**[MVC和WebForm区别](https://www.cnblogs.com/kongsq/p/9885771.html)**

WebForm的理解

1、 WebForm概念

ASP.NETWebform提供了一个类似于Winform的事件响应GUI模型（event-drivenGUI），隐藏了HTTP、HTML、JavaScript等细节，将用户界面构建成一个服务器端的树结构控件（Control），每个控件通过ViewState保持自己的状态，并自动把客户端的js事件和服务器端的事件联系起来。这种做法使得开发WinForm和WebForm程序具有相近的开发体验，填平WinForm开发（有状态、面向对象的）和WebForm开发（无状态、面向HTML的）之间的鸿沟。

2、WebForm 请求流程

比如访问 http://www.qidian.com/Default.aspx

1、Http请求（物理地址：Default.aspx）

①发送请求： 浏览器向服务器发送请求报文，此时由IIS虚拟目录接收。（通过配置过IIS，把网站挂载在服务器上，通过访问虚拟目录的方式访问网站的。）

②转交请求： 服务器端的IIS软件接收到请求后，把请求交给.NET FrameWork进行处理。

③创建页面类对象：.NET FrameWork根据请求的地址index.aspx,会创建对应的index\_aspx类的对象（页面对象）。

IIS的内部机制：

实现一个IHttphandler的接口

该接口实现一个ProcessRequest方法

该方法会调用对应页面的Page\_load方法

处理的业务逻辑或者是访问数据库的代码

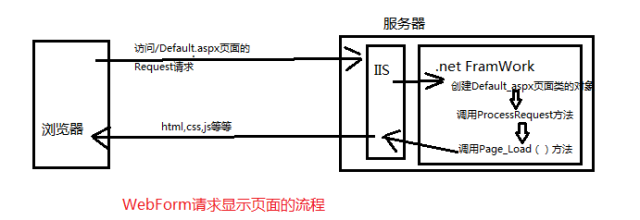
要输出的Html或者其它内容

2、返回给浏览器

①转交回复： 通过IIS传输出给浏览器，要输出的html元素或其他内容（html+js+css等）

②解析为图形界面：浏览器解析html代码，并翻译为图形化界面

如图：



2、 WebForm的优点：

1.   有大量的服务器控件支持，比如：GridView、Repeater等控件可以方便的进行数据绑定，从而减少的大量代码的编写。

2.   学习成本低，由于微软封装的比较深，造成深入学习的难度加大。

3.   基于事件驱动编程，如：click事件等，aspx和cs文件分离，即显示逻辑和处理逻辑分离。

4.   支持视图状态，每个控件以“隐藏域”的形式存在当前表单页面未达到“有状态”，即ViewState。

3、 WebForm的缺点

1.   由于使用的ViewState会增加页面的负担，造成性能不是很高。

2.   代码重用性不高，缺少对并行开发的支持，

3.   因为采用code-behind 代码后植技术，使aspx页面与cs紧密耦合度太高。

4.   对Seo不友好，因为URL指定具体的aspx页面。

5.   因为紧密耦合度太高、使用大量的事件处理函数，不利于单元测试。

二、       MVC的理解

1、 MVC的概念

Asp.netMvc架构模式是一种 低耦合、可测试的web应用程序框架，它是基于CLR和成熟的MVC架构构建的。ASP.NET MVC不支持ViewState和服务器控件。

2、 MVC的请求流程

比如访问：http://www.google.com.hk/FirstPage/Default

1、Http请求（逻辑地址：FirstPage/Default）

①发送请求（FirstPage/Default）

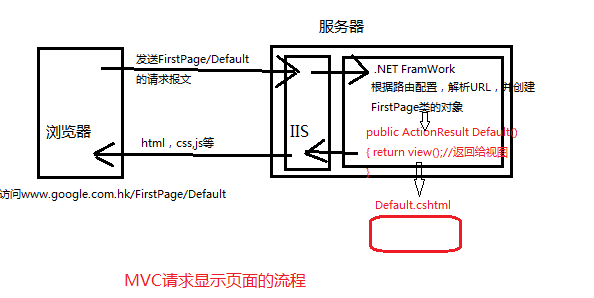
②转交请求（同上）

③创建类对象+方法

.NET FrameWork根据路由配置，解析URL，并创建news类的对象，并调用对象的index方法。通过View方法加载视图，然后访问视图文件夹下的index.cshtml

2、返回给浏览器

如图：



3、 MVC的优点

1.   架构降低了程序间的耦合性。

2.   不支持ViewState，页面更加干净，可以提升程序的性能。

3.   支持并行开发，可扩展性好，继承了asp.net的特性，表单验证、缓存、会话等。

4.   由于程序耦合度低，可以比较顺利的进行单元测试。

5.   通过修改路由规则，可以控制生成自定义的url，因此控制生成seo友好的url将更加容易。

6.   强类型view实现，更安全，更高效。

4、 MVC的缺点

需要有一定的html、css、js、jquery前端技术，也就增加了一些学习的成本。

MVC和WebForm的选择场景

新开发的项目建议采用MVC。

想要快速开发的中小型项目可以选择WebForm。

比较关注单元测试、性能、SEO、代码重用性的话建议采用MVC。

最后结语

MVC和WebForm都有自己的优缺点，目前两种技术都存在，证明都有自己的价值，虽然现在MVC应用越来越广发，我们不能一味的否定WebForm。一个高级程序员用WebForm实现的网站性能并不一定比初级程序员用MVC实现的网站性能差，最重要的还是努力提升自己的技术能力才是王道！