## 로그인 인터셉터 처리

### 작성 순서

* HandlerInterceptorAdapter 클래스를 상속받아 인터셉터 클래스를 정의한다.
* 실행 전 처리는 preHandle() 메서드를, 실행 후 처리는 postHandle() 메서드를 Overriding 하여 정의한다.
* servlet-context.xml을 다음과 같이 추가한다.
  + 상단 <beans:bean>태그 내에 xmlns:mvc 부분 추가

<beans:beans xmlns=*"http://www.springframework.org/schema/mvc"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:beans=*"http://www.springframework.org/schema/beans"*

xmlns:context=*"http://www.springframework.org/schema/context"*

**xmlns:mvc=*"http://www.springframework.org/schema/mvc"***

xsi:schemaLocation=*"http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd"*>

* + <beans:bea> 태그로 로그인 인터셉터 정의

<beans:bean id=*"loginInterceptor"* class=*"global.sesoc.web5.interceptor.LoginInterceptor"*></beans:bean>

* + <mvc:interceptors>로 인터셉트를 사용할 경로들 지정

<mvc:interceptors>

<mvc:interceptor>

<!-- 특정 네임스페이스 하위의 모든 경로에 대하여 다음과 같이 와일드카드 지정 가능 -->

<!-- <mvc:mapping path="/board/\*\*" /> -->

<mvc:mapping path=*"/board/write"* />

<mvc:mapping path=*"/board/download"* />

<mvc:mapping path=*"/board/delete"* />

<mvc:mapping path=*"/board/edit"* />

<mvc:mapping path=*"/board/replyWrite"* />

<mvc:mapping path=*"/board/replyDelete"* />

<mvc:mapping path=*"/board/replyEdit"* />

<mvc:mapping path=*"/customer/update"* />

<!-- 위의 경로들은 다음의 인터셉터를 경유한다. -->

<beans:ref bean=*"loginInterceptor"* />

</mvc:interceptor>

</mvc:interceptors>

### LoginInterceptor.java

**package** global.sesoc.web5.interceptor;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

**import** org.springframework.web.servlet.handler.HandlerInterceptorAdapter;

/\*\*

\* 사용자 로그인 확인 인터셉터. HandlerInterceptorAdapter를 상속받아서 정의.

\*/

**public** **class** LoginInterceptor **extends** HandlerInterceptorAdapter {

**private** **static** **final** Logger ***logger*** = LoggerFactory.*getLogger*(LoginInterceptor.**class**);

//콘트롤러의 메서드 실행 전에 처리

@Override

**public** **boolean** preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)

**throws** Exception {

***logger***.debug("LoginInterceptor 실행");

//세션의 로그인 정보 읽기

HttpSession session = request.getSession();

String loginId = (String) session.getAttribute("loginId");

//로그인되지 않은 경우 로그인 페이지로 이동

**if** (loginId == **null**) {

//request.getContextPath()로 루트 경로를 구하여 절대 경로로 처리

response.sendRedirect(request.getContextPath() + "/customer/login");

**return** **false**;

}

//로그인 된 경우 요청한 경로로 진행

**return** **super**.preHandle(request, response, handler);

}

}

### servlet-context.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans:beans xmlns=*"http://www.springframework.org/schema/mvc"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:beans=*"http://www.springframework.org/schema/beans"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"*

xsi:schemaLocation=*"http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd"*>

<!-- DispatcherServlet Context: defines this servlet's request-processing infrastructure -->

<!-- Enables the Spring MVC @Controller programming model -->

<annotation-driven />

<!-- Handles HTTP GET requests for /resources/\*\* by efficiently serving up static resources in the ${webappRoot}/resources directory -->

<resources mapping=*"/resources/\*\*"* location=*"/resources/"* />

<!-- Resolves views selected for rendering by @Controllers to .jsp resources in the /WEB-INF/views directory -->

<beans:bean class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

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

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

</beans:bean>

<context:component-scan base-package=*"global.sesoc.web5"* />

<!-- 로그인 인터셉터 정의 -->

<beans:bean id=*"loginInterceptor"* class=*"global.sesoc.web5.interceptor.LoginInterceptor"*></beans:bean>

<!-- 로그인 인터셉터를 경유해서 처리할 경로들 지정 -->

<!-- 현 XML파일 상단의 beans 태그 내에 xmlns:mvc="http://www.springframework.org/schema/mvc" 추가하고 사용 -->

<mvc:interceptors>

<mvc:interceptor>

<!-- 특정 네임스페이스 하위의 모든 경로에 대하여 다음과 같이 와일드카드 지정 가능 -->

<!-- <mvc:mapping path="/board/\*\*" /> -->

<mvc:mapping path=*"/board/write"* />

<mvc:mapping path=*"/board/download"* />

<mvc:mapping path=*"/board/delete"* />

<mvc:mapping path=*"/board/edit"* />

<mvc:mapping path=*"/board/replyWrite"* />

<mvc:mapping path=*"/board/replyDelete"* />

<mvc:mapping path=*"/board/replyEdit"* />

<mvc:mapping path=*"/customer/update"* />

<!-- 위의 경로들은 다음의 인터셉터를 경유한다. -->

<beans:ref bean=*"loginInterceptor"* />

</mvc:interceptor>

</mvc:interceptors>

</beans:beans>

## 에러페이지 지정

### 작성 순서

* 예외를 처리할 클래스를 정의
* 클래스에 @ControllerAdvice 어노테이션을 붙인다.
* 예외를 처리할 메서드를 정의
* 메서드 상단에 처리할 예외의 타입과 함께 @ExceptionHandler 어노테이션을 붙인다.
* 포워딩할 JSP 파일을 만든다.
* 없는 경로의 요청이 들어왔을 때, 404페이지를 노출하지 않고 예외를 발생시키려면 web.xml의 <servlet>태그 내에 다음과 같은 내용을 추가한다.

<!-- 404 에러 페이지로 넘기지 않고 익셉션을 발생 -->

<init-param>

<param-name>throwExceptionIfNoHandlerFound</param-name>

<param-value>true</param-value>

</init-param>

### Web5ExceptionHandler.java

**package** global.sesoc.web5.exception;

**import** org.springframework.web.bind.annotation.ControllerAdvice;

**import** org.springframework.web.bind.annotation.ExceptionHandler;

/\*\*

\* 웹 애플리케이션의 공통 예외 처리 클래스

\*/

@ControllerAdvice

**public** **class** Web5ExceptionHandler {

/\*\*

\* 모든 종류의 예외를 처리

\* **@return** JSP파일 경로

\*/

@ExceptionHandler (Exception.**class**)

**public** String errorHandler() {

**return** "/error";

}

}

### error.jsp

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

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>오류</title>

</head>

<body>

<h1>[ 오류 발생 ]</h1>

<p>잠시 후 다시 이용해주시기 바랍니다.</p>

</body>

</html>

### web.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/javaee"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"* version=*"2.5"*>

<context-param>

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

<param-value>/WEB-INF/spring/root-context.xml</param-value>

</context-param>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<servlet>

<servlet-name>appServlet</servlet-name>

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

<init-param>

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

<param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>

</init-param>

<!-- 404 에러 페이지로 넘기지 않고 익셉션을 발생 -->

<init-param>

<param-name>throwExceptionIfNoHandlerFound</param-name>

<param-value>true</param-value>

</init-param>

<!-- /404 에러 페이지로 넘기지 않고 익셉션을 발생 -->

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

</servlet>

<servlet-mapping>

<servlet-name>appServlet</servlet-name>

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

</servlet-mapping>

<filter>

<filter-name>encodingFilter</filter-name>

<filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>

<init-param>

<param-name>encoding</param-name>

<param-value>utf-8</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>encodingFilter</filter-name>

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

</filter-mapping>

</web-app>