**ASP 3.0高级编程（三）(1)**

　　访问和更新kies集合

　　kies的值比asp其他集合（例如fr和servervariables）的值要复杂得多。kie是一小块由浏览器存贮在客户端系统上的文本，且随同每次请求发往它们应用于的域中的服务器。

　　asp使得应用kie较为容易，可以从request对象的kies集合中获得所有随同请求发出的kie值，并可创建或修改kie，通过respnse对象的kies集合发回给用户。

　　kie包含可用两种方式构造的信息，单值kie提供其值给代码是通过一个一般的类asp集合。然而，集合的每个成员可能本身也是一个集合，包含这种信息的kie通过称为多值（ultiple-value）kie。

　　创建一个单值的kie较为简单，如下所示：

　　respnse.kies(“ite-nae”)=“ite-value”

　　创建一个多值的kie，可以使用如下命令：

　　respnse.kies(“ite-nae”)(“sub-ite-nae”)=“sub-ite-value”

　　设置kie应用的域及路径及其有效期，我们使用：

　　respnse.kies(“ite-nae”).dain=“dain-url”

　　respnse.kies(“ite-nae”).path=“virtual-path”

　　respnse.kies(“ite-nae”).expires=#date#

　　通常，客户只在对创建kie的目录中的页面提出请求时，才将kie随请示发住服务器。通过指定path属性，可以指定站点中何处这个kie是合法的，并且这个kie将随请求发送。如果kie随对整个站点的页面请求发送，设置path为“/”。

　　假如expires属性没有设置，关闭当前的浏览器实例时，kie将被自动消除。

　　注意，我们在向浏览器发送任何输出时，已经创建了kie。因为，这些kie是页面http报头的一部分。

　　在asp3.0中，缓冲的缺省状态是打开的，且没有输出被发送，除非使用respnse.flush指定做这个工作或者页面已到末端。这意味着创建kie的代码可以在页面上的任何位置，直到任何输出“刷新”（flush）到客户端前，它都可以被执行。

　　要读现有的kie，使用request.kies集合。可以单独访问其中的项目，方法类似于创建它们时使用的方法。

　　strsinglevalue=request.kies(“ite-nae”)

　　strsubitevalue=request.kies(“ite-nae”)(“sub-ite-nae”)

　　注意request.kies集合（和所有其他request集合一样）是只读的。respnse.kies集合是只写的，事实上可以访问这个集合中一系列kie的名称，而不是它们的值。

　　遍历kies集合

　　为了使用kies集合更加方便，可使用名称为haskeys的附加属性。假如访问的kie本身也是个集合，即它是一个多值的kie，这将返回true。使用haskeys属性，可以遍历完整的request.kies集合，从而获得所有kie的列表及它们的值。

　　freahbjiteinrequest.kies

　　ifrequest.kies(bjite).haskeythen

　　‘useantherfreahtiterateallsubkeys

　　freahbjitekeyinrequest.kies(bjite)

　　respnse.ritebjite“(““)=“\_

　　(bjite)(bjitekey)“

　　”

　　next

　　else

　　‘printutthekiestringasnral

　　respnse.ritebjite“=”(bjite)“

　　”

　　endif

　　next

　　这非常类似于前面的从request.fr集合中提取多个值的复杂代码。但是这里可以使用haskeys属性来判别每个条目是否为一个集合。而在fr例子里，必须查询request.fr(ite\_nae).unt属性，这是因为fr集合（和所有的除kie外的其他集合）成员不可能是真正的集合。asp只是做了“幕后”的工作，得到了每个多条目集合的值。

　　fr和querystring的差异

　　了解了访问各种asp集合的技术以后，需要解决另一个问题是：fr和querystring集合之间的差异是什么？假如准备使用asp，毫无疑问应该清楚这种差异，但需要参考http工作方式来重新认识，理解它们。

　　通过http从eb服务器请求页面或其他资源，有两个通用的方法。可使用get方法直接获得资源，也可使用pst把值传给相应资源。get方法是缺省的，可以看一下本章前面的一个http请求的实例：#p#分页标题#e#

　　7/8/9910:27:16sentget/stre/dnlad.asphttp/1.1

　　假如把一个或多个成对的名称/值附在请求页面的url后，就变成请求的查询字符串，且在querystring集合中提供给asp页面。单击eb页面、eail消息或其它文档的超链接，或在浏览器的地址栏中输入地址并按回车，或单击浏览器中的links或favrites按钮，所有这些都要使用get方法。

　　因此，对这些动作中传递值给asp的唯一方法是通过querystring集合，把值附在url后。

　　出现在request.querystring集合中并被访问的值，与前面看到的fr集合实例中的工作方式相同。url和查询字符串的结合：

　　http://ysite./press\_page.asp?firstnae=prisilla=desartes

　　可以采用如下方式访问在querystring集合中提供的值：

　　strfirstnae=request.querystring(“firstnae”)‘return“prisilla”

　　strlastnae=request.querystring(“lastnae”)‘return“desartes”

　　strra=request.querystring

　　‘return“firstnae=prisilla=desartes”

　　窗体的get和pst方法

　　在一个页面内使用段时，可以设置打开的fr标记的ethd属性值为“get”或“pst”，缺省值为“get”。假如使用“get”或省略其属性，浏览器将该值绑定在页面所有控件上，成为一个查询字符串，且附在被请求页面的url上。

　　当这个请求到达eb服务器时，其值由asp的request.querystring集合提供。然而，假如设置ethd属性为“pst”，浏览器将值包装进发送服务器的http报头中，通过request.fr集合提供给asp。

　　通过来说，可以在所有的htl窗体中使用pst方法。然而，浏览器或服务器的url字符串长度存在一定的限制。因此，附有长的字符串可能会引起溢出和某些字符串的字符被截掉。同时，查询字符串出现在浏览器的地址栏和所有的保存的链接和收藏夹中。不仅如此，还显露了通过eb服务器时在http请求中不想显示的值，它也可能出现你的服务器和其他路由服务器的日志文件中。在http请求报头中的值很少是可见的，并且不出现在日志文件中。

　　使用pst方法需要注意的小问题是，当用户重新下载时，窗体的值将不再保留，其值为空且必须重新输入。然而，当附在url上时，其值被存储为一个链接，将被保留，因此将出现在所有的url与字符串结合的请求中，这或许是个优点也可能是个缺点，这根据应用而定（一些浏览器在客户端上能够在一定范围内自动保留一个页面上的值）。

　　另一点是url与查询字符串的结合体不能包含任何空格或其他非法字符，否则的话，navigatr和一些其他的浏览器将出现问题。非法字符是那些用来分隔url和查询字符串的部分，例如“/”、“:”、“?”和“”（ie能够自动将空格转换为正确的格式——加号“+”，但其他的非法字符不能处理）。asp服务器对象提供urlende方法处理这种变换，以后章节讨论相关内容。

　　查看request和respnse对象内容

　　到目前，主要讨论了一些理论问题，没有列举特别的实例。因为已经讨论过的内容多数情况下互相之间是密切相关的。然而本书为这一章提供了一系列的实例页面，说明request和respnse对象的大多数属性。应用所讲过的实例，能够理解这些页面，并可进行相应的修改，用它们作为试验实例。

　　本章及其他所有章节的代码样例均提供给用户，可以从rx出版社站点下载。

　　http://ebdev.rx..uk/bks/2610

　　/stre/details.asp?de=2610

　　必须首先在eb服务器的/igindex.aspx"target="\_blank"title="">图所示：

　　下图显示一个htl窗体的实例，包含一些预先设置好的值，可以按自己的想法编辑这些值，然后点击“subit”按钮。

　　这将打开一个页面，如下图所示，显示集合和ttalbytes属性的全部内容。第一屏显示的是fr、querystring和kies集合。

　　注意，假如在窗体页中编辑了htl控件的值，对于kies集合以及其他的集合，在读者的计算机上的页面上可能显示不同的值。#p#分页标题#e#

　　可以从“frlletin”段中看到窗体上的htl控件的值如何在asp的request.fr集合中表示。也可以用原始的页（名称为request\_fr.asp）来试验和检测，以了解创建窗体的htl和如何与相应值相联系。

　　这个页面后面是lientertifiate集合。这里是空的，因为服务器不要求客户端提供证书。下面是的servervariables集合，下图的屏幕图显示的是集合中包含的有用的值。

　　在本书的后面附录中可以找到所有servervariables集合成员的一个列表，及其值的说明。然而，可从前面讨论的在请求页面时从客户端发出的http报头中见到这些成员。当请求收到后，eb服务器也增加它本身的一些值到集合中，正如上面可以看到的运行在iis5.0创建的页面那样。

　　1）页面是如何工作的

　　为了创建这个页面，使用了本章前面在对fr集合和如何访问其值的讨论中所看到的完全相同的代码。例如，遍历所有的集合（除request.kies外），使用：

　　freahbjiteinrequest.lletin\_nae

　　respnse.ritebjite“=“(bjite)“

　　”

　　next

　　遍历kies集合，可以使用：

　　freahbjiteinrequest.kies

　　ifrequest.kies(bjite).haskeysthen

　　‘useantherfr...eahtiterateallkeysfditinary

　　freahbjitekeyinrequest.kies(bjite)

　　resnse.ritebjitebjite“(““)=“\_

　　(bjite)(bjitekey)“

　　”

　　next

　　else

　　‘printutthekiestringasnral

　　respnse.ritebjite“=“(bjite)“

　　”

　　endif

　　next

　　为获得ttalbytes属性，可简单地使用：

　　request.ttalbytes=

　　读者应该注意到，在两个集合中出现的某些值不是从窗体的htl控件中直接得到的。querystring集合包括了两个名为hapter和saple的值，如下图所示：

　　为在请求中创建这两个值，将一个字符串附在窗体的atin属性的url上，这是可以接受的。工作方式与附在一个元素的href属性上相类似。查询字符的值出现在querystring集合中，且被pst的窗体控件值出现在fr集合中。

　　为防止非ie类的浏览器出现错误，必须将查询字符串中的空格用加号“+”来代替，读者将在第4章的server对象的urlende方法中看到更多这种情况。

　　2）创建客户端的kie

　　为了确保至少有些值出现在request.kies集合中，增加一些客户端脚本代码到原始的页面request\_fr.asp。将创建名称为visitunt的多值kie。另一个kie是由另一个页面创建的，且已经存在于浏览器中。如下图所示，读者可以看到另外的kie。

　　这是一段窗体被装载时设置文档对象的kies属性的客户端代码：

　　另外，必须将内容进行编码，以便它能被正确地传送到服务器（同样的规则也适用于将查询字符串附到url上）。在第4章中，讨论server对象的urlende方法时，读者将了解到更多细节。

　　２、查看respnse对象的成员

　　回到hapter02实例的最初的default.asp页面，这次选择“usingtherespnsebjet”链接，这个页面显示的是respnse对象的集合和属性的内容，且提供到所有respnse对象方法的链接。

　　下图是使用浏览器netsapeuniatr4.61的屏幕，用来证明使用的是纯服务器端和跨平台兼容技术。需要注意的是kies集合是为respnse对象而建立的，仅显示kie的名称而不显示其值。浏览该页时，可能得不到kie或得到与这个页面不同的kie。

　　（由于豆豆没有安装netsapeuniatr4.61，仍用ie5.5显示，并无什么区别）

　　各种respnse属性说明了将要用来创建http报头的一些信息。http报头页面的其他部分（htl和文本内容）被发往到客户端。这些属性中的一些以及所有的respnse对象的方法均有链接，允许读者打开另一个页面来显示其使用情况。我们稍后再回到这些页面。#p#分页标题#e#

　　页面中的属性是通过读取相应的属性并插入到页面中创建的。由于这些是动态链接，通过元素来选择。

　　respnse.ahentrl

　　=

　　链接到每个方法是通过链接元素，页面中唯一复杂的部分是respnse.kies集合。通过只能访问kie，读取request.kies集合中的值。当访问respnse.kie集合时，必须在发送任何输出到客户端之前结束对它的所有引用。因此在页面的上部，通过遍历集合创建页面的htl放在一个局部字符串变量中。

　　strkies=“”

　　‘eannlyreadthekeynaesandntthevaluesbeause

　　‘therespnse.kieslletinis‘ritenly’

　　freahbjiteinrespnse.kies

　　ifrespnse.kies(bjite).haskeysthen

　　‘useantherfreahtiterateallsubkeys

　　freahbjitekeyinrespnse.kies(bjite)

　　strkies=strkies“(““)

　　”

　　next

　　else

　　‘printutthekieasnral

　　strkies=strkies“

　　”

　　endif

　　next

　　然后在页面的适当点上插入结果。

　　therespnse.kieslletin

　　respnse.kies

　　isarite-nlylletinsthevaluesanntbedisplayed

　　asp中的kie的使用

　　在前面所看到页面中，一些集合、属性和方法已经链接到其他的页面，用来显示request和respnse对象的各个特性细节，我们将在本章的其余的部分研究这些内容，我们将学习那些提供给asp代码使用的集合、方法和属性的各种技术。

　　在本章前面已经看到了如何使用request.kies和respnse.kies集合来创建和阅读kie，点击上面两个页面中的任一个的“kies”链接时，这个页面包含一些设置了三个kie的值的asp代码，且在页面上显示被执行的代码，如下图所示：

　　点击“shkies”的链接时，kie的内容就显示出来了。这是通过遍历request.kies集合而得到的，这与在上一页所用的方式完全相同，如下图所示：

　　这个屏幕图显示的是运行前面看到的设置kie值的代码的结果。可能会看到其他已经存贮在计算机系统里的kie。然而，假如现在关闭浏览器然后重新打开浏览器，然后运行显示kie的页面，除了tiedkie外，所有的kie都不见了，这是由于只有这个tiedkie具有有效期的设置，其他的在浏览器关闭时，自动消失了。

　　1）kie中存储用户的细节情况

　　可以使用kie来存储这两类值：当浏览器关闭时我们不想保存的值（例如用户的注册信息）以及在用户访问站点时要保留的值。在每种情况下kie的值对于来自用户浏览器的每个页面请求的asp都是可用的。

　　然而，需要记住的是，kie只有在对kie中的虚拟路径（path）内的页面发出请求时，才会发往服务器。缺省时，假如path的值在kie中没有设置，则其值为创建kie的页面的虚拟路径。为使一个kie发往一个站点的所有页面，需要使用path=“/”。

　　这里是个实例，从自定义的lgin页面中，将用户的注册信息存贮在一个kie中，由于没有应用有效期，kie值仅在关闭这个浏览器这前保留：

　　...

　　request.kies(“user”)(“uid”)=“”

　　request.kies(“user”)(“pd”)=“”

　　request.kies(“user”).path=“/adinstuff”‘nlyappliestadinpages

　　...

　　现在，在用户从adinstuff目录或其子目录请求的每个页面中，都可以找到这个kie。假如它不存在，可以将用户重定向到注册页面：

　　if(request.kies(“user”)(“uid”)“alexher”)\_

　　r(request.kies(“user”)(“pd”)“seret”)then

　　respnse.rediret“lgin.asp?usernae=”(“user”)(“uid”)

　　endif

　　...

　　由于把kie中的用户名放在respnse.rediret的url查询字符串中，假如在口令输入时出现错误且希望用户不必重新键入用户名，可以在lgin.asp页面中使用它：#p#分页标题#e#

　　2）修改现有的kie

　　可以使用asp修改现有的kie，但不能只修改kie中的一个值。当更新一个在respnse.kies集合中的kie时，现有的值将丢失。我们可以用如下代码创建一个kie，可以使用：

　　respnse.kies(“visitunt”)(“startdate”)=dtstart

　　respnse.kies(“visitunt”)(“lastdate”)=n

　　respnse.kies(“visitunt”)(“visits”)=str(intvisits)

　　respnse.kies(“visitunt”).path=“/”‘applytentiresite

　　respnse.kies(“visitunt”).expires=dateadd(“”,3,n)

　　假如想要更新visits和lastdate的值，必须先不需改变的所有值，然后重写整个的kie：

　　datdtart=respnse.kies(“visitunt”)(“startdate”)

　　intvisits=respnse.kies(“visitunt”)(“visits”)

　　respnse.kies(“visitunt”)(“startdate”)=dtstart

　　respnse.kies(“visitunt”)(“lastdate”)=n

　　respnse.kies(“visitunt”)(“visits”)=str(intvisits)

　　respnse.kies(“visitunt”).path=“/”

　　respnse.kies(“visitunt”).expires=dateadd(“”,3,n+1)且对于几乎所有的其他respnse方法和属性，应该在写入任何内容（即打开标记或任何文本或其他的htl）到响应之前完成这个工作。