* WebForm：

ViewState是ASP.NET中用来保存[WEB控件](https://www.baidu.com/s?wd=WEB%E6%8E%A7%E4%BB%B6&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvPvNbPj9WnWRYnH7-uHu90ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHnvn1f4rHn)回传时状态值一种机制。在[WEB窗体](https://www.baidu.com/s?wd=WEB%E7%AA%97%E4%BD%93&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvPvNbPj9WnWRYnH7-uHu90ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHnvn1f4rHn)(FORM)的设置为runat="server",这个窗体(FORM)会被附加一个隐藏的属性\_VIEWSTATE。\_VIEWSTATE中存放了所有控件在ViewState中的状态值。   
  
ViewState是类Control中的一个域，其他所有控件通过继承Control来获得了ViewState功能。它的类型是system.Web.UI.StateBag，一个名称/值的对象集合。   
  
当请求某个页面时，ASP.NET把所有控件的状态[序列化](https://www.baidu.com/s?wd=%E5%BA%8F%E5%88%97%E5%8C%96&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvPvNbPj9WnWRYnH7-uHu90ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHnvn1f4rHn)成一个字符串，然后做为窗体的隐藏属性送到客户端。当客户端把页面回传时，ASP.NET分析回传的窗体属性，并赋给控件对应的值。当然这些全部是由ASP.NET负责的.

ViewState的原理  
1.浏览器请求Default.aspx页面  
2.在服务器端 发现创建的ViewState 这个时候 会自动创建一个名字叫做\_\_VIEWSTATE(双下滑线 全部是大写)  
的隐藏域 其隐藏域的值经过base64[加密](http://www.2cto.com/Article/jiami/)以后返回到浏览器端这一加密过程在页面生命周期  
的SaveState事件中的SaveAllState方法中完成  
3.当浏览器提交表单的时候 将\_\_VIEWSTATE的隐藏域也一起提交到服务端 这个时候 页面生命周期的ReadState事件  
的ReadAllState方法会将加密后的值反base64解密 最后将值赋值给名字叫做name的ViewState  
4.最后来操作ViewState中的值

优点：

一、耗费的服务器资源较少（和Application、Session相比）。因为，视图状态数据都写入了客户端计算机中。  
  
二、易于维护。默认情况下，.NET系统自动启用对控件状态数据的维护。  
  
三、增强的安全功能。视图状态中的值经过哈希计算和压缩，并且针对Unicode实现进行编码，其安全性要高于使用隐藏域。  
  
缺点：  
  
一、性能注意事项。由于视图状态存储在页本身，因此如果存储较大的值，即使在视图状态分块的情况下，用户显示页和发送页时的速度仍然可能减慢。  
  
二、设备限制。移动设备可能没有足够的内存容量来存储大量的视图状态数据。因此，移动设备上的服务器控件时，将使用其他的实现方法。  
  
三、潜在的安全风险。视图状态存储在页上的一个或多个隐藏域中。虽然视图状态以哈希格式存储数据，但他能被篡改。如果直接查看页输出源，能看到隐藏域中的信息，这导致潜在的安全性问题。

☆ MVC：

MVC代表: *模型-视图-控制器* 。MVC是一个架构良好并且易于测试和易于维护的开发模式。基于MVC模式的应用程序包含：

· **M**odels： 表示该应用程序的数据并使用验证逻辑来强制实施业务规则的**数据类**。

· **V**iews： 应用程序动态生成 HTML所使用的模板文件。

· **C**ontrollers： 处理浏览器的请求，取得数据模型，然后指定要响应浏览器请求的视图模板。

ASP.NET MVC会调用不同的控制器类（和其内部不同的操作方法）这取决于传入URL。所使用的ASP.NET MVC的默认URL路由逻辑使用这样的格式来判定哪些代码以便调用：

/[Controller]/[ActionName]/[Parameters]

你也可在App\_Start/RouteConfig.cs文件内通过配置URL路由解析规则

public ActionResult Welcome(string name="TianJun",int num=1)

{

ViewBag.Name = name;

ViewBag.Num = num;

return View(); //调用：http://localhost:1160/Home/Welcome?Name=zhangsan&Num=2

}

<http://localhost:1160/Home/Welcome?Name=zhangsan&Num=2>

在Asp.net MVC 3 web应用程序中，我们会用到[ViewData](http://msdn.microsoft.com/en-us/library/system.web.mvc.viewpage.viewdata.aspx)与[ViewBag](http://msdn.microsoft.com/en-us/library/system.web.mvc.controllerbase.viewbag(v=vs.98).aspx)，对比一下：

|  |  |
| --- | --- |
| **ViewData** | **ViewBag** |
| 它是Key/Value字典集合 | 它是[dynamic](http://msdn.microsoft.com/en-us/library/dd264741.aspx)类型对像 |
| 从Asp.net MVC 1 就有了 | ASP.NET MVC3 才有 |
| 基于Asp.net 3.5 framework | 基于Asp.net 4.0与.net framework |
| ViewData比ViewBag快 | ViewBag比ViewData慢 |
| 在ViewPage中查询数据时需要转换合适的类型 | 在ViewPage中查询数据时不需要类型转换 |
| 有一些类型转换代码 | 可读性更好 |

使用Layout属性设置一个不同的布局页面，或者设置为null指明不使用布局文件

如：

@{

Layout = "~/Views/Shared/\_Layout.cshtml";

}

或：

@{

Layout = null;

}

持久化(Persistence)，是指在应用程序中能永久地保存各个处理状态信息的机制。如果没有持久化这个机制，状态只能保存在内存中，机器关机后就会丢失。

为了让开发人员不用手动去做这种“匹配”工作，人们研发了很多ORM工具（如Entity Framework、NHibernate等）。ORM(Object Relation Mapping)工具，顾名思义，它的角色就是为了解决“关系”和“面向对象”之间的“失配”，它可以使得开发人员不用过多关心持久层而可以花更多的时间专注于业务。