

240305 스프링 프레임워크 11일차 :: 쥬르 없는 집

Springclass-bbs0502

MemberController

// 회원정보 수정 폼 요청

@RequestMapping("/memberUpdateForm")

public String memberUpdateForm(Model model) {

// 모델 처리를 필요 없음 - 로그인 할 때 member 객체를 세션 - 뷰에 가서 출력

return "member/memberUpdateForm";

}

@PostMapping("/memberUpdateResult")

public String memberUpdateResult(Model model,

Member member, String pass1, String emailId,

String emailDomain, String mobile1, String mobile2, String mobile3,

String phone1, String phone2, String phone3,

@RequestParam(value = "emailGet", required = false,

defaultValue = "false") boolean emailGet) {

member.setPass(pass1);

member.setEmail(emailId + "@" + emailDomain);

member.setMobile(mobile1 + "-" + mobile2 + "-" + mobile3);

```
if (phone2.equals("") || phone3.equals("")) {  
    member.setPhone("");  
} else {  
    member.setPhone(phone1 + "-" + phone2 + "-" + phone3);  
}
```

```
member.setEmailGet(Boolean.valueOf(emailGet));
```

```
service.updateMember(member);  
model.addAttribute("member", member);
```

```
return "redirect:/boardList";  
}
```

MemberDaoImpl

```
@Override
```

```
public String memberPassCheck(String id) {  
    return sqlSession.selectOne(NAME_SPACE + ".memberPassCheck", id);  
}
```

```
@Override
```

```
public void updateMember(Member m) {  
    sqlSession.update(NAME_SPACE + ".updateMember", m);  
}
```

MemberService

// 회원 정보 수정시 비밀번호 체크

```
public boolean memberPassCheck(String id, String pass);
```

// 회원 정보 수정

```
public void updateMember(Member m);
```

MemberServiceImpl

@Override

```
public boolean memberPassCheck(String id, String pass) {
```

```
String dbPass = memberDao.memberPassCheck(id);
```

```
boolean result = false;
```

```
if(passwordEncoder.matches(pass, dbPass)) {
```

```
result = true;
```

```
}
```

```
return result;
```

```
}
```

@Override

```
public void updateMember(Member m) {
```

```
m.setPass(passwordEncoder.encode(m.getPass()));
```

```
memberDao.updateMember(m);
```

```
}
```

MemberMapper

```
<select id="memberPassCheck" resultType="String">
```

```
SELECT
```

pass

FROM member

WHERE id=#{id}

</select>

<update id="updateMember" parameterType="Member">

UPDATE member

SET pass=#{pass}, email=#{email}, mobile=#{mobile},

zipcode=#{zipcode}, address1=#{address1}, address2=#{address2},

phone=#{phone}, email_get=#{emailGet}, reg_date=SYSDATE()

WHERE id=#{id}

</update>\

member.js

\$("#memberUpdateForm").on("submit", function() {

if(! \$("#btnPassCheck").prop("disabled")) {

alert("비밀번호를 확인해 주세요");

return false;

}

return joinFormCheck();

});

\$("#btnPassCheck").click(function() {

let oldId = \$("#id").val();

let oldPass = \$("#oldPass").val();

```
if($.trim(oldPass).length == 0) {  
    alert("비밀번호를 입력해 주세요");  
    return false;  
}  
  
let data = "id=" + oldId + "&pass=" + oldPass;  
  
// ajax  
$.ajax({  
    url: "passCheckAjax",  
    type: "post",  
    data: data,  
    dataType: "json",  
    success: function(resData) {  
        console.log(resData.result);  
        if(resData.result) { // 비밀번호 맞는 경우  
            alert("비밀번호가 확인되었습니다.\n비밀번호를 수정해 주세요");  
            $("#btnPassCheck").prop("disabled", true);  
            $("#oldPass").prop("readonly", true);  
            $("#pass1").focus();  
        } else { // 비밀번호가 틀린 경우  
            alert("비밀번호가 틀립니다.");  
            $("#oldPass").val("").focus();  
        }  
    },  
},
```

```
error: function() {  
    console.log("error");  
}  
});
```

```
});
```

servlet-context

```
<beans:bean id="loginCheckInterCeptor"  
class="cohttp://m.springstudy.bbs.interceptor.LoginCheckInterceptor" />
```

```
<interceptors>  
<interceptor>  
<mapping path="/boardDetail" />  
<mapping path="/add*" />  
<mapping path="/write*" />  
<mapping path="/update*" />  
<mapping path="/memberUpdate*" />  
<beans:ref bean="loginCheckInterCeptor" />  
</interceptor>  
</interceptors>
```

memberUpdateForm

교안에서 긁어오기

AjaxProcessController

@Controller

```
public class AjaxProcessController {
```

@Autowired

private MemberService service;

// 뷰가 있나요? JSON(JavaScript Object Notation)

// [{"result": "false"}, ...]

@RequestMapping("/passCheckAjax")

@ResponseBody

public Map<String, Object> memberPassCheck(String id, String pass) {

boolean result = service.memberPassCheck(id, pass);

Map<String, Object> map = new HashMap<>();

map.put("result", result);

return map;

}

}

LoginCheckInterceptor

@Override

public void afterCompletion(HttpServletRequest arg0, HttpServletResponse arg1,

Object arg2, Exception arg3)

throws Exception {

}

@Override

public void postHandle(HttpServletRequest arg0, HttpServletResponse arg1,

Object arg2, ModelAndView arg3)

throws Exception {

```
}
```

@Override

```
public boolean preHandle(HttpServletRequest arg0, HttpServletResponse arg  
1, Object arg2) throws Exception {
```

```
if(arg0.getSession().getAttribute("isLogin") == null) {
```

```
arg1.sendRedirect("loginForm");
```

```
return false;
```

```
}
```

```
return true;
```

```
}
```

좋아요공감

공유하기

통계

게시글 관리