视图解析器ViewResolver接口负责将处理结果生成View视图。常用的有四种

1. **InternalResourceViewResolver视图解析器**

该视图解析器用于***完成对当前Web应用内部资源的封装与跳转***。而对于内部资源的查找规则是，将ModelAndView中指定的视图名称与为视图解析器配置的前辍与后辍相结合的方式，拼接成一个Web应用内部资源路径。拼接规则是：前辍+视图名称+后辍。



InternalResourceView解析器会把处理器方法返回的模型属性都存放到对应的request中，然后将请求转发到目标URL。

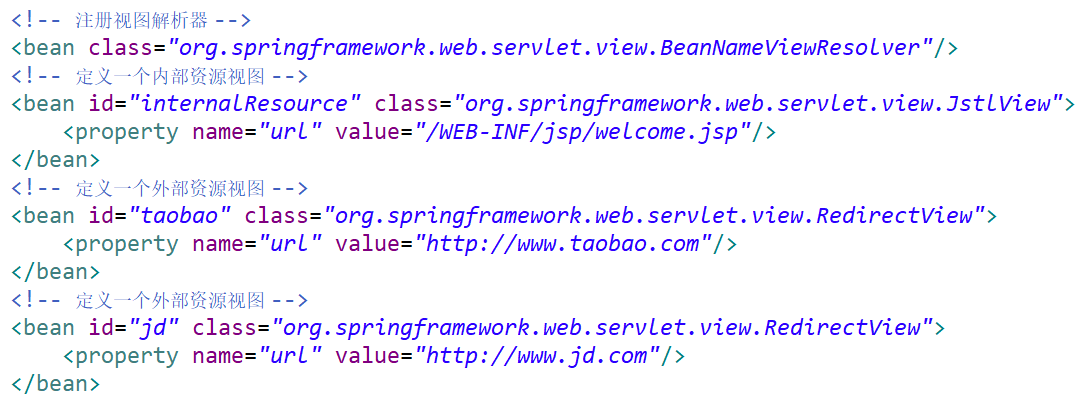
1. **BeanNameViewResolver视图解析器**

InternalResourceViewResolver解析器存在两个问题，使其使用很不灵活：

* 只可以完成将内部资源封装后的跳转。但无法转向外部资源，如外部网页。
* 对于内部资源的定义，也只能定义一种格式的资源：存放于同一目录的同一文件类型的资源文件。

BeanNameViewResolver视图解析器，顾名思义就是***将资源封装为“Spring容器中注册的Bean实例”***，ModelAndView通过设置视图名称为该Bean的id属性值来完成对该资源的访问。所以在springmvc.xml中，可以定义多个View视图Bean，让处理器中ModelAndView通过对这些Bean的id的引用来完成向View中封装资源的跳转。

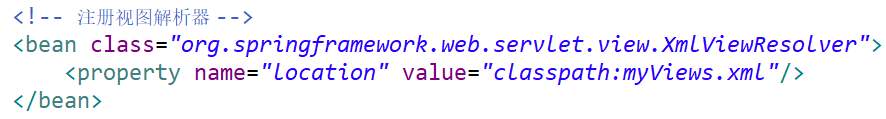
* RedirectView：定义外部资源视图对象
* JstlView：定义内部资源视图对象



在处理器中，只要将ModelAndView的视图名称指定为Spring容器中定义好的Bean的id，则可完成对该资源的访问。

1. **XmlViewResolver视图解析器**

BeanNameViewResolver视图解析器存在一些问题，当需要定义的View视图对象很多时，就会使springmvc.xml文件变得很大，很臃肿，不便于管理。所以可以将这些View视图对象专门抽取出来，单独定义为一个xml文件。这时就需要使用XmlViewResolver解析器了。



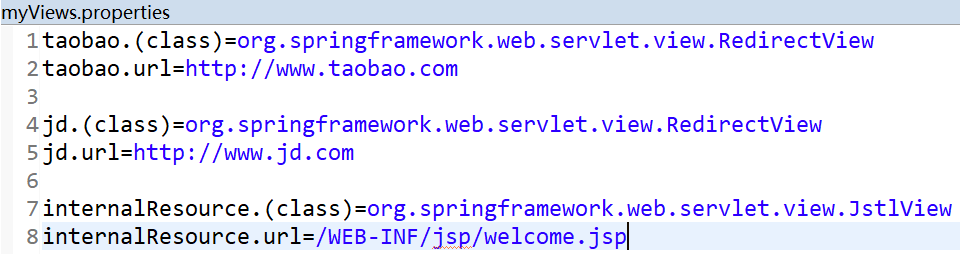
1. **ResourceBundleViewResolver视图解析器**

对于View视图对象的注册，除了使用xml文件外，也可以在properties文件中进行注册。

该属性文件需要定义在类路径下，即src下。而对于属性文件的写法，是有格式要求的：

**资源名称.(class)=封装资源的View全限定性类名**

**资源名称.url=资源路径**



1. **视图解析器的优先级**

视图解析器有一个order属性，专门用于设置多个视图解析器的优先级。数字越小，优先级越高。数字相同，先注册的优先级高。一般不为InternalResourceViewResolver解析器指定优先级，即让其优先级是最低的。

