SpringMVC工作原理

Spring MVC 分离了控制器、模型对象、分派器以及处理程序对象的角色，这种分离让它们更容易进行定制。

Spring的MVC框架主要由DispatcherServlet、处理器映射、处理器(控制器)、视图解析器、视图组成。

SpringMVC原理图

SpringMVC接口解释：

DispatcherServlet接口： Spring提供的前端控制器，所有的请求都有经过它来统一分发。在DispatcherServlet将请求分发给Spring Controller之前，需要借助于Spring提供的HandlerMapping定位到具体的Controller。

HandlerMapping接口： 能够完成客户请求到Controller映射。

Controller接口： 需要为并发用户处理上述请求，因此实现Controller接口时，必须保证线程安全并且可重用。 Controller将处理用户请求，这和Struts Action扮演的角色是一致的。一旦Controller处理完用户请求，则返回ModelAndView对象给DispatcherServlet前端控制器，ModelAndView中包含了模型（Model）和视图（View）。

从宏观角度考虑，DispatcherServlet是整个Web应用的控制器；从微观考虑，Controller是单个Http请求处理过程中的控制器，而ModelAndView是Http请求过程中返回的模型（Model）和视图（View）。 ViewResolver接口：Spring提供的视图解析器（ViewResolver）在Web应用中查找View对象，从而将相应结果渲染给客户。

SpringMVC运行原理：

1、客户端请求提交到DispatcherServlet

2、由DispatcherServlet控制器查询一个或多个HandlerMapping，找到处理请求的Controller

3、DispatcherServlet将请求提交到Controller

4、Controller调用业务逻辑处理后，返回ModelAndView

5、DispatcherServlet查询一个或多个ViewResoler视图解析器，找到ModelAndView指定的视图

6、视图负责将结果显示到客户端

DispatcherServlet是整个Spring MVC的核心。它负责接收HTTP请求组织协调Spring MVC的各个组成部分。

其主要工作有以下三项：

1、截获符合特定格式的URL请求。

2、初始化DispatcherServlet上下文对应的WebApplicationContext，并将其与业务层、持久化层的WebApplicationContext建立关联。

3、初始化Spring MVC的各个组成组件，并装配到DispatcherServlet中。

注明：Spring MVC Controller默认是单例的：

单例的原因有二：

1、为了性能。

2、不需要多例（方法级别，没有共享属性的情况下，线程安全的。）

3、SpringIOC默认注入就是单例的。

最佳实践：

1、不要在controller中定义成员变量。

2、万一必须要定义一个非静态成员变量时候，则通过注解@Scope("prototype")，将其设置为多例模式。