# springmvc配置servlet的注意事项

今天用springmvc搭建项目时，遇到了一个以前没有注意到的问题，在此记录一下。

问题描述

web.xml（部分配置）

<welcome-file-list>

welcome-file>/index.jsp</welcome-file>

</welcome-file-list>

<!-- springmvc servlet -->

<servlet>

<description>spring mvc servlet</description>

<servlet-name>springMvc</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>

classpath:spring/spring-mvc.xml

</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>springMvc</servlet-name>

<url-pattern>/\*</url-pattern>

</servlet-mapping>

<!-- 配置session超时时间，单位分钟 -->

<session-config>

<session-timeout>30</session-timeout>

</session-config>

<!-- 错误处理 -->

<error-page>

<error-code>404</error-code>

<location>/404.jsp</location>

</error-page>

<error-page>

<error-code>500</error-code>

<location>/500.jsp</location>

</error-page>

<error-page>

<exception-type>java.lang.Throwable</exception-type>

<location>/500.jsp</location>

</error-page>

sping-mvc.xml的配置（部分）：

<mvc:annotation-driven>

<mvc:message-converters register-defaults="true">

<!-- StringHttpMessageConverter编码为UTF-8，防止乱码 -->

<bean class="org.springframework.http.converter.StringHttpMessageConverter">

<constructor-arg value="UTF-8"/>

<property name = "supportedMediaTypes">

<list>

<bean class="org.springframework.http.MediaType">

<constructor-arg index="0" value="text"/>

<constructor-arg index="1" value="plain"/>

<constructor-arg index="2" value="UTF-8"/>

</bean>

<bean class="org.springframework.http.MediaType">

<constructor-arg index="0" value="\*"/>

<constructor-arg index="1" value="\*"/>

<constructor-arg index="2" value="UTF-8"/>

</bean>

</list>

</property>

</bean>

<!-- 避免IE执行AJAX时,返回JSON出现下载文件 -->

<!-- <bean id="fastJsonHttpMessageConverter"

class="com.alibaba.fastjson.support.spring.FastJsonHttpMessageConverter">

<property name="supportedMediaTypes">

<list>

<value>application/json;charset=UTF-8</value>

</list>

</property>

</bean> -->

</mvc:message-converters>

</mvc:annotation-driven>

<mvc:default-servlet-handler />

<!--对静态资源文件的访问 -->

<mvc:resources mapping="/static/\*\*" location="/WEB-INF/static/" />

<!-- 自动扫描controller包下的所有类，如果@Controller注入为bean -->

<context:component-scan base-package="com.tooklili.app.web.controller" />

<!-- 对模型视图名称的解析，即在模型视图名称添加前后缀 -->

<bean

class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="viewClass"

value="org.springframework.web.servlet.view.JstlView" />

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

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

</bean>

<!-- 配置多文件上传 -->

<bean id="multipartResolver"

class="org.springframework.web.multipart.commons.CommonsMultipartResolver">

<property name="defaultEncoding">

<value>UTF-8</value>

</property>

<property name="maxUploadSize">

<!-- 上传文件大小限制为100M，100\*1024\*1024 -->

<value>104857600</value>

</property>

<property name="maxInMemorySize">

<value>4096</value>

</property>

</bean>

<!-- 自定义异常处理器 -->

<bean id="exceptionHandler" class="com.tooklili.app.web.exception.MyExceptionHandler"/>

测试控制器类：

package com.tooklili.app.web.controller.test;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

/\*\*

\* 测试控制器

\* @author shuai.ding

\* @date 2017年5月26日下午5:45:25

\*/

@Controller

@RequestMapping("/test")

public class TestController {

@RequestMapping("/test")

@ResponseBody

public String test(){

return "test";

}

@RequestMapping("toView")

public String toView(){

return "view";

}

}

页面webapp/WEB-INF/views/view.jsp

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Insert title here</title>

</head>

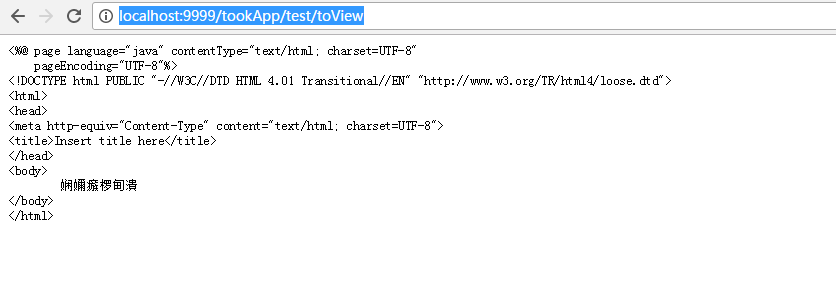
<body>

测试页面

</body>

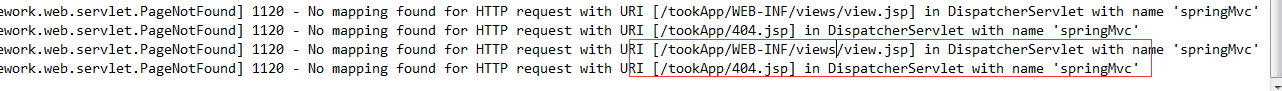
</html>

通过上面配置，访问\*\*\*\*\*/test/toView页面



jsp没有被解析

再看一个诡异现象，如果将spring-mvc.xml中的<mvc:default-servlet-handler />配置注释掉，再次访问\*\*\*\*\*/test/toView页面，此时页面不能访问到。



后台的打印是不是也很诡异，这些现象都是怎么回事呢?

原因分析

关于第一个jsp没有被解析的问题，通过上述描述，应该和<mvc:default-servlet-handler />有关系。

关于将<mvc:default-servlet-handler />配置注释掉，出现服务端跳转的页面又被spingmvc的DispatcherServlet拦截到了，找不到映射关系，然后跳到了404.jsp,但又一次被springmvc的DispatcherServlet拦截到了，最终，浏览器只能展示404的错误页面。从现象看来springmvc的servlet配置有问题。

查看自己的配置，发现springmvc的servlet的映射地址配置的是/\*,可能这个地方有问题。

在这我们来探讨一下两个问题：

1. springmvc的servlet的配置中/和/\*有什么区别
2. <mvc:default-servlet-handler />的作用是什么

关于问题1：

/\*是拦截了所有的请求，包括视图解析后跳转的.jsp页面，/则不会拦截视图解析后的jsp页面。

关于问题2：

<mvc:default-servlet-handler />是一个默认的servlet，主要作用去找静态的资源。总结一下就是：

servlet在找页面时，走的是dispatcherServlet路线。找不到的时候会报404

加上这个默认的servlet时候，servlet在找不到的时候会去找静态的内容。

<mvc:default-servlet-handler />会把"/\*\*" url,注册到SimpleUrlHandlerMapping的urlMap中,把对静态资源的访问由HandlerMapping转到

org.springframework.web.servlet.resource.DefaultServletHttpRequestHandler处理并返回。DefaultServletHttpRequestHandler就是各个Servlet容器自己的默认Servlet。

所以当我们访问\*\*\*\*\*/test/toView页面，

1. 首先经过dispatcherServlet，找到页面view.jsp
2. 访问view.jsp,再次经过dispatcherServlet，springmvc中没有这样的映射，随后走了<mvc:default-servlet-handler />默认的servlet，页面当做静态文件访问了。
3. 如果没有配置<mvc:default-servlet-handler />，找不到映射，浏览器返回404。

解决方法

Springmvc的servlet配置改成如下：

<!-- springmvc servlet -->

<servlet>

<description>spring mvc servlet</description>

<servlet-name>springMvc</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>

classpath:spring/spring-mvc.xml

</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>springMvc</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

总结一下

对于springmvc的servlet拦截匹配规则，即（<url-pattern>...</url-pattern>

）：

1. 拦截\*.do、\*.html,例如：/a/test.do

这个配置比较传统，简单，不易出错，不会导致静态文件（jpg、css、js）被拦截。

1. 拦截/,例如：/a/test

可以实现现在很流行的REST风格，会导致静态文件（jpg、css、js）也被拦截，此时需要配置，让静态文件可访问，如配置<mvc:resources/>和<mvc:default-servlet-handler />。

1. 拦截/\*

这是个错误配置，请求可以转到action，但转到jsp时，会再次被拦截，最终导致页面访问不了的现象。