

## 一、MVC 模型

## 1、Model (模型) : javabean

## 2、View (视图) : jsp

通常视图是依据模型数据创建的。

### 3、Controller (控制器) : servlet

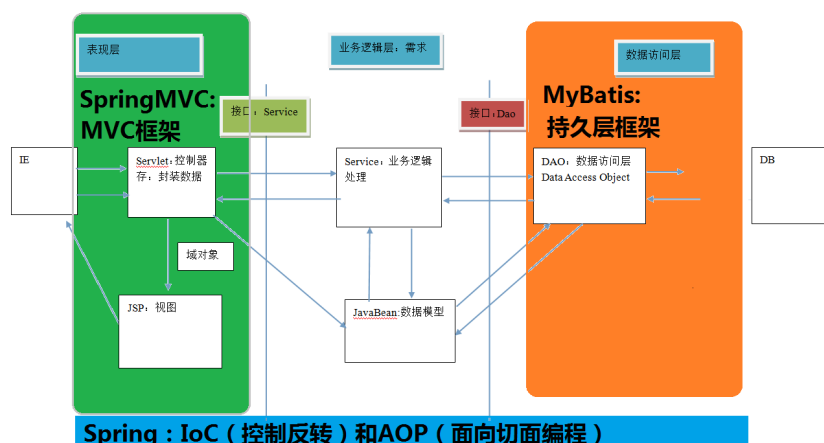
它相对于前两个不是很好理解，这里举个例子：

此时除了 is 的校验之外，服务器端也应该有数据准确性的校验，那么校验就是控制器的该做的。

当校验失败后，由控制器负责把错误页面展示给使用者。

如果校验成功，也是控制器负责把数据填充到模型，并且调用业务层实现完整的业务需求。

## 二、SpringMVC位置 (表现层)



### 三、SpringMVC 的优势

### 1、清晰的角色划分：

### 前端控制器 (DispatcherServlet)

请求到处理器映射 (HandlerMapping)

## 处理器适配器 (HandlerAdapter)

视图解析器 (ViewResolver)

处理器或页面控制器 (Controller)

验证器 ( Validator)

命令对象 (Command 请求参数绑定到的对象就叫命令对象)

表单对象 (Form Object 提供给表单展示和提交到的对象就叫表单对象)。

2、分工明确，而且扩展点相当灵活，可以很容易扩展，虽然几乎不需要。

3、由于命令对象就是一个 POJO，无需继承框架特定 API，可以使用命令对象直接作为业务对象。

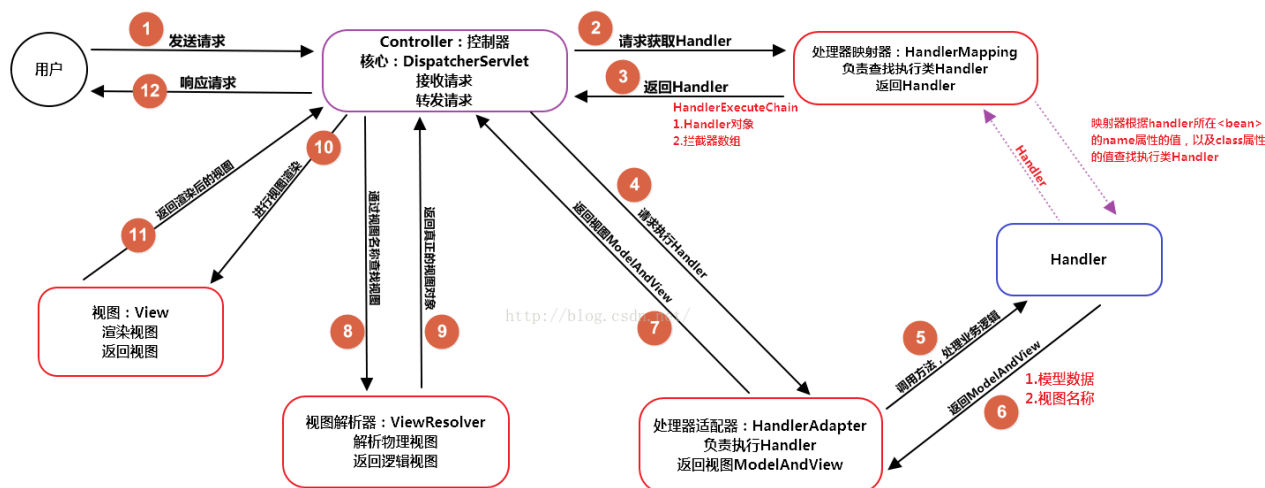
4、和 Spring 其他框架无缝集成，是其它 Web 框架所不具备的。

5、可适配，通过 HandlerAdapter 可以支持任意的类作为处理器。

- 6、可定制性，HandlerMapping、ViewResolver 等能够非常简单的定制。
- 7、功能强大的数据验证、格式化、绑定机制。
- 8、利用 Spring 提供的 Mock 对象能够非常简单的进行 Web 层单元测试。
- 9、本地化、主题的解析的支持，使我们更容易进行国际化和主题的切换。
- 10、强大的 JSP 标签库，使 JSP 编写更容易。

.....还有比如RESTful风格的支持、简单的文件上传、约定大于配置的契约式编程支持、基于注解的零配置支持等等。

## 四、SpringMVC的执行流程（重点）



1. 用户发起请求到前端控制器（Controller）
2. 前端控制器没有处理业务逻辑的能力，需要找到具体的模型对象处理（Handler），到处理器映射器（HandlerMapping）中查找Handler对象（Model）。
3. HandlerMapping返回执行链，包含了2部分内容：① Handler对象、② 拦截器数组
4. 前端处理器通过处理器适配器包装后执行Handler对象。
5. 处理业务逻辑。
6. Handler处理完业务逻辑，返回ModelAndView对象，其中view是视图名称，不是真正的视图对象。
7. 将ModelAndView返回给前端控制器。
8. 视图解析器（ViewResolver）返回真正的视图对象（View）。
9. （此时前端控制器中既有视图又有Model对象数据）前端控制器根据模型数据和视图对象，进行视图渲染。
10. 返回渲染后的视图（html/json/xml）返回。
11. 给用户产生响应。

## 五、SpringMVC 和 Struts2 的优略分析

### 1、共同点：

它们都是表现层框架，都是基于 MVC 模型编写的。  
 它们的底层都离不开原始 ServletAPI。  
 它们处理请求的机制都是一个核心控制器。

### 2、区别：

Spring MVC 的入口是 Servlet, 而 Struts2 是 Filter

Spring MVC 是基于方法设计的, 而 Struts2 是基于类, Struts2 每次执行都会创建一个动作类。所以 Spring MVC 会稍微比 Struts2 快些。

Spring MVC 使用更加简洁, 同时还支持 JSR303, 处理 ajax 的请求更方便

(JSR303 是一套 JavaBean 参数校验的标准, 它定义了很多常用的校验注解, 我们可以直接将这些注解加在我们 JavaBean 的属性上面, 就可以在需要校验的时候进行校验了。)

Struts2 的 OGNL 表达式使页面的开发效率相比 Spring MVC 更高些, 但执行效率并没有比 JSTL 提升, 尤其是 struts2 的表单标签, 远没有 html 执行效率高。