

这个就是视图解析器的一个例子

下面详细介绍

1.视图解析器的工作流程

①．调用目标方法，会返回一个String类型或者View类型或者ModelMap类型或者ModelAndView类型，springmvc都会给封装成一个ModelAndView对象

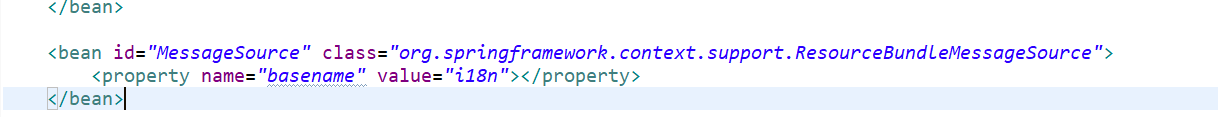
②SpringMvc借助借助视图解析器（ViewResolver，）得到最终的视图对象（view，注意默认是InternalResourceView），最终调用view的render方法得到响应结果，最终的视图可以是JSP，也可能是Excel、JFreeChart等各种表现形式的视图

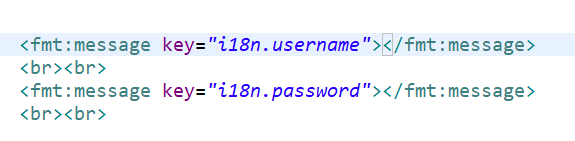
③上面实例中的InternalResourceView是来搞Jsp的，InternalResourceResolver是支持InternalResourceView的解析器，这个主要是搞在同一个web应用下，通过转发的那个结果

2.第二个视图对象（JSTLView）

1）配置同上，，，只要加入了JSTL的jar包，这个view就变成了JSTLView了

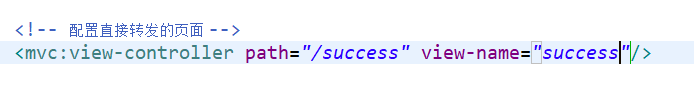
2）如果想要用JSTL中fmt标签，则需要在SpringMvc的配置文件中配置国际化资源文件





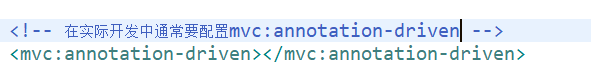


3）想直接转发到页面，不想经过handler，可以如下配置



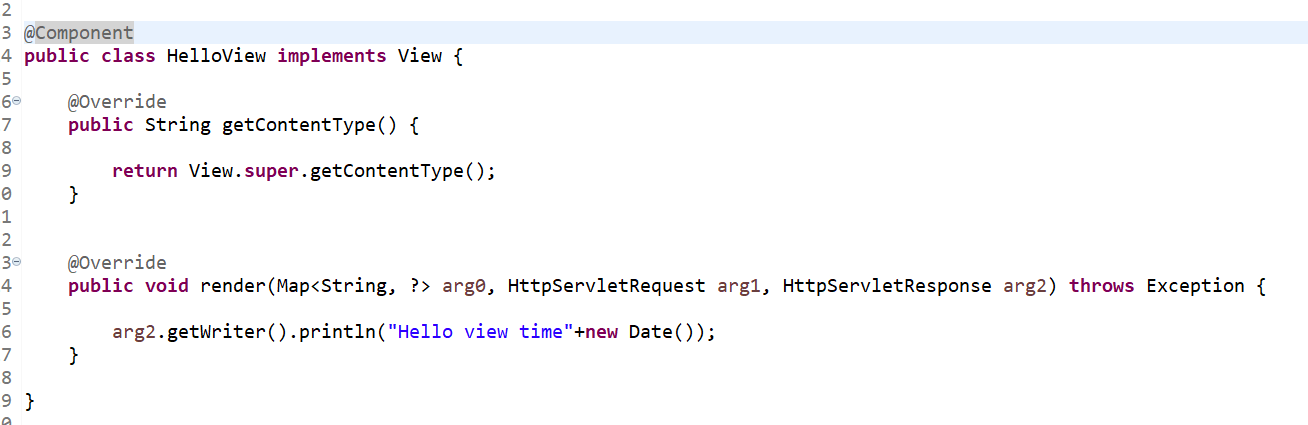
但是，你会发现配置了这个之后，是可以不经过handler直接到页面了，但是会导致以前的以前所写的经过handler的，都不管用了

解决方式（原理暂时不详，但是后面有原理）

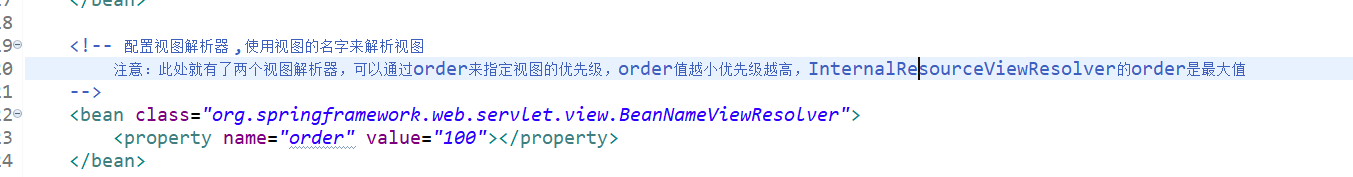


2.自定义视图：springmvc要整合其他视图（Excel，JFreechar）都是靠自定义视图来实现的

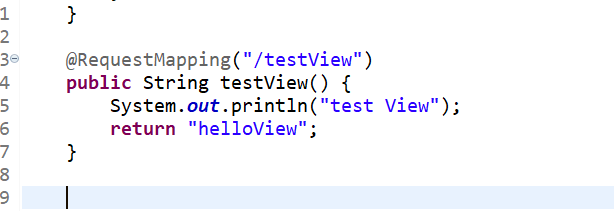
①．写一个类，实现View接口，实现其中的render 方法还有getContentType（现在的接口里能实现方法了，所以这玩意有了简单实现），render方法里来表明该视图干点什么事



②．在spring的配置文件中配置BeanNameViewResolver视图解析器，该视图解析器使用名字解析视图，然后在刚刚所写的视图解析器上加上@Component



③然后，写一个jsp页面，给一个超链接，然后在controller里写一个方法，写上@RequestMapping对应这个超链接，返回值为刚刚写的View实现类的类名的第一个字母小写



3.关于重定向

一般情况下，控制器方法返回字符串类型的值会被当成逻辑视图名处理

如果返回的字符串中带forward：或redirect：前缀时，SpringMVC会对他们进行特殊处理：将forward和redirect：当成指示符，其后的字符串作为URL来处理

forward执行转发操作

redirect执行重定向操作（emmm，转发重定向什么的，忘得有点快啊）

