

# 11장. SpringBoard2 – 회원 관리



*Security - login*



# SpringBoard2 – 회원 관리

[Home](#)[회원가입](#)[로그인](#)[회원목록](#)[게시판](#)

**Welcome to our community...**



© Full Stack Software Engineer : Hi-Media



# SpringBoard2 – 회원 관리

## ◆ 프로젝트 계층 구조

```

v SpringBoard2 [springwork master]
  v src/main/java
    v com.cloud.controller
      > BoardController.java
      > CommonController.java
      > HomeController.java
      > MemberController.java
      > SampleController.java
    > com.cloud.domain
    v com.cloud.mapper
      > BoardMapper.java
      > MemberAuthMapper.java
      > MemberMapper.java
    > com.cloud.security.domain
    v com.cloud.service
      > BoardService.java
      > BoardServiceImpl.java
      > CommonService.java
      > CommonServiceImpl.java
      > MemberService.java
      > MemberServiceImpl.java

```

```

v webapp
  > resources
  > upload
  v WEB-INF
    > classes
    > spring
    v views
      > board
      v member
        > memberList.jsp
        > memberView.jsp
        > signup.jsp
      > sample
        > accessError.jsp
        > customLogin.jsp
        > customLogin2.jsp
        > customLogout.jsp
        > footer.jsp
        > home.jsp
        > index.jsp
        > login.jsp
        > menu.jsp
    > web.xml

```



# SpringBoard2 – 회원 관리

```
<%@ taglib uri="http://www.springframework.org/security/tags" prefix="security" %>
<body>
    <nav>
        <security:authorize access="isAnonymous()">
            <ul>
                <li><a href="/">Home</a></li>
                <li><a href="/member/signup">회원가입</a></li>
                <li><a href="/customLogin">로그인</a></li>
                <li><a href="/member/memberList">회원목록</a></li>
                <li><a href="/board/boardList">게시판</a></li>
            </ul>
        </security:authorize>

        <security:authentication property="principal" var="pinfo"/>
        <security:authorize access="isAuthenticated()" >
            <ul>
                <li><a href="/">Home</a></li>
                <li><a href="/member/memberView?userid=<c:out value='${pinfo.username}'/>'>나의 정보
                <li><a href="/customLogout">(<c:out value='${pinfo.username}' />님)로그아웃</a></li>
                <li><a href="/member/memberList">회원목록</a></li>
                <li><a href="/board/boardList">게시판</a></li>
            </ul>
        </security:authorize>
    </nav>
```

menu.jsp



# SpringBoard2 – 회원 관리

```
<title>Welcome~</title>
<link rel="stylesheet" href="/resources/css/style.css">
</head>
<body>
  <jsp:include page="./menu.jsp" />
  <div id="container">
    <section id="main">
      <div class="title">
        <h2>Welcome to our community...</h2>
      </div>

      <div>
        
      </div>
    </section>
  </div>
  <jsp:include page="./footer.jsp" />
</body>
```

index.jsp

```
<footer>
  &copy; Full Stack Software Engineer : Hi-Media
</footer>
```

footer.jsp



# SpringBoard2 – 회원 관리

## ◆ CSS - style

```
@charset "UTF-8";
/* 공통 스타일 */
*{margin: 0; padding: 0;}
li{list-style: none;}
hr{margin: 20px;}
h3,h4{margin: 10px;}
a{text-decoration: none; color: ■ #000}

/* navbar 스타일 */
nav{margin-top: 20px; text-align: right; margin-right: 100px;}
nav ul li{display: inline-block; margin: 10px;
border-top: 3px solid ■ #222; padding: 10px 15px;}
nav ul li:hover{background: □ #eee;}

/* layout */
#container{width: 100%; margin: 0 auto; text-align: center;}
.title{height: 130px; background: □ beige; line-height: 130px;
margin-bottom: 20px;}
.msg{height: 80px; background: □ pink; line-height: 80px; color: ■ brown;}
footer{margin-left: 20%; margin-top: 10%}
```



# SpringBoard2 – 회원 관리

## ◆ CSS - style

```
/* login 스타일 */
.tbl_login{margin: 20px auto; width: 380px;}
.tbl_login, td{border: 1px solid #ccc; border-collapse: collapse;
padding: 10px;}
.tbl_login input{width: 200px; height: 30px; border: 1px solid #aaa;
border-radius: 2px;}
.tbl_login input[type="submit"], input[type="reset"]{width: 60px; height: 30px;}
#login .error{color: #f00; margin-top: 10px;}

/* signup 스타일 */
.tbl_signup{margin: 20px auto; width: 400px;}
.tbl_signup, td{border: 1px solid #ccc; border-collapse: collapse;
padding: 10px;}
.tbl_signup input{width: 200px; height: 30px; border: 1px solid #aaa;
border-radius: 2px;}
.tbl_signup input[type="submit"], input[type="button"], input[type="reset"]{width: 60px; height: 30px;}
.tbl_signup td:nth-child(1){width: 120px}
.tbl_signup .time{text-align: left;}
```



# SpringBoard2 – 회원 관리

## ◆ 회원 설계

```
@Data
public class MemberVO implements Serializable{

    private static final long serialVersionUID = 15L;

    private String userid;
    private String userpw;
    private String username;
    private String enabled;

    private Date regDate;
    private Date updateDate;
    private List<AuthVO> authList;
}
```

```
@Data
public class AuthVO implements Serializable{

    private static final long serialVersionUID = 20L;

    private String userid;
    private String auth;
}
```





# SpringBoard2 – 회원 관리

## ◆ Mapper 설계

```
public interface MemberMapper {  
  
    public void insertMember(MemberVO member); //회원 가입  
  
    public List<MemberVO> getMemberList(); //회원 목록  
  
    public MemberVO read(String userid); //회원 조회(검색)  
  
    public void deleteMember(MemberVO member); //회원 탈퇴  
  
    public void updateMember(MemberVO member); //회원 수정  
  
}
```

```
public interface MemberAuthMapper {  
  
    //회원 가입(권한 부여)  
    public void insertMemberAuth(AuthVO auth);  
  
}
```



# SpringBoard2 – 회원 관리

```
<mapper namespace="com.cloud.mapper.MemberMapper">
  <resultMap type="com.cloud.domain.MemberVO" id="memberMap">
    <id property="userid" column="userid"/>
    <result property="userid" column="userid"/>
    <result property="userpw" column="userpw"/>
    <result property="username" column="username"/>
    <result property="regDate" column="regdate"/>
    <result property="updateDate" column="updatedate"/>
    <collection property="authList" resultMap="authMap" />
  </resultMap>

  <resultMap type="com.cloud.domain.AuthVO" id="authMap">
    <result property="userid" column="userid"/>
    <result property="auth" column="auth"/>
  </resultMap>
```

MemberMapper.xml



# SpringBoard2 – 회원 관리

```
<!-- 회원 정보 -->
<select id="read" resultMap="memberMap">
    SELECT mem.userid, userpw, username, enabled, regdate, updatedate, auth
    FROM tbl_member mem
        LEFT OUTER JOIN tbl_member_auth auth
            on mem.userid = auth.userid
    WHERE mem.userid = #{userid}
</select>

<!-- 회원 목록 -->
<select id="getMemberList" resultMap="memberMap">
    SELECT mem.userid, mem.username, mem.regdate, auth.auth
    FROM tbl_member mem, tbl_member_auth auth
    WHERE mem.userid = auth.userid
</select>
```

MemberMapper.xml



# SpringBoard2 – 회원 관리

```
<!-- 회원 가입 -->
<insert id="insertMember">
    INSERT INTO tbl_member(userid, userpw, username, enabled)
    VALUES (#{userid}, #{userpw}, #{username}, '1')
</insert>

<!-- 회원 삭제 -->
<delete id="deleteMember">
    DELETE FROM tbl_member WHERE userid = #{userid}
</delete>

<!-- 회원 수정 -->
<update id="updateMember">
    UPDATE tbl_member
    SET userpw = #{userpw}, username = #{username}
    WHERE userid = #{userid}
</update>
```

MemberMapper.xml



# SpringBoard2 – 회원 관리

MemberAuthMapper.xml

```
<mapper namespace="com.cloud.mapper.MemberAuthMapper">
  <insert id="insertMemberAuth">
    INSERT INTO tbl_member_auth(userid, auth)
    VALUES (#{userid}, #{auth})
  </insert>
</mapper>
```



# SpringBoard2 – 회원 관리

## ◆ Service 설계

```
public interface MemberService {  
  
    public void signup(MemberVO member); //회원 가입  
  
    public List<MemberVO> getMemberList(); //회원 목록  
  
    public MemberVO read(String userid); //회원 상세보기  
  
    public void delete(MemberVO member); //회원 삭제  
  
    public void update(MemberVO member); //회원 수정  
}
```



# SpringBoard2 – 회원 관리

```
@Service
public class MemberServiceImpl implements MemberService{

    @Autowired
    private MemberMapper mapper;

    @Autowired
    private MemberAuthMapper authMapper;

    @Autowired
    private PasswordEncoder pwencoder;

    @Override
    public void signup(MemberVO member) {
        //비밀번호 암호화
        String encPw = pwencoder.encode(member.getUserpw());
        member.setUserpw(encPw);

        mapper.insertMember(member);

        //기본 ROLE_USER 부여
        AuthVO auth = new AuthVO();
        auth.setUserid(member.getUserid());
        auth.setAuth("ROLE_USER");
        authMapper.insertMemberAuth(auth);
    }
}
```

MemberServiceImpl.java



# SpringBoard2 – 회원 관리

```
@Override
public List<MemberVO> getMemberList() {
    return mapper.getMemberList();
}

@Override
public MemberVO read(String userid) {
    return mapper.read(userid);
}

@Override
public void delete(MemberVO member) {
    mapper.deleteMember(member);
}

@Override
public void update(MemberVO member) {
    //비밀번호 암호화
    String encPw = pwencoder.encode(member.getUserpw());
    member.setUserpw(encPw);

    mapper.updateMember(member);
}
```





# SpringBoard2 – 회원 관리

## ◆ Controller 설계

```
@AllArgsConstructor
@Log4j
@RequestMapping("/member/*")
@Controller
public class MemberController {

    private MemberService service; //생성자 주입

    //회원 가입 폼 요청
    @GetMapping("/signup")
    public void signUp() {
        Log.info("회원 가입 폼");
    }

    @PostMapping("/signup")
    public String signUp(MemberVO member, RedirectAttributes rttr) {
        service.signUp(member);
        //rttr.addFlashAttribute("result", member.getUserid());
        return "redirect:/board/boardList";
    }
}
```



# SpringBoard2 – 회원 관리

## ◆ Controller 설계

```
@AllArgsConstructor
@Log4j
@RequestMapping("/member/*")
@Controller
public class MemberController {
    private MemberService service; //생성자 주입

    //회원 가입 폼 요청
    @GetMapping("/signup")
    public void signUp() {
        Log.info("회원 가입 폼");
    }

    //회원 가입 처리
    @PostMapping("/signup")
    public String signUp(MemberVO member) {
        service.signUp(member);
        return "redirect:/customLogin";
    }
}
```



# SpringBoard2 – 회원 관리

## ◆ Controller 설계

```
//회원 목록 보기
@GetMapping("/memberList")
@PreAuthorize("isAuthenticated()")
public String getMemberList(Model model) {
    List<MemberVO> memberList = service.getMemberList();
    model.addAttribute("memberList", memberList);
    return "/member/memberList";
}

//회원 상세 보기
@GetMapping("/memberView")
public String read(String userid, Model model) {
    MemberVO member = service.read(userid);
    model.addAttribute("member", member);
    return "/member/memberView";
}
```



# SpringBoard2 – 회원 관리

## ◆ Controller 설계

```
//회원 삭제
@GetMapping("/delete")
public String delete(MemberVO member) {
    service.delete(member);
    return "redirect:/member/memberList";
}

//회원 수정
@PostMapping("/update")
public String update(MemberVO member) {
    service.update(member);
    return "redirect:/member/memberList";
}
```



# SpringBoard2 – 회원 관리

## 회원 가입

아이디	<input type="text" value="ID"/>
비밀번호	<input type="text" value="PASSWORD"/>
비밀번호 확인	<input type="text" value="PASSWORD_CONFIRM"/>
이름	<input type="text"/>
<input type="button" value="등록"/> <input type="button" value="취소"/>	



# SpringBoard2 – 회원 관리

```
<form action="/member/signup" method="post" name="regForm"
  onsubmit="return checkForm()">
  <table class="tbl_signup">
    <tr>
      <td>아이디</td>
      <td><input type="text" name="userid" placeholder="ID"></td>
    </tr>
    <tr>
      <td>비밀번호</td>
      <td><input type="password" name="userpw" placeholder="PASSWORD"></td>
    </tr>
    <tr>
      <td>비밀번호 확인</td>
      <td><input type="password" name="userpw_confirm" placeholder="PASSWORD_CONFIRM"></td>
    </tr>
    <tr>
      <td>이름</td>
      <td><input type="text" name="username"></td>
    </tr>
    <tr>
      <td colspan="2">
        <input type="submit" value="등록">
        <input type="reset" value="취소">
      </td>
    </tr>
  </table>
  <input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}" />
</form>
```

signup.jsp



# SpringBoard2 – 회원 관리

## ◆ 유효성 검사

/\*유효성 검사 조건

1. 아이디는 영문자, 숫자 포함 4-12자 이하로 입력해주세요
2. 비밀번호는 영문자, 숫자, 특수문자 포함 8~12자로 입력해주세요
3. 비밀번호를 동일하게 입력하세요
4. 이름을 입력해주세요

\*/

**function** checkForm(){

let form = document.regForm; //폼 이름

let id = form.userid.value; //아이디 입력값

let pwd1 = form.userpw.value; //비밀번호 입력값

let pwd2 = form.userpw\_confirm.value; //비밀번호 확인 입력값

let name = form.username.value; //이름 입력값

//정규식 변수 할당

let regExpId = /^[a-zA-Z0-9]\*\$/ //영문자, 숫자만(^-시작, \*-반복)

let regExpPwd1 = /[a-zA-Z0-9]/ //영문자, 숫자

let regExpPwd2 = /[~!@#\$%^&\*()\_+]/ //특수문자

let regExpPwd3 = /[ㄱ-ㅎㅏ-ㅣ가-힣]/ //한글

validation.js



# SpringBoard2 – 회원 관리

## ◆ 유효성 검사

```
if(id.length < 4 || id.length > 12 || !regExpId.test(id)){
    alert("아이디는 영문자, 숫자 포함 4-12자 이하로 입력해주세요 ");
    form.userid.focus();
    form.userid.select();
    return false;
}else if(pwd1.length < 8 || pwd1.length > 12 ||
        !regExpPwd1.test(pwd1) || !regExpPwd2.test(pwd1) ||
        RegExpPwd3.test(pwd1)){
    alert("비밀번호는 영문자, 숫자, 특수문자 포함 8-12자 이하로 입력해주세요 ");
    form.userpw.focus();
    form.userpw.select();
    return false;
}else if(pwd1 != pwd2){
    alert("비밀번호를 동일하게 입력해주세요");
    form.userpw_confirm.focus();
    form.userpw_confirm.select();
    return false;
}else if(name == ""){
    alert("이름을 입력해주세요");
    form.username.focus();
    return false;
}
```





# SpringBoard2 – 회원 관리

[Home](#)[나의 정보](#)[\(admin님\)로그아웃](#)[회원목록](#)[게시판](#)

## 회원 목록

번호	아이디	이름	가입일
1	cloud	구름이	2022/51/04 02:51:05
2	admin	관리자	2022/51/04 02:51:37

© Full Stack Software Engineer : Hi-Media



# SpringBoard2 – 회원 관리

```
<div id="container">
  <section id="list">
    <div class="title">
      <h2>회원 목록</h2>
    </div>
    <table class="tbl_list">
      <thead>
        <tr>
          <th>번호</th>
          <th>아이디</th>
          <th>이름</th>
          <th>가입일</th>
        </tr>
      </thead>
      <tbody>
        <c:forEach items="${memberList}" var="member" varStatus="status">
          <tr>
            <td><c:out value="${status.count}" /></td>
            <td><a href="/member/memberView?userid=<c:out value="${member.userid}" />">
              <c:out value="${member.userid}" /></a></td>
            <td><c:out value="${member.username}" /> </td>
            <td><fmt:formatDate value="${member.regDate}" pattern="yyyy/mm/dd hh:mm:ss" />
          </tr>
        </c:forEach>
      </tbody>
    </table>
  </section>
```

memberList.jsp



# SpringBoard2 – 회원 관리

## ◆ 관리자 인증(보안)

아이디가 admin인 회원을 관리자로 등급 변경

```
-- 관리자로 등급 변경  
UPDATE tbl_member_auth SET auth = 'ROLE_ADMIN' WHERE userid='admin';
```

AUTH	USERID
1 ROLE_USER	cloud
2 ROLE_ADMIN	admin

security-context.xml

```
<!-- URI 경로 접근 -->  
<security:http>  
  <security:intercept-url pattern="/sample/all" access="permitAll" />  
  <security:intercept-url pattern="/sample/member" access="hasRole('ROLE_MEMBER')" />  
  <security:intercept-url pattern="/member/memberList" access="hasRole('ROLE_ADMIN')" />  
</security:http>
```



# SpringBoard2 – 회원 관리

## ◆ 관리자 인증(보안)

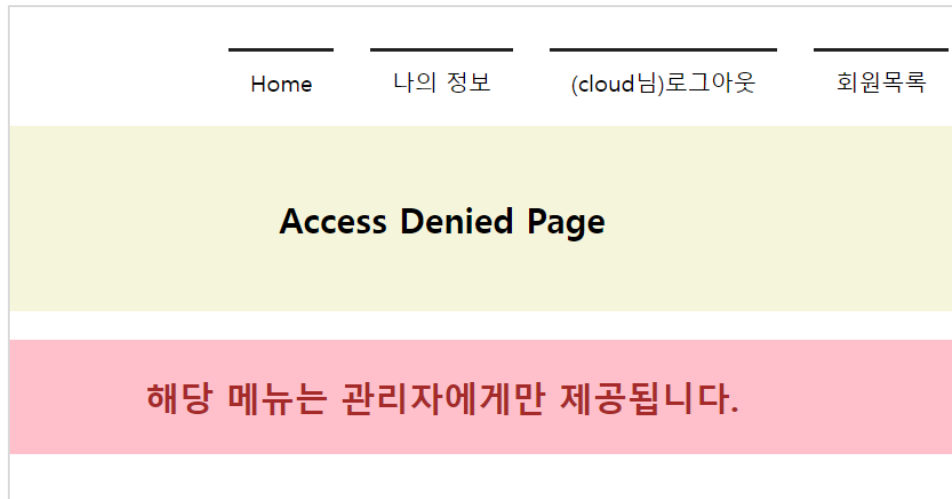
MemberController의 회원 목록 보기 @PreAuthorize("isAuthenticated()")로 인증

```
//회원 목록 보기
@GetMapping("/memberList")
@PreAuthorize("isAuthenticated()")
public String getMemberList(Model model) {
    List<MemberVO> memberList = service.getMemberList();
    model.addAttribute("memberList", memberList);
    return "/member/memberList";
}
```



# SpringBoard2 – 회원 관리

<권한이 없는 회원이 접근하면 에러 처리>



```
<jsp:include page="./menu.jsp" />
<div id="container">
  <section id="login">
    <div class="title">
      <h2>Access Denied Page</h2>
    </div>
    <h2 class="msg"><c:out value="${msg}" /> </h2>
  </section>
</div>
<jsp:include page="./footer.jsp" />
```

access\_Error.jsp



# SpringBoard2 – 회원 관리

[Home](#)   [나의 정보](#)   [\(cloud님\)로그아웃](#)   [회원목록](#)

## 나의 정보

아이디	<input type="text" value="cloud"/>
비밀번호	<input type="password" value="....."/>
비밀번호 확인	<input type="password" value="....."/>
이름	<input type="text" value="구름이"/>
가입일	2022-09-04 02:51:05
<div><input type="button" value="수정"/>   <input type="button" value="탈퇴"/>   <input type="button" value="목록"/></div>	



# SpringBoard2 – 회원 관리

- 회원 정보

```
<script>
    function checkView(){
        let form = document.viewForm;
        let pwd1 = form.userpw.value;
        let pwd2 = form.userpw_confirm.value;

        if(pwd1 != pwd2){
            alert("비밀번호를 동일하게 입력하세요");
            form.userpw_confirm.select();
            return false;
        }
    }
</script>
</head>
<body>
    <jsp:include page="../menu.jsp" />
    <div id="container">
        <div class="title">
            <h1>나의 정보</h1>
        </div>
        <div>
            <form action="/member/update" method="post" name="viewForm"
                onsubmit="return checkView()">
                <table class="tbl_signup">
                    <tr>
                        <td>아이디</td>
                        <td><input type="text" name="userid" value="${member.userid}"
                            readonly="readonly"></td>
                    </tr>
                </table>
            </form>
        </div>
    </div>
</body>
</html>
```



# SpringBoard2 – 회원 관리

memberView.jsp

```
<tr>
  <td>비밀번호</td>
  <td><input type="password" name="userpw" value="${member.userpw}"></td>
</tr>
<tr>
  <td>비밀번호 확인</td>
  <td><input type="password" name="userpw_confirm" value="${member.userpw}"></td>
</tr>
<tr>
  <td>이름</td>
  <td><input type="text" name="username" value="${member.username}"></td>
</tr>
<tr>
  <td>가입일</td>
  <td class="time">
    <fmt:formatDate value="${member.regDate}" pattern="yyyy-MM-dd hh:mm:ss"/>
  </td>
</tr>
<tr>
  <td colspan="2" align="center">
    <input type="submit" value="수정">
    <a href="/member/delete?userid=${member.userid}"
      onclick="return confirm('정말로 탈퇴하시겠습니까?')">
      <input type="button" value="탈퇴"></a>
    <a href="/member/memberList"><input type="button" value="목록"></a>
  </td>
</tr>
</table>
```





# SpringBoard2 – 회원 관리

- 회원 삭제

localhost:8080 내용:  
정말로 탈퇴하시겠습니까>

[확인](#) [취소](#)

[회원목록](#)

## 나의 정보

아이디	<input type="text" value="cloud"/>
비밀번호	<input type="password" value="....."/>



# SpringBoard2 – 회원 관리

- 회원 수정

localhost:8080 내용:  
비밀번호를 동일하게 입력하세요

확인

회원목록

## 나의 정보

아이디	<input type="text" value="cloud"/>
비밀번호	<input type="password" value="....."/>
비밀번호 확인	<input type="password" value="....."/>
이름	<input type="text" value="구름이"/>

