视图与视图解析器

1. 配置视图解析器

**配置视图解析器 ，如何把Handler的方法返回值解析成一个物理视图**

<bean class=*"org.springframework.web.servlet.view.****InternalResourceViewResolver****"*>

<!-- 需要配置两个属性 -->

<!-- **前缀** -->

<property name=*"prefix"* value=*"/WEB-INF/views/"*></property>

<!-- **后缀** -->

<property name=*"suffix"* value=*".jsp"*></property>

</bean>

**handler方法返回值可以通过视图解析器，解析成实际的物理视图**

视图解析器会把返回值解析成

**前缀+返回值+后缀**

前缀后缀在配置文件中配置

如：/WEB-INF/views/success.jsp

**InternalResourceViewResolver这个视图解析器时专门为转发使用的**

* 1. 配置直接转发的视图

path映射到name由视图解析器解析后的视图，进行直接转发，意思是，请求path的URL就直接转发到name被视图解析器解析后的url，无需经过handler

<mvc:view-controller path=*"/success"* view-name=*"success"*/>

在这里访问/success 就会访问到*/WEB-INF/views/*success.jsp

但是会影响到其他映射到handler的请求

解决方案：

配置

<mvc:annotation-driven></mvc:annotation-driven>

* 1. 自定义视图
     + 1. 配置

<bean class=*"org.springframework.web.servlet.view.BeanNameViewResolver"*>

<property name=*"order"* value=*"100"*></property>

</bean>

自定义视图解析器：***BeanNameViewResolver***

要为其设置一个优先级order

* + - 1. 写一个自定义视图类

//默认名字为类首字母小写helloView

@Component

**public** **class** HelloView **implements** View {

// 返回类型

**public** String getContentType() {

**return** "text/html";

}

// 渲染视图

**public** **void** render(Map<String, ?> arg0, HttpServletRequest arg1, HttpServletResponse arg2) **throws** Exception {

arg2.getWriter().println("hello view" + **new** Date());

}

}

* + - 1. 使用

@RequestMapping("/testView")

**public** String testView() {

System.***out***.println("testView");

**return** "helloView";

}