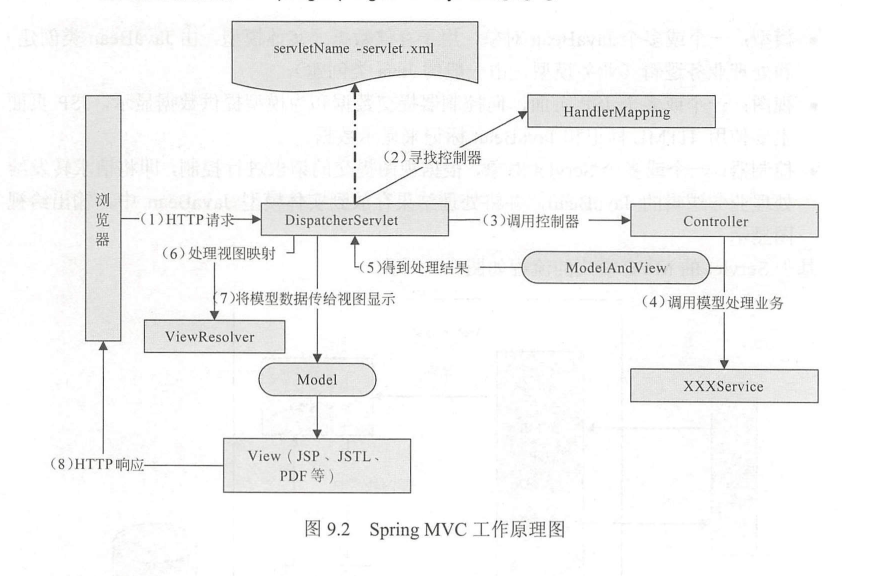
# 认识MVC模式

MVC是Model、View、Controller的缩写，它们的所包含的含义分别如下：

1. Model（模型层），用于存储数据以及处理用户请求的业务逻辑
2. View（视图层），用于显示数据与提交数据
3. Controller（控制层），用于根据请求来完成页面跳转

**Spring MVC工作原理：**

Spring MVC框架主要由DispatcherServlet、处理器映射、控制器、视图解析器、视图组 成，其工作原理如下所示：



可以从工作原理总结出Spring MVC的工作流程，如下：

1. 将客户端请求提交到DispatcherServlet；
2. 由DispatcherServlet控制器寻找一个或多个HandlerMapping，找到处理请求的 Controller；
3. DispatcherServlet将请求提交到Controller；
4. Controller调用业务逻辑处理后返回ModelAndView
5. DispatcherServlet寻找一个或多个ViewResolver视图解析器，找到ModeAndView指 定的视图；
6. 视图负责将结果显示到客户端。

**Spring MVC的主要接口：**

从Spring MVC的工作原理和工作流程上可以得知，Spring MVC有4个主要接口，即

DispatcherServlet、HandlerMapping、Controller和ViewResolver。

1）Spring MVC所有的请求都经过DispatcherServlet来同一分发，在DispatcheServlet将 请求分发给Controller之前需要借助Spring MVC提供的HandlerMapping定位到具体 的Controller。

2）HandlerMapping接口负责完成客户请求到Controller映射。

3）Controller接口将处理用户请求，这和Java Servlet扮演的角色是一致的。一旦Controller 处理完用户请求，将返回ModelAndView对象给DispatcherServlet前端控制器， ModelAndView中包含了模型（Model）和视图（View）。从宏观角度考虑， DispatcherServlet是整个Web应用的控制器；从微观角度考虑，Controller是单个Http 请求处理过程中的控制器，而ModelAndView是Http请求过程中返回的模型（Model） 和视图（View）。

4）ViewResolver接口（视图解析器）在Web应用中负责查找View对象，从而将相应 结果渲染给用户。

**Spring MVC开发环境的搭建：**

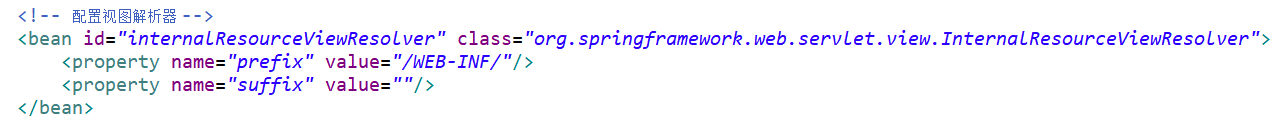
Spring MVC开发环境的搭建分为3步：

1. 导入相关JAR包
2. 创建SpringMVC全局文件
3. 配置前端控制器

**视图解析器：**

如果将jsp页面文件放在了WEB-INF目录外，则必使用视图解析器；而如果将jsp页面 等放在了WEB-INF目录中，则推荐使用视图解析器。

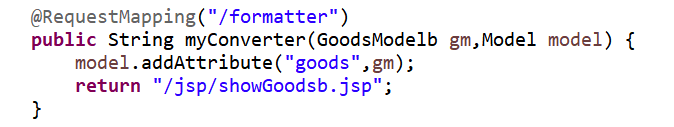
视图解析器的配置是在Spring MVC配置文件中完成的，如下：



在视图解析器中，存在两个属性，

·prefix：表示前缀。

·suffix：表示后缀。推荐值空，即value=””

视图解析的主要作用是为了使请求处理方法中带有逻辑视图的字符串的编写更加方 便。如当在Spring MVC配置文件中配置了如上的视图解析器时，在某个控制类中的存 在某个请求处理方法，如下：

虽然返回的是”/jsp/showGoodb.jsp”，但是由于配置了视图解析器，所以实际上代表 的是”/WEB/jsp/showGoodsb.jsp”，即跳转到WEB-INF/jsp目录中的showGoodsb.jsp 页面。