**·MVC原理——Web MVC**

**1、Web开发中的请求-响应模型：**



在Web世界里，具体步骤如下：

1、  Web浏览器（如IE）**发起请求**，如访问http://sishuok.com

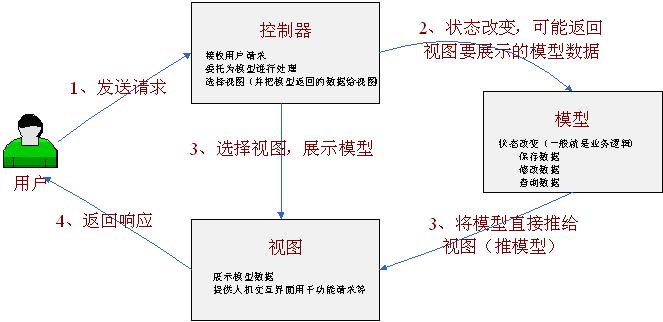
2、  Web服务器（如Tomcat）**接收请求**，**处理请求**（比如用户新增，则将把用户保存一下），最后**产生响应**（一般为html）。

3、web服务器**处理完成后，返回内容给web客户端**（一般就是我们的浏览器），客户端**对接收的内容进行处理**（如web浏览器将会对接收到的html内容进行**渲染**以展示给客户）。

因此，在Web世界里：都是Web客户端发起请求，Web服务器接收、处理并产生响应。一般**Web服务器是不能主动通知Web客户端更新内容**。虽然现在有些技术如服务器推（如Comet）、还有现在的HTML5 websocket可以实现Web服务器主动通知Web客户端。

**2、标准MVC模型概述**

MVC模型：是一种架构型的模式，本身不引入新功能，只是帮助我们将开发的结构组织的更加合理，使**展示与模型分离**、**流程控制逻辑**、**业务逻辑调用与展示逻辑分离**。如图



首先让我们了解下MVC（Model-View-Controller）三元组的概念：

**Model（模型）**：数据模型，**提供要展示的数据**，因此**包含数据和行为**，可以认为是领域模型或JavaBean组件（包含数据和行为），不过现在一般都分离开来：Value Object（数据） 和服务层（行为）。也就是**模型提供了模型数据查询和模型数据的状态更新等功能**，**包括数据和业务**。

**View（视图）**：**负责进行模型的展示**，一般就是我们见到的用户界面，客户想看到的东西。

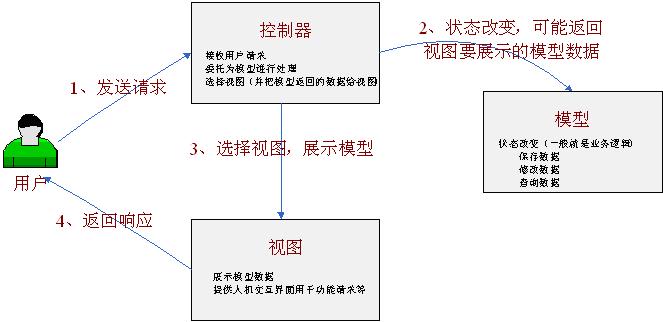
**Controller（控制器）**：**接收用户请求，委托给模型进行处理**（状态改变），处理完毕后**把返回的模型数据返回给视图**，由视图负责展示。 也就是说控制器做了个调度员的工作。

 从上图我们还看到，在标准的MVC中模型能主动推数据给视图进行更新（**观察者设计模式**，在模型上注册视图，当**模型更新时自动更新视图**），但在Web开发中模型是无法主动推给视图（无法主动更新用户界面），因为在Web开发是**请求-响应模型**。

 那接下来我们看一下在Web里MVC是什么样子，我们称其为 Web MVC 来区别标准的MVC。

**3、Web MVC概述**

模型-视图-控制器概念和标准MVC概念一样，请参考1.2，我们再看一下Web MVC标准架构，如下图：



在Web MVC模式下，模型无法主动推数据给视图，如果用户想要视图更新，需要再发送一次请求（即请求-响应模型）。