AJAX 数据库 ASP 总结

ASP 快速参考

来自菜鸟教程的 ASP 快速参考。打印出来，放入口袋，以备随时使用。

基础语法

ASP 脚本由 <% 和 %> 包围。向浏览器写输出：

```
<html>
<body>
<% response.write("Hello World!") %>
</body>
</html>
```

ASP 中的默认语言是 VBScript。如需使用其他脚本语言，请在 ASP 页面顶部插入一段语言说明：

```
<%@ language="javascript" %>
<html>
<body>
<%
....
%>
```

表单和用户输入

Request.QueryString 用于收集 method="get" 的表单中的值。使用 GET 方法从表单传送的信息对所有的用户都是可见的（出现在浏览器的地址栏），并且对所发送信息的量也有限制。

Request.Form 用于收集使用 method="post" 的表单中的值。使用 POST 方法从表单传送的信息对用户是不可见的，并且对所发送信息的量没有限制。

ASP Cookies

cookie 常用用于识别用户。cookie 是一种服务器留在用户计算机上的小文件。每当同一台计算机通过浏览器请求页面时，这台计算机将会发送 cookie。

Response.Cookies 命令用于创建 cookie：

```
<%
Response.Cookies("firstname")="Alex"
Response.Cookies("firstname").Expires="May 10,2002"
%>
```

注释：Response.Cookies 命令必须出现在 <html> 标签之前！

"Request.Cookies" 命令用于取回 cookie 值：

```
<%
fname=Request.Cookies("firstname")
response.write("Firstname=" & fname)
%>
```

引用文件

通过使用 #include 指令，您可以在服务器执行 ASP 文件之前，把另一个 ASP 文件的内容插入到这个 ASP 文件中。#include 指令用于创建函数、页眉、页脚或者其他多个页面上需要重复使用的元素等。

语法：

```
<!--#include virtual="somefile.inc"-->
```

或者

```
<!--#include file ="somefile.inc"-->
```

请使用关键词 virtual 来指示以虚拟目录开始的路径。如果一个名为 "header.inc" 的文件位于虚拟目录 /html 中，下面这行代码会插入 "header.inc" 文件中的内容：

```
<!-- #include virtual ="/html/header.inc" -->
```

请使用关键词 file 来指示一个相对路径。相对路径是以含有引用文件的目录开始的。如果您在 html 目录中有一个文件，且 "header.inc" 文件位于 html 头部，下面这行代码将在您的文件中插入 "header.inc" 文件中的内容：

```
<!-- #include file ="headersheader.inc" -->
```

请使用带有语法 (..) 的关键词 file 来引用更高层级目录中的文件。

Global.asa

Global.asa 文件是一个可选的文件，它可包含被 ASP 应用程序中每个页面访问的对象、变量和方法的声明。

注释：Global.asa 文件必须存放在 ASP 应用程序的根目录中，而且每个应用程序只能有一个 Global.asa 文件。

Global.asa 文件只能包含下列内容：

- Application 事件
- Session 事件
- <object> 声明
- TypeLibrary 声明
- #include 指令

Application 和 Session 事件

在 Global.asa 中，您可以告诉 application 和 session 对象当 application/session 开始时做什么，当 application/session 结束时做什么。完成这项任务的代码被放置在事件句柄中。**注释：**由于我们无法在 Global.asa 文件中使用 ASP 的脚本分隔符 (<% 和 %>) 插入脚本，我们需要把子例程放置在 HTML 的 <script> 标签内部：

```
<script language="vbscript" runat="server">
sub Application_OnStart
' some code
```

```
end sub
sub Application_OnEnd
' some code
end sub
sub Session_OnStart
' some code
end sub
sub Session_OnEnd
' some code
end sub
</script>
```

<object> 声明

可通过使用 <object> 标签在 Global.asa 文件中创建带有 session 或者 application 作用域的对象。**注释：**<object> 标签应位于 <script> 标签外部！

语法：

```
<object runat="server" scope="scope" id="id"
{progid="progID"|classid="classID"}>
.....
</object>
```

TypeLibrary 声明

TypeLibrary（类型库）是一个容器，其中装有对应于 COM 对象的 DLL 文件。通过在 Global.asa 文件中包含对 TypeLibrary 的调用，可以访问 COM 对象的常量，同时 ASP 代码也能更好地报告错误。如果您的 Web 应用程序依赖于已在类型库中声明的数据类型的 COM 对象，您可以在 Global.asa 中对类型库进行声明。

语法：

```
<!--METADATA TYPE="TypeLib"
file="filename"
uuid="typelibraryuuid"
version="versionnumber"
lcid="localeid"
-->
```

Session 对象

Session 对象用于存储关于用户会话（session）的信息，或者更改用户会话（session）设置。存储于 Session 对象中的变量存储单一用户的信息，并且对于应用程序中的所有页面都是可用的。

集合

- Contents - 包含所有通过脚本命令追加到 session 的条目。
- StaticObjects - 包含了所有使用 HTML 的 <object> 标签追加到 session 的对象。

- Contents.Remove(*item/index*) - 从 Contents 集合删除一个项目。
- Contents.RemoveAll() - 从 Contents 集合删除全部项目。

属性

- CodePage - 规定显示动态内容时使用的字符集。
- LCID - 设置用于显示动态内容的区域标识符。
- SessionID - 返回 session id
- Timeout - 设置或返回 session 的超时时间。

方法

- Abandon - 撤销 session 对象中的所有对象。

Application 对象

在一起协同工作以完成某项任务的一组 ASP 文件称为一个应用程序。Application 对象用于把这些文件捆绑在一起。所有的用户分享一个 Application 对象。Application 对象存有会被应用程序中的许多页面使用的信息（比如数据库连接信息）。

集合

- Contents - 包含所有通过脚本命令追加到应用程序中的项目。
- StaticObjects - 包含所有使用 HTML 的 <object> 标签追加到应用程序中的对象。
- Contents.Remove - 从 Contents 集合中删除一个项目。
- Contents.RemoveAll - 从 Contents 集合中删除所有的项目。

方法

- Lock - 防止用户修改 Application 对象中的属性。
- Unlock - 允许用户修改 Application 对象中的属性。

Response 对象

Response 对象用于从服务器向用户发送输出的结果。

集合

- Cookies(name) - 设置 cookie 的值。如果 cookie 不存在，则创建 cookie ，并设置指定的值。

属性

- Buffer - 规定是否缓冲输出。当输出设置缓冲时，服务器会阻止向浏览器的响应，直到所有的服务器脚本均被处理，或者直到脚本调用了 Flush 或 End 方法。如果要设置此属性，它应当位于 ASP 文件中的 <html> 标签之前。
- CacheControl - 设置代理服务器是否可以缓存由 ASP 产生的输出。如果设置为 Public，则代理服务器会缓存页面。
- Charset(*charset_name)* - 将字符集的名称（比如 "ISO8859-1"）追加到 Response 对象中的内容类型报头。

- ContentType - 设置 Response 对象的 HTTP 内容类型（比如 "text/html", "image/gif", "image/jpeg", "text/plain"）。默认是 "text/html"。
- Expires - 设置页面在失效前的浏览器缓存时间（分钟）。
- ExpiresAbsolute - 设置浏览器上页面缓存失效的日期和时间。
- IsClientConnected - 指示客户端是否已从服务器断开。
- Pics(*pics_label*) - 向 response 报头的 PICS 标签追加值。
- Status - 规定由服务器返回的状态行的值。

方法

- AddHeader(*name, value*) - 向 HTTP 响应添加新的 HTTP 报头和值。
- AppendToLog *string* - 向服务器记录项目（server log entry）的末端添加字符串。
- BinaryWrite(*data_to_write*) - 在没有任何字符转换的情况下直接向输出写数据。
- Clear - 清除已缓冲的输出。使用该方法来处理错误。如果 Response.Buffer 未设置为 true，该方法将产生 run-time 错误。
- End - 停止处理脚本，并返回当前的结果。
- Flush - 立即发送已缓冲的输出。如果 Response.Buffer 未设置为 true，该方法将产生 run-time 错误。
- Redirect(*url*) - 把用户重定向到另一个 URL。
- Write(*data_to_write*) - 向用户写文本。

Request 对象

当浏览器向服务器请求页面时，这个行为就被称为一个 request（请求）。Request 对象用于从用户那里获取信息。

集合

- ClientCertificate - 包含了存储在客户证书中的所有的字段值。
- Cookies(*name*) - 包含了 HTTP 请求中发送的所有的 cookie 值。
- Form(*element_name*) - 包含了使用 post 方法由表单发送的所有的表单（输入）值。
- QueryString(*variable_name*) - 包含了 HTTP 查询字符串中所有的变量值。
- ServerVariables(*server_variable*) - 包含了所有的服务器变量值。

属性

- TotalBytes - 返回在请求正文中客户端发送的字节总数。

方法

- BinaryRead - 取回作为 post 请求的一部分而从客户端发送至服务器的数据。

Server 对象

Server 对象用于访问服务器上的属性和方法。

属性

- ScriptTimeout - 设置或返回在一段脚本终止前它所能运行时间（秒）的最大值。

方法

- CreateObject(*type_of_object*) - 创建对象的实例。
- Execute(*path*) - 从 ASP 文件内部执行另一个 ASP 文件。在被调用的 ASP 文件执行完毕后，控制权返回原先的 ASP 文件。
- GetLastError() - 返回可描述已发生错误状态的 ASPError 对象。
- HTMLEncode(*string*) - 对字符串应用 HTML 编码。
- MapPath(*path*) - 把相对或虚拟路径映射为物理路径。
- Transfer(*path*) - 把所有状态信息发送到另一个文件以备处理。在传送之后，程序的控制权不会返回原先的 ASP 文件。
- URLEncode(*string*) - 对字符串应用 URL 编码规则。

← AJAX 数据库 ASP 总结 →