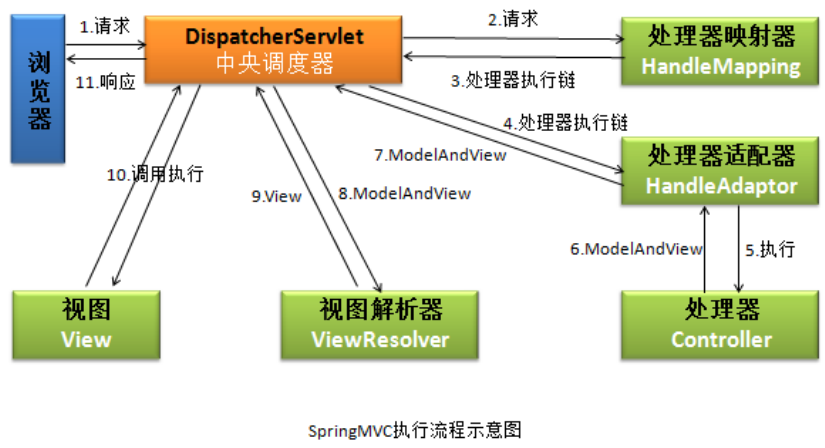
1. **执行流程简单分析**



1. 浏览器提交请求到中央调度器
2. 中央调度器直接将请求转给处理器映射器。
3. 处理器映射器会根据请求，找到处理该请求的处理器，并将其***封装***为处理器执行链后返回给中央调度器。
4. 中央调度器根据处理器执行链中的处理器，找到能够执行该处理器的处理器适配器。
5. 处理器适配器调用执行处理器。
6. ***处理器将处理结果及要跳转的视图封装到一个对象 ModelAndView 中，并将其返回给处理器适配器。***
7. 处理器适配器直接将结果返回给中央调度器。
8. 中央调度器调用视图解析器，将 ModelAndView 中的视图名称封装为视图对象。
9. 视图解析器将封装了的视图对象返回给中央调度器
10. ***中央调度器调用视图对象，让其自己进行渲染，即进行数据填充，形成响应对象。***
11. 中央调度器响应浏览器。
12. **API简要说明**
13. DispatcherServlet

中央调度器，也称为前端控制器，在MVC架构模式中充当控制器C，DispatcherServlet是整个流程的控制中心，由它调用诸如处理器映射器、处理器适配器、视图解析器等其它组件处理用户请求。中央调度器的存在降低了组件之间的耦合度。

1. HandlerMapping

处理器映射器，负责根据用户请求找到相应的将要执行的Handler，即处理器。即用于完成将用户请求映射为要处理该请求的处理器，并将处理器封装为处理器执行链传给中央调度器。

1. HandlAdapter

处理器适配器，通过HandlerAdapter对处理器进行执行，这是适配器模的应用，通过扩展适配器可以对更多类型的处理器进行执行。中央调度器会根据不同的处理器自动为处理器选择适配器，以执行处理器。

1. Handler

处理器，也称为后端控制器，在DispatcherServlet的控制下Handler调用Service层对具体的用户请求进行处理。由于Handler涉及到具体的用户业务请求，所以一般情况下需要程序员根据业务需求自己开发Handler。

1. ViewResolver

视图解析器，负责将处理结果生成View视图，ViewResolver首先将逻辑视图名解析为物理视图名，即具体的页面地址，再生成View视图对象。最后将处理结果通过页面形式展示给用户。

SpringMVC框架提供了很多的View视图类型，包括：JstlView、RedirectView等。一般情况下需要通过页面标签或页面模版技术将模型数据通过页面展示给用户，需要由程序员根据业务需求开发具体的页面。

1. **DispatcherServlet的默认配置**

在流程介绍中所述的重要的处理器映射器、处理器适配器、视图解析器等，都在哪里，不用做配置吗？

当然不是！这些内容均在DispatcherServlet的默认配置DispatcherServlet.properties文件中被定义。这个文件与DispatcherServlet类在一个包下，而且是当Spring配置文件中没有指定配置时使用的默认情况。

从上面的内容可知，这些组件的默认配置如下：

