**Spring MVC Handler Interceptors**

Handler interceptors are provided by Spring’s handler mapping mechanism, and they can be used to apply a certain functionality to requests. For example, an interceptor may be used to check for a principal.

In order to create a handler interceptor, we must make use oforg.springframework.web.servlet.HandlerInterceptor. A custom interceptor must implement this interface. The HandlerInterceptor defines three methods that can be used for preprocessing and postprocessing requests:

* preHandle(..) is called before the actual handler is executed. It returns a boolean value. When it returns true, the handler execution chain continues, whereas when it returns false, theDispatcherServlet assumes the interceptor itself has taken care of requests (and, for example, rendered an appropriate view) and does not continue executing the other interceptors and the actual handler in the execution chain.
* postHandle() is called after the handler is executed.
* afterCompletion() is called after the complete request has finished.

Here, we will create an interceptor implementation, that makes use of preHandle() and postHandle() methods. We will create a simple project with a Controller, an interceptor, a view (jsp) and we will add the necessary configuration files. Then we will use an application server to run the example. The interceptor will keep time in ms every time it is invoked

## Create a Controller – View – Model example

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 {

**private** **static** **final** Logger ***logger*** = Logger

.*getLogger*(HelloWorldController.**class**);

@Override

**protected** ModelAndView handleRequestInternal(HttpServletRequest request,

HttpServletResponse response) **throws** Exception {

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

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

***logger***.info("Hello World Controller is invoked! ");

**return** model;

}

}

helloWorld.jsp

<html>

<body>

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

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

</body>

</html>

<bean

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

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

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

</bean>

<!-- 1. BeanNameUrlHandlerMapping -->

<bean

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

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

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

## Create an interceptor

TimeInterceptor.java class is an interceptor that implements theorg.springframework.web.servlet.handler.HandlerInterceptorAdapter interface. It overrides the preHandle() andpostHandle() methods, where a log message is produced with the time that every method is invoked.

**public** **class** TimeInterceptor **extends** HandlerInterceptorAdapter {

**private** **static** **final** Logger ***logger*** = Logger

.*getLogger*(TimeInterceptor.**class**);

**public** **boolean** preHandle(HttpServletRequest request,

HttpServletResponse response, Object handler) **throws** Exception {

**long** time = System.*currentTimeMillis*();

request.setAttribute("time", time);

***logger***.info(" pre handle method, time in ms is: " + time);

**return** **true**;

}

**public** **void** postHandle(HttpServletRequest request,

HttpServletResponse response, Object handler,

ModelAndView modelAndView) **throws** Exception {

**long** totalTime = System.*currentTimeMillis*()

- (Long) request.getAttribute("time");

modelAndView.addObject("totalTime", totalTime);

***logger***.info(" post handle method, totalTime passed: " + totalTime

+ "ms");

}

}

## Configure the interceptor

The interceptor must be configured in the spring-servlet.xml file. All HandlerMapping classes extending fromAbstractHandlerMapping have an interceptors property, where the interceptor bean is defined through the ref tag.

<bean

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

<property name=*"interceptors"*>

<list>

<ref bean=*"timeInterceptor"* />

</list>

</property>

</bean>

<bean id=*"timeInterceptor"*

class=*"in.spring4buddies.application.interceptor.TimeInterceptor"* />

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