

WEB 数据库接口技术及应用

李 平 刘 彬 (北京联合大学信息学院 100101)

摘要: 本文讲述了企业网站的建设中 ASP 文件的处理过程、ASP 的内置对象及其应用。并通过程序设计实例重点分析了 WEB 数据库接口 ADO (Active Data Objects) 的主要对象, 及如何通过 ADO 从数据库中获取数据。最后, 结合程序讲解了应用四种游标执行 SQL 查询结果的传送方法和 ADO 的上锁类型。

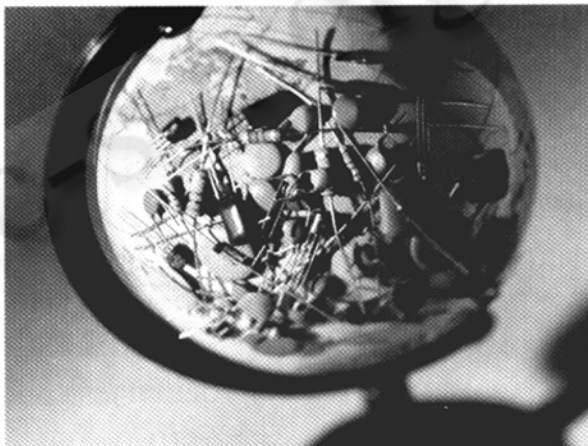
关键词: ASP 技术 WEB 数据库接口 游标 ASP 脚本 内置对象 上锁

对于企业网站, 其主要功能就是要宣传自己的产品。对于传统网站的建设来说, 由于其 Web 是 HTML 的格式, 因此在网站的结构设计完毕之后, 要分别为每一个产品制作其简介的页面。这样做对于网站建设初期有一定的好处, 例如网站建设速度快、方法简单等。但对于网站在后期的维护却是非常困难的。

当企业产品不断的增加或更新, 随之而来的是要对相应产品的页面进行添加或修改。而对于 ASP 技术来说, 便不存在这样的问题。由于 ASP 可以通过数据库访问对象轻松地与数据库相连接, 系统可以将用户所要查询的产品信息从数据库中提出, 再利用 ASP 动态的特性, 动态产生一个页面来介绍产品。这样对网站建设来说, 我们只须制作一页便可以介绍企业的所有产品。对于维护人员, 只须通过使用已做好专用页面便可对 Web 数据库进行维护和管理。在此基础上, 企业还可拥有自己的注册用户; 可以为不同的访问者分配权限; 实现网上订货、在线接收订单等功能。

此外 ASP 技术可以使企业的网站具有更加丰富的内容。我们可以根据日期的不同让用户看到不同的网页, 让你的网页每一天都是新的。还可以在网站上创建留言板等程序, 可以让用户与企业、用户与用户进行更进一步的交流。

用 ASP 代替 HTML 来建设网络, 首先要对网络结构进行整体规划, 然后要建造相关数据库, 最后制作相



关的 ASP 网页程序。这样 ASP 网站的前期工作量很大, 相对于简单的 HTML 结构的网站来说制作的周期比较长, ASP 网站的结构一旦设计完毕在以后的维护中一般就不会改变了。对于企业的投资, 投入是一次性的, 网站建设完毕后不会因增加产品而增加网页的制作成本。

1 利用 ASP 技术建设网站

1.1 ASP 的处理过程

那么 ASP 文件是怎样处理的呢? 首先用户从服务器请求某 .asp 页。Web 服务器检查文件的扩展名, 查看是不是特殊的程序处理请求。如果是 .asp 扩展名, Web 服务器就确定它应当调用 ASP 来处理该页。如果该页以前从未被请求过, 或上一次请求过之后更改过, 就必须重新分析, 让 Web 服务器对其进行语法检查和编译。在分析的过程中, HTML 和脚本编程代码被分开。脚本编程引擎来自 IIS 的资源来执行代码。IIS 也负责外部的 ActiveX 对象处理输入和输出。下一步是将 ASP 文件中的脚本输出和静态 HTML 代码进行合并。最后将最终的 HTML 在 HTTP 响应中返回给用户。

1.2 ASP 的内置对象及其应用

ASP 有 6 个内置的对象, 它们分别是 Application, Session, Request, Response, Server 和 Object Context。它们分别有自己的属性、集合和方法, 来实现不同的功能。下面通过实例讲述如何使用这些内置对象。

当我们建设的网站需要完成在线购物或在线接收订单时,这就需要确定用户的身份。要解决这样的问题往往是采用用户先注册后登录的方式来确认用户身份。

在建立用户的注册和登录时我们会遇到一些问题:第一,我们是用什么方法来获得用户所填写的表单;第二,是如何为登录成功的用户分配权限;第三,是数据的有效性;

在解决用什么方法来获得用户所填写的表单时有两个选择:一是采用表单的GET方法,二是POST方法。这两种方法有着根本的区别,GET方法是将表单中的数据追加到查询字符串的URL中,然后通过使用Request.QueryString集合来读取数据。而POST方法是将表单的数据直接发送给服务器,然后用Request.From集合来读取数据,结果是没有字符串附加到URL中,并且字段的数目和大小几乎没有限制。因此在一般情况下,表单的发送是采用POST方法。此外POST方法的另一个优点是可以从客户的硬盘向服务器传输文件。

在用户的使用权限方面,我们要充分发挥ASP的Session对象。由于Session对象是用来维护每个用户的个人的信息。也就是说,当用户访问网站主页的时候系统便会为其创建Session对象。Session对象是针对每一个单独的个体,因此它不会受其他访问者的影响。所以我们可以为每一个登录成功的用户创建一个Session对象,并且用IsEmpty()函数来检查该Session对象是否存在,如果存在则证明用户登录成功,否则便是登录失败。以此类推,我们可以建立不同的Session对象以用来区分不同的权限。关于数据的有效性。首先我们要确定哪些数据是要用户必须填写的,而哪些是可写可不写的。然后是确定所需要数据的类型,这一点是非常重要的。数据的类型应与Web数据库中相应的类型是一致的,否则在向数据库写入数据的时候将会出现错误。在数据类型不正确的时候,我们可以采用不同的转换函数来转换数据。但这样做并不好,因为有些数据即使被转换成为有效的数据类型,而数据本身却并不是我们想要的。所以最好的解决办法是提示用户输入正确的数据。此外,我们还需要过滤掉一些意想不到的错误,例如HTML中的<SCRIPT>标记,以防止一些人在此标记中输入恶意的程序来破坏系统。

在收集用户的注册信息时,我们还应该记录一些不需要用户填写的数据,如用户注册时的时期、时间、IP地址等。甚至还可收集一些用户的喜好,为用户创建个性化界面和服务。

此外Session对象还有一些其他的功能,它可以避免一些想不到的问题,例如:当我们知道某一个文件在Web服务器中的具体位置,在IE的地址栏中输入该文件的绝对路径,如果该页没有进行保护就可以将该页打开。我们可以通过Session对象和IsEmpty()函数联合使用利用分配权限道理,来对该页进行这方面的保护。Application对象可以用来存储、读取用户共享的应用程序信息。Session对象还可以和Application对象联用,用来记录网站的总共访问人数、每一页的访问次数和实时的在线人数。如果为每一个访问者分配权限用其作为标记,并为每一个页面做一标记,我们可以实时地知道每一个在线访问者在网站的位置。对于企业来说其最关心的功能应是在线的注册和登录以及在线的订货。而这无一需要对数据库进行读取、写入和查询的操作。

2 Web 数据库接口

2.1 ADO 简介

ActiveX Data Objects(ADO, ActiveX 数据对象)是一个提供了数据库和ASP之间的接口。可以在ASP中使用能通过OLE DB和ODBC访问的数据库。在目前还有一些数据库访问技术,ADO虽然在某些方面并不是最好的,但它却包含了一些技术的精华,它为数据管理程序(如Microsoft SQL Server或Microsoft Access)提供了一个应用程序水平的界面,用户可以在ASP中直接用ADO与数据库通信。而且ADO是获得对任何数据库访问的最舒适的方式,它用较小的内存负载和磁盘记录提供了高速访问。当位于用于大量用户、同时访问Web服务器上时,这些优点显得更加重要了。

因为ADO是与编程语言无关的,所以可以从VBScript、Jscript、Visual Basic、Java和其他任何能处理对象的语言中访问它。

ADO提供了许多对象,可帮助我们连接到数据库并操作其数据。下面是ADO的主要对象:

- * Connection: 代表到数据库源的链接。
- * Recordset: 由从数据库查询返回的记录和这些记录中的一个游标组成。
- * Field: 包含单栏数据以及关于这个数据的信息。
- * Property: 包含底层提供者提供的关于ADO对象的动态信息。
- * Error: 包含从提供者返回的扩展错误信息。
- * Command: 提供定义特定命令的能力,你可以用

不同的参数多次对数据库执行自己定义的命令。

* Parameter : 可以是与 Command 对象相关联的单个参数或形参。

2.2 通过 ADO 从数据库中获取数据

在 ADO 的主要对象中, 使用最为频繁的就是 Connection 对象和 Recordset 对象。因为我们要想访问数据库, 就必须创建一个与数据库源的连接。当对数据库进行查询时, 其结果的存放位置便是我们最关心的。

可以通过对 Connection 对象执行 Execute 方法来创建 Recordset 对象。相反, Recordset 对象的 Active Connection 属性指示了 Recordset 当前所属的连接, Recordset 对象具有一个 Fields 集合, 它包含了 Recordset 的所有 Field 对象。对于数据库的访问、查询及一些基本的操作, 我们可以创建相应的 SQL 语句, 然后用 Connection 对象的 execute 方法来执行以实现以上操作。虽然 ADO 为我们提供了一些对数据库操作的方法, 使用起来也非常简单, 但由于 SQL 语句在执行起来具有更快的速度和更高的性能, 因此在这里还是建议使用 SQL。由于对于不同的数据库 SQL 会在某方面有些差异, 所以在编写时要特别注意。

在创建 SQL 语句的时候, 应特别注意对于单引号 (') 的处理, 由于在 SQL 中使用单引号来分割字段, 所以当单引号作为字符出现时则用两个单引号来表示 (""), 因此当其作为数据时我们需要用 REPLACE () 函数来把 "" 变为 ""。

下面的程序以 Windows NT 为平台是用于建立一个与 db1.mdb 的连接, 并且从 typeName 和 typeID 字段中获取数据, 然后创建一个下拉菜单, 最后动态生成菜单项:

```
<%  
strconn="Driver={Microsoft Access Driver (*.mdb)};  
DBQ=" & Server.MapPath("\iisadmin") & "\website\  
db1.mdb;" // 创建连接字符串 //  
  
set objconn=server.createobject("adodb.connection") //  
创建一个名字为 objconn 的 Connection 对象 //  
objconn.open strconn // 打开与数据库的连接 //  
%>  
  
<select name="select"> //HTML <SELECT> 标记, 用  
来创建一个下拉菜单 //  
  
<option value="0" selected> 请选择 </option>  
  
<%  
  
strq="select typeID,typeName from type" // 创建查询  
SQL 语句 //
```

```
set objrs=objconn.execute(strq) // 用 Connection 的  
execute 方法创建 Recordset 对象 //
```

```
while not objrs.eof // 如果当前游标位置不是记录集  
的底, 则进行循环 //
```

```
typeName=objrs("typeName")
```

```
typeID=objrs("typeID") // 将 typeName 和 typeID 字段  
的值, 分别赋得不同的变量 //
```

```
%>
```

```
<option value="<%=typeID%>"><%=typename%></  
option> // 创建动态的下拉菜单选项 //
```

```
<%
```

```
objrs.movenext // 游标向前移动 //
```

```
wend
```

```
%>
```

```
</select>
```

从以上程序可以看到, 通过 ADO 可以很轻松的获取数据。值得注意的是, 在获取数据时, 一定要控制游标的移动, 否则就很可能进入死循环。还有在我们不需要与数据库连接时, 应即时的断开连接。

2.3 游标和上锁类型

现在我们可以轻松的得到数据库中的数据, 但现在存在一个问题, 就是从数据库中查找的结果是通过什么传送给 Web 的呢?

我们现在可以看到, 当用 Connection 对象执行了 SQL 的查询语句后, 其结果将传送给 Recordset 对象, 然后需要通过游标来读取每一条记录并传送给 Web。

在前面的程序中, 提到了游标的概念。在我们打开 Recordset 时, 不论是无意还是有意的, 都会带有某个特定的游标和上锁类型。那么由于两方面原因, 先来理解一下游标和上锁类型。

首先, 使用不同的游标和上锁类型对 ASP 脚本的运行有着戏剧性的影响。若选择错误, 则 ASP 可能无法获取并显示数据库清单。第二, 只有使用正确的游标和上锁类型时, 才可以利用 Recordset 对象的某些属性。

正确选择 Recordset 的游标和锁类型应平衡考虑两个方面: 运行性能的需要和特定 Recordset 属性的需要。

有些程序在编写时其算法和语法均无问题, 但在调试时却总是出错, 无法达到想实现的功能, 纠其根源往往是没有选择好正确的游标和上锁类型。在执行返回一系列数据行的 SQL 查询时, 数据行通过游标传送到 ADO 应用程序。有 4 种类型的游标确定查询结果的传送方法:

* 前向游标: 此游标的缺省游标。当打开 Recordset 时没有指定游标类型的时候, 则 Recordset 由前向游标打开。用它打开 Recordset, 只能按从头到尾顺序取得结果, 它不支持滚动, 只能在结果间单向移动, 但其提供了最快的运行性能。

* 静态游标: 反映第一次打开游标时表中数据的状态, 游标无法查明底层表中的数据行是否更新、删除或添加了新数据, 不过与只能前移的游标不同, 静态游标可以在结果间前后滚动。

* 键盘驱动游标: 此游标支持滚动, 可以查询表中底层数据行的某些变化, 但不是全部。它特别是可以准确反映数据是否更新过。但它不能查明其他用户是否曾插入或删除过数据行。

* 动态游标: 动态游标是最丰富的游标类型。游标打开时可以查询其他用户对表的任何改动, 而且支持滚动。我们在编写程序时, 一般应该以性能优先为准则来选择游标。也就是说, 在不需要特殊功能的时候应用前向游标来打开 Recordset 对象。

此外, 还有一种处理 ADO 游标的选择, 那就是选择打开客户机端或服务端游标。在使用客户机端游标时, 在 ASP 的主机上打开并管理游标。而使用服务端游标时, 在数据库的主机上打开并管理游标。使用客户机端游标的好处在于它将数据库服务器上的工作转移。所有查询结果都送到 ASP 应用程序的主机。它还可以提高 ASP 应用程序的性能。因为游标与应用程序位于同一台机器上, 可以在数据行之间迅速移动游标。在此, 如果你担心数据库服务器的负担过重, 则应使用客户机端游标。若网络服务器的负担过重或网络太拥挤, 则应使用缺省的服务器端游标。

下面的程序可以看到如何设置游标:

```
<!--#include virtual="/adovbs.inc"-->
<%
set objconn=server.createobject("adodb.connection")
strconn="Driver={Microsoft Access Driver (*.mdb)};
DBQ=" & Server.MapPath("\ iisadmin")
& "\ website \ db1.mdb;"
objconn.open strconn
strq="select typeID,typeName from type"
set objrs=server.createobject("adodb.recordset") //创建
```

Recordset 对象 //

```
objrs.cursortype=adOpenStatic //将游标设置为静态游标//
```

```
objrs.cursorlocation=aduseclient //将游标设置为客户端游标//
```

```
objrs.open strq,objconn //打开 Recordset 对象//
%>
```

下面来看一下 ADO 的上锁类型。在数据库驱动的网站可能要同时支持上面上百个访问同一个数据库的用户。则现在存在一个问题, 那就是当数个用户同时在数据库表中更新、读取数据, 则会出现冲突, 而使系统不能正常工作。为了解决此类冲突我们引进了上锁类型的概念, 可以使用 Recordset 的 LockType 属性。LockType 属性可以拥有不同的值: adLockReadOnly, adLockPessimistic, adLockOptimistic, adLockBatchOptimistic。它们分别代表以不同的方式打开 Recordset。

若 ASP 仅仅用来显示来自数据库表的数据, 则不应改变缺省的上锁类型 (adLockReadOnly)。当要用 ADO 的原始方法 (如 AddNew) 更新数据 (非 SQL 方法), 则要选择另一种上锁类型。

3 优化 ASP 脚本

为了使 ASP 脚本运行的更快, 可采取以下措施:

- (1) 尽可能的使用局部变量。
- (2) 避免调整动态数组的大小。
- (3) 打开 Option Explicit。
- (4) 将集合值复制到局部变量中去。
- (5) 关闭不使用的 Session 对象。
- (6) 使用浏览器的高速缓存。■

参考文献

- 1 《Active Server Pages 宝典》[美] Eric A. Smith 著, 电子工业出版社 1999。
- 2 《Active Server Pages 应用大全——ASP 与数据库的整合》廖信彦著, 清华大学出版社 2000.4。
- 3 《Web 数据库开发进阶》唐东 著, 北京人民邮电出版社 1999。
- 4 《Active Server Pages & Web 数据库》王国荣 著, 北京人民邮电出版社 1999。