

1、用户向服务器发送请求，请求被SpringMVC的前端控制器DispatcherServlet截获。

2、DispatcherServlet对请求的URL（统一资源定位符）进行解析，得到URI(请求资源标识符)，然后根据该URI，调用HandlerMapping获得该Handler配置的所有相关的对象，包括Handler对象以及Handler对象对应的拦截器，这些对象都会被封装到一个HandlerExecutionChain对象当中返回。

3、DispatcherServlet根据获得的Handler，选择一个合适的HandlerAdapter。HandlerAdapter的设计符合面向对象中的单一职责原则，代码结构清晰，便于维护，最为重要的是，代码的可复制性高。HandlerAdapter会被用于处理多种Handler，调用Handler实际处理请求的方法。

4、提取请求中的模型数据，开始执行Handler(Controller)。在填充Handler的入参过程中，根据配置，spring将帮助做一些额外的工作

 消息转换：将请求的消息，如json、xml等数据转换成一个对象，将对象转换为指定的响应信息。

 数据转换：对请求消息进行数据转换，如String转换成Integer、Double等。

 数据格式化：对请求的消息进行数据格式化，如将字符串转换为格式化数字或格式化日期等。

 数据验证：验证数据的有效性如长度、格式等，验证结果存储到BindingResult或Error中。

 5、Handler执行完成后，向DispatcherServlet返回一个ModelAndView对象，ModelAndView对象中应该包含视图名或视图模型。

6、根据返回的ModelAndView对象，选择一个合适的ViewResolver(视图解析器)返回给DispatcherServlet。

7、ViewResolver结合Model和View来渲染视图。

8、将视图渲染结果返回给客户端。

以上8个步骤，DispatcherServlet、HandlerMapping、HandlerAdapter和ViewResolver等对象协同工作，完成SpringMVC请求—>响应的整个工作流程，这些对象完成的工作对于开发者来说都是不可见的，开发者并不需要关心这些对象是如何工作的，开发者，只需要在Handler(Controller)当中完成对请求的业务处理。