

高等院校信息技术规划教材

.NET Web 企业应用开发实战

黄锐军 编著

清华大学出版社

高等院校信息技术规划教材

.NET Web 企业应用开发实战

黄锐军 编著

清华大学出版社
北京

内 容 简 介

本书通过企业项目开发案例深入介绍.NET Web 企业应用开发高级技术,共包含 4 个综合实践项目。第一个项目是基于百度的天气预报查询程序,讲解数据的 XML 与 JSON 序列化问题以及客户端与网站服务器数据上传与下载的方法。第二个项目是基于 Web 的图片共享程序,讲解通过自定义协议客户端与服务器的交互问题来实现数据上传与下载。第三个项目是基于微信的成绩查询程序,讲解 Web Service 程序的技术方法以及微信公众号程序的开发技术。第四个项目是基于 WCF 的试题练习程序,讲解 WCF 服务器与客户端程序的开发技术。所有项目的服务器程序大都采用三层架构设计思想,客户端采用 WPF 窗体程序。

本书的特点是实践性强,由浅入深,循序渐进,每个项目都分解成若干节,每节以案例展示、技术要点、服务器程序、客户端程序、拓展训练的结构展开。

本书可以作为高等院校的教材,也可以作为相关技术人员学习.NET Web 程序设计的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

.NET Web 企业应用开发实战/黄锐军编著. —北京:清华大学出版社,2017
(高等院校信息技术规划教材)
ISBN 978-7-302-46531-7

I. ①N… II. ①黄… III. ①计算机网络—程序设计—高等学校—教材 IV. ①TP393.09

中国版本图书馆 CIP 数据核字(2017)第 025620 号

责任编辑:张 玥 战晓雷

封面设计:常雪影

责任校对:白 蕾

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印装者:三河市春园印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:17

字 数:393 千字

版 次:2017 年 5 月第 1 版

印 次:2017 年 5 月第 1 次印刷

印 数:1~2000

定 价:39.50 元

产品编号:072803-01

前言

foreword

目前国内有大量关于 ASP.NET 网站设计的高校教材,但是微软公司的 .NET Web 技术远远不限于 ASP.NET 网页设计,还有更高级别的 Web Service 及 WCF 等技术。在引进的国外教程中有一些涉及这方面的技术,但是这些书籍一般内容很杂,实践证明这些书籍教学效果欠佳,不适合用作高校的教材。针对我国高校的教学实际,作者编写了这本由企业项目驱动的教程。

本教程包含 4 个项目,第一个项目是基于百度的天气预报查询程序,讲解数据的 XML 与 JSON 序列化问题,同时讲解客户端与网站服务器数据上传与下载的方法。第二个项目是基于 Web 的图片共享程序,讲解通过自定义协议客户端与服务器的交互问题来实现数据上传与下载。第三个项目是基于微信的成绩查询程序,讲解 Web Service 程序的技术方法,同时讲解微信公众号程序的开发技术。第四个项目是基于 WCF 的试题练习程序,讲解 WCF 服务器与客户端程序的开发技术。这几个项目的客户端大都设计成 WPF 的窗体程序。

要学好一门编程技术,不应该拘泥于该技术的规范细节,而应该大量使用该技术来编写程序,在实践中学习与巩固基本知识,锻炼编程能力。本书的特点是实践性强,每节内容都采用案例展示、技术要点、服务器程序、客户端程序、拓展训练的结构展开。读者先通过“案例展示”了解要做什么,通过“技术要点”学习要用到什么技术,通过“服务器程序”与“客户端程序”学习程序的编写方法,最后通过“拓展训练”巩固与拓展所学习的知识。

每个项目都配有一些编程练习,有些练习项目是比较复杂的,需要读者花费比较长的时间来完成。

本教程可以作为高等院校的教材,建议总学时安排在 60 学时左右,其中讲授与上机实习的学时比例为 1:1 左右。

本书是作者省级精品资源在线开放课程“DotNetWeb 编程应用程序实践”的配套教材,课程网站 <http://www.dotnetweb.com.cn> 提供大量的教学资源,课程还开通了微信公众号 DotNetWeb 微课教学。

由于作者水平有限,书中难免有错误或不足之处,敬请广大读者批评指正。



微信公众号: DotNetWeb

作 者

2017 年 1 月于深圳

目录

Contents

项目一 基于百度的天气预报程序	1
1.1 服务器与客户端程序	2
1.1.1 案例展示	2
1.1.2 技术要点	2
1.1.3 服务器程序	2
1.1.4 客户端程序	5
1.1.5 拓展训练	6
1.2 XML 数据的网络传输	8
1.2.1 案例展示	8
1.2.2 技术要点	8
1.2.3 服务器程序	10
1.2.4 客户端程序	12
1.2.5 拓展训练	14
1.3 JSON 数据的网络传输	15
1.3.1 案例展示	15
1.3.2 技术要点	15
1.3.3 服务器程序	16
1.3.4 客户端程序	18
1.3.5 拓展训练	20
1.4 城市天气状况查询	20
1.4.1 案例展示	20
1.4.2 技术要点	21
1.4.3 服务器程序	23
1.4.4 客户端程序	23
1.4.5 拓展训练	25
1.5 城市天气预报查询	26
1.5.1 案例展示	26

1.5.2	技术要点	27
1.5.3	服务器程序	31
1.5.4	客户端程序	31
1.5.5	拓展训练	34
1.6	城市天气预报程序的实现	34
1.6.1	技术要点	34
1.6.2	服务器程序	37
1.6.3	客户端程序	37
1.6.4	拓展训练	44
练习一		51
项目二	基于 Web 的图片共享程序	54
2.1	用户信息的发送	55
2.1.1	案例展示	55
2.1.2	技术要点	55
2.1.3	服务器程序	57
2.1.4	客户端程序	61
2.1.5	拓展训练	63
2.2	用户注册与登录	65
2.2.1	案例展示	65
2.2.2	技术要点	65
2.2.3	服务器程序	69
2.2.4	客户端程序	70
2.2.5	拓展训练	71
2.3	图片上传与存储	72
2.3.1	案例展示	72
2.3.2	技术要点	72
2.3.3	服务器程序	75
2.3.4	客户端程序	76
2.3.5	拓展训练	79
2.4	图片共享与设置	79
2.4.1	案例展示	79
2.4.2	技术要点	80
2.4.3	服务器程序	82
2.4.4	客户端程序	83
2.4.5	拓展训练	87
2.5	图片下载与浏览	88
2.5.1	案例展示	88

2.5.2	技术要点	89
2.5.3	服务器程序	90
2.5.4	客户端程序	92
2.5.5	拓展训练	95
2.6	图片共享程序的实现	97
2.6.1	技术要点	97
2.6.2	服务器程序	100
2.6.3	客户端程序	108
2.6.4	拓展训练	120
练习二		121
项目三	基于微信的成绩查询程序	122
3.1	课程记录管理	124
3.1.1	案例展示	124
3.1.2	技术要点	125
3.1.3	服务器程序	129
3.1.4	客户端程序	131
3.1.5	拓展训练	135
3.2	学生记录管理	135
3.2.1	案例展示	135
3.2.2	技术要点	136
3.2.3	服务器端程序	138
3.2.4	客户端程序	139
3.2.5	拓展训练	143
3.3	成绩记录管理	145
3.3.1	案例展示	145
3.3.2	技术要点	146
3.3.3	服务器程序	149
3.3.4	客户端程序	149
3.3.5	拓展训练	153
3.4	微信开发平台搭建	154
3.4.1	案例展示	154
3.4.2	技术要点	154
3.4.3	服务器申请	156
3.4.4	服务器绑定	156
3.4.5	拓展训练	158
3.5	微信事件与文本回复	159
3.5.1	案例展示	159

3.5.2	技术要点	160
3.5.3	接口程序	162
3.5.4	部署程序	164
3.5.5	拓展训练	165
3.6	微信成绩查询	165
3.6.1	案例展示	165
3.6.2	技术要点	166
3.6.3	接口程序	168
3.6.4	部署程序	170
3.6.5	拓展训练	170
3.7	微信成绩查询程序的实现	172
3.7.1	技术要点	172
3.7.2	服务器程序	175
3.7.3	客户端管理程序	183
3.7.4	拓展训练	185
练习三		188
项目四	基于 WCF 的试题练习程序	189
4.1	用户注册登录	190
4.1.1	案例展示	190
4.1.2	技术要点	191
4.1.3	服务器程序	192
4.1.4	客户端程序	198
4.1.5	拓展训练	201
4.2	试题记录增加	203
4.2.1	案例展示	203
4.2.2	技术要点	203
4.2.3	服务器程序	206
4.2.4	客户端程序	207
4.2.5	拓展训练	210
4.3	试题记录删除	210
4.3.1	案例展示	210
4.3.2	技术要点	211
4.3.3	服务器程序	213
4.3.4	客户端程序	213
4.3.5	拓展训练	218
4.4	试题记录更新	220
4.4.1	案例展示	220

4.4.2	技术要点	221
4.4.3	服务器程序	223
4.4.4	客户端程序	224
4.4.5	拓展训练	229
4.5	用户试题练习	231
4.5.1	案例展示	231
4.5.2	技术要点	231
4.5.3	服务器程序	234
4.5.4	客户端程序	235
4.5.5	拓展训练	239
4.6	试题练习程序的实现	245
4.6.1	技术要点	245
4.6.2	服务器程序	246
4.6.3	客户端程序	250
4.6.4	拓展训练	255
练习四		261

基于百度的天气预报程序

设计一个天气预报程序是很有价值的,问题是如何去找天气的数据。实际上目前很多网站都提供天气预报的实时数据,百度天气预报就是其中之一,可以从这些网站上获取任何一个城市的天气预报数据,把它整合到自己的程序中。图 1-1 就是一个 WPF (Windows Premtation Foundation)的窗体程序,启动后可以选择任何一个省份的任何一个城市,就显示出该城市 4 天内的天气预报,数据来自百度天气预报网站。



图 1-1 百度天气预报

这是一个典型的 C/S(Client/Server, 客户/服务器)结构的程序,客户端是一个 WPF 的窗体程序,服务器是提供天气预报的百度网站。但百度是一个 Web 网站,给出的天气预报数据一般是 JSON 结构的,客户端要通过 HTTP 协议与这个网站进行交互,其程序结构如图 1-2 所示。



图 1-2 百度天气预报程序结构

1.1 服务器与客户端程序

1.1.1 案例展示

在 D:\web 目录下创建一个服务器程序 server.ashx, 并把该目录设置为一个 Web 服务站点, 通过浏览器访问地址 `http://localhost/web/server.ashx` 就可以看到输出的信息“Hello World 我的服务器”, 如图 1-3 所示。

再设计一个 WPF 窗体的客户端程序, 在客户端程序的地址栏中输入网址 `http://localhost/web/server.ashx`, 单击“连接”按钮就可以看到服务器返回的信息“Hello World 我的服务器”, 如图 1-4 所示。



图 1-3 浏览器访问服务器



图 1-4 窗体程序访问服务器

1.1.2 技术要点

客户端访问服务器的核心类是 WebClient 类, 该类在 System.Net 空间中定义, 该类可以用于客户端与 Web 服务器端进行通信, 它有很多重要的方法, 其中一个方法是

```
public String DownloadString(String url)
```

这个方法的作用是通过 GET 方法访问地址为 url 的网站, 返回该网站的信息。例如:

```
WebClient client=new WebClient();  
client.Encoding=Encoding.UTF8;  
String s=client.DownloadString("http://localhost/web/server.ashx");
```

程序首先创建一个 WebClient 对象 client, 然后设置它的编码为因特网默认的 UTF8 编码, 最后调用 DownloadString 方法访问网站, 并返回网站的信息。

1.1.3 服务器程序

1. 创建网页

在 D 盘根目录建立 web 文件夹, 用 Visual Studio 打开网站 D:\web, 执行“网站”→“添加新项”命令, 出现“添加新项”对话框, 如图 1-5 所示。选择“一般处理程序”, 这种程序的扩展名是 ashx, 在“名称”中输入 server.ashx 后单击“添加”按钮, 在网站中就增加了

一个程序 server.ashx。



图 1-5 添加新项对话框

server.ashx 程序的代码如下：

```
<%@WebHandler Language="C#" Class="server" %>
using System;
using System.Web;
public class server : IHttpHandler
{
    public void ProcessRequest(HttpContext context)
    {
        context.Response.ContentType="text/plain";
        context.Response.Write("Hello World 我的服务器");
    }
    public bool IsReusable {
        get {
            return false;
        }
    }
}
```

其中 ProcessRequest 是主程序入口,context 对象是上下文对象,该对象包含了在网页设计中常见的 Request、Response 等对象,通过语句

```
context.Response.ContentType="text/plain";
```

设置要输出的文本的类型是纯文本,通过语句

```
context.Response.Write("Hello World 我的服务器");
```

输出“Hello World 我的服务器”。

2. 创建网站

上面建立的服务器程序 server.ashx 可以通过 Visual Studio 直接运行,但是程序调试好后要放在一个网站下独立运行,这要把 D:\web 目录设置为网站,操作步骤如下:

(1) 打开“控制面板”的“管理工具”,选择“Internet Information Services (IIS)管理器”,弹出对话框。

(2) 默认有一个 Default Web Site 的网站,选择该网站,并右击,弹出快捷菜单,选择“添加应用程序”命令,弹出“添加应用程序”对话框,如图 1-6 所示。在对话框中选择“物理路径”为 D:\web,别名可以设置为 web(也可以是其他名字),单击“确定”按钮后关闭对话框。

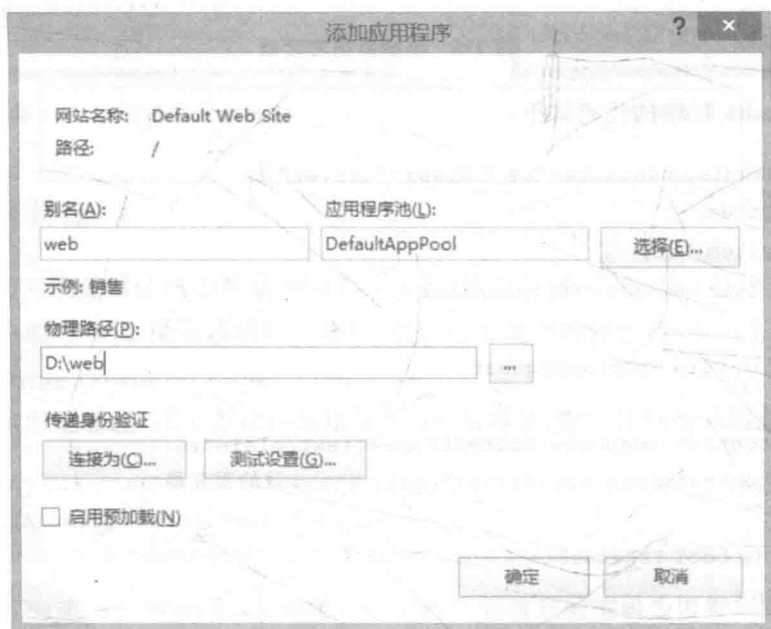


图 1-6 “添加应用程序”对话框

(3) 这时可以看到在 Default Web Site 下增加了一个名为 web 的项目,该项目就是默认网站下的一个站点,如图 1-7 所示。

在浏览器地址栏中输入网址 <http://localhost/web/server.ashx> 就可以看到图 1-3 所示的结果。在浏览器中查看网页源代码,可以看到服务器输出的只有一句“Hello World 我的服务器”的纯文本,没有别的 HTML 格式信息,如图 1-8 所示。



图 1-7 增加 web 站点



图 1-8 在浏览器中查看网页源代码

1.1.4 客户端程序

1. 界面设计

本项目把客户端程序设计成一个 WPF 程序,在窗体 MainWindow 中放入一个名为 txtUrl 的 TextBox 存放地址、一个名为 btCon 的 Button 连接按钮和一个名为 msg 显示标签 TextBlock,MainWindow.xaml 代码如下:

```
<Grid>
    <StackPanel Orientation="Vertical">
        <TextBox x:Name="txtUrl" Text="http://localhost/web/server.ashx" />
        <Button x:Name="btCon" Content="连接" Width="40" Click="btCon_Click" />
        <TextBlock x:Name="msg" />
    </StackPanel>
</Grid>
```

2. 程序设计

相应的 MainWindow.xaml.cs 程序如下:

```
namespace client
{
    public partial class MainWindow : Window
    {
        WebClient client;
        public MainWindow()
        {
            InitializeComponent();
        }
        private void btCon_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                WebClient client=new WebClient();
                client.Encoding=Encoding.UTF8;
                Uri uri=new Uri(txtUrl.Text, UriKind.Absolute);
                msg.Text=client.DownloadString(uri);
            }
            catch (Exception exp) { msg.Text=exp.Message; }
        }
    }
}
```

运行该程序,就得到图 1-4 所示的结果。

1.1.5 拓展训练

必须指出的是,WebClient 类的 DownloadString 方法在执行时会阻塞程序,等待服务器的响应,如果服务器长期没有响应,该方法就会一直等待下去,直到等待时间超出了最大的等待时间就抛出一个异常。这种等待过程称为同步过程,同步过程会阻塞程序,在网络程序中,除非能够确保服务器响应是很及时的,一般不建议采用同步过程,而是采用异步过程。

WebClient 提供了异步访问方法 DownloadStringAsync,声明如下:

```
public void DownloadStringAsync(String url)
```

该方法与同步方法 DownloadString 最大的不同就在于它不返回任何信息,在执行后不用等待服务器响应就立即返回,因此不会阻塞程序。该方法执行时只是确定了要访问指定的服务器,服务器什么时候响应是不可以预计的,它不负责等待服务器的返回。

但是一旦服务器响应,客户端又怎么样得到服务器响应的结果呢?实际上在异步方法执行前要先设置好一个回调函数,服务器响应后就触发该回调函数执行,获取服务器

响应结果。回调函数的设置如下：

```
void client_DownloadStringCompleted  
(object sender, DownloadStringCompletedEventArgs e)  
{  
}
```

其中 `client_DownloadStringCompleted` 是回调函数名称, 参数 `e` 的类型是 `DownloadStringCompletedEventArgs`。在服务器响应后就会触发该执行该函数, 并把参数 `e` 传递给该函数, 通过 `e.Result` 就得到服务器响应的信息。采用异步方法的程序如下:

```
private void btCon_Click(object sender, RoutedEventArgs e)  
{  
    try  
    {  
        WebClient client=new WebClient();  
        client.Encoding=Encoding.UTF8;  
        client.DownloadStringCompleted+=client_DownloadStringCompleted;  
        Uri uri=new Uri(txtUrl.Text, UriKind.Absolute);  
        client.DownloadStringAsync(uri);  
        msg.Text="等待服务器响应...";  
    }  
    catch(Exception exp) { msg.Text=exp.Message; }  
}  
  
void client_DownloadStringCompleted(object sender,  
DownloadStringCompletedEventArgs e)  
{  
    msg.Text=e.Result;  
}
```

运行时单击“连接”按钮可以立即看到标签上显示“等待服务器响应...”, 过一会服务器返回信息, 触发 `client_DownloadStringCompleted` 函数执行, 就看到服务器返回的信息“Hello World 我的服务器”。由此可见, 执行 `client.DownloadStringAsync(uri)` 时并不阻塞程序以等待服务器相应, 而是立即执行 `msg.Text = "等待服务器响应..."` 语句, 待服务器响应后就触发 `client_DownloadStringCompleted` 函数, 其中 `e.Result` 为服务器的返回信息。

一般为了监测服务器的错误, 回调函数可以写成如下形式:

```
void client_DownloadStringCompleted(object sender,  
DownloadStringCompletedEventArgs e)  
{  
    if(e.Eorr==null) msg.Text=e.Result;  
    else showMsg(e.Error.Message);  
}
```