**Spring MVC Handler Mapping**

This is an example of how to handle URL requests mapping in Spring MVC. In particular, we shall check on three handler mapping classes that Spring MVC provides for handling URL requests. They are all used to define a mapping between URL requests and handler objects. We will create a simple project with a Controller, a view (jsp) and we will add the necessary configuration files, and then we will use an application server to run the example, making use of all handler mapping classes.

**Create a Controller – View – Model**

The HelloWorldController extends the AbstractController provided by Spring, and overrides thehandleRequestInternal(HttpServletRequest request, HttpServletResponse response) method, where aorg.springframework.web.servlet.ModelAndView is created by a handler and returned to be resolved by theDispatcherServlet.

**public** **class** HelloWorldController **extends** AbstractController {

@Override

**protected** ModelAndView handleRequestInternal(HttpServletRequest request,

HttpServletResponse response) **throws** Exception {

ModelAndView model = **new** ModelAndView("helloWorld");

model.addObject("msg", "hello world!");

**return** model;

}

}

The view is a simple jsp page, that shows the value of the attribute that was set to the HelloWorldController. It must be placed in /WEB-INF/ folder.

helloWorld.jsp

<html>

<body>

<h1>Spring 4.0.2 MVC HelloWorld Controller</h1>

<h3>Your message is : ${msg}</h3>

</body>

</html>

The files that we must configure in the application are the web.xml file and the mvc-dispatcher-servlet.xml file. The web.xml file is the file that defines everything about an application that a server needs to know. It is placed in /WEB-INF/ directory of the application. The <servlet> element declares the DispatcherServlet. When the DispatcherServlet is initialized, the framework will try to load the application context from a file named [servlet-name]-servlet.xml located in/WEB-INF/ directory. So, we have created the mvc-dispatcher-servlet.xml file, that will be explained below. The <servlet-mapping> element of web.xml file specifies what URLs will be handled by the DispatcherServlet.

<servlet>

<servlet-name>spring</servlet-name>

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

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

</servlet>

<servlet-mapping>

<servlet-name>spring</servlet-name>

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

</servlet-mapping>

**BeanNameUrlHandlerMapping**

The BeanNameUrlHandlerMapping class maps URL requests to beans names. It is the default handler mapping class, so it is the one created by the DispatcherServlet when Spring cannot find any handler mapping class declared. An example of using the BeanNameUrlHandlerMapping class is shown below. There are two beans declared, the first one’s name ishelloWorld.htm and its class is the HelloWorldController. So the BeanNameUrlHandlerMapping will map any helloWorldURL request to this Controller. The second bean’s name is the hello\*.htm and its class is also the HelloWorldController. So, in this case, the BeanNameUrlHandlerMapping will map any URL request that starts with hello (such as helloWorld,helloAll) to the HelloWorldController.

<bean

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

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

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

</bean>

<bean

class=*"org.springframework.web.servlet.handler.BeanNameUrlHandlerMapping"* />

<bean name=*"/helloWorld.htm"*

class=*"in.spring4buddies.application.controller.HelloWorldController"* />

<bean name=*"/hello\*.htm"*

class=*"in.spring4buddies.application.controller.HelloWorldController"* />

So, check what happens when the calling the URL helloWorld.htm, helloGeeks.htm

<http://localhost:8080/sf-mvc4-basic-handler-mapping/helloWorld.htm>

<http://localhost:8080/sf-mvc4-basic-handler-mapping/helloGeeks.htm>

## ControllerClassNameHandlerMapping

The ControllerClassNameHandlerMapping class uses a convention to determine the mapping between request URLs and the Controller instances that are to handle those requests. In this case, there is no need to declare a bean name for the Controller. In the example below, the ControllerClassNameHandlerMapping will map to the HelloWorldController all URL requests that start with helloWorld, or helloWorld\*. In the ControllerClassNameHandlerMapping bean declaration there are two properties to configure, the caseSensitive, which is set to true, and the pathPrefix, which is set to/spring4buddies/. These properties allow ControllerClassNameHandlerMapping to also map to the HelloWorldController all URL requests with uppercase characters, like helloWorldSpring4buddies, as also URL requests with path prefix like/spring4buddies/helloWorld.

<bean

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

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

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

</bean>

<bean

class=*"org.springframework.web.servlet.mvc.support.ControllerClassNameHandlerMapping"*>

<property name=*"caseSensitive"* value=*"true"* />

<property name=*"pathPrefix"* value=*"/spring4buddies"* />

</bean>

<bean class=*"in.spring4buddies.application.controller.HelloWorldController"* />

<http://localhost:8080/sf-mvc4-basic-handler-mapping/spring4buddies/helloWorld>

<http://localhost:8080/sf-mvc4-basic-handler-mapping/spring4buddies/helloWorldSpring4buddies.htm>

## SimpleUrlHandlerMapping

The SimpleUrlHandlerMapping provides a property called mappings so as to be configured. This property is set in the bean declaration and consists of key value mapping pairs. It can be set in two ways, as shown below:

<bean class=*"org.springframework.web.servlet.handler.SimpleUrlHandlerMapping"*>

<property name=*"mappings"*>

<props>

<prop key=*"/helloWorld.htm"*>helloWorldController</prop>

<prop key=*"/\*/hello.htm"*>helloWorldController</prop>

<prop key=*"/hello\*.htm"*>helloWorldController</prop>

</props>

</property>

</bean>

<bean id=*"helloWorldController"*

class=*"in.spring4buddies.application.controller.HelloWorldController"* />

or

<bean class=*"org.springframework.web.servlet.handler.SimpleUrlHandlerMapping"*>

<property name=*"mappings"*>

<value>

/helloWorld.htm=helloWorldController

/\*/hello.htm=helloWorldController

/hello\*.htm=helloWorldController

</value>

</property>

</bean>

<bean id=*"helloWorldController"*

class=*"in.spring4buddies.application.controller.HelloWorldController"* />

Note that the Controller bean declaration uses an id property, which is used in the SimpleUrlHandlerMapping bean declaration for the mapping

## Handler mapping priorities

The handler mapping implementations described can be mixed and used together. The only thing that needs to be configured is the priority of each mapping class, so that Spring MVC DispatcherServlet will know which handler mapping implementation to use with what priority. The priority can be set as a property in every mapping bean declaration, as shown below:

<!--ControllerClassNameHandlerMapping -->

<bean

class=*"org.springframework.web.servlet.mvc.support.ControllerClassNameHandlerMapping"*>

<property name=*"caseSensitive"* value=*"true"* />

<property name=*"pathPrefix"* value=*"/spring4buddies"* />

<property name=*"order"* value=*"1"* />

</bean>

<!-- SimpleUrlHandlerMapping -->

<bean class=*"org.springframework.web.servlet.handler.SimpleUrlHandlerMapping"*>

<property name=*"mappings"*>

<props>

<prop key=*"/helloWorld.htm"*>helloWorldController</prop>

<prop key=*"/\*/hello.htm"*>helloWorldController</prop>

<prop key=*"/hello\*.htm"*>helloWorldController</prop>

</props>

</property>

<property name=*"order"* value=*"0"* />

</bean>

<bean id=*"helloWorldController"*

class=*"in.spring4buddies.application.controller.HelloWorldController"* />

in this case, both ControllerClassNameHandlerMapping and SimpleUrlHandlerMapping are used, but the first one to handle a URL request will be the SimpleUrlHandlerMapping.