

운동, 여행, 취미 매칭 플랫폼

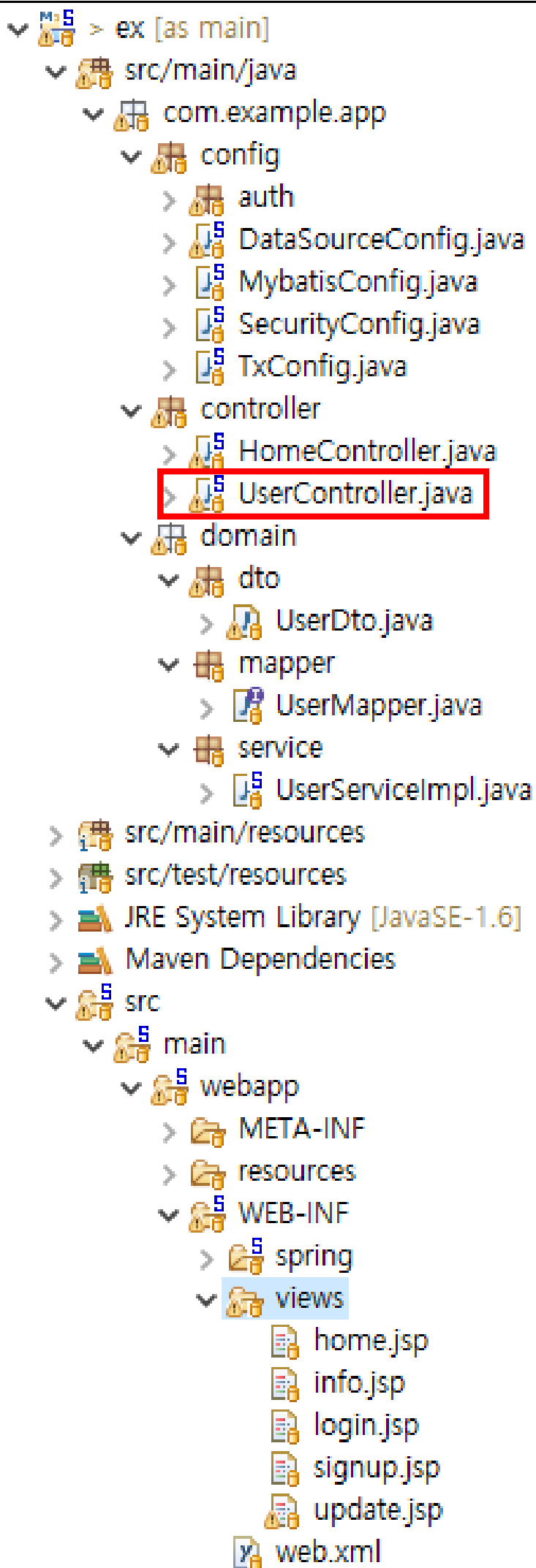
# PlayMatch

## 서버 프로그램 구현

### 2팀

서은빈, 문상희, 신현우, 최서연

코드



• UserController.java

```
package com.example.app.controller;

import javax.validation.Valid;

@Controller
@Slf4j
public class UserController {

    @Autowired
    UserServiceImpl userServiceImpl;

    @GetMapping("/signup")
    public void join() {
        log.info("GET /signup..");
    }

    @PostMapping("/signup")
    public String join_post(@Valid UserDto dto, BindingResult bindingResult, Model model,
        RedirectAttributes redirectAttributes) {
        log.info("POST /signup.." + dto);

        for (FieldError error : bindingResult.getFieldErrors()) {
            log.info("Error Field : " + error.getField() + " Error Msg : " + error.getDefaultMessage());
            model.addAttribute(error.getField(), error.getDefaultMessage());
            return "/signup";
        }

        boolean isJoin = userServiceImpl.userJoin(dto);

        if (isJoin) {
            redirectAttributes.addFlashAttribute("message", "회원가입 성공");
            return "redirect:/login";
        } else {
            return "/signup";
        }
    }
}
```

코드

```
▼ ex [as main]
  ▼ src/main/java
    ▼ com.example.app
      ▼ config
        > auth
        > DataSourceConfig.java
        > MybatisConfig.java
        > SecurityConfig.java
        > TxConfig.java
      ▼ controller
        > HomeController.java
        > UserController.java
      ▼ domain
        ▼ dto
          > UserDto.java
        ▼ mapper
          > UserMapper.java
        ▼ service
          > UserServiceImpl.java
      > src/main/resources
      > src/test/resources
      > JRE System Library [JavaSE-1.6]
      > Maven Dependencies
    ▼ src
      ▼ main
        ▼ webapp
          > META-INF
          > resources
          ▼ WEB-INF
            > spring
            ▼ views
              home.jsp
              info.jsp
              login.jsp
              signup.jsp
              update.jsp
            web.xml
```

```
@GetMapping("/login")
public void login() {
    log.info("GET /login..");
}

@GetMapping("/remove")
public String remove(@RequestParam(value = "email") String email, RedirectAttributes redirectAttributes) {
    log.info("GET /remove..");

    boolean isLeave = userServiceImpl.userLeave(email);

    if (isLeave) {
        redirectAttributes.addFlashAttribute("message", "회원탈퇴 성공");
        return "redirect:/login";
    } else {
        return "/";
    }
}

@GetMapping("/update")
public void update() {
    log.info("GET /update..");
}

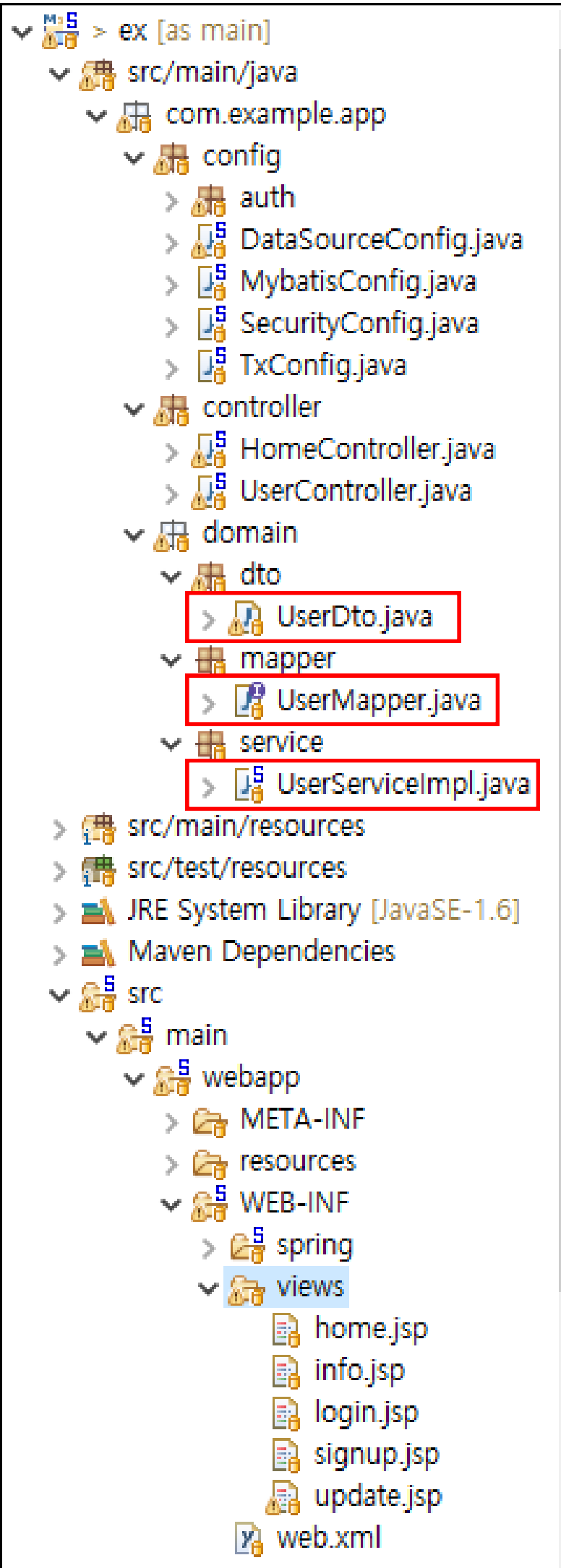
@PostMapping("/update")
public String update_post(@Valid UserDto dto, BindingResult bindingResult, Model model, RedirectAttributes redirectAttributes) {
    log.info("POST /update.." + dto);

    boolean isUpdate = userServiceImpl.userUpdate(dto);
    UserDto userDto = userServiceImpl.userSelect(dto.getEmail());

    if (isUpdate) {
        model.addAttribute("userDTO", userDto);
        return "/info";
    } else {
        return "/";
    }
}

@GetMapping("/user")
public void user(Model model) {
    Authentication authentication = SecurityContextHolder.getContext().getAuthentication();
    log.info("Value : " + authentication.getName());
    log.info("Value : " + authentication.getAuthorities());
    model.addAttribute("username", authentication.getName());
    model.addAttribute("auth", authentication);
}
}
```

코드



• UserDto

```
1 package com.example.app.domain.dto;
2
3 import java.time.LocalDate;
14
15 @Data
16 @NoArgsConstructor
17 @AllArgsConstructor
18 @Builder
19 public class UserDto {
20     private String email;
21     private String name;
22     private String nickname;
23     @NotBlank(message="password 를 입력하세요")
24     private String password;
25     private String role;
26 }
27
```

• UserMapper

```
1 package com.example.app.domain.mapper;
2
3 import org.apache.ibatis.annotations.Delete;
10
11 @Mapper
12 public interface UserMapper {
13
14     @Insert(value = "insert into user_tbl values(#{email},#{name},#{nickname},#{password},#{role})")
15     public int insert(UserDto userDto);
16
17     @Select(value = "select * from user_tbl where email = #{email}")
18     public UserDto selectAt(String username);
19
20     @Delete(value = "delete from user_tbl where email = #{email}")
21     public int delete(String email);
22
23     @Update(value = "update user_tbl set name=#{name}, nickname=#{nickname} where email = #{email}")
24     public int update(UserDto userDto);
25 }
26
```

• UserServiceImpl

```
1 package com.example.app.domain.service;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 @Service
6 @Slf4j
7 public class UserServiceImpl {
8     @Autowired
9     private PasswordEncoder passwordEncoder;
10
11     @Autowired
12     private UserMapper userMapper;
13
14     @Transactional(noRollbackFor = Exception.class)
15     public boolean userJoin(UserDto userDto) {
16
17         userDto.setPassword(passwordEncoder.encode(userDto.getPassword()));
18         userDto.setRole("ROLE_USER");
19         int result = userMapper.insert(userDto);
20         return result > 0;
21     }
22
23     @Transactional(noRollbackFor = Exception.class)
24     public boolean userUpdate(UserDto userDto) {
25         System.out.println(userDto.getEmail());
26         int result = userMapper.update(userDto);
27         return result > 0;
28     }
29
30     @Transactional(noRollbackFor = Exception.class)
31     public boolean userLeave(String email) {
32
33         int result = userMapper.delete(email);
34         return result > 0;
35     }
36
37     @Transactional(noRollbackFor = Exception.class)
38     public UserDto userSelect(String email) {
39
40         UserDto userDto = userMapper.selectAt(email);
41         return userDto;
42     }
43 }
44
```

회원가입 시스템 구현

1. 회원가입

- 이름, 닉네임, 비밀번호, 이메일 입력 시 회원가입 가능
- 이메일이 Id 역할
- 우측 사용 테이블

Table Name: user\_tbl

Schema: testdb

Charset/Collation: utf8mb4 utf8mb4\_09C

Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
email	VARCHAR(20)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
nickname	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
password	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
role	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

logo

임시 회원가입 페이지입니다.  
사용하고 있는 비밀번호를 입력하지마세요.

이름

사용자 닉네임

비밀번호

이메일

인증번호 입력

인증번호전송

회원가입

계정이 있으신가요? 로그인

logo

임시 회원가입 페이지입니다.  
사용하고 있는 비밀번호를 입력하지마세요.

이름

닉네임

....

aa@aa.a

인증번호 입력

인증번호전송

회원가입

계정이 있으신가요? 로그인

회원 로그인 회원가입 성공

아이디

비밀번호

☐ 아이디 저장하기

제출

회원가입

아이디찾기

비밀번호찾기

로그인 시스템 구현

2. 로그인

- 로그인 성공 시 회원 정보 확인 페이지로 이동

회원 로그인

회원가입 성공

aa@aa.a

....

☐ 아이디 저장하기

제출

[회원가입](#) | [아이디찾기](#) | [비밀번호찾기](#)

HOME

EMAIL : aa@aa.a

NAME : 이름

NICKNAME : 닉네임

PASSWORD : \$2a\$10\$6F47i7E.3NCFn8ebRrh4e.0tRkZcTU7yGP.G53VmA5UwTB3wKpS7m

ROLE : ROLE\_USER

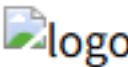
[정보수정](#) [회원탈퇴](#) [로그아웃](#)

	email	name	nickname	password	role
▶	aa@aa.a	이름	닉네임	\$2a\$10\$LaWjRNV8H6mumtscxG4zSOuh1sCC.v...	ROLE_USER
●	NULL	NULL	NULL	NULL	NULL

로그인 시스템 구현

3. 정보 수정

- 회원 정보 수정시 반영 된 정보 출력

회원정보 수정

Email

aa@aa.a

NAME

이름

NickName

닉네임

수정완료

HOME

EMAIL : aa@aa.a

NAME : 이름바꾸기

NICKNAME : 닉네임

PASSWORD : \$2a\$10\$6F47i7E.3NCFn8ebRrh4e.0tRkZcTU7yGP.G53VmA5UwTB3wKpS7m

ROLE : ROLE\_USER

[정보수정](#) [회원탈퇴](#) [로그아웃](#)

	email	name	nickname	password	role
▶	aa@aa.a	이름바꾸기	닉네임	\$2a\$10\$6F47i7E.3NCFn8ebRrh4e.0tRkZcTU7y...	ROLE_USER
✱	NULL	NULL	NULL	NULL	NULL

로그인 시스템 구현

4. 회원 탈퇴

- 회원 탈퇴 시 DB에서 정보 삭제

	email	name	nickname	password	role
▶	aa@aa.a	이름바꾸기	닉네임	\$2a\$10\$6F477E.3NCFn8ebRrh4e.0tRkZcTU7y...	ROLE_USER
✖	NULL	NULL	NULL	NULL	NULL

	email	name	nickname	password	role
✖	NULL	NULL	NULL	NULL	NULL