**Spring MVC分析**

**1. 跟踪spring MVC的请求**

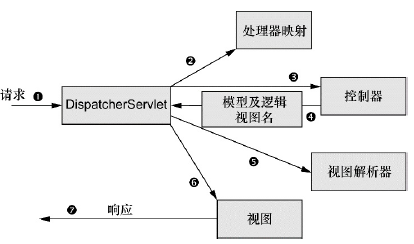


图1 spring MVC请求处理流程

①请求离开浏览器，带着用户所请求内容的信息，包含请求的URL（和用户提交的表单信息），到达Spring的dispatcherServlet。

②dispatcherServlet以会查询一个或者多个handler mapping来确定请求的下一站在哪里。Handler mapping会根据请求所携带的URL信息进行决策。

③，DispatcherServlet会将请求发送给选中的控制器。到了控制器，请求会卸下其负载（用户提交的信息）并耐心等待控制器处理这些信息。（实际上，设计良好的控制器本身只处理很少甚至不处理工作，而是将业务逻辑委托给一个或多个服务对象进行处理）

④（控制器完成逻辑处理后，将信息存储在model中，并指定渲染模版view）将model和view（通常只指定视图名称）发送回dispatcherServlet。

⑤DispatcherServlet将会使用视图解析器（view resolver） 来将逻辑视图名匹配为一个特定的视图实现。

⑥它的最后一站是视图的实现（可能是jsp），在这里它交付模型数据（model）。

⑦视图（view）将使用模型数据（model）渲染输出。

**2. spring MVC的总体实现**

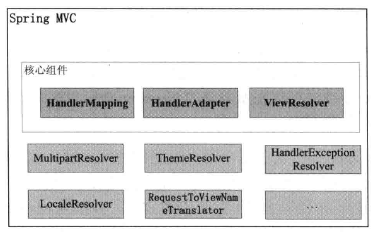


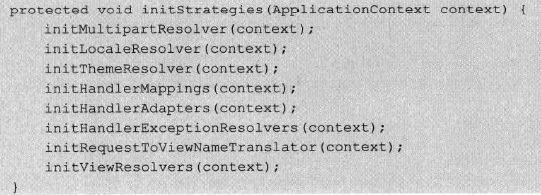
图2 spring mvc的组件图

在spring mvc框架中，有三个组件是用户必须定义和拓展的：定义URL映射规则、实现业务逻辑的Handler实例对象、渲染模版资源。而连接Handler实例对象和模版渲染的纽带就是Model模型了。

要搞清楚Spring MVC如何工作，主要看DispatcherServlet的代码。其中，DispatcherServlet类继承了httpServlet。

**2.1 DispatcherServlet的初始化工作**

DispatcherServlet继承了httpServlet，在servlet的init方法调用时DispatcherServlet执行Spring MVC的初始化工作。主要初始化图2中springMVC的8个组件。



其中：

* initHandlerMappings：用于定义用户设置的请求映射关系。将用户请求的URL映射成一个个Handler实例。
* initHandlerAdapters：用于根据Handler的类型定义不同的处理规则。在HandlerMapping返回处理请求的Controller实例后，需要一个帮助定位具体请求方法的处理类，这个类就是HandlerAdapter，HandlerAdapter是处理器适配器，Spring MVC通过HandlerAdapter来实际调用处理函数。
* initViewResolvers：用于将view解析成页面，在ViewResolvers中可以设置多个解析策略。

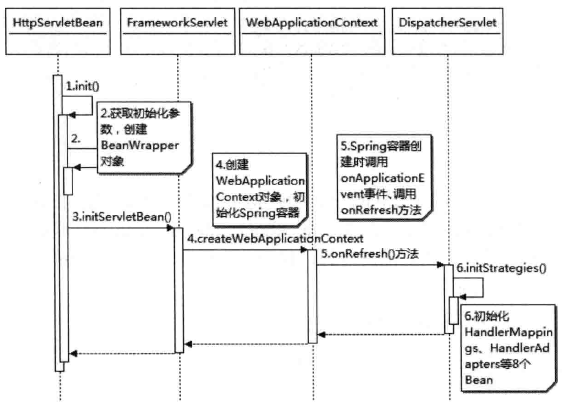


图3 SpringMVC初始化时序图

**2.2 control设计**

Spring MVC的control主要有HandlerMapping和HandlerAdapter两个组件提供。HandlerMapping负责映射用户的URL和对应的处理类，HandlerMapping并没有规定这个URL与应用的处理类如何映射，在HandlerMapping接口中只定义了一个URL必须返回一个有HandlerExceptionChain代表的处理链，我们可以在这个处理链中添加任意的handlerAdapters实例来处理这个URL对应的请求。

**2.2.1 HandlerMapping**

HandlerMapping（默认使用BeanNameUrlHandlerMapping，根据bean的name属性映射到URL中）的初始化工作完成两个最重要的工作就是将URL与Handler的对应关系保存在handlerMap集合中，并将所有的interceptors对象保存在adaptedInterceptors数组中，等请求到来时执行所有的adaptedInterceptors数组中的interceptor对象。所有的interceptor对象必须实现handlerInterceptor接口。

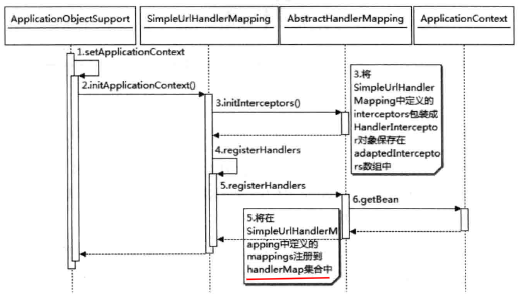


图4 handlerMapping初始化时序图

**2.2.2 HandlerAdapter**

HandlerAdapter是处理适配器，它的作用一句话概括就是调用具体的方法对用户发来的请求进行处理。当handlerMapping获取到执行请求的controller时，DispatcherServlet会根据controller对应的controller类型来调用相应的HandlerAdapter进行处理。

HandlerAdapter处理的流程大致分三个步骤：

①DispatcherServlet会根据配置文件信息注册HandlerAdapter，若没有配置，会注册默认的HandlerAdapter。存储到它的属性HandlerAdapters集合中。

②DispatcherServlet会根据handlerMapping传过来的controller与已经注册好了的HandlerAdapter一一匹配，看哪一种HandlerAdapter是支持该controller类型的。

③（找到的）HandlerAdapter会调用自己的handle方法，handle方法运用Java的反射机制执行controller的具体方法来获得ModelAndView。

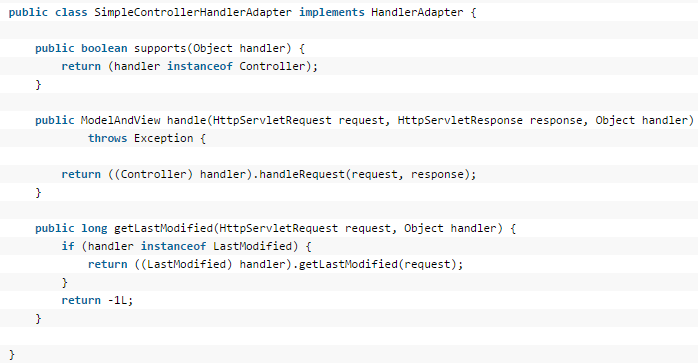


图5 默认的adapter源码示例

**2.2.3 Control的调用逻辑**

**2.3 model设计**

**2.4 view设计**