

# Spring MVC – How To Include JS Or CSS Files In A JSP Page



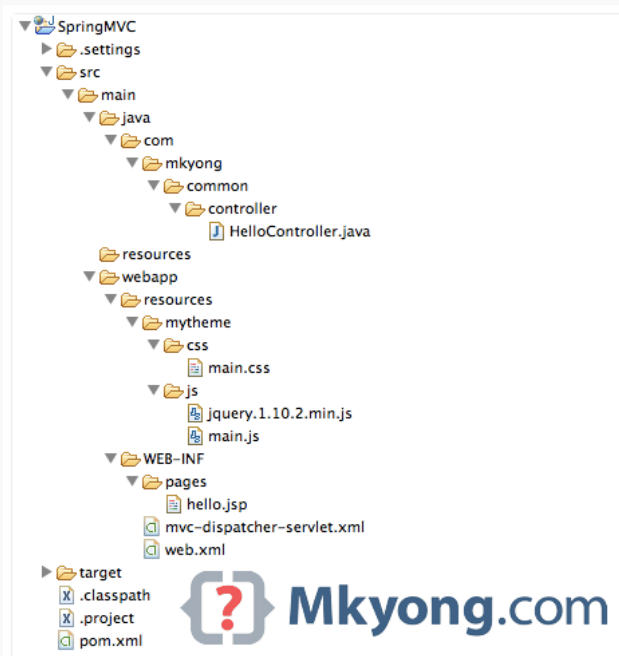
Posted on November 19, 2013 , Last modified : November 18, 2013

By [mkyong](#)

In this tutorial, we will show you how to include **JavaScript** and **CSS** files in a JSP page, with the **Spring MVC** framework environment.

## 1. Project Directory

Review the final project directory structure. A standard Maven folder structure, put the JS and CSS files under the `webapp\resources` folder.



## 2. Resource Mapping

Declares `mvc:resources` , to map "url path" to a physical file path location.

`mvc-dispatcher-servlet.xml`

```
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:mvc="http://www.springframework.org/schema/mvc"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
        http://www.springframework.org/schema/mvc
        http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context-3.0.xsd">

    <context:component-scan base-package="com.mkyong.common.controller" />

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

### Recent Posts



[Multiple SSH private keys Examples](#)



[MongoDB - Allow remote access](#)



[Logback - different log file for each thread](#)



[Java - Get number of available processors](#)



[How to change Eclipse theme](#)

### Popular Tutorials



[Android Tutorial](#)



[JSF 2.0 Tutorial](#)



[Spring Tutorial](#)



[Maven Tutorial](#)



[Hibernate Tutorial](#)

```

    <property name="prefix">
        <value>/WEB-INF/pages/</value>
    </property>
    <property name="suffix">
        <value>.jsp</value>
    </property>
</bean>

<mvc:resources mapping="/resources/**" location="/resources/mytheme/" />

<mvc:annotation-driven />

</beans>

```

In above example, any requests from this url pattern `/resources/**`, Spring will look for `/resources/mytheme/` instead. For example,

1. Requested URL = example.com/resources/js/default.js  
Spring Converted to this = example.com/resources/mytheme/js/default.js
2. Requested URL = example.com/resources/css/default.css  
Spring Converted to this = example.com/resources/mytheme/css/default.css

### 3. Include in JSP Page

To include JS and CSS in a JSP page, uses the standard `c:url` JSTL tag.

hello.jsp

```

<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<html>
<head>
<link href="<c:url value="/resources/css/main.css" />" rel="stylesheet">
<script src="<c:url value="/resources/js/jquery.1.10.2.min.js" />"></script>
<script src="<c:url value="/resources/js/main.js" />"></script>
</head>
<body>
    <h1>1. Test CSS</h1>

    <h2>2. Test JS</h2>
    <div id="msg"></div>

</body>
</html>

```

/webapp/resources/mytheme/js/main.js

```

jQuery(document).ready(function($) {

    $('#msg').html("This is updated by jQuery")

});

```

/webapp/resources/mytheme/css/main.css

```

h1{
    color:red;
}

```

#### Alternatively

For those who hates JSTL, you can use the "page context" variable to get the resources like this :

```

<link href="<${pageContext.request.contextPath}/resources/css/main.css" rel="stylesheet" >

```

### 4. Demo

<http://localhost:8080/SpringMVC/>