

운동, 여행, 취미 매칭 플랫폼

PlayMatch

애플리케이션 테스트 수행

2팀

서은빈, 문상희, 신현우, 최서연

1. 유저 로그인 코드

```
@PostMapping("/login")
public ResponseEntity<?> userLogin(@RequestBody @Validated UserDTO userDTO) {
    Map<String, Object> resp = new HashMap<>();

    String userId = userDTO.getUserId();
    String password = userDTO.getPassword();
    JwtTokenDTO jwtTokenDTO = authService.login(userId, password);
    resp.put("jwtToken", jwtTokenDTO.getAccessToken());

    ResponseCookie cookie1 = ResponseCookie.from("accessToken", jwtTokenDTO.getAccessToken())
        .httpOnly(true)
        .path("/")
        .maxAge(Duration.ofDays(1))
        .build();

    ResponseCookie cookie2 = ResponseCookie.from("refreshToken", jwtTokenDTO.getRefreshToken())
        .httpOnly(true)
        .path("/")
        .maxAge(Duration.ofDays(1))
        .build();

    userDTO = authService.selectOne(userId);
    userDTO.setPassword(null);
    userDTO.setRole(null);
    userDTO.setPlatform("0");

    resp.put("userDTO", userDTO);

    return ResponseEntity.ok().header(HttpHeaders.SET_COOKIE, cookie1.toString(), cookie2.toString()).body(resp);
}
```

1. 유저 로그인 test 결과

POST

http://localhost:8100/api/auth/login

Send

Params

Authorization

Headers (11)

Body

Scripts

Settings

Cookies

Beautify

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

{

2

"userId": "admin",

3

"password": "1234"

4

}

Body

Cookies (2)

Headers (16)

Test Results

200 OK

247 ms

1.48 KB

Visualize

{}

JSON

Preview

Visualize

1

{

2

"jwtToken": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhZG1pb2IiImF1dG8iOiJST0xFTU0iIiwidXNlcmlkIjoiaWVhZG1pb2IiLCJpYXQiOiJlbnRpdGzMTU5NjEsImV4cCI6MTc0ODMxNzc2MX0.eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhZG1pb2IiImF1dG8iOiJST0xFTU0iIiwidXNlcmlkIjoiaWVhZG1pb2IiLCJpYXQiOiJlbnRpdGzMTU5NjEsImV4cCI6MTc0ODMxNzc2MX0.",

3

"userDTO": {

4

"id": 0,

5

"userId": "admin",

6

"password": null,

7

"name": "admin",

8

"nickName": "7777",

9

"gender": "male",

10

"birthday": "1986-01-01",

11

"phone": "01012345678",

12

"address": "대구",

13

"profile": "http://localhost:8100/profile/admin",

14

"manner": 300,

15

"introduction": "관리자 계정",

16

"points": 100,

17

"createAt": "2025-05-10",

18

"platform": "0",

19

"role": null,

20

"private": false

21

}

22

}

2. 유저 조회

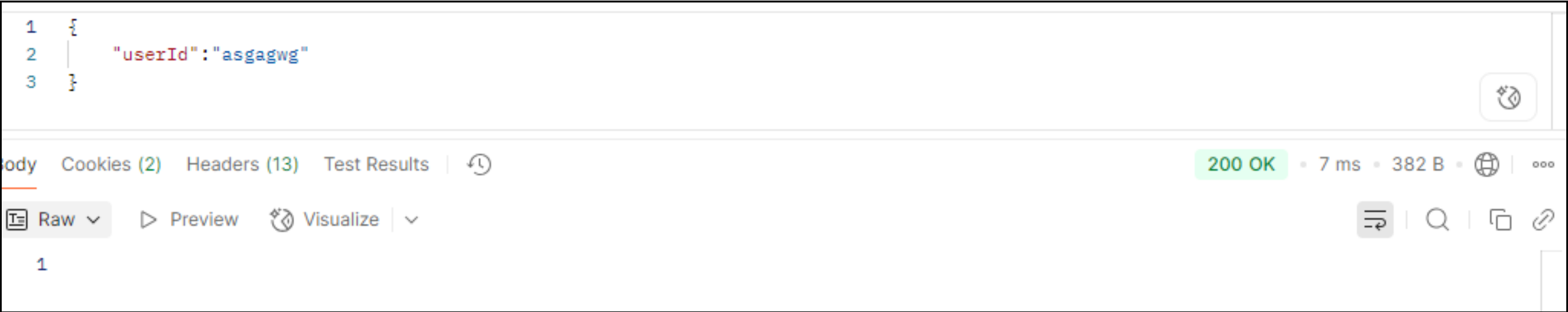
회원가입 - 중복 아이디 체크

```
@PostMapping("/check-id")
public ResponseEntity<?> userCheck(@RequestBody Map<String,String> req) {

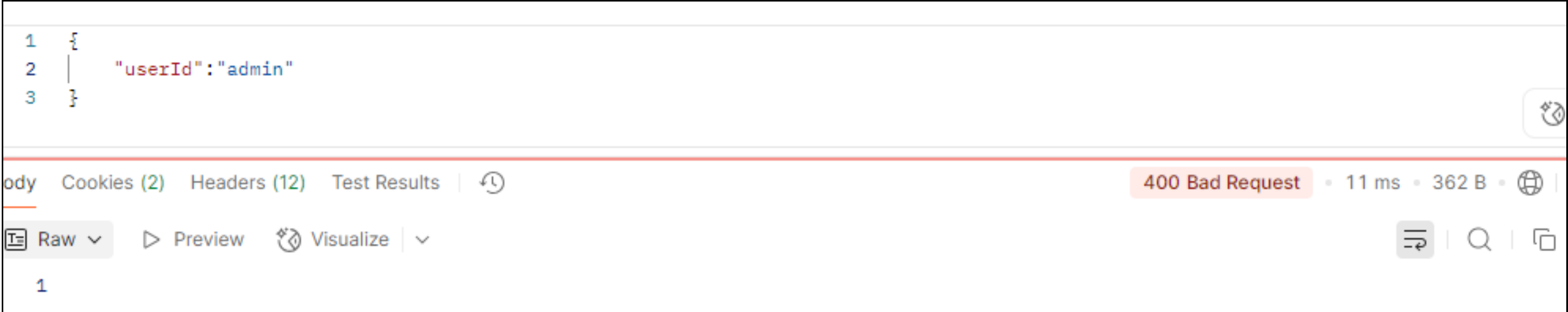
    String userId = (String) req.get("userId");
    UserDTO userDTO = authService.selectOne(userId);
    if(userDTO == null) {
        return ResponseEntity.ok().body(null);
    } else {
        return ResponseEntity.status(HttpStatus.BAD_REQUEST).body(null);
    }
}
```

2. 유저 조회 test 결과

중복 아닐 시 성공 : 200 OK 상태 전송



중복일 시 실패 : 400 Bad 상태 전송



3. 유저 내정보 수정 코드

회원 정보 - 주소 수정

```
@PostMapping("infoUpdate")
public ResponseEntity<?> infoUpdate(@RequestBody Map<String, Object> req){
    Authentication authentication = SecurityContextHolder.getContext().getAuthentication();

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

    UserDTO userDTO = null;

    String userId = authentication.getName();
    String value = (String)req.get("value");
    String type = (String)req.get("type");

    if(type.equals("phone")) {
        log.info("폰번호 변경");
    } else { // 나머지
        boolean isOk = userService.changeUserInfo(userId,value,type);
    }

    userDTO = userService.serchUserOne(userId);
    userDTO.setPassword("");
    userDTO.setRole("");

    resp.put("userDTO",userDTO);
    return ResponseEntity.ok().body(resp);
}
```

3. 유저 내정보 수정 test 결과

HTTP

http://localhost:8100/api/user/infoUpdate

Save

Share

POST

http://localhost:8100/api/user/infoUpdate

Send

Params

Authorization

Headers (11)

Body

Scripts

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

{

2

"value": "서울",

3

"type": "address"

4

}

Body

Cookies (2)

Headers (14)

Test Results

200 OK

53 ms

762 B

Visualize

{}

JSON

Preview

Visualize

1

{

2

"userDTO": {

3

"id": 0,

4

"userId": "admin",

5

"password": "",

6

"name": "admin",

7

"nickName": "7777",

8

"gender": "male",

9

"birthday": "1986-01-01",

10

"phone": "01012345678",

11

"address": "서울",

12

"profile": "http://localhost:8100/profile/admin",

13

"manner": 300,

14

"introduction": "관리자 계정",

15

"points": 100,

16

"createAt": "2025-05-10",

17

"platform": null,

18

"role": "",

19

"private": false

20

}

21

}

4. 유저 삭제 코드

```
@PostMapping("/remove")
public ResponseEntity<?> userRemove() throws IOException {

    Authentication authentication = SecurityContextHolder.getContext().getAuthentication();

    String userId = authentication.getName();

    String userPath = "src/main/resources/Users/" + userId;

    File userDir = new File(userPath);

    FileUtils.deleteDirectory(userDir);

    boolean isDelete = authService.leaveUser(userId);

    return ResponseEntity.ok().body(null);
}
```


4. 유저 삭제 test 결과

	Key	Value	Description	⋮ Bulk Edit Presets ▾
<input checked="" type="checkbox"/>	Cookie	 ⓘ accessToken=eyJhbGciOiJIUzI1NiJ9.eyJzdWliOiJhZG1pbilslmF1dGgiOiJST0xFX0FETUIOlwidXNlcklkloiYW		
<input checked="" type="checkbox"/>	Postman-Token	 ⓘ WRtaW4iLCJpYXQiOiE3NDgzMTU5NjEsImV4cCI6MTc0ODMxNzc2MX0		
<input checked="" type="checkbox"/>	Content-Type	 ⓘ i9-dJI8oUaLc325gMdQI-S-Q8pVbXE_otYZBEsHfZhU;		
<input checked="" type="checkbox"/>	Content-Length	 ⓘ refreshToken=eyJhbGciOiJIUzI1NiJ9.eyJzdWliOiJhZG1pbilslmF1dGgiOiJST0xFX0FETUIOlwidXNlcklkloiYW		
<input checked="" type="checkbox"/>	Host	 ⓘ WRtaW4iLCJpYXQiOiE3NDgzMTU5NjEsImV4cCI6MTc0ODQwMjMxO		
		 ⓘ .lpAvcppHt3z1wBUIMdEsECz5J85FMjVND07MzK9zcj0		

POST

⌵

http://localhost:8100/api/auth/remove

Send

⌵

Params

●

Authorization

●

Headers (10)

Body

Scripts

Settings

Cookies

☒

none

☐

form-data

☐

x-www-form-urlencoded

☐

raw

☐

binary

☐

GraphQL

This request does not have a body

Body

Cookies (2)

Headers (13)

Test Results

↺

200 OK

• 586 ms

• 382 B

• 🌐

• ⋮

Raw

⌵

▶ Preview

🔗 Visualize

⌵

↺

🔍

📄

🔗

1