

# 로그인 구현하기 (템플릿)

# 로그인 구현하기 - 1

## • 설명

WEB-INF/views/member/login.jsp 파일에 아래 코드 붙여넣기

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<!doctype html>
<html lang="ko" class="h-100">
<head>
<%@ include file="../common/head.jsp" %>
  <link rel="canonical" href="https://getbootstrap.kr/docs/5.2/examples/sign-in/">
  <!-- Custom styles for this template -->
  <link href="/resources/css/auth/signin.css" rel="stylesheet">
</head>

<body class="text-center">
<main class="form-signin w-100 m-auto">
  <form>
    
    <h1 class="h3 mb-3 fw-normal">Please sign in</h1>
    <div class="form-floating">
      <input type="email" class="form-control" id="floatingInput" placeholder="name@example.com">
      <label for="floatingInput">Email address</label>
    </div>
    <div class="form-floating">
      <input type="password" class="form-control" id="floatingPassword" placeholder="Password">
      <label for="floatingPassword">Password</label>
    </div>
    <div class="checkbox mb-3">
      <label>
        <input type="checkbox" value="remember-me"> Remember me
      </label>
    </div>
    <button class="w-100 btn btn-lg btn-primary" type="submit">Sign in</button>
    <p class="mt-5 mb-3 text-muted">© 2017–2023</p>
  </form>
</main> </body>
</html>
```

## 로그인 구현하기 - 2

- **설명**

resources/css/member/signin.css 파일에 아래 코드 붙여넣기

[illegible]

# 로그인 구현하기 - 3

- 설명

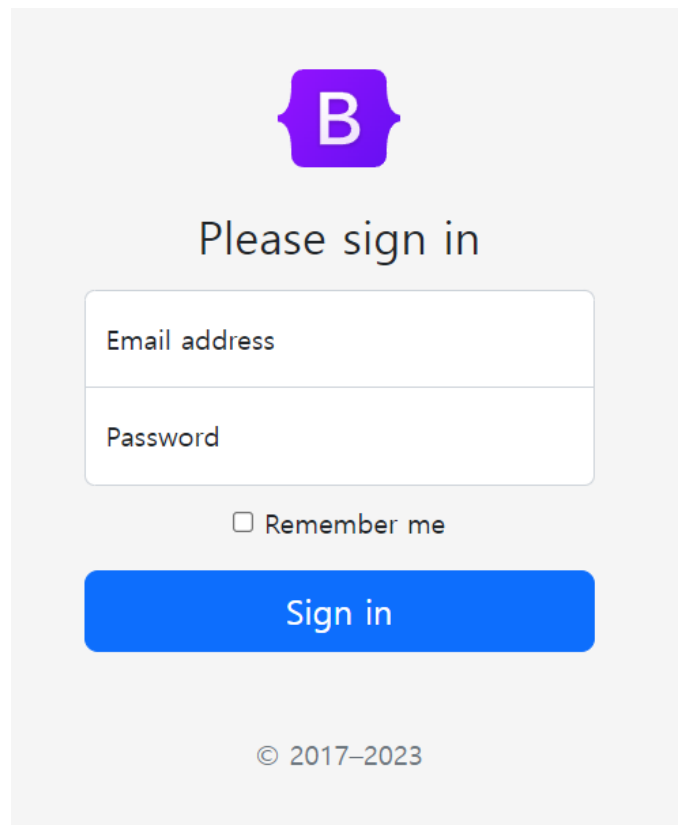
kr.co.greenart.common.controller 패키지에 IndexController 추가  
\* / 경로로 올 때 로그인 페이지로 이동시키는 용도

```
@Controller
public class IndexController {

    @RequestMapping("/")
    public String loginIndex(Model model) {
        return "member/login";
    }
}
```

## 로그인 구현하기 - 4

- 최종 화면



A mockup of a login screen. At the top center is a purple square icon with a white letter 'B'. Below the icon is the text 'Please sign in'. Underneath is a white rectangular form with two input fields: 'Email address' and 'Password'. Below the form is a checkbox labeled 'Remember me'. At the bottom of the form area is a blue button with the text 'Sign in'. At the very bottom of the screen is the copyright notice '© 2017-2023'.

B

Please sign in

Email address

Password

☐ Remember me

Sign in

© 2017-2023

# 로그인 구현하기 - 5

## • 더미 데이터 생성

로그인을 구현하기 위한 테이블, 시퀀스 생성 및 데이터 삽입

```
CREATE TABLE member (  
  m_idx NUMBER PRIMARY KEY,  
  m_email VARCHAR2(100) NOT NULL,  
  m_name VARCHAR2(50) NOT NULL,  
  m_password VARCHAR2(100) NOT NULL,  
  m_indate DATE DEFAULT SYSDATE,  
  m_remove_date DATE  
  CONSTRAINT unique_email UNIQUE (e_email)  
);  
  
CREATE SEQUENCE member_seq START WITH 1 INCREMENT BY 1;  
  
INSERT INTO member (m_idx, m_email, m_name, m_password)  
VALUES (member_seq.NEXTVAL, 'islandtim@naver.com', '김재섭', 'qwer1234');  
  
INSERT INTO member (m_idx, m_email, m_name, m_password)  
VALUES (member_seq.NEXTVAL, 'user1@example.com', '김철수', 'qwer1234');  
  
INSERT INTO member (m_idx, m_email, m_name, m_password)  
VALUES (member_seq.NEXTVAL, 'user2@example.com', '홍길동', 'qwer1234');
```

# 로그인 구현하기

# 로그인 구현하기 - 1

## • 설명

jsp에서 컨트롤러로 데이터를 요청하기 위해 아래와 같이 입력

\* method : 사용자의 비밀번호는 중요한 데이터이기 때문에 반드시 post로 전달해야 함

\* 주의 : 각 태그의 name은 실제 DTO의 변수 이름과 일치해야 함

```
<body class="text-center">
  <main class="form-signin w-100 m-auto">
    <form action="auth/Login.do" method="post">
      
      <h1 class="h3 mb-3 fw-normal">Please sign in</h1>

      <div class="form-floating">
        <input type="email" class="form-control" id="floatingInput" name="email" placeholder="name@example.com">
        <label for="floatingInput">Email address</label>
      </div>
      <div class="form-floating">
        <input type="password" class="form-control" id="floatingPassword" name="pwd" placeholder="Password">
        <label for="floatingPassword">Password</label>
      </div>

      <div class="checkbox mb-3">
        <label>
          <input type="checkbox" value="remember-me"> Remember me
        </label>
      </div>
      <button class="w-100 btn btn-lg btn-primary" type="submit">Sign in</button>
      <p class="mt-5 mb-3 text-muted">© 2017-2023</p>
    </form>
  </main>
</body>
```



## 로그인 구현하기 - 2

- kr.co.greenart.member.controller.MemberController

Object.isNull : 객체가 비어있다면 true  
논리부정(!)을 사용했으니 비어있다면 false  
비어있지 않다면 true

Spring Framework에게 해당 URL로 요청  
(컨트롤러 → 컨트롤러)

```
@PostMapping("/login.do")
public String loginIndex(Member au, HttpSession session) {
    Member loginUser = memberService.loginMember(au);

    if(!Objects.isNull(loginUser)) {
        session.setAttribute("memberIdx", loginUser.getMemberIdx());
        session.setAttribute("msg", "로그인 되었습니다.");
        return "redirect:/board/list.do"; // 페이지 리다이렉트
    }
}
```

## 로그인 구현하기 - 3

- `kr.co.greenart.member.model.Member`

```
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class Member {
    private int memberId;
    private String memberEmail;
    private String memberName;
    private String memberPassword;
    private Date memberIndate;
    private Date memberRemoveDate;
}
```

## 로그인 구현하기 - 4

- kr.co.greenart.member.model.MemberServiceImpl

```
package kr.co.greenart.member.model.service;

import org.mybatis.spring.SqlSessionTemplate;

@Service
public class MemberServiceImpl implements MemberService{

    @Autowired
    private SqlSessionTemplate sqlSession;

    @Autowired
    private MemberDao memberDao;

    @Override
    public Member loginMember(Member au) {
        return memberDao.loginMember(sqlSession, au);
    }
}
```

## 로그인 구현하기 - 5

- `kr.co.greenart.member.model.MemberService`

```
public interface MemberService {  
    Member loginMember(Member au);  
}
```

## 로그인 구현하기 - 6

- `kr.co.greenart.member.model.MemberDao`

```
@Repository
public class MemberDao {
    public Member loginMember(SqlSessionTemplate sqlSession, Member au) {
        return sqlSession.selectOne("memberMapper.loginMember", au);
    }
}
```

## 로그인 구현하기 - 7

- resources/mappers/mybatis-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN" "http://mybatis.org/dtd/mybatis-3-config.dtd" >
<configuration>
  <settings>
    <setting name="jdbcTypeForNull" value="NULL" />
    <setting name="mapUnderscoreToCamelCase" value="true" />
  </settings>

  <typeAliases>
    <typeAlias type="kr.co.greenart.board.model.dto.Board" alias="board" />
    <typeAlias type="kr.co.greenart.member.model.dto.Member" alias="member" />
  </typeAliases>

  <mappers>
    <mapper resource="/mappers/board/board-mapper.xml" />
    <mapper resource="/mappers/member/member-mapper.xml" />
  </mappers>
</configuration>
```

## 로그인 구현하기 - 8

- resources/mappers/member/member-mapper.xml

```
<mapper namespace="memberMapper">
  <resultMap type="member" id="memberResultSet">
    <result column="M_IDX" property="memberIdx"></result>
    <result column="M_EMAIL" property="memberEmail"></result>
    <result column="M_NAME" property="memberName"></result>
    <result column="M_PASSWORD" property="memberPassword"></result>
    <result column="M_INDATE" property="memberIndate"></result>
    <result column="M_REMOVE_DATE" property="memberRemoveDate"></result>
  </resultMap>

  <select id="loginMember" resultMap="memberResultSet">
    SELECT * FROM member
    WHERE M_EMAIL = #{memberEmail}
    AND M_PASSWORD = #{memberPassword}
    AND M_REMOVE_DATE IS NULL
  </select>

</mapper>
```

M\_REMOVE\_DATE 컬럼이 NULL인것만 조회  
\* M\_REMOVE\_DATE = NULL : 회원 탈퇴하지 않은 유저

로그인 구현하기  
(alert 창 띄우기)



# 로그인 구현하기 - alert 창 띄우기

- 설명

MemberController의 session msg 키에 "로그인 되었습니다." 저장

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

    @Autowired
    private MemberServiceImpl memberService;

    @PostMapping("/login.do")
    public String loginIndex(Member au, HttpSession session, Model model) {
        Member loginUser = memberService.loginMember(au);

        if(!Objects.isNull(loginUser)) {
            session.setAttribute("memberIdx", loginUser.getMemberIdx());
            session.setAttribute("msg", "로그인 되었습니다.");
            return "redirect:/board/list.do"; // 페이지 리다이렉트
        } else {
            return "member/login";
        }
    }
}
```

# 로그인 구현하기 - alert 창 띄우기

- 설명

BoardController에 저장했던 msg 세션의 값을 msg 변수에 저장

```
@GetMapping("list.do")
public String boardList(@RequestParam(value="cpage", defaultValue="1") int currentPage,
                        HttpSession session,
                        Model model) {

    // 로그인 메시지
    String msg = (String) session.getAttribute("msg");

    model.addAttribute("list", list); // 객체 바인딩
    model.addAttribute("row", row);
    model.addAttribute("pi", pi);
    model.addAttribute("msg", msg);

    return "board/boardList"; // /WEB-INF/views/boardList.jsp 파일로 포워딩
```

# 로그인 구현하기 - alert 창 띄우기

- 설명

WEB-INF/views/common/head.jsp 파일에 alert 스크립트를 불러올 수 있도록 코드 작성

```
<!-- Alert Script -->
<script src="/resources/js/common/alert.js" ></script>
<script>
    const msg = '<%= request.getAttribute("msg") %>'
    alertFunction(msg);
</script>
```

# 로그인 구현하기 - alert 창 띄우기

- 설명

resources/js/common/alert.js 파일 생성

```
function alertFunction(msg) {  
  if (msg !== "") {  
    alert(msg);  
  }  
}
```

로그인 구현하기  
(sweetAlert2 적용하기)

# 로그인 구현하기 – sweetAlert2 적용하기

- 설명

WEB-INF/views/common/head.jsp 파일에 CDN 및 스크립트 추가

```
<!-- sweetAlert2 CDN -->
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/sweetalert2@11.4.10/dist/sweetalert2.min.css">
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11.4.10/dist/sweetalert2.min.js"></script>

<!-- Alert Script -->
<script src="/resources/js/common/alert.js" ></script>
<script>
    const msg = '<%= request.getAttribute("msg") %>'
    if(msg != "null") {
        $(document).ready(function () {
            alertFunction(msg);
        });
    }
</script>
```

# 로그인 구현하기 – sweetAlert2 적용하기

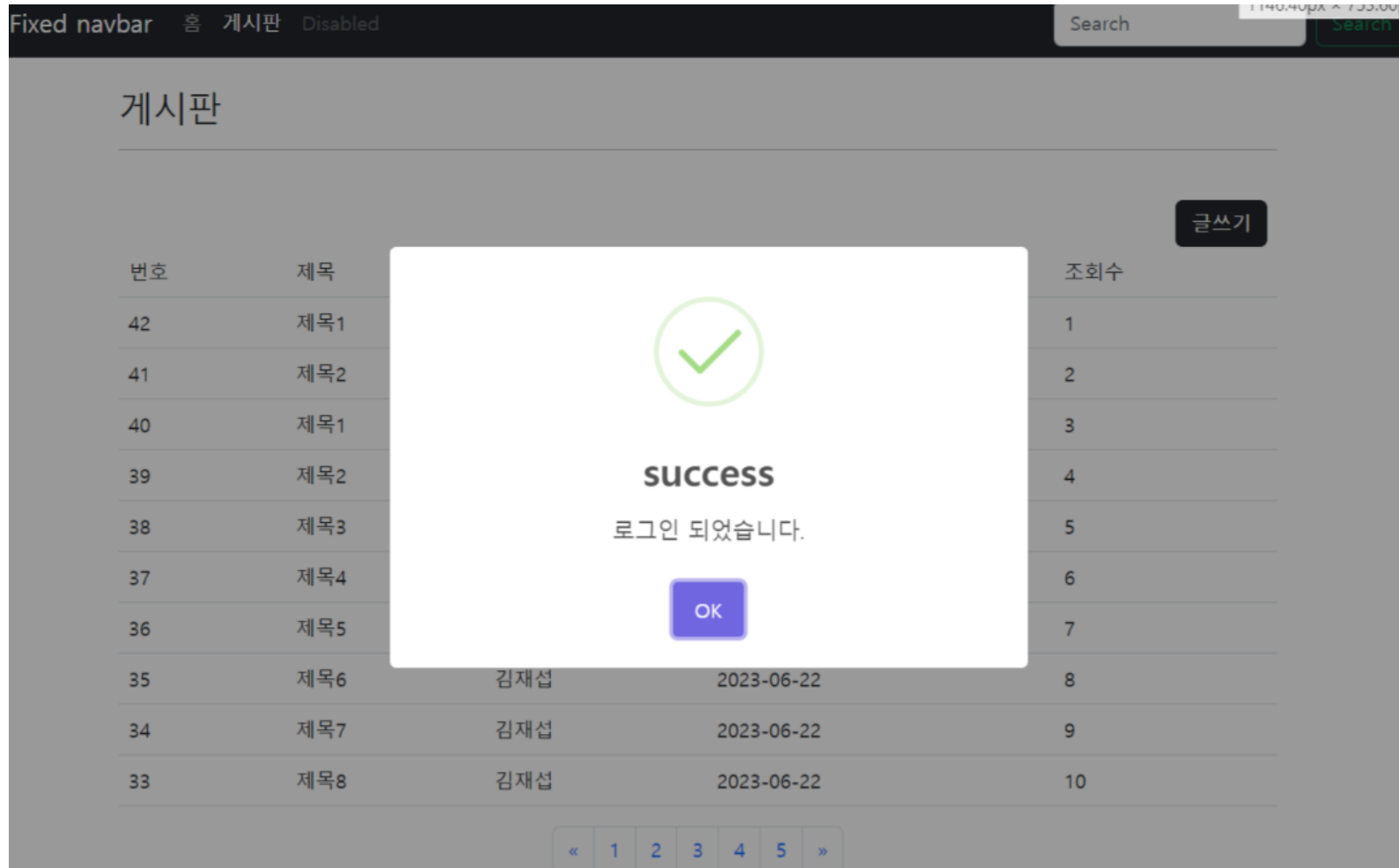
- 설명

resources/js/common/alert.js 파일에 아래 스크립트 추가

```
function alertFunction(msg) {  
    Swal.fire({  
        icon: 'success',  
        title: 'success',  
        text: msg,  
    });  
}
```

# 로그인 구현하기 – sweetAlert2 적용하기

- 최종 화면





로그인 구현하기  
(로그인 실패 팝업창 구현)

# 로그인 구현하기 – sweetAlert2 적용하기

## • MemberController

sweetAlert2 icon을 바꾸기 위한 status 변수 생성

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

    @Autowired
    private MemberServiceImpl memberService;

    @PostMapping("/login.do")
    public String loginIndex(Member au, HttpSession session, Model m) {
        Member loginUser = memberService.loginMember(au);

        if(!Objects.isNull(loginUser)) {
            session.setAttribute("memberIdx", loginUser.getMemberIdx());
            session.setAttribute("msg", "로그인 되었습니다.");
            session.setAttribute("status", "success");
            return "redirect:/board/list.do"; // 페이지 리다이렉트
        } else {
            String msg = "아이디 또는 비밀번호를 확인 해주세요.";
            String status = "error";
            model.addAttribute("msg", msg);
            model.addAttribute("status", status);
            return "member/login";
        }
    }
}
```

# 로그인 구현하기 – sweetAlert2 적용하기

- member/login.jsp

로그인 실패 이후 다시 로그인 버튼을 클릭했을 때 제대로 된 경로 (localhost/member/login.do)로 이동하기 위해 절대 경로로 변경

```
<body class="text-center">
  <main class="form-signin w-100 m-auto">
    <form action="/member/login.do" method="post">
      Please sign in</h1>
```

# 로그인 구현하기 – sweetAlert2 적용하기

- common/head.jsp

새로 생성된 status를 변수에 담고 msg와 status 변수가 모두 비어있지 않을 때  
alertFunction(msg, status) 실행

```
<!-- Alert Script -->
<script src="/resources/js/common/alert.js" ></script>
<script>
    const msg = '<%= request.getAttribute("msg") %>'
    const status = '<%= request.getAttribute("status") %>'
    if(msg !== null && msg !== "null" &&
        status !== null && status !== "null") {
        $(document).ready(function () {
            alertFunction(msg, status);
        });
    }
</script>
```

# 로그인 구현하기 – sweetAlert2 적용하기

- `resources/js/common/alert.js`

매개 변수로 받은 status를 icon과 title에 사용

```
function alertFunction(msg, status) {  
    Swal.fire({  
        icon: status,  
        title: status,  
        text: msg,  
    });  
}
```

# 로그인 구현하기 – sweetAlert2 적용하기

- BoardController

로그인에 성공했을때도 status 변수를 넘겨줘야 하기 때문에 addAttribute 추가 후 msg와 status의 세션 제거

```
// 로그인 메시지
String msg = (String) session.getAttribute("msg");
String status = (String) session.getAttribute("status");

model.addAttribute("list", list); // 객체 바인딩
model.addAttribute("row", row);
model.addAttribute("pi", pi);
model.addAttribute("msg", msg);
model.addAttribute("status", status);

session.removeAttribute("msg");
session.removeAttribute("status");
return "board/boardList"; // /WEB-INF/views/boardList.jsp 파일로 포워딩
```