Else
        .Add("btnFirst", "首页").Kind = 1
        .Add("btnPrev", "上一页").Kind = 1
    End If
    If Endrow < count Then
        .Add("btnNext", "下一页","","List.htm?page=" & page + 1)
        .Add("btnLast", "末页","","List.htm?page=" & pages - 1)
    Else
      .Add("btnNext", "下一页").Kind = 1
```

```
        .Add("btnNext", "末页").Kind = 1
    End If
End With
With wb.AddButtonGroup("","btg2", False) '生成操作按钮
    .Add("btnAdd", "增加订单").Attribute = "onclick='addnew()'"
    .Add("btnEdit", "编辑订单").Attribute = "onclick='edit()'"
    .Add("btnDelete", "删除订单").Attribute = "onclick='del()'"
End With
wb.AppendHTML("<script src='./lib/table.js'></script>") '引入脚本文件
e.WriteString(wb.Build)
```

7.55.22 显示动态菜单

一般用户可以忽略本节内容。

本节的任务是设计一个动态菜单，如果当前行部门列的内容为"技术部"时，隐藏“菜单项目 1”：





设计步骤：

1、在"d:\web"目录下，建立一个子目录 lib，在这个目录建立一个文本文件，文件名为"activesheet.js"，文件内容为：

```
function dynaActiveSheet(){
  if(table1.rows[table1.rowSel].cells[0].innerHTML=="技术部"){
    hide("menu1");
  }
  else{
    show("menu1");
  }
}
```

提示：rowSel(注意区分大小写)属性是 Foxtable 为 WeUI 扩展的一个表格属性，用于返回选定行的位置，对应还有一个 colSel 属性，用于返回选定列的位置。

2、HttpRequest 事件代码：

```
Select Case e.Path
```

```
    Case "table.htm"
        Dim wb As New WeUI
        With wb.AddTable("","table1")
            .AfterSelChange  =  "dynaActiveSheet()"  '置顶选择不同单元格后要执行的js行数,注意区分大小写
            .head.AddRow("部门","姓名","年龄","电话","地址") '表头
            .body.AddRow("技术部","张三","36","110","中国北京") '数据
            .body.AddRow("技术部","李四","38","110","中国上海")
            .body.AddRow("技术部","王五","39","110","中国深圳")
            .body.AddRow("生产部","赵六","39","110","中国深圳")
            .body.AddRow("生产部","刘七","39","110","中国深圳")
            .ActiveSheet = "menu" '指定菜单
        End With
        With wb.AddActionSheet("","menu") '设计菜单
            .Add("menu1", "菜单项目1")
            .Add("menu2", "菜单项目2")
            .Add("menu3", "菜单项目3")
            .Add("menu4","取消","",True)
        End With
        wb.AppendHTML("<script src='./lib/activesheet.js'></script>") '引入脚本文件
        e.WriteString(wb.Build)
End Select
```

提示： AfterSelChange 属性用于指定在表格中选择不同单元格后，要执行的 js 事件代码，这里是要执行我们在第一步定义好的 js 函数，注意 js 函数是区分大小写的，而且括号不能省略。

7.55.23 又一个综合示例

本节是一个综合性比较强的例子，包括三个页面。
list.htm 用于分页显示数据，有四个页面切换按钮：



NO.	产品	客户	数量	单价	日期
1	PD05	CS03	650.00	17.00	2016-01-02
2	PD02	CS01	400.00	20.00	2016-01-03
3	PD05	CS04	320.00	17.00	2016-01-03
4	PD01	CS03	80.00	18.00	2016-01-04
5	PD03	CS03	490.00	10.00	2016-01-04
6	PD04	CS04	240.00	17.60	2016-01-07
7	PD05	CS03	160.00	17.00	2016-01-07
8	PD01	CS04	200.00	14.40	2016-01-08
9	PD03	CS02	200.00	8.00	2016-01-08
10	PD01	CS02	800.00	18.00	2016-01-10

连续两次点击同一单元格，会出现一个菜单，且在筛选和非筛选的状态下，显示的菜单是不同的：

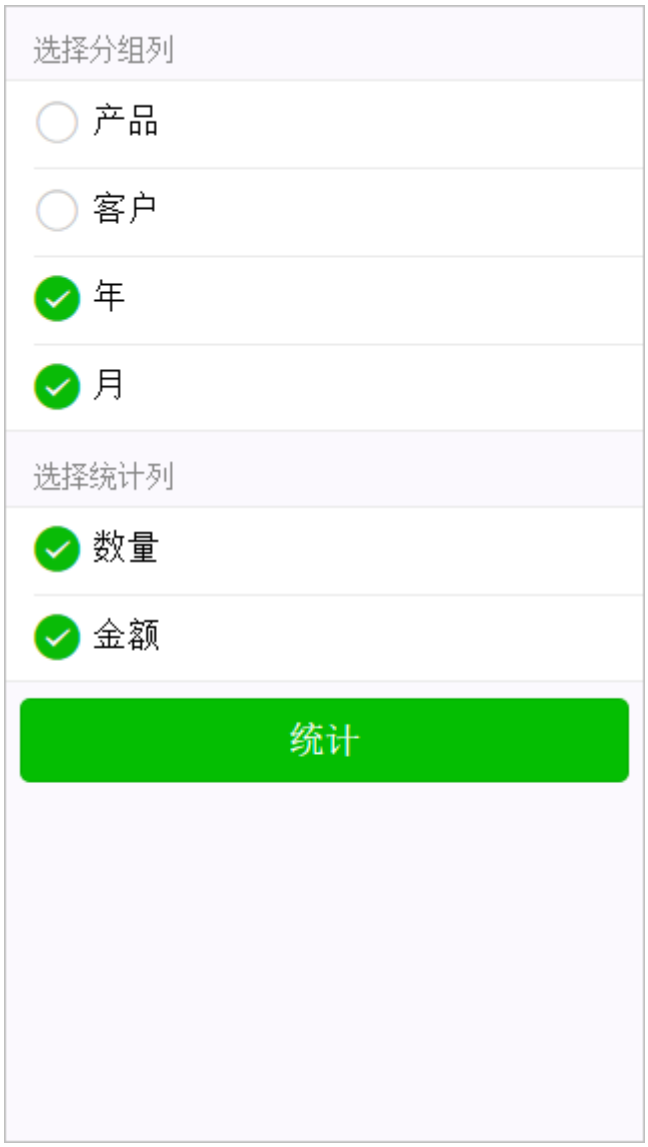

NO.
产品
客户
数量
单价
日期
1 PD05 CS03 650.00 17.00 2016-01-02
2 PD02 CS01 400.00 20.00 2016-01-03
3 PD05 CS04 320.00 17.00 2016-01-03
4 PD01 CS03 80.00 18.00 2016-01-04
5 PD03 CS03 490.00 10.00 2016-01-04
6 PD04 CS04 240.00 17.60 2016-01-07
7 PD05 CS03 160.00 17.00 2016-01-07
8 PD01 CS04 200.00 14.40 2016-01-08
9 PD03 CS02 200.00 8.00 2016-01-08
10 PD01 CS02 800.00 18.00 2016-01-10
第一页
上一页
下一页
最末页
数据筛选
数据统计
取消



filter.htm 用于输入筛选条件：



tongji.htm 用于统计数据（包括统计结果的显示），如果已经进行了筛选，则仅统计筛选后的数据：


选择分组列
产品
客户
年
月
选择统计列
数量
金额
统计

年	月	数量	金额
2016	1	7560.00	114878.50
2016	2	8990.00	156231.50
2016	3	8420.00	141334.00
2016	4	10910.00	187344.00
2016	5	9080.00	160196.50
2016	6	4950.00	92011.00



为避免 HttpRequest 事件代码过程，本节全部采用自定义函数实现。

1、HttpRequest 事件代码：

```
Select Case e.Path
    Case "filter.htm"
        Functions.Execute("Filter",e)
    Case "list.htm"
        Functions.Execute("List",e)
    Case "tongji.htm"
        Functions.Execute("Statistics",e)
End Select
```

2、自定义函数 Filter 用于生成筛选页面 filter.htm：

```
Dim e As RequestEventArgs = args(0)
Dim wb As New WeUI
wb.AddForm("","form1","list.htm")
With wb.AddInputGroup("form1","ipg1","数据筛选")
```

```
    .AddSelect("product","产品","|PD01|PD02|PD03|PD04|PD05")
    .AddSelect("customer","客户","|CS01|CS02|CS03|CS04|CS05")
    .AddInput("startdate","开始日期","date")
    .AddInput("enddate","结束时间","date")
End With
With wb.AddButtonGroup("form1","btg1",True)
    .Add("btn1", "确定", "submit")
End With
e.WriteString(wb.Build)
```

3、自定义函数GetFilter用于根据生成filter.htm提交的内容合成条件表达式：

```
Dim e As RequestEventArgs = args(0)
Dim wb As WeUI = Args(1)
Dim flt As String
If e.PostValues.ContainsKey("product") Then
    flt = "产品  = '" & e.PostValues("product") & "'" '合成条件
    wb.AppendCookie("product", e.PostValues("product"),30) '将值写入cookie中
Else
    wb.DeleteCookie("product") '删除cookie
End If
If e.PostValues.ContainsKey("customer") Then
    If flt > "" Then
        flt = flt & " and "
    End If
    flt = flt & "客户  = '" & e.PostValues("customer") & "'" '合成条件
    wb.AppendCookie("customer", e.PostValues("customer")) '将值写入cookie中
Else
    wb.DeleteCookie("customer") '删除cookie
End If
If e.PostValues.ContainsKey("startdate") Then
    If flt > "" Then
        flt = flt & " and "
    End If
    flt = flt & "日期  >= '" & e.PostValues("startdate") & "'"
    wb.AppendCookie("startdate", e.PostValues("startdate"))
Else
    wb.DeleteCookie("startdate")
End If
If e.PostValues.ContainsKey("enddate") Then
    If flt > "" Then
        flt = flt & " and "
    End If
    flt = flt & "日期  <= '" & e.PostValues("enddate") & "'"
    wb.AppendCookie("enddate", e.PostValues("enddate"))
```

```
Else
    wb.DeleteCookie("enddate")
End If
Return flt
```

提示：GetFilter函数需要输入的条件保存到Cookie中，而Coolie的保存和删除是通过WeUI完成的，所以在调用GetFilter的时候，需要将WeUI作为第二个参数传递过去。

4、自定义函数GetCookieFilter用于根据Cookie合成条件表达式：

```
Dim flt As String
Dim e As RequestEventArgs = args(0)
If e.Cookies.ContainsKey("product")  Then
    flt = "产品  = '" & e.Cookies("product") & "'"
End If
If e.Cookies.ContainsKey("customer")  Then
    If flt > "" Then
        flt = flt & " and "
    End If
    flt = flt & "客户  = '" & e.Cookies("customer") & "'"
End If
If e.Cookies.ContainsKey("startdate")  Then
    If flt > "" Then
        flt = flt & " and "
    End If
    flt = flt & "日期  >= '" & e.Cookies("startdate") & "'"
End If
If e.Cookies.ContainsKey("enddate")  Then
    If flt > "" Then
        flt = flt & " and "
    End If
    flt = flt & "日期  <= '" & e.Cookies("enddate") & "'"
End If
Return flt
```

5、List函数用于生成list.htm：

```
Dim e As RequestEventArgs = args(0)
Dim wb As New WeUI
Dim flt As String
If e.GetValues.ContainsKey("unfilter") Then '如果是取消筛选
    wb.ClearCookie() '清除Cookie
ElseIf e.PostValues.Count > 0
    flt = Functions.Execute("GetFilter",e, wb) '根据输入内容合成条件，注意WeUI也需要传递过去
Else
```

```
    flt = Functions.Execute("GetCookieFilter",e) '根据Cookie合成条件
End If
'获取要显示的页码
Dim page As Integer = 0 '默认page为0,显示第一页
Dim pageRows As Integer = 10 '每页10行
If e.GetValues.ContainsKey("page") Then   '如果地址中有page参数
    Integer.TryParse(e.GetValues("page"), page) '提取page参数
End If
Dim StartRow As Integer = page * pageRows + 1 '此页第一行
Dim EndRow As Integer = (page + 1) * pageRows '此页最后一行
'提取此页数据
Dim cmd As New SQLCommand
cmd.ConnectionName = "orders" '记得设置数据源名称
cmd.CommandText = "Select Count(*) From {订单}"
If flt > "" Then
    cmd.CommandText = cmd.CommandText & " where " & flt
End If
Dim Count As Integer = cmd.ExecuteScalar() '获取总的行数
Dim Pages As Integer = Math.Ceiling(Count/PageRows) '计算出总页数
cmd.CommandText = "Select * From (Select Row_Number() Over(Order by 日期) As [NO.], 产品,客户,数量,单价,
日期 From 订单 "
If flt > "" Then
    cmd.CommandText = cmd.CommandText & " where " & flt
End If
cmd.CommandText = cmd.CommandText & ") As a "
cmd.CommandText = cmd.CommandText & "   Where [NO.]>= " & StartRow & " And [NO.] <= " & EndRow
'合成网页
With wb.AddTable("","Table1")
    .ActiveSheet = "menu" '指定菜单
    .CreateFromDataTable(cmd.ExecuteReader)
End With
With wb.AddButtonGroup("","btg2", False)
    If page > 0 Then
        .Add("btnFirst", "第一页","","List.htm?page=0")
        .Add("btnPrev", "上一页","","List.htm?page=" & page - 1)
    Else
        .Add("btnFirst", "第一页","button").Kind = 1
        .Add("btnPrev", "上一页","button").Kind = 1
    End If
    If Endrow < count Then
        .Add("btnNext", "下一页","","List.htm?page=" & page + 1)
        .Add("btnLast", "最末页","","List.htm?page=" & pages - 1)
    Else
        .Add("btnNext", "下一页","button").Kind = 1
```

```
        .Add("btnLast", "最末页","button").Kind = 1
    End If
End With
With wb.AddActionSheet("","menu") '设计菜单
    If flt = "" Then
        .Add("mnuFilter", "数据筛选","filter.htm")
    Else
        .Add("mnuUnFilter", "取消筛选","list.htm?unfilter=true")
    End If
    .Add("mnuStatistics", "数据统计","tongji.htm")
    .Add("mnuCancel","取消","",True)
End With
e.WriteString(wb.Build)
```

6、自定函数Statistics用于生成页面tongji.htm:

```
Dim e As RequestEventArgs = args(0)
Dim wb As New WeUI
If e.PostValues.Count = 0 Then '分组统计设置
    wb.AddForm("","form1","tongji.htm")
    With wb.AddCheckGroup("form1","rdg1","选择分组列")
        .Add("fz1","产品",)
        .Add("fz2","客户")
        .Add("fz3","年")
        .Add("fz4","月")
    End With
    With wb.AddCheckGroup("form1","rdg2","选择统计列")
        .Add("tj1","数量")
        .Add("tj2","金额")
    End With
    With wb.AddButtonGroup("form1","btg1",True)
        .Add("btn1", "统计", "submit")
    End With
Else '显示统计结果
    Dim gp As new SQLGroupTableBuilder("统计表1","订单")
    gp.ConnectionName = "orders"
    gp.Filter = Functions.Execute("GetCookieFilter",e) '根据Cookie合成条件
    If e.PostValues.ContainsKey("fz1") Then
        gp.Groups.AddDef("产品")
    End If
    If e.PostValues.ContainsKey("fz2") Then
        gp.Groups.AddDef("客户")
    End If
    If e.PostValues.ContainsKey("fz3") Then
        gp.Groups.AddDef("日期",DateGroupEnum.Year,"年")
```

```
    End If
    If e.PostValues.ContainsKey("fz4") Then
      gp.Groups.AddDef("日期",DateGroupEnum.Month,"月")
    End If
    If e.PostValues.ContainsKey("tj1") Then
      gp.Totals.AddDef("数量")
    End If
    If e.PostValues.ContainsKey("tj2") Then
      gp.Totals.AddExp("金额","数量 * 单价")
    End If
    If gp.Groups.Count = 0 OrElse gp.Totals.Count = 0 Then
      wb.InsertHTML("请选择分组列和统计列!")
    Else
      With wb.AddTable("","Table1")
        .CreateFromDataTable(gp.Build(True))
      End With
      With wb.AddButtonGroup("","btg1", False)  '水平排列
        .Add("btn6", "重新统计","button","tongji.htm")
        .Add("btn7", "返回列表","button","list.htm?page=0")
      End With
    End If
End If
e.WriteString(wb.Build)
```

7.56 使用关联表

7.56.1 后台表结构

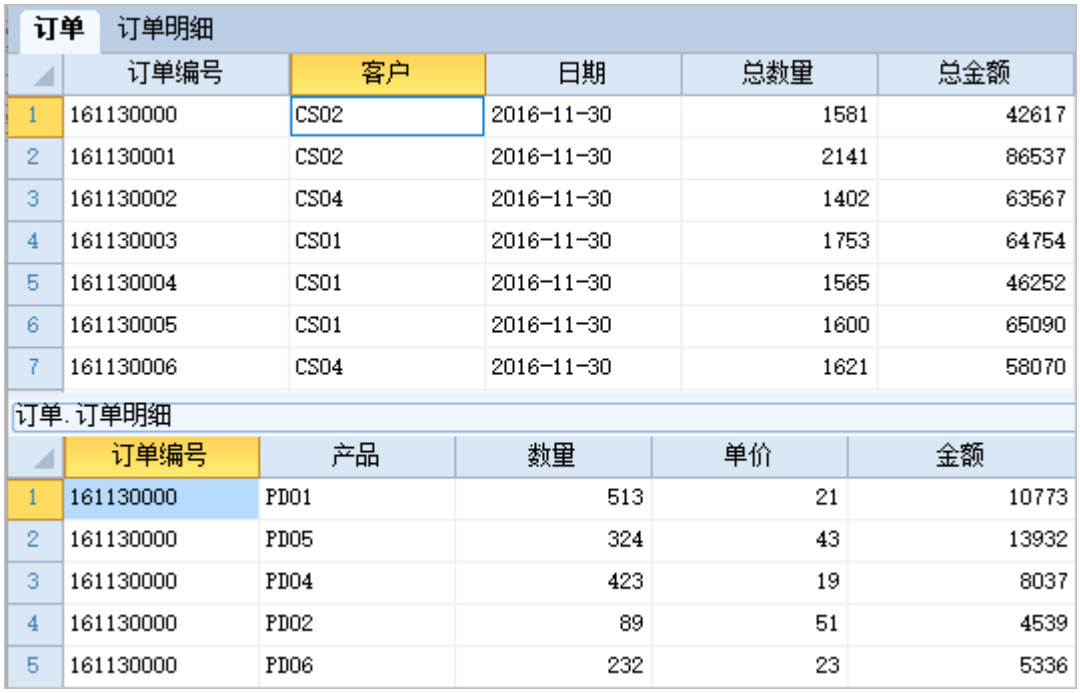
通过网页显示和编辑关联表，对于专业程序员也是一个不小的挑战，当然由于这是 Foxtable，技术实现上是不会有多少难度的，只是在设计逻辑的理解上，一些用户可能会有困惑。

我们会提供一个完整的示例，不仅是为了方便大家理解，更重要的是，即使你暂时理解不了，直接在这个例子的基础上改改表名和列名，就能套用在自己的系统中。

接下来你会看到，整个任务的完成，基本不要求你懂网页设计，因为绝大多数任务都是直接通过 Foxtable 完成的。
而且代码数量很少，从客户端到服务端，从分页浏览后台数据，到从订单的增加、删除和修改，再到订单明细的增加、删除和修改，以及多个菜单和按钮的生成，完成所有这些任务，不过 200 行左右的代码而已。

我们首先介绍一下后台的表结构。

后台数据库有订单和订单明细两个表，二者通过订单编号建立关联，结构如下图所示：



订单 订单明细

	订单编号	客户	日期	总数量	总金额
1	161130000	CS02	2016-11-30	1581	42617
2	161130001	CS02	2016-11-30	2141	86537
3	161130002	CS04	2016-11-30	1402	63567
4	161130003	CS01	2016-11-30	1753	64754
5	161130004	CS01	2016-11-30	1565	46252
6	161130005	CS01	2016-11-30	1600	65090
7	161130006	CS04	2016-11-30	1621	58070

订单.订单明细

	订单编号	产品	数量	单价	金额
1	161130000	PD01	513	21	10773
2	161130000	PD05	324	43	13932
3	161130000	PD04	423	19	8037
4	161130000	PD02	89	51	4539
5	161130000	PD06	232	23	5336

订单明细表的金额，以及订单表的总数量和总金额，都是表达式列，后台数据库不存在这三列。

希望设计一个手机网页管理系统，该系统有要完成三个任务：

1、分页浏览订单

2、订单的增加、修改和删除

3、订单明细的增加、修改和删除。

软件开发没有定式

关联表的显示和编辑是一个综合性的例子，我会按照比较容易理解，且相对比较高效合理的方式来给大家编写这个例子。

大家掌握好这个例子之后，基本上就可以开发一些比较复杂的手机管理系统了。

但是软件开发是没有定式的，完成同样的任务，会有很多种方法，我的例子是针对初学者而言的，希望大家不要被例子局限了思路。

7.56.2 如何传递命令

网页是无法直接操作后台数据的，我们需要利用 get 方式，按照事先约定的方式，将各种参数传递给服务器，服务器根据收到的参数，来执行相应的操作。

例如要编辑一个订单，客户端肯定要将订单编号（或主键）传递给服务器，服务器才知道用户要编辑的是那一个订单，还需要将当前页码传递给服务器，这样编程完成之后，客户端才能正确返回原来的页面。

所以要编辑一个订单，客户端通常需要给服务器发送以下访问请求：

edit.htm?page=2&oid=161130078

page 为页码，oid 为订单编号，完整的意思是：

我目前位于第 3 页(页码从 0 开始)，我要编辑订单编号为 161130078 的订单，编辑完成之后，请让我返回第 3 页吧。

事先约定好每个页面的 get 参数的名称和含义，以便于通过 get 方式正确传递数据给服务器，是能否完成设计的关键，否则很容易将自己绕晕，最后难以为继。
在 Foxtable 开发网页管理系统，基本上就是两部分：

1、客户端发送包含 get 参数的访问请求给服务器。
2、服务端收到访问请求，从中提取 get 参数，执行相应的操作后，生成新的页面，发回给客户端。
所以接下来我要先理清楚每个页面要完成的任务，以及对应的 get 参数要求。

7.56.3 list.htm 的设计思路

首先我们需要一个页面来分页浏览订单，这个网页我们命名为 list.htm。

为节省显示空间，list.htm 只显示三个常用的按钮，分别是增加订单、上一页和下一页：

订单编号	客户	日期	数量	金额
161130093	CS02	2016-11-30	1445	48060.00
161130092	CS05	2016-11-30	1615	44060.00
161130091	CS04	2016-11-30	1721	51899.00
161130090	CS01	2016-11-30	2048	55735.00
161130089	CS03	2016-11-30	2078	108751.00
161130088	CS01	2016-11-30	1477	39688.00
161130087	CS03	2016-11-30	840	23852.00
161130086	CS02	2016-11-30	1680	63798.00
161130085	CS02	2016-11-30	1376	50836.00
161130084	CS04	2016-11-30	1325	45198.00

增加订单 上一页 下一页

连续点击某个单元格，可以显示一个上拉菜单，通过这个菜单可以进行跟多的操作，例如编辑或删除当前订单：



list.htm 的 get 参数说明：

- **page 参数**

  page 参数用于指定要显示的页面，例如：

  list.htm?page=2

  表示要显示第 3 页(页码从 0 开始编号)，如果省略 page 参数，则显示第一页。

- **deloid 参数**

deloid 参数用于指定要删除订单的订单编号，例如：

list.htm?page=2&deloid=161130078

表示要先删除订单编号为 161130078 的订单，然后显示第 3 页。

7.56.4 edit.htm 的设计思路

通常我们还需要两个页面，分别用于增加订单和编辑订单。
我们可以按照之前的示例一样，用两个页面（addnew.htm 和 edit.htm）来完成这两项任务，简单也容易理解。

但实际上，不管是增加订单，还是编辑订单，其实都是一回事，前者是先增加一个订单再编辑这个订单，后者是直接编辑现有的订单
所以我们可以将二者合并为一个页面，用不同的 get 参数，来完成不同的任务，我们将这个页面命名为：edit.htm。

将新增和编辑合并在一个页面处理，编码会变得复杂一点，但是需要维护的页面和函数会减少一半，总的代码数量也会减少很多，还是值得的。
为便于大家理解，我们会对有关函数逐行讲述，但是对于代码逻辑能力不强的用户，我还是建议分开处理。

由于手机屏幕显示面积不大，而且 WeUI 的表格暂时没有数据编辑功能，所以订单明细的录入我们也采用输入框完成，例如：



为节省空间，edit.htm 默认只显示三个按钮，连续点击订单明细表格中同一单元格，会出现一个上拉菜单：



edit.htm 的 get 参数说明

通过 edit.htm，我们可以：

1、编辑订单
2、新增订单
3、编辑订单明细
4、新增订单明细
5、删除订单明细

不同的任务用户不同的 get 参数来表示，我们首先约定所有 get 参数的名称和含义：

- **page 参数**

  page 参数表示用户在新增或编辑订单之前，在 list.htm 中正在浏览第几页数据。

- **oid**

  oid 参数用于指定要编辑的订单的订单编号。

- **addnext 参数**

  addnext 是一个逻辑参数，用于表明是否需要新增一个订单明细。

- **did**

  did 参数用于指定要编辑的订单明细的主键。

- **deldid**

  deldid 参数用于指定要删除的订单明细的主键

下面我们分 5 种情况来介绍 edit.htm 的 get 参数的使用：

编辑订单

如果要编辑一个订单，需要向服务器发送访问请求：

edit.htm?page=2&oid=161203005

意思是我目前正在浏览第 3 页数据，现在要编辑订单编号为 161203005 的订单。

收到这个访问请求后，服务器返回的页面如下：



新增订单

如果要新增订单，需要向服务器发送访问请求：

edit.htm?page=2

服务器收到这个访问请求之后，知道用户正在浏览第 3 页，由于没有发现 oid 参数，所以知道用户要新增一个订单。

如是服务器新增一个订单，并自动生成订单编号。

另外，既然是新增订单，肯定也需要新增订单明细，所以服务器最后发送给用户的页面为：



新增订单明细

在 edit.htm 中，单击"增加明细"按钮，发送给服务器的访问请求为：

edit.htm?page=2&oid=161203005&addnext=true

服务器收到这个访问请求后，从 get 参数分析：

用户正在编辑编号为 161203005 的订单，之前正在 list.htm 浏览第 3 页数据，现在要新增一个订单明细。

如是服务器发送以下页面给用户：



编辑订单明细

在 edit.htm 中的订单明细表格中，连续点击同一单元格，从弹出的菜单中执行"编辑明细"，发送给服务器的访问请求为：

edit.htm?page=2&oid=161203005&did=567

服务器收到这个访问请求后，从 get 参数分析：

用户正在编辑编号为 161203005 的订单，之前正在 list.htm 浏览第 3 页数据，现在要编辑主键为 567 的订单明细。

如是服务器发送以下页面给用户：



删除订单明细

在 edit.htm 中的订单明细表格中，连续点击同一单元格，从弹出的菜单中执行"删除明细"，发送给服务器的访问请求为：

edit.htm?page=2&oid=161203005&deldid=568

服务器收到这个访问请求后，从 get 参数分析：

用户正在编辑编号为 161203005 的订单，之前正在 list.htm 浏览第 3 页数据，现在要删除主键为 568

的订单明细。
如果服务器先删除主键为 568 的订单明细，然后将以下页面发送给用户，新的页面不再包括被删除的订单明细：



现在设计思路已经清晰，我们可以开始客户端和服务端的编码工作了。

7.56.5 准备 JS 文件

我们的代码分客户端和服务端两部分。
服务端就是 Foxtable 的 HttpRequest 事件。
客户端就是简单的 JavaScript 代码，用于按照我们之前约定的 get 参数规则，向服务器发送反问请求，非常简单。

在目录"d:\web\lib"新建一个文本文件"order.js"，内容为：

```
function edit(){
   location="edit.htm?page=" + table1.pagenumber + "&oid=" + table1.primarykey;
}

function del(){
   location="list.htm?page=" + table1.pagenumber + "&deloid=" + table1.primarykey;
}

function addnew(){
   location="edit.htm?page=" + table1.pagenumber;
}

function addDetail(){
   location=form1.action + "&addnext=true";
}

function editDetail(){
   location = form1.action + "&did=" + detailtable.primarykey;
}

function delDetail() {
   location = form1.action + "&deldid=" + detailtable.primarykey;
}

function calc(){
    document.getElementById("金额").value = document.getElementById("数量").value * document.getElementById("单价").value;
}
```

上代码中的 table1 为 list.htm 中的订单表格，form1 为 edit.htm 中的输入表单，detailtable 为 edit.htm 中的订单明细表格。

除了最后一个 calc 函数用于输入过程自动计算金额外，其他函数都是根据我们上一节 理顺 的 get 参数规则，根据要完成的任务向服务器发送访问请求，只是简单的字符串合并而已，都很好理解。

一个例子

以编辑订单为例，当我们在 list.htm 的菜单中执行“编辑订单”，会执行 edit 函数，该函数的代码为：

```
function edit(){
   location="edit.htm?page=" + table1.pagenumber + "&oid=" + table1.primarykey;
}
```

假定我们正在访问第 3 页，选定订单的订单编号为 161130078，此时 table1.pagenumber 等于 2，table1.primarykey 等于 161130078，以上代码合成的链接为：

edit.htm?page=2&oid=161130078

等于告诉服务器：我正在浏览第 3 页，我现在要编辑订单编号为 161130078 的订单。
后面在介绍 Foxtable 端的代码时，还会对以上 js 函数逐个讲述。

你有疑问？

我估计少数用户会发出这样的疑问，JS 代码文件明明存放在服务端，怎么就是客户端代码呢？
其实在运行的时候，JS 代码文件会先被下载到客户端，然后再执行，用户可以通过浏览器查看 JS 文件的内容，这一点和普通的网页或图片没有不同，都是由客户端浏览器负责显示或执行的。
HttpRequest 才是真正的服务端代码，直接在服务端运行，用户是看不到代码的。
所以千万不要在 JS 代码中出现敏感信息哦。

7.56.6 HttpRequest 事件代码

我计划所有任务都用自定义函数来完成，所以 HttpRequest 事件代码很简单：

```
Select Case e.Path
    Case "list.htm"
        Functions.Execute("List",e) '分页显示
    Case "edit.htm"
        If e.PostValues.Count > 0 Then
            Functions.Execute("Save",e) '保存表单数据
        End If
        Functions.Execute("Edit",e) '生成订单编辑页面
End Select
```

由于edit.htm承担了太多的任务，为避免单个函数代码过长，我们用两个函数处理edit.htm，Save函数负责保存表单数据，Eidt函数负责生成网页。

接下来我们分别介绍这3个自定义函数，实际上有4个自定义函数，因为还有一个用于生成订单编号的函数。

7.56.7 List 函数

当用户访问list.htm时，服务器会将访问请求转给自定义函数list处理。

list函数用于分页显示数据，这里假定使用的是SQL Server数据源，其代码为：

```
Dim e As RequestEventArgs = args(0)
'订单删除代码
If e.GetValues.ContainsKey("deloid") Then '如果提交了deloid参数,则删除对应的订单.
        DataTables("订单").SQLDeleteFor("订单编号='" & e.GetValues("deloid") & "'")
        DataTables("订单明细").SQLDeleteFor("订单编号='" & e.GetValues("deloid") & "'")
End If
```

```
'获取要显示的页
Dim page As Integer = 0 '默认page为0,显示第一页
Dim pageRows As Integer = 10 '每页10行
If e.GetValues.ContainsKey("page") Then   '如果地址中有page参数
        Integer.TryParse(e.GetValues("page"), page) '提取page参数
End If
Dim StartRow As Integer = page * pageRows + 1 '此页第一行
Dim EndRow As Integer = (page + 1) * pageRows '此页最后一行
'获取该页数据
Dim cmd As New SQLCommand
cmd.ConnectionName = "orders" '记得设置数据源名称
cmd.CommandText = "Select Count(*) From {订单}"
Dim Count As Integer = cmd.ExecuteScalar() '获取总的行数
Dim Pages As Integer = Math.Ceiling(Count/PageRows) '计算出总页数
cmd.CommandText = "Select * From (Select   Row_Number() Over(Order by 订单.订单编号 desc ) As RowNum,订单.订单编号,日期,客户,Sum(数量) As 数量,sum(数量*单价) As 金额"
cmd.CommandText = cmd.CommandText & " From 订单 Left JOIN 订单明细 ON 订单明细.订单编号 = 订单.订单编号 Group By {订单}.订单编号,日期,客户) As a "
cmd.CommandText = cmd.CommandText &  "   Where RowNum >= " & StartRow & " And RowNum <= " & EndRow
Dim dt As DataTable = cmd.ExecuteReader
'根据此页数据生成表格
Dim wb As New WeUI
With wb.AddTable("","Table1")
        .PageNumber = page '设置页码
        .ActiveSheet = "menu" '指定菜单
        .Primarykey = "订单编号" '指定主键,只要是能唯一区分行的列即可,并非一定要表的实际主键.
        .CreateFromDataTable(dt, False,"","","订单编号","客户","日期","数量","金额")
End With
'设计菜单
With wb.AddActionSheet("","menu")
        .Add("mnudAdd", "增加订单").Attribute="onclick='addnew()'" '调用js函数
        .Add("mnuEdit", "编辑订单").Attribute ="onclick='edit()'"
        .Add("mnuDelete", "删除订单").Attribute ="onclick=""show('dlg1')"""
        .Add("mnuFirst","第一页","List.htm?page=0",True)
        .Add("mnuLast","最末页","List.htm?page=" & pages - 1)
        .Add("mnuCancel","取消","",True)
End With
With wb.AddDialog("","dlg1", "删除确认","您确定要删除当前订单吗?")
        .AddButton("btnCancel","取消").Kind = 1
        .AddButton("btnOK","确定").Attribute = "onclick='del()'"
End With
'生成换页按钮
```

```
With wb.AddButtonGroup("","btg1", False)
        .Add("btnAdd", "增加订单").Attribute = "onclick='addnew()'"
        If page > 0 Then
                .Add("btnPrev", "上一页","","List.htm?page=" & page - 1)
        Else
                .Add("btnPrev", "上一页").Kind = 1
        End If
        If Endrow < count Then
                .Add("btnNext", "下一页","","List.htm?page=" & page + 1)
        Else
                .Add("btnNext", "下一页").Kind = 1
        End If
End With
wb.AppendHTML("<script src='./lib/order.js'></script>") '引入脚本文件
e.WriteString(wb.Build)
```

```
2
5
3
5
4
5
5
5
6
5
7
5
8
5
9
6
0
6
1
```

代码解析

list.htm一般带有一个page参数，例如：
list.htm?page=2
表示要显示第3页数据。

如果要删除订单，还需要带有deloid参数，例如：
list.htm?page=2&deloid=161130078
表示要删除编号为161130078的订单，然后显示第3页数据。

第3行代码判断GetValues是否由deloid参数。
如果有，则在第4行代码删除对应的订单，在第5行代码删除对应的订单明细。

第8行代码定义了一个变量page，这个变量表示要显示的页面，默认为0.，也就是显示第1页。
第10行到第12行代码，判断GetValues集合是否包括page参数，如果有的话， 将其转换为整数保存在变量page中。
第13行和第14行代码，分别计算出此页第一行数据的顺序号，和最后一行数据的顺序号。
第19行代码统计出订单表的总行数，第20行代码据此算出总页数。

第21行到第24行代码，利用SQL语句提取出这一页数据，保存在DataTable型变量dt中。
需要注意几点：
1、顺序号根据订单编号降序生成，保证新增订单显示在第一页。
2、后台的订单表其实没有数量和金额两列，所以要在select语句中用聚合函数计算出每个订单的数量和金额。

3、订单和订单明细表用的是Left Join方式联结，确保没有录入订单明细的订单，也能出现在查询结果中。

第27行到第32行代码根据这页数据生成表格。
第28行代码设置了表的页码。
第29行代码设置了表格的上拉菜单名称。
第30行代码指定了主键列名称，这个名称不一定是订单表真正的主键，只要是能唯一区分每个订单的列即可。

第34行到第41行生成了上拉菜单，菜单名称必须和第29行代码使用的名称保持一致。
第42行到第45行代码生成了一个对话框，用于删除订单。
第47到第59行代码生成了操作按钮。
除了上一页和下一页命令，其他命令都是调用js函数。
这些js函数定义在“d:\web\lib\order.js”文件中，我们之前已经对此进行了介绍。
第60行代码引用了这个js函数文件。
综上所述，list.htm 默认只显示三个常用的按钮，分别是增加订单、上一页和下一页：

订单编号	客户	日期	数量	金额
161130093	CS02	2016-11-30	1445	48060.00
161130092	CS05	2016-11-30	1615	44060.00
161130091	CS04	2016-11-30	1721	51899.00
161130090	CS01	2016-11-30	2048	55735.00
161130089	CS03	2016-11-30	2078	108751.00
161130088	CS01	2016-11-30	1477	39688.00
161130087	CS03	2016-11-30	840	23852.00
161130086	CS02	2016-11-30	1680	63798.00
161130085	CS02	2016-11-30	1376	50836.00
161130084	CS04	2016-11-30	1325	45198.00

增加订单　上一页　下一页

连续点击某个单元格，可以显示一个上拉菜单，通过这个菜单可以进行跟多的操作，例如编辑或删除当前订单：



现在我们结合具体代码，分别看看 list.htm 是如何实现新增订单、编辑订单和删除订单的：

删除订单

当我们在菜单中点击"删除订单"时，会显示一个对话框（第37行代码）：



如果在对话框中单击确定按钮，会执行js函数del(第44行代码)。

del函数的代码为：

```
function del(){
    location="list.htm?page=" + table1.pagenumber + "&deloid=" + table1.primarykey;
}
```

假定我们正在访问第3页，选定订单的订单编号为161130078，此时table1.pagenumber等于2，table1.primarykey等于161130078，以上代码合成的链接为：

list.htm?page=2&deloid=161130078

服务器收到这个访问请求之后，触发HttpRequest事件，转给List函数负责处理，List函数执行第3行到第6行代码，删除编号为161130078的订单及其订单明细。

编辑订单

当我们在菜单中点击"编辑订单“时，会执行 js 函数 edit，这个函数的代码为：

```
function edit(){
   location="edit.htm?page=" + table1.pagenumber + "&oid=" + table1.primarykey;
}
```

假定我们正在访问第 3 页，选定订单的订单编号为 161130078，此时 table1.pagenumber 等于 2，table1.primarykey 等于 161130078，以上代码合成的链接为：

edit.htm?page=2&oid=161130078

服务器收到这个访问请求之后，触发 HttpRequest 事件，转给 Edit 函数负责处理，Edit 函数是如何处理以上访问请求的，在介绍 Edit 函数的时候会详细讲述。

增加订单

当我们单击"增加订单"按钮时，会调用js函数addnew，这个函数的代码为：

```
function addnew(){
     location="edit.htm?page=" + table1.pagenumber;
}
```

假定我们正在访问第3页，此时table1.pagenumber等于2，以上代码合成的链接为：

edit.htm?page=2

服务器收到这个访问请求之后，触发HttpRequest事件，转给Edit函数负责处理，Edit函数是如何处理以上访问请求的，在介绍Edit函数的时候会详细讲述。

7.56.8 Edit 函数

Edit 函数用于生成订单编辑页面 edit.htm。

按照我们之前的设计思路，订单的增加与编辑，以及订单明细的增加、编辑和删除，都在这个页面完成，所以 Edit 函数的代码比较长，超过 100 行。

该函数的代码为：

```
Dim e As RequestEventArgs = args(0)
Dim wb As New weui
'删除订单明细
If e.GetValues.ContainsKey("deldid") Then
    DataTables("订单明细").SQLDeleteFor("[_Identify] = " & e.GetValues("deldid"))
End If
```

```
'订单编辑
Dim pr As DataRow '订单
Dim srs As List(of DataRow) '订单明细集合
Dim Page As Integer '页码变量
If e.GetValues.ContainsKey("page") Then   '如果地址中有page参数
    Integer.TryParse(e.GetValues("page"), page) '提取page参数
End If
If e.GetValues.ContainsKey("oid") = False Then '如果没有传递订单编号,则新增与一个订单
    pr = DataTables("订单").SQLAddNew() '
    pr("订单编号") = Functions.Execute("GetOrderID") '利用自定义函数GetOrderID为新增订单生
成编号.
    pr("日期") = Date.Today()
Else '如果传递了订单编号,则找出此订单进行编辑
    pr = DataTables("订单").SQLFind("订单编号='" & e.GetValues("oid") & "'")
    If pr Is Nothing Then '多用户情况下,必须考虑其他用户删除订单的可能.
        wb.InSertHtml("此订单已被其他用户删除!")
        e.WriteString(wb.Build)
        Return ""   '必须返回
    End If
    srs =   DataTables("订单明细").SQLSelect("订单编号='" & pr("订单编号")   & "'") '获取订单明细
End If
Dim url As String = "edit.htm?page=" & page &   "&oid=" & pr("订单编号")   '传递页码和订单编号
wb.AddForm("","form1",url)
With wb.AddInputGroup("form1","ipg1",iif(e.GetValues.ContainsKey("oid"),"编辑订单","新增订单"))
    With .AddInput("订单编号","编号","text")
        .Value = pr("订单编号")
        .Readonly = True
    End With
    .AddInput("客户","客户","text").Value = pr("客户")
    .AddInput("日期","日期","date").Value = pr("日期")
    If e.GetValues.ContainsKey("oid")   Then '如果是旧订单,则汇总显示数量和金额
        Dim qty As Integer
        Dim amt As Integer
        For Each sr As DataRow In srs
            qty = qty + sr("数量")
            amt = amt + sr("数量") * sr("单价")
        Next
        .AddInput("总数量","总数量","number").value = qty
        .AddInput("总金额","总金额","number").value = amt
    End If
End With
'订单明细编辑
Dim mr As DataRow '要编辑的订单明细
Dim IsNew As Boolean '此变量用于标记是否要新增明细
```

```
If e.GetValues.ContainsKey("oid") = False OrElse e.GetValues.ContainsKey("addnext") '如果是新增订单,或这包括addnext参数
    IsNew = True '将IsNew参数设置为True,表明需要新增订单明细
ElseIf e.GetValues.ContainsKey("did") Then '如果传递了订单明细主键
    mr = DataTables("订单明细").SQLFind("[_Identify]=" & e.GetValues("did")) '找出此订单明细进行编辑
End If
If IsNew OrElse mr IsNot Nothing Then
    With wb.AddInputGroup("form1","ipg2",iif(IsNew,"新增明细","编辑明细"))
        .Attribute = "onchange='calc()'" '调用js函数,自动计算金额
        If IsNew Then '如果是新增订单明细
            .AddInput("产品","产品","text")
            .AddInput("数量","数量","number")
            .AddInput("单价","单价","number").Step= "0.01"
            .AddInput("金额","金额","number")
        Else
            .AddHiddenValue("DetailID",mr("_Identify")) '插入一个隐藏的订单明细主键,此值将随表单数据一并提交到服务器.
            .AddInput("产品","产品","text").Value = mr("产品")
            .AddInput("数量","数量","number").value = mr("数量")
            With .AddInput("单价","单价","number")
                .Step= "0.01"
                .value = mr("单价")
            End With
            .AddInput("金额","金额","number").value = mr("数量") * mr("单价") '后台没有金额列,要通过数量和单价计算得出
        End If
    End With
End If
'生成订单明细表格
If e.GetValues.ContainsKey("oid") AndAlso srs.count > 0 Then '如果不是新增订单,且订单明细行数大于0,则生成订单明细表格.
    With wb.AddTable("form1","detailTable") '为了区分,明细表的名字设为detailTable
        .head.AddRow("产品","数量","单价","金额")
        .ActiveSheet = "menu" '指定菜单
        For Each sr As DataRow In srs
            With .Body.AddRow(sr("产品"),sr("数量"),sr("单价"))
                .AddCell(sr("数量") * sr("单价"))
                .Primarykey = sr("_Identify") '为此行指定主键值
            End With
        Next
    End With
    '设计菜单
    With wb.AddActionSheet("","menu")
```

```
95        .Add("mnudAdd", "增加明细").Attribute="onclick='addDetail()'" '调用js函数
96        .Add("mnuEdit", "编辑明细").Attribute ="onclick='editDetail()'"
97        .Add("mnuDelete", "删除明细").Attribute ="onclick=""show('dlg1')"""
98        .Add("mnuCancel","取消","",True)
99    End With
10 End If
0  With wb.AddDialog("","dlg1", "删除确认","您确定要删除当前明细吗?")
10    .AddButton("btnCancel","取消").Kind = 1
1     .AddButton("btnOK","确定").Attribute = "onclick='delDetail()'"
10 End With
2  With wb.AddButtonGroup("form1","btg1",False)
10    .Add("btn1", "增加明细", "submit").FormAction = url & "&addnext=true" '加上addnext参数,表
3  示保存后进入增加明细状态
10    .Add("btn2", "保存", "submit") '正常提交,保存后进入编辑状态
4     If  e.GetValues.ContainsKey("oid") = False Then '如过是新增订单
10        .Add("btn3", "取消", "button","list.htm?page=" & page & "&deloid=" & pr("订单编号")) '删除
5  新增订单后返回列表
10    ElseIf IsNew OrElse mr IsNot Nothing  Then '如果在给旧订单新增或修改明细,则直接返回编辑
6  状态
10        .Add("btn3", "取消", "button",url) '返回编辑状态
7     Else
10        .Add("btn3", "返回", "button","list.htm?page=" & page) '返回列表
8     End If
10 End With
9  pr.Save()  '必须保存,而且必须在最后保存,因为SQLAddNew增加的行,保存之后就会销毁,无法再
11 调用
0  If e.PostValues.Count > 0 Then '如果是通过提交按钮访问,则给一个已经保存的提示给用户,时长
11 500毫秒
1     wb.AddToast("","t1", "已经保存",0).Msec= 500
11 End If
2  wb.AppendHTML("<script src='./lib/order.js'></script>") '引入脚本文件
11 e.WriteString(wb.Build) '生成网页
3
11
4
11
5
```

我们前面已经提到，上面的函数完成了5项任务，所以我们要分5种情况来解析代码。
我强烈建议你直接看上面的代码，自行分析，代码的逻辑很清晰，而且我加上了注释，理解起来不困难，实在看不懂的时候，再来看下面的解析不迟。
只要你明白了之前已经理顺的设计思路，看懂上述代码应该不会有问题的。

增加订单

我们已经知道，当用户在list.htm中，单击"增加订单"按钮，会向服务器发送访问请求：
edit.htm?page=2
服务端收到这个访问请求后，触发HttpRequest事件，将请求转给Edit函数处理。

以下是edit函数的处理流程：

第11行到第13行代码，从GetValues集合中提取page参数，保存在变量page中，page变量的值等于2。
由于没有传递oid参数，第14行代码的条件成立，执行第15行到第17行代码。
第15行代码新增一个订单。
第16行代码利用自定义函数GetOrderID为新订单生成订单编号。
第17行代码将订单日期设置为当天日期。

假定新增订单编号为161203006，第27行代码合成了一个链接，保存在url中，内容为：
edit.htm?page=2&oid=161203006
第28行代码新增一个表单，数据接收链接为url，即：edit.htm?page=2&oid=161203006。
第29到第46行代码，生成了订单输入框。
由于第36行代码的条件不成立，不会生成总数量和总金额输入框。

第50行代码的条件成立，IsNew参数设置为True，表明需要新增一个订单明细。
第59到第62行代码，为订单明细生成输入框。

第57行代码调用了js函数calc，这个函数的代码为：
```
function calc(){
    document.getElementById("金额").value = document.getElementById("数量").value * document.getElementById("单价").value;
}
```
这样用户输入订单明细的数量和单价后，能自动计算出金额。
注意这里没有单独给数量和单价输入框定义onchange事件，而是给他们的父容器定义onchange事件，在父容器定义的事件，对于子元素有效，JavaScript这个特性真不错。

第76行行代码的条件不成立，所以不会生成明细表格及相关菜单。
第101行代码定义了一个提交按钮，标题为"保存"。
第100行代码定义了一个提交按钮，标题为"增加明细",数据接收链接为:edit.htm?page=2&oid=161203006&addnext=true
这样用户单击"保存明细"后,服务器通过分许get参数，就知道用户要保存本次输入结果，然后开始下一个明细的输入。
第103行代码定义了一个取消按钮，链接为:list.htm?page=2&deloid=161203006
这样用户单击取消按钮，会返回list.htm，并显示第3页数据，在显示数据之前会先删除新增加的订单。

第111行代码判断本次访问是否含有有表单数据。
如果本次访问包括有表单数据，则在第112行代码创建一个Toast，显示0.5秒，内容为"已经保存".
需要注意的时候，表单数据的保存是通过Save函数完成的，Edit函数只负责生成网页内容。

综上所示，edit.htm在新增订单状态下的显示内容为：



编辑订单

假定客户端正在访问list.htm，浏览到第3页，选择该页中订单编号为161203005的订单，然后执行菜单命令“编辑订单”。
服务端会收到以下访问请求：
edit.htm?page=2&oid=161203005
服务端收到这个访问请求后，触发HttpRequest事件，将请求转给Edit函数处理。

下面是Edit函数的处理流程，和增加订单相同的部分就不重复讲述了：

第19行代码根据传递过来的订单编号找出订单，保存在变量pr中。
第25行代码提取出这个订单所有的订单明细，保存在集合srs中。
第29行到第46行代码，为这个订单生成了输入框。
由于后台的订单表并不存在总数量和总金额两列，所以第37到第44行代码，累加了订单明细的数量和金额，赋值给对应输入框的value属性。

由于没有传递addnext和did参数，所以第50行代码和第52行代码的条件都不成立。
因此第55行代码的条件也不成立，本次请求不会生成订单明细输入框。

第76到第94行代码，生成订单明细表格，以及操作明细的菜单。
由于订单明细表的金额是表达式列，无法用CreateDataTable自动生成，所以只能逐行生成表格，逐行设置主键。

第107行代码定义了一个返回按钮，链接为:list.htm?page=2
这样用户单击返回按钮，会返回list.htm，并显示第3页数据。

综上所述，在编辑订单状态下，edit.htm显示内容为：



新增明细

假定用户正在编辑编号为161203005，单击了"新增明细"按钮。
服务器会收到以下访问请求：
edit.htm?page=2&oid=161203005&addnext=true
服务端收到这个访问请求后，触发HttpRequest事件，由于PostValue集合包括有表单数据，所以HttpRequest事件先执行Save函数保存数据，然后将请求转给Edit函数处理。

Edit函数的处理流程和编辑订单基本相同，下面只讲述不同的地方：

由于访问请求中包括addnext参数，第50行代码的条件成立。

第51行代码将变量IsNew设置为True，表明需要新增明细。
第55行和第58行代码的条件成立，执行第59到第62行代码，为新增明细定义输入框。

第104行代码的条件成立，第105代码生成一个取消按钮，这个取消按钮的链接地址为：edit.htm?page=2&oid=161203005
返回按钮的类型不是submit，所以单击这个按钮，不会提交数据，而是直接返回到订单编辑状态，等于取消了本次新增明细操作。

综上所述，在新增明细状态下，edit.htm的显示内容为：



小提示：如果想直接增加订单明细，不想保存当前输入结果，可以点击菜单中的"增加明细"命令。

编辑明细

假定用户正在编辑编号为161203005的订单，连续点击订单明细表格的同一单元格，在弹出的菜单中执行"编辑明细"命令，会调用js函数editDetail。
editDetail函数的代码为：

```
function editDetail(){
   location = form1.action + "&did=" + detailtable.primarykey;
}
```

form1.action获取表单的action属性(数据接收链接)，值为:edit.htm?page=2&oid=161203005
假定用户点击的订单明细的主键为567，那么执行js函数editDetail，会向服务器发出访问请求：
edit.htm?page=2&oid=161203005&did=567
服务端收到这个访问请求后，触发HttpRequest事件，将请求转给Edit函数处理。

edit函数的处理流程和新增明细基本相同，不同的部分只有：

由于访问请求中包括did参数(订单明细主键)，第52行代码的条件成立。
第53行代码根据did参数，找出对应的订单明细，保存再变量mr中。
第64到第71行代码，为此订单明细生成输入框。
第64行代码，插入了一个隐藏的输入框，名为DetailID，值为此明细的主键。
这样服务器收到用户提交的表单数据后，通过判断PostValues集合是否包括DetailID参数，就能知道是用户新增明细还是编辑明细。
第71行代码，由于订单明细表的金额列是表达式列，并不存在于后台数据表，所以需要用代码计算出金额，并赋值给金额输入框的value属性

综上所述，在编辑明细状态下，edit.htm的显示内容为：



删除明细

假定用户正在编辑编号为161203005，连续点击订单明细表格的同一单元格，在弹出的菜单中执行"删除明细"命令， 会显示一个对话框(第91行代码)：



如果用户单击确定，会调用js函数delDetail（第97行代码），该函数的代码为：

```
function delDetail() {
   location = form1.action + "&deldid=" + detailtable.primarykey;
}
```

假定用户在订单明细表格中点击的是主键为568的明细，那么执行js函数delDetail，会向服务器发出访问请求：

edit.htm?page=2&oid=161203005&deldid=568

服务端收到这个访问请求后，触发HttpRequest事件，将请求转给Edit函数处理。

Edit函数的处理流程如下：

第4行代码判断GetValues集合是否包括deldid参数。
如果包括过deldid参数，则执行第5行代码，删除主键等于deldid参数的订单明细。
接下来处理流程，和编辑订单请求完全相同，就不重复了。

综上所述，删除订单明细后，edit.htm重新进入订单编辑状态，显示内容为：



7.56.9 Save 函数

Save 函数负责保存表单内容，包括订单和订单明细，其代码为：

```
Dim e As RequestEventArgs = args(0)
'保存订单
Dim dr As DataRow = DataTables("订单").SQLFind("订单编号='" & e.PostValues("订单编号") & "'")
Dim nms() As String = {"客户","日期"}  '''
If dr IsNot Nothing Then
    For Each nm As String In nms
        If e.PostValues.ContainsKey(nm) Then
            dr(nm)= e.PostValues(nm)
        End If
    Next
End If
dr.Save()
'保存明细
Dim valid As Boolean '用于判断用户是否 输入了订单明细数据
nms = New String() {"产品","数量","单价" }
For Each nm As String In nms
    If e.PostValues.ContainsKey(nm) Then
        valid = True '如果 输入了订单明细数据,将valid变量设置为True
    End If
Next
If valid Then '如果提交了订单明细数据
    Dim sr As DataRow
    If e.PostValues.ContainsKey("DetailID") Then '如果传递了订单明细主键
        sr = DataTables("订单明细").SQLFind("[_Identify]=" & e.PostValues("DetailID"))
        If sr Is Nothing  Then  '多用户环境,必须考虑其他用户删除此明细的可能,避免程序报错
            Return ""
        End If
    Else
        sr = DataTables("订单明细").SQLAddNew() '增加一个订单明细
        sr("订单编号") = dr("订单编号")
    End If
    For Each nm As String  In nms
        sr(nm) = e.PostValues(nm)
    Next
    sr.Save()  '必须保存,而且必须在最后保存,因为用SQLAddNew增加的行,一旦保存,就不能再引用此行
End If
```

```
7
2
8
2
9
3
0
3
1
3
2
3
3
3
4
3
5
3
6
```

上述代码很好理解，只是需要注意第23行代码，通过判断PostValues集合中是是否包括DetailID，来判断出用户是编辑明细，还是新增明细。
你要是不记得DetailID是怎么来的，请回头看一下Edit函数的代码。
另外代码还采取了措施，避免保存空的订单明细，用户必须给订单明细至少输入一列的内容，此明细才会被保存。

SQLAddNew和主键

用SQLAddNew增加行的时候，并没有真正在后台数据库增加行，直到用代码保存此行，此行才会写入后台数据库，保存之后，此行不能再访问。
这带来一个问题，如果表使用的是自动增量型主键，那么行的主键只能在保存后生成，但是SQLAddNew增加的行在保存后又无法再访问，所以我们是无法获取新增行的主键 值。
如果你需要获取行的主键，请使用DataTable的AddNew方法增加行，保存之后，会自动生成行的主键，还可以继续访问此行，获取包括主键在内的所有列的数据。

7.56.10 GetOrderID 函数

GetOrderID函数用于生成订单编号
一开始我是直接在Edit函数中编写代码生成订单编号的，后来之所以将生成编号的代码单独提出来做成一个函数，是考虑到用户可能需要直接在Foxtable输入订单。

编号格式为两位年两位月和两位日，后接三位顺序号。

GetOrderID函数代码为：

```
Static Lastbh As String '通过这个Static变量可以访问上一次生成的编号,避免重复Compute
Dim prefix As String = Format(Date.now,"yyMMdd") '本次编号前缀
If Lastbh >"" AndAlso Lastbh.SubString(0,6) = prefix Then '如果上一次生成编号的前缀和本次相同
    Lastbh = prefix & Format(Cint(Lastbh.SubString(6,3)) + 1,"000") '在上次编号的基础上递增1
Else
    '取数据表中同前缀的最大编号
    Dim max As String=DataTables("订单").SQLCompute("Max(订单编号)","订单编号 like '" & prefix & "%'")
    If max = "" Then '如果不存在同前缀的编号
        Lastbh  = prefix & "001"  '同前缀的第一个编号
    Else
        Lastbh = prefix & Format(CInt(max.SubString(6,3)) + 1,"000") '在同前缀最大编号的基础上递增1
    End If
End If
Return Lastbh
```

注意上述代码中，Lastbh是一个静态变量，每次执行都可以通过此变量获取上一次生成的编号，这样处理的好处是，不用次次执行SQLCompute生成新的编号，提高了效率。

关于静态变量，参考：使用静态变量

7.56.11 继续扩展功能

7.56.11.1 加上数据筛选功能

知识准备

为了方便大家理解掌握，不要被枝节问题干扰，所以前面的例子并没有设计数据筛选功能。

当然设计一个普通的数据筛选并不复杂，可以参考：

数据筛选

数据筛选和分页

不过本节的筛选涉及到父表(订单)和子表(订单明细)，所以处理起来有点特别。

我们知道，一个订单可以订单多个产品，假定我们要筛选出订购过产品"PD01"的订单，由于订单表并不包括产品列，需要用如下的语句查询：

```
Select * From {订单} Where 订单编号 in (Select Distinct 订单编号 From 订单明细 Where 产品 = 'PD01')
```

上面的查询有两个 Select 语句，执行步骤为：

1、从订单明细表中找出产品为 PD01 的不重复的订单编号，执行红色部分的 Select 语句：

```
Select Distinct 订单编号 From 订单明细 Where 产品 = 'PD01'
```

2、然后从订单表中找出订单编号包括在以上查询结果中的订单：

Select * From {订单} Where 订单编号 in (第一次查询的结果)

设计要求

1、菜单中增加一个"筛选数据"命令：



2、执行“数据筛选”命令，显示数据筛选页(filter.htm)：



3、在数据筛选页输入条件后，单击筛选按钮，显示符合条件的数据，且此时菜单会显示一个“撤销筛选”命令：



设计思路

筛选部分设计的思路非常简单，在 list 函数中增加代码，根据用户输入的条件合成条件表达式，为避免 list 函数的代码过程，我们可以另外增加一个函数(GetFilter)用于合成条件表达式。

至于撤销筛选，我们可以传递一个约定的 get 参数给 list.htm，例如：

List.htm?page=0&unfilter=true

list 函数判断存在 unfilter 参数后，清除筛选条件后，显示第一个数据。

相关代码

1、在 order.js 中增加一个函数，代码为：

```
function filter() {
    location="filter.htm?page=" + table1.pagenumber
}
```

在菜单中点击"筛选数据"命令，将执行此函数。

2、增加一个自定义函数 Filter，用于生成筛选页面(Filter.htm)：

```
Dim e As RequestEventArgs = args(0)
Dim wb As New weui
Dim Page As Integer
If e.GetValues.ContainsKey("page") Then
    Integer.TryParse(e.GetValues("page"), page)
End If
wb.AddForm("","form1","list.htm?page=0")
With wb.AddInputGroup("form1","ipg1","数据筛选")
    .AddInput("订单编号","订单编号","text")
    .AddInput("客户","客户","text")
    .AddInput("产品","产品","text")
    .AddInput("开始日期","开始日期","date")
    .AddInput("结束日期","结束日期","date")
End With
With wb.AddButtonGroup("form1","btg1",False)
    .Add("btn1", "筛选", "submit")
    .Add("btn1", "返回", "button","list.htm?page=" & Page) '返回原页面
End With
e.WriteString(wb.Build())
```

注意表单的数据接收页为：

list.htm?page=0

将输入的条件传递给List函数，有List函数负责筛选出数据，并显示第一页符合条件的数据。

3、增加一个GetFilter函数，供List函数调用，用于根据Filter.htm的输入结果合成条件表达式，这段代码有点长，不过很好理解：

```
Dim bh As String '订单编号
Dim cp As String '产品
Dim kh As String   '客户
Dim rq1 As String '开始日期
Dim rq2 As String '结束日期
Dim Filter As String   '条件表达式
Dim e As RequestEventArgs = args(0)
Dim wb As WeUI = args(1)
If e.PostValues.Count > 0 Then '如果是通过表单输入了筛选条件
    If e.PostValues.ContainsKey("订单编号") Then
        bh =   e.PostValues("订单编号")
        wb.Appendcookie("bh", bh)
    Else
        wb.DeleteCookie("bh")
    End If
    If e.PostValues.ContainsKey("产品") Then
```

```
        cp = e.PostValues("产品")
        wb.Appendcookie("cp", cp)
    Else
        wb.DeleteCookie("cp")
    End If
    If e.PostValues.ContainsKey("客户") Then
        kh = e.PostValues("客户")
        wb.Appendcookie("kh", kh)
    Else
        wb.DeleteCookie("kh")
    End If
    If e.PostValues.ContainsKey("开始日期") Then
        rq1= e.PostValues("开始日期")
        wb.Appendcookie("rq1", rq1)
    Else
        wb.DeleteCookie("rq1")
    End If
    If e.PostValues.ContainsKey("结束日期") Then
        rq2 = e.PostValues("结束日期")
        wb.Appendcookie("rq2",rq2)
    Else
        wb.DeleteCookie("rq2")
    End If
Else '否则从Cookie中提取筛选条件
    If e.Cookies.ContainsKey("bh") Then
        bh = e.Cookies("bh")
    End If
    If e.Cookies.ContainsKey("cp") Then
        cp = e.Cookies("cp")
    End If
    If e.Cookies.ContainsKey("kh") Then
        kh = e.Cookies("kh")
    End If
    If e.Cookies.ContainsKey("rq1") Then
        rq1 = e.Cookies("rq1")
    End If
    If e.Cookies.ContainsKey("rq2") Then
        rq2 = e.Cookies("rq2")
    End If
End If
If bh > "" Then '如果输入了订单编号,其他条件可以忽略
    Return "产品编号  = '" & bh & "'"
End If
If cp > "" Then '如果输入产品
```

```
    '产品列在子表订单明细,不是在父表订单,注意这种根据子表列合成父表条件的技巧
    Filter =  "订单.订单编号 in (Select Distinct 订单编号 From {订单明细} Where 产品= '" & cp  & "')"
End If
If  kh > "" Then
    If Filter > "" Then
        Filter = Filter & " And "
    End If
    Filter = Filter & "客户  = '" & kh & "'"
End If
If  rq1 > "" Then
    If Filter > "" Then
        Filter = Filter & " And "
    End If
    Filter = Filter & "日期  >= '" & rq1 & "'"
End If
If  rq2 > "" Then
    If Filter > "" Then
        Filter = Filter & " And "
    End If
    Filter = Filter & "日期  <= '" & rq2 & "'"
End If
Return Filter
```

4、修改 HttpRequest 事件代码，粗体部分是新加上的：

```
Select Case e.Path
    Case "list.htm"
        Functions.Execute("List",e) '分页显示
    Case "edit.htm"
        If e.PostValues.Count > 0 Then
            Functions.Execute("Save",e) '保存表单数据
        End If
        Functions.Execute("Edit",e) '生成订单编辑页面
    Case "filter.htm"
        Functions.Execute("Filter",e)
    Case "order.xls"
        Functions.Execute("CreateXLS",e)
End Select
```

5、最后我们需要对 List 函数稍作修改，增加一些内容，新增加的内容我用粗体显示，可以看到增加内容很少：

```
Dim e As RequestEventArgs = args(0)
Dim wb As New WeUI
```

```
'订单删除代码
If e.GetValues.ContainsKey("deloid") Then '如果提交了deloid参数,则删除对应的订单.
    DataTables("订单").SQLDeleteFor("订单编号='" & e.GetValues("deloid") & "'")
    DataTables("订单明细").SQLDeleteFor("订单编号='" & e.GetValues("deloid") & "'")
End If
'获取要显示的页
Dim page As Integer = 0 '默认page为0,显示第一页
Dim pageRows As Integer = 10 '每页10行
If e.GetValues.ContainsKey("page") Then   '如果地址中有page参数
    Integer.TryParse(e.GetValues("page"), page) '提取page参数
End If
Dim StartRow As Integer = page * pageRows + 1 '此页第一行
Dim EndRow As Integer = (page + 1) * pageRows '此页最后一行
Dim Filter As String   '条件表达式
If e.GetValues.ContainsKey("unfilter") Then
    wb.ClearCookie() '清除Cookie
Else
    Filter = Functions.Execute("GetFilter",e， wb)'合成条件表达式
End If
'获取该页数据
Dim cmd As New SQLCommand
cmd.ConnectionName = "orders" '记得设置数据源名称
cmd.CommandText = "Select Count(*) From {订单}"   & iif(Filter > "", " Where "   & Filter, "")
Dim Count As Integer = cmd.ExecuteScalar() '获取总的行数
Dim Pages As Integer = Math.Ceiling(Count/PageRows) '计算出总页数
cmd.CommandText = "Select * From (Select  Row_Number() Over(Order by 订单.订单编号 desc ) As
RowNum,订单.订单编号,日期,客户,Sum(数量) As 数量,sum(数量*单价) As 金额" & _
" From 订单 Left JOIN 订单明细 ON 订单明细.订单编号 = 订单.订单编号 " & iif(Filter > "", "
Where " & Filter ,"") & _
" Group By {订单}.订单编号,日期,客户) As a  Where RowNum >= " & StartRow & " And RowNum <= "
& EndRow
Dim dt As DataTable = cmd.ExecuteReader
'根据此页数据生成表格
With wb.AddTable("","Table1")
    .PageNumber = page '设置页码
    .ActiveSheet = "menu" '指定菜单
    .Primarykey = "订单编号" '指定主键,只要是能唯一区分行的列即可,并非一定要表的实际主键.
    .CreateFromDataTable(dt, False,"","","订单编号","客户","日期","数量","金额")
End With
'设计菜单
With wb.AddActionSheet("","menu")
    .Add("mnudAdd", "增加订单").Attribute="onclick='addnew()'" '调用js函数
    .Add("mnuEdit", "编辑订单").Attribute ="onclick='edit()'"
    .Add("mnuDelete", "删除订单").Attribute ="onclick=""show('dlg1')"""
```

```
    .Add("mnuFirst","第一页","List.htm?page=0",True)
    .Add("mnuLast","最末页","List.htm?page=" & pages - 1)
    If Filter = "" Then
      .Add("mnuFilter","筛选数据").Attribute = "onclick='filter()'"
    Else
      .Add("mnuUnFilter","撤销筛选","List.htm?page=0&unfilter=true")
    End If
    .Add("mnuCancel","取消","",True)
End With
With wb.AddDialog("","dlg1", "删除确认","您确定要删除当前订单吗?")
    .AddButton("btnCancel","取消").Kind = 1
    .AddButton("btnOK","确定").Attribute = "onclick='del()'"
End With
'生成换页按钮
With wb.AddButtonGroup("","btg1", False)
    .Add("btnAdd", "增加订单").Attribute = "onclick='addnew()'"
    If page > 0 Then
      .Add("btnPrev", "上一页","","List.htm?page=" & page - 1)
    Else
      .Add("btnPrev", "上一页").Kind = 1
    End If
    If Endrow < count Then
      .Add("btnNext", "下一页","","List.htm?page=" & page + 1)
    Else
      .Add("btnNext", "下一页").Kind = 1
    End If
End With
wb.AppendHTML("<script src='./lib/order.js'></script>") '引入脚本文件
e.WriteString(wb.Build)
```

7.56.11.2 生成 Excel 格式的订单

本节的任务是在订单编辑页面，加上一个按钮，用于生成 Excel 格式的订单：



整个设计逻辑很简单，假定当前订单编号为“161203001”，用户单击“生成 Excel 格式订单”按钮后，会向服务器发送访问请求：

order.xls?oid=161203001

HttpRequest 事件收到这个访问请求后，将请求转给自定义函数(CreateXLS)处理，此函数提取出编号为 161203001 的订单和订单明细，据此生成 Excel 报表，并将生成的报表发送给用户。

所有的代码加起来只有 18 行：

设计步骤

1、首先要设计好一个 Excel 报表模板：

	A	B	C	D	E	F	G
1							<END>
2	<订单,1>	湛江佳信公司订单					
3							
4		<"编号:" & [订单编号]>					
5		<"客户:" & [客户] >			<"日期:" & [日期]>		
6	<HeaderRow>	产品	数量	单价	金额	备注	
7	<订单明细,-9>	[产品]	[数量]	[金额]	[金额]		
8	<订单>	合计:[明细,Sum(数量)]		[%订单明细,Sum(金额)]			
9	<END>						

2、增加一个自定义函数 CreateXls，用于生成 Excel 报表：

```
Dim e As RequestEventArgs = args(0)
Dim oid As String
If e.GetValues.ContainsKey("oid") Then
    oid = e.GetValues("oid")
Else
    Return ""
End If
Dim Book As New XLS.Book(ProjectPath & "Attachments\订单.xls")
Dim fl As String = ProjectPath & "Reports\订单.xls"
book.AddDataTable("订单","orders","Select * from {订单} where  订单编号= '" & oid & "'") '添加父表
book.AddDataTable("订单明细","orders","Select *,数量 * 单价 As 金额 from {订单明细} where 订单编号= '" & oid & "'") '添加子表
book.AddRelation("订单","订单编号","订单明细","订单编号") '建立关联
e.WriteBook(book,"订单" & oid & ".xls",False)
```

代码逻辑很简单，根据传递过来的订单编号，找出对应的订单和订单明细，生成 Excel 报表。

参考：

根据后台数据生成 Excel 报表

提升 Excel 报表效率

3、修改 HttpRequst 事件代码，粗体部分是我们新增加的代码：

```
Select Case e.Path
    Case "list.htm"
        Functions.Execute("List",e) '分页显示
    Case "edit.htm"
        If e.PostValues.Count > 0 Then
            Functions.Execute("Save",e) '保存表单数据
        End If
        Functions.Execute("Edit",e) '生成订单编辑页面
    Case "filter.htm"
        Functions.Execute("Filter",e)
```

```
    Case "order.xls"
        Functions.Execute("CreateXLS",e)
End Select
```

3、最后在 Edit 函数增 3 行代码（最后面的粗体部分），用于添加生成 Excel 报表的按钮：

```
Dim e As RequestEventArgs = args(0)
Dim wb As New weui
'删除订单明细
If e.GetValues.ContainsKey("deldid") Then
    DataTables("订单明细").SQLDeleteFor("[_Identify] = " & e.GetValues("deldid"))
End If
'订单编辑
Dim pr As DataRow '订单
Dim srs As List(of DataRow) '订单明细集合
Dim Page As Integer '页码变量
If e.GetValues.ContainsKey("page") Then   '如果地址中有page参数
    Integer.TryParse(e.GetValues("page"), page) '提取page参数
End If
If e.GetValues.ContainsKey("oid") = False Then '如果没有传递订单编号,则新增与一个订单
    pr = DataTables("订单").SQLAddNew() '
    pr("订单编号") = Functions.Execute("GetOrderID") '利用自定义函数GetOrderID为新增订单生成编号.
    pr("日期") = Date.Today()
Else '如果传递了订单编号,则找出此订单进行编辑
    pr = DataTables("订单").SQLFind("订单编号='" & e.GetValues("oid") & "'")
    If pr Is Nothing Then '多用户情况下,必须考虑其他用户删除订单的可能.
        wb.InSertHtml("此订单已被其他用户删除!")
        e.WriteString(wb.Build)
        Return ""   '必须返回
    End If
    srs =   DataTables("订单明细").SQLSelect("订单编号='" & pr("订单编号")   & "'") '获取订单明细
End If
Dim url As String = "edit.htm?page=" & page &   "&oid=" & pr("订单编号")   '传递页码和订单编号
wb.AddForm("","form1",url)
With wb.AddInputGroup("form1","ipg1",iif(e.GetValues.ContainsKey("oid"),"编辑订单","新增订单"))
    With .AddInput("订单编号","编号","text")
        .Value = pr("订单编号")
        .Readonly = True
    End With
    .AddInput("客户","客户","text").Value = pr("客户")
    .AddInput("日期","日期","date").Value = pr("日期")
    If e.GetValues.ContainsKey("oid")   Then '如果是旧订单,则汇总显示数量和金额
        Dim qty As Integer
        Dim amt As Integer
        For Each sr As DataRow In srs
```

```
        qty = qty + sr("数量")
        amt = amt + sr("数量") * sr("单价")
      Next
      .AddInput("总数量","总数量","number").value = qty
      .AddInput("总金额","总金额","number").value = amt
    End If
End With
'订单明细编辑
Dim mr As DataRow '要编辑的订单明细
Dim IsNew As Boolean '此变量用于标记是否要新增明细
If e.GetValues.ContainsKey("oid") = False OrElse e.GetValues.ContainsKey("addnext") '如果是新增订单,或这包括addnext参数
    IsNew = True '将IsNew参数设置为True,表明需要新增订单明细
ElseIf e.GetValues.ContainsKey("did") Then '如果传递了订单明细主键
    mr = DataTables("订单明细").SQLFind("[_Identify]=" & e.GetValues("did")) '找出此订单明细进行编辑
End If
If IsNew OrElse mr IsNot Nothing  Then
    With wb.AddInputGroup("form1","ipg2",iif(IsNew,"新增明细","编辑明细"))
      .Attribute = "onchange='calc()'" '调用js函数,自动计算金额
      If IsNew Then '如果是新增订单明细
        .AddInput("产品","产品","text")
        .AddInput("数量","数量","number")
        .AddInput("单价","单价","number").Step= "0.01"
        .AddInput("金额","金额","number")
      Else
        .AddHiddenValue("DetailID",mr("_Identify")) '插入一个隐藏的订单明细主键,此值将随表单数据一并提交到服务器.
        .AddInput("产品","产品","text").Value = mr("产品")
        .AddInput("数量","数量","number").value = mr("数量")
        With .AddInput("单价","单价","number")
          .Step= "0.01"
          .value = mr("单价")
        End With
        .AddInput("金额","金额","number").value = mr("数量") * mr("单价") '后台没有金额列,要通过数量和单价计算得出
      End If
    End With
End If
'生成订单明细表格
If e.GetValues.ContainsKey("oid") AndAlso  srs.count > 0 Then '如果不是新增订单,且订单明细行数大于0,则生成订单明细表格.
    With wb.AddTable("form1","detailTable") '为了区分,明细表的名字设为detailTable
      .head.AddRow("产品","数量","单价","金额")
```

```
        .ActiveSheet = "menu" '指定菜单
        For Each sr As DataRow In srs
            With .Body.AddRow(sr("产品"),sr("数量"),sr("单价"))
                .AddCell(sr("数量") * sr("单价"))
                .Primarykey = sr("_Identify") '为此行指定主键值
            End With
        Next
    End With
    '设计菜单
    With wb.AddActionSheet("","menu")
        .Add("mnudAdd", "增加明细").Attribute="onclick='addDetail()'" '调用js函数
        .Add("mnuEdit", "编辑明细").Attribute ="onclick='editDetail()'"
        .Add("mnuDelete", "删除明细").Attribute ="onclick=""show('dlg1')"""
        .Add("mnuCancel","取消","",True)
    End With
End If
With wb.AddDialog("","dlg1", "删除确认","您确定要删除当前明细吗?")
    .AddButton("btnCancel","取消").Kind = 1
    .AddButton("btnOK","确定").Attribute = "onclick='delDetail()'"
End With
With wb.AddButtonGroup("form1","btg1",False)
    .Add("btn1", "增加明细", "submit").FormAction = url & "&addnext=true" '加上addnext参数,表示保存后进入增加明细状态
    .Add("btn2", "保存", "submit") '正常提交,保存后进入编辑状态
    If  e.GetValues.ContainsKey("oid") = False Then '如过是新增订单
        .Add("btn3", "取消", "button","list.htm?page=" & page & "&deloid=" & pr("订单编号")) '删除新增订单后返回列表
    ElseIf IsNew OrElse mr IsNot Nothing  Then '如果在给旧订单新增或修改明细,则直接返回编辑状态
        .Add("btn3", "取消", "button",url) '返回编辑状态
    Else
        .Add("btn3", "返回", "button","list.htm?page=" & page) '返回列表
    End If
End With
With wb.AddButtonGroup("form1","btg2",False)
    .Add("btn4", "生成Excel格式订单", "button","order.xls?oid=" & pr("订单编号"))
End With
pr.Save() '必须保存,而且必须在最后保存,因为SQLAddNew增加的行,保存之后就会销毁,无法再调用
If e.PostValues.Count > 0 Then '如果是通过提交按钮访问,则给一个已经保存的提示给用户,时长500毫秒
    wb.AddToast("","t1", "已经保存",0).Msec= 500
End If
wb.AppendHTML("<script src='./lib/order.js'></script>") '引入脚本文件
e.WriteString(wb.Build) '生成网页
```

8. 客户端类

8.1 HttpClient

B/S 程序在一定程度上，也可以看作是 C/S 程序，只是此时客户端是一个浏览器而已。
和 HttpServer 配合的客户端并非一定要浏览器，也可以是其它程序。
我们可以在其它程序中编码模仿浏览器，向 HttpServer 发送访问请求，模拟提交 Cookie 和表单数据，以及上传文件，并接收从 HttpServer 返回的 Cookie、数据 和文件。
如果你不是专业程序员，完成上述任务会有困难，所以我们为此提供了一个 HttpClient 类。
HttpClient 可以完成浏览器所有的工作，只不过这一切都是通过编码进行的，而且是非常简单的代码。
在线用户数控制、独占式编辑、流程管理和不重复编号的生成等等，我们以前都是通过 OpenQQ 来实现，由于 OpenQQ 是封闭的，且有严谨的控制逻辑，所以使用起来会繁琐一些。
所有这些都可以通过 HttpClient 实现，而且更简单，这是因为 HttpClient 和 HttpServer 是开放的，某种程度也可以认为是随意的 ，你甚至可以用很简单的代码完成一个数据中间件的编写工作。
我并不打算将这些经典的示例用 HttpServer+HttpClient 重写一次，因为原理和逻辑都是一样的。
重要提示
1、HttpServer 和 HttpClient 必须分别运行在不同的项目中，否则程序将会锁死.
2、如果客户端项目和服务端项目在同一台电脑运行，二者都必须以 Windows 管理员身份运行，否则客户端项目将无法正常编译代码。
3、切记以上两点。

8.2 接收数据

HttpClient 的使用非常简单，只需两行代码即可从服务器获取数据。
一个例子
1、在服务端项目的 HttpRequest 事件中设置代码：

```
Select Case e.Path
    Case "gettime.htm"
        e.WriteString(Date.now)
End Select
```

2、在客户端项目通过下面的代码，即可获取服务器时间：

```
Dim hc As New HttpClient("http://127.0.0.1/gettime.htm")
Dim dt As Date = hc.GetData()
MessageBox.Show(dt)
```

过程很简单，首先是新建一个HttpClient，新建HttpClient的语法为：

```
Dim 变量名 As New HttpClient(网址)
```

然后通过HtpClient的GetData方法将访问请求提交到服务器，并从服务器获取返回的数据。

再次提示：服务端和客户端必须分开两个项目运行，否则Foxtable将被锁死。

8.3 发送数据

HttpClient 可以直接将字符型数据发送到服务器，只需将其 Content 属性设为要发送的数据即可。

一个例子

1、在服务端项目的 HttpRequest 事件中设置代码：

```
Select Case e.Path
    Case "test.htm"
        If e.PlainText > "" Then
            e.WriteString("服务器收到以下数据:" & vbcrlf & e.PlainText)
        Else
            e.WriteString("服务器未收到数据")
        End If
End Select
```

2、在客户端项目的命令窗口执行下面的代码，会提示服务器收到的数据：

```
Dim hc As New HttpClient("http://127.0.0.1/test.htm")
hc.Timeout = 5 '设置超时为5秒
hc.Content = "我是Foxtable"
Dim ret As String = hc.GetData()
MessageBox.Show(ret)
```

提示HttpRequest会将收到的纯文本数据保存在e参数PlainText中。

8.4 收发 Cookie

HttpClient 有两个和 Cookie 相关的字典属性，分别是：

- sCookies
  要发送到服务器的所有 Cookie。

- rCookies
  从服务器返回的所有 Cookie。

一个例子

1、在服务端项目的 HttpRequest 事件中设置代码：

```
Dim Verified As Boolean
If e.Cookies.ContainsKey("username") AndAlso e.Cookies.ContainsKey("password") Then
    '实际开发的时候,请改为根据用户表验证身份
    Dim username As String   = e.Cookies("username")
    Dim password As String   = e.Cookies("password")
    If username = "张三" AndAlso password = "888" Then
        Verified   = True
    End If
End If
If Verified = False Then
    e.AppendCookie("error","用户身份验证失败!") '通过Cookie返回错误信息.
    Return
End If
Select Case e.Path
    Case "gettime.htm"
        e.WriteString(Date.now)
End Select
```

2、在客户端项目的命令窗口运行以下代码，可以获取服务器时间：

```
Dim hc As New HttpClient("http://127.0.0.1/gettime.htm")
hc.sCookies.Add("username","张三")
hc.sCookies.Add("password","888")
Dim val As String = hc.GetData()
If hc.rCookies.ContainsKey("error") Then
    MessageBox.show(hc.RCookies("error"), "错误",  MessageBoxButtons.OK, MessageBoxIcon.Warning)
Else
    Dim dt As Date = val
    MessageBox.Show(dt)
End If
```

上述代码中，用户名和密码是通过Cookie提交到服务器的。
如果修改代码中的用户名或密码，执行后会提示“用户身份验证失败”，这个提示是服务器通过Cookie返回的。

8.5 模拟表单

HttpClient 有个 FormData 属性，这是一个字典，用于添加需要向服务器提交的表单数据。

一个例子

1、服务端项目的 HttpRequest 事件代码:

```
Select Case e.Path
    Case "AddNew.htm"
        Dim nms() As String = {"产品","客户","数量","单价","日期"}
        For  Each nm As String In nms
            If e.PostValues.ContainsKey(nm) = False Then
                e.AppendCookie("error","请输入" & nm)
                Return
            End If
        Next
        Dim dr As DataRow = DataTables("订单").AddNew()
        For  Each nm As String In nms
            dr(nm) = e.PostValues(nm)
        Next
        dr.Save()
        e.WriteString(dr("_Identify")) '返回主键列的值给客户端
        dr.Remove() '移除此行
End Select
```

2、客户端项目可以通过以下代码增加订单，并获得新增订单的主键：

```
Dim hc As New HttpClient("http://127.0.0.1/AddNew.htm")
hc.FormData.Add("产品","PD01")
hc.FormData.Add("客户","CS01")
hc.FormData.Add("数量","100")
hc.FormData.Add("单价","98")
hc.FormData.Add("日期",Date.Today)
Dim ret As String = hc.getData()
If hc.rCookies.ContainsKey("error") Then
    MessageBox.Show(hc.rCookies("error"), "提示", MessageBoxButtons.OK ,MessageBoxIcon.Warning)
Else
    MessageBox.Show("增加行成功,新增行的主键为:" & ret)
End If
```

8.6 接收文件

HttpClient 的 GetFile 方法用于接收从服务器返回的文件，语法为：
GetFile 的语法为：
GetFile(FielName)
FileName：将接收到的文件保存在本地的文件名，含路径。
如果成功接收并保存文件，GetFile 会返回 True，否则返回 False。
一个例子
1、服务端项目的 HttpRequest 事件代码：

```
Select Case e.Path
    Case "download.htm"
        e.WriteFile("d:\web\abc.jpg")
End Select
```

2、客户端项目可以通过以下代码下载文件：

```
Dim hc As New HttpClient("http://127.0.0.1/download.htm")
If hc.GetFile("c:\data\123.jpg") Then
    MessageBox.show("文件下载成功","提示")
Else
    MessageBox.show("文件下载失败","提示")
End If
```

下载的文件会保存为 c:\data\123.jpg

8.7 上传文件

通过 HttpClient 上传文件非常简单，只需将要上传的文件添加到 HttpClient 的 Files 集合即可。
向 Files 添加上传文件的语法为：
Add(ID，FileName)

ID	文件上传组件的 ID。
FileName	要上传的文件，含路径，一个 ID 下可以添加多个文件。

关于 HttpServer 是如何接收和保存客户端上传文件的，请参考：文件的上传与接收
一个例子
1、服务端项目的 HttpRequest 事件代码：

```
Dim Verified As Boolean
If e.PostValues.ContainsKey("username") AndAlso e.PostValues.ContainsKey("password") Then
    '实际开发的时候,请改为根据用户表验证身份
```

```
    Dim username As String   = e.PostValues("username")
    Dim password As String   = e.PostValues("password")
    If username = "张三" AndAlso password = "888" Then
        Verified   = True
    End If
End If
If Verified = False Then
    e.AppendCookie("error","用户身份验证失败!") '通过 Cookie 返回错误信息.
    Return
End If
Select Case e.Path
    Case "upload.htm"
        Dim sb As New StringBuilder
        sb.AppendLine("服务器收到的文件有:")
        For Each key As String In e.Files.Keys
            sb.AppendLine(Key)
            For Each fl As String In e.Files(key)
                sb.AppendLine("    " & fl)
                e.SaveFile(key,fl,"d:\web\uploadfiles\" & fl) '保存接收到的文件
            Next
        Next
        e.WriteString(sb.ToString)
End Select
```

2、客户端可以通过以下代码上传文件：

```
Dim hc As New HttpClient("http://127.0.0.1/upload.htm")
hc.FormData.Add("username","张三")
hc.FormData.Add("password","888")
hc.Files.Add("up1","c:\data\abc.jpg")
hc.Files.Add("up1","c:\data\订单.xls")
hc.Files.Add("up2","c:\data\附件.zip")
Dim ret As String = hc.getData()
If hc.rCookies.ContainsKey("error") Then
    MessageBox.Show(hc.rCookies("error"))
Else
    MessageBox.Show(ret)
End If
```

提示：本示例的身份信息通过表单传递，而不是通过Cookie，使用哪种方式都可以，记得实际开发的时候，不要使用username和password这样指示性很强的名称。

8.8 接收 DataTable

HttpRequest 事件有个 WriteDataTable 方法，用于向客户端发送 DataTable，语法为：
WriteDataTable(DataTable)

或
WriteDataTable(SelectString, ConnectionName)

DataTable	要发送的一个现有 DataTable
SelectString	SelectString 语句，据此生成一个 DataTable 发送给客户端。
ConnectionName	数据源名称

对应的 HttpClient 有个 GetDataTable 方法，用于接收服务端发来的 DataTable。

一个例子

1、服务端项目的 HttpRequest 事件代码：

```
Dim Verified As Boolean
If e.PostValues.ContainsKey("username") AndAlso e.PostValues.ContainsKey("password") Then
    '实际开发的时候,请改为根据用户表验证身份
    Dim username As String  = e.PostValues("username")
    Dim password As String  = e.PostValues("password")
    If username = "EP01" AndAlso password = "888" Then
        Verified  = True
    End If
End If
If Verified = False Then
    e.AppendCookie("error","用户身份验证失败!") '通过Cookie返回错误信息.
    Return
End If
Select Case e.Path
    Case "query.htm"
        Dim scd As String  = "select Top 100 * from 订单 Where 雇员 = '" & e.PostValues("username")
& "'"
        If e.PostValues.ContainsKey("filter") Then
            scd = scd & " and (" & e.PostValues("filter") & ")"
        End If
        e.WriteDataTable(scd, "数据源名称")
End Select
```

重要提示：
在合成表达式的时候，必要条件(只加载登录用户负责的的订单，这里是：雇员='EP01')和查询条件，必须按照以下格式合并：
必要条件 And (查询条件)
如果查询条件没有用括号括起来，那么用户可以很轻松地突破权限设置，获取所有数据。

2、在客户端项目新建一个窗口，插入一个 Table 和按钮，按钮的 Click 事件代码设置为：

```
Dim hc As New HttpClient("http://127.0.0.1/query.htm")
hc.FormData.Add("username","EP01")
hc.FormData.Add("password","888")
hc.FormData.Add("filter","产品='PD01'")
Dim dt As DataTable = hc.GetDataTable()
If hc.rCookies.ContainsKey("error") Then
    MessageBox.show(hc.rCookies("error"), "错误" ,MessageBoxButtons.OK, MessageBoxIcon.Warning)
ElseIf dt IsNot Nothing Then
    Tables("窗口1_Table1").DataSource = dt
Else
        MessageBox.Show("接收DataTable失败，原因未知!", "错误" ,MessageBoxButtons.OK ,MessageBoxIcon.Warning)
End If
```

打开窗口，单击按钮，Table 控件即可显示从服务端获取的数据。

8.9 获取内容信息

我们可以获取内容信息，例如最近一次修改时间、内容类型、内容长度、状态码、状态字符串和头部信息等。
例如：

```
Dim htc As new HttpClient("http://www.foxtable.com/bbs/images/foxtable.gif")
htc.GetFile("c:\aaa\test.gif")
OutPut.Show("更新日期:" & htc.LastModified)
OutPut.Show("状态码:" & htc.StatusCode)
OutPut.Show("状态描述:" & htc.StatusDescription)
OutPut.Show("内容类型:" & htc.ResponseContentType)
OutPut.Show("内容长度:" & htc.ResponseContentLength)
For Each key As String  In htc.ResponseHeaders.Keys  '显示服务器返回的头部信息
    OutPut.Show(Key & ":" & htc.ResponseHeaders(key))
Next
```

第三方开发接口示例

下面是进行企业微信开发的时候，一段下载临时素材的代码：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/media/get?access_token={0}&media_id={1}"
Dim mediaID As String = "1GDEyyL7RIo868rKmIf3ThZoCMkd69VvH3wPBMpQj9w3uq7dmdxlbNjh669KPLyhF7EEOpBszls7BSmgMFgn07A"  '要获取素材的ID
```

```
Dim hc As new HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken"),mediaID))
Dim fl As String  = "c:\data\abc.jpg" '要保存为的本地文件
If hc.GetFile(fl) Then
    If hc.ResponseContentType.StartsWith("application/json") Then
        MessageBox.Show(Filesys.ReadAllText(fl))
        FileSys.DeleteFile(fl) '删除文件
    Else
        MessageBox.Show("图片素材下载成功!")
    End If
Else
    MessageBox.Show("图片素材下载失败!")
End If
```

下载临时素材,会有两种可能：

1、获取成功，返回文件。
2、获取失败，返回JSON格式的错误信息。

但是HttpClient的GetFile会直接将返回内容保存为文件，不管是返回的内容是素材文件还是错误信息。
我们可以判断HttpClient的ResponseContentType属性，如果其值为"application/json"，说明返回的不是素材文件，而是错误提示信息，然后从此文件读取错误信息显示给用户，最后删除文件。

接下来你会看到，我们还有更好的办法应付此类场景。

8.10 分两步获取信息和内容

HttpClient 有个 GetInformation 方法，用于获取内容信息，获取成功则返回 True，否则返回 False。

此方法执行之后并不会关闭和服务端的连接，我们可以继续用 GetFile、GetData 等方法获取内容本身。

有了这个方法，我们可以首先获取内容信息，然后根据内容信息进行下一步的操作，而无需重新连接服务器。
示例一
服务端的 HttpRequest 事件代码为：

```
Select Case e.Path
    Case "test.jpg"
        If Date.Now.Hour < 10 Then
            e.ResponseHeaders("Error") = "001"
            e.WriteString("文件还未准备好,请10点之后再下载!")
```

```
    Else
      e.WriteFile("d:\web\test.jpg")
    End If
End Select
```

如果客户在上午 10 点之前访问，则在头部信息中加上错误标记，并返回文本形式的错误信息。

客户端代码：

```
Dim htc As new HttpClient("http://127.0.0.1/test.jpg")
If  htc.GetInformation() Then
  Dim err As String  = htc.ResponseHeaders("Error")
  If  err> "" Then
    MessageBox.Show(htc.GetData() ,"错误", MessageBoxButtons.OK, MessageBoxIcon.Error)
  Else
    htc.GetFile("c:\data\test.jpg")
    MessageBox.Show("文件下载成功", "提示", MessageBoxButtons.OK, MessageBoxIcon.Information)
  End If
Else
    MessageBox.Show("文件下载失败", "提示", MessageBoxButtons.OK,MessageBoxIcon.Error)
End If
```

示例二

服务端可以只发送内容信息，而不发送内容，例如服务端的 HttpRequest 事件代码为：

```
Select Case e.Path
  Case "test.jpg"
    If Date.Now.Hour < 10 Then
      e.ResponseHeaders("Error") = "The file is not ready"
      e.Handled = True
    Else
      e.WriteFile("d:\web\test.jpg")
    End If
End Select
```

如果客户在上午 10 点之前访问，则通过头部信息返回错误"The file is not ready!"

重要提示：

1、如果不打算发送内容本身，只发送内容信息，那么务必将 e 参数 Handled 设置为 True，否则客户端将持续等待，直到超时。
2、头部信息不支持中文，解决方法后面会介绍。
客户端代码：

```
Dim htc As new HttpClient("http://127.0.0.1/test.jpg")
If  htc.GetInformation() Then
    Dim err As String  = htc.ResponseHeaders("Error")
    If  err> "" Then
       MessageBox.Show(err,"错误", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
       htc.GetFile("c:\data\test.jpg")
         MessageBox.Show(" 文 件 下 载 成 功 ", " 提 示 ",  MessageBoxButtons.OK,
MessageBoxIcon.Information)
    End If
Else
       MessageBox.Show("文件下载失败", "提示", MessageBoxButtons.OK,MessageBoxIcon.Error)
End If
```

第三方开发接口示例

有了 GetInformation 方法，我们进行第三方接口开发的时候，会更加方便，还是以上一节企业微信下载临时素材为例，改进后的代码为：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/media/get?access_token={0}&media_id={1}"
Dim mediaID As String =
"1GDEyyL7RIo868rKmIf3ThZoCMkd69VvH3wPBMpQj9w3uq7dmdxlbNjh669KPLyhF7EEOpBszls7BSmgM
Fgn07A"  '要获取素材的ID
Dim hc As new HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken"),mediaID))
Dim fl As String  = "c:\data\abc.jpg" '要保存为的本地文件
If hc.GetInformation() Then
    If hc.ResponseContentType.StartsWith("application/json") Then
       MessageBox.Show(hc.GetData())
    Else
       hc.GetFile(fl)
       MessageBox.Show("图片素材下载成功!")
    End If
Else
    MessageBox.Show("图片素材下载失败!")
End If
```

8.11 单纯获取内容信息

如果只需获取内容信息，而不需要获取内容本身，可以参考下面的代码。

例如：

```
Dim htc As new HttpClient("http://www.foxtable.com/bbs/images/foxtable.gif")
htc.GetInformation()
htc.Close() '必须执行 Close 方法关闭连接
OutPut.Show("更新日期:" & htc.LastModified)
OutPut.Show("状态码:" & htc.StatusCode)
OutPut.Show("状态描述:" & htc.StatusDescription)
OutPut.Show("内容类型:" & htc.ResponseContentType)
OutPut.Show("内容长度:" & htc.ResponseContentLength)
For Each key As String  In htc.ResponseHeaders.Keys  '显示服务器返回的头部信息
    OutPut.Show(Key & ":" & htc.ResponseHeaders(key))
Next
```

重要提示：

如果不获取内容本身，那么一定要执行 Close 方法关闭 HttpClient，否则和服务器之间的链接将一直存在，直到超时，导致服务器资源得不到及时释放。

8.12 再谈头部信息

我们已经知道，Http 服务器和客户端之间可以通过 Post、Get、Cookies 传递信息，其中 Post 和 Get 都是单向的，Cookies 是双向的。

前面两节，我们用了一个新的数据传递方法：通过头部(Headers)传递信息。

Headers 一样可以双向传递，不同的是 Cookies 可以存储在客户端供下次访问使用，而 Headers 只能是临时的。
此外 Headers 不支持中文，如果需要用中文，可用 UrlEncode 和 UrlDecode 函数进行编码和解码。

你可能已经知道，服务端和客户端之间默认已经有不少头部信息，例如传递客户端的浏览器类型、服务器类型、HTTP 协议版本等等。

服务端的 HttpRequest 事件有两个和头部信息相关的属性：

- Headers
  用于获取客户端发来的头部信息。

- ResponseHeaders
  用于设置向客户端发送的头部信息。

客户端类 HttpClient 有两个和头部信息相关的属性：

- Headers
  用于设置向服务端发送的头部信息。

- ResponseHeaders
  用于获取服务端发来的头部信息。

在针对一些第三方 API 接口进行开发时，可能需要使用头部信息，参考：阿里云全国物流查询接口。

示例

服务端的 HttpRrquest 事件代码

```
Select Case e.Path
    Case "test.jpg"
        '获取客户端发送来的头部信息
        Dim author As String   =   UrlDecode(e.Headers("author"))
        Dim age As String   = e.Headers("age")
        '向客户端发送头部信息
        e.ResponseHeaders.Add("brief",URLEncode("你不是因为厉害才开始,你是因为开始才厉害"))
        e.ResponseHeaders.Add("pages",600)
        '发送实际内容
        e.WriteFile("d:\web\test.jpg")
End Select
```

客户端代码：

```
Dim htc As new HttpClient("http://127.0.0.1/test.jpg")
htc.Headers.Add("author", UrlEncode("舒淇")) '向服务端附加头部信息
htc.Headers.Add("age", 30)
If   htc.GetInformation() Then
    '获取服务端返回的头部信息
    Dim brief As String    = UrlDecode(htc.ResponseHeaders("brief"))
    Dim pages As Integer = htc.ResponseHeaders("pages")
    '获取实际内容
    htc.GetFile("c:\data\test.jpg")
        MessageBox.Show("文件下载成功", "提示", MessageBoxButtons.OK,
MessageBoxIcon.Information)
Else
    MessageBox.Show("文件下载失败", "提示", MessageBoxButtons.OK,MessageBoxIcon.Error)
End If
```

8.13 一个经典示例

实现网络环境下实现复杂编号，一直是个不轻松的任务，Foxtable 2017 之前的实现方法，你可以参考：

网络环境下的复杂编号

用 OpenQQ 实现网络环境下的编号
可以看到，即使有了 OpenQQ，完成这个任务还是比较麻烦。

我们现在改用 HttpServer+HttpClient 来实现，你会发现代码简单了很多。

我们用其中最复杂的一个例子为例讲述。

假定有个名为"工程"的表，需要按月自动生成编号，根据工程代码按顺序编号，前 4 位是工程代码，然后是 4 位年，2 位月，最后 4 位是顺序号，如下图所示：

	工程代码	制单日期	单据编号
1	NGAY	2012-03-02	NGAY-201203-0001
2	NGAY	2012-03-03	NGAY-201203-0002
3	NGAY	2012-03-04	NGAY-201203-0003
4	NGAY	2012-03-05	NGAY-201203-0004
5	NGAY	2012-03-06	NGAY-201203-0005
6	NGAY	2012-03-07	NGAY-201203-0006
7	NGAY	2012-03-08	NGAY-201203-0007
8	NGAY	2012-03-09	NGAY-201203-0008
9	NGAY	2012-03-01	NGAY-201203-0009
10	QYGC	2012-03-02	QYGC-201203-0001
11	QYGC	2012-03-03	QYGC-201203-0002
12	QYGC	2012-03-04	QYGC-201203-0003
13	QYGC	2012-03-05	QYGC-201203-0004
14	QYGC	2012-03-06	QYGC-201203-0005
15	QYGC	2012-04-14	QYGC-201204-0001
16	QYGC	2012-04-14	QYGC-201204-0002
17	QYGC	2012-04-14	QYGC-201204-0003

重要提示：客户端和服务端必须分开两个项目运行，否则会锁死。
设计步骤：
1、服务端项目的 HttpRequest 事件代码：

```
Static bhs As new Dictionary(of String,Integer) '这个字典用于记录每个前缀的下一序号
If  e.Path =  "bianhao.htm"
   If e.PostValues.ContainsKey("制单日期") AndAlso e.PostValues.ContainsKey("工程代码") Then
      Dim max As Integer
      Dim bh As String
```

```
        Dim prefix As String = e.PostValues("工程代码") & "-" &  Format(CDate(e.PostValues("制单日期")), "yyyyMM")
        If bhs.ContainsKey(prefix) = False Then '如果字典中不存在次前缀的序号,
            Dim cmd As New SQLCommand
            'cmd.ConnectionName = "数据源名称" '外部表请设置数据源名称'
            cmd.CommandText = "Select max(单据编号) From {工程} Where  单据编号  Like '" & prefix & "%'"
            bh = cmd.ExecuteScalar
            If bh IsNot Nothing Then
                bh = bh.SubString(12)
                Integer.TryParse(bh,max)
            End If
            bhs.Add(prefix,max)
        Else
            max = bhs(prefix)
        End If
        max = max + 1
        bhs(prefix) = max
        e.WriteString(Prefix & "-" & Format(max,"0000"))
    End If
End If
```

2、在客户端项目工程表的 DataColChanged 事件加上代码：

```
Select Case e.DataCol.name
    Case "工程代码","制单日期"
        Dim dr As DataRow = e.DataRow
        If dr.IsNull("工程代码") = False AndAlso dr.IsNull("制单日期") = False Then
            Dim hc As new HttpClient("http://127.0.0.1/bianhao.htm")
            hc.FormData.Add("工程代码",dr("工程代码"))
            hc.FormData.Add("制单日期",dr("制单日期"))
            dr("单据编号") = hc.GetData()
        End If
End Select
```

9. JSON 解析

9.1 解析 JSON

JSON 的解析，Foxtable 采用的是第三方库 Newtonsoft.JSON。

Foxtable 已经引入了以下两个命名空间：

Newtonsoft.Json
Newtonsoft.Json.Linq

你可以直接在代码中使用 Newtonsoft.JSON 进行 JSON 的系列化和反系列化。

如果已经熟悉了 Newtonsoft.JSON，可以跳过本节内容。

示例一

简单JSON对象的解析和注意事项。

在命令窗口执行：

```
Dim json As String = "{'name':'李云龙','age':'36'}"
Dim jo As JObject = JObject.Parse(json)
Output.Show(jo("name"))
Output.Show(jo("age"))
```

显示的结果是：

```
李云龙
36
```

JSON对象成员可以直接转换为相应的数据类型，例如：

```
Dim json As String = "{'name':'李云龙','age':'36'}"
Dim jo As JObject = JObject.Parse(json)
Dim Name As String  = jo("name")
Dim Age As Integer = jo("age")
```

但是有一个很奇怪的现象，直接将JSON成员和字符串组合会出错，例如以下代码是无法执行的的：

```
Dim json As String = "{'name':'李云龙','age':'36'}"
Dim jo As JObject = JObject.Parse(json)
Output.Show("姓名:" & jo("name") & "  年龄:" & jo("age"))
```

必须改为:

```
Dim json As String = "{'name':'李云龙','age':'36'}"
Dim jo As JObject = JObject.Parse(json)
Output.Show("姓名:" & jo("name").ToString() & "  年龄:" & jo("age").ToString())
```

此外JSON对象是区分大小写的，所以上述代码中，如果你改用JO("Name")是取不到值的，只能是jo("name")。

实际上你用JO("Name")取值，并不会出错，而是返回一个空值(Nothing)，我们可以用这个特性判断JSON对象是否包括指定名称的成员，例如：

```
Dim json As String = "{'name':'李云龙','age':'36'}"
Dim jo As JObject = JObject.Parse(json)
Output.show( jo("Name") Is Nothing )
Output.show( jo("Test") Is Nothing )
```

显示的结果是：

```
True
True
```

示例二

解析嵌套对象。

在命令窗口执行：

```
Dim json As String = "{'name':'李云龙','age':'36','card':{'bank':'工行','account':'123456'}}"
Dim jo As JObject = JObject.Parse(json)
Output.Show(jo("name"))
Output.Show(jo("age"))
Output.Show(jo("card")("bank"))
Output.Show(jo("card")("account"))
```

上面的JSON对象嵌套了一个子对象card，这个子对象有自己的属性(bank和account)。

显示的结果为：

```
李云龙
36
工行
123456
```

示例三

JSON 值数组的解析。

在命令窗口执行：

```
Dim json As String = "['张三','李四','王五']"
Dim ja As JArray = JArray.Parse(json)
For i As Integer = 0 To ja.Count - 1
```

```
    output.show(ja(i))
Next
```

显示的结果是：

```
张三
李四
王五
```

JSON 数组 的对象为 JArray，数组成员的类型为 JToken，所以也可以用下面的代码遍历 JSON 数组:

```
Dim json As String = "['张三','李四','王五']"
For Each v As JToken In JArray.Parse(json)
    Output.show(v)
Next
```

示例四

JSON对象中的值数组的解析。

在命令窗口执行：

```
Dim json As String   = "{'touser':['zhansan', 'lisi'],'msgtype':'text'}"
Dim jo As JObject = JObject.Parse(json)
Dim ja As JArray = jo("touser")
For i As Integer = 0 To ja.Count - 1
   Output.Show(ja(i).ToString)
Next
Output.Show(jo("msgtype").Tostring)
```

上面的JSON对象中，touser属性是一个值数组，这个数组包括两个成员(zhansan和lisi)。

执行后显示的结果为：

```
zhansan
lisi
text
```

同样也可以通过JToken来遍历，得到的结果是一样的：

```
Dim json As String   = "{'touser':['zhansan', 'lisi'],'msgtype':'text'}"
Dim jo As JObject = JObject.Parse(json)
For Each jt As JToken In jo("touser")
   Output.Show(jt)
Next
```

```
Output.Show(jo("msgtype").Tostring)
```

示例五

JSON对象数组的解析。

在命令窗口执行：

```
Dim json As String = "[{'name':'李云龙','age':'36'},{'name':'黄晓明','age':'28'}]"
Dim ja As JArray = Jarray.Parse(json)
For i As Integer = 0 To ja.Count - 1
    OutPut.Show("姓名:"  & ja(i)("name").ToString() &  "  年龄:" & ja(i)("age").ToString())
Next
```

或者：

```
Dim json As String = "[{'name':'李云龙','age':'36'},{'name':'黄晓明','age':'28'}]"
For Each ep As JToken In JArray.Parse(json)
    Output.Show("姓名:"  & ep("name").ToString() &  "  年龄:" & ep("age").ToString())
Next
```

显示的结果为：

```
姓名:李云龙  年龄:36
姓名:黄晓明  年龄:28
```

示例六

嵌套的对象数组解析。

在命令窗口执行：

```
Dim json As String = "{'name':'李云龙','card':[{'bank':'工行','account':'123'},{'bank':'建行','account':'678'}]}"
Dim jo As JObject = JObject.Parse(json)
Output.Show(jo("name"))
For Each jt As JToken In jo("card")
    Output.Show(jt("bank").ToString & "|" & jt("account").ToString)
Next
```

上面的 JSON 对象，其 card 属性是一个对象数组，包括两个银行卡信息。

显示的结果为：

```
李云龙
```

工行|123
建行|678

9.2 解析实例

下面是阿里云全国快递查询接口返回的数据格式，顶层对象包括的属性非常多，其属性 showapi_res_body 是一个嵌套对象，这个嵌套对象本身也有很多属性， 其中的 data 属性是一个对象数组，这个数组的每个成员包括 time 和 context 属性：


```
{
  "showapi_res_code": 0,
  "showapi_res_error": "",
  "showapi_res_body": {
    "mailNo": "929601675231",
    "update": 1488784549365,
    "updateStr": "2017-03-06 15:15:49",
    "ret_code": 0,
    "flag": true,
    "status": 4,
    "tel": "95338",
    "expSpellName": "shunfeng",
    "data": [
      {
        "time": "2017-03-03 08:42:59",
        "context": "已签收,感谢使用顺丰,期待再次为您服务"
      },
      {
        "time": "2017-03-03 07:40:22",
        "context": "正在派送途中,请您准备签收(派件人:李正国,电话:18907153726)"
      },
      {
        "time": "2017-03-03 06:08:45",
        "context": "快件到达 【武汉硚口区复兴村营业点】"
      },
      {
        "time": "2017-03-03 04:48:26",
        "context": "快件在【武汉吴家山集散中心】已装车，准备发往 【武汉硚口区复兴村营业点】"
      },
      {
        "time": "2017-03-03 00:07:27",
        "context": "快件到达 【武汉吴家山集散中心】"
      },
```

```
    {
     "time": "2017-03-02 22:54:14",
     "context": "快件在【武汉总集散中心】已装车，准备发往 【武汉吴家山集散中心】"
    },
    {
     "time": "2017-03-02 22:54:02",
     "context": "快件到达 【武汉总集散中心】"
    },
    {
     "time": "2017-03-02 15:00:33",
     "context": "快件在【深圳总集散中心】已装车，准备发往 【武汉总集散中心】"
    },
    {
     "time": "2017-03-02 14:47:39",
     "context": "快件到达 【深圳总集散中心】"
    },
    {
     "time": "2017-03-02 10:24:37",
     "context": "快件在【江门江海集散中心】已装车，准备发往下一站"
    },
    {
     "time": "2017-03-02 05:41:54",
     "context": "快件到达 【江门江海集散中心】"
    },
    {
     "time": "2017-03-01 22:31:24",
     "context": "快件在【湛江麻章集散中心】已装车，准备发往 【江门江海集散中心】"
    },
    {
     "time": "2017-03-01 22:24:37",
     "context": "快件到达 【湛江麻章集散中心】"
    },
    {
     "time": "2017-03-01 20:02:01",
     "context": "快件在【湛江市赤坎文保北村营业点】已装车，准备发往 【湛江麻章集散中心】"
    },
    {
     "time": "2017-03-01 19:49:28",
     "context": "顺丰速运 已收取快件"
    }
   ],
   "expTextName": "顺丰速运"
  }
```

```
}
```

首先将上面的 JSON 数据复制到剪贴版，然后在命令窗口执行下面的代码：

```
Dim jo As JObject = Jobject.Parse(ClipBoard.GetText)
If jo("showapi_res_body")("data") IsNot Nothing Then
    For Each jt As JToken In jo("showapi_res_body")("data")
        Output.Show(jt("time").ToString & " | " & jt("context").ToString)
    Next
End If
```

可以得到解析结果：

```
2017-03-03 08:42:59 | 已签收,感谢使用顺丰,期待再次为您服务
2017-03-03 07:40:22 | 正在派送途中,请您准备签收(派件人:李正国,电话:18907153726)
2017-03-03 06:08:45 | 快件到达 【武汉硚口区复兴村营业点】
2017-03-03 04:48:26 | 快件在【武汉吴家山集散中心】已装车，准备发往 【武汉硚口区复兴村营业点】
2017-03-03 00:07:27 | 快件到达 【武汉吴家山集散中心】
2017-03-02 22:54:14 | 快件在【武汉总集散中心】已装车，准备发往 【武汉吴家山集散中心】
2017-03-02 22:54:02 | 快件到达 【武汉总集散中心】
2017-03-02 15:00:33 | 快件在【深圳总集散中心】已装车，准备发往 【武汉总集散中心】
2017-03-02 14:47:39 | 快件到达 【深圳总集散中心】
2017-03-02 10:24:37 | 快件在【江门江海集散中心】已装车，准备发往下一站
2017-03-02 05:41:54 | 快件到达 【江门江海集散中心】
2017-03-01 22:31:24 | 快件在【湛江麻章集散中心】已装车，准备发往 【江门江海集散中心】
2017-03-01 22:24:37 | 快件到达 【湛江麻章集散中心】
2017-03-01 20:02:01 | 快件在【湛江市赤坎文保北村营业点】已装车，准备发往 【湛江麻章集散中心】
2017-03-01 19:49:28 | 顺丰速运 已收取快件
```

可以看到，用 JObject 解析 JSON 数据是异常方便的。

9.3 生成 JSON

我们当然可以手工编码生成 JSON，但是比较繁琐，容易出错。

如果采用 Newtonsoft.JSON 生成，会变得简单可靠。

Foxtable 已经引入了以下两个命名空间：

Newtonsoft.Json
Newtonsoft.Json.Linq

你可以直接在代码中使用 Newtonsoft.JSON 进行 JSON 的系列化和反系列化。

如果已经熟悉了 Newtonsoft.JSON，可以跳过本节内容。

示例一

根据现有对象生成 JSON。

例如根据当前登录用户生成 JSON，可在命令窗口执行：

```
Dim jo As JObject = JObject.FromObject(User)
Dim js As String = jo.ToString()
Output.Show(js)
```

显示的结果为：

```
{
"Name": "张三",
"Group": null,
"Type": 2,
"Tag": null,
"Roles": "经理",
"Default": false,
"ExtendedValues": {}
}
```

示例二

动态生成 JSON。

在命令窗口执行：

```
Dim jo As New JObject
jo("Name") = "张三"
jo("Group") = "VIP"
jo("Type") = 2
jo("Default") = False
Output.Show(jo.ToString)
```

显示的结果为：

```
{
"Name": "张三",
"Group": "VIP",
"Type": 2,
"Default": false
}
```

示例三

生成包括嵌套对象的JSON。

在命令窗口执行：

```
Dim jo As New JObject
jo("name") = "李云龙"
jo("age") = "36"
jo("card") = New JObject
jo("card")("bank") = "工行"
jo("card")("account") = "12345678"
Output.Show(jo.ToString)
```

显示的结果为：

```
{
  "name": "李云龙",
  "age": "36",
  "card": {
    "bank": "工行",
    "account": "12345678"
  }
}
```

示例四

生成包括数组的JSON。

在命令窗口执行：

```
Dim jo As New JObject
jo("Name") = "张三"
jo("Group") = "VIP"
Dim ja As New Jarray '定义数组
ja.Add("manager")
ja.Add("developer")
jo("Roles") = ja '将Roles设置为前面定义的数组
Output.Show(jo.ToString)
```

显示的结果为：

```
{
  "Name": "张三",
  "Group": "VIP",
```

```
  "Roles": [
    "manager",
    "developer"
  ]
}
```

下面是一段完全等效的代码，希望大家体会，在实际开发过程中灵活运用：

```
Dim jo As New JObject
Dim values() As String = {"manager","developer"}
jo("Name") = "张三"
jo("Group") = "VIP"
jo("Roles") = New Jarray(values)
Output.Show(jo.ToString)
```

示例五

生成包括对象数组的JSON。

在命令窗口执行：

```
Dim jo As New JObject
Dim ja As New JArray
jo("dept") = "销售部"
jo("mpr") = "赵刚"
jo("staff") =   ja '
ja.Add(New JObject) '给数组添加两个对象成员
ja.Add(New JObject)
ja(0)("name") = "李云龙"
ja(0)("age") = 36
ja(1)("name") = "黄晓明"
ja(1)("age") = 26
Output.Show(jo.ToString)
```

显示的结果为：

```
{
  "dept": "销售部",
  "mpr": "赵刚",
  "staff": [
    {
      "name": "李云龙",
      "age": 36
    },
    {
```

```
    "name": "黄晓明",
    "age": 26
   }
  ]
}
```

对象数组也是一个数组，只不过这个数组的成员也是JObject，JObject的成员可以是数组，数组的成员可以是JObject，二者可以层层嵌套，生成任意复杂的JSON。

9.4 生成实例

在后面学习微信接口的时候，你会发现很多数据都需要用 JSON 格式提交到微信服务器。

例如下面是一个 JSON 格式的微信菜单数据，我们将这段数据提交到微信相关接口后，可以生成一个微信菜单。
这个菜单包括两个顶层按钮，第一个顶层按钮"今日歌曲"是一个普通按钮，第二个顶层按钮"功能"是一个菜单，这个菜单包括三个子菜单按钮，分别是"搜索"、"视频"和"赞一下我们"：

```
{
  "button": [
   {
    "type": "click",
    "name": "今日歌曲",
    "key": "V1001_TODAY_MUSIC"
   },
   {
    "name": "功能",
    "sub_button": [
     {
      "type": "view",
      "name": "搜索",
      "url": "http://www.soso.com/"
     },
     {
      "type": "view",
      "name": "视频",
      "url": "http://v.qq.com/"
     },
     {
      "type": "click",
      "name": "赞一下我们",
      "url": "V1001_GOOD"
     }
```

```
    ]
   }
  ]
}
```

我们可以用代码生成以上格式的数据，请务必仔细体会：

```
Dim mnu As New JObject '菜单对象
Dim button As New JArray
mnu("button") = button '菜单对象只有一个属性 button,这个是一个数组,包括所有顶层菜单按钮.
'增加第一个顶层按钮.
button.Add(New Jobject)
button(0)("type") = "click"
button(0)("name") = "今日歌曲"
button(0)("key") = "V1001_TODAY_MUSIC"
'增加第二个顶层按钮,这个按钮其实是一个菜单.
button.Add(New Jobject)
button(1)("name") = "功能"
button(1)("sub_button") = New Jarray '第二个顶层按钮的 sub_button 属性是一个数组,包括所有的子菜单按钮
Dim SubButton As JArray = button(1)("sub_button") '为方便后面的编码，将子菜单数组保存在变量 SubButton 中
'增加第一个子菜单按钮
SubButton.Add(New Jobject)
SubButton(0)("type") = "view"
SubButton(0)("name") = "搜索"
SubButton(0)("url") = "http://www.soso.com/"
'增加第二个子菜单按钮
SubButton.Add(New Jobject)
SubButton(1)("type") = "view"
SubButton(1)("name") = "视频"
SubButton(1)("url") = "http://v.qq.com/"
'增加第三个子菜单按钮
SubButton.Add(New Jobject)
SubButton(2)("type") = "click"
SubButton(2)("name") = "赞一下我们"
SubButton(2)("url") = "V1001_GOOD"
Output.Show(mnu.ToString) '生成 JSON 字符串
```

你也许觉得代码有点长，因为这是"硬"生成，但是自行用工具编写 JSON 数据非常繁琐，也很容易出错，多数时候我们会通过数据表输入菜单设置，然后用代码遍历数据表中的行自动生成 JSON 数据。

实际上我在编写上面这段实例代码时，已经考虑到你未来的需要，你可以很轻松地将上面这段代码改为根据数据表中的行遍历生成，这在《微信接口》这一章会有介绍。

9.5 压缩 JSON

CompressJson 是 Foxtable 提供的一个用于压缩 JSON 字符串的函数，该函数的参数可以是一个 JObject 或 JArray，也可以是一个 JSON 字符串。

我们知道 JObject 等的 ToString 方法，输出的 JSON 字符串是经过格式化的，存在空格缩进和换行，以方便阅读。

例如：

```
Dim json As String = "{'touser':['zhansan', 'lisi'],'msgtype':'text'}"
Dim jo As JObject = JObject.Parse(json)
Output.Show(jo.ToSTring)
```

输出结果为：

```
{
  "touser": [
    "zhansan",
    "lisi"
  ],
  "msgtype": "text"
}
```

如果用CompressJson函数输出，则会压缩掉多余的空格和换行，例如：

```
Dim json As String = "{'touser':['zhansan', 'lisi'],'msgtype':'text'}"
Dim jo As JObject = JObject.Parse(json)
Output.Show(CompressJson(jo))
```

输出结果为：

```
{"touser":["zhansan","lisi"],"msgtype":"text"}
```

如果你需要在数据库中存储JSON字符串，先用CompressJson压缩一下比较合适。

9.6 关于引号

JSON 可以用单引号，也可以用双引号，例如：

```
Dim json As String = "{'name':'李云龙','age':'36'}"
```

也可以写为：

```
Dim json As String = "{""name"":""李云龙"",""age"":""36""}"
```

为方便阅读，本文档很多地方采用了单引号编写 JSON。

但实际上双引号才是标准的写法，兼容性最好，例如微信使用的 JSON，就必须是双引号的。

所以自行编写 JSON 字符串的时候，最好全部使用双引号。

好消息是，Newtonsoft.JSON 生成的 JSON 都是双引号的，所以基本上我们无需为这个问题分心。

9.7 解析 XML

现在互联网的数据传递，JSON 的使用率已经远远超过 XML，我也推荐大家使用 JSON。

但是微信有点奇葩，有的接口是 XML，有的是 JSON，开发者需要处理两种类型的数据，而且有时还需要转换。

为了提高大家的学习和开发效率，我们增加了一个新的类型 XObject，让大家可以将 XML 数据直接作为 JSON 数据进行解析。

XObject 和 JObject 在数据解析方面是完全相同的，可以说唯一差别就是类名不同，几乎不需要额外的学习就能掌握。

如果你熟悉.net，可以直接用.net 的相关类库进行 XML 的解析和生成，也一样的简单，效率会更高一些。

至于普通用户，还是用 XObject 来得快捷方便。
由于 Foxtable 和 System.XML.Linq 命名空间都有名为 XObject 的类，实际使用时为避免冲突，必须加上命名空间 Foxtable，例如：
Dim xo As **Foxtable**.XObject

示例一
解析简单的 XML。
源 XML 数据结构：

```
<XML>
<to>George</to>
<from>John</from>
<content>Don't forget the meeting!</content>
</XML>
```

在命令窗口执行下面的代码：

```
Dim xml As String = "<xml><to>George</to><from>John</from><content>Don't forget the
```

```
meeting!</content></xml>"
Dim xo As Foxtable.XObject = Foxtable.XObject.Parse(xml)
Output.Show(xo("to"))
Output.Show(xo("from"))
Output.Show(xo("content"))
```

显示的结果为：

```
George
John
Don't forget the meeting!
```

可以看到解析XML和解析JSON的方法几乎是一样的。

示例二

嵌套 XML 的解析，源 XML 数据结构：

```
<XML>
<公司>宏兴贸易</公司>
<部门>
<名称>销售部</名称>
<员工>张三</员工>
<员工>李四</员工>
</部门>
<部门>
<名称>行政部</名称>
<员工>王五</员工>
<员工>赵六</员工>
</部门>
</XML>
```

在命令窗口执行以下代码：

```
Dim xml As String = "<xml><公司>宏兴贸易</公司><部门><名称>销售部</名称><员工>张三</员工><员工>李四</员工></部门>"
xml = xml & "<部门><名称>行政部</名称><员工>王五</员工><员工>赵六</员工></部门></xml>"
Dim xo As Foxtable.XObject = Foxtable.XObject.Parse(xml)
Output.Show(xo("公司"))
For Each bm As JToken In xo("部门")
    Output.show(bm("名称"))
    For Each yg As JToken In bm("员工")
        Output.Show(yg)
    Next
Next
```

显示的结果为：

```
宏兴贸易
销售部
张三
李四
行政部
王五
赵六
```

在解析XML的过程中，同一父节点之下的同名结点，会被解析为一个数组，为了验证这一点，可以在命令窗口执行：

```
Dim xml As String = "<xml><公司>宏兴贸易</公司><部门><名称>销售部</名称><员工>张三</员工><员工>李四</员工></部门>"
xml = xmL & "<部门><名称>行政部</名称><员工>王五</员工><员工>赵六</员工></部门></xml>"
Dim xo As Foxtable.XObject = Foxtable.XObject.Parse(xml)
Output.Show(xo.ToString())
```

显示的结果为：

```
{
  "公司": "宏兴贸易",
  "部门": [
    {
      "名称": "销售部",
      "员工": [
        "张三",
        "李四"
      ]
    },
    {
      "名称": "行政部",
      "员工": [
        "王五",
        "赵六"
      ]
    }
  ]
}
```

可以看到，部门和员工，都是以 JSON 数组的形式存在，其中部门是对象数组，员工是值数组。

通过上面的代码，大家也可以看到将 XML 数据转为 JSON 数据是多么的简单。

既然是数组，就可以用 JArray 进行解析：

```
Dim xml As String = "<xml><公司>宏兴贸易</公司><部门><名称>销售部</名称><员工>张三</员工><员工>李四</员工></部门>"
xml = xmL & "<部门><名称>行政部</名称><员工>王五</员工><员工>赵六</员工></部门></xml>"
Dim xo As Foxtable.XObject = Foxtable.XObject.Parse(xml)
Output.Show(xo("公司"))
Dim bms As JArray = xo("部门")
For i As Integer = 0 To bms.Count - 1
    Output.show(bms(i)("名称"))
    Dim ygs As JArray = bms(i)("员工")
    For n As Integer = 0 To ygs.Count - 1
        Output.Show(ygs(n))
    Next
Next
```

当然用 For Each 语句处理起来更加简洁一些。

我们可以直接用索引提取数据，例如第一个部门的第二个员工，可以表示为：

```
Dim nm As String = xo("部门")(0)("员工")(1)
```

9.8 生成 XML

我们也可以用 XObject 生成 XML，和用 JObject 生成 JSON 的方法是一回事，只是前者用 ToXML 方法，后者用 ToString 方法。

示例一

在命令窗口执行：

```
Dim jo As New Foxtable.XObject
jo("Name") = "张三"
jo("Group") = "VIP"
jo("Type") = 2
jo("Default") = False
Output.Show(jo.ToXML)
```

显示的结果为：

<XML><Name>张三</Name><Group>VIP</Group><Type>2</Type><Default>false</Default></XML>

示例二

生成嵌套的XML。

在命令窗口执行：

```
Dim xo As New Foxtable.XObject
xo("name") = "李云龙"
xo("age") = "36"
xo("card") = New JObject
xo("card")("bank") = "工行"
xo("card")("account") = "12345678"
Output.Show(xo.ToXML)
```

显示的结果为：

<xml><name>李云龙</name><age>36</age><card><bank>工行</bank><account>12345678</account></card></xml>

示例三

生成带数组的XML。

在命令窗口执行：

```
Dim jo As New Foxtable.XObject
jo("Name") = "张三"
jo("Group") = "VIP"
jo("Roles") = New Jarray("manager","developer") '定义数组
jo("Type") = 2
Output.Show(jo.ToXML)
```

显示的结果为：

<XML><Name>张三</Name><Group>VIP</Group><Roles>manager</Roles><Roles>Developer</Roles><Type>2</Type></XML>

示例四

生成包括对象数组的XML。

在命令窗口执行：

```
Dim jo As New Foxtable.XObject
Dim ja As New JArray
jo("dept") = "销售部"
jo("mpr") = "赵刚"
jo("staff") =  ja '
For i As Integer = 1 To 2
    Dim so As New JObject
    If i = 1 Then
        so("name") = "李云龙"
        so("age") = 36
    Else
        so("name") = "黄晓明"
        so("age") = 26
    End If
    ja.Add(so)
Next
Output.Show(jo.ToXML)
```

显示的结果为：

<XML><dept>销售部</dept><mpr>赵刚</mpr><staff><name>李云龙</name><age>36</age></staff><staff><name>黄晓明</name><age>26</age></staff></XML>

9.9 CDATA 结点

XML 的 CDATA 结点可以理解为数据结点，此类结点的内容会被原样不动解析，内容中的所有标记都会被忽略，也就是说 CDATA 结点可以包括任意特殊字符。
微信接收方的消息都是 XML 结构，其中很多结点就是 CDATA 的：

```
<xml>
<ToUserName><![CDATA[gh_c78d6e907523]]></ToUserName>
<FromUserName><![CDATA[o_0W1wG4xCDJTpdqlTfjSdVjlznQ]]></FromUserName>
<CreateTime>1483933403</CreateTime>
<MsgType><![CDATA[text]]></MsgType>
<Content><![CDATA[测试文本]]></Content>
<MsgId>6373445435737863016</MsgId>
</xml>
```

其中有 4 个结点是 CDATA 的，CDATA 结点内容 以“<![CDATA[”开始，以“]]>”结束。
解析 CDATA 结点

将上面的 XML 数据复制到剪贴板，在命令窗口执行下面的代码：

```
Dim xo As XObject = XObject.Parse(ClipBoard.GetText())
```

```
Output.Show(xo("ToUserName"))
Output.Show(xo("FromUserName"))
Output.Show(xo("CreateTime"))
Output.Show(xo("MsgType"))
Output.Show(xo("Content"))
Output.Show(xo("MsgId"))
```

显示的结果为：

```
gh_c78d6e907523
o_0W1wG4xCDJTpdqlTfjSdVjlznQ
1483933403
text
测试文本
6373445435737863016
```

可以看到 CDATA 结点的解析和普通结点是一样的。

增加 CDATA 结点

XObject 的 AddCDATA 方法用于新增 CDATA 结点，此方法的语法为：

```
AddCDATA(Key, Value)
```

Key：　结点名称
Value：结点值
例如在命令窗口执行：

```
Dim xo As New XObject
xo("name") = "李云龙"
xo("age") = 38
xo.AddCDATA("remark","1+ 2 <> 2")
xo("rank") = "少将"
Output.Show(xo.ToXML)
```

显示的结果为：

```
<XML><name> 李 云 龙 </name><age>38</age><remark><![CDATA[1+  2  <>  2]]></remark><rank> 少 将 </rank></XML>
```

10. 微信接口

10.1 关于微信接口

微信接口分为公众号和企业号，公众号又分为服务号和订阅号。

公众号(包括服务号和订阅号)接口说明：
https://mp.weixin.qq.com/wiki/
企业号接口说明：
http://qydev.weixin.qq.com/wiki/

订阅号和服务号虽同属公众号，接口格式一致，但是腾讯在接口提供方面存在差异，参考：
https://mp.weixin.qq.com/wiki?id=mp1445241432&lang=zh_CN
微信接口非常庞大，我们很难一一介绍，所以这里只介绍常用接口。

其实不管是什么接口，调用的方法都是一样的，不同的只是提交和接收的数据格式不一样。

所以掌握了常用接口的调用，其它接口的调用，大家参考腾讯的接口说明，稍微调整一下代码即可。

如果使用 Foxtable 调用其他第三方接口，也是一样的原理。

10.2 WXBizCrypt 类

腾讯官方提供了微信消息加解密算法的通用代码， 我们已经将其编译成 DLL，并整合在 Foxtable 中，命名空间为"Tencent.WXBizMsgCrypt"，如果你熟悉用.net 进行微信开发，你可以直接调用。

但是对多数用户来说，直接使用封装在 DLL 中的函数进行加解密还是比较麻烦的，我个人就曾经因为一个参数顺序问题，浪费了一个多小时。

为此我们另外提供了一个类 WXBizCrypt，在腾讯官方代码的基础上进行了简写，更适合非专业人员使用。

定义 WXBizCrypt 类的语法为：

Dim wbiz As New WXBizCrypt(AppId, Token, EncodingAESKey)

AppId	开发者 ID，对于企业号来说，就是 CorpID。
Token	设置回调 URL 时指定的 Token。

EncodingAESKey　　设置回调 URL 时指定的 EncodingAESKey。

WXBizCrypt 包括四个函数：

- **DecryptMsg**

  DecryptMsg 函数用于解密消息，语法为：

  DecryptMsg(e)

  e 为 HttpRequest 事件的 e 参数。

  返回值是对 e.PlainText 属性进行解密的结果，如果解密失败，则返回一个错误值。

- **EncryptMsg**

  EncryptMsg 函数用于加密消息，语法为：

  EncryptMsg(e, Message)

  e 为 HttpRequest 事件的 e 参数，Message 为要加密的消息。

  返回值为对 Message 加密后的结果，如果加密失败，则返回一个错误值。

- **GetErorDesc**

  GetErorDesc 函数用于获取错误值对应的错误描述，语法为：

  GetErorDesc(ErrorCode)

  ErrorCode 是一个整数型的错误值。

  函数返回的是 ErrorCode 对应的错误描述。

- **CheckQYSignature**

  CheckQYSignature 专门用于企业号的接入验证，语法为：

  CheckQYSignature(e)

  如果接入验证成功返回 0，否则返回一个错误值。

10.3 公众号接口

10.3.1 申请测试号

如果你已经有微信公众号或企业号，可以跳过本节内容。

对于新手，你可以先申请一个微信公众平台接口测试帐号，不需要提交任何任何资料，而且即时生效。

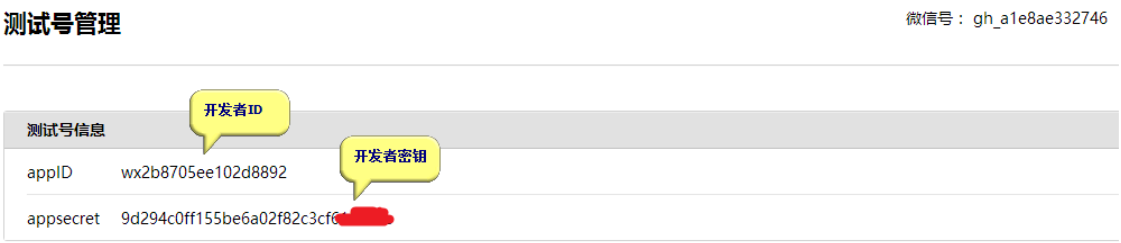
申请测试号的页面地址为：

http://mp.weixin.qq.com/debug/cgi-bin/sandbox?t=sandbox/login

在此页面中单击登录按钮，用你的微信扫一下出现的二维码，即可完成申请。

之后要登录测试号，也是通过以上连接的。

申请成功后，会出现以下内容，其中 appID 为开发者 ID，appsecret 为开发者密钥，有了这两样，才能进行接下来的微信开发：



为方便测试，我们还需要用微信关注此测试号。

用微信扫描测试号的二维码，即可关注此测试号：



注意上图中的微信号，并非用户真实的微信号，而是 OpenID，我们在开发过程中，用 OpenID 来区分不同的用户。

同一个微信号，关于不同的公众号，产生的 OpenID 是不同的，腾讯这样处理，是为了更好地保护用户隐私。

今后文档中提及的用户微信号，通常指的就是这个 OpenID。

10.3.2 接入 HTTP 服务

用户发到公众号(包括测试号)的信息，都是直接发给腾讯的微信服务器的，我们怎样才能在 Foxtable 中接收并回复用户发来的信息呢？

很简单，我们只需在 Foxtable 端开启一个 HTTP 服务，并在微信后台进行配置，通知微信服务器将收到的用户信息，转发给此 HTTP 服务即可。

所以我们首先要在后台接入一个 HTTP 服务，通常也称为回调模式设置。

出于简单和适用更多用户考虑，本文档的不少内容是基于测试号的，但相关内容同样适用正式的公众号，测试号和公众号在编程上是相同的，不同的只是后台管理页面。

在测试号的后台管理页面中，按下图所示进行填写 HTTP 服务的地址和 Token：



此时我们还没有办法提交设置，因为单击提交按钮后，微信服务器会发送一个 GET 访问请求到上述 URL，Foxtable 需要对这个访问请求做出正确的回应 （验证真实性），才能接入成功。

关于真实性验证的方法，可以参考微信官方文档：

http://mp.weixin.qq.com/wiki/4/2ccadaef44fe1e4b0322355c2312bfa8.html

根据官方说明，我们将 HttpRequet 事件代码设为：

```
Select Case e.path
    Case "wefox"
        If e.Request.HttpMethod = "GET" Then
            Dim token = "foxtable" '必须和设置的 Token 相同
            Dim signature As String = e.GetValues("signature")
            Dim timestamp As String = e.GetValues("timestamp")
            Dim nonce As String = e.GetValues("nonce")
            Dim echostr As String = e.GetValues("echostr")
            Dim aryTmp() As String = {token,timestamp,nonce}
            Array.Sort(aryTmp)
```

```
        Dim strTmp As String = String.Join("", aryTmp)
                                                    strTmp                                                    =
Web.Security.FormsAuthentication.HashPasswordForStoringInConfigFile(strTmp, "SHA1")
        If signature = strTmp.ToLower() Then
          e.WriteString(echostr)
        End If
      End If
End Select
```

现在我们可以在测试号的设置页面，单击提交按钮提交设置了。

实际开发的时候，可以将验证部分的代码做成一个自定义函数，函数名为VerifySignature，代码代码为：

```
Dim e As RequestEventArgs = Args(0)
Dim token = "foxtable" '必须和设置的Token相同
Dim signature As String = e.GetValues("signature")
Dim timestamp As String = e.GetValues("timestamp")
Dim nonce As String = e.GetValues("nonce")
Dim echostr As String = e.GetValues("echostr")
Dim aryTmp() As String = {token,timestamp,nonce}
Array.Sort(aryTmp)
Dim strTmp As String = String.Join("", aryTmp)
strTmp = Web.Security.FormsAuthentication.HashPasswordForStoringInConfigFile(strTmp, "SHA1")
Return signature = strTmp.ToLower
```

然后将HttpRequet事件代码改为：

```
Select Case e.path
  Case "wefox"
    If e.Request.HttpMethod.ToUpper = "GET"
      If Functions.Execute("VerifySignature",e) Then
        e.WriteString(e.GetValues("echostr"))
      End If
    End If
End Select
```

10.3.3 获取 access_token

关于获取 access_token 的接口说明，参考：

https://mp.weixin.qq.com/wiki/14/9f9c82c1af308e3b14ba9b973f99a8ba.html

除了被动回复消息，所有微信接口的调用，都必须提交 access_token。

什么是 access_token 呢？

你要调用微信接口，微信服务器肯定要验证你的身份，按理提交开发者 ID(appID)和开发者密钥（appsecret）即可。

但是开发者 ID 和开发者密钥是是固定不变的，频繁通过 POST 方式提交，会存在一定的安全隐患。

access_token 可以说是 appID 和 appsecret 的替代者，我们定期向微信服务器提交 appID 和 appsecret，微信服务器动态生成一个 access_token，之后所有的接口调用，都 要提交此 access_token 作为身份验证凭据。

access_token 的寿命只有 2 小时，也就是说 2 小时后，我们必须重新获取新的 access_token，使用新的 access_token 调用微信接口。

需要注意的是，access_token 一天只能生成 2000 次。

根据接口说明，我们新建一个自定义函数 GetAccessToken 用于获取 access_token，代码为：

```
Static CreateTime As Date '记录最近一次生成 access_token 的时间
Static AccessTocken As String '记录最近一次生成的 access_token
Dim AppId As String= "wx8acdb7df5beb68fd" '开发者 ID
Dim AppSecret As String = "a5a3b06a790d4200a151e6b13c3dc263" '开发者密匙
Dim tp As TimeSpan = Date.Now - CreateTime
Dim url As String = "https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid={0}&secret={1}"
If tp.TotalSeconds > 3600 Then
    Dim hc As New HttpClient(CExp(url,AppId,AppSecret))
    Dim ret As String = hc.GetData()
    If ret = "" Then '如果失败,再尝试一次
        hc.GetData()
    End If
    CreateTime = Date.Now()
    Dim jo As JObject = JObject.Parse(ret)
    If jo("errcode") Is Nothing Then
        AccessTocken = jo("access_token")
    Else
```

```
        MessageBox.show("获取 access_token 失败,原因:" & vbcrlf & jo("errmsg").ToString)
    End If
End If
Return AccessTocken
```

以上代码中的开发者 ID(AppId)和开发者密钥(AppSecret)必须改为自己申请的测试号或公众号的，否则你所有代码都可能无法正常运行。

提示：

1、为保险起见，以上代码每 1 小时刷新一次 access_token，而不是接口说明中的 2 小时。
2、代码用静态变量保存上一次获取的 access_token 以及获取时间，每次获取之前都先比较上一次获取的时间，如果没有超过 1 小时，则直接返回之前获取的 access_token，否则调用接口获取新的 access_token。
3、关于静态变量，参考：使用静态变量
4、本文档将会大量使用 CExp 函数动态合成微信接口地址，关于 CExp 函数，请参考：动态合成表达式之二

现在你可以很方便地获取 access_token 了，例如在命令窗口执行：

```
Dim access_token As String = Functions.Execute("GetAccessToken")
Output.Show(access_token)
```

10.3.4 消息接口

10.3.4.1 接收用户消息

微信服务器收到用户发来的消息后，会以 POST 方式，将消息转发给 Foxtable 端的 HTTP 服务。

消息的格式是纯文本的 XML，保存在 HttpRequest 事件的 e 参数 PlainText 中。

在接收用户消息之前，首先我们要了解普通用户消息的格式，参考：

https://mp.weixin.qq.com/wiki/17/f298879f8fb29ab98b2f2971d42552fd.html
1、首先新建一个数据表，用于保存收到的用户消息，表名为 Message，表结构如下：

查看表结构

列名	类型	长度	标题
FromUserName	字符型	32	
CreateTime	日期时间型		
MsgType	字符型	16	
MsgId	字符型	20	
Format	字符型	16	
MediaId	字符型	96	
PicUrl	字符型	168	
ThumbMediaId	字符型	64	
Content	备注型		
AgentID	整数		
Recognition	备注型		

提示：列名请和和接口消息中的键名保持一致，包括大小写，以方便编码。

2、然后将 HttpRequest 事件代码改为：

```
Select Case e.path
    Case "wefox"
        If e.Request.HttpMethod = "GET"
            If Functions.Execute("VerifySignature",e) Then
                e.WriteString(e.GetValues("echostr"))
            End If
        ElseIf e.Request.HttpMethod = "Post"
            Dim xo As XObject = XObject.Parse(e.PlainText)
            Dim st As New Date(1970,1,1,8,0,0)
            Select Case xo("MsgType")
                Case  "text","image","voice","video","shortvideo"  '为了简单,这里只保存文本、图片、语音和视频消息。
                    Dim dr As DataRow = DataTables("Message").AddNew()
                    dr("FromUserName") = xo("FromUserName")
                    dr("CreateTime") = st.AddSeconds(xo("CreateTime"))
                    dr("MsgType") = xo("MsgType")
                    dr("MsgId") = xo("MsgId")
                    dr("MediaId") = xo("MediaId")
                    dr("ThumbMediaId") = xo("ThumbMediaId")
                    dr("PicUrl") = xo("PicUrl")
                    dr("Content") = xo("Content")
                    dr("Format") = xo("Format")
                    dr("Recognition") = xo("Recognition")
                    dr.Save()
            End Select
        End If
End Select
```

现在即可将收到的用户消息保存在数据表中。

你也可以将代码改为：

```
Select Case e.path
    Case "wefox"
        If e.Request.HttpMethod = "GET"
            If Functions.Execute("VerifySignature",e) Then
                e.WriteString(e.GetValues("echostr"))
            End If
        ElseIf e.Request.HttpMethod = "Post"
            Dim xo As XObject = XObject.Parse(e.PlainText)
            Dim st As New Date(1970,1,1,8,0,0)
            Select Case xo("MsgType")
                Case "text","image","voice","video","shortvideo"
                    Dim dr As DataRow = DataTables("Message").AddNew()
                    dr("CreateTime") = st.AddSeconds(xo("CreateTime"))
                                Dim nms() As String = {"FromUserName","MsgType","MsgId","Content","MediaId","ThumbMediaId","PicUrl","Format","Recognition"}
                    For Each nm As String In nms
                        dr(nm) = xo(nm)
                    Next
                    dr.Save()
            End Select
        End If
End Select
```

可以看到，设计表的时候，列名和消息键名保持一致(包括大小写)，可以简化编码。

同样实际开发的时候，建议将保存消息的代码做成自定义函数，以简化 HttpRequest 事件的代码。

过滤重复消息

微信服务器将收到的用户消息转发给 Foxtable 端的 HTTP 服务，如果 Foxtbale 不能在 5 秒内做出回应，微信服务器会再次转发此消息，总共会重复三次，这意味着如果用户量比较大，服务端处理不及时，可能会收到三条同样的消息，显然我们需要编码过滤掉重复的消息。

为了过滤掉重复消息，可以将 HttpRequest 事件代码改为：

```
Select Case e.path
    Case "wefox"
        If e.Request.HttpMethod = "GET"
            If Functions.Execute("VerifySignature",e) Then
                e.WriteString(e.GetValues("echostr"))
            End If
```

```
    ElseIf e.Request.HttpMethod = "Post"
        Dim xo As XObject = XObject.Parse(e.PlainText)
        Dim st As New Date(1970,1,1,8,0,0)
        Select Case xo("MsgType")
            Case "text","image","voice","video","shortvideo"
                If DataTables("Message").Find("MsgID = '" & xo("MsgId").ToString() & "'") IsNot Nothing Then '如果有相同 MsgID 的信息直接返回
                    Return
                End If
                Dim dr As DataRow = DataTables("Message").AddNew()
                dr("FromUserName") = xo("FromUserName")
                dr("CreateTime") = st.AddSeconds(xo("CreateTime"))
                dr("MsgType") = xo("MsgType")
                dr("MsgId") = xo("MsgId")
                dr("MediaId") = xo("MediaId")
                dr("ThumbMediaId") = xo("ThumbMediaId")
                dr("PicUrl") = xo("PicUrl")
                dr("Content") = xo("Content")
                dr("Format") = xo("Format")
                dr("Recognition") = xo("Recognition")
                dr.Save()
        End Select
    End If
End Select
```

10.3.4.2 验证消息真假

由于 HTTP 服务在网路世界是公开的，因此第三方可以按照微信消息格式，向 Foxtable 端的 HTTP 服务提交虚假的微信消息。

所以对于非常注重安全的系统，有必要验证消息是否来自微信服务器。

方法和接入验证是一样的，不管是 get 方式还是 post 方式，微信服务器都都会包括 signature、timestamp、nonce 和 echostr 等 4 个用于真实性验证的参数。

微信服务器在进行接入验证的时候，是用 get 方式访问 Foxtable 端的 HTTP 服务，提交消息的时候，用的则是 Post 方式。

我们之前只对接入提供了验证，如果要加上消息的真伪验证，只需将代码改为：

```
Select Case e.path
    Case "wefox"
        If Functions.Execute("VerifySignature",e) = False Then '如果验证失败，直接返回
            Return
        End If
```

```
    If e.Request.HttpMethod = "GET"
      e.WriteString(e.GetValues("echostr"))
    ElseIf e.Request.HttpMethod = "Post"
      Dim xo As XObject = XObject.Parse(e.PlainText)
      Dim st As New Date(1970,1,1,8,0,0)
      Select Case xo("MsgType")
        Case  "text","image","voice","video","shortvideo"  '为了简单,这里只保存文本、图片、语音和视频消息。
          Dim dr As DataRow = DataTables("Message").AddNew()
          dr("FromUserName") = xo("FromUserName")
          dr("CreateTime") = st.AddSeconds(xo("CreateTime"))
          dr("MsgType") = xo("MsgType")
          dr("MsgId") = xo("MsgId")
          dr("MediaId") = xo("MediaId")
          dr("ThumbMediaId") = xo("ThumbMediaId")
          dr("PicUrl") = xo("PicUrl")
          dr("Content") = xo("Content")
          dr("Format") = xo("Format")
          dr("Recognition") = xo("Recognition")
          dr.Save()
      End Select
    End If
End Select
```

需要注意的是，本节接下来的例子，都只会对接入进行验证，而不会验证消息真伪。

此外如果开启了安全模式，消息的收发都将采用密文形式，此时就没必要验证消息的真实性了，因为伪造的消息肯定无法正常解密。

10.3.4.3 被动回复消息

关于被动回复消息的接口说明，请参考：

https://mp.weixin.qq.com/wiki/1/6239b44c206cab9145b1d52c67e6c551.html
可以看到被动回复的消息，也是 XML 格式的。

被动回复文本消息

如果要被动回复文本消息，请参考以下的 HttpRequest 事件代码：

```
Select Case e.path
  Case "wefox"
    If e.Request.HttpMethod = "GET"
      If Functions.Execute("VerifySignature",e) Then
        e.WriteString(e.GetValues("echostr"))
```

```
        End If
    ElseIf e.Request.HttpMethod = "Post"
        Dim xo As XObject = XObject.Parse(e.PlainText)
        Dim st As New Date(1970,1,1,8,0,0)
        Select Case xo("MsgType")
            Case "text","image","voice","video","shortvideo"
                '如果要保存收到的消息,请参考上一节在此处编写代码.
                '下面是被动回复文本消息的代码:
                Dim so As New XObject()
                so("ToUserName") = xo("FromUserName")
                so("FromUserName") = xo("ToUserName")
                so("CreateTime") = (Date.Now - st).TotalSeconds()
                so("MsgType")= "text"
                so("Content") = "您好,我是 Foxtable,今天是" & Format(Date.Today,"yyyy 年 MM 月 dd 日
")
                e.WriteString(so.ToXML)
        End Select
    End If
End Select
```

现在用户向我们的测试号发送消息，会收到类似以下内容的回复：

您好,我是Foxtable,今天是2017年02月01日

当然实际开发的时候，我们会根据用户消息的不同，经过处理之后，回复不同的内容。

另外同样建议将回复消息的代码做成自定义函数，以简化HttpRequest事件代码。

被动回复图文消息

如果要被动回复图文消息，请参考以下HttpRequest事件代码：

```
Select Case e.path
  Case "wefox"
    If e.Request.HttpMethod = "GET"
      If Functions.Execute("VerifySignature",e) Then
        e.WriteString(e.GetValues("echostr"))
      End If
    ElseIf e.Request.HttpMethod = "Post"
      Dim xo As XObject = XObject.Parse(e.PlainText)
      Dim st As New Date(1970,1,1,8,0,0)
      Select Case xo("MsgType")
        Case "text","image","voice","video","shortvideo"
          '如果要保存收到的消息,请参考上一节在此处编写代码.
          '下面是被动回复图文消息的代码:
```

```
            Dim so As New XObject()
            Dim ja As New JArray
            so("ToUserName") = xo("FromUserName")
            so("FromUserName") = xo("ToUserName")
            so("CreateTime") = (Date.Now - st).TotalSeconds()
            so("MsgType") = "news"
            so("ArticleCount") = 1 '图文数量
            For cnt As Integer = 1 To 1 '可以回最多8条图文,这里只回1条
              Dim ao As New JObject
              ao("Title") = "特朗普提名保守派大法官" '图文标题
              ao("Description") = "特朗普在白宫提名尼尔·戈萨奇为美国最高法院大法官" '图文简介
                                                    ao("PicUrl")                                    =
"https://ss1.baidu.com/6ONXsjip0QIZ8tyhnq/it/u=1632653074,1222564348&fm=80&w=179&h=119&img.JPEG" '图片
              ao("Url") = "http://news.ifeng.com/a/20170201/50640529_0.shtml" '单击阅读原文后跳转的连接
              ja.Add(ao)
            Next
            so("Articles") = New JObject()
            so("Articles")("item") = ja
            e.WriteString(so.ToXML)
        End Select
    End If
End Select
```

现在用户向我们的测试号发送消息，会收到类似以下内容的图文回复：



准备素材

不管是被动回复还是主动发送，除了文本和图文消息，其它要发送给用户的消息，包括图片、视频和语音，都必须事先将相关文件上传到微信服务器。

因为微信服务器并不会直接给用户发送文件，而是发送一个素材ID(MediaId)，每一个素材ID都对应者一个素材文件。

关于素材的管理，请参考后面的《素材接口》这一章。

被动回复图片消息

必须事先将要回复的图片上传到微信服务器，因为回复的只能是图片素材的ID。

被动回复图片消息参考代码：

```
Select Case e.path
```

```
    Case "wefox"
        If e.Request.HttpMethod = "GET"
            If Functions.Execute("VerifySignature",e) Then
                e.WriteString(e.GetValues("echostr"))
            End If
        ElseIf e.Request.HttpMethod = "Post"
            Dim xo As XObject = XObject.Parse(e.PlainText)
            Dim st As New Date(1970,1,1,8,0,0)
            Select Case xo("MsgType")
                Case "text","image","voice","video","shortvideo"
                    '如果要保存收到的消息,请参考上一节在此处编写代码.
                    '下面是被动回复图片消息的代码:
                    Dim so As New XObject()
                    so("ToUserName") = xo("FromUserName")
                    so("FromUserName") = xo("ToUserName")
                    so("CreateTime") = (Date.Now - st).TotalSeconds()
                    so("MsgType") = "image"
                    so("Image") = New JObject
                    so("Image")("MediaId") = "eHDdfBQmRtOJ-PtkD6WENicqBE7-tFFczZ6p5ZUnz0k" '图片素材
ID
                    e.WriteString(so.ToXML)
            End Select
        End If
End Select
```

被动回复语音消息

必须事先将要回复的语音文件上传到微信服务器，因为回复的只能是语音素材的ID。

被动回复语音消息参考代码：

```
Select Case e.path
    Case "wefox"
        If e.Request.HttpMethod = "GET"
            If Functions.Execute("VerifySignature",e) Then
                e.WriteString(e.GetValues("echostr"))
            End If
        ElseIf e.Request.HttpMethod = "Post"
            Dim xo As XObject = XObject.Parse(e.PlainText)
            Dim st As New Date(1970,1,1,8,0,0)
            Select Case xo("MsgType")
                Case "text","image","voice","video","shortvideo"
                    '如果要保存收到的消息,请参考上一节在此处编写代码.
                    '下面是被动回复语音消息的代码:
                    Dim so As New XObject()
```

```
            so("ToUserName") = xo("FromUserName")
            so("FromUserName") = xo("ToUserName")
            so("CreateTime") = (Date.Now - st).TotalSeconds()
            so("MsgType") = "voice"
            so("Voice") = New JObject
            so("Voice")("MediaId") = "eHDdfBQmRtOJ-PtkD6WENrp8gcW8NSKXkp9sOzn3NoQ" '语音素
材ID
            e.WriteString(so.ToXML)
        End Select
    End If
End Select
```

被动回复视频消息

必须事先将要回复的视频文件上传到微信服务器，因为回复的只能是是因素材的ID。

被动回复视频消息参考代码：

```
Select Case e.path
  Case "wefox"
    If e.Request.HttpMethod = "GET"
      If Functions.Execute("VerifySignature",e) Then
        e.WriteString(e.GetValues("echostr"))
      End If
    ElseIf e.Request.HttpMethod = "Post"
      Dim xo As XObject = XObject.Parse(e.PlainText)
      Dim st As New Date(1970,1,1,8,0,0)
      Select Case xo("MsgType")
        Case "text","image","voice","video","shortvideo"
          '如果要保存收到的消息,请参考上一节在此处编写代码.
          '下面是被动回复语音消息的代码:
          Dim so As New XObject()
          so("ToUserName") = xo("FromUserName")
          so("FromUserName") = xo("ToUserName")
          so("CreateTime") = (Date.Now - st).TotalSeconds()
          so("MsgType") = "video"
          so("Video") = New JObject
          so("Video")("MediaId") = "eHDdfBQmRtOJ-PtkD6WENoqiLnKOzd9llepgH1VioLM" '语音素材
ID
          so("Video")("Title") = "视频标题"
          so("Video")("Description") = "视频内容简介"
          e.WriteString(so.ToXML)
      End Select
    End If
End Select
```

10.3.4.4 主动发送消息

在收到用户消息的 48 小时内，我们可以主动给用户发送消息（微信称之为客服消息）。

关于客服消息的接口说明，请参考：

https://mp.weixin.qq.com/wiki/11/c88c270ae8935291626538f9c64bd123.html
客服消息的为 JSON 格式，且必须提交 access_token 作为访问凭据。

主动发送文本消息

以下是主动发送文本消息的示例代码，你可以在命令窗口运行测试：

```
Dim jo As New JObject
jo("touser") = "ofjtFwBSZ5cNqTKLSKx2TNEAxfBI"
jo("msgtype") = "text"
jo("text") =   New JObject
jo("text")("content") = "您好,我是Foxtable"
Dim URL As String = "https://api.weixin.qq.com/cgi-bin/message/custom/send?access_token={0}"
Dim hc As New HttpClient(CExp(URL, Functions.Execute("GetAccessToken")))
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData())
If jo("errcode") = "0" Then
    MessageBox.Show("消息发送成功")
Else
    MessageBox.show("消息发送失败,原因:" & vbcrlf   & jo("errmsg").ToString)
End If
```

主动发送动态图文消息

下面是主动发送动态图文消息的示例，你可以在命令窗口运行测试：

```
Dim jo As New JObject
Dim ja As New JArray
jo("touser") = "ofjtFwBSZ5cNqTKLSKx2TNEAxfBI"
jo("msgtype") = "news"
For i As Integer = 1 To 1 '可以一次发送最多8个图文消息,这里只发送了1个
    Dim ao As New JObject()
    ao("title") = "特朗普提名保守派大法官" '标题
    ao("description") = "特朗普在白宫提名尼尔·戈萨奇为美国最高法院大法官" '描述
    ao("url") = "http://news.ifeng.com/a/20170201/50640529_0.shtml" '连接
                                                ao("picurl")                                    =
"https://ss1.baidu.com/6ONXsjip0QIZ8tyhnq/it/u=1632653074,1222564348&fm=80&w=179&h=119&i
mg.JPEG" '图片
    ja.Add(ao)
```

```
Next
jo("news") =   New JObject()
jo("news")("articles") = ja
Dim URL As String = "https://api.weixin.qq.com/cgi-bin/message/custom/send?access_token={0}"
Dim hc As New HttpClient(CExp(URL,Functions.Execute("GetAccessToken")))
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData())
If jo("errcode") = "0" Then
    MessageBox.Show("消息发送成功")
Else
    MessageBox.show("消息发送失败,原因:" & vbcrlf   & jo("errmsg").ToString)
End If
```

除了上面的动态图文消息，还有一种永久图文消息，永久图文消息是一种素材，需要事先上传到服务器，发送的时候，将素材ID发给客户即可。

准备素材

不管是被动回复还是主动发送，除了文本和动态图文消息，其它要发送给用户的消息，包括图片、视频、语音以及永久图文消息，都必须事先将相关文件上传到微信服务器。

因为微信服务器并不会直接给用户发送文件，而是发送一个素材ID(MediaId)，每一个素材ID都对应者一个素材文件。

关于素材的管理，请参考后面的《素材接口》这一章。

主动发送图片消息

必须事先将要发送的图片上传到微信服务器，因为我们发送的只能是图片素材的 ID，无法直接发送文件。

主动发送图片消息参考代码：

```
Dim jo As New JObject
jo("touser") = "oQdguwzDwS696vh5vnXU3u0aD9fk"
jo("msgtype") = "image"
jo("image") =   New JObject
jo("image")("media_id") = "eHDdfBQmRtOJ-PtkD6WENicqBE7-tFFczZ6p5ZUnz0k"
Dim URL As String = "https://api.weixin.qq.com/cgi-bin/message/custom/send?access_token={0}"
Dim hc As New HttpClient(CExp(URL, Functions.Execute("GetAccessToken")))
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData())
If jo("errcode") = "0" Then
    MessageBox.Show("消息发送成功")
Else
```

```
    MessageBox.show("消息发送失败,原因:" & vbcrlf   & jo("errmsg").ToString)
End If
```

主动发送视频消息

必须事先将要发送的视频上传到微信服务器，因为我们发送的只能是视频素材的 ID，无法直接发送文件。

主动发送视频消息参考代码：

```
Dim jo As New JObject
jo("touser") = "oQdguwzDwS696vh5vnXU3u0aD9fk"
jo("msgtype") = "video"
jo("video") =   New JObject
jo("video")("media_id") = "eHDdfBQmRtOJ-PtkD6WENoqiLnKOzd9llepgH1VioLM"
jo("video")("title") = "视频标题"
jo("video")("description") = "视频内容简介"
Dim URL As String = "https://api.weixin.qq.com/cgi-bin/message/custom/send?access_token={0}"
Dim hc As New HttpClient(CExp(URL, Functions.Execute("GetAccessToken")))
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData())
If jo("errcode") = "0" Then
    MessageBox.Show("消息发送成功")
Else
    MessageBox.show("消息发送失败,原因:" & vbcrlf   & jo("errmsg").ToString)
End If
```

主动发送语音消息

必须事先将要发送的语音上传到微信服务器，因为我们发送的只能是语音素材的 ID，无法直接发送文件。

主动发送语音消息参考代码：

```
Dim jo As New JObject
jo("touser") = "oQdguwzDwS696vh5vnXU3u0aD9fk"
jo("msgtype") = "voice"
jo("voice") =   New JObject
jo("voice")("media_id") = "eHDdfBQmRtOJ-PtkD6WENrp8gcW8NSKXkp9sOzn3NoQ"
Dim URL As String = "https://api.weixin.qq.com/cgi-bin/message/custom/send?access_token={0}"
Dim hc As New HttpClient(CExp(URL, Functions.Execute("GetAccessToken")))
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData())
If jo("errcode") = "0" Then
    MessageBox.Show("消息发送成功")
Else
    MessageBox.show("消息发送失败,原因:" & vbcrlf   & jo("errmsg").ToString)
End If
```

主动发送永久图文消息

和动态合成的图文消息不同，永久图文消息需要事先上传到服务器，之后将素材 ID 发送给用户。

主动发送永久图文消息的参考代码：

```
Dim jo As New JObject
jo("touser") = "oQdguwzDwS696vh5vnXU3u0aD9fk"
jo("msgtype") = "mpnews"
jo("mpnews") =   New JObject
jo("mpnews")("media_id") = "eHDdfBQmRtOJ-PtkD6WENpJyI8iZ7Qy2l5SvQS58Jz0" '素材 ID
Dim URL As String = "https://api.weixin.qq.com/cgi-bin/message/custom/send?access_token={0}"
Dim hc As New HttpClient(CExp(URL, Functions.Execute("GetAccessToken")))
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData())
If jo("errcode") = "0" Then
    MessageBox.Show("消息发送成功")
Else
    MessageBox.show("消息发送失败,原因:" & vbcrlf   & jo("errmsg").ToString)
End If
```

10.3.4.5 接收事件推送

关于接收事件推送的接口说明，请参考：

https://mp.weixin.qq.com/wiki/7/9f89d962eba4c5924ed95b513ba69d9b.html

本节只介绍用户关注和取消关注事件，其他类型的事件将在相关章节介绍。

本节的任务是：

1、创建一个用户表，用于保存用户信息。
2、有用户关注测试号，则在用户表增加一行，用于登记此用户信息。
3、有用户取消关注测试号，则在用户表删除对应的行。

显然要完成上述任务，我们还需要了解获取用户信息的接口，参考：

https://mp.weixin.qq.com/wiki/1/8a5ce6257f1d3b2afb20f83e72b72ce9.html
首先设计结构如下图所示的表，表名为 WXUsers，用于记登记已关注用户的信息：

查看表结构

列名	类型	长度	标题
openid	字符型	32	
nickname	字符型	32	
sex	字符型	1	
city	字符型	16	
country	字符型	16	
province	字符型	16	
headimgurl	字符型	168	
groupid	字符型	16	
remark	字符型	64	
language	字符型	8	
subscribe_time	日期时间型		
tagid_list	字符型	32	

然后将 HttpRequest 事件代码设置为：

```
Select Case e.path
    Case "wefox"
        If e.Request.HttpMethod.ToUpper = "GET"
            If Functions.Execute("VerifySignature",e) Then
                e.WriteString(e.GetValues("echostr"))
            End If
        ElseIf e.Request.HttpMethod.ToUpper = "Post"
            Dim xo As XObject = XObject.Parse(e.PlainText)
            Dim st As New Date(1970,1,1,8,0,0)
            Select Case xo("MsgType")
                Case "text","image"
                    '保存和回复文本和图片消息的代码
                Case "event" '如果是事件
                    Select Case xo("Event")
                        Case "subscribe" '关注事件
                            Dim url As String = "https://api.weixin.qq.com/cgi-bin/user/info?access_token={0}&openid={1}&lang=zh_CN"
                            url = CExp(url, Functions.Execute("GetAccessToken"), xo("FromUserName"))
                            Dim hc As New HttpClient(url)
                            Dim jo As JObject = Jobject.Parse(hc.GetDaTa)
                            If jo("errcode") IsNot Nothing Then
                                '这里编写代码记录错误信息,方便查阅
                            Else
                                Dim dr As DataRow = DataTables("WXUsers").AddNew()
                                Dim nms() As String = {"openid","nickname","sex","city","country","province","headimgurl","groupid","remark","language"}
                                For Each nm As String In nms
                                    dr(nm) = jo(nm)
                                Next
                                dr("subscribe_time") = Date.Now()
```

```
                    dr.Save()
                End If
            Case "unsubscribe" '取消关注事件
                DataTables("WXUsers").DeleteFor("openid = '" & xo("FromUserName").ToString() & "'" )
        End Select
    End Select
  End If
End Select
```

实际开发的时候，建议大家将相关代码做成自定义函数，避免 HttpRequest 事件过于复杂。

在事件中回复消息

你也可以在事件中被动回复消息，例如希望在用户关注后发送一条欢迎语，可以参考代码：

```
Select Case e.path
  Case "wefox"
    If e.Request.HttpMethod.ToUpper = "GET"
      If Functions.Execute("VerifySignature",e) Then
        e.WriteString(e.GetValues("echostr"))
      End If
    ElseIf e.Request.HttpMethod.ToUpper = "Post"
      Dim xo As XObject = XObject.Parse(e.PlainText)
      Dim st As New Date(1970,1,1,8,0,0)
      Select Case xo("MsgType")
        Case "text","image","voice","video","shortvideo"
          '保存和收到的消息
        Case "event" '如果是事件
          Select Case xo("Event")
            Case "subscribe" '关注事件
                Dim url As String = "https://api.weixin.qq.com/cgi-bin/user/info?access_token={0}&openid={1}&lang=zh_CN"
                url = CExp(url, Functions.Execute("GetAccessToken"), xo("FromUserName"))
                Dim hc As New HttpClient(url)
                Dim jo As JObject = Jobject.Parse(hc.GetDaTa)
                If jo("errcode") IsNot Nothing Then
                  '这里编写代码记录错误信息,方便查阅
                Else
                  '保存消息
                  Dim dr As DataRow = DataTables("WXUsers").AddNew()
                  Dim nms() As String = {"openid","nickname","sex","city","country","province","headimgurl","groupid","remark","language"}
                  For Each nm As String In nms
                    dr(nm) = jo(nm)
                  Next
                  dr("subscribe_time") = Date.Now()
```

```
                    dr.Save()
                    '回复欢迎语
                    Dim so As New XObject()
                    so("ToUserName") = xo("FromUserName")
                    so("FromUserName") = xo("ToUserName")
                    so("CreateTime") = (Date.Now - st).TotalSeconds()
                    so("MsgType")= "text"
                    so("Content") = "您好,感谢您关注 FoxTable,您的关注就是我们的动力."
                    e.WriteString(so.ToXML)
                End If
            Case "unsubscribe" '取消关注事件
                DataTables("WXUsers").DeleteFor("openid = '" & xo("FromUserName").ToString() & "'" )
        End Select
    End Select
End If
End Select
```

10.3.4.6 发送模版消息

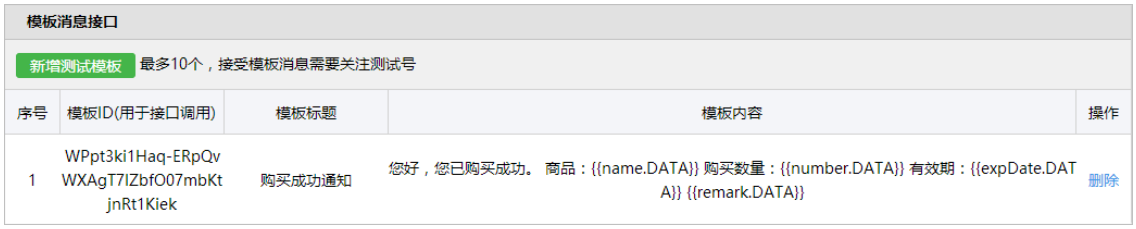
模版消息主要用于下发业务通知,没有 48 小时的限制，但必须遵守腾讯的运营规范，以免被封号：

https://mp.weixin.qq.com/wiki?action=doc&id=mp1433751288&t=0.41837619353429356

关于模版消息接口，可以参考：

https://mp.weixin.qq.com/wiki?action=doc&id=mp1433751277&t=0.8783707802201653
模版消息接口的内容很多，但实际上我们只需了解其中的发送接口即可，因为模版管理通常是在后台完成的。

测试号可以自定义消息模版，正式的公众号只能从模板库中选择模版。
下图是我在测试号中自定义的一个模版：



模版中类似{{name.DATA}}这样的内容，要在发送的时候替换为实际的内容，注意替换的时候使用的键 名不包括".DATA"，例如{{number.DATA}}的键 名是 number。
下面的代码是我基于上述模版编写的发送代码，你可以直接在命令窗口执行测试：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/template/send?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
```

```
Dim jo As New JObject
Dim ja As New JArray
jo("touser") = "ofjtFwBSZ5cNqTKLSKx2TNEAxfBI"
jo("template_id") = "WPpt3ki1Haq-ERpQvWXAgT7IZbfO07mbKtjnRt1Kiek" '模板ID
jo("url") = "http://www.foxtable.com"
jo("data") = New JObject()
jo("data")("name") = New JObject()
jo("data")("name")("value") = "Foxtable"
jo("data")("name")("color") = "#173177"
jo("data")("number") = New JObject()
jo("data")("number")("value") = "2"
jo("data")("number")("color") = "#173177"
jo("data")("expDate") = New JObject()
jo("data")("expDate")("value") = "2019年12月27日"
jo("data")("expDate")("color") = "#173177"
jo("data")("expDate") = New JObject()
jo("data")("expDate")("value") = "2019年12月27日"
jo("data")("expDate")("color") = "#173177"
jo("data")("remark") = New JObject()
jo("data")("remark")("value") = "如有疑问,可联系客服QQ:800014337"
jo("data")("remark")("color") = "#173177"
hc.content = jo.Tostring()
jo = JObject.Parse(hc.getdata)
If jo("errcode") = "0" Then
    MessageBox.show("模版消息发送成功!")
Else
    Messagebox.Show(jo.ToString)
End If
```

下图是手机端收到的模版消息：



可以看到，模版消息内容是可以设置内容颜色的，常用颜色值可以参考：

http://www.w3school.com.cn/cssref/css_colornames.asp

接收模版消息发送完成事件

微信服务器在完成模版消息发送后，会推送一个事件到 Foxtable 端的 Http 服务，这样我们就能知道此模版消息是否已经送达用户。

要记录模版消息是否送达，我们首先要新建一个数据表，表名为 TemplateMessage，结构为：

查看表结构

列名	类型	长度	标题
OpenID	字符型	32	
MsgID	长整数		
Status	字符型	32	
Description	备注型		

然后修改发送代码为：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/template/send?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject
Dim ja As New JArray
jo("touser") = "ofjtFwBSZ5cNqTKLSKx2TNEAxfBI"
jo("template_id") = "WPpt3ki1Haq-ERpQvWXAgT7IZbfO07mbKtjnRt1Kiek" '模板ID
jo("url") = "http://www.foxtable.com"
jo("data") = New JObject()
jo("data")("name") = New JObject()
jo("data")("name")("value") = "Foxtable"
jo("data")("name")("color") = "#173177"
jo("data")("number") = New JObject()
jo("data")("number")("value") = "2"
jo("data")("number")("color") = "#173177"
jo("data")("expDate") = New JObject()
jo("data")("expDate")("value") = "2019年12月27日"
jo("data")("expDate")("color") = "#173177"
jo("data")("expDate") = New JObject()
jo("data")("expDate")("value") = "2019年12月27日"
jo("data")("expDate")("color") = "#173177"
jo("data")("remark") = New JObject()
jo("data")("remark")("value") = "如有疑问,可联系客服QQ:800014337"
jo("data")("remark")("color") = "#173177"
hc.content = jo.Tostring()
jo = JObject.Parse(hc.getdata)
If jo("errcode") = "0" Then
    Dim dr As DataRow = DataTables("TemplateMessage").AddNew()
    dr("MsgID") = jo("msgid") '动态生成的消息ID,每个消息都不同
    dr("OpenID") = "ofjtFwBSZ5cNqTKLSKx2TNEAxfBI" '接收方OpenID
    dr("Description") = "这是消息内容说明"
    dr.Save()
Else
    Messagebox.Show(jo.ToString)
End If
```

最后将 Foxtable 端的 HttpRequest 事件代码设置为：

```
Select Case e.path
    Case "wefox"
        If e.Request.HttpMethod.ToUpper = "GET"
            If Functions.Execute("VerifySignature",e) Then
                e.WriteString(e.GetValues("echostr"))
            End If
        ElseIf e.Request.HttpMethod.ToUpper = "Post"
            Dim xo As XObject = XObject.Parse(e.PlainText)
            Dim st As New Date(1970,1,1,8,0,0)
            Select Case xo("MsgType")
                Case "text","image"
                    '保存和回复文本和图片消息的代码
                Case "event"
                    Select Case xo("Event")
                        Case "subscribe" '关注事件
                            '处理用户关注事件的代码
                        Case "unsubscribe" '取消关注事件
                            '处理用户取消关注事件的代码'
                        Case "LOCATION" '上报地理位置事件
                            '处理用户上报地理位置的代码
                        Case "TEMPLATESENDJOBFINISH"
                            Dim dr As DataRow = DataTables("TemplateMessage").Find("MsgID = " & xo("MsgID").ToString)
                            If dr Is Nothing Then
                                dr = DataTables("TemplateMessage").AddNew()
                                dr("MsgID") = xo("MsgID") '动态生成的消息ID,每个消息都不同
                                dr("OpenID") = xo("FromUserName")'接收方OpenID
                            End If
                            dr("Status") = xo("Status") '记录发送状态
                            dr.Save()
                    End Select
            End Select
        End If
End Select
```

大小写的问题

微信的接口有些混乱，以上面的代码为例，有个细节不知你是否已经留意到，发送代码中有一行：

dr("MsgID") = jo("msgid") '动态生成的消息 ID,每个消息都不同

而 Foxtbale 端的 HttpRequest 事件中有一行类似代码：

dr("MsgID") = xo("MsgID") '动态生成的消息 ID,每个消息都不同

前者是jo("msgid"),后者是xo("MsgID"),前者用的是小写的msgid,后者则是混合了大小写MsgID。

我们在编写代码的时候，必须按照接口说明，严格区分大小写，否则是无法取到数据的。

10.3.4.7 加密解密消息

要学习本节内容，必须有正式的微信公众号。

测试号只有明文模式，正式的公众号可以用选择明文模式还是安全模式收发消息：



需要说明的是，安全模式是针对回调方式（包括消息和事件），我们只需调整 Foxtable 端的 HttpRequest 事件代码，对消息进行加密或解密，而主动发送消息的代码不需要做任何调整。

关于微信消息的加密和解密，请参考：

https://mp.weixin.qq.com/wiki/14/70e73cedf9fd958d2e23264ba9333ad2.html

我们已经将腾讯提供的 C#算法代码编译成 DLL，并整合在 Foxtable 中，命名空间为"Tencent.WXBizMsgCrypt"，如果你熟悉用.net 进行微信开发，你可以直接调用。

但是对多数用户来说，直接使用封装在 DLL 总的函数进行加解密还是比较麻烦的，我个人就曾经因为一个参数顺序问题，浪费了一个多小时。

为方便普通用户的使用，我们另外提供了一个类 WXBizCrypt，将相关方法进行了简写，请参考：WXBizCrypt 类

定义 WXBizCrypt 的语法为：

Dim wbiz As New WXBizCrypt(AppId, Token, EncodingAESKey)

AppId	公众号的的开发者 ID。
Token	上图中指定的 Token。
EncodingAESKey	上图中指定的 EncodingAESKey。

一个示例

下面是安全模式下 HttpRequest 事件的一个简单示例，包括对收到消息的解密，以及对要发送消息的加密：

```
Select Case e.path
    Case "wefox"
        If e.Request.HttpMethod = "GET"
            If Functions.Execute("VerifySignature",e) Then
                e.WriteString(e.GetValues("echostr"))
            End If
        ElseIf e.Request.HttpMethod = "Post"            '
                                Dim                 wbiz                As                  New
WXBizCrypt("wxdd6b45ebecd2f81e","foxtable","1f3JudIWHWXTkFa880d1ZXOVbVsZParNEj4uI15wfF6")
            Dim st As New Date(1970,1,1,8,0,0)
            Dim msg   As String = wbiz.DecryptMsg(e) '解密收到的消息
            If   IsNumeric(msg) Then '如果解密失败
                Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
                Return
            End If
            Dim xo As XObject = XObject.Parse(msg)
            Select Case xo("MsgType")
                Case "text","image","voice","video","shortvideo"
                    Dim dr As DataRow = DataTables("Message").AddNew()
                    dr("FromUserName") = xo("FromUserName")
                    dr("CreateTime") = st.AddSeconds(xo("CreateTime"))
                    dr("MsgType") = xo("MsgType")
                    dr("MsgId") = xo("MsgId")
                    dr("MediaId") = xo("MediaId")
                    dr("ThumbMediaId") = xo("ThumbMediaId")
```

```
                dr("PicUrl") = xo("PicUrl")
                dr("Content") = xo("Content")
                dr("Format") = xo("Format")
                dr("Recognition") = xo("Recognition")
                '被动回复消息
                Dim so As New XObject()
                so("ToUserName") = xo("FromUserName")
                so("FromUserName") = xo("ToUserName")
                so("CreateTime") = (Date.Now - st).TotalSeconds()
                so("MsgType")= "text"
                so("Content") = "您好,我是 Foxtable,今天是" & Format(Date.Today,"yyyy 年 MM 月 dd 日
")
                msg = wbiz .EncryptMsg(e,so.ToXML ) '加密要发送的消息
                If  IsNumeric(msg) Then '如果加密失败
                   Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
                Else
                   e.WriteString(msg)
                End If
            End Select
        End If
End Select
```

10.3.5 素材接口

10.3.5.1 上传永久素材

关于上传永久素材的接口说明，请参考：
https://mp.weixin.qq.com/wiki/10/10ea5a44870f53d79449290dfd43d006.html
首先新建一个表，表名为 Materials，用于储存永久素材信息，表结构如下：

查看表结构

列名	类型	长度	标题
Type	字符型	16	
MediaID	字符型	96	
URL	字符型	168	
UpdateTime	日期时间型		
FileName	字符型	32	
Content	备注型		

上传图片素材

上传图片素材的代码，可以在命令窗口测试：

```
Dim url As String =
```

```
"https://api.weixin.qq.com/cgi-bin/material/add_material?access_token={0}&type=image"
Dim hc As new HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
hc.Files.Add("media","c:\data\sample.jpg") '指定要上传的图片文件
Dim jo As JObject = JObject.Parse( hc.getdata)
If jo("errcode") Is Nothing Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("Type") = "image"
    dr("mediaID") = jo("media_id")
    dr("URL") = jo("url")
    dr("UpdateTime") = Date.Now
    dr("FileName") = "sample.jpg"
    dr.Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

上传缩略图素材

上传缩略图素材的代码完全相同，只需将 type 参数改为 thumb 即可：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/material/add_material?access_token={0}&type=thumb"
Dim hc As new HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
hc.Files.Add("media","c:\data\006s.jpg") '指定要上传的图片文件
Dim jo As JObject = JObject.Parse( hc.getdata)
If jo("errcode") Is Nothing Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("Type") = "thumb"
    dr("mediaID") = jo("media_id")
    dr("URL") = jo("url")
    dr("UpdateTime") = Date.Now
    dr("FileName") = "006s.jpg"
    dr.Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

上传语音素材

上传语音素材的代码也基本相同，只需将 type 参数改为 voice，另语音素材上传成功后不会返回 URL：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/material/add_material?access_token={0}&type=voice"
Dim hc As new HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
hc.Files.Add("media","c:\data\sample.mp3") '指定要上传的语音文件
Dim jo As JObject = JObject.Parse( hc.getdata)
```

```
If jo("errcode") Is Nothing Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("Type") = "voice"
    dr("mediaID") = jo("media_id")
    dr("UpdateTime") = Date.Now
    dr("FileName") = "sample.mp3"
    dr.Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

上传视频素材

根据接口说明，上传视频素材有点特殊，需要同时提交一个 JSON 格式的视频素材说明，代码可以参考：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/material/add_material?access_token={0}&type=video"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
hc.Files.Add("media","c:\data\samlpe.mp4") '指定要上传的视频文件
Dim jo As New JObject
jo("title") = "视频标题"
jo("introduction") = "视频简介"
hc.FormData.Add("description", jo.ToString)
jo = JObject.Parse(hc.getdata)
If jo("errcode") Is Nothing Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("Type") = "video"
    dr("mediaID") = jo("media_id")
    dr("UpdateTime") = Date.Now
    dr("FileName") = "sample.mp4"
    dr.Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

上传图文素材

上传永久图文素材的代码参考：

```
Dim ar As New JArray
For i As Integer = 1 To 1 '可以有多段图文
    Dim ao As New JObject
    ao("title") = "标题"
    ao("thumb_media_id") = "eHDdfBQmRtOJ-PtkD6WENiWR_BUO5GO3cWv7uZAGbIk" '缩略图 ID
    ao("author") = "作者"
```

```
    ao("digest") = "图文消息摘要"
    ao("show_cover_pic") = "1" '显示封面
    ao("content") = "图文消息内容,可以包括 HTML 标签"
    ao("content_source_url") = "http://www.foxtable.com" '点击"阅读原文"后的 URL
    ar.add(ao)
Next
Dim jo As New JObject
jo("articles") = ar
Dim url As String = "https://api.weixin.qq.com/cgi-bin/material/add_news?access_token={0}"
Dim hc As new HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") Is Nothing Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("Type") = "news"
    dr("mediaID") = jo("media_id")
    dr("UpdateTime") = Date.Now
    dr("Content") = ar.ToString
    dr.Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

10.3.5.2 上传图文消息内的图片

腾讯对公众号永久素材的数量做了限制，图文消息素材、图片素材的上限为 5000，视频和语音素材的上限为 1000。

本接口所上传的图片不受 5000 个图片素材上限的限制，图片仅支持 jpg/png 格式，大小必须在 1MB 以下，返回的只有 URL。

参考代码：

```
Dim URL As String = "https://api.weixin.qq.com/cgi-bin/media/uploadimg?access_token={0}"
Dim hc As new HttpClient(CExp(URL, Functions.Execute("GetAccessToken")))
hc.Files.Add("media","c:\data\sample.jpg")
Dim jo As JObject = JObject.Parse( hc.getdata)
If jo("errcode") Is Nothing Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("URL") = jo("url")
    dr("UpdateTime") = Date.Now
    dr("FileName") = "sample.jpg"
    dr.Save()
Else
```

```
    MessageBox.Show(jo.ToString)
End If
```

10.3.5.3 下载永久素材

下载永久素材的接口说明：

http://mp.weixin.qq.com/wiki/4/b3546879f07623cb30df9ca0e420a5d0.html

图文素材和视频素材返回的是 JSON 数据，其它类型的素材返回的是文件。

返回 JSON 数据很好处理，但是返回文件处理起来稍显麻烦，因为有两种可能：

1、获取成功，返回文件。
2、获取失败，返回错误信息。

但是 HttpClient 的 GetFile 会直接将返回内容保存为文件，不管是返回的内容是文件还是错误信息。

我们可以通过判断 HttpClient 的 ResponseContentType 属性，如果其值为"text/plain"，说明返回的不是文件，而是 错误信息。

例如下载指定 ID 的图片素材到本地，可以参考代码：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/material/get_material?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim sucess As Boolean = True '用于标记是否下载成功
Dim fl As String   = "c:\data\abc.jpg" '要保存为的本地文件
Dim jo As New JObject
jo("media_id") = "eHDdfBQmRtOJ-PtkD6WENicqBE7-tFFczZ6p5ZUnz0k" '要获取的图片素材 ID
hc.Content = jo.ToString()
If hc.GetFile(fl) Then
    If hc.ResponseContentType.StartsWith("text/plain") Then
        jo = JObject.Parse(Filesys.ReadAllText(fl))
        MessageBox.Show(jo.ToSTring)  '显示错误
        FileSys.DeleteFile(fl) '删除文件
        sucess = False
    End If
Else
    sucess = False
End If
If sucess Then
    MessageBox.Show("图片素材下载成功!")
Else
    MessageBox.Show("图片素材下载失败!")
```

```
End If
```

下载永久视频素材

前面提到，通过接口获取的永久视频素材不是文件，而是一段 JSON 数据：

```
{
"title":TITLE,
"description":DESCRIPTION,
"down_url":DOWN_URL,
}
```

我们需要根据这段 JSON 数据提供的链接下载视频素材，参考代码：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/material/get_material?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim fl As String   = "c:\data\abc.mp4" '要保存为的本地文件
Dim jo As New JObject
jo("media_id") = "eHDdfBQmRtOJ-PtkD6WENmQFOu5VJhsunxEZ2BxkfJU" '要获取的视频素材 ID
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("down_url") IsNot Nothing Then
    hc = New HttpClient(jo("down_url"))
    If hc.GetFile(fl) Then
        MessageBox.Show("视频素材下载成功!")
    Else
        MessageBox.Show("视频素材下载失败!")
    End If
Else
    MessageBox.Show(jo.ToString)
End If
```

10.3.5.4 获取素材列表

获取素材列表的接口参考：

https://mp.weixin.qq.com/wiki?action=doc&id=mp1444738734&t=0.381933189174956
你可以通过下面的代码，获取所有素材信息保存在 Materials 表中：

```
DataTables("Materials").DataRows.Clear()
Dim typs() As String = {"image","voice","video","news"} """
For Each typ As String In typs
    Dim ur As String = "https://api.weixin.qq.com/cgi-bin/material/batchget_material?access_token={0}"
    Dim st As New Date(1970,1,1,8,0,0)
    Dim offset As Integer = 0
```

```
    Do
        Dim hc As new HttpClient(CExp(ur, Functions.Execute("GetAccessToken")))
        Dim jo As New JObject
        jo("type") = typ
        jo("offset") = offset
        jo("count") = 20 '一次最多获取 20 个
        hc.Content = jo.ToString()
        jo = JObject.Parse(hc.GetData)
        If jo("errcode") Is Nothing Then
            If CInt(jo("item_count")) = 0 Then
                Exit Do
            End If
            For Each item As JToken In jo("item")
                Dim dr As DataRow = DataTables("Materials").AddNew
                dr("Type") = typ
                dr("MediaID") = item("media_id")
                dr("FileName") = item("name")
                dr("URL") = item("url")
                dr("UpdateTime") = st.AddSeconds(item("update_time"))
                If item("content") IsNot Nothing Then
                    dr("Content") = item("content")("news_item").ToString()
                End If
            Next
            offset = offset + CInt(jo("item_count"))
        Else
            MessageBox.Show(jo.Tostring) '显示错误信息
            Exit Do
        End If
    Loop
Next
DataTables("Materials").Save()
```

10.3.5.5 管理临时素材

临时素材上传到微信服务器 3 天后，会被自动删除。

新增临时素材

新增临时素材的接口说明，参考：

https://mp.weixin.qq.com/wiki/15/2d353966323806a202cd2deaafe8e557.html

例如上传一个临时图片素材，可以参考代码：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/media/upload?access_token={0}&type=image"
Dim hc As new HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
```

```
hc.Files.Add("media","c:\data\sample.jpg") '要上传的文件
Dim jo As JObject = JObject.Parse( hc.getdata)
If jo("errcode") Is Nothing Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("Type") = "临时图片"
    dr("mediaID") = jo("media_id")
    dr("UpdateTime") = Date.Now
    dr("FileName") = "sample.jpg"
    dr.Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

所有素材类型的上传代码都一样，只需修改一下 type 参数即可。

唯一例外的是上传缩略图(thumb)，实际返回的 JSON 数据和腾讯官方的说明并不一致，返回的媒体 ID 键值是"thumb_media_id"，不是"media_id"，所以需要修改代码：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/media/upload?access_token={0}&type=thumb"
Dim hc As new HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
hc.Files.Add("media","c:\data\006s.jpg") '要上传的文件
Dim jo As JObject = JObject.Parse( hc.getdata)
If jo("errcode") Is Nothing Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("Type") = "临时缩略图"
    dr("mediaID") = jo("thumb_media_id")
    dr("UpdateTime") = Date.Now
    dr("FileName") = "006s.jpg"
    dr.Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

下载临时素材

下载临时素材的接口说明：

https://mp.weixin.qq.com/wiki/9/677a85e3f3849af35de54bb5516c2521.html

下载临时素材,会有两种可能：

1、获取成功，返回文件。
2、获取失败，返回 JSON 格式的错误信息。

但是 HttpClient 的 GetFile 会直接将返回内容保存为文件，不管是返回的内容是文件还是错误信息。

我们可以通过判断 HttpClient 的 ResponseContentType 属性，如果其值为"text/plain"，说明返回的不

是文件，而是提示信息。

参考代码：

```
Dim fl As String   = "c:\data\abc.jpg" '要保存为的本地文件
Dim sucess As Boolean = True '用于标记是否下载成功
Dim url As String = "https://api.weixin.qq.com/cgi-bin/media/get?access_token={0}&media_id={1}"
Dim mediaID As String = "qINAMviNpYpn5EcMwb2EmrsoCR6zUo8rJQplei1uyluhfbSVZ_esSva9NkuU8W1m" '
要下载的素材 ID
url = CExp(url, Functions.Execute("GetAccessToken"), mediaID)
Dim hc As New HttpClient(url)
If hc.GetFile(fl) Then
    If hc.ResponseContentType.StartsWith("text/plain") Then
        Dim jo As JObject = JObject.Parse(Filesys.ReadAllText(fl))
        MessageBox.Show(jo.ToString)  '显示错误信息
        FileSys.DeleteFile(fl) '删除文件
        sucess = False
    End If
Else
    sucess = False
End If
If sucess Then
    MessageBox.Show("临时素材下载成功!")
Else
    MessageBox.Show("临时素材下载失败!")
End If
```

10.3.6 用户接口

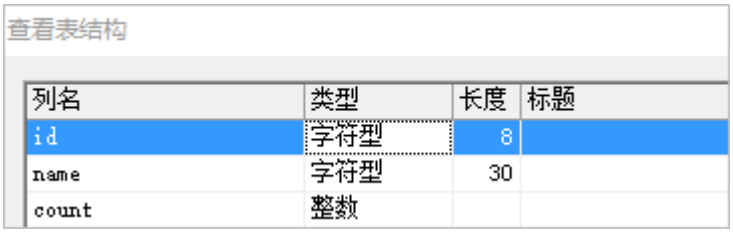
10.3.6.1 用户标签管理

标签用于给用户分类，一个公众号可以创建 100 个标签，一个用户可以有多个标签。

用户标签管理接口说明，请参考：

https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140837&t=0.3704000145509809

首先创建一个表，用于保存用户标签设置，表名为 tag，结构如下：


查看表结构

列名	类型	长度	标题
id	字符型	8	
name	字符型	30	
count	整数		

创建标签

一个公众号可以创建 100 个标签，创建标签的代码参考：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/tags/create?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
jo("tag") = New JObject()
jo("tag")("name") = "vip"
hc.Content = jo.ToString
jo = JObject.Parse(hc.GetData)
If jo("errcode") Is Nothing Then
    Dim dr As DataRow = DataTables("tag").AddNew()
    dr("id") = jo("tag")("id")
    dr("name") = jo("tag")("name")
    dr.Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

以上代码创建了一个名为 vip 的标签。

修改标签

我们可以修改标签名称，例如将 ID 为 104 的标签名改为"VIP 用户"，参考代码为：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/tags/update?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
jo("tag") = New JObject()
jo("tag")("id") = "104"
jo("tag")("name") = "VIP 用户"
hc.Content = jo.ToString
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("标签修改成功")
Else
    MessageBox.Show(jo.ToString)
End If
```

删除标签

删除标签的代码请参考：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/tags/delete?access_token={0}"
```

```
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
jo("tag") = New JObject()
jo("tag")("id") = "101"
hc.Content = jo.ToString
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    DataTables("tag").DeleteFor("id = '101'")
Else
    MessageBox.show(jo.ToString)
End If
```

列出现有标签

列出现有标签的代码请参考：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/tags/get?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As JObject = JObject.Parse(hc.GetData)
If jo("errcode") Is Nothing Then
    DataTables("tag").DataRows.Clear()
    For Each tg As JToken In jo("tags")
        Dim dr As DataRow = DataTables("tag").AddNew()
        dr("id") = tg("id") '标签id
        dr("name") = tg("name") '标签名
        dr("count") = tg("count") '此标签下的用户数
    Next
Else
    MessageBox.show(jo.ToString)
End If
```

批量为用户打标签

可以批量为用户打标签，每个用户最多可以打三个标签，代码参考：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/tags/members/batchtagging?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
Dim ja As New JArray()
ja.Add("ofjtFwBSZ5cNqTKLSKx2TNEAxfBI")
ja.Add("ofjtFwGvPBfy9aGz9OJ-qWiG5oA0")
jo("openid_list") = ja
jo("tagid") = "104"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
```

```
    MessageBox.Show("批量打标签成功!")
Else
    MessageBox.show(jo.ToString)
End If
```

批量为用户取消标签

如果要为若干用户取消指定标签，可以参考代码：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/tags/members/batchuntagging?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
Dim ja As New JArray()
ja.Add("ofjtFwBSZ5cNqTKLSKx2TNEAxfBI")
ja.Add("ofjtFwGvPBfy9aGz9OJ-qWiG5oA0")
jo("openid_list") = ja
jo("tagid") = "104"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("批量取消标签成功!")
Else
    MessageBox.show(jo.ToString)
End If
```

获取某用户的标签

例如在命令窗口执行下面的代码，可以列出指定用户的全部标签：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/tags/getidlist?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
jo("openid") = "ofjtFwBSZ5cNqTKLSKx2TNEAxfBI"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("tagid_list") IsNot Nothing Then
    For Each jk As JToken  In jo("tagid_list")
       Output.show(jk)
    Next
Else
    MessageBox.show(jo.ToString)
End If
```

获取某标签下的所有用户

例如在命令窗口执行下面的代码，可以列出 ID 为"104"标签下的所有用户：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/user/tag/get?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
jo("tagid") = "104"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") Is Nothing Then
    If CInt(jo("count")) > 0 Then
        For Each jk As JToken   In jo("data")("openid")
            Output.show(jk)
        Next
    End If
Else
    MessageBox.show(jo.ToString)
End If
```

10.3.6.2 设置用户备注名

设置用户备注名的接口说明，请参考：

https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140838&t=0.11167659567780225

参考代码：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/user/info/updateremark?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
jo("openid") = "ofjtFwBSZ5cNqTKLSKx2TNEAxfBI"
jo("remark") = "用户备注名"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("设置用户备注名成功")
Else
    MessageBox.Show(jo.ToString)
End If
```

10.3.6.3 获取用户列表

获取用户列表的接口说明，参考：

https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140840&t=0.37665316006084615
该接口返回的是用户的 OpenID。

例如在命令窗口执行以下代码，可以显示所有用户 OpenID：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/user/get?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As JObject = JObject.Parse(hc.GetData)
If jo("errcode") Is Nothing Then
    Dim sb As New StringBuilder
    For Each jk As JToken   In jo("data")("openid")
      sb.AppendLine(jk)
    Next
    Messagebox.Show(sb.ToString)
Else
    MessageBox.Show(jo.ToString)
End If
```

提示：如果你的公众号有非常多的用户，不能用 MessageBox 来显示用户列表，应该保存为一个文本文件。

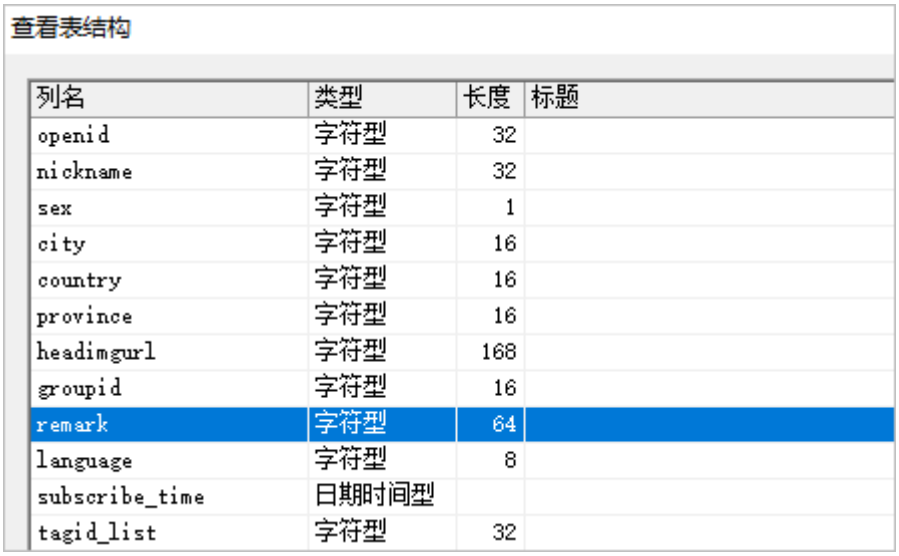
根据接口说明可知，上述代码最多只能获取前 1 万个 OpenID，如果你的公众号用户数超过 1 万，需要采用下面的代码才能获取所有 OpenID：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/user/get?access_token={0}&next_openid={1}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken"),""))
Dim sb As New StringBuilder
Dim jo As JObject = JObject.Parse(hc.GetData)
Do
    If jo("errcode") Is Nothing Then
      Dim cnt As Integer = jo("count")
      If cnt > 0 Then
        For Each jk As JToken   In jo("data")("openid")
          sb.AppendLine(jk)
        Next
        hc =   New HttpClient(CExp(url, Functions.Execute("GetAccessToken").jo("next_openid").Tostring))
        jo = JObject.Parse(hc.GetData)
      Else
        Exit Do
      End If
    Else
      MessageBox.Show(jo.ToString)
      Exit Do
    End If
Loop
FileSys.WriteAllText("c:\data\openid.txt", sb.ToString, False)
```

10.3.6.4 获取用户信息

获取用户信息的接口参考：

https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140839&t=0.7904743578958528
首先设计结构如下图所示的表，表名为 WXUsers，用于保存用户的信息：


查看表结构

列名	类型	长度	标题
openid	字符型	32	
nickname	字符型	32	
sex	字符型	1	
city	字符型	16	
country	字符型	16	
province	字符型	16	
headimgurl	字符型	168	
groupid	字符型	16	
remark	字符型	64	
language	字符型	8	
subscribe_time	日期时间型		
tagid_list	字符型	32	

建议列名和接口返回信息的键名保持一致(包括大小写)，以方便编码。

获取单个用户信息

获取单个用户的信息，可以参考：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/user/info?access_token={0}&openid={1}&lang=zh_CN"
url = CExp(url, Functions.Execute("GetAccessToken"), "ojDlHuAjZ8Z3eSm0gW1g0Lxbz54s")
Dim hc As New HttpClient(url)
Dim jo As JObject = Jobject.Parse(hc.GetDaTa)
If jo("errcode") Is  Nothing Then
    Dim dr As DataRow = DataTables("WXUsers").AddNew()
    Dim nms() As String = {"openid","nickname","sex","city","country","province","headimgurl","groupid","remark","language"}
    For Each nm As String In nms
        dr(nm) = jo(nm)
    Next
    If jo("tagid_list") IsNot Nothing Then
        dr("tagid_list") = CompressJson(jo("tagid_list")).Trim("[","]")
    End If
    dr.Save()
Else
```

```
    MessageBox.Show(jo.ToString)
End If
```

获取全部用户信息

如果希望批量刷新所有用户的信息，可以参考以下代码。

代码首先要获取所有用户的 OpenID，且每次只能获取 1 万个，然后获取用户详情，且每次只能获取 100 个用户的详情，所以存多个嵌套的循环，是本文档关于微信开发部分最复杂的一段代码。

学习以下代码的时候，请务必结合接口说明来理解。

对于部分用户来说，这段代码可能过于复杂了，如果你确实理解不了，也没有关系，直接使用即可，但如果你真的消化吸收了这段代码，估计今后的第三方接口再也难不倒你了。

具体代码：

```
DataTables("WXUsers").DataRows.Clear()
Dim ul As String = "https://api.weixin.qq.com/cgi-bin/user/get?access_token={0}&next_openid={1}" '获取用户 OpenID 列表接口
Dim il = "https://api.weixin.qq.com/cgi-bin/user/info/batchget?access_token={0}" '批量获取用户信息接口
Dim nms() As String = {"openid","nickname","sex","city","country","province","headimgurl","groupid","remark","language"} '列名
Dim hc As New HttpClient(CExp(ul, Functions.Execute("GetAccessToken"),""))
Dim ids As New List(of String) 'OpenId 集合,每次获取 10000 个
Dim ba As New Jarray '准备用来获取用户信息的 OpenId 列表,一次不能超过 100 个
Dim jo As JObject = JObject.Parse(hc.GetData)
Do '循环获取,一次只能获取 10000 个 OpenID
    If jo("errcode") Is Nothing Then
        Dim cnt As Integer = jo("count")
        If cnt = 0 Then '如果已经获取完所有用户'
            Exit Do
        End If
        For Each jk As JToken  In jo("data")("openid")
            ids.Add(jk)
        Next
        For i As Integer = 0 To ids.Count - 1  '循环获取用户详情,一次不能超过 100 个用户
            Dim uo As New JObject
            uo("openid") = ids(i)
            ba.Add(uo)
            If ba.Count = 100 OrElse i = ids.Count - 1 Then '每满 100 个就获取一次
                Dim bo As New Jobject
                bo("user_list") = ba
                hc = New HttpClient(CExp(il, Functions.Execute("GetAccessToken")))
```

```
            hc.Content = bo.ToString()
            Dim ro As JObject = JObject.Parse(hc.GetData)
            If ro("errcode") Is  Nothing Then
               Dim ia As JArray = ro("user_info_list") '获取的用户信息列表
               For Each jt As JToken In ia
                    Dim dr As DataRow = DataTables("WXUsers").Find("openid = '" & jt("openid").ToString() & "'")
                  If dr Is Nothing Then
                     dr = DataTables("WXUsers").AddNew()
                  End If
                  For Each nm As String In nms
                     dr(nm) = jt(nm)
                  Next
                  dr("tagid_list") = CompressJSON(jt("tagid_list"))
               Next
            Else
               MessageBox.Show(ro.ToString)
               Exit For
            End If
            ba.Clear() '清除集合,准备获取下一批用户详情,每批只能 100 个.
         End If
      Next
      ids.Clear '清除已经获取的 OpenID.,准备接收下一批 OpenID
                  hc = New HttpClient(CExp(ul, Functions.Execute("GetAccessToken"),jo("next_openid").Tostring))  '获取下一批 OpenID
      jo = JObject.Parse(hc.GetData)
   Else
      MessageBox.Show(jo.ToString)
      Exit Do
   End If
Loop
DataTables("WXUsers").Save()
```

获取全部用户信息是一项比较耗时的操作，建议实际开发的时候，用一个进度条显示获取进度。

10.3.6.5 获取用户地理位置

获取用户地理位置的接口说明：

https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140841&t=0.6130908688268812

用户同意上报地理位置后，每次进入公众号会话时，都会自动上报地理位置，也可以选择在会话持续期间，每 5 分钟 自动上报一次地理位置。

上报地理位置以推送 XML 数据包的形式发送到 Foxtable 端的 HTTP 服务。

一个例子

如果希望希望将用户的地理位置保存为一个日志文件中，可以参考下面的 HttpRequest 事件代码：

```
Select Case e.path
    Case "wefox"
        If e.Request.HttpMethod.ToUpper = "GET"
            If Functions.Execute("VerifySignature",e) Then
                e.WriteString(e.GetValues("echostr"))
            End If
        ElseIf e.Request.HttpMethod.ToUpper = "Post"
            Dim logFile As String  = "c:\data\location" & Format(Date.Today,"yyyyMMdd") & ".txt" '你可以选择每日或每月一个日志文件
            Dim xo As XObject = XObject.Parse(e.PlainText)
            Dim st As New Date(1970,1,1,8,0,0)
            Select Case xo("MsgType")
                Case "text","image","voice","video","shortvideo"
                    '保存和回复普通消息的代码
                Case "event" '如果是事件
                    Select Case xo("Event")
                        Case "subscribe"
                            '处理用户关注事件的代码
                        Case "unsubscribe"
                            '处理用户取消关注事件的代码'
                        Case "LOCATION" '自动上报地理位置事件
                            Dim OpenID As String  = xo("FromUserName") '用户 OpenID
                            Dim CreateTime As Date = st.AddSeconds(xo("CreateTime"))
                            Dim y As String = xo("Longitude") '经度
                            Dim x As String = xo("Latitude") '纬度
                            Filesys.WriteAllText(logFile , OpenID & "在" & CreateTime & "的位置为:" & x & "|" & y & vbcrlf , True)
                    End Select
            End Select
        End If
End Select
```

除了自动上报位置，用户还可以通过菜单主动上报位置，主动方式提交给服务器的信息更多，具体可以参考《菜单接口》这一章。

手动上报地理位置

除了自动上报地理位置，微信的系统菜单也自带有上报地理位置的功能：



手动上报的地理位置是以普通消息的形式发送到 Foxtable 端的 HTTP 服务,消息类型为 location。

下面的 HttpRequest 事件代码能同时记录自动上报和手动上报的地理位置：

```
Select Case e.path
    Case "wefox"
        If e.Request.HttpMethod.ToUpper = "GET"
            If Functions.Execute("VerifySignature",e) Then
                e.WriteString(e.GetValues("echostr"))
            End If
        ElseIf e.Request.HttpMethod.ToUpper = "Post"
            Dim logFile As String  = "c:\data\location" & Format(Date.Today,"yyyyMMdd") & ".txt" '你可以选择每日或每月一个日志文件
            Dim xo As XObject = XObject.Parse(e.PlainText)
            Dim st As New Date(1970,1,1,8,0,0)
            Dim OpenID As String  = xo("FromUserName") '用户 OpenID
            Dim CreateTime As Date = st.AddSeconds(xo("CreateTime"))
            Select Case xo("MsgType")
                Case "text","image","voice","video","shortvideo"
                    '保存和回复普通消息
                Case "location" '主动上报地理位置消息
                    Dim x As String = xo("Location_X") '经度
                    Dim y As String = xo("Location_Y") '纬度
                    Dim lb As String  = xo("Label") '地理位置
                    Filesys.WriteAllText(logFile , OpenID & "在" & CreateTime & "的位置为:" & x & "|" & y & lb & vbcrlf , True)
```

```
            Case "event" '如果是事件
                Select Case xo("Event")
                    Case "subscribe"
                        '处理用户关注事件的代码
                    Case "unsubscribe"
                        '处理用户取消关注事件的代码'
                    Case "LOCATION" '自动上报地理位置事件
                        Dim y As String = xo("Longitude") '经度
                        Dim x As String = xo("Latitude") '纬度
                        Filesys.WriteAllText(logFile , OpenID & "在" & CreateTime & "的位置为:" & x & "|"
& y & vbcrlf , True)
                End Select
            End Select
        End If
End Select
```

下面第一行是自动上报的地理位置，第二行是主动上报的地理位置，自动上报地理位置只有经纬度，而主动上报除了经纬度，还有字符型的地理位置：

ofjtFwGvPBfy9aGz9OJ-qWiG5oA0 在 2017-02-24 10:14:41 的位置为:21.257692|110.377396
ofjtFwGvPBfy9aGz9OJ-qWiG5oA0 在 2017-02-24 10:14:57 的位置为:21.257692|110.377396|广东省湛江市赤坎区人民大道北 39 号
实际上还有第三种地理位置上报方法，就是通过自定义菜单，后面在《菜单接口》讲述，但个人觉得以上两种足矣。

10.3.6.6 黑名单管理

关于黑名单管理接口，参考：

https://mp.weixin.qq.com/wiki?action=doc&id=mp1471422259_pJMWA&t=0.011829662488346548

拉黑用户

拉黑用户的代码参考：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/tags/members/batchblacklist?access_token={0}"
Dim hc As New HttpClient(CExp(url,Functions.Execute("GetAccessToken")))
Dim jo As New JObject
Dim ja As New JArray
ja.Add("ojDlHuIVSy7OUrI43K_9sTS7n8gE") '要拉黑的用户 OpenID
ja.Add("ojDlHuOzrMnI90y6P0m7uQ79eZQQ")
jo("openid_list") = ja
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("已经成功拉黑")
Else
```

```
    MessageBox.Show(jo.ToString)
End If
```

取消拉黑

取消拉黑用户的代码参考：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/tags/members/batchunblacklist?access_token={0}"
Dim hc As New HttpClient(CExp(url,Functions.Execute("GetAccessToken")))
Dim jo As New JObject
Dim ja As New JArray
ja.Add("oQdguwzDwS696vh5vnXU3u0aD9fk") '要取消拉黑的用户OpenID
ja.Add("oQdguw1qKnHQu5oIew-osH6JSIZA"
jo("openid_list") = ja
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("已经取消拉黑")
Else
    MessageBox.Show(jo.ToString)
End If
```

黑名单列表

要列出黑名单中的所有用户，请参考：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/tags/members/getblacklist?access_token={0}"
Dim hc As New HttpClient(CExp(url,Functions.Execute("GetAccessToken")))
Dim jo As New JObject
jo("begin_openid") = ""
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") Is Nothing AndAlso jo("count") <> "0" Then
    Dim sb As New StringBuilder
    For Each tk As JToken In jo("data")("openid")
        sb.AppendLine(tk.Tostring)
    Next
    Output.Show(sb.ToString())
Else
    MessageBox.Show(jo.ToString)
End If
```

10.3.7 群发接口

10.3.7.1 关于群发接口

关于群发接口的说明，请参考：

https://mp.weixin.qq.com/wiki?action=doc&id=mp1481187827_i0l21&t=0.9636354997031221
概括几点：

- 订阅号每天可以给用户群发 1 条消息，服务号每月可以给用户群发 4 条消息 ，这些限制是针对单个用户而言，并非是你调用群发接口的次数。
- 除了文本消息，其他类型的消息都必须先将素材上传到微信服务器，群发给用户的是素材的 mediaID。
- 关于消息素材的上传，请参考前面《素材接口》这一章。
- 由于群发的次数有限，所以微信提供了预览接口，可以先将消息发送到指定微信预览，确定没有问题后再群发。
- 可以一次群发给所有用户，可以根据用户标签群发，也可以根据用户 OpenID 列表群发。

接下来分别介绍消息预览、按标签群发，按 OpenID 列表群发等接口。

为方便测试，我提前给我的测试号上传了一些素材：

	mediaID	URL	Description
1	XuVl2KVO7GQG2JOGv6QLLzErZRJ3ug1l1PQtg4KUgJw	http://mmbiz.qpic.cn/mmbiz_jpg/q01SgX8ekyEcAnvUV7dqYJTWde	图片素材说明
2	XuVl2KVO7GQG2JOGv6QLL9Qm5Kbktj4K_qIDHnazgBU	http://mmbiz.qpic.cn/mmbiz_jpg/q01SgX8ekyEcAnvUV7dqYJTWde	缩略图素材说明
3	XuVl2KVO7GQG2JOGv6QLL5Ciga6xkphUAbcNEk2UUrE		语音素材说明
4	XuVl2KVO7GQG2JOGv6QLLOBi3oDe1o9RwMENqM-oZ8A		视频素材说明
5		http://mmbiz.qpic.cn/mmbiz_jpg/q01SgX8ekyEcAnvUV7dqYJTWde	图文素材内的图片说明
6	XuVl2KVO7GQG2JOGv6QLLzTT6WJPsPOyUDdwnoueJio		图文素材说明带缩略图

10.3.7.2 预览群发消息

群发的次数有限，所以最好先通过预览调整好消息内容再群发。

预览的次数是每天 100 次，所以也不要滥用。

预览各种类型的群发消息，代码基本是相同的。
预览文本消息

参考代码：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/mass/preview?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
jo("touser") = "ofjtFwBSZ5cNqTKLSKx2TNEAxfBI" '接收预览信息的 OpenID
```

```
jo("text") = New JObject
jo("text")("content") = "需要群发的文本消息"
jo("msgtype") = "text"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("已经发送预览消息,请注意查看")
Else
    MessageBox.Show(jo.ToString)
End If
```

预览图片消息

必须事先将要群发的图片上传到微信服务器，因为群发的只是图片素材的 ID，参考代码:

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/mass/preview?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
jo("touser") = "ofjtFwBSZ5cNqTKLSKx2TNEAxfBI" '接收预览信息的 OpenID
jo("image") = New JObject
jo("image")("media_id") = "XuVl2KV07GQG2J0Gv6QLLzErZRJ3ug1l1PQtg4KUgJw" '已经上传的永久图片素材 ID
jo("msgtype") = "image"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("已经发送预览消息,请注意查看")
Else
    MessageBox.Show(jo.ToString)
End If
```

预览图文消息

必须事先将要群发的图文消息上传到微信服务器，因为群发的只是图文消息的 ID，参考代码:

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/mass/preview?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
jo("touser") = "ofjtFwBSZ5cNqTKLSKx2TNEAxfBI" '接收预览信息的 OpenID
jo("mpnews") = New JObject
jo("mpnews")("media_id") = "XuVl2KV07GQG2J0Gv6QLLzTT6WJPsP0yUDdwnoueJio" '已经上传的永久图文素材 ID
jo("msgtype") = "mpnews"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
```

```
    MessageBox.Show("已经发送预览消息,请注意查看")
Else
    MessageBox.Show(jo.ToString)
End If
```

预览语音消息

必须事先将要群发的语音文件上传到微信服务器，因为群发的只是语音素材的 ID，参考代码:

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/mass/preview?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
jo("touser") = "ofjtFwBSZ5cNqTKLSKx2TNEAxfBI" '接收预览信息的 OpenID
jo("voice") = New JObject
jo("voice")("media_id") = "XuVl2KV07GQG2J0Gv6QLL5Ciga6xkphUAbcNEk2UUrE" '已经上传的永久语音素材 ID
jo("msgtype") = "voice"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("已经发送预览消息,请注意查看")
Else
    MessageBox.Show(jo.ToString)
End If
```

预览视频消息

必须事先将要群发的视频文件上传到微信服务器，因为群发的只是视频素材的 ID，参考代码:

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/mass/preview?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
jo("touser") = "ofjtFwBSZ5cNqTKLSKx2TNEAxfBI" '接收预览信息的OpenID
jo("mpvideo") = New JObject
jo("mpvideo")("media_id") = "XuVl2KV07GQG2J0Gv6QLL0Bi3oDe1o9RvMENqM-oZ8A" '已经上传的永久视频素材ID
jo("msgtype") = "mpvideo"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("已经发送预览消息,请注意查看")
Else
    MessageBox.Show(jo.ToString)
End If
```

10.3.7.3 根据标签进行群发

公众号的群发次数有限，所有必须慎重，用完了次数，你就只能等明天或者下个月了，^_^。

当你向某个用户群发消息的次数用完之后，继续向其群发消息，微信服务器并不会有任何提示，但对方肯定不会收到你群发的消息，因为微信服务器在群发的过程中自动将其排除在外了。

此外群发是是需要时间的，我们要做的只是通知微信服务器向全部或部分用户群发消息，什么时候完成得看腾讯的。

不过腾讯在群发完毕后，会推送一个事件到 Foxtable 端的 Http 服务，我们可以据此获取群发结果。

群发各种类型的消息，代码基本是相同的，需要注意的是，测试号只能群发文本消息。

群发文本消息

参考代码：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/mass/sendall?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
jo("filter") = New JObject
jo("filter")("is_to_all") = False
jo("filter")("tag_id") = 104 '向 104(标签 ID)标签下的所有用户群发
jo("text") = New JObject
jo("text")("content") = "需要群发的文本消息"
jo("msgtype") = "text"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("提交群发消息成功")
Else
    MessageBox.Show(jo("errmsg"))
End If
```

群发图片消息

必须事先将要群发的图片上传到微信服务器，因为群发的只是图片素材的 ID，参考代码:

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/mass/sendall?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
jo("filter") = New JObject
jo("filter")("is_to_all") = False
jo("filter")("tag_id") = 100 '向 100(标签 ID)标签下的所有用户群发
jo("image") = New JObject
jo("image")("media_id") = "AO9_Rt0R6isjlzuh9Mskm9VT57PP6mgf4HppbSur32Y" '要群发的图片素材 ID
jo("msgtype") = "image"
hc.Content = jo.ToString()
```

```
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("提交群发消息成功")
Else
    MessageBox.Show(jo("errcode"))
End If
```

群发图文消息

必须事先将要群发的图文消息上传到微信服务器，因为群发的只是图文素材的 ID，参考代码:

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/mass/sendall?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
jo("filter") = New JObject
jo("filter")("is_to_all") = False
jo("filter")("tag_id") = 100 '向 100(标签 ID)标签下的所有用户群发
jo("mpnews") = New JObject
jo("mpnews")("media_id") = "AO9_Rt0R6isjlzuh9Mskm5tTjK69DAjI7xozcdwStdk" '要群发的图文消息素材 ID
jo("msgtype") = "mpnews"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("提交群发消息成功")
Else
    MessageBox.Show(jo("errcode"))
End If
```

群发语音消息

必须事先将要群发的语音文件上传到微信服务器，因为群发的只是语音素材的 ID，参考代码:

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/mass/sendall?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
jo("filter") = New JObject
jo("filter")("is_to_all") = False
jo("filter")("tag_id") = 100 '向 100(标签 ID)标签下的所有用户群发
jo("voice") = New JObject
jo("voice")("media_id") = "AO9_Rt0R6isjlzuh9Mskm1VSuKR3vHgKyJkG3kHyKwg" '要群发的语音消息素材 ID
jo("msgtype") = "voice"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
```

```
    MessageBox.Show("提交群发消息成功")
Else
    MessageBox.Show(jo("errcode"))
End If
```

群发视频消息

必须事先将要群发的视频文件上传到微信服务器，因为群发的只是视频素材的 ID，参考代码:

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/mass/sendall?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
jo("filter") = New JObject
jo("filter")("is_to_all") = False
jo("filter")("tag_id") = 100 '向 100(标签 ID)标签下的所有用户群发
jo("mpvideo") = New JObject
jo("mpvideo")("media_id") = "AO9_Rt0R6isjlzuh9MskmzglvCD-tb59QcXGNbdkE3E" '要群发的 视频消息素材 ID
jo("msgtype") = "mpvideo"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("提交群发消息成功")
Else
    MessageBox.Show(jo("errcode"))
End If
```

10.3.7.4 根据 OpenID 列表群发

根据 OpenID 群发，订阅号不可用，认证后的服务号可用。

公众号的群发次数有限，所有必须慎重，用完了次数，你就只能等明天或者下个月了，^_^。

当你向某个用户群发消息的次数用完之后，继续向其群发消息，微信服务器并不会有任何提示，但对方肯定不会收到你群发的消息，因为微信服务器在群发的过程中自动将其排除在外了。

此外群发是是需要时间的，我们要做的只是通知微信服务器向全部或部分用户群发消息，什么时候完成得看腾讯的。

不过腾讯在群发完毕后，会推送一个事件到 Foxtable 端的 HTTP 服务，我们可以据此获取群发结果。

群发各种类型的消息，代码基本是相同的，需要注意的是，测试号只能群发文本消息。

群发文本消息

参考代码：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/mass/send?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
Dim ja As New JArray
ja.Add("ojDlHuK8rykTyijSgRZl4osUn09E") '添加用户 OpenID
ja.Add("ojDlHuGwo2JqybedsudAmn8feI_g")
jo("touser") = ja
jo("msgtype") = "text"
jo("text") = New JObject
jo("text")("content") = "需要群发的文本消息"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("提交群发消息成功")
Else
    MessageBox.Show(jo("errmsg"))
End If
```

群发图文消息

必须事先将要群发的图文消息上传到微信服务器，因为群发的只是图文素材的 ID，参考代码:

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/mass/send?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
Dim ja As New JArray
ja.Add("ojDlHuK8rykTyijSgRZl4osUn09E") '添加用户 OpenID
ja.Add("ojDlHuGwo2JqybedsudAmn8feI_g")
jo("touser") = ja
jo("mpnews") = New JObject
jo("mpnews")("media_id") = "AO9_Rt0R6isjlzuh9Mskm5tTjK69DAjI7xozcdwStdk" '要群发的图文消息素材 ID
jo("msgtype") = "mpnews"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("提交群发消息成功")
Else
    MessageBox.Show(jo("errcode"))
End If
```

群发视频消息

必须事先将要群发的视频文件上传到微信服务器，因为群发的只是视频素材的 ID，参考代码:

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/mass/send?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
Dim ja As New JArray
ja.Add("ojDlHuK8rykTyijSgRZl4osUn09E") '添加用户 OpenID
ja.Add("ojDlHuGwo2JqybedsudAmn8feI_g")
jo("touser") = ja
jo("mpvideo") = New JObject
jo("mpvideo")("media_id") = "AO9_Rt0R6isjlzuh9MskmzglvCD-tb59QcXGNbdkE3E" '要群发的视频消息素材 ID
jo("mpvideo")("title") = "视频标题"
jo("mpvideo")("description") = "视频内容简介"
jo("msgtype") = "mpvideo"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("提交群发消息成功")
Else
    MessageBox.Show(jo("errcode"))
End If
```

群发语音消息

必须事先将要群发的语音文件上传到微信服务器，因为群发的只是语音素材的 ID，参考代码:

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/mass/send?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
Dim ja As New JArray
ja.Add("ojDlHuK8rykTyijSgRZl4osUn09E") '添加用户 OpenID
ja.Add("ojDlHuGwo2JqybedsudAmn8feI_g")
jo("touser") = ja
jo("voice") = New JObject
jo("voice")("media_id") = "AO9_Rt0R6isjlzuh9Mskm1VSuKR3vHgKyJkG3kHyKwg" '要群发的语音消息素材 ID
jo("msgtype") = "voice"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("提交群发消息成功")
Else
    MessageBox.Show(jo("errcode"))
End If
```

10.3.7.5 获取群发结果

微信服务器在完成群发消息任务后，会向 Foxtable 端的 Http 服务推送一个消息，我们可以据此获取群发结果。

首先我们要新建一个表，表名为 MassMessage，用于存储消息群发结果，表结构如下：

查看表结构

列名	类型	长度	标题
MsgID	长整数		消息ID
Status	字符型	32	发送状态
TotalCount	整数		发送人数
SentCount	整数		成功送达人数
ErrorCount	整数		发送失败人数
Description	字符型	64	本次纯发说明

以群发文本消息为例，将群发代码修改为：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/message/mass/send?access_token={0}"
Dim hc As New HttpClient(CExp(url, Functions.Execute("GetAccessToken")))
Dim jo As New JObject()
Dim ja As New JArray
ja.Add("ofjtFwBSZ5cNqTKLSKx2TNEAxfBI") '添加用户OpenID
ja.Add("ofjtFwGvPBfy9aGz9OJ-qWiG5oA0")
jo("touser") = ja
jo("msgtype") = "text"
jo("text") = New JObject
jo("text")("content") = "需要群发的文本消息"
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    Dim dr As DataRow = DataTables("MassMessage").AddNew
    dr("MsgID") = jo("msg_id")
    dr("Description") = "本次群发任务说明"
Else
    MessageBox.Show(jo("errmsg"))
End If
```

修改代码的目的是为例在群发消息提交成功后，获取饭回来的消息 ID(MsgID)，微信服务器完成群发之后，推送的事件中包括消息 ID。

最后修改 Foxtable 端的 HttpRequest 事件代码为：

```
Select Case e.path
    Case "wefox"
        If e.Request.HttpMethod.ToUpper = "GET"
            If Functions.Execute("VerifySignature",e) Then
                e.WriteString(e.GetValues("echostr"))
```

```
        End If
    ElseIf e.Request.HttpMethod.ToUpper = "Post"
        Dim xo As XObject = XObject.Parse(e.PlainText)
        Dim st As New Date(1970,1,1,8,0,0)
        Select Case xo("MsgType")
            Case "text","image"
                '保存和回复文本和图片消息的代码
            Case "event"
                Select Case xo("Event")
                    Case "subscribe" '关注事件
                        '处理用户关注事件的代码
                    Case "unsubscribe" '取消关注事件
                        '处理用户取消关注事件的代码'
                    Case "LOCATION" '上报地理位置事件
                        '处理用户上报地理位置的代码
                    Case "TEMPLATESENDJOBFINISH"
                        '处理模板消息发送事件的代码
                    Case "MASSSENDJOBFINISH"
                        Dim dr As DataRow = DataTables("MassMessage").Find("MsgID = " &
xo("MsgID").ToString)
                        If dr Is Nothing Then
                            dr = DataTables("MassMessage").AddNew()
                            dr("MsgID") = xo("MsgID") '动态生成的消息ID,每个消息都不同
                        End If
                        dr("Status") = xo("Status") '记录发送状态
                        dr("TotalCount") = xo("TotalCount") '尝试发送人数
                        dr("SentCount") = xo("SentCount") '成功发送人数
                        dr("ErrorCount") = xo("ErrorCount") '发送失败人数
                        dr.Save()
                End Select
        End Select
    End If
End Select
```

我建议你将 HttpRequest 事件中的每个任务都做成自定义函数，避免 HttpRequest 事件代码过程。

10.3.8 二维码接口

10.3.8.1 关于二维码接口

关于二维码接口的说明，可以参考：

https://mp.weixin.qq.com/wiki?action=doc&id=mp1443433542&t=0.6276053744844635

目前有 2 种类型的二维码：
1、临时二维码，是有过期时间的，最长可以设置为在二维码生成后的 30 天（即 2592000 秒）后过期，但能够生成较多数量。临时二维码主要用于帐号绑定等不要求二维码永久保存的业务场景
2、永久二维码，是无过期时间的，但数量较少（目前为最多 10 万个）。永久二维码主要用于适用于帐号绑定、用户来源统计等场景。
每次生成二维码，都需要指定一个场景 ID。

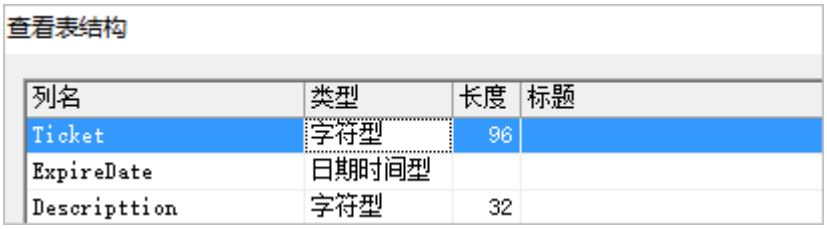
用户扫描带场景 ID 的二维码时，可能推送以下两种事件：

1、如果用户还未关注公众号，则用户可以关注公众号，关注后微信会将带场景值关注事件推送到 Foxtable 端。
2、如果用户已经关注公众号，在用户扫描后会自动进入会话，微信也会将带场景值扫描事件推送到 Foxtable 端。
接下来我们分别介绍生成临时二维码、生成永久二维码、接收二维码推送事件等接口的使用。
第三方二维码
本节介绍的二维码接口，针对的是由微信服务器生成的带场景 ID 的二维码，如果你想通过手机扫描第三方的二维码，并将扫描结果发送到 Foxtable 端，请菜单后面的《菜单接口》这一章。

10.3.8.2 生成临时二维码

临时二维码，是有过期时间的，最长可以设置为在二维码生成后的 30 天（即 2592000 秒）后过期。

首先我们创建一个表，用于记录生成的二维码，表名为 QRCode，结构如下：


查看表结构

列名	类型	长度	标题
Ticket	字符型	96	
ExpireDate	日期时间型		
Descripttion	字符型	32	

生成临时二维码的代码可以参考：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/qrcode/create?access_token={0}"
Dim hc As New HttpClient(CExp(url,Functions.Execute("GetAccessToken")))
Dim jo As New JObject
jo("expire_seconds") = 86400 '有效时间,单位为秒
jo("action_name") = "QR_SCENE" '类型为临时二维码
jo("action_info") = New JObject
jo("action_info")("scene") = New JObject
jo("action_info")("scene")("scene_id") = 112 '指定此二维码的场景 ID
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") Is Nothing Then
```

```
    Dim dr As DataRow = DataTables("QRCode").AddNew()
    dr("Ticket") = jo("ticket")
    dr("ExpireDate") = Date.Now .AddSeconds(jo("expire_seconds"))
    dr("Descripttion") = "二维码说明"
    dr.Save()
Else
    MessageBox.Show(jo("errcode"))
End If
```

二维码生成成功后，会返回一个 ticket，通过这个 ticket 可以获取生成的二维码，格式为：

https://mp.weixin.qq.com/cgi-bin/showqrcode?ticket=返回的 ticket

假定返回的 ticket 为：

gQGj8DwAAAAAAAAAAS5odHRwOi8vd2VpeGluLnFxLmNvbS9xLzAyZWo2QjlHY1Y5N1Uxek92b3hvMUoAAgRyzpZYAwSAUQEA

那么可以通过下面的网址，访问生成的 ticket:

https://mp.weixin.qq.com/cgi-bin/showqrcode?ticket=gQGj8DwAAAAAAAAAAS5odHRwOi8vd2VpeGluLnFxLmNvbS9xLzAyZWo2QjlHY1Y5N1Uxek92b3hvMUoAAgRyzpZYAwSAUQEA

如果要将生成的二维码下载到本机，可以参考代码：

```
Dim ticket As String = "gQGj8DwAAAAAAAAAAS5odHRwOi8vd2VpeGluLnFxLmNvbS9xLzAyZWo2QjlHY1Y5N1Uxek92b3hvMUoAAgRyzpZYAwSAUQEA"
Dim hc As New HttpClient("https://mp.weixin.qq.com/cgi-bin/showqrcode?ticket=" & ticket)
hc.GetFile("c:\data\qrcode.jpg")
```

10.3.8.3 创建永久二维码

永久二维码，是无过期时间的，但数量较少（目前为最多 10 万个）。

创建永久二维码的代码参考：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/qrcode/create?access_token={0}"
Dim hc As New HttpClient(CExp(url,Functions.Execute("GetAccessToken")))
Dim jo As New JObject
jo("action_name") = "QR_LIMIT_SCENE" '类型为永久二维码
jo("action_info") = New JObject
jo("action_info")("scene") = New JObject
jo("action_info")("scene")("scene_id") = 112 '指定此二维码的场景 ID
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
```

```
If jo("errcode") Is Nothing Then
    Dim dr As DataRow = DataTables("QRCode").AddNew()
    dr("Ticket") = jo("ticket")
    dr("Descripttion") = "永久二维码说明"
    dr.Save()
Else
    MessageBox.Show(jo("errcode"))
End If
```

永久二维码的场景 ID 可以是字符串，生成带字符串场景 ID 的永久二维码的代码可以参考：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/qrcode/create?access_token={0}"
Dim hc As New HttpClient(CExp(url,Functions.Execute("GetAccessToken")))
Dim jo As New JObject
jo("action_name") = "QR_LIMIT_STR_SCENE" '永久二维码,场景 ID 为字符串
jo("action_info") = New JObject
jo("action_info")("scene") = New JObject
jo("action_info")("scene")("scene_str") = "字符串形式的场景 ID" '指定此二维码的场景场景 ID
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") Is Nothing Then
    Dim dr As DataRow = DataTables("QRCode").AddNew()
    dr("Ticket") = jo("ticket")
    dr("Descripttion") = "永久二维码说明"
    dr.Save()
Else
    MessageBox.Show(jo("errcode"))
End If
```

永久二维码的获取和下载，方法和临时二维码完全一致，例如下面的代码同时完成了二维码的生成和下载任务：

```
Dim url As String = "https://api.weixin.qq.com/cgi-bin/qrcode/create?access_token={0}"
Dim hc As New HttpClient(CExp(url,Functions.Execute("GetAccessToken")))
Dim jo As New JObject
jo("action_name") = "QR_LIMIT_SCENE" '类型为永久二维码
jo("action_info") = New JObject
jo("action_info")("scene") = New JObject
jo("action_info")("scene")("scene_id") = 112 '指定此二维码的场景ID
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") Is Nothing Then
    Dim dr As DataRow = DataTables("QRCode").AddNew()
    dr("Ticket") = jo("ticket")
    dr("Descripttion") = "永久二维码说明"
    dr.Save()
```

```
    '下载生成好的二维码
    hc = New HttpClient("https://mp.weixin.qq.com/cgi-bin/showqrcode?ticket=" & dr("Ticket"))
    hc.GetFile("c:\data\qrcode.jpg")
Else
    MessageBox.Show(jo("errcode"))
End If
```

10.3.8.4 接收二维码扫描事件

用户扫描带场景 ID 的二维码，会向 Foxtable 端的 HTTP 服务推送事件，事件推送分两种情况：

- 用户已经关注时，推送的是扫描 SCAN，事件参数 EventKey 为二维码的情景 ID，事件参数 Ticket 为二维码的 Ticket。
- 用户未关注时，推送的是关注事件 subscribe，和平常的关注事件不同，此时 subscribe 事件也有包括 EventKey 和 Ticket 参数，需要注意的是 subscribe 事件的 EventKey 由字符串"qrscene_"和二维码的情景 ID 组成。

接收处理二维码扫描事件，可以参考以下 HttpRequest 事件代码：

```
Select Case e.path
    Case "wefox"
        If e.Request.HttpMethod.ToUpper = "GET"
            If Functions.Execute("VerifySignature",e) Then
                e.WriteString(e.GetValues("echostr"))
            End If
        ElseIf e.Request.HttpMethod.ToUpper = "Post"
            Dim xo As XObject = XObject.Parse(e.PlainText)
            Dim st As New Date(1970,1,1,8,0,0)
            Select Case xo("MsgType")
                Case "text","image"
                    '保存和回复文本和图片消息的代码
                Case "event"
                    Select Case xo("Event")
                        Case "subscribe" '关注事件
                            '处理用户关注事件的代码
                            Dim eKey As String = xo("EventKey")
                            If eKey IsNot Nothing AndAlso eKey.StartsWith("qrscene_")Then ' 判
断是否是通过扫描二维码关注
                                Dim openid As String = xo("FromUserName") '获取用户OpenID
                                Dim ticket As String = xo("Ticket") '获取二维码的ticket
                                Dim sceneid As String = eKey.SubString(8) '获取二维码的情景ID,排除前缀qrscene_
                                '被动回复消息
```

```
                    Dim so As New XObject()
                    so("ToUserName") = xo("FromUserName")
                    so("FromUserName") = xo("ToUserName")
                    so("CreateTime") = (Date.Now - st).TotalSeconds()
                    so("MsgType") = "text"
                    so("Content") = "您好,谢谢关注,您扫描的二维码的情景ID为" & sceneid
                    e.WriteString(so.ToXML)
                End If
            Case "unsubscribe" '取消关注事件
                '处理用户取消关注事件的代码'
            Case "LOCATION" '上报地理位置事件
                '处理用户上报地理位置的代码
            Case "TEMPLATESENDJOBFINISH"
                '处理模板消息发送事件的代码
            Case "SCAN"
                Dim openid As String = xo("FromUserName") '获取用户OpenID
                Dim ticket As String = xo("Ticket") '获取二维码的ticket
                Dim sceneid As String = xo("EventKey") '获取二维码的情景ID
                '被动回复消息
                Dim so As New XObject()
                so("ToUserName") = xo("FromUserName")
                so("FromUserName") = xo("ToUserName")
                so("CreateTime") = (Date.Now - st).TotalSeconds()
                so("MsgType") = "text"
                so("Content") = "您好,您扫描的二维码的情景ID为" & sceneid
                e.WriteString(so.ToXML)
        End Select
    End Select
  End If
End Select
```

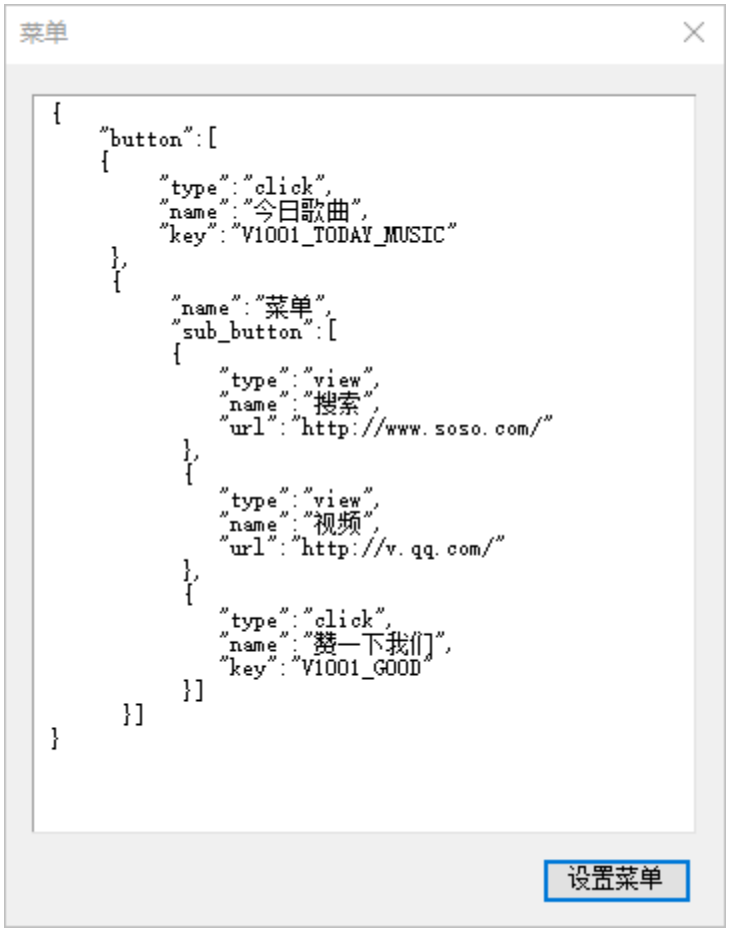
为避免代码冗余，也为了精简 HttpRequest 事件代码，建议将相关代码做成自定义函数。

10.3.9 菜单接口

10.3.9.1 生成默认菜单

关于生成默认菜单接口的说明，请参考：
https://mp.weixin.qq.com/wiki?action=doc&id=mp1421141013&t=0.4168286294440158

直接用代码生成JSON格式的菜单数据比较繁琐，实际开发额时候，可以设计一个窗口，用于输入菜单设置：



窗口的“设置菜单”按钮的代码为：

```
Dim URL As String = " https://api.weixin.qq.com/cgi-bin/menu/create?access_token={0}"
Dim hc As New HttpClient(CExp(URL,Functions.Execute("GetAccessToken")))
hc.Content = e.Form.Controls("TextBox1").Text
Dim jo As JObject = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("菜单设置成功!")
Else
    MessageBox.Show(jo("errcode"))
End If
```

根据表生成

首先设计一个表，名为Menu，用于编辑菜单，结构如下：

	type	name	url	key	media_id	parent
1	view	主页	http://www.foxtable.com			
2		菜单一				
3		菜单二				
4	view	搜索	http://www.soso.com/			菜单一
5	pic_sysphoto	系统拍照发图		rselfmenu_0_0		菜单一
6	pic_photo_or_album	拍照或者相册发		rselfmenu_0_1		菜单一
7	pic_weixin	微信相册发图		rselfmenu_0_2		菜单一
8	click	赞一下我们		rselfmenu_0_3		菜单一
9	scancode_push	扫码推事件		rselfmenu_1_0		菜单二
10	scancode_waitmsg	扫码带提示		rselfmenu_1_1		菜单二
11	location_select	发送位置		rselfmenu_1_2		菜单二
12	media_id	下载图片			eHDdfBQmRt0J-PtkD6WENicqBE7-tFFczZ6p5ZUnz0k	菜单二
13	view_limited	下载图文			eHDdfBQmRt0J-PtkD6WENpJyI8iZ7Qy2l5SvQS58Jz0	菜单二

我根据接口说明，在表中输入了一些测试数据(parent列用于输入父菜单的name属性，如果parent为空，表示这是一级菜单)，涵盖了所有十种菜单按钮类型，对应的菜单如下图：


中国联通
05:40
微信(1)
wxid_x5ezy8eqrx9312...
搜索
系统拍照发图
拍照或者相册发图
微信相册发图
赞一下我们
主页
菜单一
菜单二



然后我们在Foxtable设计一个按钮用于生成菜单，按钮代码如下：

```
Dim mrs As List(of DataRow)  = DataTables("Menu").Select("[parent] is null","_SortKey") '删选出一级按钮对应的行
Dim mo As New JObject  '一级菜单对象
Dim ma As New Jarray '一级菜单按钮数组
mo("button") = ma '一级菜单属性就一个button属性,这是一个数组属性
For Each mr As DataRow In mrs
```

```
    Dim so As New JObject
    so("name") = mr("name").Tostring
    Dim srs As List(of DataRow) = DataTables("Menu").Select("[parent] = '" & mr("name") & "'", "_SortKey") '筛选出二级按钮
    Dim nms As String() = {"type","name","url","key","media_id"} '注意列名和菜单属性名一直,方便编码,注意大是区分大小写的
    If srs.Count = 0 Then '如果此一级菜单按钮对应的二级按钮数为0,那么次一级菜单按钮就是一个普通的命令按钮
        For Each nm As String In nms '设置一级菜单按钮的户型
            If mr.IsNull(nm) = False Then
                so(nm) = mr(nm).ToString()
            End If
        Next
    Else '如果此一级菜单按钮有对应的二级菜单按钮,那么此一级菜单按钮是一个菜单按钮
        Dim sa As New JArray '定义一个数组,包括此一级按钮下的所有二级按钮
        For Each sr As DataRow In srs '逐个生成二级按钮
            Dim jo As New JObject '定义二级按钮对象
            For Each nm As String In nms '设置二级按钮属性
                If sr.IsNull(nm) = False Then
                    jo(nm) = sr(nm).ToString()
                End If
            Next
            sa.Add(jo)
        Next
        so("sub_button") = sa '一级菜单按钮的sub_button属性是一个数组,包括所有二菜单按钮
    End If
    ma.Add(so)
Next
Dim URL As String = " https://api.weixin.qq.com/cgi-bin/menu/create?access_token={0}"
Dim AccessToken As String = Functions.Execute("GetAccessToken")
Dim hc As New HttpClient(CExp(URL,AccessToken))
hc.Content = mo.Tostring()
Dim ro As JObject = JObject.Parse(hc.GetData)
If ro("errcode") = "0" Then
    MessageBox.Show("菜单设置成功!")
Else
    MessageBox.Show(ro("errcode"))
End If
```

显然，通过数据表来编写菜单，要轻松很多。

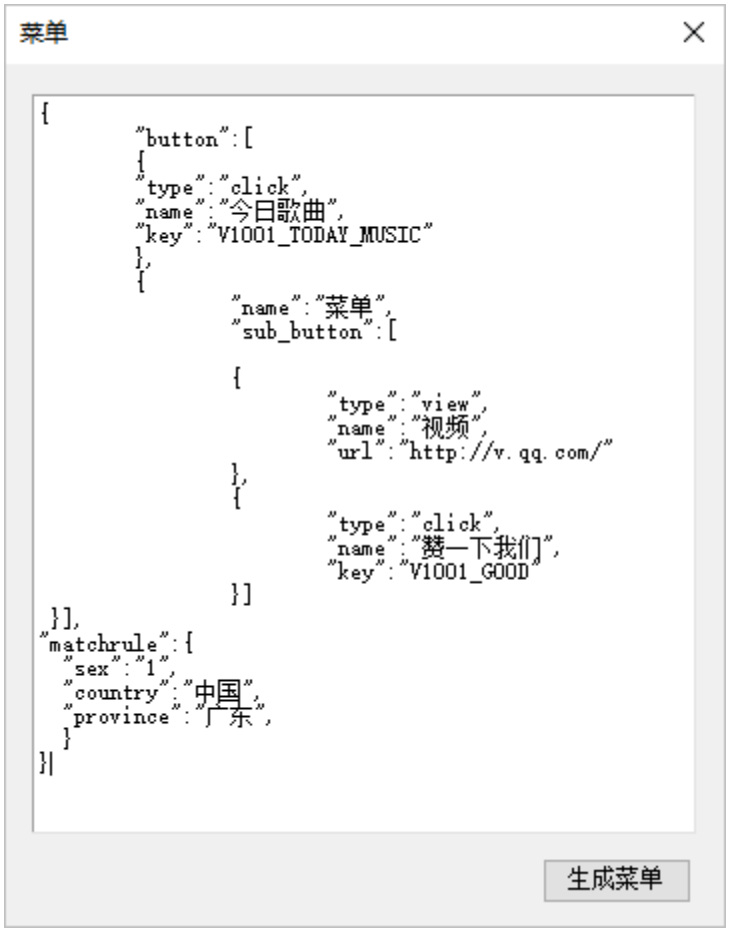
10.3.9.2 生成个性化菜单

关于生成个性化菜单接口的说明，请参考：

https://mp.weixin.qq.com/wiki?action=doc&id=mp1455782296&t=0.16927251183776681

生成个性化菜单

直接用代码生成 JSON 格式的菜单数据比较繁琐，实际开发额时候，可以设计一个窗口，用于输入菜单设置：



生成菜单按钮的代码为：

```
Dim URL As String = "https://api.weixin.qq.com/cgi-bin/menu/addconditional?access_token={0}"
Dim hc As New HttpClient(CExp(URL,Functions.Execute("GetAccessToken")))
hc.Content = e.Form.Controls("TextBox1").Text
Dim jo As JObject = JObject.Parse(hc.GetData)
If jo("errcode") Is Nothing Then
    MessageBox.Show("菜单设置成功, menuid 为: " & jo("menuid").ToString())
Else
    MessageBox.Show(jo("errcode"))
End If
```

其实个性化菜单和默认菜单基本没有差别，你设置了菜单匹配条件的就是个性化菜单，否则就是默认菜单，此外个性化菜单生成后会返回一个 menuid。

查询某用户匹配的个性化菜单

个性化菜单可以定义多个，如果要查询某个用户匹配的个性化菜单，可以参考下面的代码：

```
Dim URL As String = "https://api.weixin.qq.com/cgi-bin/menu/trymatch?access_token={0}"
Dim hc As New HttpClient(CExp(URL,Functions.Execute("GetAccessToken")))
Dim jo As New JObject
jo("user_id") = "oQdguw1qKnHQu5oIew-osH6JSIZA" '用户微信号或 OpenID
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") Is Nothing Then
    MessageBox.Show(jo.ToString)
Else
    MessageBox.Show(jo("errcode"))
End If
```

10.3.9.3 接收菜单事件

单击任何自定义菜单按钮，微信服务器都会向 Foxtable 的 HTTP 服务推送一个事件，即使是直接跳转到网页的按钮，也一样会推送事件。

需要注意的是，所有向服务器传送图片的菜单命令，会发送两次消息，第一次是菜单命令事件，第二次是普通的 image 消息，第二次才会真正将图片的 MediaID 和 URL 传递给 Foxtable。

接收并处理菜单事件，可以参考以下 HttpRequest 事件代码，这段代码将接收到的菜单事件记录在一个日志文件中：

```
Select Case e.path
   Case "wefox"
      If e.Request.HttpMethod.ToUpper = "GET"
         If Functions.Execute("VerifySignature",e) Then
            e.WriteString(e.GetValues("echostr"))
         End If
      ElseIf e.Request.HttpMethod.ToUpper = "Post"
         Dim xo As XObject = XObject.Parse(e.PlainText)
         Dim st As New Date(1970,1,1,8,0,0)
         Select Case xo("MsgType")
            Case "text","image","voice","video","shortvideo"
               '保存收到的消息
               Dim dr As DataRow = DataTables("Message").AddNew()
               dr("FromUserName") = xo("FromUserName")
               dr("CreateTime") = st.AddSeconds(xo("CreateTime"))
               dr("MsgType") = xo("MsgType")
               dr("MsgId") = xo("MsgId")
               dr("MediaId") = xo("MediaId")
               dr("PicUrl") = xo("PicUrl")
```

```
                dr("Content") = xo("Content")
                dr.Save()
            Case "event"
                Dim OpenId As String = xo("FromUserName")
                Dim key As String = xo("EventKey")
                Dim CreateTime As Date = st.AddSeconds(xo("CreateTime")) '触发事件时间
                Dim logFile As String  = "c:\data\location" & Format(Date.Today,"yyyyMMdd") & ".txt" '你可以选择每日或每月一个日志文件
                Select Case xo("Event")
                    Case "subscribe" '关注事件
                        '处理用户关注事件的代码
                    Case "unsubscribe" '取消关注事件
                        '处理用户取消关注事件的代码'
                    Case "LOCATION" '自动上报地理位置事件
                        '处理自动上报地理位置的代码
                    Case "TEMPLATESENDJOBFINISH"
                        '处理模板消息发送事件的代码
                    Case "SCAN"
                        '处理扫描情景二维码事件的代码
                    Case "CLICK" '普通单击事件
                        Filesys.WriteAllText(logFile , OpenId & "于" & CreateTime & "单击按钮:" & key & vbcrlf, True)
                    Case "VIEW" '单击跳转到网页事件
                        Filesys.WriteAllText(logFile , OpenId & "于" & CreateTime & "访问网页:" & key & vbcrlf, True)
                    Case "scancode_push","scancode_waitmsg" '两种扫描二维码事件
                        Dim ScanResult As String = xo("ScanCodeInfo")("ScanResult") '获取二维码扫描结果
                        Filesys.WriteAllText(logFile , OpenId & "于" & CreateTime & "扫描二维码:" & ScanResult & vbcrlf, True)
                    Case "pic_sysphoto","pic_photo_or_album","pic_weixin" '三种发送图片事件
                        Dim Count As Integer = xo("SendPicsInfo")("Count") '获取图片数量
                        Filesys.WriteAllText(logFile , OpenId & "于" & CreateTime & "提交图片数:" & Count & vbcrlf, True)
                    Case "location_select" '主动上报地理位置事件
                        Dim lx As String = xo("SendLocationInfo")("Location_X")
                        Dim ly As String = xo("SendLocationInfo")("Location_Y")
                        Dim lb As String = xo("SendLocationInfo")("Label")
                        Dim pn As String = xo("SendLocationInfo")("Poiname")
                        Filesys.WriteAllText(logFile , OpenId & "于" & CreateTime & "上报位置:" & lx &"|" & ly & "|" &  lb & "|" & pn & vbcrlf, True)
                End Select
            End Select
        End If
End Select
```

10.3.9.4 查询自定义菜单

查询自定义菜单（包括默认菜单和个性化菜单）的接口说明，请参考：
https://mp.weixin.qq.com/wiki?action=doc&id=mp1421141014&t=0.2272236121056963
参考代码：

```
Dim ur As String = "https://api.weixin.qq.com/cgi-bin/menu/get?access_token={0}"
Dim hc As new HttpClient(Cexp(ur,Functions.Execute("GetAccessToken")))
FileSys.WriteAllText("c:\data\men.txt",hc.GetData,False)
```

执行以上代码，可以将菜单配置保存为一个文本文件。
如果希望生成的文本文件可读性更强，可以用 JObject 格式化一下：

```
Dim ur As String = "https://api.weixin.qq.com/cgi-bin/menu/get?access_token={0}"
Dim hc As new HttpClient(Cexp(ur,Functions.Execute("GetAccessToken")))
Dim jo As JObject = JObject.Parse(hc.GetData)
FileSys.WriteAllText("c:\data\men.txt", jo.ToString, False)
```

10.3.9.5 菜单的删除与修改

删除所有菜单
删除菜单的接口说明，请参考：

https://mp.weixin.qq.com/wiki?action=doc&id=mp1421141015&t=0.19727748596737232

代码很简单：

```
Dim ur As String = "https://api.weixin.qq.com/cgi-bin/menu/delete?access_token={0}"
Dim hc As new HttpClient(Cexp(ur,Functions.Execute("GetAccessToken")))
Dim jo As New JObject
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("菜单删除成功")
Else
    MessageBox.Show(jo("errcode"))
End If
```

删除个性化菜单

删除个性化菜单的接口说明，请参考：

https://mp.weixin.qq.com/wiki?action=doc&id=mp1455782296&t=0.7361369594310105

代码参考：

```
Dim ur As String = "https://api.weixin.qq.com/cgi-bin/menu/delconditional?access_token={0}"
Dim hc As new HttpClient(Cexp(ur,Functions.Execute("GetAccessToken")))
Dim jo As New JObject
jo("menuid") = 407808988 '要删除的个性化菜单 ID
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("菜单删除成功")
Else
    MessageBox.Show(jo("errcode"))
End If
```

关于修改菜单

修改默认菜单很简单，重新发布一次即可。

个性化菜单是没有办法修改的，即使重新发布一个匹配条件完全一样的个性化菜单，也不会覆盖原来的菜单，而是生成一个新的菜单，你可以从返回的 MenuID 判断得出这个结论。

微信服务器为用户匹配个性话菜单的时候，在匹配条件完全一致的情况下，优先匹配最新发布的菜单。

10.3.9.6 用数据表管理多个菜单

普通用户可以忽略本节内容。

自行编写 JSON 格式的菜单，是一件比较繁琐和容易出错的工作，特别是有大量的个性化菜单需要维护的时候。

我们也可以用数据表管理和生成菜单，之前我们在介绍生成默认菜单的时候，已经提供了一个简单的例子。

本节的例子更进一步，可以通过数据表管理和生成多个菜单，包括默认菜单和数量不等的个性化菜单。

1、首先我们设计一个表，名为 Menus，表结构如下：

	menu	matchrule	MenuID
1	Default		
2	Menu1	tag_id=2&sex=1	407835285
3	Menu2	country=中国&province=广东	407835289

menu 列为菜单名称。

matchrule 列为菜单的匹配条件，为了方便输入，我们这里模仿了 get 数据格式"键 1=值 1&键 2=值 2"。

如果 matchrule 为空，也就是没有设置匹配条件，表示这是默认菜单。

默认菜单只有一个，虽然你可以设置多个默认菜单，但是新生成的会覆盖之前生成的。

个性化菜单可以有多个，微信服务器生成个性化菜单生成之后，会返回一个 MenuID，我们需要将此值保存在 MenuID 列，方便维护。

2、再设计一个名为 MenuButtons 的表，用于输入菜单按钮，表结构为：

	type	name	url	key	media_id	parent	Menu
1	view	主页	http://www.foxtable.com/				Default
2		菜单一					Default
3		菜单二					Default
4	view	搜索	http://www.soso.com/			菜单一	Default
5	pic_sysphoto	系统拍照发图		rselfmenu_0_0		菜单一	Default
6	click	赞一下我们		rselfmenu_0_1		菜单一	Default
7	scancode_push	扫描二维码		rselfmenu_1_0		菜单二	Default
8	location_select	发送位置		rselfmenu_1_1		菜单二	Default
9	view_limited	下载图文			eHDdfBQmRtOJ-PtkD6WENpJyI8iZ7Qy2	菜单二	Default
10	click	今日歌曲		V1001_TODAY_MUSIC			Menu1
11		菜单					Menu1
12	view	搜索	http://www.soso.com/			菜单	Menu1
13	view	视频	http://v.qq.com/			菜单	Menu1
14	click	赞一下我们	V1001_GOOD	V1001_GOOD		菜单	Menu1
15	click	今日头条		V1002_TODAY_TOUTAIO			Menu2
16		菜单					Menu2
17	view	搜索	http://www.baidu.com/			菜单	Menu2
18	view	视频	http://v.qq.com/			菜单	Menu2
19	click	赞一下我们	V1001_GOOD	V1002_GOOD		菜单	Menu2

Menu 列和上一个表的 menu 列对应，用于输入此按钮所属的菜单。

菜单按钮有一级和二级之分，parent 列用于输入上菜单按钮名称，如果不设置，表示此按钮为一级菜单按钮。

type 为 view 的按钮，必须输入 url 值；type 为 media_id 或 view_limited 的按钮，必须输入 media_id 值(素材 ID)；其它类型的按钮，必须输入 key 值，否则会出错。

上面两个表定义了三个菜单，default 为默认菜单，Menu1 和 Menu2 为个性化菜单。

提示：所有列名都必须和接口说明中的属性名保持一致（包括大小写），以方便编码。
3、最后在菜单或窗口设计一个按钮，用于基于上面两个表生成菜单，代码为：

```
'首先要删除以前的菜单,这是必须的,否则个性化菜单会越来越多.
Dim ur As String = "https://api.weixin.qq.com/cgi-bin/menu/delete?access_token={0}"
Dim hc As new HttpClient(Cexp(ur,Functions.Execute("GetAccessToken")))
Dim RO As New JObject
RO = JObject.Parse(hc.GetData)
If RO("errcode") <> "0" Then
    MessageBox.Show(RO("errcode"))
```

```
    Return
End If
'基于数据表生成菜单.
For Each menuRow As DataRow In DataTables("Menus").DataRows 'Menu 表中每一行都对应一个菜单
    Dim FRows As List(of DataRow) = DataTables("MenuButtons").Select(CExp("menu='{0}' And [parent] is null",menuRow("menu")),"_SortKey")
    Dim nms As String() = {"type","name","url","key","media_id"}
    Dim FButtons As New Jarray
    For Each FRow As DataRow In FRows '生成一级菜单
        Dim SRows As List(of DataRow)
        SRows = DataTables("MenuButtons").Select(CExp("menu='{0}' And [parent] = '{1}'", menuRow("menu"), FRow("name")),"_SortKey")
        Dim FButton As New JObject
        If sRows.Count = 0 Then
            For Each nm As String In nms
                If FRow.IsNull(nm) = False Then
                    FButton(nm) = FRow(nm).ToString()
                End If
            Next
        Else
            Dim SButtons As New JArray
            For Each SRow As DataRow In SRows '生成二级菜单
                Dim SButton As New JObject
                For Each nm As String In nms
                    If SRow.IsNull(nm) = False Then
                        SButton(nm) = SRow(nm).ToString()
                    End If
                Next
                SButtons.Add(SButton)
            Next
            FButton("name") = FRow("name").ToString()
            FButton("sub_button") = SButtons
        End If
        FButtons.Add(FButton)
    Next
    Dim MO As New JObject
    MO("button") = FButtons
    If MenuRow.IsNull("matchrule") = False '如果设置了过滤条件,那就是个性化菜单了
        Dim FO As New JObject
        Dim fts() As String = MenuRow("matchrule").Split("&")
        For Each ft As String In fts
            Dim kv() As String = ft.Split("=")
            FO(kv(0)) = kv(1)
        Next
```

```
        MO("matchrule") = FO
        ur = "https://api.weixin.qq.com/cgi-bin/menu/addconditional?access_token={0}"
    Else
        ur = "https://api.weixin.qq.com/cgi-bin/menu/create?access_token={0}"
    End If
    hc = New HttpClient(CExp(ur,Functions.Execute("GetAccessToken")))
    hc.Content = MO.ToString()
    RO = JObject.Parse(hc.GetData)
    If  RO("menuid") IsNot Nothing Then
        menuRow("MenuID") = RO("menuid")
    ElseIf RO("errcode") <> "0" Then
        MessageBox.Show(RO("errcode"))
    End If
Next
```

10.3.10 网页授权接口

10.3.10.1 网页简单授权

一般用户可以忽略本节内容。

关于微信网页授权的接口说明，请参考：

https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140842&t=0.06929239949617943

微信网页授权有两种，一种只能获取用户的 openid，另一种可以获取用户的详细信息，为了区分，我们称前者为简单授权，称后者为详情授权。

这一节我们介绍简单授权，简单授权只能获取用户的 openid，适合企业内部的管理系统（实际上，企业内部管理系统更适合用企业号，而不是公众号）。

简单网页授权 的一般流程：
1、将目标网页的地址包装在一个授权链接中，这个链接是指向腾讯的授权服务器的，格式如下：

https://open.weixin.qq.com/connect/oauth2/authorize?appid=APPID&redirect_uri=REDIRECT_URI&response_type=code&scope=snsapi_base&state=STATE#wechat_redirect

红色部分是需要我们合成进去的：

APPID	开发者 ID。
REDIRECT_URI	目标网页地址，注意： 1、需要用 UrlEncode 函数进行处理后再合成进去。

	2、一定要完整的路径。 你可参考下面的代码生成授权 URL： Dim ul1 As String = "https://open.weixin.qq.com/connect/oauth2/authorize?appid={0}&redirect_uri={1}&response_type=code&scope=snsapi_base&state={2}#wechat_redirect" Dim ul2 As String = UrlEncode("http://wexin.foxtable.com") ul1 = CExp(ul1,"wx8acdb7df5beb68fd",ul2,"123") Output.Show(ul1)
STATE	自定义参数，重定向后会在目标网页地址中用 get 方式附上此参数。

2、用户打开目标网页，后台在 cookie 中检查是否存在登录信息，如果存在，直接显示目标网页内容。

3、如果未在 cookie 中发现登录信息，则引导用户跳转到第一步生成的授权链接。
4、微信服务器收到这个 授权链接请求后，会生成一个 CODE，然后将 CODE 和上面的 STATE 参数以 get 形式附加到目标网页地址中,格式为：

redirect_uri/?code=CODE&state=STATE

经过授权链接中转后，用户最终访问的就是以上链接，等于又回到了目标网页，只是多了 CODE 和 STATE 参数。

5、Foxtable 收到上述访问请求后，根据 CODE 参数调用微信接口获取访问者的 openid，再根据 openid 判断用户是否有权访问系统。

看起来步骤很多，事件编码却很简单。
一个简单例子

1、在微信后台将网页授权域名设置为"wexin.foxtable.com"，注意不能加"http"。

2、修改 WXUsers 表的结构，增加一个 permit 列，用于标记某个用户是否允许访问系统:

	openid	nickname	sex	city	country	province	headimgurl	groupid	remark	language	tagid_lis	permit
1	ofjtFwBSZ5cNq	老挨踢	1	湛江	中国	广东	http://wx.qlo	0		zh_CN	[]	☑
2	ofjtFwGvPBfy9	老码农	0				http://wx.qlo	104		zh_CN	[104]	☐
3	ofjtFwELCpcSf	□□K	2	广州	中国	广东	http://wx.qlo	0		zh_CN	[]	☑
4	ofjtFwCb5pHrW	哀月的羽	2	湛江	中国	广东	http://wx.qlo	0		zh_CN	[]	☐
5	ofjtFwIAgbhy6	黄博	1	湛江	中国	广东	http://wx.qlo	106		zh_CN	[106]	☐

3、Foxtable 的 HttpRequest 事件代码设置为：

```
If e.host = "wexin.foxtable.com" Then '需要授权才能访问的域名
    Dim OpenID As String
    Dim sb As New StringBuilder
```

```
    sb.AppendLine("<meta name='viewport' content='width=device-width,initial-scale=1,user-scalable=1'>")
    If e.GetValues.ContainsKey("code") Then '如果通过授权链接跳转而来,就根据传递过来的 code 参数调用接口,获取用户的 OpenID
                Dim ul As String = "https://api.weixin.qq.com/sns/oauth2/access_token?appid={0}&secret={1}&code={2}&grant_type=authorization_code"
        ul = CExp(ul,"wx8acdb7df5beb68fd","a5a3b06a790d4200a151e6b13c3dc263",e.GetValues("code"))
        Dim hc As new HttpClient(ul)
        Dim jo As JObject = JObject.Parse(hc.GetData)
        If jo("openid") IsNot Nothing Then
            OpenID = jo("openid")
            e.AppendCookie("openid",OpenID) '将 openid 存储在 Cookie 中
        End If
    Else
        OpenId =  e.Cookies("openid") '否则从 cookie 中提取 openid
    End If
    Dim Verified As Boolean
    Dim dr As DataRow = DataTables("WXUsers").Find("openid  ='" & OpenID & "'") '根据 openid 找出对应的行
    If OpenId > "" AndAlso dr IsNot Nothing AndAlso dr("permit") = True '授权成功
        Verified  = True
    ElseIf e.GetValues.ContainsKey("code") = False Then '如果授权失败,且不是通过授权链接跳转而来,那么就跳转到授权链接
                Dim ul As String = "https://open.weixin.qq.com/connect/oauth2/authorize?appid=wx8acdb7df5beb68fd&redirect_uri=http%3a%2f%2fwexin.foxtable.com&response_type=code&scope=snsapi_base&state=123#wechat_redirect"
        sb.Append("<meta http-equiv='Refresh' content='0; url=" & ul & "'>") '跳转到授权链接
        e.WriteString(sb.ToString)
        Return
    End If
    If Verified = False Then
        sb.AppendLine("你无权访问本系统")
    Else
        sb.AppendLine("Foxtable 欢迎您, <a href='http://wexin.foxtable.com'>刷新页面</a>")
    End If
    e.WriteString(sb.ToString)
End If
```

现在只有只有经过授权的微信账户，才能通过访问"http://wexin.foxtable.com",其它微信用户或者直接通过浏览器输入地址访问的用户，都将被拒绝。

10.3.10.2 网页详情授权

一般用户可以忽略本节内容。

关于微信网页授权的接口说明，请参考：

https://mp.weixin.qq.com/wiki?action=doc&id=mp1421140842&t=0.06929239949617943

微信网页授权有两种，一种只能获取用户的 openid，另一种可以获取用户的详细信息，为了区分，我们称前者为简单授权，称后者为详情授权。

这一节我们详情授权，详情授权能获取用户的所有信息，适合公开对外的系统，不管用户是否已经关注过微信号，都能获取用户详情。

如果用户已经关注过公众号，那么详情授权和简单授权一样，都是静默方式的，如果用户没有关注公众号，那么会出现一个提示页面，只有用户同意授权之后，我们才能获取用户的信息：



详情授权流程：

1、将目标网页的地址包装在一个授权链接中，这个链接是指向腾讯的授权服务器的，格式如下：

https://open.weixin.qq.com/connect/oauth2/authorize?appid=APPID&redirect_uri=REDIRECT_URI&response_type=code&scope=snsapi_userinfo&state=STATE#wechat_redirect

红色部分是需要我们合成进去的：

APPID	开发者 ID。

REDIRECT_URI	目标网页地址，注意： 1、需要用 UrlEncode 函数进行处理后再合成进去。 2、一定要完整的路径。 你可参考下面的代码生成授权 URL： Dim ul1 As String = "https://open.weixin.qq.com/connect/oauth2/authorize?appid={0}&redirect_uri={1}&response_type=code&scope=snsapi_userinfo&state={2}#wechat_redirect" Dim ul2 As String = UrlEncode("http://wexin.foxtable.com") ul1 = CExp(ul1,"wx8acdb7df5beb68fd",ul2,"123") Output.Show(ul1) 详情授权链接的 scope 参数为 snsapi_userinfo，而简单授权链接的 scope 参数为 snsapi_base，二者无其它差别。
STATE	自定义参数，重定向后会在目标网页地址中用 get 方式附上此参数。

2、用户打开目标网页，后台在 cookie 中检查是否存在登录信息，如果存在，直接显示目标网页内容。

3、如果未在 cookie 中发现登录信息，则引导用户跳转到第一步生成的授权链接。
4、微信服务器收到这个授权链接请求后，会生成一个 CODE，然后将 CODE 和上面的 STATE 参数以 get 形式附加到目标网页地址中,格式为：

redirect_uri/?code=CODE&state=STATE

经过授权链接中转后，用户最终访问的就是以上链接，等于又回到了目标网页，只是多了 CODE 和 STATE 参数。

5、Foxtable 收到上述访问请求后，根据 CODE 调用微信接口获取 openid 和 access_token（注意这不是普通 access_token，此 access_token 只能用于网页授权）。

6、根据 openid 判断此用户是否已经存在 WXUsers 表中：

- 如果存在，授权过程结束，用户直接进入系统；
- 如果不存在，则根据第 5 步获取的 access_token 和 openid 调用接口，获取用户详情登记在 WXUsers 表中后，允许用户进入系统。

可以看到，和简单授权性比，详情授权多了第 6 步，也就用获取用户详情的这一步。

看起来步骤很多，事件编码并不复杂。

一个简单例子

1、在微信后台将网页授权域名设置为"wexin.foxtable.com"，注意不能加"http"。

2、HttpRequest 事件代码如下：

```
Dim sb As New StringBuilder
sb.AppendLine("<meta name='viewport' content='width=device-width,initial-scale=1,user-scalable=1'>")
If e.host = "wexin.foxtable.com" Then '授权测试
    Dim UserName As String
    Dim OpenID As String
    If e.GetValues.ContainsKey("code") Then '如果是通过授权链接跳转而来,就从链接重提取 code 来获取 openid
        Dim ul As String = "https://api.weixin.qq.com/sns/oauth2/access_token?appid={0}&secret={1}&code={2}&grant_type=authorization_code"
        ul = CExp(ul,"wx8acdb7df5beb68fd","dec0df8a469518df30e9bdef0758c678",e.GetValues("code"))
        Dim hc As new HttpClient(ul)
        Dim jo As JObject = JObject.Parse(hc.GetData)
        If jo("openid") IsNot Nothing Then '如果获取 openid 成功(成功的话,还会同时返回一个 accesstiken,用于获取用户详情)
            OpenID = jo("openid")
            Dim dr As DataRow = DataTables("WXUsers").Find("openid ='" & Openid & "'")
            If dr IsNot Nothing Then
                UserName = dr("nickname")
            Else
                ul = "https://api.weixin.qq.com/sns/userinfo?access_token={0}&openid={1}&lang=zh_CN "
                '根据 openid 和 accesstoken 获取用户详情,注意这里这个 accesstoken 不是普通 accesston,只能用于网页授权
                hc = New HttpClient(CExp(ul, jo("access_token"), OpenId))
                jo = jo.Parse(hc.GetData)
                If jo("openid") IsNot Nothing Then
                    UserName = jo("nickname")
                    dr = DataTables("WXUsers").AddNew()
                    Dim nms() As String = {"openid","nickname","sex","city","country","province","headimgurl"} '''''
                    For Each nm As String In nms
                        dr(nm) = jo(nm)
                    Next
                    dr.Save
                Else
                    e.WriteString(jo.ToString) '在用户浏览器显示错误信息
                    Return
                End If
            End If
            e.AppendCookie("username",UserName) '用户名和 openid 存储在 Cookie 中
            e.AppendCookie("openid",OpenID)
        Else
            e.WriteString(jo.ToString) '在用户浏览器显示错误信息
            Return
```

```
        End If
    Else
        UserName = e.Cookies("username") '从 cookie 获取用户名和 openid
        OpenID = e.Cookies("openid")
        If userName = "" OrElse OpenID = "" Then
                                Dim                ul                As                String                =
"https://open.weixin.qq.com/connect/oauth2/authorize?appid=wx8acdb7df5beb68fd&redirect_uri=http%3a%2f%2f
wexin.foxtable.com&response_type=code&scope=snsapi_userinfo&state=123#wechat_redirect"
            sb.Append("<meta http-equiv='Refresh' content='0; url=" & ul & "'>") '跳转到授权链接
            e.WriteString(sb.ToString)
            Return
        End If
    End If
    If  OpenID > "" And UserName > "" Then
        '这里可以做进一步的权限判断
        sb.AppendLine("欢迎" & UserName & "光临, <a href='http://wexin.foxtable.com'>刷新页面</a>")
    Else
        sb.AppendLine("你无权访问本系统")
    End If
End If
e.WriteString(sb.ToString)
```

10.3.11 在网页实现扫码输入数据

在网页实现扫描二维码或条形码输入数据，需要借助 JS-SDK。

关于公众号 JS-SDK 的说明参考:

https://mp.weixin.qq.com/wiki?action=doc&id=mp1421141115&t=0.8138827698946394

可以看到 JS-SDK 的内容非常多，对于 Foxtable 用户而言，最有用的莫过于在网页实现扫描二维码或条形码输入 （或提交）数据的功能了，下面我们就以此功能为例，介绍一下 JS-SDK 的使用方法。

本节的任务是设计一个下图所示的页面(只能在微信中浏览此页面)，点击“扫码输入”后，能调用微信扫码功能，对二维码或条形码进行扫描，并将扫描结果填入编号输入框：



提示：本节的例子只能在微信访问。

设计步骤

1、在微信管理后台设置 JS 安全域名，只能在该域名下调用 JS-SDK，测试号可以使用 IP，正式的公众号只能使用经过备案的域名 ；我们必须通过设置好的安全域名访问网页，才能调用 JS-SDK。

2、新增一个自定义函数，函数名为"GetJsSignature"，用于生成 JS-SDK 的授权签名，代码为：

```
Static CreateTime As Date '记录最近一次生成 Ticket 的时间
Static Ticket As String '记录最近一次生成的 Ticket
Dim tp As TimeSpan = Date.Now - CreateTime
Dim ul As String = "https://api.weixin.qq.com/cgi-bin/ticket/getticket?access_token={0}&type=jsapi"
If tp.TotalSeconds > 3600 Then
    Dim hc As New HttpClient(CExp(ul,Functions.Execute("GetAccessToken")))
```

```
    Dim ret As String = hc.GetData()
    If ret = "" Then '如果失败,再尝试一次
        hc.GetData()
    End If
    CreateTime = Date.Now()
    Dim jo As JObject = JObject.Parse(ret)
    If jo("errcode") = "0" Then
        Ticket = jo("ticket")
    Else
        MessageBox.show("获取 jsapi_ticket,原因:" & vbcrlf & jo.ToString)
    End If
End If
Dim signature As String = CExp("jsapi_ticket={0}&noncestr={1}&timestamp={2}&url={3}",Ticket,args(0),args(1),args(2))
Return Web.Security.FormsAuthentication.HashPasswordForStoringInConfigFile(signature, "SHA1").ToLower()
```

2、在 d:\web\lib 目录下新建一个文本文件 jssdk.js，内容为：

```
wx.ready(function () {
    document.getElementById('scan').onclick = function () {
        wx.scanQRCode({
            needResult: 1,
            scanType: ['qrCode','barCode'],
            success: function (res) {
                document.getElementById('number').value = res.resultStr;
            }
        });
    };
});
wx.error(function (res) {
    //alert(res.errMsg);
});
```

实际开发的时候，你要变化的只有红色的两处，scan 为网页中扫码输入按钮的 id，number 为网页中产品编号输入框的 id。

3、最后将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "test.htm",""
        Dim wb As New weui
        wb.AppendHTML("<script  src='http://res.wx.qq.com/open/js/jweixin-1.2.0.js'></script>",True) '引入 JS-SDK 库
        wb.AppendHTML("<script src='./lib/jssdk.js'></script>") '引入脚本文件
        '在页面注入权限验证配置
        Dim st As New Date(1970,1,1,8,0,0)
        Dim appid As String = "wx8acdb7df5beb68fd" '开发者 ID
```

```
    Dim timestamp As Integer = CInt((Date.Now - st).TotalSeconds()) '时间戳
    Dim noncestr As String = Rand.NextString(16) '随机字符
    Dim url As String  = e.Request.URL.ToString '当前页面地址
    Dim signature As String = Functions.Execute("GetJsSignature", noncestr, timestamp, url) '生成权限验证签名
                Dim cfg As String = "wx.config({appId:'{0}',timestamp:{1},nonceStr:'{2}',signature:'{3}',jsApiList:['scanQRCode']});"
    wb.AppendHTML("<script>" & CExp(cfg,appid,timestamp,noncestr,signature) & "</script>",True)
    '开始正常生成网页内容
    wb.AddForm("","form1","test.htm")
    With wb.AddInputGroup("form1","ipg1","产品输入")
      .AddInput("product","产品","text")
      With .AddInputCell("ic1")
        .AddLabel("lbh","编号",0)
        .AddInput("number","text",1)
        .AddVcodeButton("scan","扫码输入",2) '增加二维码扫描按钮,2 表示显示在右边
      End With
    End With
    With wb.AddButtonGroup("form1","btg1",True)
      .Add("btn1", "确定", "submit")
    End With
    e.WriteString(wb.Build) '生成网页
End Select
```

至此，我们的网页扫码输入功能就算设置完成，可以看到 JS-SDK 的使用并不复杂。

如果你希望直接将扫码得到的结果发送到 Foxtable 进行处理，而不是填入输入框，可以参考：SendAjaxText

当然你也可以根须扫描结果合成新的页面地址访问之。

10.3.12 中控服务器的设计

我们知道公众号可以通过 AppID 和 AppSecret 生成 AccessToken，用于调用公众号接口的 身份验证凭据。

和企业号不同，公众号的 AccessToken 每刷新一次，都会返回不不同的结果，如果你有多个电脑处理微信业务，这将引发冲突，导致程序不稳定。

例如假定 A 电脑已经刷新并缓存了 AccessToken（因为 AccessToken 一天只能获取 2000 次，所以必须缓存），在未来 1 个小时内都不再刷新 AccessToken，而是直接使用缓存的 AccessToken。

如果后来 B 电脑开始处理业务，于是 B 电脑刷新并缓存 AccessToken，这将导致 A 电脑已经缓存的 AccessToken 失效，使得 A 电脑无法继续处理业务。

显然如果有多个电脑处理业务，只能由其中一台电脑负责刷新并缓存 AccessToken，其它业务电脑

都向此电脑查询获得 AccessToken，这台统一负责刷新和缓存 AccessToken 的电脑， 称为中控服务器。
中控服务器的设计，你可以考虑以下三种方案之一：

1. 在中控服务器开启一个计划，每小时刷新一次 AccessToken，将最新的 Access 保存在数据库中，其它电脑通过查询数据库获取 AccessToken。
2. 在中控服务器开启 QQServer 或 UDPClient，其它电脑发送约定格式的字符串到中控服务器，由中控服务器返回 GetAccessToken 函数的执行结果。
3. 在中控服务器开启 HttpServer，其它电脑通过 HttpClient 发送访问请求到约定地址，由中控服务器返回 GetAccessToken 函数的执行结果。

以上说的是多个电脑处理微信业务，如果是同一台电脑的多个项目处理微信业务，一样需要将其中一个项目作为中控项目，统一负责 AccessToken 的刷新和缓存。
中控服务器（或中控项目）的设计非常简单，以第三种方案为例：

1、在中控服务器 （或中控项目）新建一个自定义函数，名称为 GetAccessToken，用于刷新和缓存 AccessToken，代码为：

```
Static CreateTime As Date '记录最近一次生成 access_token 的时间
Static AccessTocken As String '记录最近一次生成的 access_token
Dim AppId As String= "wx8acdb7df5beb68fd" '开发者 ID
Dim AppSecret As String = "a5a3b06a790d4200a151e6b13c3dc263" '开发者密匙
Dim tp As TimeSpan = Date.Now - CreateTime
Dim url As String = "https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid={0}&secret={1}"
If tp.TotalSeconds > 3600 Then
    Dim hc As New HttpClient(CExp(url,AppId,AppSecret))
    Dim ret As String = hc.GetData()
    If ret = "" Then '如果失败,再尝试一次
        hc.GetData()
    End If
    CreateTime = Date.Now()
    Dim jo As JObject = JObject.Parse(ret)
    If jo("errcode") Is Nothing Then
        AccessTocken = jo("access_token")
    Else
        MessageBox.show("获取 access_token 失败,原因:" & vbcrlf & jo("errmsg").ToString)
    End If
End If
Return AccessTocken
```

这个函数之前就已经存在，实际上设计中控服务器的代码，只是接下来第 2 步和第 3 步中的几行代码而已.
2、在中控服务器（或中控项目）开启 HttpServer，在 HttpRequest 事件加上代码：

```
Select Case e.path
    Case "GetToken"
        Dim UserName As String = e.PostValues("name")
        Dim Password As String = e.PostValues("password")
        '验证用户身份
        e.WriteString(Functions.Execute("GetAccessToken"))
End Select
```

3、其它业务电脑（或其它项目）可通过下面的代码获取 AccessToken：

```
Dim hc As New HttpClient("http://127.0.0.1/GetToken/") '请改为中控服务器的实际 IP
hc.FormData.Add("UserName","老三") '添加用户名和密码
hc.FormData.Add("Password","888")
Dim Token =  hc.GetData '获取 AccessToken
MessageBox.Show(Token)
```

上面的代码采用了用户名和密码来避免 AccessToken 被非法获取，实际开发的时候，也可以采用字符串加密解密的方式保护 AccessToken，或者两种方式都用。

关于字符串的加密和解密，可以参考：EncryptText

10.3.13 被动回复还是主动发送？

公众号收到用户消息后，如果要被动回复消息，必须在 5 秒内完成。

如果要要主动发送消息，必须在 48 小时内完成。

48 小时后，只能用模板消息(格式有限制)或群发消息(条数有限制)的形式完成。
由于 5 秒的限制，所以必须保持服务端项目处于轻负荷状态，而有些消息的回复，是需要经过一定的查询计算或处理才能形成回复结果，会耗费一定的时间，此时采用被动回复时不恰当的，只能采用主动发送。

我建议对于达到一定规模的系统，负责接收客户消息的服务端项目不要涉及具体业务的处理，收到消息后，直接将消息存入数据库，然后回复一个类似“消息收到，正在处理...”的提示给客户 ，或直接回复一个空消息。同时开启另一个项目，专门负责从数据库中提取新收到的消息，进行查询计算或处理，形成回复结果后，用主动发送的方式，将结果发送给客户。

这样的设计并不复杂，服务端项目和业务项目共用一个数据库，业务项目开启一个计划，定期从数据库追载新的消息进行处理即可，同时建议将处理完成的行(消息)从数据表中移除，避免表中数据越来越多。如果希望有更快捷的响应，可以利用 OpenQQ 和 UDPClient 连接服务端项目和业务项目，这样响应更及时，也能降低数据库服务器的负荷。
多个项目协同工作，必须考虑 AccessToken 的缓存冲突问题，请参考：中控服务器的设计

10.4 企业号接口

10.4.1 接入 HTTP 服务

企业号进行HTTP服务接入验证的时候，其echostr参数是加密的，需要先解密后回写才能完成接入验证。

为了方便非专业用户，我们在WXBizCrypt类中提供了一个CheckQYSignature函数，可以 用很简单的代码完成接入验证。

企业号可以增加多个应用，每一个应用都可以独立设置回调地址(HTTP服务地址)，设置时需要指定Token和EncodingAESKey：



定义WXBizCrypt的语法为：

Dim wbiz As New WXBizCrypt(CorpID, Token, EncodingAESKey)

CorpID	企业号的 CorpID。
Token	上图中指定的 Token。
EncodingAESKey	上图中指定的 EncodingAESKey。

下面是Foxtable端的HttpRequest事件，用于完成企业号的接入验证：

```
Select Case e.path
   Case "wefox"
                        Dim                  wbiz                  As                  New
WXBizCrypt("wxa31aba4cd83af57e","foxtable","ilsmyivvRPNj0qxSiSzWCnqm7cy1w1RcS6w2LBhsh7J")
      If e.Request.HttpMethod = "GET"
        wbiz.CheckQYSignature(e)
      End If
End Select
```

如果接入验证失败，你可以采用下面的代码获得错误描述：

```
Select Case e.path
   Case "wefox"
                        Dim                  wbiz                  As                  New
WXBizCrypt("wxa31aba4cd83af57e","foxtable","ilsmyivvRPNj0qxSiSzWCnqm7cy1w1RcS6w2LBhsh7J")
      If e.Request.HttpMethod = "GET"
        Dim ret As Integer = wbiz.CheckQYSignature(e)
        If ret <> 0 Then '如果接入验证失败
          Dim err As String = wbiz.GetErorDesc(ret) '获取错误描述
        End If
      End If
End Select
```

10.4.2 获取 AccessToken

企业号通过 CorpID 和 Secret 获取 AccessToken。

CorpID 是企业号的标识，每个企业号拥有一个唯一的 CorpID；Secret 是管理组凭证密钥，不同的管理组有不同的 Secret。

我们可以定义一个自定义函数，名为 GetQYAccessToken，用于获取企业号的 AccessToken，代码参考：

```
Static CreateTime As Date '记录最近一次生成access_token的时间
Static AccessTocken As String '记录最近一次生成的access_token
Dim CorpID As String=   "ww7deed3af842576ed" '企业号的CorpID
Dim Secret As String = "sO7RjwoisX82kB_gcRY4isEU8ZC6sJVcGs6CDlz8Nkc" '管理组的Secret
```

```
Dim tp As TimeSpan = Date.Now - CreateTime
If tp.TotalSeconds > 3600 Then
    Dim url As String = "https://qyapi.weixin.qq.com/cgi-bin/gettoken?corpid={0}&corpsecret={1}"
    Dim hc As New HttpClient(CExp(url, CorpID, Secret))
    Dim ret As String = hc.GetData()
    If ret = "" Then '如果失败,再尝试一次
        hc.GetData()
    End If
    CreateTime = Date.Now()
    Dim jo As JObject = JObject.Parse(ret)
    If jo("access_token") IsNot Nothing Then
        AccessTocken = jo("access_token")
    Else
        MessageBox.show("获取access_token失败,原因:" & vbcrlf & jo.ToString)
    End If
End If
Return AccessTocken
```

企业号的AccessToken在有效期内重复获取，返回的值是一样的，所以企业号并不需要设置中控服务 器。

10.4.3 消息接口

10.4.3.1 接收用户消息

微信服务器收到用户发来的消息后，会以POST方式，将消息转发给Foxtable端的HTTP服务。

消息的格式是纯文本的XML，保存在HttpRequest事件的e参数PlainText中。

企业号的消息都是经过加密的，收到的消息要先解密，发送的消息也要先加密，Foxtable内置了一个WXBizCrypt类，专门用于微信消息的加解密。

1、首先新建一个数据表，用于保存收到的用户消息，表名为Message，表结构如下：

查看表结构

列名	类型	长度	标题
AgentID	整数		
UserID	字符型	32	
CreateTime	日期时间型		
MsgType	字符型	16	
MsgId	长整数		
Format	字符型	16	
MediaId	字符型	64	
PicUrl	字符型	168	
ThumbMediaId	字符型	64	
Content	备注型		

2、将FoxTable端的HttpRequest时事件代码设置为：

```
Select Case e.path
    Case "wefox"
                                Dim                wbiz                As                New
WXBizCrypt("wxa31aba4cd83af57e","foxtable","ilsmyivvRPNj0qxSiSzWCnqm7cy1w1RcS6w2LBhsh7J")
        If e.Request.HttpMethod = "GET"
            Dim ret As Integer = wbiz.CheckQYSignature(e)
            If ret <> 0 Then '如果接入验证失败
                Dim err As String = wbiz.GetErorDesc(ret) '获取错误描述
            End If
        ElseIf e.Request.HttpMethod = "Post"            '
            Dim st As New Date(1970,1,1,8,0,0)
            Dim msg   As String = wbiz.DecryptMsg(e) '解密消息
            If   IsNumeric(msg) Then '如果解密失败
                Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
                Return
            End If
            Dim xo As XObject = XObject.Parse(msg)
            Select Case xo("MsgType")
                Case "text","image","voice","video","shortvideo"
                    Dim dr As DataRow = DataTables("Message").AddNew()
                    dr("AgentID") = xo("AgentID")
                    dr("UserID") = xo("FromUserName")
                    dr("CreateTime") = st.AddSeconds(xo("CreateTime"))
                    dr("MsgType") = xo("MsgType")
                    dr("MsgId") = xo("MsgId")
                    dr("MediaId") = xo("MediaId")
                    dr("ThumbMediaId") = xo("ThumbMediaId")
                    dr("PicUrl") = xo("PicUrl")
                    dr("Content") = xo("Content")
```

```
            dr("Format") = xo("Format")
        End Select
    End If
End Select
```

过滤重复消息

微信服务器将收到的用户消息转发给Foxtable端的HTTP服务，如果Foxtbale不能在5秒内做出回应，微信服务器会再次转发此消息，总共会重复三次，这意味着如果用户量比较大，服务端处理不及时，可能会收到三条同样的消息，显然我们需要编码过滤掉重复的消息。

为了过滤掉重复消息，可以将HttpRequest事件代码改为：

```
Select Case e.path
  Case "wefox"
                    Dim          wbiz          As          New
WXBizCrypt("wxa31aba4cd83af57e","foxtable","ilsmyivvRPNj0qxSiSzWCnqm7cy1w1RcS6w2LBhsh7J")
    If e.Request.HttpMethod = "GET"
      Dim ret As Integer = wbiz.CheckQYSignature(e)
      If ret <> 0 Then '如果接入验证失败
        Dim err As String = wbiz.GetErorDesc(ret) '获取错误描述
      End If
    ElseIf e.Request.HttpMethod = "Post"          '
      Dim st As New Date(1970,1,1)
      Dim msg  As String = wbiz.DecryptMsg(e) '解密消息
      If  IsNumeric(msg) Then '如果解密失败
        Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
        Return
      End If
      Dim xo As XObject = XObject.Parse(msg)
      Select Case xo("MsgType")
        Case "text","image","voice","video","shortvideo"
          If DataTables("Message").Find("MsgID = '" & xo("MsgId").ToString() & "'") IsNot Nothing
Then '如果有相同MsgID的信息直接返回
            Return
          End If
          Dim dr As DataRow = DataTables("Message").AddNew()
          dr("AgentID") = xo("AgentID")
          dr("UserID") = xo("FromUserName")
          dr("CreateTime") = st.AddSeconds(xo("CreateTime"))
          dr("MsgType") = xo("MsgType")
          dr("MsgId") = xo("MsgId")
          dr("MediaId") = xo("MediaId")
          dr("ThumbMediaId") = xo("ThumbMediaId")
```

```
            dr("PicUrl") = xo("PicUrl")
            dr("Content") = xo("Content")
            dr("Format") = xo("Format")
        End Select
    End If
End Select
```

10.4.3.2 被动回复消息

我们可以在 HttpRequest 事件中被动回复消息，被动回复的消息也是 XML 格式的。

被动回复的消息必须先进行加密，Foxtable 内置了一个 WXBizCrypt 类，专门用于微信消息的加解密。

除了要加解密，企业号被动回复消息和公众号几乎一模一样。

被动回复文本消息

如果要被动回复文本消息，请参考以下的 HttpRequest 事件代码：

```
Select Case e.path
    Case "wefox"
                    Dim wbiz As New WXBizCrypt("wxa31aba4cd83af57e","foxtable","ilsmyivvRPNj0qxSiSzWCnqm7cy1w1RcS6w2LBhsh7J")
        If e.Request.HttpMethod = "GET"
            Dim ret As Integer = wbiz.CheckQYSignature(e)
            If ret <> 0 Then '如果接入验证失败
                Dim err As String = wbiz.GetErorDesc(ret) '获取错误描述
            End If
        ElseIf e.Request.HttpMethod = "Post"            '
            Dim st As New Date(1970,1,1,8,0,0)
            Dim msg  As String = wbiz.DecryptMsg(e) '解密消息
            If  IsNumeric(msg) Then '如果解密失败
                Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
                Return
            End If
            Dim xo As XObject = XObject.Parse(msg)
            Select Case xo("MsgType")
                Case "text","image","voice","video","shortvideo"
                    '如果要保存收到的消息,请参考上一节在此处编写代码.
                    '下面是被动回复文本消息的代码:
                    Dim so As New XObject()
                    so("ToUserName") = xo("FromUserName")
```

```
            so("FromUserName") = xo("ToUserName")
            so("CreateTime") = (Date.Now - st).TotalSeconds()
            so("MsgType")= "text"
            so("Content") = "您好,我是 Foxtable,今天是" & Format(Date.Today,"yyyy 年 MM 月 dd 日
")
            msg = wbiz .EncryptMsg(e,so.ToXML ) '加密要发送的消息
            If  IsNumeric(msg) Then '如果加密失败
              Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
            Else
              e.WriteString(msg)
            End If
        End Select
    End If
End Select
```

被动回复图文消息

如果要被动回复图文消息，请参考以下 HttpRequest 事件代码：

```
Select Case e.path
  Case "wefox"
                    Dim          wbiz          As          New
WXBizCrypt("wxa31aba4cd83af57e","foxtable","ilsmyivvRPNj0qxSiSzWCnqm7cy1w1RcS6w2LBhsh
7J")
    If e.Request.HttpMethod = "GET"
      Dim ret As Integer = wbiz.CheckQYSignature(e)
      If ret <> 0 Then '如果接入验证失败
        Dim err As String = wbiz.GetErorDesc(ret) '获取错误描述
      End If
    ElseIf e.Request.HttpMethod = "Post"          '
      Dim st As New Date(1970,1,1,8,0,0)
      Dim msg  As String = wbiz.DecryptMsg(e) '解密消息
      If  IsNumeric(msg) Then '如果解密失败
        Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
        Return
      End If
      Dim xo As XObject = XObject.Parse(msg)
      Select Case xo("MsgType")
        Case "text","image","voice","video","shortvideo"
          '如果要保存收到的消息,请参考上一节在此处编写代码.
          '下面是被动回复文本消息的代码:
          Dim so As New XObject()
          Dim ja As New JArray
          so("ToUserName") = xo("FromUserName")
          so("FromUserName") = xo("ToUserName")
          so("CreateTime") = (Date.Now - st).TotalSeconds()
```

```
            so("MsgType") = "news"
            so("ArticleCount") = 1 '图文数量
            For cnt As Integer = 1 To 1 '可以回最多 8 条图文,这里只回 1 条
                Dim ao As New JObject
                ao("Title") = "特朗普提名保守派大法官" '图文标题
                ao("Description") = "特朗普在白宫提名尼尔 •戈萨奇为美国最高法院大法官" '图文简介
                ao("PicUrl") = "https://ss1.baidu.com/6ONXsjip0QIZ8tyhnq/it/u=1632653074,1222564348&fm=80&w=179&h=119&img.JPEG" '图片
                ao("Url") = "http://news.ifeng.com/a/20170201/50640529_0.shtml" '单击阅读原文后跳转的连接
                ja.Add(ao)
            Next
            so("Articles") = New JObject()
            so("Articles")("item") = ja
            msg = wbiz .EncryptMsg(e,so.ToXML ) '加密要发送的消息
            If IsNumeric(msg) Then '如果加密失败
                Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
            Else
                e.WriteString(msg)
            End If
        End Select
    End If
End Select
```

现在用户向我们的测试号发送消息，会收到类似以下内容的图文回复：



准备素材

不管是被动回复还是主动发送，除了文本和图文消息，其它要发送给用户的消息，包括图片、视频和语音，都必须事先将相关文件上传到微信服务器。

因为微信服务器并不会直接给用户发送文件，而是发送一个素材 ID(MediaId)，每一个素材 ID 都对应者一个素材文件。

关于素材的管理，请参考后面的《素材接口》这一章。

被动回复图片消息

必须事先将要回复的图片上传到微信服务器，因为回复的只能是图片素材的 ID。

被动回复图片消息参考代码：

```
Select Case e.path
    Case "wefox"
```

```
                Dim wbiz As New WXBizCrypt("wxa31aba4cd83af57e","foxtable","ilsmyivvRPNj0qxSiSzWCnqm7cy1w1RcS6w2LBhsh7J")
    If e.Request.HttpMethod = "GET"
      Dim ret As Integer = wbiz.CheckQYSignature(e)
      If ret <> 0 Then '如果接入验证失败
        Dim err As String = wbiz.GetErorDesc(ret) '获取错误描述
      End If
    ElseIf e.Request.HttpMethod = "Post"  '
      Dim st As New Date(1970,1,1,8,0,0)
      Dim msg  As String = wbiz.DecryptMsg(e) '解密消息
      If  IsNumeric(msg) Then '如果解密失败
        Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
        Return
      End If
      Dim xo As XObject = XObject.Parse(msg)
      Select Case xo("MsgType")
        Case "text","image","voice","video","shortvideo"
          '如果要保存收到的消息,请参考上一节在此处编写代码.
          '下面是被动回复图片消息的代码:
          Dim so As New XObject()
          so("ToUserName") = xo("FromUserName")
          so("FromUserName") = xo("ToUserName")
          so("CreateTime") = (Date.Now - st).TotalSeconds()
          so("MsgType") = "image"
          so("Image") = New JObject
                so("Image")("MediaId") = "2UgIup9iQcz60YIjrpxPFSUTYoPdTAqCz52PXUruWM6isWNdPOQwRgPY3lzJmsKLxPeSWqLMUMREyg9K-4MO8bg" '图片素材 ID
          msg = wbiz .EncryptMsg(e,so.ToXML ) '加密要发送的消息
          If  IsNumeric(msg) Then '如果加密失败
            Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
          Else
            e.WriteString(msg)
          End If
      End Select
    End If
End Select
```

被动回复语音消息

必须事先将要回复的语音文件上传到微信服务器，因为回复的只能是语音素材的 ID。

被动回复语音消息参考代码:

```
Select Case e.path
    Case "wefox"
                                Dim                          wbiz                          As                          New
WXBizCrypt("wxa31aba4cd83af57e","foxtable","ilsmyivvRPNj0qxSiSzWCnqm7cy1w1RcS6w2LBhsh7J")
        If e.Request.HttpMethod = "GET"
            Dim ret As Integer = wbiz.CheckQYSignature(e)
            If ret <> 0 Then '如果接入验证失败
                Dim err As String = wbiz.GetErorDesc(ret) '获取错误描述
            End If
        ElseIf e.Request.HttpMethod = "Post"            '
            Dim st As New Date(1970,1,1,8,0,0)
            Dim msg   As String = wbiz.DecryptMsg(e) '解密消息
            If   IsNumeric(msg) Then '如果解密失败
                Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
                Return
            End If
            Dim xo As XObject = XObject.Parse(msg)
            Select Case xo("MsgType")
                Case "text","image","voice","video","shortvideo"
                    '下面是被动回复语音消息的代码:
                    Dim so As New XObject()
                    so("ToUserName") = xo("FromUserName")
                    so("FromUserName") = xo("ToUserName")
                    so("CreateTime") = (Date.Now - st).TotalSeconds()
                    so("MsgType") = "voice"
                    so("Voice") = New JObject
                                                    so("Voice")("MediaId")                                    =
"20U2cc1RA4G8ZOU573NDgQlF0BsBHtiyl5FfxkxUc5jbZxFZqqsD3Ab2AGywhU-s-SP6Jgjs76RciJfWZQ68OVg" '语音素材 ID
                    msg = wbiz .EncryptMsg(e,so.ToXML ) '加密要发送的消息
                    If   IsNumeric(msg) Then '如果加密失败
                        Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
                    Else
                        e.WriteString(msg)
                    End If
            End Select
        End If
End Select
```

被动回复视频消息

必须事先将要回复的视频文件上传到微信服务器，因为回复的只能是是因素材的 ID。

被动回复视频消息参考代码：

```
Select Case e.path
    Case "wefox"
                                Dim                          wbiz                          As                          New
WXBizCrypt("wxa31aba4cd83af57e","foxtable","ilsmyivvRPNj0qxSiSzWCnqm7cy1w1RcS6w2LBhsh7J")
        If e.Request.HttpMethod = "GET"
            Dim ret As Integer = wbiz.CheckQYSignature(e)
            If ret <> 0 Then '如果接入验证失败
                Dim err As String = wbiz.GetErorDesc(ret) '获取错误描述
            End If
        ElseIf e.Request.HttpMethod = "Post"            '
            Dim st As New Date(1970,1,1,8,0,0)
            Dim msg   As String = wbiz.DecryptMsg(e) '解密消息
            If   IsNumeric(msg) Then '如果解密失败
                Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
                Return
            End If
            Dim xo As XObject = XObject.Parse(msg)
            Select Case xo("MsgType")
                Case "text","image","voice","video","shortvideo"
                    '下面是被动回复语音消息的代码:
                    Dim so As New XObject()
                    so("ToUserName") = xo("FromUserName")
                    so("FromUserName") = xo("ToUserName")
                    so("CreateTime") = (Date.Now - st).TotalSeconds()
                    so("MsgType") = "video"
                    so("Video") = New JObject
                                                    so("Video")("MediaId")                                  =
"24rRqRLyJMJXT-FAtspbOc1pkPwZleQ9t6PRBMK-hQY-0pPjSRmWQUUGjTda_pVn0SwbFGzjX4DtONQNqCQe7rg" '语音素材 ID
                    so("Video")("Title") = "视频标题"
                    so("Video")("Description") = "视频内容简介"
                    msg = wbiz .EncryptMsg(e,so.ToXML ) '加密要发送的消息
                    If   IsNumeric(msg) Then '如果加密失败
                        Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
                    Else
                        e.WriteString(msg)
                    End If
            End Select
        End If
End Select
```

10.4.3.3 主动发送消息

企业号主动发送消息没有 48 小时的限制，而发送次数也几乎不受限制。

主动发送消息的时候，要指定应用 ID(AgentID)。

主动发送文本消息

可以直接向用户列表发送，例如下面的代码向账号为"laoliu"和"zhangsan"的用户发送消息：

```
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/message/send?access_token={0}"
Dim hc As new HttpClient(Cexp(ur,Functions.Execute("GetQYAccessToken")))
Dim jo As New JObject
jo("touser") = "laoliu|zhangsan"
jo("msgtype") = "text"
jo("agentid") = 9
jo("text") = New JObject
jo("text")("content") = "您好,这是测试信息"
hc.Content = jo.ToString
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.show("消息发送成功!")
Else
    MessageBox.show(jo.ToString)
End If
```

注意这里的"laoliu"和"zhangsan"并非 OpenId，而是企业号通讯录中的用户账号(UserID)。
也可以按部门或标签发送，下面的代码会给两个部门(部门 ID 分别为 3 和 9)的所有成员发送消息：

```
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/message/send?access_token={0}"
Dim hc As new HttpClient(Cexp(ur,Functions.Execute("GetQYAccessToken")))
Dim jo As New JObject
jo("toparty") = "9|3"
jo("msgtype") = "text"
jo("agentid") = 9
jo("text") = New JObject
jo("text")("content") = "您好,这是测试信息"
hc.Content = jo.ToString
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.show("消息发送成功!")
Else
    MessageBox.show(jo.ToString)
End If
```

接下来的的例子，都只提供向用户列表发送的示例代码。

主动发送动态图文消息

下面是主动发送动态图文消息的示例，你可以在命令窗口运行测试：

```
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/message/send?access_token={0}"
Dim hc As new HttpClient(Cexp(ur,Functions.Execute("GetQYAccessToken")))
Dim jo As New JObject
Dim ja As New JArray
jo("touser") = "laoliu|zhangsan"
jo("msgtype") = "news"
jo("agentid") = 9
For i As Integer = 1 To 1 '可以一次发送最多 8 个图文消息,这里只发送了 1 个
    Dim ao As New JObject()
    ao("title") = "特朗普提名保守派大法官" '标题
    ao("description") = "特朗普在白宫提名尼尔·戈萨奇为美国最高法院大法官" '描述
    ao("url") = "http://news.ifeng.com/a/20170201/50640529_0.shtml" '连接
                                                    ao("picurl")                                                    =
"https://ss1.baidu.com/6ONXsjip0QIZ8tyhnq/it/u=1632653074,1222564348&fm=80&w=179&h=119&img.JPEG" '图片
    ja.Add(ao)
Next
jo("news") =   New JObject()
jo("news")("articles") = ja
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.show("消息发送成功!")
Else
    MessageBox.show(jo.ToString)
End If
```

准备素材

不管是被动回复还是主动发送，除了文本和动态图文消息，其它要发送给用户的消息，包括图片、视频、语音以及永久图文消息，都必须事先将相关文件上传到微信服务器。

因为微信服务器并不会直接给用户发送文件，而是发送一个素材 ID(MediaId)，每一个素材 ID 都对应者一个素材文件。

关于素材的管理，请参考后面的《素材接口》这一章。

主动发送图片消息

必须事先将要发送的图片上传到微信服务器，因为我们发送的只能是图片素材的 ID，无法直接发送文件。

主动发送图片消息参考代码：

```
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/message/send?access_token={0}"
Dim hc As new HttpClient(Cexp(ur,Functions.Execute("GetQYAccessToken")))
Dim jo As New JObject
jo("touser") = "laoliu|zhangsan"
jo("msgtype") = "image"
jo("agentid") = 9
jo("image") = New JObject
jo("image")("media_id") = "2UgIup9iQcz60YIjrpxPFSUTYoPdTAqCz52PXUruWM6isWNdPOQwRgPY3lzJmsKLxPeSWqLMUMREyg9K-4MO8bg"
hc.Content = jo.ToString
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.show("消息发送成功!")
Else
    MessageBox.show(jo.ToString)
End If
```

主动发送语音消息

必须事先将要发送的语音上传到微信服务器，因为我们发送的只能是语音素材的 ID，无法直接发送文件。

主动发送语音消息参考代码：

```
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/message/send?access_token={0}"
Dim hc As new HttpClient(Cexp(ur,Functions.Execute("GetQYAccessToken")))
Dim jo As New JObject
jo("touser") = "laoliu|zhangsan"
jo("msgtype") = "voice"
jo("agentid") = 9
jo("voice") = New JObject
jo("voice")("media_id") = "20U2cc1RA4G8ZOU573NDgQlF0BsBHtiyl5FfxkxUc5jbZxFZqqsD3Ab2AGywhU-s-SP6Jgjs76RciJfWZQ68OVg"
hc.Content = jo.ToString
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.show("消息发送成功!")
Else
    MessageBox.show(jo.ToString)
End If
```

主动发送视频消息

必须事先将要发送的视频上传到微信服务器，因为我们发送的只能是视频素材的 ID，无法直接发送文件。

主动发送视频消息参考代码：

```
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/message/send?access_token={0}"
Dim hc As new HttpClient(Cexp(ur,Functions.Execute("GetQYAccessToken")))
Dim jo As New JObject
jo("touser") = "laoliu|zhangsan"
jo("msgtype") = "video"
jo("agentid") = 9
jo("video") = New JObject
jo("video")("media_id") = "24rRqRLyJMJXT-FAtspbOc1pkPwZleQ9t6PRBMK-hQY-0pPjSRmWQUUGjTda_pVn0SwbFGzjX4DtONQNqCQe7rg"
hc.Content = jo.ToString
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.show("消息发送成功!")
Else
    MessageBox.show(jo.ToString)
End If
```

主动发送文件消息

相对于公众号，企业号可以发送普通的文件消息，这一点优势非常明显，例如我们可以直接向用户发送 Excel 和 Word 报表。

必须事先将要发送的文件上传到微信服务器，因为我们发送的只能是文件素材的 ID，无法直接发送文件。

主动发送文件消息参考代码：

```
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/message/send?access_token={0}"
Dim hc As new HttpClient(Cexp(ur,Functions.Execute("GetQYAccessToken")))
Dim jo As New JObject
jo("touser") = "laoliu|zhangsan"
jo("msgtype") = "file"
jo("agentid") = 9
jo("file") = New JObject
jo("file")("media_id") = "2LidksRrG1ybDoGS2fvGIaT0yZtTqYqn7zXT_5kJ4FptZhLB2EX25FQ15UAnn24583CHK_1F8QttulZ6YCZJd5w"
hc.Content = jo.ToString
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.show("消息发送成功!")
Else
```

```
    MessageBox.show(jo.ToString)
End If
```

主动发送永久图文消息

和动态合成的图文消息不同，永久图文消息需要事先上传到服务器，之后将素材 ID 发送给用户。

主动发送永久图文消息的参考代码：

```
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/message/send?access_token={0}"
Dim hc As new HttpClient(Cexp(ur,Functions.Execute("GetQYAccessToken")))
Dim jo As New JObject
jo("touser") = "laoliu|zhangsan"
jo("msgtype") = "mpnews"
jo("agentid") = 9
jo("mpnews") = New JObject
jo("mpnews")("media_id") = "2Tb6YJ_zqB3FXrUdJRcco6Ck6CHuL4fbLO2u7tB3_yoiEeiJS3jvlMVzpRoE3KcEi"
hc.Content = jo.ToString
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.show("消息发送成功!")
Else
    MessageBox.show(jo.ToString)
End If
```

10.4.3.4 接收事件推送

企业号接收事件推送和公众号基本相似，只是有个解密和加密的过程。

假定你需要自动生成一个日志文件，记录用户关注、取消关注、位置以及进入应用的信息：

```
laoliu 于 2017-02-14 07:06:30 关注 0
laoliu 于 2017-02-14 07:06:31 关注 9
laoliu 于 2017-02-14 07:06:34 访问 9
laoliu 于 2017-02-14 07:06:54 访问 9
zhangsan 于 2017-02-14 07:07:09 访问 9
zhangsan 于 2017-02-14 07:07:14 访问 9
zhangsan 在 2017-02-14 07:07:16 的位置为:21.257669|110.377129
```

完成这样的任务很简单，可以参考下面的 HttpRequest 事件代码：

```
If e.Path = "wefox"
                Dim wbiz As New
WXBizCrypt("wxa31aba4cd83af57e","foxtable","ilsmyivvRPNj0qxSiSzWCnqm7cy1w1RcS6w2LBhsh
```

```
7J")
    If e.Request.HttpMethod = "GET"
        Dim ret As Integer = wbiz.CheckQYSignature(e)
        If ret <> 0 Then '如果接入验证失败
            Dim err As String = wbiz.GetErorDesc(ret) '获取错误描述
        End If
    ElseIf e.Request.HttpMethod = "Post"            '
        Dim st As New Date(1970,1,1,8,0,0)
        Dim msg   As String = wbiz.DecryptMsg(e) '解密
        If   IsNumeric(msg) Then '如果解密失败
            Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
            Return
        End If
        Dim xo As XObject = XObject.Parse(msg)
        Select Case xo("MsgType")
            Case "text","image","voice","video","shortvideo" '处理消息
                '保存消息的代码
            Case "event" '处理事件
                Dim UserID As String   = xo("FromUserName") '用户的 UserID
                Dim CreateTime As Date = st.AddSeconds(xo("CreateTime")) '事件触发时间
                Dim AgentID As String = xo("AgentID") '应用 ID,如果为 0 表示针对整个企业号.
                Dim logFile As String   = "c:\data\wxlog" & Format(Date.Today,"yyyyMMdd") & ".txt" '你可以选择每日或每月一个日志文件
                Select Case xo("Event")
                    Case "subscribe" '关注事件
                        Filesys.WriteAllText(logFile , UserID & "于" & CreateTime & "关注" & AgentID & vbcrlf, True)
                    Case "unsubscribe" '取消关注事件
                        Filesys.WriteAllText(logFile , UserID & "于" & CreateTime & "取消关注" & AgentID & vbcrlf ,True)
                    Case "enter_agent" '进入应用事件
                        Filesys.WriteAllText(logFile , UserID & "于" & CreateTime & "访问" & AgentID & vbcrlf , True)
                    Case "LOCATION" '上报地理位置事件
                        Dim y As String = xo("Longitude") '经度
                        Dim x As String = xo("Latitude") '纬度
                        Filesys.WriteAllText(logFile , UserID & "在" & CreateTime & "的位置为:" & x & "|" & y & vbcrlf , True)
                End Select
        End Select
    End If
End If
```

在事件中被动回复消息

在事件中也可以被动回复消息，记得消息一定要先加密才能回复，录入下面的代码在用户关注后，

自动回复一条欢迎语：

```
If e.Path = "wefox"
                    Dim wbiz As New WXBizCrypt("wxa31aba4cd83af57e","foxtable","ilsmyivvRPNj0qxSiSzWCnqm7cy1w1RcS6w2LBhsh7J")
    If e.Request.HttpMethod = "GET"
        Dim ret As Integer = wbiz.CheckQYSignature(e)
        If ret <> 0 Then '如果接入验证失败
            Dim err As String = wbiz.GetErorDesc(ret) '获取错误描述
        End If
    ElseIf e.Request.HttpMethod = "Post"            '
        Dim st As New Date(1970,1,1,8,0,0)
        Dim msg   As String = wbiz.DecryptMsg(e) '解密
        If   IsNumeric(msg) Then '如果解密失败
            Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
            Return
        End If
        Dim xo As XObject = XObject.Parse(msg)
        Select Case xo("MsgType")
            Case "text","image","voice","video","shortvideo" '处理消息
                '保存消息的代码
            Case "event" '处理事件
                Dim UserID As String   = xo("FromUserName") '用户的 UserID
                Dim CreateTime As Date = st.AddSeconds(xo("CreateTime")) '事件触发时间
                Dim AgentID As String = xo("AgentID") '应用 ID,如果为 0 表示针对整个企业号.
                Dim logFile As String   = "c:\data\wxlog" & Format(Date.Today,"yyyyMMdd") & ".txt" '你可以选择每日或每月一个日志文件
                Select Case xo("Event")
                    Case "subscribe" '关注事件
                        Filesys.WriteAllText(logFile , UserID & "于" & CreateTime & "关注" & AgentID & vbcrlf, True)
                        '被动回复消息
                        Dim so As New XObject()
                        so("ToUserName") = xo("FromUserName")
                        so("FromUserName") = xo("ToUserName")
                        so("CreateTime") = (Date.Now - st).TotalSeconds()
                        so("MsgType")= "text"
                        so("Content") = "今天是个好日子,欢迎" & UserID & "的加入!"
                        msg = wbiz .EncryptMsg(e,so.ToXML ) '加密要发送的消息
                        If   IsNumeric(msg) Then '如果加密失败
                            Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
                        Else
                            e.WriteString(msg)
                        End If
```

```
            Case "unsubscribe" '取消关注事件
                Filesys.WriteAllText(logFile , UserID & "于" & CreateTime & "取消关注" & AgentID & vbcrlf ,True)
            Case "enter_agent" '进入应用事件
                Filesys.WriteAllText(logFile , UserID & "于" & CreateTime & "访问" & AgentID & vbcrlf , True)
            Case "LOCATION" '上报地理位置事件
                Dim y As String = xo("Longitude") '经度
                Dim x As String = xo("Latitude") '纬度
                Filesys.WriteAllText(logFile , UserID & "在" & CreateTime & "的位置为:" & x & "|" & y & vbcrlf , True)
        End Select
    End Select
  End If
End If
```

10.4.4 菜单接口

10.4.4.1 菜单接口

企业号不同的应用，可以设置不同的菜单，所以在调用企业号的菜单接口时，除了提交 AccessTokn，还需要提交 AgentId(应用 ID)。
实际开发额时候，可以设计一个窗口，用于 菜管理菜单：



"设置菜单"按钮的代码为：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/menu/create?access_token={0}&agentid={1}"
Dim AccessToken = Functions.Execute("GetQYAccessToken") 'AccessTiken
Dim AgentId As String = e.Form.Controls("NumericComboBox1").Value '应用 ID
Dim MenuJson As String = e.Form.Controls("TextBox1").Value '在文本框中输入的 Json 格式的菜单
Dim hc As New HttpClient(CExp(ul, AccessToken, AgentId))
hc.Content = MenuJson
Dim jo As JObject = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("菜单设置成功!")
Else
    MessageBox.Show(jo.ToString)
End If
```

"删除菜单"按钮的代码为：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/menu/delete?access_token={0}&agentid={1}"
Dim AccessToken = Functions.Execute("GetQYAccessToken") 'AccessTiken
Dim AgentId As String = e.Form.Controls("NumericComboBox1").Value '应用 ID
Dim hc As New HttpClient(CExp(ul, AccessToken, AgentId))
Dim jo As JObject = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
```

```
    MessageBox.Show("菜单已经删除!")
Else
    MessageBox.Show(jo.ToString)
End If
```

"提取菜单"按钮用于提取现有菜单设置，其代码为：

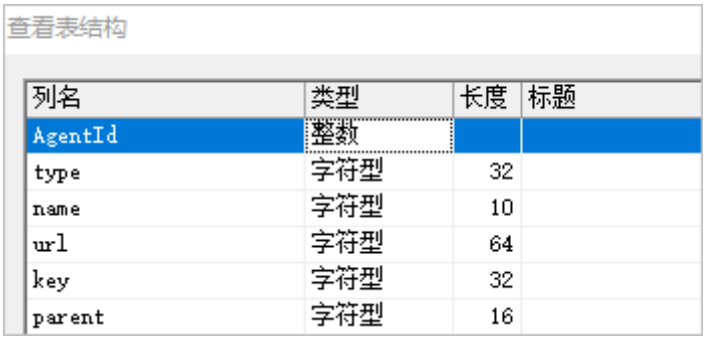
```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/menu/get?access_token={0}&agentid={1}"
Dim AccessToken = Functions.Execute("GetQYAccessToken") 'AccessTiken
Dim AgentId As String = e.Form.Controls("NumericComboBox1").Value '应用 ID
Dim hc As New HttpClient(CExp(ul, AccessToken, AgentId))
Dim jo As JObject = JObject.Parse(hc.GetData)
If jo("menu") IsNot Nothing Then
    e.Form.Controls("TextBox1").Value = jo("menu").ToString()
Else
    MessageBox.Show(jo.ToString)
End If
```

实际测试的时候，发现必须先给应用开启回调(接入 HTTP 服务)才能为其设置菜单。

10.4.4.2 用表管理菜单

直接为企业号的众多应用编辑 JSON 格式的菜单，是比较繁琐的，我们可以自己开发一个用表管理菜单的小功能。

1、首先设计一个表，结构如下：


查看表结构

列名	类型	长度	标题
AgentId	整数		
type	字符型	32	
name	字符型	10	
url	字符型	64	
key	字符型	32	
parent	字符型	16	

2、AgentID 列是整数型，用于输入应用 ID；parent 列用于输入上级菜单名称，如果不输入，表示此按钮为一级菜单按钮；type 为 view 的按钮，必须输入 url 值；其它类型的按钮，必须输入 key 值，否则会出错。

例如：

	AgentId	type	name	url	key	parent
1	9	view	主页	http://www.foxtable.com/		
2	9		菜单一			
3	9		菜单二			
4	9	view	搜索	http://www.soso.com/		菜单一
5	9	pic_sysphoto	系统拍照发图		rselfmenu_0_0	菜单一
6	9	click	赞一下我们		rselfmenu_0_1	菜单一
7	9	scancode_push	扫描二维码		rselfmenu_1_0	菜单二
8	9	location_select	发送位置		rselfmenu_1_1	菜单二
9	9	view	搜索	http://www.baidu.com/		菜单二
10	11	click	今日歌曲		V1001_TODAY_MUSIC	
11	11		菜单			
12	11	view	搜索	http://www.soso.com/		菜单
13	11	view	视频	http://v.qq.com/		菜单
14	11	click	赞一下我们	V1001_GOOD	V1001_GOOD	菜单

3、最后在 Foxtable 的菜单或窗口中增加一个按钮，用于根据上面的表生成应用菜单，按钮代码为：

```
For Each AgentID As String In DataTables("Menus").GetValues("AgentId")
    Dim FRows As List(of DataRow) = DataTables("Menus").Select("AgentId = " & AgentID & " And [parent] Is null","_SortKey")
    Dim nms As String() = {"type","name","url","key"} """
    Dim FButtons As New Jarray
    For Each FRow As DataRow In FRows
        Dim SRows As List(of DataRow)
        SRows = DataTables("Menus").Select(CExp("AgentId='{0}' And [parent] = '{1}'",AgentID, FRow("name")),"_SortKey")
        Dim FButton As New JObject
        If sRows.Count = 0 Then
            For Each nm As String In nms
                If FRow.IsNull(nm) = False Then
                    FButton(nm) = FRow(nm).ToString()
                End If
            Next
        Else
            Dim SButtons As New JArray
            For Each SRow As DataRow In SRows
                Dim SButton As New JObject
                For Each nm As String In nms
                    If SRow.IsNull(nm) = False Then
                        SButton(nm) = SRow(nm).ToString()
                    End If
                Next
                SButtons.Add(SButton)
            Next
            FButton("name") = FRow("name").ToString()
            FButton("sub_button") = SButtons
        End If
        FButtons.Add(FButton)
```

```
    Next
    Dim MO As New JObject
    MO("button") = FButtons
                    Dim ul As String =
"https://qyapi.weixin.qq.com/cgi-bin/menu/create?access_token={0}&agentid={1}"
    Dim AccessToken = Functions.Execute("GetQYAccessToken") 'AccessTiken
    Dim hc As New HttpClient(CExp(ul, AccessToken, AgentId))
    hc.Content = Mo.ToSTring
    Dim jo As JObject = JObject.Parse(hc.GetData)
    If jo("errcode") <> "0" Then
        MessageBox.Show(jo.ToString)
    End If
Next
```

10.4.4.3 接收菜单事件

企业号接收菜单事件的代码和公众号相似，只是需要在处理之前先解密事件参数。

需要注意的是，所有向服务器传送图片的菜单命令，会发送两次消息，第一次是菜单命令事件，第二次是普通的 image 消息，第二次才会真正将图片的 MediaID 和 URL 传递给 Foxtable，所以通常并不需要处理图片发送按钮的菜单事件，而是处理常规的 image 消息。

下面是企业号接收处理菜单事件的参考代码，这个代码将接收到的菜单事件记录在一个日志文件中：

```
If e.Path = "wefox" Then
                    Dim wbiz As New
WXBizCrypt("wxa31aba4cd83af57e","foxtable","ilsmyivvRPNj0qxSiSzWCnqm7cy1w1RcS6w2LBhsh
7J")
    If e.Request.HttpMethod = "GET"
        Dim ret As Integer = wbiz.CheckQYSignature(e)
        If ret <> 0 Then '如果接入验证失败
            Dim err As String = wbiz.GetErorDesc(ret) '获取错误描述
        End If
    ElseIf e.Request.HttpMethod = "Post"            '
        Dim st As New Date(1970,1,1,8,0,0)
        Dim msg   As String = wbiz.DecryptMsg(e) '解密
        If  IsNumeric(msg) Then '如果解密
            Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
            Return
        End If
        Dim xo As XObject = XObject.Parse(msg)
        Select Case xo("MsgType")
            Case "text","image","voice","video","shortvideo" '处理消息
```

```
            Dim dr As DataRow = DataTables("Message").AddNew()
            dr("AgentID") = xo("AgentID")
            dr("UserID") = xo("FromUserName")
            dr("CreateTime") = st.AddSeconds(xo("CreateTime"))
            dr("MsgType") = xo("MsgType")
            dr("MsgId") = xo("MsgId")
            dr("MediaId") = xo("MediaId")
            dr("ThumbMediaId") = xo("ThumbMediaId")
            dr("PicUrl") = xo("PicUrl")
            dr("Content") = xo("Content")
            dr("Format") = xo("Format")
        Case "event" '处理事件
            Dim CorpID As String = xo("ToUserName") '企业号的 CorpID
            Dim UserID As String  = xo("FromUserName") '用户的 UserID
            Dim CreateTime As Date = st.AddSeconds(xo("CreateTime")) '触发事件时间
            Dim AgentID As String = xo("AgentID") '应用 ID
            Dim key As String = xo("EventKey") '如果是菜单事件,用于获取按钮的 Key 或 URL,其它事件返回空
            Dim logFile As String  = "c:\data\wxlog" & Format(Date.Today,"yyyyMMdd") & ".txt" '你可以选择每日或每月一个日志文件
            Select Case xo("Event")
                Case "subscribe" '关注事件
                Case "unsubscribe" '取消关注事件
                Case "LOCATION" '上报地理位置事件
                Case "enter_agent" '进入应用事件
                Case "click" '普通单击事件
                    Filesys.WriteAllText(logFile , UserID & "于" & CreateTime & "单击按钮:" & Key & vbcrlf, True)
                Case "view" '单击跳转到网页事件
                    Filesys.WriteAllText(logFile , UserID & "于" & CreateTime & "访问网页:" & Key & vbcrlf, True)
                Case "scancode_push","scancode_waitmsg" '两种扫描二维码事件
                    Dim ScanResult As String = xo("ScanCodeInfo")("ScanResult") '获取二维码扫描结果
                    Filesys.WriteAllText(logFile , UserID & "于" & CreateTime & "扫描二维码:" & ScanResult & vbcrlf, True)
                Case "pic_sysphoto","pic_photo_or_album","pic_weixin" '三种发送图片事件
                    Dim Count As Integer = xo("SendPicsInfo")("Count") '获取图片数量
                    Filesys.WriteAllText(logFile , UserID & "于" & CreateTime & "提交了:" & Count & "张图片." & vbcrlf, True)
                Case "location_select" '主动上报地理位置事件
                    Dim lx As String = xo("SendLocationInfo")("Location_X")
                    Dim ly As String = xo("SendLocationInfo")("Location_Y")
                    Dim lb As String = xo("SendLocationInfo")("Label")
                    Dim pn As String = xo("SendLocationInfo")("Poiname")
```

```
                Filesys.WriteAllText(logFile , UserID & "于" & CreateTime & "上报位置:" & lx & "|" & ly & "|" & lb & "|" & pn   & vbcrlf, True)
            End Select
        End Select
    End If
End If
```

10.4.5 素材接口

10.4.5.1 永久素材接口

永久素材上传后，接口返回一个返回素材资源标识 ID：media_id，这个 media_id 是可以复用的，同一个 media_id 可用于消息的多次发送。

我们首先创建一个名为"Materials"的表，用于保存永久素材信息，表结构如下：

查看表结构

列名	类型	长度	标题
Type	字符型	16	
MediaID	字符型	128	
UpdateTime	日期时间型		
FileName	字符型	48	
Content	备注型		

上传永久图片素材

企业号上传永久图片素材参考代码：

```
Dim url As String = "https://qyapi.weixin.qq.com/cgi-bin/material/add_material?type=image&access_token={0}"
Dim hc As new HttpClient(CExp(url, Functions.Execute("GetQYAccessToken")))
hc.Files.Add("media","c:\data\sample.jpg") '指定要上传的图片文件
Dim jo As JObject = JObject.Parse( hc.getdata)
If jo("errcode") = "0" Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("Type") = "image"
    dr("MediaID") = jo("media_id")
    dr("FileName") = "sample.jpg"
    dr("UpdateTime") = Date.Now
    dr.Save()
```

```
Else
    MessageBox.Show(jo.ToString)
End If
```

上传永久语音素材

企业号上传永久语音你素材参考代码：

```
Dim url As String = "https://qyapi.weixin.qq.com/cgi-bin/material/add_material?type=voice&access_token={0}"
Dim hc As new HttpClient(CExp(url, Functions.Execute("GetQYAccessToken")))
hc.Files.Add("media","c:\data\sample.mp3") '指定要上传的语音文件
Dim jo As JObject = JObject.Parse( hc.getdata)
If jo("errcode") = "0" Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("Type") = "voice"
    dr("MediaID") = jo("media_id")
    dr("FileName") = "sample.mp3"
    dr("UpdateTime") = Date.Now
    dr.Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

上传永久视频素材

企业号上传永久视频素材参考代码：

```
Dim url As String = "https://qyapi.weixin.qq.com/cgi-bin/material/add_material?type=video&access_token={0}"
Dim hc As new HttpClient(CExp(url, Functions.Execute("GetQYAccessToken")))
hc.Files.Add("media","c:\data\test.mp4") '指定要上传的视频文件
Dim jo As JObject = JObject.Parse( hc.getdata)
If jo("errcode") = "0" Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("Type") = "video"
    dr("MediaID") = jo("media_id")
    dr("FileName") = "test.mp4"
    dr("UpdateTime") = Date.Now
    dr.Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

上传永久文件素材

和公众号不同，企业号可以上传普通文件作为永久素材，参考代码：

```
Dim url As String = "https://qyapi.weixin.qq.com/cgi-bin/material/add_material?type=file&access_token={0}"
Dim hc As new HttpClient(CExp(url, Functions.Execute("GetQYAccessToken")))
hc.Files.Add("media","c:\data\订单.xls") '指定要上传的普通文件
Dim jo As JObject = JObject.Parse( hc.getdata)
If jo("errcode") = "0" Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("Type") = "file"
    dr("MediaID") = jo("media_id")
    dr("FileName") = "订单.xls"
    dr("UpdateTime") = Date.Now
    dr.Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

上传永久图文消息

企业号上传永久图文消息的参考代码：

```
Dim ar As New JArray
For i As Integer = 1 To 1 '可以有多段图文
    Dim ao As New JObject
    ao("title") = "标题"
    ao("thumb_media_id") = "2UgIup9iQcz60YIjrpxPFSUTYoPdTAqCz52PXUruWM6isWNdPOQwRgPY3lzJmsKLxPeSWqLMUMREyg9K-4MO8bg" '缩略图 ID
    ao("author") = "作者"
    ao("content_source_url") = "http://www.foxtable.com" '点击"阅读原文"后的 URL
    ao("content") = "图文消息内容,可以包括 HTML 标签"
    ao("digest") = "图文消息摘要"
    ao("show_cover_pic") = "1" '显示封面
    ar.add(ao)
Next
Dim jo As New JObject
jo("mpnews") = New JObject
jo("mpnews")("articles") = ar
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/material/add_mpnews?access_token={0}"
Dim hc As new HttpClient(CExp(ur, Functions.Execute("GetQYAccessToken")))
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("Type") = "mpnews"
    dr("MediaID") = jo("media_id")
```

```
    dr("Content") = ar.ToString
    dr.Save()
Else
    MessageBox.Show(jo("errcode"))
End If
```

10.4.5.2 获取永久素材

通过 media_id 获取上传的图文消息、图片、语音、文件、视频素材。

获取图文消息素材

获取永久图文消息素材，请参考代码：

```
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/material/get?access_token={0}&media_id={1}"
Dim atk As String = Functions.Execute("GetQYAccessToken") 'AccessToken
Dim mid As String = "2Tb6YJ_zqB3FXrUdJRcco6Ck6CHuL4fbLO2u7tB3_yoiEeiJS3jvlMVzpRoE3KcEi" '要获取素材的 ID
Dim hc As new HttpClient(CExp(ur, atk, mid))
MessageBox.Show(hc.GetData)
```

上述代码返回的表示图文消息内容的字符串。

获取其它永久素材

除了永久图文素材返回的是字符串，其它永久素材返回的都是文件。

下面的代码用于下载一个永久图片素材，可以在命令窗口进行测试：

```
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/material/get?access_token={0}&media_id={1}"
Dim atk As String = Functions.Execute("GetQYAccessToken") 'AccessToken
Dim mid As String = "2UgIup9iQcz60YIjrpxPFSUTYoPdTAqCz52PXUruWM6isWNdPOQwRgPY3lzJmsKLxPeSWqLMUMREyg9K-4MO8bg" '要获取素材的 ID
Dim hc As new HttpClient(CExp(ur, atk, mid))
Dim fl As String  = "c:\data\abc.jpg" '要保存为的本地文件
If hc.GetFile(fl) Then
    If hc.ResponseContentType.StartsWith("application/json") Then
        MessageBox.Show(Filesys.ReadAllText(fl))
        FileSys.DeleteFile(fl) '删除文件
    Else
        MessageBox.Show("图片素材下载成功!")
    End If
Else
    MessageBox.Show("图片素材下载失败!")
```

```
End If
```

10.4.5.3 删除永久素材

企业号删除永久素材的代码参考：

```
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/material/del?access_token={0}&media_id={1}"
Dim atk As String = Functions.Execute("GetQYAccessToken") 'AccessToken
Dim mid As String =
"2nOy2I4oiisCFDHBeabDw3EEEVU47sMdqKft_34uqtRwUTiffbCQE0Ez77nSrw2gcIMULxPHXY5CmjZ5MFr
6ySw" '要删除的素材 ID
Dim hc As new HttpClient(CExp(ur, atk, mid))
Dim jo As JObject= JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("素材已经被删除")
Else
    MessageBox.Show(jo.ToString)
End
```

10.4.5.4 获取素材总数

获取永久素材总数的代码参考：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/material/get_count?access_token={0}"
Dim hc As New HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken")))
Dim jo As JObject = JObject.Parse(hc.GetData)
MessageBox.Show(jo.ToString)
```

10.4.5.5 获取素材列表

获取后台所有素材的代码参考：

```
DataTables("Materials").DataRows.Clear()
Dim typs() As String = {"image","voice","video","mpnews","file"}
For Each typ As String In typs
    Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/material/batchget?access_token={0}"
    Dim st As New Date(1970,1,1,8,0,0)
    Dim offset As Integer = 0
    Do
        Dim hc As new HttpClient(CExp(ur, Functions.Execute("GetQYAccessToken")))
        Dim jo As New JObject
        jo("type") = typ
```

```
        jo("offset") = offset
        jo("count") = 20
        hc.Content = jo.ToString()
        jo = JObject.Parse(hc.GetData)
        If jo("errcode") = "0" Then
            Dim Items As Jarray = jo("itemlist")
            If Items.Count = 0 Then
                Exit Do
            Else
                offset = offset + items.Count
            End If
            For Each item As JToken In Items
                Dim dr As DataRow = DataTables("Materials").AddNew
                dr("Type") = typ
                dr("MediaID") = item("media_id")
                dr("FileName") = item("filename")
                dr("UpdateTime") = st.AddSeconds(item("update_time"))
                If item("content") IsNot Nothing Then
                    dr("Content") = item("content")("articles").ToString()
                End If
            Next
        Else
            MessageBox.Show(jo.Tostring) '显示错误信息
            Exit Do
        End If
    Loop
Next
DataTables("Materials").Save()
```

10.4.5.6 上传图文消息内的图片

企业号上传图文消息内使用的图片，可以参考代码：

```
Dim url As String = "https://qyapi.weixin.qq.com/cgi-bin/media/uploadimg?access_token={0}"
Dim hc As new HttpClient(CExp(url, Functions.Execute("GetQYAccessToken")))
hc.Files.Add("media","c:\data\006.jpg") '指定要上传的图片文件
Dim jo As JObject = JObject.Parse( hc.getdata)
If jo("errcode") Is Nothing Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("URL") = jo("url")
    dr("Description") = "永久图文消息内的图片"
    dr.Save()
Else
    MessageBox.Show(jo.ToString)
```

```
End If
```

10.4.5.7 管理临时素材

临时素材上传到微信服务器 3 天后会被自动删除。

上传临时素材

例如上传一个临时图片素材，参考代码为：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/media/upload?access_token={0}&type=image"
Dim hc As new HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken")))
hc.Files.Add("media","c:\data\qrcode.jpg")
Dim jo As JObject = JObject.Parse(hc.GetData)
If jo("media_id") IsNot Nothing  Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("type") = "临时图片"
    dr("MediaID") = jo("media_id")
    dr("FileName") = "qrcode.jpg"
    dr("UpdateTime") = Date.Now
Else
    MessageBox.Show(jo.ToString)
End If
```

上传语音（voice）、视频（video）和普通文件(file)等临时素材的代码和以上代码基本一样，只需将第一行代码的 type 参数（绿色加粗部分）相应地改为 voice、video 或 file 即可。

下载临时素材

下载临时素材的代码参考：

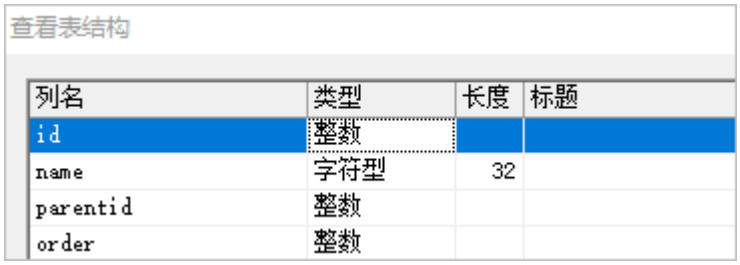
```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/media/get?access_token={0}&media_id={1}"
Dim mediaID As String = "1GDEyyL7RIo868rKmIf3ThZoCMkd69VvH3wPBMpQj9w3uq7dmdxlbNjh669KPLyhF7EEOpBszls7BSmgMFgn07A"  '要获取素材的 ID
Dim hc As new HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken"),mediaID))
Dim fl As String  = "c:\data\abc.jpg" '要保存为的本地文件
If hc.GetFile(fl) Then
    If hc.ResponseContentType.StartsWith("application/json") Then
        MessageBox.Show(Filesys.ReadAllText(fl))
        FileSys.DeleteFile(fl) '删除文件
    Else
        MessageBox.Show("图片素材下载成功!")
    End If
Else
    MessageBox.Show("图片素材下载失败!")
```

```
End If
```

10.4.6 通讯录接口

10.4.6.1 部门接口

首先我们要创建一个表，用于存储部门信息，表名为 Departments，结构如下：



获取部门列表

获取所有部门信息的代码参考：

```
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/department/list?access_token={0}&id={1}"
Dim hc As New HttpClient(CExp(ur,Functions.Execute("GetQYAccessToken"),1))
Dim jo As JObject = JObject.Parse(hc.GetData())
If jo("errcode") = "0" Then
    DataTables("Departments").DataRows.Clear()
    For Each jt As JToken In jo("department")
        Dim dr As DataRow= DataTables("Departments").AddNew()
        dr("id") = jt("id")
        dr("name") = jt("name")
        dr("parentid") = jt("parentid")
        dr("order") = jt("order")
        dr.Save
    Next
Else
    MessageBox.Show(jo.Tostring)
End If
```

创建部门

创建部门的时候，部门 ID 可以由接口自动生成，例如：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/department/create?access_token={0}"
Dim hc As New HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken")))
Dim jo As New JObject
```

```
jo("name") = "辉迅软件技术部"
jo("parentid") = 1 '指定父部门 id,1 表示根部门
jo("order") =  20
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    Dim dr As DataRow = DataTables("Departments").AddNew()
    dr("name") = "辉迅软件技术部"
    dr("parentid") = 1
    dr("order") = 20
    dr("id") = jo("id") '获取自动生成的部门 id
Else
    MessageBox.Show(jo.ToString)
End If
```

修改部门

例如要将 ID 为部门 11 的部门名称改为"辉迅软件市场部"，可以参考代码：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/department/update?access_token={0}"
Dim hc As New HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken")))
Dim jo As New JObject
jo("name") = "辉迅软件市场部" '指定新的部门名称
jo("parentid") = 1
jo("order") =  20
jo("id") = 11
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    Dim dr As DataRow = DataTables("Departments").Find("id = 11")
    dr("name") = "辉迅软件市场部"
    dr.Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

提示：除了部门 ID，其他属性都能修改。

删除部门

例如要删除 ID 为 11 的部门，可以参考代码：

```
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/department/delete?access_token={0}&id=11"
Dim hc As New HttpClient(CExp(ur,Functions.Execute("GetQYAccessToken")))
Dim jo As JObject = JObject.Parse(hc.GetData())
If jo("errcode") = "0" Then
    DataTables("Departments").DeleteFor("id = 11")
```

```
    DataTables("Departments").Save
Else
    MessageBox.Show(jo.Tostring)
End If
```

提示:不能删除有子部门或成员的部门。

10.4.6.2 成员接口

和公众号人人都能关注不同，企业号只有在通讯录中的用户才能关注，本接口用于管理通讯录中的成员。

创建成员

创建成员的代码可以参考：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/user/create?access_token={0}"
Dim hc As New HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken")))
Dim jo As New JObject
jo("userid") = "zhaoliu"
jo("name") =  "赵六"
jo("department") = New Jarray(3,9) '部门 ID 为 3 和 9
jo("position") =  "产品经理"
jo("mobile")= "15913215421"
jo("gender") = "1"
jo("email") = "zhangsan@gzdev.com"
jo("weixinid") = "zhangsan4dev"
jo("avatar_mediaid") =
"2ajDqSlzedZ5A3eFCAb73lbjkSkkbliE0u0oq6xXNtM49xvyJ7O6k7JBC10V5wfR5IONGxoiqG9EOs5H9LB_dzw"
hc.Content = Jo.ToString
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("用户创建成功")
Else
    MessageBox.Show(jo.ToString)
End If
```

提示:

1、创建成员可以指定的属性很多，但是只有 userid、name 和 department 三个属性是必须指定的。
2、mobile、email 和 wexindid 三者必须至少指定一个。

更新成员

例如要将用户"zhaoliu"的职务改为"总经理"，可以参考代码：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/user/update?access_token={0}"
Dim hc As New HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken")))
Dim jo As New JObject
jo("userid") = "zhaoliu"
jo("position") =  "总经理"
hc.Content = jo.ToString
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("用户更新成功")
Else
    MessageBox.Show(jo.ToString)
End If
```

提示：只需指定要修改的属性值即可，未指定值的属性将保持不变。

删除成员

例如要删除 userid 为"zhaoliu"的成员，可以参考代码：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/user/delete?access_token={0}&userid={1}"
Dim hc As New HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken"),"zhaoliu"))
Dim jo As JObject = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("用户删除成功")
Else
    MessageBox.Show(jo.ToString)
End If
```

批量删除成员

可以一次删除多个成员，代码参考：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/user/batchdelete?access_token={0}"
Dim hc As New HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken")))
Dim jo As New JObject
jo("useridlist") = New Jarray("laosan","zhaoliu", "yanggang")
hc.Content= jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("用户删除成功")
Else
    MessageBox.Show(jo.ToString)
End If
```

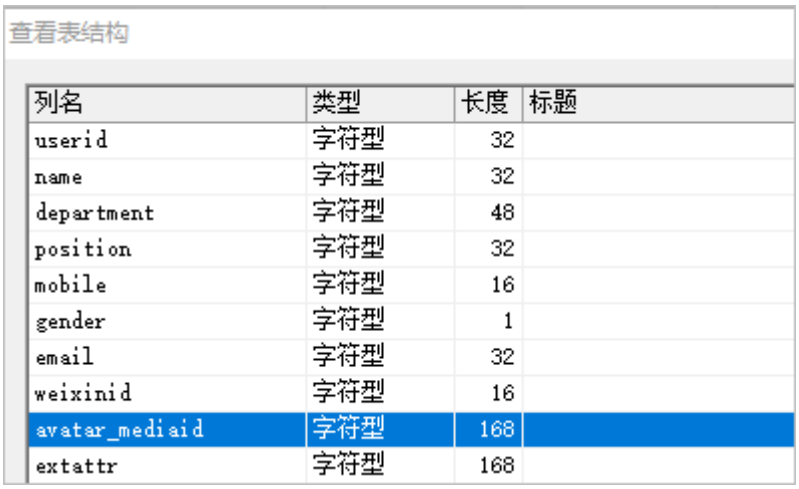
获取指定成员信息

要获取指定成员信息，可参考代码：

```
Dim ul As String  = "https://qyapi.weixin.qq.com/cgi-bin/user/get?access_token={0}&userid={1}"
ul= CExp(ul,Functions.Execute("GetQYAccessToken"),"yanggang")
Dim hc As new HttpClient(ul)
Dim jo As JObject = JObject.Parse(hc.GetData)
MessageBox.Show(jo.ToString)
```

获取成员列表

首先我们新建一个名为 Users 的表，用于管理成员，表结构为：


查看表结构

列名	类型	长度	标题
userid	字符型	32	
name	字符型	32	
department	字符型	48	
position	字符型	32	
mobile	字符型	16	
gender	字符型	1	
email	字符型	32	
weixinid	字符型	16	
avatar_mediaid	字符型	168	
extattr	字符型	168	

实际上我们只能获取指定部门的成员列表，如果要获取所有成员列表，只需将部门 id 设置为 1，因为 1 表示根部门，根部门包括所有成员。

成员列表只返回成员的 userid、department 和 name 三个字段的信息。

获取成员列表保存在 Users 表中，可以参考代码：

```
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/user/simplelist?access_token={0}&department_id=1&fetch_child=1&status=0"
Dim hc As New HttpClient(CExp(ur,Functions.Execute("GetQYAccessToken")))
Dim jo As JObject = JObject.Parse(hc.GetData())
If jo("errcode") = "0" Then
    DataTables("Users").DataRows.Clear
    For Each jt As JToken In jo("userlist")
        Dim dr As DataRow = DataTables("Users").AddNew()
        dr("userid") = jt("userid")
        dr("name") = jt("name")
        If jt("department") IsNot Nothing Then
            dr("department") = CompressJSon(jt("department")).Trim("[","]")
        End If
    Next
    DataTables("Users").Save()
```

```
Else
    MessageBox.Show(jo.ToString)
End If
```

获取成员详情列表

同样要获取所有成员的详情列表，需要将部门 id 参数设置为 1，详情略表能和获取成员的全部信息。

获取成员详情列表保存在 Users 表中，可参考代码：

```
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/user/list?access_token={0}&department_id=1&fetch_child=1&status=0"
Dim hc As New HttpClient(CExp(ur,Functions.Execute("GetQYAccessToken")))
Dim jo As JObject = JObject.Parse(hc.GetData())
If jo("errcode") = "0" Then
    DataTables("Users").DataRows.Clear
    Dim nms() As String = {"userid","name","position","mobile","gender","email","weixinid","avatar","status"} """
    For Each jt As JToken In jo("userlist")
        Dim dr As DataRow = DataTables("Users").AddNew()
        For Each nm As String In nms
            dr(nm) = jt(nm)
        Next
        If dr("department") IsNot Nothing
            dr("department") = CompressJson(jt("department")).Trim("[","]")
        End If
        If jt("extattr") IsNot Nothing Then
            dr("extattr") = ComPressJson(jt("extattr")("attrs")).Trim("[","]")
        End If
    Next
    DataTables("Users").Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

10.4.6.3 标签接口

标签是除部门外，另一种分类用户的方法，实际上部门也可以打标签的，所以标签也可以给部门分类。

创建标签

创建标签的参考代码：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/tag/create?access_token={0}"
Dim hc As New HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken")))
Dim jo As New JObject()
jo("tagname") = "UI"
jo("tagid") = 11
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("标签创建成功")
Else
    MessageBox.Show(jo.ToString)
End If
```

修改标签名称

例如将 tagid 为 11 的标签名改为"UI design"，参考代码：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/tag/update?access_token={0}"
Dim hc As New HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken")))
Dim jo As New JObject()
jo("tagname") = "UI design"
jo("tagid") = 11
hc.Content = jo.ToString()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("标签修改成功")
Else
    MessageBox.Show(jo.ToString)
End If
```

删除标签

例如要删除 tagid 为 11 的标签，参考代码：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/tag/delete?access_token={0}&tagid={1}"
ul = CExp(ul,Functions.Execute("GetQYAccessToken"),11)
Dim hc As New HttpClient(ul)
Dim jo As JObject= JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("标签删除成功")
Else
    MessageBox.Show(jo.ToString)
End If
```

获取标签列表

如果要列出所有标签，可以参考代码：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/tag/list?access_token={0}"
Dim hc As New HttpClient(CExp(ul, Functions.Execute("GetQYAccessToken")))
Dim jo As JObject =  JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    Dim sb As New StringBuilder
    For Each jt As JToken In jo("taglist")
        sb.AppendLine(jt("tagid").ToString & "|" & jt("tagname").ToString)
    Next
    MessageBox.Show(sb.ToString)
Else
    MessageBox.Show(jo.ToString)
End If
```

增加标签成员

可以一次给多个用户或部门打上某个标签，例如下面的代码，给 tagid 为 1 的标签增加了两个成员和一个部门：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/tag/addtagusers?access_token={0}"
Dim jo As New JObject
jo("tagid") = 1
jo("userlist") = New JArray("zhaoliu","zhansan") '要打标签的成员
jo("partylist") = New Jarray(9) '要打标签的部门
Dim hc As New HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken")))
hc.Content = jo.ToString()
jo =  JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("打标签成功")
Else
    MessageBox.Show(jo.ToString)
End If
```

获取标签成员

例如要获取 tagid 为 1 的标签的所有成员，并保存为一个文本文件，参考代码：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/tag/get?access_token={0}&tagid={1}"
ul = CExp(ul,Functions.Execute("GetQYAccessToken"),1)
Dim hc As New HttpClient(ul)
Dim jo As JObject= JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    Dim sb As New StringBuilder
    sb.AppendLine("Users:")
    For Each jt As JToken In jo("userlist")
        sb.AppendLine(jt("name").ToString & "|" & jt("userid").ToString)
    Next
    sb.AppendLine("Parties:")
```

```
    For Each jt As JToken In jo("partylist")
        sb.AppendLine(jt)
    Next
    FileSys.WriteAllText("c:\data\TagUsers.txt",sb.ToString,False)
Else
    MessageBox.Show(jo.ToString)
End If
```

删除标签成员

以下代码从 tagid 为 1 的标签中删除了两个用户和一个部门：

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/tag/deltagusers?access_token={0}"
Dim jo As New JObject
jo("tagid") = 1
jo("userlist") = New JArray("zhaoliu","zhansan") '要删除标签的成员
jo("partylist") = New Jarray(9) '要删除标签的部门
Dim hc As New HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken")))
hc.Content = jo.ToString()
jo =  JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("删除标签成功")
Else
    MessageBox.Show(jo.ToString)
End If
```

10.4.6.4 异步接口

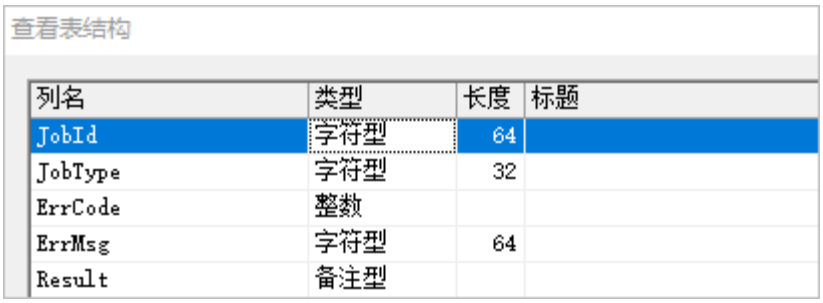
异步任务接口用于大批量数据的处理，提交后接口即返回，企业号会在后台继续执行任务。执行完成后，通过任务事件通知企业获取结果。

异步接口目前只用于通讯录的更新。

用异步接口最大的好处是我们可以用 Excel 编辑好通讯录(csv 格式)，然后通过接口上传到企业号，批量更新通讯录，更新完成后，会发送一个事件通知到 Foxtable 端的 HTTP 服务，我们可以据此获取批量操作的结果。

准备工作

1、首先设计一个名为 BatchJob 的表，用于记录异步操作信息，表结构如下：


查看表结构

列名	类型	长度	标题
JobId	字符型	64	
JobType	字符型	32	
ErrCode	整数		
ErrMsg	字符型	64	
Result	备注型		

2、在 HtppRequest 事件增加代码，用于接收异步操作完成事件，参考代码：

```
If e.Path = "wefox" OrElse e.path = "wefox1"
                Dim wbiz As New WXBizCrypt("wxa31aba4cd83af57e","foxtable","ilsmyivvRPNj0qxSiSzWCnqm7cy1w1RcS6w2LBhsh7J")
    If e.Request.HttpMethod = "GET"
        Dim ret As Integer = wbiz.CheckQYSignature(e)
        If ret <> 0 Then '如果接入验证失败
            Dim err As String = wbiz.GetErorDesc(ret) '获取错误描述
        End If
    ElseIf e.Request.HttpMethod = "POST"            '
        Dim st As New Date(1970,1,1,8,0,0)
        Dim msg  As String = wbiz.DecryptMsg(e) '解密
        If  IsNumeric(msg) Then '如果解密失败
            Dim err As String = wbiz.GetErorDesc(CInt(msg)) '获取错误描述
            Return
        End If
        Dim xo As XObject = XObject.Parse(msg)
        Select Case xo("MsgType")
            Case "text","image","voice","video","shortvideo" '普通消息
            Case "event" '处理事件
                Dim CorpID As String = xo("ToUserName") '企业号的 CorpID
                Dim UserID As String  = xo("FromUserName") '用户的 UserID
                Dim CreateTime As Date = st.AddSeconds(xo("CreateTime")) '发送事件
                Dim AgentID As String = xo("AgentID") '应用 ID
                Dim key As String = xo("EventKey") '如果是菜单事件,用于获取按钮的 Key 或 URL,其它事件返回空|
                Dim logFile As String  = "c:\data\wxlog" & Format(Date.Today,"yyyyMMdd") & ".txt" '你可以选择每日或每月一个日志文件
                Select Case xo("Event")
                    Case "subscribe" '关注事件
                    Case "unsubscribe" '取消关注事件
                    Case "LOCATION" '上报地理位置事件
                    Case "enter_agent" '进入应用事件
```

```
            Case "click" '普通单击事件
            Case "view" '单击跳转到网页事件
            Case "scancode_push","scancode_waitmsg" '两种扫描二维码事件
            Case "pic_sysphoto","pic_photo_or_album","pic_weixin" '三种发送图片事件
            Case "location_select" '主动上报地理位置事件
            Case "batch_job_result" '异步任务完成事件
                Dim Filter As String = "JobId = '" & xo("BatchJob")("JobId").ToString & "'"
                DataTables("BatchJob").AppendLoad(Filter)
                Dim dr As DataRow = DataTables("BatchJob").Find(Filter)
                If  dr IsNot Nothing Then
                    dr("JobType") = xo("BatchJob")("JobType")
                    dr("ErrCode") = xo("BatchJob")("ErrCode")
                    dr("ErrMsg") = xo("BatchJob")("ErrMsg")
                    dr.Save()
                End If
            End Select
        End Select
    End If
End If
```

批量更新成员

异步操作目前可以实现批量更新成员，全员覆盖成员，全员覆盖部门三项操作。

下面以批量更新成员为例，介绍一下操作过程。

1、首先从下载腾讯提供的模板，地址为：

http://qydev.weixin.qq.com/download/batch_user_sample.csv

2、模版文件是 csv 格式，这是一个文本文件，你可以直接用记事本打开编辑，或者用 Excel 打开编辑：

	A	B	C	D	E	F	G
1	姓名	帐号	微信号	手机号	邮箱	所在部门	职位
2	杨刚	yanggang	ygang759	15913112120	zhangsan@gzmailteam.c	9	产品经理
3	谭波	tanbo	tanbo12	18913112121	lisi@gzmailteam.com	2;3	工程师

如果用 Excel 编辑，编辑结束后，不能直接保存，只能执行另存命令，选择文件格式为"csv"保存，切记切记。

注意事项：

- 模板中的部门需填写部门 ID，多个部门用分号分隔，部门 ID 必须为数字
- 文件中存在、通讯录中也存在的成员，更新成员在文件中指定的字段值

- 文件中存在、通讯录中不存在的成员，执行添加操作
- 通讯录中存在、文件中不存在的成员，保持不变
- 成员字段更新规则：文件中有指定的字段，以指定的字段值为准；文件中没指定的字段，不更新

3、假定基于模板编辑好的成员数据文件为"c:\data\users.csv"，我们先要将此文件上传到到微信服务器才能调用，代码为：

```
Dim url As String = "https://qyapi.weixin.qq.com/cgi-bin/material/add_material?type=file&access_token={0}"
Dim hc As new HttpClient(CExp(url, Functions.Execute("GetQYAccessToken")))
hc.Files.Add("media","c:\data\users.csv") '指定要上传的普通文件
Dim jo As JObject = JObject.Parse( hc.getdata)
If jo("errcode") = "0" Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("type")= "file"
    dr("mediaID") = jo("media_id")
    dr("UpdateTime") = Date.Now
    dr("FileName") = "users.csv"
    dr.Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

实际上这是将成员数据文件作为一个素材上传的，具体参考后面的《素材接口》这一章。

4、成员数据文件上传到微信服务器后，会返回一个素材 ID(media_id)，通过下面的代码，可以批量更新成员信息了：

```
Dim url As String = "https://qyapi.weixin.qq.com/cgi-bin/batch/syncuser?access_token={0}"
Dim hc As new HttpClient(CExp(url, Functions.Execute("GetQYAccessToken")))
Dim jo As New JObject
jo("media_id") = "23GjbO9LxhlPSImEU3opz4pFFG7e71Liz7359NOIAOCFa7NVsn1iBuLNguROwqr64vCd2CdzFvIeI3k8VHFmjuQ" '通讯录文件的素材 ID
jo("callback") = New JObject
jo("callback")("url") = "http://120.198.123.19/wefox/"
jo("callback")("token") = "foxtable"
jo("callback")("encodingaeskey") = "ilsmyivvRPNj0qxSiSzWCnqm7cy1w1RcS6w2LBhsh7J"
hc.Content = jo.Tostring()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    Dim dr As DataRow = DataTables("BatchJob").AddNew()
    dr("JobId") = jo("jobid")
    dr.Save()
```

```
Else
    MessageBox.Show(jo.ToString)
End If
```

重要提示：

1、以上提交异步任务的代码，不能在在服务端项目运行，必须在另外的项目运行，否则会提示 40057 错误。
2、两个项目必须共用 BatchJob 表。
3、在开发阶段，如果你不想单独新建项目测试，可以另外启动一个 Foxtable，打开服务端项目(不要开启 HTTP 服务)测试上述代码。

5、服务期在批量更新成员信息完成后，会发送一个通知事件到 Foxtable 的 HTTP 服务，我们之前在 HttpRequwst 设置的代码在接收到事件后，会在 BatchJob 表显示异步任务的执行结果：

	JobId	JobType	ErrCode	ErrMsg	Result
1	8NrJdgcX4AOfDt89k9GpORy72NpqIRPrMpFgN43Bz-U	sync_user	0	ok	

可以通过下面的代码获取异步任务更详细的执行结果，保存在 Result 列中：

```
Dim cr As Row   = Tables("BatchJob").Current
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/batch/getresult?access_token={0}&jobid={1}"
Dim AccessToken As String = Functions.Execute("GetQYAccessToken")
Dim JobId As String = cr("JobId")
Dim hc As New HttpClient(CExp(ul,AccessToken,JobId))
Dim jo As JObject= JObject.Parse(hc.GetData)
cr("Result")= jo.ToString()
```

执行结果的内容通常如下：

```
{
  "errcode": 0,
  "errmsg": "ok",
  "type": "sync_user",
  "total": 2,
  "percentage": 100,
  "remain_time": 0,
  "status": 3,
  "result": [
    {
      "userid": "taobo",
      "errcode": 0,
      "errmsg": "ok",
      "action": 1
    },
    {
```

```
      "userid": "yanggang",
      "errcode": 0,
      "errmsg": "ok",
      "action": 1
    }
  ]
}
```

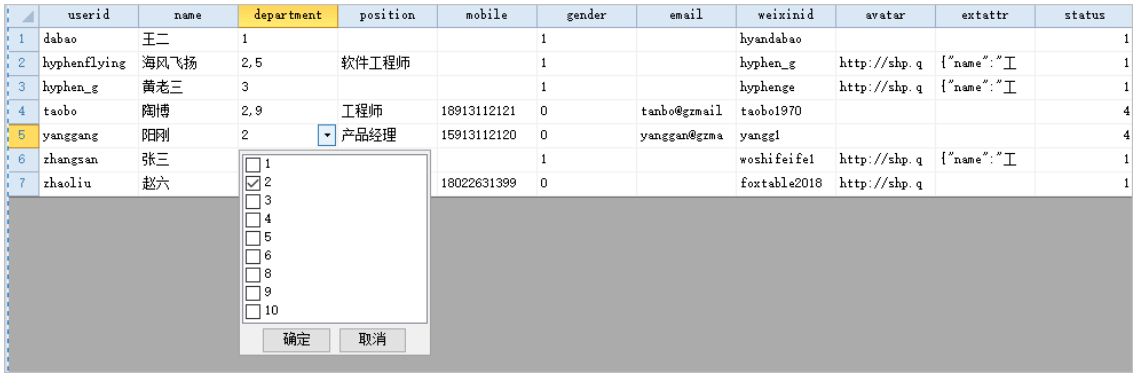

10.4.6.5 一个成员管理系统

前面介绍了微信企业号成员管理接口的使用方法，但是真正基于 Foxtable 设计一个实用的成员管理系统，估计不少用户会感到迷茫。

本节的任务是设计一个简单实用的成员管理系统，你可以通过这个系统成员的增加、删除、修改和批量更新，可以反向操作，从后台下载单个或所有成员信息到系统中。

本系统使用到的表包括 Users(成员信息)、Materials(素材信息)、BatchJob(异步任务)三个表，这些表我们已经在前面的章节设计后，如果你忘记了，请回头复习。

其中最重要的是 Users 表,用于编辑成员信息(你再也不用为生成 CSV 格式的通讯录文件而烦恼了)，建议将 department 列设置为多值字段，方便编辑：



本节的任务是设计一个下图所示的窗口，完成日常的成员管理工作：



下载当前用户信息

你可以在 Users 表中选定一行，然后单击按钮“下载但前用户信息”，即可从微信后台下载此用户的信息覆盖当前行。

代码为：

```
Dim r As Row = Tables("Users").Current
Dim nms() As String = {"userid","name","position","mobile","gender","email","weixinid","avatar","status"} """
Dim ul As String  = "https://qyapi.weixin.qq.com/cgi-bin/user/get?access_token={0}&userid={1}"
ul= CExp(ul,Functions.Execute("GetQYAccessToken"),r("userid"))
Dim hc As new HttpClient(ul)
Dim jo As JObject = JObject.Parse(hc.GetData)
For Each nm As String In nms
    r(nm) = jo(nm)
Next
If jo("department") IsNot Nothing
    r("department") = CompressJson(jo("department")).Trim("[","]")
End If
If jo("extattr") IsNot Nothing Then
    r("extattr") = ComPressJson(jo("extattr")("attrs")).Trim("[","]")
End If
```

下载所有用户信息按钮

```
Dim ur As String = "https://qyapi.weixin.qq.com/cgi-bin/user/list?access_token={0}&department_id=1&fetch_child=1&status=0"
Dim hc As New HttpClient(CExp(ur,Functions.Execute("GetQYAccessToken")))
Dim jo As JObject = JObject.Parse(hc.GetData())
```

```
If jo("errcode") = "0" Then
                    Dim          nms()          As          String          =
{"userid","name","position","mobile","gender","email","weixinid","avatar","status"} """
    For Each jt As JToken In jo("userlist")
        Dim dr As DataRow = DataTables("Users").Find("userid = '" & jt("userid").ToString & "'")
        If dr Is Nothing Then
            dr = DataTables("Users").AddNew()
        End If
        For Each nm As String In nms
            dr(nm) = jt(nm)
        Next
        If jt("department") IsNot Nothing
            dr("department") = CompressJson(jt("department")).Trim("[","]")
        End If
        If jt("extattr") IsNot Nothing Then
            dr("extattr") = ComPressJson(jt("extattr")("attrs")).Trim("[","]")
        End If
    Next
    DataTables("Users").Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

更新\创建用户

此按钮兼具创建用户和更新用户的功能。

单击此按钮，首先检查后台是否存在此用户，如果存在，则用当前行数据更新此用户信息，如果不存在，则根据当前行数据创建一个新的用户。

按钮代码：

```
Dim r As Row = Tables("Users").Current
Dim ul As String  = "https://qyapi.weixin.qq.com/cgi-bin/user/get?access_token={0}&userid={1}"
ul= CExp(ul,Functions.Execute("GetQYAccessToken"),r("userid"))
Dim hc As new HttpClient(ul)
Dim jo As JObject = JObject.Parse(hc.GetData)
If jo("errcode")= "60111" Then '如果不存在此成员
    ul = "https://qyapi.weixin.qq.com/cgi-bin/user/create?access_token={0}" '使用新增成员接口
ElseIf jo("errcode") = "0" Then '如果存在此用户
    ul = "https://qyapi.weixin.qq.com/cgi-bin/user/update?access_token={0}" '使用更新成员接口
Else '如果发生错误
    MessageBox.Show(jo.ToString)'
    Return
End If
hc = New HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken")))
```

```
jo = New JObject
Dim nms() As String = {"userid","name","position","mobile","gender","email","weixinid"}
For Each nm As String In nms
    jo(nm) = r(nm).ToString
Next
jo("department") = New Jarray(r("department").Split(","))
hc.Content = jo.Tostring
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    MessageBox.Show("用户更新/更新成功")
Else
    MessageBox.Show(jo.ToString)
End If
```

更新所有用户信息

单击此按钮会用 Users 表中的数据更 i 性能后台通讯录，对于后台已经存在的用户，会用 User 表中的数据更新其信息，对于后台不存在的用户，会基于 User 表中的数据创建新用户。

其实这个一个按钮完成了三项任务：

1、基于 Users 表的数据生成一个 csv 格式的通讯略文件，此文件必须严格按照腾讯官方的模版生成，且编码必须为 UTF8。
2、上传生成好的通讯录文件到微信服务器。
3、通知微信服务器执行一个异步任务，基于此通讯录文件更新后台通讯录。

具体代码：

```
'第一步:按照腾讯官方提供的模板生成 CSV 格式的通讯录文件
Dim sb As New StringBuilder
sb.AppendLine("姓名,帐号,微信号,手机号,邮箱,所在部门,职位")
Dim nms() As String  = {"name","userid","weixinid","mobile","email","department","position"} """
For Each r As Row In Tables("Users").Rows
    For i As Integer = 0 To nms.Length -1
        If nms(i) = "department" Then
            sb.Append(r(nms(i)).Replace(",", ";")) '注意 CSV 文件中的部门要用分号隔开的.
        Else
            sb.Append(r(nms(i)))
        End If
        If i = nms.Length-1 Then
            sb.AppendLine()
        Else
            sb.Append(",")
        End If
    Next
Next
```

```
Dim fln As String = "c:\data\newuers.csv"
FileSys.WriteAllText(fln,sb.ToString,False,Encoding.utf8)
'第二步:上传生成好的模板文件,并获取 media_id
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/material/add_material?type=file&access_token={0}"
Dim hc As new HttpClient(CExp(ul, Functions.Execute("GetQYAccessToken")))
Dim mid As String   '
hc.Files.Add("media",fln) '指定要上传的普通文件
Dim jo As JObject = JObject.Parse( hc.getdata)
If jo("errcode") = "0" Then
    Dim dr As DataRow = DataTables("Materials").AddNew()
    dr("type")= "file"
    dr("mediaID") = jo("media_id")
    dr("UpdateTime") = Date.Now
    dr("FileName")= FileSys.GetName(fln)
    dr.Save()
    mid = jo("media_id")
Else
    MessageBox.Show(jo.ToString)
    Return
End If
'第三步:向服务器提交异步更新通讯录任务
ul = "https://qyapi.weixin.qq.com/cgi-bin/batch/syncuser?access_token={0}"
hc = New HttpClient(CExp(ul, Functions.Execute("GetQYAccessToken")))
jo = New JObject
jo("media_id") = mid '上一步获取的上传后的通讯录文件的 media_id
jo("callback") = New JObject
jo("callback")("url") = "http://120.198.123.19/wefox/"
jo("callback")("token") = "foxtable"
jo("callback")("encodingaeskey") = "ilsmyivvRPNj0qxSiSzWCnqm7cy1w1RcS6w2LBhsh7J"
hc.Content = jo.Tostring()
jo = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    Dim dr As DataRow = DataTables("BatchJob").AddNew()
    dr("JobId") = jo("jobid")
    dr.Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

关于接收获取异步任务执行完毕通知，以及获取异步任务执行结果，请参考上一节《异步接口》。

删除当前用户

删除当前用户按钮的代码很简单：

```
Dim r As Row = Tables("Users").Current
```

```
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/user/delete?access_token={0}&userid={1}"
Dim hc As New HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken"),r("userid")))
Dim jo As JObject = JObject.Parse(hc.GetData)
If jo("errcode") = "0" Then
    r.Delete
    MessageBox.Show("用户删除成功")
    DataTables("Users").Save()
Else
    MessageBox.Show(jo.ToString)
End If
```

10.4.7 网页授权接口

通过企业号网页授权接口，我们可以在网页系统中，获取当前使用者在企业号中的成员信息。

企业号网页授权流程一般为：

1、将目标网页的地址包装在一个授权链接中，这个链接是指向腾讯的授权服务器的，格式如下：

https://open.weixin.qq.com/connect/oauth2/authorize?appid=CORPID&redirect_uri=REDIRECT_URI&response_type=code&scope=snsapi_base&state=STATE#wechat_redirect
红色部分是需要我们合成进去的：

CORPID	企业号的 CORPID。
REDIRECT_URI	目标网页地址，注意： 1、需要用 UrlEncode 函数进行处理后再合成进去。 2、一定要完整的路径。 你可参考下面的代码生成授权 URL： Dim ul1 As String = "https://open.weixin.qq.com/connect/oauth2/authorize?appid={0}&redirect_uri={1}&response_type=code&scope=snsapi_base&state={2}#wechat_redirect" Dim ul2 As String = UrlEncode("http://wexin.foxtable.com") ul1 = CExp(ul1,"wxa31aba4cd83af57e",ul2,"123") Output.Show(ul1) 详情授权链接的 scope 参数为 snsapi_userinfo，而简单授权链接的 scope 参数为 snsapi_base，二者无其它差别。
STATE	自定义参数，重定向后会在目标网页地址中用 get 方式附上此参数。

2、用户打开目标网页，后台在 cookie 中检查是否存在登录信息，如果存在，直接显示目标网页内容。

3、如果未在 cookie 中发现登录信息，则引导用户跳转到第一步生成的授权链接。

4、微信服务器收到这个授权链接请求后，会生成一个 CODE，然后将 CODE 和上面的 STATE 参数以 get 形式附加到目标网页地址中,格式为：

redirect_uri/?code=CODE&state=STATE

经过授权链接中转后，用户最终访问的就是以上链接，等于又回到了目标网页，只是多了 CODE 和 STATE 参数。

5、根据上一步获得的 CODE 参数调用微信接口，获取用户的 UserId 和 DeviceId，如果非企业成员，获取的则是 OpenId 和 DeviceId。

看起来步骤很多，实际编码却很简单。

一个简单例子

1、在微信后台将网页授权域名设置为"wexin.foxtable.com"，注意不能加"http"。

2、修改 Users 表的结构，增加一个 permit 列，用于标记某个用户是否允许访问系统：

	userid	name	department	position	mobile	gender	email	weixinid	avatar	extattr	status	permit
1	dabao	王二	1	主管		1		hyandabao			1	☑
2	hyphenflyin	海风飞扬	2,5	软件工程师		1		hyphen_g	http://shp.	{"name":"工	1	☐
3	hyphen_g	黄老三	3			1		hyphenge	http://shp.	{"name":"工	1	☐
4	taobo	陶博	2,9	工程师	18913112121	0	tanbo@gzmai	taobo1970			4	☐
5	yanggang	阳刚	2	产品经理	15913112120	0	yanggan@gzm	yangg1			4	☐
6	zhangsan	张三	1,2,9	打杂的		1		woshifeife1	http://shp.	{"name":"工	1	☑
7	zhaoliu	赵六	1,9	总经理	18022631399	0		foxtable201	http://shp.		1	☑

3、Foxtable 端的 HttpRequst 事件代码为：

```
If e.host = "wexin.foxtable.com" Then '需要授权才能访问的域名
    Dim UserId As String
    Dim UserName As String
    Dim sb As New StringBuilder
                                sb.AppendLine("<meta                                name='viewport'
content='width=device-width,initial-scale=1,user-scalable=1'>")
    If  e.GetValues.ContainsKey("code")  Then  '如果通过授权链接跳转而来,就根据传递过来的  code
参数调用接口,获取用户的 UserId
                        Dim          ul          As          String          =
"https://qyapi.weixin.qq.com/cgi-bin/user/getuserinfo?access_token={0}&code={1}"
        ul = CExp(ul,Functions.Execute("GetQYAccessToken"),e.GetValues("code"))
        Dim hc As new HttpClient(ul)
        Dim jo As JObject = JObject.Parse(hc.GetData)
        If jo("UserId") IsNot Nothing Then
            UserId = jo("UserId")
        End If
    Else
```

```
        UserId = e.Cookies("userid") '否则从 cookie 中提取 userid 和 username
    End If
    Dim Verified As Boolean
    Dim dr As DataRow = DataTables("Users").Find("userid ='" & UserId & "'") '根据 openid 找出对应的行
    If UserId  > "" AndAlso dr IsNot Nothing AndAlso dr("permit") = True '授权成功
        Verified  = True
        UserName = dr("name")
        e.AppendCookie("userid",UserId) '将 userid 和 username 存储在 Cookie 中
    ElseIf e.GetValues.ContainsKey("code") = False Then '如果授权失败,且不是通过授权链接跳转而来,那么就跳转到授权链接
                Dim ul As String = "https://open.weixin.qq.com/connect/oauth2/authorize?appid=wxa31aba4cd83af57e&redirect_uri=http%3a%2f%2fwexin.foxtable.com&response_type=code&scope=snsapi_base&state=123#wechat_redirect"
        sb.Append("<meta http-equiv='Refresh' content='0; url=" & ul & "'>") '跳转到授权链接
        e.WriteString(sb.ToString)
        Return
    End If
    If Verified = False Then
        sb.AppendLine("你无权访问本系统")
    Else
        sb.AppendLine("欢迎" & UserName & " , <a href='http://wexin.foxtable.com'>刷新页面</a>")
    End If
    e.WriteString(sb.ToString)
End If
```

10.4.8 在网页实现扫码输入数据

在网页实现扫描二维码或条形码输入数据，需要借助企业号的 JS-SDK。

企业号 JS-SDK 的内容非常多，对于 Foxtable 用户而言，最有用的莫过于在网页实现扫描二维码或条形码输入数据的功能了，下面我们就以此功能为例，介绍一下 JS-SDK 的使用方法。

本节的任务是设计一个下图所示的页面(只能在微信中浏览此页面)，点击“扫码输入”后，能调用微信扫码功能，对二维码或条形码进行扫描，并将扫描结果填入编号输入框：



提示：本节的例子只能在微信访问

设计步骤

1、在微信管理后台设置可信域名，只能在该域名下调用 JS-SDK，域名必须经过备案。

2、新增一个自定义函数，函数名为"GetJsSignature"，用于生成 JS-SDK 的授权签名，代码为：

```
Static CreateTime As Date '记录最近一次生成 Ticket 的时间
Static Ticket As String '记录最近一次生成的 Ticket
Dim tp As TimeSpan = Date.Now - CreateTime
Dim ul As String = "https://qyapi.weixin.qq.com/cgi-bin/get_jsapi_ticket?access_token={0}"
If tp.TotalSeconds > 3600 Then
    Dim hc As New HttpClient(CExp(ul,Functions.Execute("GetQYAccessToken")))
```

```
    Dim ret As String = hc.GetData()
    If ret = "" Then '如果失败,再尝试一次
        hc.GetData()
    End If
    CreateTime = Date.Now()
    Dim jo As JObject = JObject.Parse(ret)
    If jo("errcode") = "0" Then
        Ticket = jo("ticket")
    Else
        MessageBox.show("获取 jsapi_ticket,原因:" & vbcrlf & jo.ToString)
    End If
End If
Dim signature As String = CExp("jsapi_ticket={0}&noncestr={1}&timestamp={2}&url={3}",Ticket,args(0),args(1),args(2))
Return Web.Security.FormsAuthentication.HashPasswordForStoringInConfigFile(signature, "SHA1").ToLower()
```

2、在 d:\web\lib 目录下新建一个文本文件 jssdk.js，内容为：

```
wx.ready(function () {
    document.getElementById('scan').onclick = function () {
        wx.scanQRCode({
            needResult: 1,
            scanType: ['qrCode','barCode'],
            success: function (res) {
                document.getElementById('number').value = res.resultStr;
            }
        });
    };
});
wx.error(function (res) {
    //alert(res.errMsg);
});
```

实际开发的时候，你要变化的只有红色的两处，scan 为网页中扫码输入按钮的 id，number 为网页中产品编号输入框的 id。

3、最后将 HttpRequest 事件代码设置为：

```
Select Case e.Path
    Case "test.htm",""
        Dim wb As New weui
        wb.AppendHTML("<script  src='http://res.wx.qq.com/open/js/jweixin-1.0.0.js'></script>",True) '引入 JS-SDK 库
        wb.AppendHTML("<script src='./lib/jssdk.js'></script>") '引入脚本文件
        '在页面注入权限验证配置
        Dim st As New Date(1970,1,1,8,0,0)
        Dim appid As String = "wxa31aba4cd83af57e" 'CorpID
```

```
    Dim timestamp As Integer = CInt((Date.Now - st).TotalSeconds()) '时间戳
    Dim noncestr As String = Rand.NextString(16) '随机字符
    Dim url As String  = e.Request.URL.ToString '当前页面地址
    Dim signature As String = Functions.Execute("GetJsSignature", noncestr, timestamp, url) '生成权限验证签名
                Dim cfg As String = "wx.config({appId:'{0}',timestamp:{1},nonceStr:'{2}',signature:'{3}',jsApiList:['scanQRCode']});"
    wb.AppendHTML("<script>" & CExp(cfg,appid,timestamp,noncestr,signature) & "</script>",True)
    '开始正常生成网页内容
    wb.AddForm("","form1","test.htm")
    With wb.AddInputGroup("form1","ipg1","产品输入")
      .AddInput("product","产品","text")
      With .AddInputCell("ic1")
        .AddLabel("lbh","编号",0)
        .AddInput("number","text",1)
        .AddVcodeButton("scan","扫码输入",2) '增加二维码扫描按钮,2 表示显示在右边
      End With
    End With
    With wb.AddButtonGroup("form1","btg1",True)
      .Add("btn1", "确定", "submit")
    End With
    e.WriteString(wb.Build) '生成网页
End Select
```

至此，我们的网页扫码输入功能就算设置完成，可以看到 JS-SDK 的使用并不复杂。

如果你希望直接将扫码得到的结果发送到 Foxtable 进行处理，而不是填入输入框，可以参考：SendAjaxText

当然你也可以根须扫描结果合成新的页面地址访问之。

10.5 客户上传文件的下载

客户发送图片、语音和视频文件到公众号或企业号时，Foxtable 的 HTTP 服务收到的消息，会包括一个 MediaId，如果是图片，还有一个 PicUrl 属性，我们通过 MediaId 或 PicUrl 可以下载用户上传的文件。

好消息是用户上传的文件，多数时候都是图片，而 Foxtable 的 PictureBox 控件可以直接显示这些图片，将其绑定到 Message 表的 PicUrl 列即可，不太好的消息是 Foxtabled 的 PictureViewer 不能直接显示这些图片，但这不是大问题，因为一条消息只有一个图片，用不着 PictureViewer。

对于语音和视频文件，必须先下载下来才能打开，建议不要直接在 HTTP 服务所在项目下载这些文件，而是采用单独的项目负责下载和管理，要知道 HTTP 服务必须在 5 秒内对微信服务的访问请求做出相应，否则会认为服务已经停止，所以确保服务端项目处于轻负荷状态是很重要的。

如果你比较专业，可以考虑自己开发多线程的文件下载功能，这样在文件下载的时候不至于阻塞主线程。

11. 更多接口

11.1 阿里云全国物流查询接口

利用 Foxtable 2017，可以很方便第对第三方接口进行开发，实际上微信接口就是第三方接口。

本节再提供一个微信之外的接口开发示例：阿里云全国物流查询接口
此接口需要购买，不过非常便宜，在我写这个文档的时候，你只需花 1 分钱，就可以买 100 次的使用权：

https://market.aliyun.com/products/57126001/cmapi010996.html#sku=yuncode499600008
购买之后，获取 AppCode 进行开发。
购买页面有接口使用说明，按照说明，我们开发出一下功能，每一个功能都只有四行代码，相比阿里云的示例，要简单得多。

假定我们的 AppCode 为：7858a126750949e4a36371580f420413，快递单号为：929601675231
根据快递单号查询快速公司名

参考代码：

```
Dim hc As New HttpClient("http://ali-deliver.showapi.com/fetchCom?nu=929601675231")
hc.Headers.Add("Authorization","APPCODE " & "7858a126750949e4a36371580f420413")
Dim jo As JObject = Jobject.Parse(hc.GetData)
Output.Show(jo.ToString)
```

根据快递单号查询物流信息

参考代码：

```
Dim hc As New HttpClient("http://ali-deliver.showapi.com/showapi_expInfo?com=shunfeng&nu=929601675231")
hc.Headers.Add("Authorization","APPCODE " & "7858a126750949e4a36371580f420413")
Dim jo As JObject = Jobject.Parse(hc.GetData)
Output.Show(jo.ToString)
```

所有代码返回的都是 JSON 数据，你可以对 JSON 数据进行进一步的处理，例如：

```
Dim hc As New HttpClient("http://ali-deliver.showapi.com/showapi_expInfo?com=shunfeng&nu=929601675231")
```

```
hc.Headers.Add("Authorization","APPCODE " & "7858a126750949e4a36371580f420413")
Dim jo As JObject = Jobject.Parse(hc.GetData)
If jo("showapi_res_body")("data") IsNot Nothing Then
    For Each jt As JToken In jo("showapi_res_body")("data")
        Output.Show(jt("time").ToString & " | " & jt("context").ToString)
    Next
End If
```

显示的内容为：

2017-03-02 10:24:37 | 快件在【江门江海集散中心】已装车，准备发往下一站
2017-03-02 05:41:54 | 快件到达 【江门江海集散中心】
2017-03-01 22:31:24 | 快件在【湛江麻章集散中心】已装车，准备发往 【江门江海集散中心】
2017-03-01 22:24:37 | 快件到达 【湛江麻章集散中心】
2017-03-01 20:02:01 | 快件在【湛江市赤坎文保北村营业点】已装车，准备发往 【湛江麻章集散中心】
2017-03-01 19:49:28 | 顺丰速运 已收取快件

查询所有快递公司信息

参考代码：

```
Dim hc As New HttpClient("http://ali-deliver.showapi.com/showapi_expressList?maxSize=500")
hc.Headers.Add("Authorization","APPCODE " & "7858a126750949e4a36371580f420413")
Dim jo As JObject = Jobject.Parse(hc.GetData)
Output.Show(jo.ToString)
```

12. 类型参考

12.1 HttpServer

HttpServer 是一个全局变量，用于开启和关闭 Http 服务。
HttpServer 的属性有：

- **IsRunning**

  逻辑型，用于判断 HppServer 是否正在运行。

- **Prefixes**

  集合，包括所有定义好的 Http 服务。

- **WebPath**

  字符型，用于设置静态文件路径，参考：去掉通用事件头

- **Extensions**
  字符型集合，用于设置可自动发送的静态文件后缀名，参考：去掉通用事件头

HttpServer 的方法有：

- **Start**

  启动 Http 服务。

- **Close**
  关闭 Http 服务。

开启服务

```
HttpServer.Prefixes.Add("http://*/")
HttpServer.WebPath = "d:\web"  '指定静态文件存储位置
HttpServer.Start()
```

默认端口为 80，可以改端口，例如：

```
HttpServer.Prefixes.Add("http://*:32177/")
HttpServer.WebPath = "d:\web"  '指定静态文件存储位置
HttpServer.Start()
```

如果不是默认端口，访问的时候需要加上端口号：

http://192.168.100.1:32177/

关闭服务

```
HttpServer.Close()
```

12.2 HttpRequest

HttpRequest 的 e 参数属性有：

- Host
  字符型，返回客户端访问的服务器 IP 地址(或域名)。

- Port
  整数型，返回客户端访问的服务器端口号。

- Path
  字符型，返回客户端访问的路径，含文件名。

  假定用户输入的访问路径是"http://192.168.0.100:32188/fox/",这三个 e 参数的返回值分别是：

  Host: 192.168.0.100
  Port: 32188
  Path: fox

  假定用户输入访问路径是"http://192.168.0.100:32188/fox/china.htm"，这三个 e 参数的返回值分别是：

  Host: 192.168.0.100
  Port: 32188
  Path: fox\china.htm

- Values
  集合，返回接收到的数据，包括通过 post 和 get 方式提交的数据。

- PostValues
  集合，包括通过 post 方式提交的数据，参考：Values、GetValue 和 PostValues

- GetValues
  集合，包括通过 get 方式提交的数据，参考：Values、GetValue 和 PostValues

- Cookies
  集合，返回接收到的 Cookie，参考：使用 Cookie

- Files
  集合，返回接收到的文件，参考：文件的上传和接收

- PlainText
  字符型，返回收到的纯文本数据，参考：sendAjaxText

- ResponseEncoding
  字符型，用于设置向客户端发送文本内容时采用的编码，参考：网页出现乱码啦

- Request
  返回一个 HttpListenerRequest 对象，适合专业开发人员使用。

- Response
  返回一个 HttpListenerResponse 对象，适合专业人员使用。

- CacheTime
  设置缓存时间，单位为秒，在指定缓存时间内，将不再访问服务器重新下载内容。

- Headers
  用于获取客户端发来的头部信息，参考：再谈头部信息

- ResponseHeaders
  用于设置向客户端发送的头部信息，参考：再谈头部信息

HttpRequest 的 e 参数方法有：

- WriteString
  向客户端发送文本内容，参考：关于 WriteString

- WriteFile
  向客户端发送文件，参考：我想显示一个图片

- WriteDataTable
  向客户端发送 DataTable，需要和 HttpClient 配合使用，参考：接收 DataTable

- SaveFile
  保存接收到的文件，参考：文件的上传与接收

- AppendCookie
  增加 Cookie，参考：使用：使用 Cookie

- Redirect
  跳转到指定页面，例如将 HttpRequst 事件代码设置为：

```
Select Case e.Path
    Case "foxtable.htm"
        e.Redirect("http://www.foxtable.com")
    Case "sina.htm"
        e.Redirect("default.htm")
    Case "default.htm"
        e.WriteString("Hello World")
End Select
```

现在访问地址" http://127.0.0.1/foxtable.htm"，打开的是foxtable的官方网站，而访问地址" http://127.0.0.1/sina.htm"，显示的是页面default.htm的内容：Hello Word。

12.3 HttpClient

HttpClient 的属性有：

属性	说明
Content	字符型，用于设置要发送到服务器的数据，参考：发送数据
ContentType	字符型，用于设置数据类型，默认为"text/plain"，一般无需设置。
FormData	字典，用于添加需要发送到服务器的表单数据，参考：模拟表单
sCookies	字典，用于添加需要发送到服务器的 Cookie，参考：收发 Cookie
rCookies	字典，包括所有服务器返回的 Cookies，参考：收发 Cookie
Files	字典，包括所有需要发送到服务器的文件，参考：上传文件
TimeOut	整数型，用于设置获取服务器响应的超时时限，单位为秒，默认是 100 秒。
Headers	字典，用于添加 HTTP 头，参考：阿里云全国物流查询接口。
UserAgent	获取或设置 User-agent HTTP 标头的值。
ReadWriteTimeout	整数型，用于获取读写服务器数据的超时时限，单位为秒，默认是 300 秒。
RequestEncoding	字符型，用于设置向服务器发送数据时所使用的字符编码，默认为"utf-8"，一般无需设置。
ResponseEncoding	字符型，用于设置解析收到的服务器数据时所采用的字符编码，默认为"utf-8"，一般无需设置。
LastModified	获取服务器返回内容的最近一次更新时间，参考：获取内容信息
StatusCode	获取服务器返回的状态码，参考：获取内容信息
StatusDescription	获取服务器返回的状态字符串，参考：获取内容信息
ResponseContentType	获取服务器返回内容的类型，参考：获取内容信息
ResponseContentLength	获取服务器返回内容的长度，参考：获取内容信息
ResponseHeaders	回去服务器返回内容的头部信息，参考：获取内容信息

HttpClient 的方法有：

方法	说明
GetData	用于接收从服务器返回的数据，参考：接收数据
GetFile	用于接收从服务器返回的文件，参考：接收文件

GetDataTable	用于接收从服务器返回的 DataTable，参考：接收 DataTable

12.4 WeUI

WeUI 的属性有：

属性	说明
Title	字符型，用于设置网页标题
WeUIPath	字符型，用于指定 WeUI 库文件的路径，默认为"/weui"。

WeUI 的方法有：

方法	说明
AddActionSheet	用于增加 ActionSheet，参考：使用 ActionSheet
AddArticle	用于增加 Article，参考：使用 Article
AddButtonGroup	用于增加按钮分组，参考：定义按钮
AddCheckGroup	用于增加复选列表组，参考：复选列表项
AddDialog	用于增加 Dialog，参考：使用 Dialog
AddForm	用于增加表单，参考：定义表单
AddGallery	用于增加 Gallery，参考：使用 Gallery
AddGrid	用于增加 Grid，参考：使用 Grid
AddInputGroup	用于增加输入框分组，参考：文本输入框 数值输入框 日期输入框 列表输入框 InputCell 逻辑开关 多行文本框
AddListGroup	用于增加列表项组，参考：使用 List
AddMsgPage	用于增加 MSgPage，参考：使用 MsgPage
AddPage	用于增加页面，参考：使用 Page
AddPageFooter	用于增加 PageFooter，参考：使用 PageFooter
AddPageTitle	用于增加 PageTitle，参考：使用 PageTitle
AddPanelGroup	用于增加 Panel 分组，参考：使用 Panel
AddPreview	用于增加 Preview，参考：使用 Preview
AddProgress	用于增加 Progress，参考：使用 Progress
AddRadioGroup	用于增加单选列表组，参考：使用单选列表项
AddTabBar	用于增加 TabBar，参考：使用 TabBar
AddTable	用于增加表格，参考：生成表格
AddToast	用于增加 Toast,参考：使用：使用 Toast
AddToptips	用于增加 Toptoips，参考使用：使用 Toptoips
AppendHTML	用于在 head 或 body 的快结束位置附加内容，参考：引用外部文件

InsertHTML	用于在当前位置插入 HTML 代码，参考 InsertHTML
ImageToBase64	用于将图片转换为 base64 字符串，参考：动态生成图表
Save	将 WeUI 的内容保存一个一个文件。 Dim wb As new WeUI With wb.AddMsgPage("","msgpage","操作完成","内容详情,可根据实际需要安排") .AddButton("btn1","确定") .AddButton("btn2","取消").kind = 1 .AddExtra("详细信息","http://www.foxtable.com/") End With wb.Save("d:\web\tst.html")
Build	根据 WeUI 的内容生成一个字符串，这是最常用的一个方法。
AppendCookie	用于添加 Cookie，参考：使用 Cookie
DeleteCookie	用于删除 Cookie，参考：使用 Cookie
ClearCookie	用于清除所有 Cookie，参考：使用 Cookie

12.5 WXBizCrypt

腾讯官方提供了微信消息加解密算法的通用代码， 我们已经将其编译成 DLL，并整合在 Foxtable 中，命名空间为"Tencent.WXBizMsgCrypt"，如果你熟悉用.net 进行微信开发，你可以直接调用。

但是对多数用户来说，直接使用封装在 DLL 中的函数进行加解密还是比较麻烦的，我个人就曾经因为一个参数顺序问题，浪费了一个多小时。

为此我们另外提供了一个类 WXBizCrypt，在腾讯官方代码的基础上进行了简写，更适合非专业人员使用。

定义 WXBizCrypt 类的语法为：

Dim wbiz As New WXBizCrypt(AppId, Token, EncodingAESKey)

AppId	开发者 ID，对于企业号来说，就是 CorpID。
Token	设置回调 URL 时指定的 Token。
EncodingAESKey	设置回调 URL 时指定的 EncodingAESKey。

WXBizCrypt 包括四个函数：

- **DecryptMsg**

  DecryptMsg 函数用于解密消息，语法为：

DecryptMsg(e)

e 为 HttpRequest 事件的 e 参数。

返回值是对 e.PlainText 属性进行解密的结果，如果解密失败，则返回一个错误值。

- **EncryptMsg**

  EncryptMsg 函数用于加密消息，语法为：

  EncryptMsg(e, Message)

  e 为 HttpRequest 事件的 e 参数，Message 为要加密的消息。

  返回值为对 Message 加密后的结果，如果加密失败，则返回一个错误值。

- **GetErorDesc**

  GetErorDesc 函数用于获取错误值对应的错误描述，语法为：

  GetErorDesc(ErrorCode)

  ErrorCode 是一个整数型的错误值。

  函数返回的是 ErrorCode 对应的错误描述。

- **CheckQYSignature**

  CheckQYSignature 专门用于企业号的接入验证，语法为：

  CheckQYSignature(e)

  如果接入验证成功返回 0，否则返回一个错误值。

12.6 JObject

JObject 用于生成和解析 JSON 数据。

参考：

解析 JSON

生成 JSON

12.7 XObject

XObject 用于生成和解析 XML 数据。

参考：

解析 XML

生成 XML

12.8 使用缓存

一些网页文件，例如 css、js、图片文件，很少会频繁更新，为提高效率，可以使用缓存：在一定时间内将文件缓存在客户端，这样客户端再次使用这些文件时，可直接使用缓存文件，无需再次访问服务器下载，以提高效率。
HttpRequest 时间有个 CatcheTime 属性，用于设置缓存时间，单位为秒。
示例

例如希望除 html 和 htm 之外的文件，全部在客户端缓存一个小时：

```
Dim fl As String = "d:\web\" & e.path
If filesys.FileExists(fl)
    Dim idx As Integer = fl.LastIndexOf(".")
    Dim ext As String   = fl.SubString(idx)
    Select Case ext
      Case ".jpg",".gif",".png",".bmp",".wmf",".js",".css" ,".zip",".rar"
        e.CacheTime = 3600 '缓存文件一个小时
        e.WriteFile(fl)
        Return
      Case ".html",".htm"
        e.WriteFile(fl)
        Return
    End Select
End If
```

提示：

1、这只是一个示例，实际上静态的网页也应该缓存的。

2、在缓存期内刷新页面(例如在地址栏再次按回车或单击刷新按钮)，依然会重新访问服务器下载文件。

12.9 常用函数

- **CompressJson**

  CompressJson 用于压缩 JSON 数据，参考：压缩 JSON

- **UrlEncode**

  UrlEncode 用于对字符串进行 URL 编码，例如：

```
Dim ul As String = "http://www.foxtable.com/?page=123"
Output.Show(UrlEncode(ul))
```

  显示结果为：

  http%3a%2f%2fwww.foxtable.com%2f%3fpage%3d123

- **UrlDecode**

  UrlDecode 用于对字符串进行 URL 解码，例如：

```
Dim ul As String = "http%3a%2f%2fwww.foxtable.com%2f%3fpage%3d123"
Output.Show(UrlDecode(ul))
```

  显示结果为：

  http://www.foxtable.com/?page=123

- **HTMLEncode**

  HTMLEncode用于对字符串进行HTML编码，例如：

```
Dim v As String = "1 + 2 <> 3"
Output.Show(HTMLEncode(v))
```

  显示结果为：

  1 + 2 <> 3

- **HTMLEncode**

HTMLDecode用于对字符串进行HTML解码，例如：

```
Dim v As String = "1 + 2 &lt;&gt; 3"
Output.Show(HTMLDecode(v))
```

13. 异步编程

13.1 关于异步编程

一般用户可以忽略本章内容。
现在的电脑基本都是多核的，操作系统可以同时运行多个程序，而一个程序也可以同时执行多个任务。
Foxtable 默认是单线程的，一段代码执行完毕后，才能执行后续代码，从任务角度看，每次只能处理一个任务，多个任务必须排队执行 ，这种单线程执行方式通常称为同步执行。
普通的客户端程序，单线程足以满足要求。
但是对于服务端，例如 OpenQQ 和 HttpServer(高级开发版才有)，如果采用单线程方式，在用户量比较大的情况下， 由于需要排队等候，用户会感到明显的卡顿，甚至出现超时错误。
所以对于服务端程序，最好能同时响应多个用户的访问请求，缩短单个用户的等候时间，这就需要同时开启多个线程，多段代码分别在不同的线程中同时执行，从任务角度看，就是多个任务被同时处理，无需排队等候。
这种多线程编程模式，通常称为异步编程。
传统的异步编程对于专业程序员，也是比较复杂的，所幸 Foxtable 进行了简化，基本上不涉及复杂的概念和技巧，相信大家都能掌握。

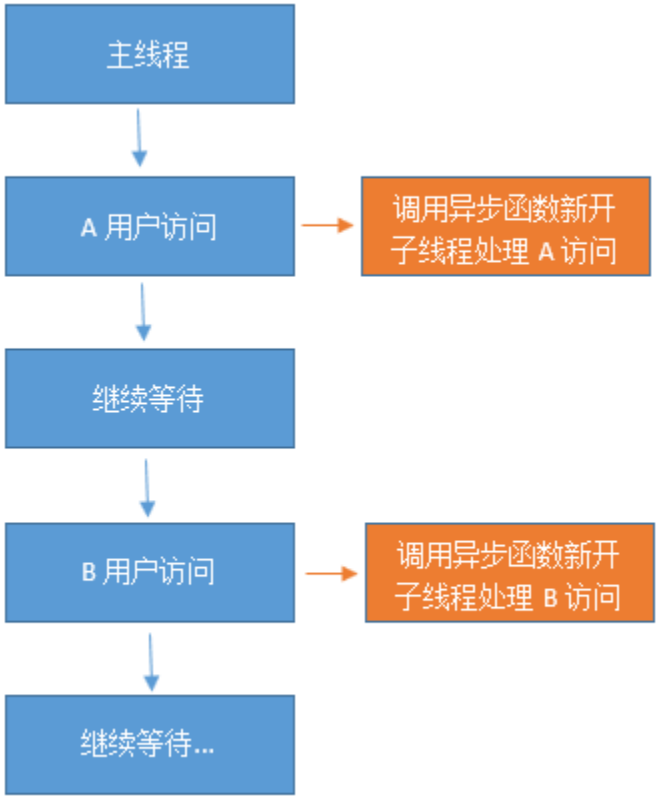
13.2 主线程和子线程

在 Foxtable 中，所有代码，包括 Foxtable 自身的代码，以及用户进行二次开发所编写的代码，默认都在同一个线程中运行，这个线程称为主线程。
UI(用户界面)的显示和刷新，以及对用户操作的响应（例如用户单击某个按钮，选择某个单元格），也都是在主线程中完成的，所以如果主线程经常处于长时间大负荷的运算中，那么用户会感觉程

序有明显的卡顿甚至假死。
对于并发访问量比较大的服务端程序，如果所有访问请求都在主线程中处理，UI 将经常无法响应（当然对于服务端程序，这并不重要），更重要的是单个用户的等候时间也会延长甚至出现超时错误，影响用户感受。
我们可以在主线程之外，另外开启一些线程，用于处理每个用户的访问请求，这种称之为子线程。

开启子线程

Foxtable 用异步函数开启子线程，例如采用异步函数编程的服务端程序，其运行过程如下图：

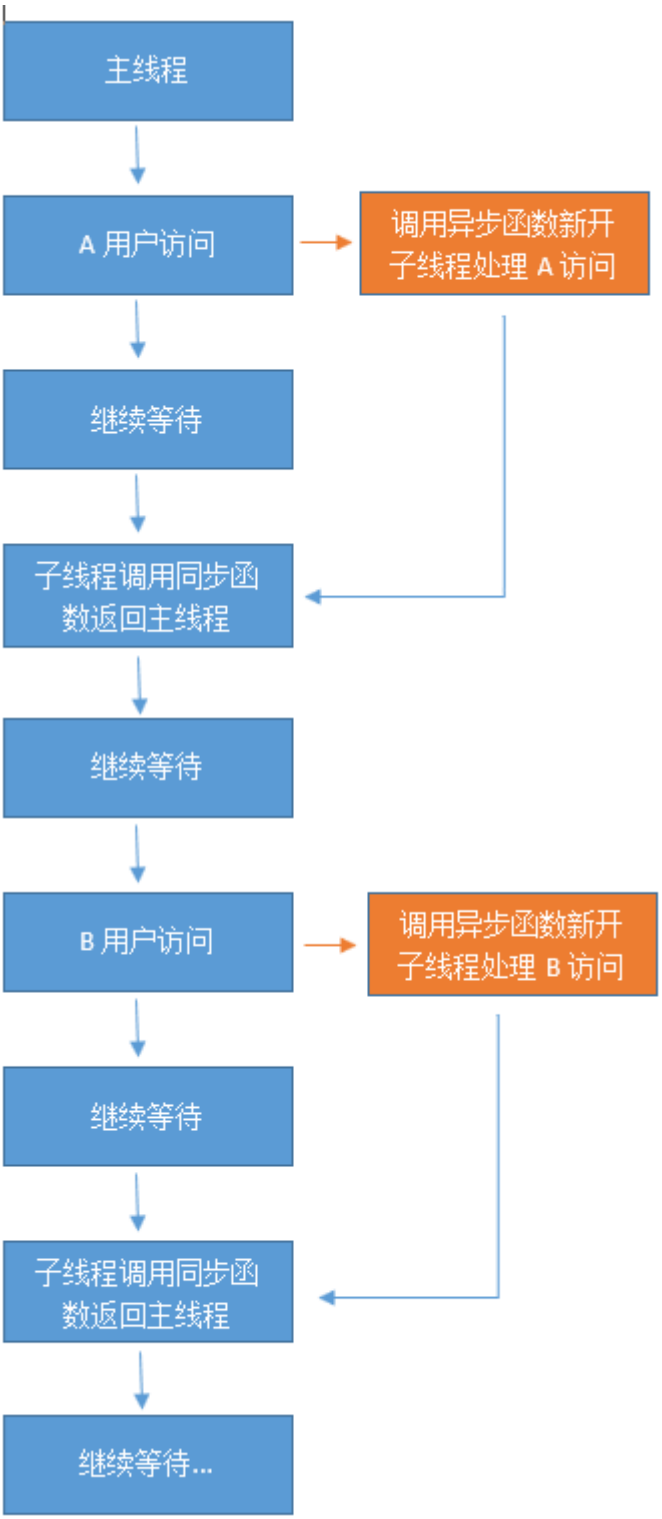


上图中蓝色为主线程，黄色为子线程，可以看到由于每次收到新的用户访问请求后，都会调用异步函数，新开一个线程专门用于处理此用户的访问，所以主线程始终不会被阻塞，可以随时接入新的访问请求。

返回主线程

在主线程中创建的类，例如 DataTable、Table、窗口和菜单等等，都不允许在子线程中访问。

如果收到用户访问请求之后，只是直接读写后台数据，然后将处理结果发送给用户，那么没有问题，而且 Foxtable 增强了 SQLCommand 之后，处理后台数据也更加方便了。
但是确实有时候，子线程需要调用上述对象，例如我们需要在服务端的主界面动态显示在线用户。
还有一种情况，我们开启了多个线程分别负责一部分数据运算工作，完成后需要将计算结果返回给主线程统一处理。
上述情况要求我们必须提供从子线程返回主线程的功能。
子线程通过调用同步函数返回主线程，下面是另一个服务端程序，子线程处理完毕后，通过调用同步函数回到主线程，其运行过程如下图：



图中蓝色为主线程，黄色为子线程，同样主线程不会因为有用户访问而处于阻塞状态，不同的是子线程执行完毕后，调用同步函数返回到主线程继续执行部分代码。

不管是异步函数还是同步函数，都可以传递参数，所以主线程和子线程之间可以相互传递数据。

子线程并非只能执行完毕后才能调用同步函数，实际上子线程运行过程中，可以随时调用同步函数返回主线程执行，等同步函数在主线程执行完毕后（也可以不等待），继续执行子线程中的后续代码，且调用同步函数的次数不限。

真的很简单

综上所述，在 Foxtable 进行异步编程非常的简单：
1、主线程调用异步函数开启一个子线程执行需要异步执行的代码。
2、子线程调用同步函数返回主线程执行需要同步执行的代码。

13.3 异步函数和同步函数

我们在自定义函数的时候，并不存在常规函数、异步函数和同步函数的区分，全部都是一样的定义，参考：自定义函数
那么如何实现异步函数和同步函数的效果呢？
以前调用一个自定义函数函数的语法为：
Functions.Execute(函数名, 参数 1, 参数 2, 参数 3....)
Foxtable 2018 除了 Execute，新增加了三个用于调用函数的方法，分别是 AsyncExecute、SyncExecute 和 BeginSyncExecute，这三个方法的语法和 Execute 完全相同。
同一个函数，如果用 AsyncExecute 方法调用，就是异步函数，如果用 SyncExecute 或 BeginSyncExecute 方法调用，就是同步函数。
现在我们有四个方法用于调用自定义函数了：

方法	说明
Execute	最常用的函数调用方法，可以在主线程使用，也可以在子线程使用。 如果在主线程通过 Execute 方法调用某函数，此函数就运行在主线程。 如果在子线程通过 Execute 方法调用某函数，此函数就运行在子线程。 也就是说，Execute 不会改变函数的运行线程，被调用的函数始终和调用方处于同一个线程中。
AsyncExecute	用异步方式调用函数，一般在主线程使用。 如果在主线程(A)通过 AsyncExecute 调用函数，被调用的函数将在一个新的子线程(B)中运行。 如果在子线程(B)通过 AsyncExecute 调用函数，被调用的函数将在一个新的子线程(C)中运行。 也就是说 AsyncExecute 始终会开启一个新的线程运行函数，所以 AsyncExecute 方法一般仅在在主线程使用。 由于 AsyncExecute 并不会等函数执行完毕，就会执行返回调用方线程继续执行后续代码，所以 AsyncExecute 不会返回函数执行结果。
SyncExecute BeginSyncExecute	用同步方式调用函数，二者区别如下： 子线程通过 BeginSyncExecute 调用函数后，不会做任何的等待，立即返回原来位置继续执行后后续代码，BeginSyncExecute 不会返回函数的结果 子线程通过 SyncExecute 调用函数，会一直等待，直到函数执行完毕，才会返回子线程继续执行后续代码，SyncExecute 会返回函数的结果。 多数时候，我们都会用 BeginSyncExecute 方法调用同步函数，如果希望获取同步函数的结果，可以使用 SyncExecute 方法调用。 二者都只应该在子线程使用，因为在主线程使用毫无意义。

在 Foxtable 进行异步编程，离不开同步函数和异步函数，但并不是真的存在这样的函数，只是通

过不同的调用方法来实现同步和异步的效果，但为便于讲述和理解，我们会继续采用同步函数和异步函数的说法。

13.4 异步函数示例

我们增加一个名为"Task"的函数，代码为：

```
Dim Sum As Integer
For i As Integer = 1 To 10
    Sleep(20) '暂停线程 20 毫秒,模拟需要耗时的任务.
    Sum = Sum + i
Next
Return Sum
```

提示： Sleep 是新增加的一个方法，用于暂停当前线程，这里用于模拟需要耗时的任务。

上述函数执行一次需要时间 200 毫秒(10 * 20)，我们在命令窗口按顺序执行此函数 50 次：

```
Dim dt As Date = Date.Now
For i As Integer = 1 To 50
    Functions.Execute("Task")
Next
Dim tp As TimeSpan = Date.Now - dt
Output.Show(tp.TotalSeconds) '显示总的耗时
```

用时大概 10 秒多一点，和实际计算的时间相符（50 * 200 = 10000 毫秒），在这 10 秒时间内，你无法执行任何操作，单击菜单或表格，都不会有响应。

我们修改一下代码，用 AsyncExecute 代替 Execute：

```
Dim dt As Date = Date.Now
For i As Integer = 1 To 50
    Functions.AsyncExecute("Task")
Next
Dim tp As TimeSpan = Date.Now - dt
Output.Show(tp.TotalSeconds) '显示总的耗时
```

由于改用了 AsyncExecute 调用，所以函数 Task 成为了异步函数

你猜猜用时是多少？ 10 秒？1 秒？0.1 秒？ 还是更少呢？

以上答案都不对，实际上用时是 0 秒，并不是说执行 50 次函数不需要时间，而是这里采用了异步执行方式，每个函数都在独立的线程中运行，不会阻塞主线程，不会导致程序无响应，这就是异步执行的好处。

我们将这种采用异步方式执行的函数，称为异步函数； AsyncExecute 方法并不会返回函数的执行结果，这是可以理解的，因为 AsyncExecute 开启一个新的线程来执行函数后，会立即返回主线程，并不会等函数执行完毕，自然无法获取函数的执行结果。

如何获取异步函数的执行结果呢，我们在下一节讲述。

异步函数的问题

首先我们在全局代码中定义一个 Public 变量：

```
Public Total As Integer
```

关于 Public 变量，参考：Public 变量

首先增加一个名为"AddTotal"的函数，代码为：

```
For i As Integer = 1 To 100
    Total = Total + 1
Next
```

然后在命令窗口用 AsyncExecute 方法异步执行此函数 100 次：

```
Total = 0
For i As Integer = 1 To 100
    Functions.AsyncExecute("AddTotal")
Next
Sleep (2000) '等待 2 秒,待所有子线程线程执行完毕
Return Total
```

你可能认为 Total 的最终结果是 10000，可反复测试后你会发现，Total 只会接近 10000，而且每次执行结果都会不同。

如果你不相信，认为只是一些子线程没有执行完毕而已，那么你可以将 Sleep 的参数增大，延长等待时间，你会发现 Total 的值确实不会 等于 10000。

为什么会这样？因为主线程用 AsyncExecute 方法异步执行 AddTotal 函数 100 次，等于开启了 100 个子线程同时执行 AddTotal 方法；假定某个时刻 Total 的值为 99，此时 A 线程取 Total 的值得到 99，准备加 1，并将结果保存回 Total 变量，假定 A 线程在将结果保存 回 Total 变量之前，另一个线程也取 Total 变量的值，也得到 99，同样加 1 后保存回 Total 变量，这样两个线程执行完毕之后，Total 的值是 100，而不是我们期望的 101。这里只是结果不正确而已，如果你采用 var 变量代替 public 变量，那么不仅仅是结果不正确的问题，程序可能会直接崩溃。

要解决这个问题，必须单独定义一个同步函数用于累加异步函数结果，这是下一节要讲述的内容。

13.5 同步函数示例

本节我们介绍同步函数，同步函数通常用于集中处理异步函数的运行结果。
上一节我们提出了一个问题，用异步函数进行累加，会因为冲突产生错误的结果。
那么如何得到正确的结果呢？
我们可以定义一个函数，这个函数同时只能有一个线程调用，如果有多个线程同时调用此函数，那么就必须排队，上一个线程调用完毕后，下一个线程才会继续，我们称这类函数称为同步函数。
有了同步函数，这样可以开启多个线程执行分别进行计算，然后将结果传递给此同步函数，统一进行处理，以避免上一节提到的冲突问题。

同步函数使用示例

1、定义 Public 变量

首先我们在全局代码中定义一个 Public 变量：

```
Public Total As Integer
```

2、定义同步函数

定义一个名为 AddTotal 的函数，其代码为：

```
Total = Total + Args(0)
```

这个函数将在异步函数中被同步调用，用于累加各异步函数产生的结果。

3、定义异步函数

然后定义一个名为 AddSubtotal 的函数，这个函数将在主线程中被异步调用，其代码为：

```
Dim Sum As Integer
For i As Integer = 1 To 100
    Sum = Sum + 1
Next
Functions .BeginSyncExecute("AddTotal",Sum) '调用同步函数 AddTotal 累加结果
```

4、测试结果

然后在命令窗口执行代码：

```
Total = 0
For i As Integer = 1 To 100
    Functions.AsyncExecute("AddSubtotal")
Next
```

在命令窗口删除以上代码，然后执行：

```
Return Total
```

现在你反复测试，会发现 Total 的值始终都等于 10000。

5、执行过程与原理

主线程用 AsyncExecute 方法异步执行 AddSubtotal 函数 100 次，等于开启了 100 个子线程同时执行 AddSubtotal 方法，各子线程中的 AddSubtotal 方法完成数字累加后，用 BeginSyncExecute 方法同步执行函数 AddTotal，将结果累加到全局变量 Total，因为即使有多个子线程同时用 BeginSyncExecute 方法调用 AddTotal，也只会有一个线程调用成功，其余需要排队等候执行，所以不会再有冲突发生。

6、慎用 Sleep 方法！

认真学习的用户可能已经想到了，这里为啥要两次执行？ 直接在主线程用 Sleep 方法等 2 秒，待子线程执行完毕显示 Total 的值即可啊：

```
Total = 0
For i As Integer = 1 To 100
    Functions.AsyncExecute("AddSubtotal")
Next
Sleep (2000)
Return Total
```

结果出乎意料，Total 的值为 0，为什么呢？

因为同步函数是在主线程中执行的，而上述代码暂停主线程 2 秒，所有子线程通过 BeginSyncExecute 调用 AddTotal 方法时，也都需要排队等候两秒才执行，所以 2 秒后主线程获取 Total 值时，可能还没有一个子线程将结果累加到 Total 变量，所以 Total 的结果为 0。

同样的原因，我们不应该在同步函数中用 Sleep 方法，这会导致主线程和其他所有同步函数处于等待状态，总之除非用于测试，否则：

不要在主线程用 Sleep！

不要在同步函数用 Sleep！

自动等待异步函数执行完毕

作为一个面向非专业人士的开发平台，Foxtable 并未提供等待异步函数执行完毕的功能，实际开发过程中也少有此种需要发生。
如果你确实需要在全部异步函数执行完毕后，自动执行特定的代码，可以参考下面的代码。
继续以前面的异步累加为例：

1、定义 Public 变量

首先我们在全局代码中定义一个 Public 变量：

```
Public Total As Integer
Public TCount As Integer
```

我们在这里增加了一个计数器变量TCount。

2、定义同步函数

定义一个名为 AddTotal 的函数，其代码为：

```
Total = Total + Args(0)
TCount   = TCount - 1
If Tcount = 0 Then
    MessageBox.Show("计算完成,结果为:" & Total)
End If
```

这个函数将在异步函数中被同步调用，用于累加各异步函数产生的结果。

3、定义异步函数

然后定义一个名为 AddSubtotal 的函数，这个函数将在主线程中被异步调用，其代码为：

```
Dim Sum As Integer
For i As Integer = 1 To 100
    Sum = Sum + 1
Next
Functions .BeginSyncExecute("AddTotal",Sum) '调用同步函数 AddTotal 累加结果
```

4、测试结果

然后在命令窗口执行代码：

```
Total = 0
TCount = 100
For i As Integer = 1 To TCount
    Functions.AsyncExecute("AddSubtotal")
Next
```

原理很简单，TCount 变量的初始值为 100，表示异步函数要调用 100 次，每次异步函数(AddSubtotal)执行完毕后，都会调用同步函数(AddTotal)，此函数每执行一次，就将 TCount 减 1，当 TCount 的值等于 0，表示所有的 100 个异步函数都执行完毕了。

13.6 能用的和不能用的

凡是在主线程中创建的对象，例如 Table、Datatable，窗口及窗口控件，菜单及菜单组件，都不应该在异步函数中调用。

我们可以在异步函数中生成临时的 DataTable 使用。
不能在异步函数中读写 Var 变量，不能读写设置等。 Public 变量可以在异步函数中使用，但最好不要跨线程更新数据，否则很容易得到错误的结果。
如果异步函数要使用上述任何对象或数据，请另外定义一个函数来完成之，然后在异步函数中用同步方式调用此函数，有关异步函数如何调用同步函数并传递参数，我们在上一节已经介绍了。
默认不能在异步函数使用 GroupTableBuilder 和 CrossTableBuilder 统计数据，因为二者的数据来自 DataTable，但如果将其 FromServer 属性设置为 True，或者使用 Select 语句作为统计数据来源，则可以在异步函数中使用，因为此时统计的是后台数据。
可以在异步函数中使用 SQLGroupTableBuilder 和 SQLCrossTableBuilder 进行数据统计。
如果需要在异步函数中生成 Excel 报表，请尽量根据后台数据生成，参考：Excel 报表与后台数据
Word 报表无法暂时无法根据后台数据生成，所以目前不要在异步函数中使用 Word 报表，未来应该会改进。
以下类型，都可以直接在异步函数中使用：
SQLCommand
SQLGroupTableBuilder
SQLCrossTableBuilder
SQLJoinTableBuilder
通常我们在服务端事件，例如 HttpRequest 事件中使用这些类，将得到的统计查询结果发送给客户端。

关于数据源的选择

作为桌面数据库，Access 的并发能力非常有限，同时访问的用户一多，很容易崩溃。对于高并发的系统，如果要异步编程访问后台数据库， 请尽量使用 SQL Server。

13.7 异步函数与 UI

我们建议尽量不要在异步函数中有和 UI 相关的代码，否则很容易产生错误。
如果要在异步函数更新 UI，应该定义一个函数用于更新 UI，然后在异步函数同步调用之。
本节的例子和上一节类似，只是希望在状态栏上显示进度条，关注状态栏，请参考：状态栏

1、定义 Public 变量

首先我们在全局代码中定义一个 Public 变量：

```
Public Total As Integer
```

2、定义同步函数

定义一个名为 AddTotal 的函数，其代码为：

```
Total = Total + args(0)
StatusBar.ProgressBar.Value = StatusBar.ProgressBar.Value + 1
StatusBar.Message2 = Total
If  StatusBar.ProgressBar.Value = 100  '如果运行了 100 次,表示全部子线程运行完毕
    MessageBox.show(" 运 行 完 毕 , 累 计 结 果 是 :" & Total, " 提 示 ",
MessageBoxButtons.OK ,MessageBoxIcon.Information)
End If
```

这个函数将在异步函数中被同步调用，用于累加各异步函数产生的结果，在状态栏显示最新结果，并更新进度条。

3、定义异步函数

然后定义一个名为 AddSubtotal 的函数，这个函数将在主线程中被异步调用，其代码为：

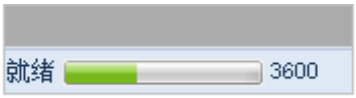
```
Dim Sum As Integer
For i As Integer = 1 To 100
   Sum = Sum + 1
Next
Sleep(1000)  '暂停子线程 1 秒,模拟耗时的操作
Functions.BeginSyncExecute("AddTotal",Sum)  '同步调用函数 AddTotal
```

4、测试结果

然后在命令窗口执行代码：

```
Total = 0
StatusBar.ProgressBar.Visible = True
StatusBar.ProgressBar.Value = 0
For i As Integer = 1 To 100
   Functions.AsyncExecute("AddSubtotal")
Next
```

执行的过程，会在状态栏显示进度以及 Total 的当前值：



在我的电脑测试，上述代码用时 3 秒多完成所有计算，而且运行过程可以继续进行其他操作，不受影响。

我们可以测试一下，不用异步函数执行，将 AsyncExecute 改为 Execute：

```
Total = 0
StatusBar.ProgressBar.Visible = True
```

```
StatusBar.ProgressBar.Value = 0
For i As Integer = 1 To 100
    Functions.Execute("AddSubtotal")
Next
```

用时超过 100 秒，而且运行过程系统卡死，不能进行任何操作，这就是异步执行重负荷代码的好处。

13.8 不要滥用异步编程

专门用一节提示大家不要滥用异步编程，因为这真的很重要。
在 2018 版之前，Foxtable 并没有提供异步编程功能，但一样可以编写出性能优越的管理软件。
对于基于 Foxtbale 开发的大多数客户端程序，异步编程并不会带来性能的提升，只会增加维护和调试难度，甚至影响稳定性。
对于基于 Foxtable 开发的，但并发访问数并不多的服务端程序（OpenQQ 和 HttpServer），通常也没有必要采用异步编程。
只有基于 Foxtable 开发的，且存在高并发访问的服务端程序（OpenQQ 和 HttpServer），或者因为特殊原因，导致服务端需要较长时间才能处理完成单次的访问请求，才可以考虑异步编程。
最后需要提醒的是，线程的创建和销毁，也是需要时间的，一些很简单的任务采用多线程来完成，也许得不偿失。

13.9 HttpServer 与异步编程

13.9.1 HttpServer 与异步编程

HttpRequest 事件中的 e 参数有两个方法，WriteString 和 WriteFile，分别用于向客户端发送文件文本和文件，这两个方法本身就是异步执行的，通常不会影响负载能力。
但 HttpRequest 事件中的代码是在主线程中同步执行的，如果要这里进行一些负荷较重的计算，例如数据的统计查询，复杂网页的生成，将客户上传的文件存储在本地等等，而且用户量比较大，那么就可以考虑异步编程了。
我们并不能简单地将 HttpRequest 事件中的代码转移到自定义函数中，然后在 HttpRequest 事件中用异步方式调用此函数就完事，因为默认主线程执行完毕， 系统就会关闭和客户端的信道，客户端不会显示任何内容，同时因为信道已经关闭，异步函数向客户端发送信息的时候会出错。
所以 HttpRequest 事件采用异步编程，首先要通知系统，我将采用异步编程，主线程完成之后不能关闭信道 ，最后要在异步函数运行结束后通知系统关闭信道。
HttpRequest 事件代码结构为：

```
e.AsyncExecute = True   '通知系统,将采用异步方式生成网页
Functions.AsyncExecute("函数名", e) '异步调用函数生成网页
```

异步函数代码结构为：

```
Dim e As RequestEventArgs = args(0)
'生成网页
'...
'发送网页
e.Handled = True '通知系统异步函数执行完毕,可以关闭信道
```

接下来我们会用两个例子演示HttpServer异步编程。

13.9.2 用异步函数生成网页

首先请回顾一下在网页生成汇总模式的方法，参考：生成汇总模式
本节要完成同样的任务，不过改用异步函数实现，以提高负载能力。
设计步骤：
1、首先增加一个自定义函数 List，这个函数除了第一行和最后一行代码，其余原样来自之前的 Httrequest 事件，所以准备异步函数并不复杂：

```
Dim e As RequestEventArgs = args(0)
'获取要显示的页
Dim page As Integer = 0 '默认 page 为 0,显示第一页
Dim pageRows As Integer = 13 '每页 13 行
If e.GetValues.ContainsKey("page") Then  '如果地址中有 page 参数
    Integer.TryParse(e.GetValues("page"), page) '提取 page 参数
End If
Dim StartRow As Integer = page * pageRows + 1 '此页第一行
Dim EndRow As Integer = (page + 1) * pageRows '此页最后一行
'获取该页数据
Dim cmd As New SQLCommand
cmd.ConnectionName = "orders" '记得设置数据源名称
cmd.CommandText = "Select Count(*) From {订单}"
Dim Count As Integer = cmd.ExecuteScalar() '获取总的行数
cmd.CommandText = "Select * From (Select Row_Number() Over(Order by 产品,客户,日期) As [NO.], 产品, 客户, 数量, 单价, 数量 * 单价 as 金额,日期 From 订单) As a "
cmd.CommandText = cmd.CommandText & "  Where [NO.]>= " & StartRow & " And [NO.] <= " & EndRow
Dim dt As DataTable = cmd.ExecuteReader
'根据此页数据生成网页
Dim wb As New WeUI
```

```
With wb.AddTable("","Table1")
    Dim nms() As String
    Dim qty As Integer
    Dim amt As Double
    .Head.AddRow("No.","产品","客户","数量","单价","金额","日期")
    For i As Integer = 0 To dt.DataRows.count - 1
        Dim r As DataRow = dt.DataRows(i)
        If i > 0 Then
            Dim lr As DataRow = dt.DataRows(i-1)
            If r("客户") <> lr("客户") Then
                qty = dt.compute("sum(数量)","产品='" & lr("产品") & "' And 客户='" & lr("客户") & "'")
                amt = dt.compute("sum(金额)","产品='" & lr("产品") & "' And 客户='" & lr("客户")
& "'")
                With .Body.AddRow()
                    .Attribute = "style='background-color:#F0FFFF'"
                    .AddCell("小计 " & lr("客户"),"colspan='3'")
                    .AddCells(qty,"",amt,"")
                End With
            End If
            If r("产品") <> lr("产品") Then
                qty = dt.compute("sum(数量)","产品='" & lr("产品") & "'")
                amt = dt.compute("sum(金额)","产品='" & lr("产品") & "'")
                With .Body.AddRow()
                    .Attribute = "style='background-color:#FFFFE0'"
                    .AddCell("小计 " & lr("产品"),"colspan='3'")
                    .AddCells(qty,"",amt,"")
                End With
            End If
        End If
        .Body.AddRow(r("NO."),r("产品"),r("客户"),r("数量"),r("单价"),r("金额"),r("日期"))
    Next
    qty = dt.compute("sum(数量)")
    amt = dt.compute("sum(金额)")
    If EndRow >= Count Then
        .Body.AddRow("总计","","",qty,"",amt,"").Attribute = "style='background-color:#98FB98'"
    End If
End With
With wb.AddButtonGroup("","btg2", False)
    If page > 0 Then
        .Add("btnPrev", "上一页","","List.htm?page=" & page - 1)
    End If
    If Endrow < count  Then
        .Add("btnNext", "下一页","","List.htm?page=" & page + 1)
    End If
```

```
End With
e.WriteString(wb.Build)
e.Handled = True '通知系统异步函数执行完毕,可以关闭信道
```

2、将 HttpRequest 事件代码改为：

```
Select Case e.Path
    Case "list.htm"
        e.AsyncExecute = True   '通知系统,将采用异步方式生成网页
        Functions.AsyncExecute("List", e) '异步调用函数生成网页
End Select
```

我们的异步 Http 服务就此搭建完成，符合越大，负载能力提升就越明显。

13.9.3 异步接收数据和文件

首先请回顾一下如何接收上传的图片，参考：接收上传的文件
本节要完成同样的任务，不过文件和数据的接收部分要改为异步执行。
原代码中，用来生成 AddNew.htm 的代码非常简单，负荷极低，如果改为异步执行，创建和销毁线程的时间可能会长于生产网页的时间，得不偿失，所以这部分继续同步执行。
接收数据和文件的部分，负荷要重一些，所以我们改为异步执行。
需要注意的是，在异步函数中不能用 AddNew 增加给 DataTable 增加行，需要改用 SQL 语句增加行，Foxtable 新版本的 SQLCommand 支持参数化，可以更加方便地 完成此类任务，参考：参数化 SQLCommand
此外要注意，可能多个用户同时提交数据，且上传的文件名相同，如果多个子线程同时写一个文件，会因冲突而出错，我们可以在主线程用一个整数变量递增文件序号，或者用随机字符作为文件名，这也是我们本节采用的方法。
设计步骤：
1、新增一个自定义函数 SaveData，代码为：

```
Dim e As RequestEventArgs = args(0)
Dim Name As String   = e.PostValues("姓名")
Dim Xueli As String   = e.PostValues("学历")
Dim Age As Integer = Val(e.PostValues("年龄"))
Dim Files As String = ""
For Each key As String In e.Files.Keys
    If key = "up1" Then
        For Each fln As String In e.Files(key)
            '生成随机文件名,后缀名不变
            Dim newName As String = rand.NextString(10) &   fln.SubString(fln.LastIndexOf("."))
            e.SaveFile(key, fln, ProjectPath & "Attachments\" & newName)
            Files = Files & IIF(Files > "" , vbcrlf, "") & newName
        Next
    End If
```

```
Next
Dim cmd As new SQLCommand
cmd.ConnectionName = "orders"
cmd.CommandText = "Insert Into 员工 (姓名, 年龄, 学历, 照片) Values(?,?,?,?)"
cmd.Parameters.Add("@姓名",Name)
cmd.Parameters.Add("@年龄",Age)
cmd.Parameters.Add("@学历",XueLi)
cmd.Parameters.Add("@照片",Files)
cmd.ExecuteNonQuery()
Dim wb As New WeUI
With wb.AddMsgPage("","msgpage","增加成功", "好好学习,天天向上") '生成成功提示页
    .AddButton("btn1","继续增加","addnew.htm")
End With
e.WriteString(wb.Build)
e.Handled = True   '通知系统异步函数执行完毕,可以关闭信道
```

2、将 HttpRequest 事件代码改为：

```
Dim wb As New weui
Select Case e.Path
    Case "addnew.htm"
        If e.PostValues.Count = 0 Then
            wb.AddForm("","form1","addnew.htm")
            With wb.AddInputGroup("form1","ipg1","增加员工")
                .AddInput("姓名","姓名","Text") '前一个"姓名"是ID,后一个"姓名"是标题
                .AddInput("年龄","年龄","number")
                .AddSelect("学历","学历","大专|本科|硕士|博士")
                .AddUploader("up1","照片",True) 'True表示允许上传多个文件
            End With
            With wb.AddButtonGroup("form1","btg1",True)
                .Add("btn1", "确定", "submit")
            End With
            e.WriteString(wb.Build)
        Else
            e.AsyncExecute = True   '通知系统异步执行,不要关闭信道
            Functions.AsyncExecute("SaveData",e) '异步调用函数 处理接收到的数据和文件
        End If
End Select
```

13.10 OpenQQ 与异步编程

13.10.1 QQServer 与异步编程

QQServer 的负载远低于 HttpServer，一般并无采用异步编程的必要，但我们依然为 QQServer 提供了异步编程的功能。
QQServer 有三个事件和 HttpRequest 一样，在调用异步函数之前，需要将 e 参数的 AsyncExcute 属性设置为 True，保持信道不被关闭；在异步函数执行完毕之后，将 e 参数的 Handled 属性设置为 True，通知系统关闭信道。
这三个事件别为：
UserLogging(e 参数类型为 Server_UserLoggingEventArgs)
ReceivedMessage（e 参数类型为 Server_MessageEventargs）
UserChangePassword(e 参数类型为 Server_ChangeUserPasswordEventArgs)

提示：如果 ReceivedMessage 和 UserChangePassword 不需要通过 e 参数的 ReturnValue 属性返回信息，则无需保持信道。

以UserLogging事件为例，异步编程的时候，HttpRequest事件代码结构为:

```
e.AsyncExecute = True   '通知系统不要关闭信道
Functions.AsyncExecute("函数名", e) '异步调用函数处理登录请求
```

异步函数代码结构为:

```
Dim e As Server_UserLoggingEventArgs = args(0)
'处理登录请求
'...
e.Handled = True '通知系统关闭信道
```

13.10.2 QQServer 异步登录示例

在 CaseStudy 目录下有个文件: QQServer.Table，此文件的设计请参考：设置用户登录事件

现在我们将此文件的 UserLogging 事件改为异步执行，设计步骤：

1、首先准备一个名为 ValidateLog 事件，代码如下，注意三行加粗的代码：

```
Dim e As Server_UserLoggingEventArgs= args(0)
```

```
Dim pts() As String = e.UserName.Split(".")
Dim cmd As New SQLCommand
'cmd.ConnectionName = "数据源名称"
cmd.CommandText = "Select 密码 From {用户} Where 部门 = '{0}' And 姓名 = '{1}'"
Cmd.CommandText = CExp(cmd.CommandText,pts(0), pts(1))
Dim pwd As String = cmd.ExecuteScalar()
If pwd <> e.Password Then '如果密码错误
    e.Message = "登录密码错误!" '返回给登录者的提示信息
    e.Handled = True '退出前一定要将 Handled 设置为 True
    Return Nothing
End If
'添加本部门的用户为好友
cmd.CommandText = "Select 部门,姓名 From {用户} Where 部门 = '" & pts(0) & "'"
For Each dr2 As DataRow In cmd.ExecuteReader().DataRows
    Dim nm As String = dr2("部门") & "." & dr2("姓名")
    If nm <> e.UserName Then '如果不是登录者本人
       e.Buddies.Add(nm)
    End If
Next
e.Success = True '允许用户登录
e.Handled = True '退出前一定要将 Handled 设置为 True
```

2、将 QQServer 的 UserLoging 事件代码改为：

```
e .AsyncExecute = True '通知系统不要关闭信道
Functions.AsyncExecute("ValidateLog", e) '异步调用函数处理登录请求
```

14. Web 数据源

14.1 关于三层架构和 Web 数据源

Foxtable 可以直接连接后台数据库，默认是两层架构，客户端是 Foxtable，服务端是数据库。
Foxtable 2018 开始提供 Web 数据源，服务端可以创建一个本地数据源，然后通过 HttpRequest 事件公开给客户端。
客户端不再直接和服务端的数据库打交道，而是和服务端的 HttpRequest 事件交互，现在客户端和服务端中间多了一个 Web 层，成了三层结构。
相当多的企业因各种原因，不允许在网络上开放数据库端口，而 Web 数据源和三层架构的出现，为他们解决了这个大问题。
Web 数据源和传统数据源有本质区别，但是 Foxtable 的开发人员做了大量的工作，使得 Web 数据源和传统数据源在开发和使用上没有任何区别，原二层架构系统的代码在转为三层架构之后，代码不需要任何的改变。

我们可以在一分钟内将一个二层结构的管理系统转换为三层结构，或将一个三层结构的管理系统转换为两层结构。

目前已知的区别只有一点：三层结构的系统，不能在客户端增加表、删除表或修改表结构，但三层结构的管理系统，已经有服务端项目，本就应该在服务端项目设计表，所以这不是一个问题，而是开发人员有意而为之。

在 Web 数据源出现之前，如果要开发基于互联网的管理系统，必须使用 SQL Server，现在我们也可以用 Access 了，大大简化服务端的搭建和准备工作，当然由于 Access 的并发量和负载能力都有限，所以仅适合搭建一些中小型的管理系统，大的管理系统依然应该采用 SQL Server 数据库。

14.2 在服务端建立 Web 数据源

要使用 Web 数据源，必须有一个服务端项目，此项目运行在服务器，用于向客户端提供 Web 数据源。

Web 数据源的建立非常简单，如果不需要身份验证，只需一行代码就能完成 Web 数据源的建立工作。

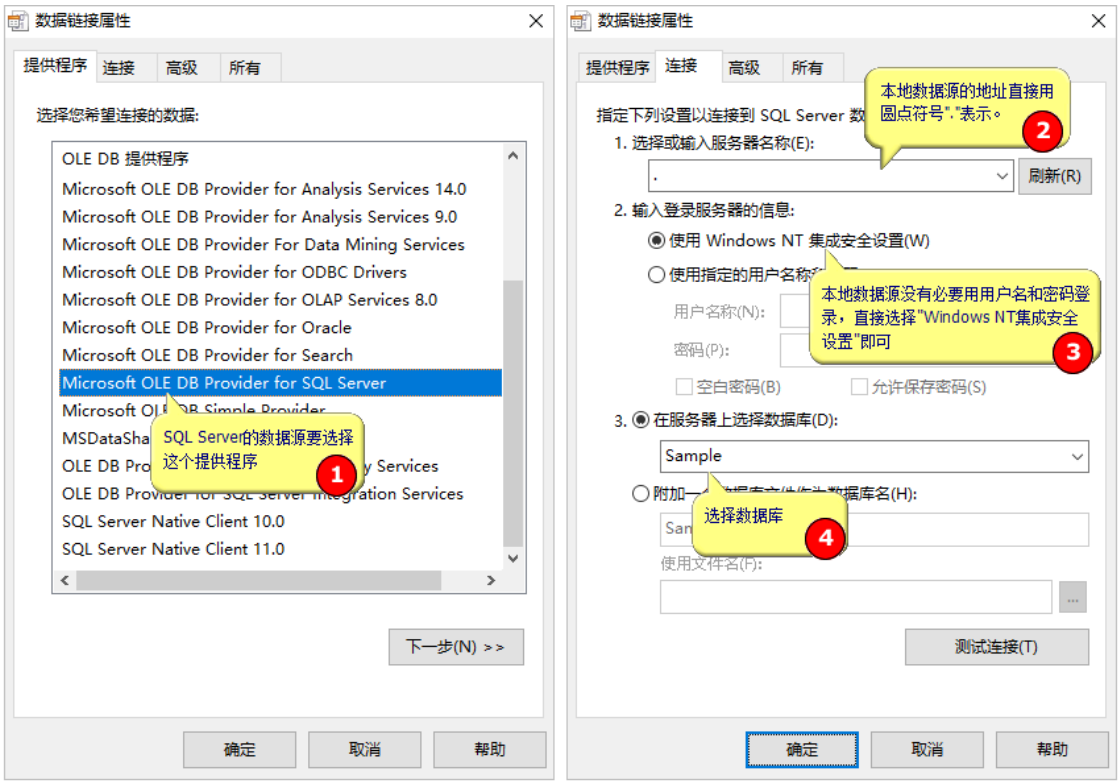
设计步骤：

1、首先我们在服务端的 AfterOpenProject 事件中加上以下代码，用于开启 Web 服务：

```
HttpServer.Prefixes.Add("http://*/")
HttpServer.WebPath = "d:\web"
HttpServer.Start()
```

如果你的服务端项目仅用于提供数据源服务，那么第二行代码可以删除。

2、假定服务器已经安装了 SQL Server，有一个名为 Sample 的数据源，我们现在建立一个数据源连接到这个数据库，使用生成器生成连接字符串的设置如下图：



生成的连接字符串为：

Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=Sample;Data Source=.

如果你使用的是 Access 数据库，那么建立数据源的过程和以前没有区别，这里就不再赘述。

4、假定服务端建立的本地数据源的名称为"Orders"，现在将 HttpRequest 事件代码设置为：

```
Dim Verified As Boolean
If e.PostValues.ContainsKey("username") AndAlso e.PostValues.ContainsKey("password") Then
    '实际开发的时候,请改为根据用户表验证身份
    Dim username As String   = e.PostValues("username")
    Dim password As String   = e.PostValues("password")
    If username = "张三" AndAlso password = "888" Then
        Verified   = True
    End If
End If
If Verified = False Then
    e.AppendCookie("Error","用户身份验证失败!") '通过 Cookie 返回错误信息.
    Return
End If
Select Case e.Path
    Case "DataServer.htm"
        e.AsDataServer("Orders")  '将一个本地数据源公开为 Web 数据源
End Select
```

提示：

1、AsDataServer 方法用于将本地数据源转为 Web 数据源，并对外公开，其参数为本地数据源名称。

2、AsDataServer 本身就是一个异步方法，可同时处理多个用户的访问请求，没有必要再次对其做异步处理。
至此我们的 Web 数据源就搭建完毕了，如果你的 Web 服务仅用于提供数据源，而且无需身份验证，那么可以将代码简化为一行：

```
e.AsDataServer("Orders")
```

服务器可以同时公开多个本地数据源，但是要注意区分路径，例如：

```
'省略的身份验证代码
Select Case e.Path
    Case "DataServer.htm"
        e.AsDataServer("Orders")
    Case "Sales.htm"
        e.AsDataServer("Sales")
End Select
```

14.3 在客户端连接 Web 数据源

请确保服务端项目已经启动。
在客户端项目连接 Web 数据源很简单，假定客户端和服务端项目在同一台电脑，为了连接上一节所创建的 Web 数据源，可以按下图所示输入连接字符串：



现在我们就可以像常规数据源一样，从这个 Web 数据源中添加数据表到 Foxtable，一样可以增加、删除和修改数据，一样可以使用加载树和筛选树，一样可以进行各种统计分析，一样可以进行各种开发，源代码也和以前一样，没有任何特殊之处。
连接 Web 数据源时，通常需要附加一些表单数据或 Cookie 数据，服务端据此进行身份验证
Web 数据源连接字符串格式为：
http://地址
-FormData-
键 1:值 1
键 2:值 2
-Cookies-
键 1:值 1
键 2:值 2
第一行为数据源的地址，"-FormData-"行之后为附加的表单数据，"-Cookies-"行之后为附加的 Cookie

数据。

可以看到 Web 数据源的创建和连接都很简单，为了方便，可以在开发过程中采用传统数据源，发布前再根据需要改为 Web 数据源即可，实际上对于已经发布的项目，也可以在传统数据源和 Web 数据源之间 随意切换。

BeforeConnectOuterDataSource 事件

在连接 Web 数据源之前，一样会触发 BeforeConnectOuterDataSource 事件，我们可以在这里动态合成连接字符串，合成字符串的时候，必须严格按照上述格式要求。

例如：

```
If e.Name = "Orders" Then
    Dim sb As New StringBuilder
    sb.AppendLine("http://127.0.0.1/DataServer.htm")
    sb.AppendLine("-FormData-")
    sb.AppendLine("UserName:张三")
    sb.AppendLine("Password:888")
    e.ConnectionString = sb.ToString()
End If
```

运行过程中创建和修改Web数据源

可以在运行过程中创建一个Web数据源，例如：

```
Dim sb As New StringBuilder
sb.AppendLine("http://127.0.0.1/DataServer.htm")
sb.AppendLine("-FormData-")
sb.AppendLine("UserName:张三")
sb.AppendLine("Password:888")
Connections.Add("Sales",sb.ToString)
```

你也可以在运行过程中修改Web数据源的属性，例如：

```
Connections("Orders").DataServer = "http://127.0.0.1/DataServer.htm"
Connections("Orders").FormData("UserName") = "李四"
Connections("Orders").FormData("PassWord") = "999"
Connections("Orders").FormData("AccessToken") = "asgkfd9is45" '添加一个新的表单项
```

提示：

1、上述属性对常规数据源无效。

2、实际开发中，请不要使用UserName和Password这种指向性很明确的词汇作为键名。

3、Foxtable已经占用的键名有： Action、Datatable、IgnoreSchema和Select。请不要在自己的代码中使用以上键名，以免冲突。

判断某个数据源是否可以连通

判断某个数据源是否可以连通的参考代码：

```
Dim sb As New StringBuilder
Dim Err As String
sb.AppendLine("http://127.0.0.1/DataServer.htm")
sb.AppendLine("-FormData-")
sb.AppendLine("UserName:张三")
sb.AppendLine("Password:888")
If Connections.TryConnect(sb.Tostring, Err) = False Then
    MessageBox.Show(err,"提示",MessageBoxButtons.OK,MessageBoxIcon.Error) '显示错误信息
Else
    MessageBox.Show("数据源可以正常连通!","提示",MessageBoxButtons.OK,MessageBoxIcon.Information)
End If
```