1、SpringMVC的概述

SpringMVC是一个Web层（控制层）的框架，主要用来负责与客户端（浏览器端）交互，以及业务逻辑的调用。

Spring框架是Service层（业务访问层）的框架。

SpringMVC的特点是简单、易用，性能佳。

SpringMVC替代了之前的Struts2框架，也是用于替代Servlet技术的功能。

2、SpringMVC相比Servlet技术的优势

（1）Servlet的开发配置相对麻烦，Servlet特别多的时候web.xml文件就会非常臃肿

（2）每个Servlet都只能处理一个功能，如果需要多个功能就需要开发多个Servlet，项目中存在大量的Servlet也会显得臃肿。

（3）获取请求参数进行类型转换，包括封装到bean的过程比较繁琐。

（4）其它开发中不方便的地方，例如：乱码问题、数据格式的处理、表单的验证等等。

3、SpringMVC的组件

（1）前端控制器（DispatcherServlet）

本质上前端控制器DispatcherServlet是一个Servlet，相当于一个中转站，所有的访问都会走到这个Servlet中，在根据配置进行中转到响应的Handler（处理器）中进行处理，获取到数据和视图后，在使用相应视图做出响应。

（2）处理器映射器（HandlerMapping）

本质上是一段映射关系，将访问路径和对应的Handler存储为映射关系，在需要时供前端控制器查阅。

（3）处理器适配器（HandlerAdapter）

本质上就是一个适配器，可以根据配置的要求找到对应的Handler来运行。前端控制器通过处理器映射器找到对应的Handler信息之后，将请求响应和对应Handler信息交由处理器适配器处理，处理器适配器找到真正的Handler来执行，Handler执行，会产生一个ModelAndView类型的结果返回给前端过滤器。

（4）视图解析器（ViewResolver）

本质上也是一个映射关系，可以将视图名称映射到真正的视图地址。

前端控制器调用处理器适配器完成后得到ModelAndView结果，将View信息传递给视图解析器得到真正的View（浏览器端的JSP页面）。

（5）视图（View）

本质上是将Handler处理器中返回的Model数据嵌入到视图解析器解析后得到的jsp页面中，向浏览器端做出响应。

4、SpringMVC的第一个入门案例

Step1：新建一个Web项目

Step2：导入开发需要的jar包

Step3：在web.xml文件中配置前端控制器DispatcherServlet

Step4：创建一个SpringMVC框架的核心配置文件

SpringMVC默认会自动在Web项目中的WEB-INF目录下去寻找（前端控制器

servlet-name）的名字-servlet.xml作为当前SpringMVC的核心配置文件。

一般习惯把核心配置文件放在src目录下，需要在web.xmml中配置一下核心配置文件的位置。

springmvc.xml核心配置文件需要引入Spring框架开发中的一些约束（beans,context,mvc...）

step5：在src下定义包，在包下开发Handler，在Handler中定义方法，通过@RequestMapping完成对方法的注解配置，并方法返回ModelAndView对象。

Step6：在springmvc.xml核心文件中配置视图解析器

Step7：前端控制器根据视图解析器的解析结果，跳转到真正的View视图页面，并在真

正的View视图页面中获取Model中的数据

SpringMVC框架的细节

5、@RequestMapping注解

通过注解方法实现路径到处理器方法的映射。

可以用在类或方法上

用在方法上表示将该方法作为一个处理器，且和指定路径做映射；

用在类上则配置的路径会作为这个类中所有处理器中方法的父路径使用。

6、@RequestMapping注解中的属性

（1）value属性

指定要将当前处理器（类或者方法）绑定到哪个访问路径上

可以配置多个路径；

路径中也可以使用\*号作为通配符。