

在我们之前在wenapp目录中会创建很多不同类型的静态资源目录，如html目录、jsp目录、js目录、images目录等等，分别存放的资源是.html文件、.jsp文件、.js文件、图片资源。

当我们启动Tomcat服务器时，输入相应静态资源的URL路径时，在我们项目的web.xml中并没有声明servlet处理这些请求静态资源的请求。所以这些静态资源的请求是由Tomcat处理的。我们可以查看Tomcat目录中的web.xml文件，发现其中声明了一个名称为default的servlet，在启动Tomcat时创建：从下面的注释中可以知道该servlet：

1、处理访问静态资源的请求

2、处理没有被映射到其他servlet的请求

```
<!-- The default servlet for all web applications, that serves static -->
<!-- resources. It processes all requests that are not mapped to other -->
<!-- servlets with servlet mappings (defined either here or in your own -->
<!-- web.xml file). This servlet supports the following initialization -->
<!-- parameters (default values are in square brackets): -->
<servlet>
    <servlet-name>default</servlet-name>
    <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-
class>
    <init-param>
        <param-name>debug</param-name>
        <param-value>0</param-value>
    </init-param>
    <init-param>
        <param-name>listings</param-name>
        <param-value>>false</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
<!--该servlet的mapping如下-->
<!-- The mapping for the default servlet -->
<servlet-mapping>
    <servlet-name>default</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
```

我们可以看到该servlet的url-pattern用的是一个"/"，这是我们之前在配置中央调度器时url-pattern使用的第二种方式。

表示的是，任何请求都由该default进行接收处理，但是当我们配置了其他的servlet，当请求的地址是映射到了对应的servlet，那么就不会到default的servlet处理。

我们之前在项目的web.xml中声明中央调度器DispatcherServlet时，url-pattern使用的是*.do的方式。我们就需要在Controller类中的控制器方法上添加以.do结尾的URL映射，不方便。

所以我们将DispatcherServlet的url-pattern设置为"/"的方式。但是这样设置之后，我们再去访问静态资源的时候会出现404的错误，就是说没有这个资源。

原因是这种设置使所有的请求包括静态与动态的资源请求都有中央调度器处理，代替了原先Tomcat中的default的servlet。但是中央调度器是分配给相应的控制器来处理这种请求的，我们没有处理静态资源的Controller类，所以就无法访问到这些资源了。

那么如何解决这种问题呢？

1、第一种处理访问静态资源请求的方式

在SpringMVC的配置文件中加入

```
<mvc:default-servlet-handler/>
```

原理是：加入这个标签后，框架会创建控制器对象：DefaultServletHttpRequestHandler

这个就相当于我们自己创建的Controller类对象，中央调度器会把访问静态资源的请求都转给这个对象进行处理，

DefaultServletHttpRequestHandler对象就把请求交给Tomcat的default这个servlet进行处理。这个类内部实现把 请求交给default这个servlet处理是调用了 request.getRequestDispatcher("/xxx").forward(request, response) 的转发方法。

<!--需要注意的是-->

该标签是和控制器方法上的RequestMapping有冲突的，需要在SpringMVC配置文件添加注解驱动标签来处理冲突

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

2、第二种处理访问静态资源请求的方式

前面仍然是把访问静态资源的请求转发给Tomcat服务器的servlet处理。

第二种方式就是我们直接用Spring框架中定义的一个专门处理访问静态资源请求的处理器（用的多，不依赖于其他人）： ResourceHttpRequestHandler

使用方式：

在SpringMVC配置文件中添加标签：<mvc:resources mapping="" location=""/>

加入该标签后框架就会创建 ResourceHttpRequestHandler这个处理器对象。让该对象处理静态资源的访问

其中有两个属性：

mapping：访问的静态资源的URL地址，可使用通配符 ** 代表处理该目录下的任意资源（因为资源不止一个）

location：静态资源在项目中的位置

我们可以这样写

```
<mvc:resources mapping="/html/**" location="WEB-INF/html"/>
```

```
<mvc:resources mapping="/images/**" location="WEB-INF/images"/>
```

上面这样写，如果资源目录较多的话会很麻烦，所以可以再webapp目录下新建一个static目录，把所有包含静态资源的目录全放进去，然后在下面这样写，就可以写一条配置语句就ok了

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

<!--需要注意的是-->

该标签也是和控制器方法上的RequestMapping有冲突的，需要在SpringMVC配置文件添加注解驱动标签来处理冲突

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