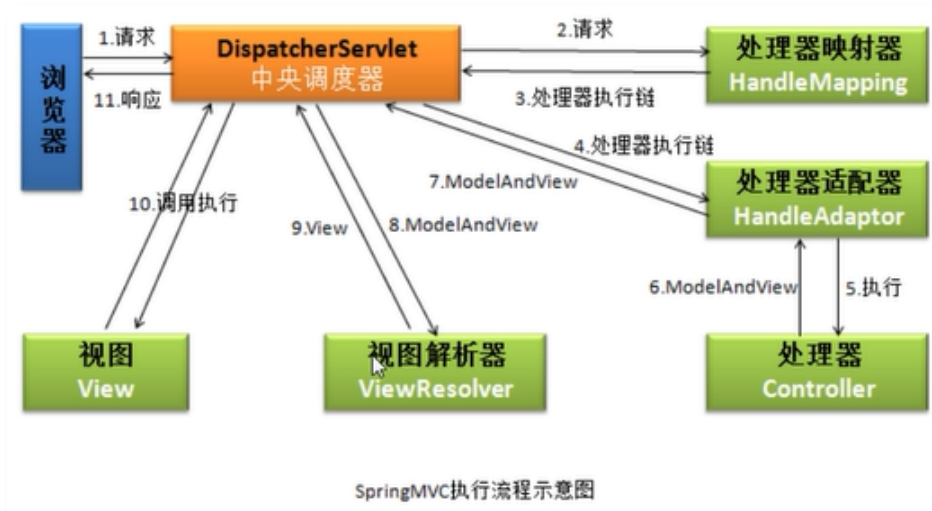


# SpringMVC底层执行流程

## 1.5 SpringMVC 执行流程（理解）

### 1.5.1 流程图



## SpringMVC接收请求到处理完成的过程

根据图中（执行代码都是在DispatcherServlet类中的）：

1、用户发起请求，请求some.do资源

--在DispatcherServlet中的代码如下：

这个执行流程主要是在这个方法中进行

```
protected void doDispatch(HttpServletRequest request, HttpServletResponse response) throws Exception
```

2、DispatcherServlet接收请求some.do，把请求交给处理器映射器HandleMapping

处理器映射器：是SpringMVC框架中的一种对象，框架中实现了HandlerMapping接口的类都称为映射器

作用：获取请求的URL，根据请求中的信息从SpringMVC容器中获取处理该请求的处理器对象

相当于：MyController controller = ctx.getBean("xxxsome.do");

HandlerMapping接口接口中有一个getHandler方法，返回值是HandlerExecutionChain类型的

其中有很多的实现类，原因是之前是没有注解来创建处理器的，需要使用各种不同的接口才能作为控制器使用

不同的接口实现的控制器被找到的方式不一致，所以我们就需要为不同的接口创建不同的映射器

现在我们使用注解的方式，使用的映射器是RequestMappingHandlerMapping实现类

然后把获取的处理器对象返回给DispatcherServlet对象

DispatcherServlet中的一个HandlerExecutionChain类来保存返回的处理器对象，这个类中有一个Object handler属性保存控制器对象，也有一个HandlerInterceptor类型的数组和List<HandlerInterceptor>集合保存拦截器对象。

通常将这个类叫做处理器执行链：HandlerExecutionChain对象，保存了返回的控制器对象和所有的拦截器对象

--在DispatcherServlet中的代码如下：

```
HandlerExecutionChain mappedHandler = null;
mappedHandler = this.getHandler(processedRequest);
```

3、DispatcherServlet把2中的执行链HandlerExecutionChain对象(保存了处理器对象)交给处理器适配器对象(多个)

处理器适配器: SpringMVC框架中实现了HandlerAdapter接口的类

作用: 处理器适配器获取执行链中的控制器对象, 然后执行其中的处理器方法(即调用doSome方法)得到返回值 ModelAndView, 框架

HandlerAdapter接口的实现类也有很多, 因为不同的映射器返回的执行链需要其对应的适配器来处理

--在DispatcherServlet中的代码如下:

```
HandlerAdapter ha = this.getHandlerAdapter(mappedHandler.getHandler());  
mv = ha.handle(processedRequest, response, mappedHandler.getHandler());
```

4、DispatcherServlet把3中获取的ModelAndView对象交给视图解析器对象

视图解析器: SpringMVC中的对象, 实现的是ViewResolver接口

作用: 从ModelAndView对象中获取到的视图的逻辑名称和其配置好的前缀与后缀组成完整的视图路径, 将视图资源的路径字符串作为创建一个View对象的参数, 获得一个该保存该视图资源的View对象

View是一个接口, 表示一个视图, 框架中jsp、html文件并不是以他们的路径的String表示, 而是用View和它的实现类来表示视图。

View接口实现类很多, 其中有一个RedirectView实现类, 我们new实现类的对象时, 把一个视图资源的路径作为该类构造方法的参数, 创建的这个View对象就可以代表这个视图了。

比如说我们使用的是: mv.setViewName("show"); //这个show.jsp框架内部转成View对象  
但也可以这样: mv.setView(new RedirectView("/view/show.jsp")); //这里我们是手动转的

而有一个实现类InternalResourceView: 是视图类, 常用, 表示jsp文件。

视图解析器会创建InternalResourceView类对象, 该对象有一个属性url=/view/show.jsp

5、DispatcherServlet把4中返回的View对象获取到, 调用View类中的方法, 把ModelAndView中的数据放到request作用域中, 然后执行对象视图的forward方法。请求结束

## 1.5.2 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。