

ASP. NET 在图书馆数字资源建设中的应用

全成千

(厦门大学图书馆, 厦门 361005)

摘要 ASP. NET 是一个提供各种网络服务的统一平台, 能较好的通过 Web 页完成数据操作、文件传送等多种功能。本文简要介绍 ASP. NET 实现数据库访问、文件上传、Email 发送及 Web 服务方式。

关键词 ASP. NET 数据库 Web 服务

ASP. NET in the Constructing of Library Digital Resources

Quan Chenggan

(Library of Xiamen University, Xiamen361005)

Abstract ASP. NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. This paper introduces the basic technology to query the database, upload files, send email and present Web Service.

Keywords ASP. NET Database Web services

从服务方式来说, 图书馆大量数字资源是通过其站点主页提供, 随着数据量的爆炸性增长和技术的不断发展, 图书馆主页为读者提供数据由原来的静态方式转向动态方式, 也就是由读者被动的接收方式转为读者主动获取的方式, 在这一转换过程中, 网页的动态发布技术起着相当重要的作用。

目前动态页面主要通过公共网关接口 (Common Gateway Interface, CGI)、Hypertext Preprocessor (PHP)、Java Server Pages (JSP) 及 Active Server Pages (ASP) 等技术实现, 在对于这些技术而言, 各有各的长处和不足, 综合考虑开发难度、维护成本及其功能, ASP 具有相当的优势。不少商业数据库查询和站点的发布都是采用 ASP 来实现。但是, ASP 运行效率较低, 功能上的局限对于图书馆数字资源的网上发布和查询有一定的限制, 2002 年, 美国微软公司正式推出 ASP.NET 技术, 在保留了 ASP 的易开发、维护等众多优点的基础上, 吸取 JSP、PHP 等技术的优点, 并增强功能, 提高运行效率, 较 ASP 等技术更适于实现图书馆动态页面和数据操作, 更好的实现与读者的互动和服务。

本文简要介绍 ASP.NET 及其在提供资源中的应用。

1 ASP.NET 与 ASP

运行 WINNT4 或 Windows 2000 或 WindowsXP 的服务器上安装 Microsoft. NET Framework 就支持 ASP. NET。ASP. NET 通过后缀名为 aspx 的文件实现, 它是一个统一的平台, 以供 VB. NET、JScript 及 C# 等多种语言开发各种网络应用程序, 其中 C# 是由微软公司新开发的下一代编程语言, 本文中的全部实例都以 C# 实现。开发 ASP.NET 程序可以用如同 NotePad 之类的普通文本编辑工具进行, 但是, 利用微软公司提供 Visual Studio. Net 集成开发工具更

为方便。

与 ASP 不同, ASP. NET 采用编译方式而不是解释方式来运行程序。ASP. NET 程序第一次运行时被系统编译后缓存, 之后的运行速度则会极大的提高。作者在同一台服务器上对 ASP 和 ASP. NET 进行循环性能测试, ASP. NET 只用少于 1 秒的时间完成了 ASP 要用 8 秒才能完成的 2000 × 2000 次循环 (循环体功能是获取当前系统时间)。美国微软公司也推荐将所有的 Web 页面文件后缀 (.htm/.html) 改为 ASP. NET 程序文件后缀 (.aspx) 以提高 Web 页面的浏览速度。

ASP. NET 程序可以象 ASP 程序一样嵌入 HTML, 例如对于下列 ASP 程序:

```
<html>
<body>
  <form>
    姓名: <input type= "text" name= "T1" value= <%
= Request.QueryString ( "T1" )%> >
    <input type= "submit" value= "提交" name=
"B1" >
  </form>
<p>
<%
if Request.QueryString ( "T1" ) <> "" then
Response.write (Request.QueryString ( "T1" ))
End If
%>
</body>
</html>
```

改写为 ASP.NET 程序为:

```

< %@ Page Language= "C#" %>
< html>
< body>
< form>
姓名: < input name= "T1" type= text value= "< %
= Request.QueryString [ "T1" ]%> " >
< input type= submit name= "Lookup" value=
"Lookup" >
< /form>
< p>
< %if (Request.QueryString [ "Lookup" ]!= null) { %
>
< %= Request.QueryString [ "T1" ]%>
< %} %>
< /body>
< /html>

```

但是, 微软公司推荐的更为先进的开发思想来开发 ASP.NET 程序, 对于上例, 则可以分写为如下两个文件 intro.aspx 与 intro.cs, 其中 intro.aspx 内容为:

```

< %@ Page Inherits = "MyCodeBehind" Src =
"Intro.cs" %>
< html>
< body>
< form action= "intro.aspx" method= "post" runat=
"server" >
Name: < asp: textbox id= "T1" runat= "server" /
>
< asp: button text= "Lookup" OnClick= "SubmitBtn_
Click" runat= "server" / >
< /form>
< asp: Label id= "Label1" runat= "server" / >
< /body>
< /html>

```

intro.cs 内容为:

```

using System.Web.UI;
using System.Web.UI.WebControls;
public class MyCodeBehind: Page {
public Label Label1;
public void SubmitBtn_Click (object sender, Sys-
tem.EventArgs e) {
Label1.Text = T1.Value;
}
}

```

把页面文件和代码文件分成两个文件及使用了面向对象的开发思想更利于代码的重用和日后的维护。

同时, ASP.NET 提供了大量的服务器端控件和可自定义控件, 通过这些控件, 开发人员能很方便的实现所需的各

2 ASP.NET 中的数据操作

数字图书馆建设工作中, 很大一部分工作是数据访问, 也就是对各种不同数据源进行数据操作。ASP.NET 通过 ADO.NET 进行数据访问, 并有功能强大的服务端控件对数据发布 (显示) 提供较完善的支持, 使得图书馆与读者之间的交互更为简洁和方便。与数据访问相关的类被封装在 System.Data、System.Data.SqlClient、System.Data.OleDb 等名字空间内。因此, 在使用 ADO.NET 中, 应用关键字 Import 引入相关的名字空间。

ADO.NET 支持对目前流行的各种数据源如大型数据库、桌面型数据库及 XML 文件等, 一般而言, ASP.NET 通过 ADO.NET 对数据源的访问过程包括如下步骤或是其中的若干个:

- (1) 创建一个数据连接;
- (2) 请求一个记录集合;
- (3) 把记录集合暂存到 DataSet;
- (4) 如果需要, 返回第 2 步; (DataSet 可以容纳多个数据集合)
- (5) 关闭数据库链路;
- (6) 在 DataSet 上作所需要的操作。

ADO.NET 中的 DataSet 在 ASP.NET 的地位相当于 ASP 中的 Recordset。除 DataSet 外, ASP.NET 还提供了 XMLReader、DataReader 等其它组件, 用户可以根据要求选择不同组件, 很方便的对各种数据源实现数据访问。

下面以 Sql Server 7 中的 pubs 库为例, 说明 ASP.NET 是如何利用 ADO.NET 进行数据更新和查询操作的。

```

< %@ Import Namespace= "System.Data" %>
< %@ Import Namespace= "System.Data.SqlClient"
%>
< script language= "C#" runat= "server" >
SqlConnection myConnection;
protected void Page_Load (Object Src, EventArgs E)
{
myConnection= new SqlConnection ("user id= sa; pass-
word= Password; initial catalog=
pubs; data source= serverIP; Connect Timeout= 30")
if (! IsPostBack)
{
BindGrid ();
}
}
public void MyDataGrid_Update (Object sender, Dataa-
GridCommandEventArgs E)
{
// 数据更新操作
String updateCmd= "UPDATE Authors SET au_ id=

```

```
@ Id, au_ lname= @ LName,
    au_ lfname= @ FName, phone= @ Phone, "+ "address
= @ Address, city= @ City,
    state = @ State, zip = @ Zip, contract = @ Contract
where au_ id= @ Id";
SqlCommand myCommand= new SqlCommand(update-
Cmd, myConnection);
myCommand.Parameters.Add ( new SqlParameter
( "@ Id ", SqlDbType.NVarChar, 11));
.....
myCommand.Connection.Open ();
try
{
myCommand.ExecuteNonQuery ();
}
catch ( SqlException e)
{
//这里进行异常处理
}
myCommand.Connection.Close ();
BindGrid ();
}
public void BindGrid ()
{
//数据选择和显示操作
SqlDataAdapter myCommand = new SqlDataAdapter
( "select * from Authors ", myConnection);
DataSet ds= new DataSet ();
myCommand.Fill ( ds, "Authors");
MyDataGrid.DataSource = ds.Tables [ "Authors " ]
.DefaultView;
MyDataGrid.DataBind ();
}
</script>
```

关于数据追加、数据删除其它所有相关的数据库操作，都能使用类似的方式通过 SqlCommand 的 ExecuteNonQuery () 等方法完成。对于查询得到的 DataSet，则可以绑定到诸如 DataGrid、DataList 等服务端控件上，通过对其相关属性及事件方法进行设定及编码，就能很方便的实现数据排序、分页显示等原来使用 ASP 较难实现的功能。

3 文件上传、Email 发送功能及 Web 服务

除了数据访问较 ASP 的功能更强大和方便，ASP.NET 还具有了不少适于图书馆与读者进行交互的新特性，如用 ASP、PHP 或是 JSP 必须借用第三方组件才能实现或无法实现的文件上传、Email 发送及 Web 服务，ASP.NET 只要

用简单的方式就能完成。

(1) 文件上传:

```
< script language= "C#" runat = "server ">
void Button 1_ Click (object Source, EventArgs) {
if ( ( FileName.Value != "" ) and ( File.PostedFile!
= null) )
{
try
{
File.PostedFile.SaveAs ( "c: \\ temp \\ " + File-
Name.Value);
}
catch (Exception exc)
{
// 异常处理
}
}
}
</script>
```

(2) 发送 Email:

```
< %@ Import Namespace= "System.Web.Mail "%>
< script Language= "C#" runat = "server ">
MailMessage mail= new MailMessage ();
mail.From= 发信人地址;
mail.To= 收件人地址;
mail.Body= "这是一个测试的信件内容 ";
mail.Subject= "信件标题 ";
SmtpMail.SmtpServer= smtp 服务器;
SmtpMail.Send ( mail);
</script>
```

(3) Web 服务:

Web 服务通过 .asmx 文件生成符合 WSDL 的 xml 文件提供各种功能。对于客户而言，只要通过 WSDL 代理类文件而并不需要知道所使用数据库的名称、用户名及密码等重要信息就能实现服务器提供的功能。在下面的例子中，作者提供一个对数据操作的 Web 服务文件。

```
< %@WebService Language= "C#" Class= "DataSer-
vice "%>
using System;
using System.Data;
using System.Data.SqlClient;
using System.Web.Services;
public class DataService {
[WebMethod]
public DataSet GetTitleAuthors ()
```

```
{
    SqlConnection myConnection = new SqlConnection
( user
```

```

id= sa; password= Password; initial catalog= pubs; data
source= serverIP; Connect Timeout= 30");
    SqlDataAdapter myCommand1= new SqlDataAdapter
( "select * from Authors ",
    myConnection);
    SqlDataAdapter myCommand2= new SqlDataAdapter
( "select * from Titles ",
    myConnection);
    DataSet ds= new DataSet ();
    myCommand1.Fill (ds, "Authors ");
    myCommand2.Fill (ds, "Titles ");
    return ds;
}
[WebMethod]
public int PutTitleAuthors (DataSet ds)
{
    return ds.Tables [0] .Rows.Count;
}
}

```

Web 服务被正确提供后, 客户使用 WSDL.EXE 命令生成代理类文件, 然后通过代理类文件就能进行被授权的数据操作。例如通过下面的代码, 客户就可以获得所要数据。

```

< % @ Import Namespace= "DataService "%>
< % @ Import Namespace= "System.Data "%>
< % @ Import Namespace= "System.Data.SqlClient "%
>
< html>
< script language= "C# " runat= "server " >
protected void Page_ Load ( Object Src, EventArgs E)
{

```

```

    DataService d= new DataService ();
    DataSet myData= d.GetTitle Authors ();
    if (! Page.IsPostBack)
    {
        Authors_ DataGrid.DataSource = myData.Tables
[ "Authors "].DefaultView;
        Authors_ DataGrid.DataBind ();
    }
}
< /script>

```

Web 服务是一项崭新的技术, 在实现图书馆数字化建设中的跨库查询、分布式查询等工作可能有较大前景。

从上述对 ASP.NET 的简单介绍中可以看出 ASP.NET 比 ASP、PHP 及 JSP 等较流行的动态页面制作技术功能更强大, 功能涉及 Web 应用程序的各个方面。在集成开发环境 VS.NET 的支持下, 开发和维护工作也很方便, 很适于数字化图书馆的建设中的数据查询和动态发布。

参考文献

- 1 陈茜, 张伟. ASP 技术在图书馆网络资源建设中的应用. 情报科学, 2001 (7)
- 2 周艳, 时晓难. 利用 ASP 技术实现图书馆主页上 Web 数据库的动态查询. 现代图书情报技术, 2001 (4)
- 3 濮德敏. 高校图书馆主页与 ASP 技术. 大学图书馆学报, 2000 (3)
- 4 曹军. ASP、PHP 和 JSP 的比较研究. 现代图书情报技术, 2001 (4)
- 5 Microsoft Co. ASP.NET Quick Start
http: www.gotdotnet.com/, 2001

(责任编辑: 滕代娣)