**로그인 처리**

**테이블만들기**

**CREATE TABLE tbl\_user(**

**userId varchar(200) primary key,**

**userPw varchar(600),**

**userNm varchar(200),**

**email varchar(300),**

**auth int**

**)**

**Vo 객체 만들기 get,set,toString 해준다.**

private String userid;

private String userpw;

private String usernm;

private String email;

private int auth = 0;

**쿼리 만들기**

<**select** id=*"login"* resultType=*"UserVO"*>

select \* from tbl\_user

where userId = #{userid}

and userPw = #{userpw}

</**select**>

Dao

public UserVO login(UserVO vo);

*DaoImpl*

*@Inject*

SqlSession session;

*@Override*

public UserVO login(UserVO vo) {

return session.selectOne("user.login",vo);

}

service

public UserVO login(UserVO vo);

serviceImpl

*@Inject*

UserDAO dao;

*@Override*

public UserVO login(UserVO vo) {

return dao.login(vo);

}

Controller *@RequestMapping*("/user/\*")

*@Inject* // byType으로 자동 주입

UserService service;

// 로그인 폼을 띄우는 부분

*@RequestMapping*(value="/login",method=*RequestMethod*.***GET***)

public String loginForm(){

return "user/loginForm"; // /login/loginForm.jsp를 띄움.

}//end of loginForm

// 로그인 처리하는 부분

*@RequestMapping*(value="/loginProcess",method=*RequestMethod*.***POST***)

public String loginProcess(HttpSession session,UserVO dto){

String returnURL ="";

if ( session.getAttribute("login") !=null ){

// 기존에 login이란 세션 값이 존재한다면

session.removeAttribute("login"); // 기존값을 제거해 준다.

}

// 로그인이 성공하면 UserVO 객체를 반환함.

UserVO vo = service.login(dto);

if ( vo !=null ){ // 로그인 성공

session.setAttribute("login", vo); // 세션에 login인이란 이름으로 UserVO 객체를 저장해 놈.

returnURL ="redirect:/board/listPage"; // 로그인 성공시 게시글 목록페이지로 바로 이동하도록 하고

}else { // 로그인에 실패한 경우

returnURL ="redirect:/login"; // 로그인 폼으로 다시 가도록 함

}

return returnURL; // 위에서 설정한 returnURL 을 반환해서 이동시킴

}

// 로그아웃 하는 부분

*@RequestMapping*(value="/logout")

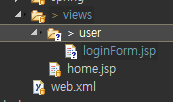
public String logout(HttpSession session) {

session.invalidate(); // 세션 전체를 날려버림

//session.removeAttribute("login"); // 하나씩 하려면 이렇게 해도 됨.

return "redirect:/board/listPage"; // 로그아웃 후 게시글 목록으로 이동하도록...함

}



AuthenticationInterceptor

// 로그인처리를 담당하는 인터셉터

public class AuthenticationInterceptor extends HandlerInterceptorAdapter{

    // preHandle() : 컨트롤러보다 먼저 수행되는 메서드

    @Override

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

            throws Exception {

        // session 객체를 가져옴

        HttpSession session = request.getSession();

        // login처리를 담당하는 사용자 정보를 담고 있는 객체를 가져옴

        Object obj = session.getAttribute("login");

        if ( obj ==null ){

            // 로그인이 안되어 있는 상태임으로 로그인 폼으로 다시 돌려보냄(redirect)

            response.sendRedirect("/login");

            return false; // 더이상 컨트롤러 요청으로 가지 않도록false로 반환함

        }

        // preHandle의return은 컨트롤러 요청 uri로 가도 되냐 안되냐를 허가하는 의미임

        // 따라서true로하면 컨트롤러 uri로 가게 됨.

        return true;

    }

    // 컨트롤러가 수행되고 화면이 보여지기 직전에 수행되는 메서드

    @Override

    Public void postHandle(HttpServletRequest request, HttpServletResponse response, Object handler,

          ModelAndView modelAndView) throws Exception {

        // TODO Auto-generated method stub

        super.postHandle(request, response, handler, modelAndView);

    }

Servler-context.xml (class에 해당 ***interceptor 경로 설정***)

<!-- 인터셉터 객체 생성 -->

**<beans:bean id=*"authenticationInterceptor"* class=*"org.spring.woo.user.interceptor.UserIntercerter"*>**

**</beans:bean>**

**<!-- Interceptor 설정 -->**

**<interceptors>**

**<interceptor>**

**<mapping path=*"/board/register"*/>**

**<mapping path=*"/board/modify"*/>**

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

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

**</interceptor>**

**</interceptors>**

**추가 C:IF로 조절하기**

<c:if test="${login.userId eq boardVO.writer }">

        <button type="submit" class="btn btn-warning">수정</button>

    <button type="submit" class="btn btn-danger">삭제</button>

    </c:if>

Auth로 권한 마다 다르게 들어가기(확인 컨트롤러 만들기)

// 로그인 처리하는 부분

*@RequestMapping*(value = "/loginProcess", method = *RequestMethod*.***POST***)

public String loginProcess(HttpSession session, UserVO dto) {

System.***out***.println("test");

String returnURL = "";

if (session.getAttribute("login") != null) {

// 기존에 login이란 세션 값이 존재한다면

session.removeAttribute("login"); // 기존값을 제거해 준다.

}

// 로그인이 성공하면 UserVO 객체를 반환함.

UserVO vo = service.login(dto);

System.***out***.println("vo : " + vo);

// id 와 password가 같으면

if (vo != null) {

// 권한 별 설정.

//0은 권한이 없는 사용자.

if (vo.getAuth() == 0) {

returnURL = "redirect:/user/loginConfirm?check=0";

//5는 admin

} else if (vo.getAuth() == 5) {

session.setAttribute("login", vo);

returnURL = "redirect:/user/loginConfirm?check=5";

} else {

// 세션에 로그인한 객체 저장

session.setAttribute("login", vo);

returnURL = "redirect:/user/loginConfirm?check=1";

}

} else { // 로그인에 실패한 경우+

returnURL = "redirect:/user/loginConfirm?check=99";

// returnURL ="redirect:/user/login"; // 로그인 폼으로 다시 가도록 함

}

return returnURL; // 위에서 설정한 returnURL 을 반환해서 이동시킴

}

*@RequestMapping*(value = "/loginConfirm", method = *RequestMethod*.***GET***)

public String loginConfirm(*@RequestParam*("check") int check) throws Exception {

return "/user/loginConfirm";

}

jsp부분

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

<!**DOCTYPE** html>

<**html**>

<**head**>

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

<**title**>loginConfirm</**title**>

</**head**>

<**body**>

<**form** action=*"loginConfirm"* id=*"form"*>

<**input** type=*"hidden"* name=*"check"* id=*"check"* value=*"*${param.check}*"*>

</**form**>

</**body**>

<**script** src=*"/resources/js/jquery-1.11.2.min.js"*></**script**>

<**script** type=*"text/javascript"*>

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

//변수선언

**var** obj =$("#form");

**var** check = $("#check").val();

//check

**if**(check == '1'){

window.location.href= '/main/mainPage';

alert("로그인에 성공 하셨습니다.");

}**else** **if**(check == '0'){

window.location.href = '/user/login';

alert("접속 권한이 낮습니다. 관리자에게 문의 하세요.");

}**else** **if**(check == '5'){

window.location.href = '/admin/adminMain';

alert("관리자 권한으로 로그인.");

}**else**{

window.location.href = '/user/login';

alert("아이디와 비번을 확인해 주세요.");

}

});

</**script**>

</**html**>

<**Connector** URIEncoding=*"UTF-8"* connectionTimeout=*"20000"* port=*"80"* protocol=*"HTTP/1.1"* redirectPort=*"8443"*/>