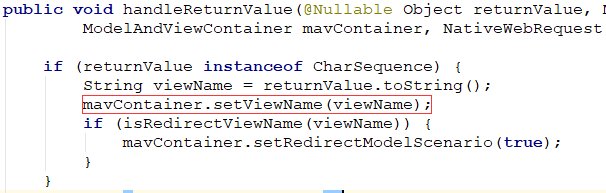
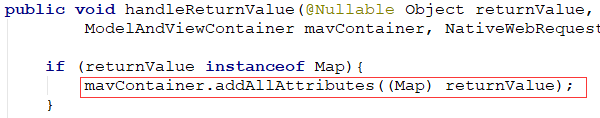
## SpringMVC的基本流程

1. 客户端发送HTTP请求，Web应用服务器（例如tomcat）接受到这个请求，如果匹配DispatcherServlet的请求路径，则Web容器将请求转交给DispatcherServlet。
2. DispatcherServlet接受到这个请求后，将根据请求的信息以及HandlerMapping的配置找到处理请求的Handler，可以将HandlerMapping看作路由控制器，将Handler看作目标主机。
3. 当找到对应的Handler之后，通过HandlerAdapter对Handler进行封装，再以统一的适配器接口调用Handler。HandlerAdatper顾名思义是一个适配器，用于统一的接口对各种Handler方法进行调用。
4. 在Handler方法调用之前会创建一个mavContainner，是ModelAndViewContainer类型，里面有ModelMap属性和View属性。如果handler方法参数有ModelMap或者Map，那么参数解析的时候就会从中mavContainner获取ModelMap，作为参数传入。在调用handler方法后会得到一个返回值returnValue，然后就是对returnValue进行处理。

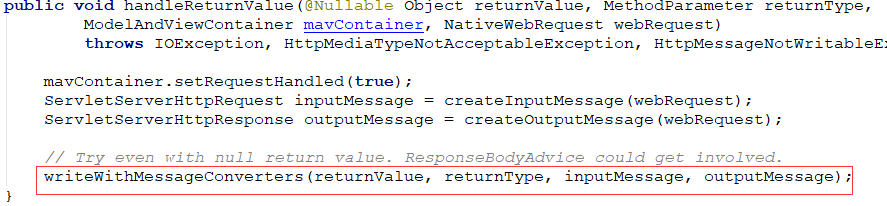
如果返回的是一个字符串，并且没有@ResponseBody注解，那么mavContainner就会设置ViewName:

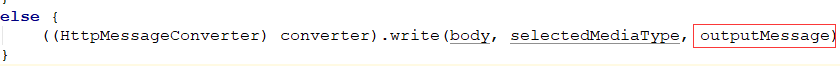


如果返回值是个Map，那么就会将该Map添加到mavContainner中：



如果handler方法上有@ResponseBody注解，那么其返回值均会写入HttpResponse中：





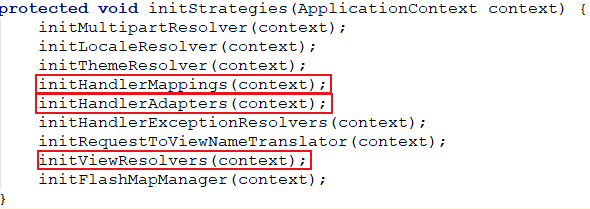
1. 处理器完成业务逻辑后，将返回一个ModelAndView给DispatcherServlet，ModelAndView中包含了视图的逻辑名和模型数据信息，ModelAndView也有可能为空。
2. ModelAndView中包含的是逻辑视图，并非是真正的视图，DispatcherServlet借由ViewResolver完成逻辑视图名到真实视图名的解析工作。
3. 当得到真实的视图对象View后，DispatcherServlet就使用这个View对象对ModelAndView中的模型数据进行渲染。
4. 最终客户端得到的响应消息可能是一个普通的HTML，可能是一个XML或者JSON，甚至是一张图片或者一个PDF。

URL🡪tomcat🡪DispatcherServlet🡪HandlerMapping🡪Handler🡪HandlerAdapter🡪ModelAndView🡪ViewResolver🡪View🡪根据ModelAndView属性进行渲染。

## 注意点：

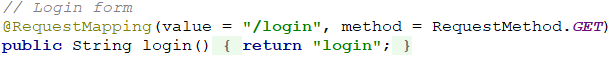
1. DispatcherServlet的初始化：

DispatcherServlet也是Servlet，在第一次访问的时候需要将其初始化加入到内存中，所以DispatcherServlet也有这样的一个初始化的方法🡪initStrategies(ApplicationContext)

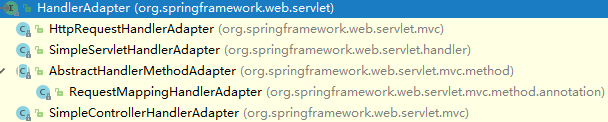


他会挨个调用初始化方法，这边主要关注的有三个，HandlerMapping，HandlerAdapter，ViewResolver的初始化。

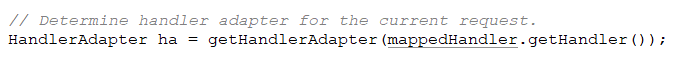
**initHandlerMappings**会从ApplicationContext中获取注入的HandlerMapping对象，如果获取到的是null，那么就会设置一些默认的handlerMapping对象，例如BeanNameUrlHandlerMapping和RequestMappingHandlerMapping，前者是根据URL直接找到与之相同的bean，该bean需要实现Controller方法。后者是将URL和一个方法匹配，该方法需要是@Controller注解类中的方法，并且方法上需要有@RequestMapping注解:



**initHandlerAdapter**会从ApplicationContext中获取已经注册的HandlerAdapter，如果没有获取到，那么使用的是SimpleControllerHandlerAdapter，该类用于支持url到Bean之间的映射。他将获取到的Bean强转为Controller，并执行相关的代码。

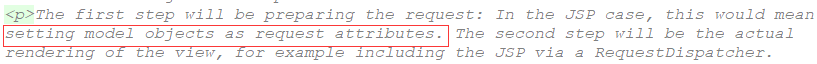


他们都实现了HanderAdapter接口，该接口可以让DispatcherServlet无限的扩展



DispatcherServlet可以通过这个接口接触所有的已经注册的handler，意味着不需要为具体的某一个handler写具体的代码。根据不同的handler来获取不同的HandlerAdapter，HandlerAdapter提供了统一的接口，具体实现各部相同，典型的适配器模式。

**initViewResolvers**会从ApplicationContext中获取注册过的ViewResolver，如果没有找到，那么就使用InternalResourceViewResolver。之前一直存在的一个问题就是，Model域和request域是否相通，这边视图渲染的时候文档给了这样一句话：



意味着对于JSP来说，model会作为属性存放到request域中。

而对于Thymeleaf来说，我并没有看到具体的将model作为属性放入request域中的操作，这边猜测是两个单独的吧。

1. 父子容器

* 在web容器启动的时候，会先读取web.xml文件中context-parm节点和listener节点，接着创建一个ServletContext作为全局共享，Spring ioc容器就存储在这里面
* 接着创建listener节点中的ContextLoaderListener，通过context-parm中的内容来创建一个Spring IOC。
* 接着Servlet会进行初始化，在初始化的过程中，会再创建一个spring mvc的ioc容器，并且设置spring ioc容器为父类容器。
* 子类容器可以访问父类容器，但是父类容器不可访问子类容器