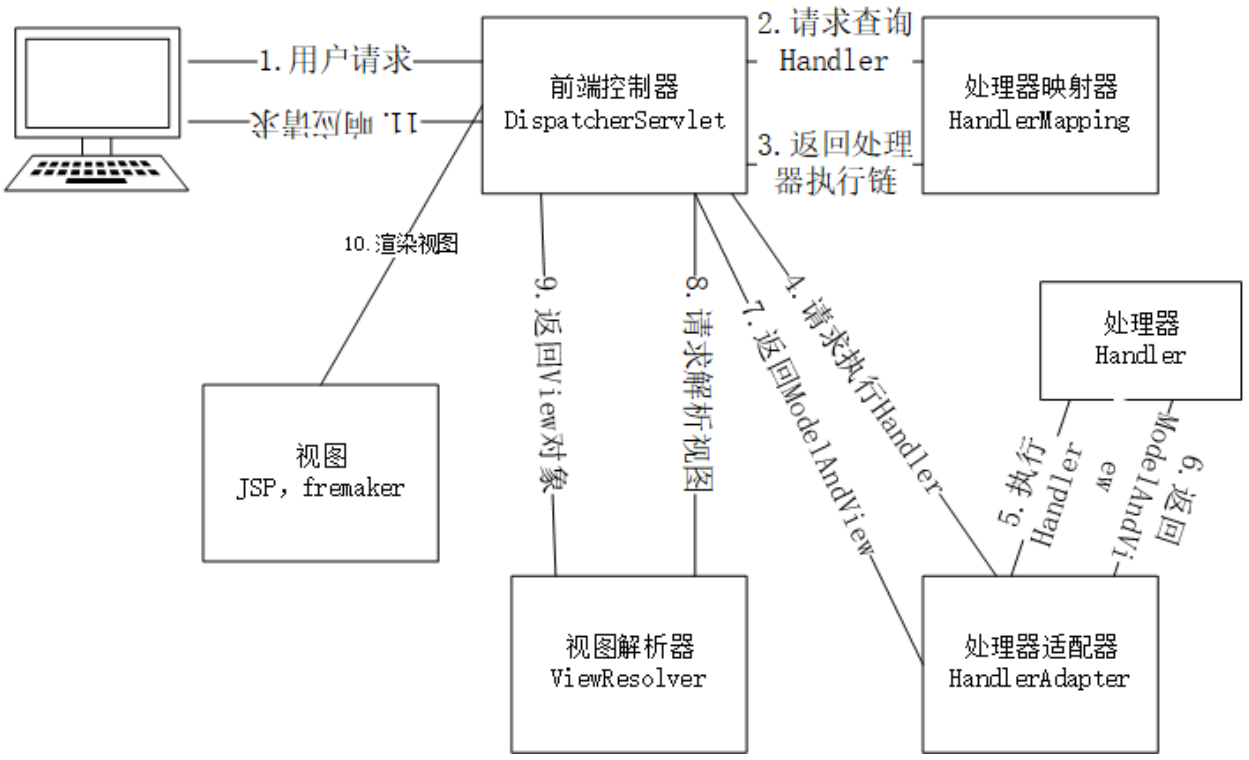
SpringMVC全名Spring Web MVC，基于Java的实现MVC设计模型的请求驱动类型的轻量级web框架

Spring MVC是对servlet的封装，简化了servlet的开发

主要作用是：1.接收请求；2.返回响应，跳转页面

主要流程如下：



流程说明

第一步：用户发送请求至前端控制器DispatcherServlet

第二步： DispatcherServlet收到请求调用HandlerMapping处理器映射器

第三步：处理器映射器根据请求Url找到具体的Handler（后端控制器），生成处理器对象及处理器拦截

器(如果 有则生成)一并返回DispatcherServlet

第四步： DispatcherServlet调用HandlerAdapter处理器适配器去调用Handler

第五步：处理器适配器执行Handler

第六步： Handler执行完成给处理器适配器返回ModelAndView

第七步：处理器适配器向前端控制器返回 ModelAndView， ModelAndView 是SpringMVC 框架的一个

底层对 象，包括 Model 和 View

第八步：前端控制器请求视图解析器去进行视图解析，根据逻辑视图名来解析真正的视图。

第九步：视图解析器向前端控制器返回View
第十步：前端控制器进行视图渲染，就是将模型数据（在 ModelAndView 对象中）填充到 request 域
第十一步：前端控制器向用户响应结果

Spring MVC 九大组件

HandlerMapping（处理器映射器）

HandlerMapping 是用来查找 Handler 的，也就是处理器，具体的表现形式可以是类，也可以是
方法。比如，标注了@RequestMapping的每个方法都可以看成是一个Handler。
Handler负责具
体实际的请求处理，在请求到达后， HandlerMapping 的作用便是找到请求相应的处理器 Handler 和 Interceptor.

HandlerAdapter（处理器适配器）

HandlerAdapter 是一个适配器。因为 Spring MVC 中 Handler 可以是任意形式的，只要能处理请
求即可。但是把请求交给 Servlet 的时候，由于 Servlet 的方法结构都是 doService(HttpServletRequest req,HttpServletResponse resp)形式的，要让固定的 Servlet 处理
方法调用 Handler 来进行处理，便是 HandlerAdapter 的职责。

HandlerExceptionResolver

HandlerExceptionResolver 用于处理 Handler 产生的异常情况。它的作用是根据异常设置
ModelAndView，之后交给渲染方法进行渲染，渲染方法会将 ModelAndView 渲染成页面。

ViewResolver

ViewResolver即视图解析器，用于将String类型的视图名和Locale解析为View类型的视图，只有一
个resolveViewName()方法。从方法的定义可以看出， Controller层返回的String类型视图名
viewName 最终会在这里被解析成为View。 View是用来渲染页面的，也就是说，它会将程序返回
的参数和数据填入模板中，生成html文件。 ViewResolver 在这个过程主要完成两件事情：

ViewResolver 找到渲染所用的模板（第一件大事）和所用的技术（第二件大事，其实也就是找到
视图的类型，如JSP）并填入参数。默认情况下， Spring MVC会自动为我们配置一个InternalResourceViewResolver,是针对 JSP 类型视图的。

RequestToViewNameTranslator

RequestToViewNameTranslator 组件的作用是从请求中获取 ViewName.因为ViewResolver 根据
ViewName 查找 View，但有的 Handler 处理完成之后,没有设置 View，也没有设置ViewName，
便要通过这个组件从请求中查找 ViewName。

LocaleResolver

ViewResolver 组件的 resolveViewName 方法需要两个参数，一个是视图名，一个是Locale。
LocaleResolver 用于从请求中解析出 Locale，比如中国 Locale 是 zh-CN，用来表示一个区域。这
个组件也是 i18n 的基础。

ThemeResolver

ThemeResolver 组件是用来解析主题的。主题是样式、图片及它们所形成的显示效果的集合。
Spring MVC 中一套主题对应一个 properties文件，里面存放着与当前主题相关的所有资源，如图
片、 CSS样式等。创建主题非常简单，只需准备好资源，然后新建一个“主题名.properties”并将资
源设置进去，放在classpath下，之后便可以在页面中使用了。 SpringMVC中与主题相关的类有
ThemeResolver、 ThemeSource和Theme。 ThemeResolver负责从请求中解析出主题名，
ThemeSource根据主题名找到具体的主题，其抽象也就是Theme，可以通过Theme来获取主题和
具体的资源。

MultipartResolver

MultipartResolver 用于上传请求，通过将普通的请求包装成MultipartHttpServletRequest 来实
现。 MultipartHttpServletRequest 可以通过 getFile() 方法 直接获得文件。如果上传多个文件，还

可以调用 getFileMap()方法得到Map<FileName， File>这样的结构，
MultipartResolver 的作用就
是封装普通的请求，使其拥有文件上传的功能。

FlashMapManager

FlashMap 用于重定向时的参数传递，比如在处理用户订单时候，为了避免重复提交，可以处理完
post请求之后重定向到一个get请求，这个get请求可以用来显示订单详情之类的信息。这样做虽然
可以规避用户重新提交订单的问题，但是在这个页面上要显示订单的信息，这些数据从哪里来获得
呢？因为重定向时么有传递参数这一功能的，如果不想把参数写进URL（不推荐），那么就可以通
过FlashMap来传递。只需要在重定向之前将要传递的数据写入请求（可以通过
ServletRequestAttributes.getRequest()方法获得）的属性
OUTPUT_FLASH_MAP_ATTRIBUTE
中，这样在重定向之后的Handler中Spring就会自动将其设置到Model中，在显示订单信息的页面
上就可以直接从Model中获取数据。 FlashMapManager 就是用来管理 FalshMap 的。