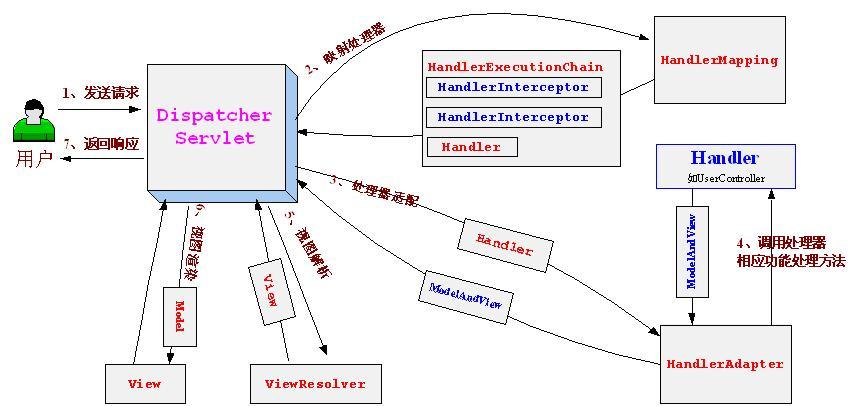
注意Spring配置文件中

|  |  |  |
| --- | --- | --- |
| 类型 | HandlerMapping个数 | HandlerAdapter个数 |
| RequestMappingUrlHandlerMapping | 2个、  BeanNameUrlHandlerMapping、RequestMappingHandlerMapping、 | RequestMappingHandlerAdapter |
| BeanNameUrlHandlerMapping | 2个，  BeanNameUrlHandlerMapping、  RequestMappingHandlerMapping | 1个、  RequestMappingHandlerAdapter |
|  |  |  |
|  |  |  |





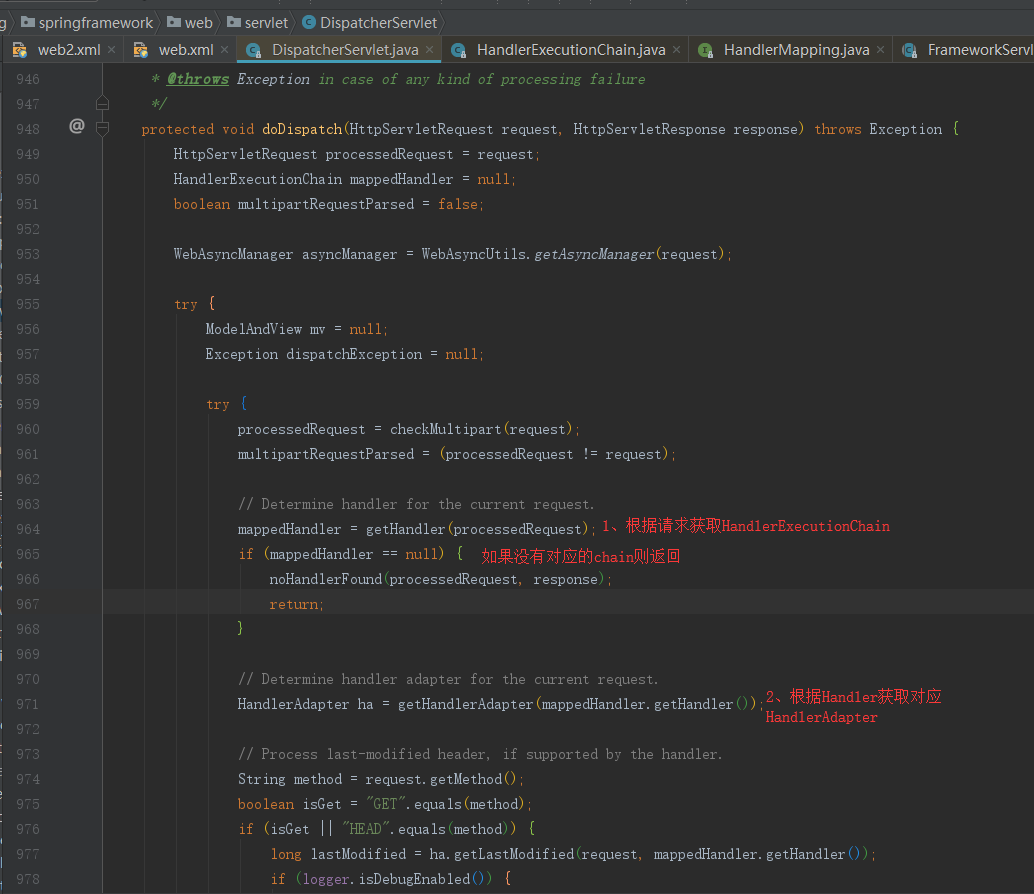
上图各个步骤对应到的代码执行步骤，所有调用都在DispatcherServlet的doDispatch方法中发起。

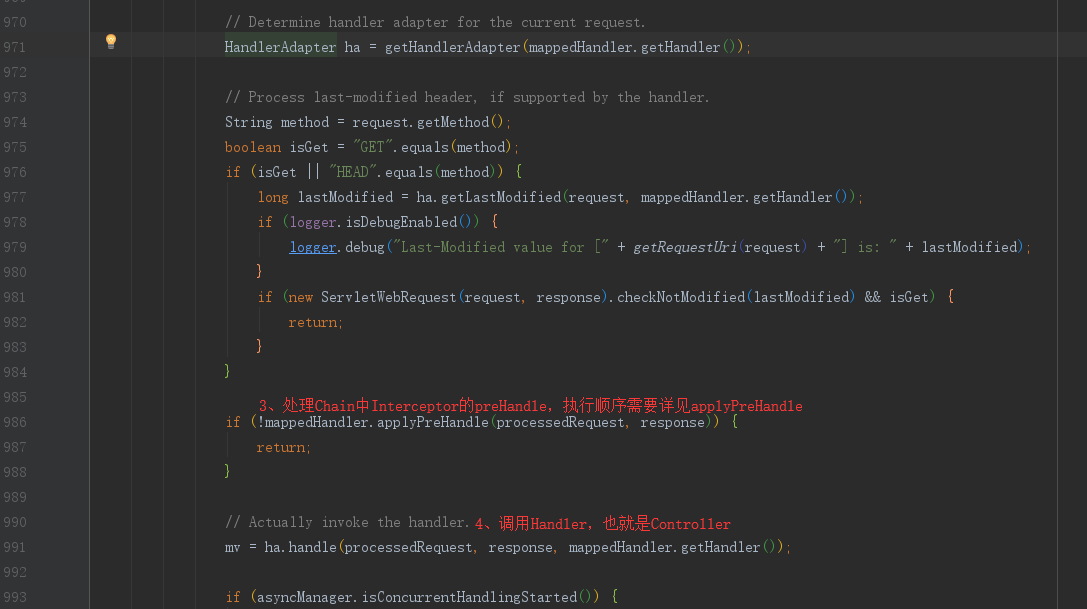
1、发起http请求，当请求到达DispatchServlet时，如果配置了filter，首先会执行filter；接下来执行DispatchServlet的doService方法，同时将调用doDispatch方法（这一步最重要）；

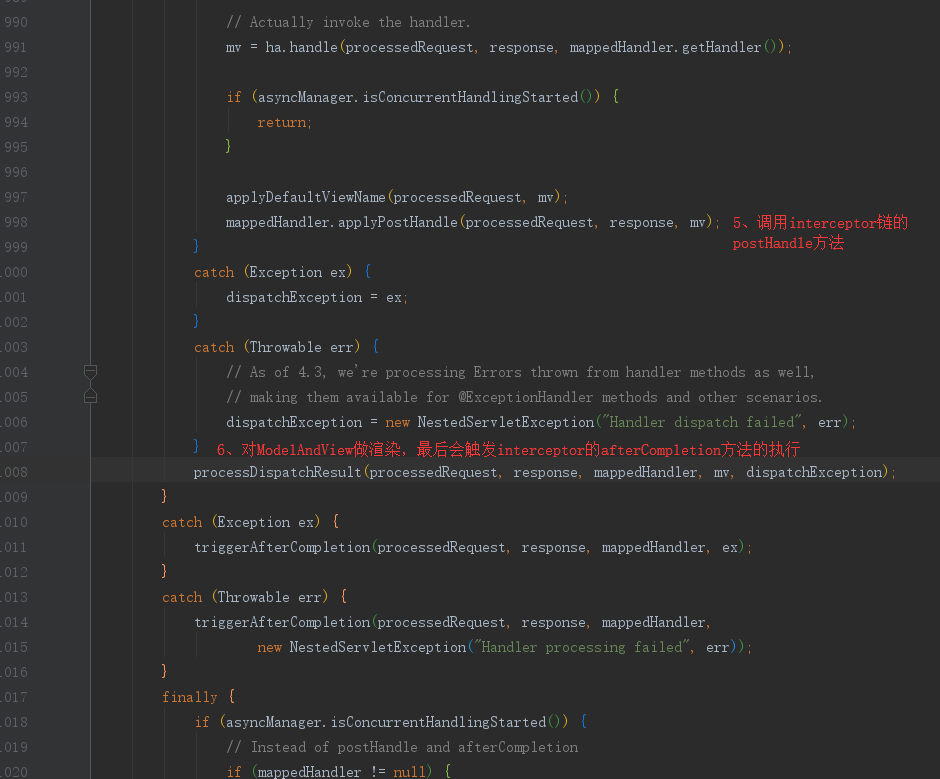
2、执行doDispatch时，首先将通过request获取对应的HandlerMapping，返回的HandlerMapping是一个HandlerExecutionChain，一个HandlerExecutionChain包含一个Handler（也就是对应的Controller）和一个Interceptor链表，链表中所有的拦截器的preHandle都要先于Controller执行（可以进行编码拦截），postHandle在Controller执行之后执行（可以修改ModelAndView），afterCompletion是在返回ModelAndView给DispatchServlet 之前执行（可以判断是否发生异常）；所有的执行都是在doDispatch方法中执行。

3、HandlerExecutionChain的执行顺序：preHandle按照Interceptor的定义顺序执行，postHandle的执行顺序与preHandle相反；具体可以参考代码applyPreHandle、applyPostHandle方法，遍历interceptor；

**代码分析**







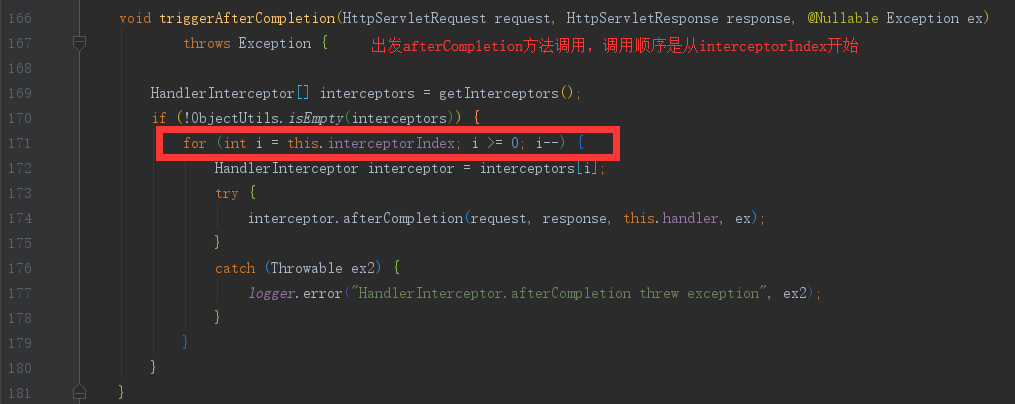
**拦截器的preHandle方法调用，此处解释了为啥preHandle的执行顺序为啥和interceptor的定义顺序一致**



**拦截器的postHandle方法调用，此处解释了为啥postHandle的执行顺序为啥和preHandle的执行顺序相反**



**触发拦截器的afterCompletion方法执行，调用顺序和preHandle的执行情况有关。AfterCompletion将在ModelAndView返回给前端之前执行**



**渲染包括两部分，视图解析和视图渲染**

