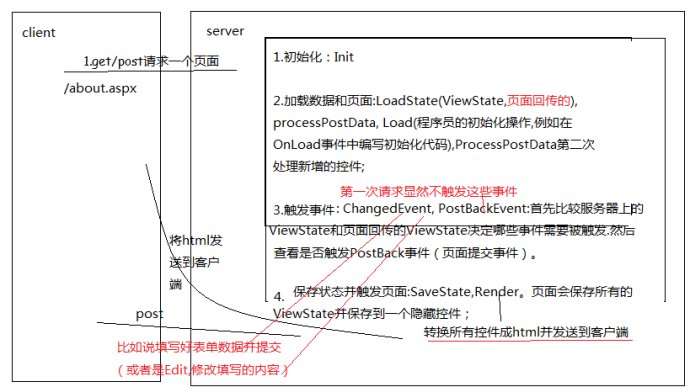
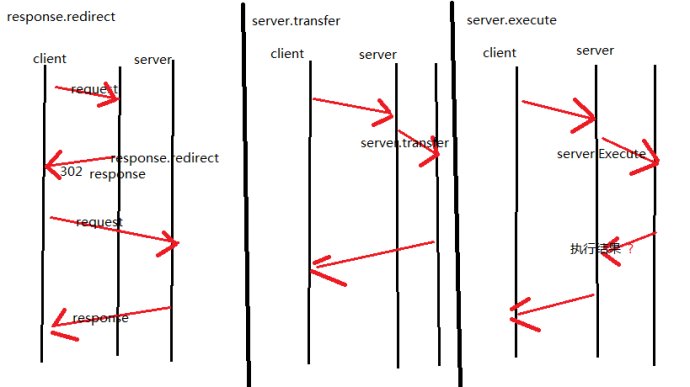
ASP.NET  
1.开发基础  
\*asp.net以什么形式运行?.net宿主的概念，ISAPI的概念，ASP.NET基本运行机制;  
.net宿主的概念:CLR被实现为一个标准的COM服务器组件,CLR具有标准的COM接口和接口GUID,在安装.net时CLR作为一个COM服务器在OS内注册;可以使用vs自带的oleview查看;CLR实际可以被非托管代码加载使用;  
  
理解难度很大，加油endeavor  
ISAPI是什么?全称为internet服务器扩展应用程序（asp.net?），而ISAPI过滤器则是指在启动ISAPI的http服务器上运行的dll,它可以用来处理特定http请求;  
asp.net是什么？是一个宿主在CLR上的ISAPI过滤器程序;ISAPI->ASP.NET->托管代码机制很复杂，简要说明:  
asp.net初始运行，加载CLR;  
iis接受一个http请求，并且判断出该请求应该被绑定到aspnet\_filter.dll这个ISAPI过滤器;  
asp.net开始处理这个http请求，判断该请求所属的网站，如果是第一个请求，新建appDomain来出处理请求，如果已有请求，则使用下游appDomain处理;  
当网站文件动态更新时，将重新加载该应用程序域;  
\*常见的httpCode有哪些?  
200  OK                       请求结束  
301  Move permanently         请求对象被永久移动到一个新的URI  
302  Move temporarily         请求对象被暂时移动到一个新的URI(response.redirect)  
303  Redicection with new access method  
304  [Not Modified](http://seo.g2soft.net/online-tools/server_status_checker.php#code-304)  
400  [Bad Request](http://seo.g2soft.net/online-tools/server_status_checker.php#code-400)  
401  [Unauthorized](http://seo.g2soft.net/online-tools/server_status_checker.php#code-401)  
403  [403 Forbidden](http://seo.g2soft.net/online-tools/server_status_checker.php#code-403)  
404  [404 Not Found](http://seo.g2soft.net/online-tools/server_status_checker.php#code-404)  
500  [Internal Server Error](http://seo.g2soft.net/online-tools/server_status_checker.php#code-500)  
501  [Not Implemented](http://seo.g2soft.net/online-tools/server_status_checker.php#code-501)  
\*GET与POST请求的区别?  
GET请求吧表单数据显现的放在URI中，并且对长度和数据值编码有所限制。POST请求把表单数据放在HTTP请求体中，没有长度的限制.详细解释如下5条:  
设计目标不同，GET作为向服务器申请资源的请求，POST作为向服务器发送数据的请求;  
GET请求在提交表单数据时，会将其添加到URL后面，具体形式:key=value,数据间用&隔开。而POST请求则直接把数据放在HTTP请求体中。也就是说GET的URL既包括服务器的URL也包含表单数据，而POST的URL只包含服务器的URL;  
HTTP协议限制了GET请求的长度，而没有限制POST请求的长度;  
HTTP协议限制GET的数据的值必须是ASCII字符，而POST却可以是整个ISO10646字符集;  
对于ASP.NET,当新建一个表单时，默认的请求方式是POST.  
\*(重点)asp.net页面的生存周期?（页面添加trace属性即可跟踪）  
所谓的页面生命周期，指的是一个ASP.NET页面类型对象从初始化到销毁 经过的步骤。在通常情况下，一旦某个请求对应到一个ASP.NET页面时，一个直接或者间接继承自System.Web.UI.Page类型的对象就会初 始化，并且开始执行页面生存周期中的所有步骤。而该类型的对象，就是客户端请求的ASP.NET页面后台托管代码.  
不分类:  
PreInit,Init, InitComplete, LoadState, ProcessPostData, PreLoad, Load, ProcessPostData(第二次), ChangedEvents, PostBackEvent, LoadComplete, PreRender, PreRenderComplete, SaveSate, SaveSateComplete, Render, Unload.  
分4个步骤:  
初始化:PreInit,Init, InitComplete--包含初始化类对象，初始化主题页面等特性，判断是否是第一次访问页面等初始功能;  
加载数据和页面:LoadState, ProcessPostData, PreLoad, Load, ProcessPostData(第二次)--首先从页面回传的数据加载ViewState对象，所有的数据都会以Base64(ViewState)编 码随页面传入服务器。然后开始处理回传的数据，也就是把表单中的键/值对存入对象中。随后开始加载页面,程序员通常可以在这里做一些初始化的编程，例如在 OnLoad事件中编写页面初始化代码，最后执行ProcessPostData，已处理Load时新添加的数据。  
两次ProcessPostData均是必须的:  
第一次：保证了页面在被加载之前，所有的数据从页面上被读入，这样页面加载时就可以访问所有的数据。  
第二次：使页面加载时新建立的控件中的数据也能够得到处理。  
触发事件:ChangedEvents, PostbackEvent.这里首先会比较ViewState中数据和页面上一次回传的数据，决定哪一些事件需要被触发，这里的事件是被逐一触发的，但顺序无法确定。然后查看是否触发PostBack事件，也就是页面提交的事件;  
保存状态并呈现页面:SaveState, SaveStateComplete, Render等步骤.首先页面会编码保存所有的ViewState数据，然后将其嵌入到页面的一个隐藏控件中。接着转换所有的控件标签并生成页面的html，并且发送回客户端。  
[](http://blog.photo.sina.com.cn/showpic.html#url=http://album.sina.com.cn/pic/0024gZI6gy6HBZsD2vH5d)  
  
2.控件和页面  
\*什么是静态/动态页面?  
静态页面是指不需要服务器代码额外处理，直接可以呈现给用户的页面，如html.相对的需要服务器针对每一个请求做处理的页面被称为动态页面，如aspx.  
注意：现在大型门户网站采用静态页面动态页面相结合的结束，配合缓存技术和负载平衡技术可以支持超大量的并行访问.  
\*简述ViewState功能和实现机制  
一般来说，构建BS结构系统最大难点在于如何保存访问的状态，和CS结构系统可以把用户数据保存在客户的内存和存储设备上不同，BS结构的系统没有得到用 户的任何授权，而把每一个访问者的数据保存在服务器代价太大，asp.net对这个问题的解决方法就是页面状态的保存:ViewState机制(Java 中的Page);  
Asp.net服务器空间从Control继承名为ViewState的属性，这使它可以更容易的加入状态管理。ViewState的类型是System.Web.UI.StateBag,它是存储名称/值对的字典;可以使用户在使用动态页面时获得连续性动作的功能;(就是说ViewState并不是存储在服务器中，而是通过不断的在服务器和客户端之间传送);  
asp.net中有些控件本身实现类保存数据的功能,而不依靠ViewState.例如TextBox，即使ViewState关闭，数据也不会丢失;ViewState可以保存任何可序列化的对象，而不仅仅限于控件的状态数据,这样的保存依然是限制在页面的范围内的.  
详细机制:  
客户申请一个新的带有ViewState字段的页面，第一次申请时ViewState字段内数据为空;  
客户提交表单，这是ViewState字段作为表单的一部分被提交，当然这时也为空;  
服务器从表单中读取ViewState的数据,解码解密，并且保存入ViewState对象，这时仍然为空;  
服务器处理完表单数据，将表单数据存入ViewState,这是ViewState有值。  
服务器把ViewState对象加密并编码，然后写入ViewSate字段，这时页面ViewState字段就包含了上次提交的表单内控件数据以及服务器代码写入ViewState的数据;  
服务器将页面发送到客户端，客户端的得到ViewState数据(这个ViewState并不做任何的处理);  
注意:对input(\_viewState)的value进行解码解密，发现viewState的真是面貌是一个xml文件，并且存储了所有ViewState对象内的值.  
\*Session有哪几种存储方法，如何设置？  
Session:一个数据字典，用于在会话中维持状态数据，从理论上来说，同一个会话过程中,Session数据是不会丢失的(关闭浏览器或主动结束会 话).用户认证(authentication)后--输入用户名密码，用户信息可以保存在Session中，包括购物车小例程。  
Session数据4中保存的方式:  
iis进程中:inetinfo.exe这个进程中;()  
状态服务器:asp.net State Service()  
该进程独立于asp.net辅助进程和iis应用程序池.保证在重启Web应用程序后仍能保留会话状态，并可使会话状态应用于多个Web服务器;  
sqlserver:()  
安装状态数据库systemroot\Microsoft.Net\Framework\versionNumber\Aspnet\_regsql.exe  
自定义:()  
定义继承自SessionStateStoreProviderBase类型的类并重写方法;  
\*如何嵌套使用GridView：实现文本需求的一种方式，还有脚本实现，flash技术等.  
        protected void GridView1\_RowDataBound(object sender, GridViewRowEventArgs e)  
        {  
            if (e.Row.RowType == DataControlRowType.DataRow)  
            {  
                GridView gvSon = e.Row.FindControl("GridView2") as GridView;  
                Label label = e.Row.Cells[0].FindControl("Label1") as Label;  
                gvSon.DataSource = GetDetailById(label.Text);  
                gvSon.DataBind();  
            }  
        }  
        private object GetDetailById(string id)  
        {  
            DataTable dt = PrepareDataSource();  
            DataTable result = dt.Clone();  
            foreach(DataRow tr in dt.Rows){  
                if (id == tr["Id"].ToString()) {  
                    DataRow newrow = result.NewRow();  
                    newrow.ItemArray = tr.ItemArray;  
                    result.Rows.Add(newrow);  
                }  
            }  
            return result;  
        }  
存在一定的疑问(关于row的操作)，模板列，模板编辑等模式的熟悉;  
\*举例实现页面跳转的方式并说明其机制;  
response.Redirect:302,迫使客户端跳转;  
Server.transfer:服务器内的跳转;  
server.execute:服务器内呼叫，新页面执行的结果追加到当前页面的响应中;类似方法调用;  
javascript:  
跳转愿望发生在客户端，很高效--是浏览器直接跳转到新页面，并且可以完全控制跳转前的工作；  
跳转愿望发生在服务器端,相对于Response.Redirect在于，可以在跳转前执行一些必要的客户端工作，例如弹出提示框,例子:  
response.write("  
超链接(a):  
[](http://blog.photo.sina.com.cn/showpic.html#url=http://album.sina.com.cn/pic/0024gZI6gy6HCfqXkYece)  
\*asp.net页面间传值方式:　页面传值是学习asp.net初期都会面临的一个问题，总的来说有页面传值、存储对象传值、ajax、类、model、表单等。但是一般来说，常用的较 简单有QueryString，Session，Cookies，Application，Server.Transfer。