API 명세서 - 2. 회원 API

전체 API 목록

API	분류	Method	API 명	설명
회원	회원가입	POST	/sign-up	계정 생성
	이메일, 닉네임 중복 확인	GET	/check	계정, 닉네임 중복 확인
	이메일 인증	POST	/check/auth	인증 코드 메일로 전송
		GET	/check/auth	메일 인증
	회원탈퇴	DELETE	/withdrawal	계정 삭제
	로그인	POST	/sign-in	로그인 후 토큰 획득
	로그아웃	GET	/sign-out	로그아웃, 토큰 해제
	계정 조회	GET	/user	모든 계정 정보 조회
		GET	/user/{id}	본인 계정 정보 조회
	계정 정보 수정	PUT	/user/{id}	계정 정보 수정
	계정 권한 수정	PUT	/user/role	계정 권한 수정

회원 가입 API

Request

Request Syntax

```
curl -X POST 'http://localhost/sign-up'
```

API 호출 방식

메서드	요청 url
POST	`/sign-up`

Request Body

```
{
    "email": "zhyun@gmail.com",
    "password": "secret!!",
```

```
    "nickname": "얼거스"
}
```

파라미터	타입	필수 여부	설명
`email`	String	`필수`	아이디로 사용 될 이메일주소
`password`	String	`필수`	비밀번호
`nickname`	String	`필수`	닉네임

Response

회원가입 성공 응답 예시

```
{
    "status": true,
    "message": "얼거스님 가입을 축하합니다! 🥳"
}
```

실패 응답 예시

- CASE 1 : 이메일 필드에 값이 없음

```
{
    "status": false,
    "message": "이메일을 입력해 주세요."
}
```

- CASE 2 : 비밀번호 필드에 값이 없음

```
{
    "status": false,
    "message": "비밀번호를 입력해 주세요."
}
```

- CASE 3 : 닉네임 필드에 값이 없음

```
{
    "status": false,
    "message": "닉네임을 입력해 주세요."
}
```

이메일 중복확인 API

Request

Request Syntax

```
curl -X GET 'http://localhost/check?email=zhyun@gmail.com'
```

API 호출 방식

메서드	요청 url
GET	`/check`

Query String

```
?email=zhyun@gmail.com
```

Key	타입	필수 여부	설명
`email`	String	`필수`	아이디로 사용 될 이메일 주소

Response

성공 응답 - 중복되지 않음

```
{
    "status": true,
    "message": "사용 가능한 이메일입니다. 이메일 인증을 진행해주세요!"
}
```

실패 응답

CASE 1 : 중복됨

```
{
    "status": false,
    "message": "이미 사용중인 이메일입니다."
}
```

CASE 2 : 이메일 형식에 어긋남

```
{
    "status": false,
    "message": "올바른 이메일 주소를 입력해 주세요."
}
```

CASE 3 : 이메일 값이 비어있음

```
{
    "status": false,
    "message": "이메일 주소를 입력해 주세요."
}
```

닉네임 중복확인 API

Request

Request Syntax

```
curl -X GET 'http://localhost/check?nickname=얼거스'
```

API 호출 방식

메서드	요청 url
GET	`/check`

Query String

```
?nickname=얼거스
```

Key	타입	필수 여부	설명
`nickname`	String	`필수`	닉네임

Response

성공 응답 - 중복되지 않음

```
{
    "status": true,
    "message": "사용 가능한 닉네임 입니다."
}
```

실패 응답

CASE 1 : 중복됨

```
{
    "status": false,
    "message": "이미 사용중인 닉네임 입니다."
}
```

CASE 2 : 닉네임 형식에 어긋남

```
{
    "status": false,
    "message": "닉네임은 6글자 이하로 작성해야 합니다."
}
```

CASE 3 : 닉네임 값이 비어있음

```
{
    "status": false,
    "message": "닉네임을 입력해 주세요."
}
```

이메일 인증 API - 이메일 전송

- api 진입시 인증 코드를 메일로 전송

Request

Request Syntax

```
curl -X POST 'http://localhost/check/auth'
```

API 호출 방식

메서드	요청 url
POST	`/check/auth`

Request Body

```
{
    "email": "zhyun@gmail.com"
}
```

파라미터	타입	필수 여부	설명
`email`	String	`필수`	아이디로 사용 될 이메일주소

Response

이메일 인증 메일 발송 성공 응답

```
{
    "status": true,
    "message": "전송 되었습니다."
}
```

이메일 인증 메일 발송 실패 응답 예시 - Case 1 : 중복확인 안 된 이메일

```
{
    "status": false,
    "message": "이메일 중복확인을 먼저 진행해주세요."
}
```

이메일 인증 메일 발송 실패 응답 예시 - Case 2 : 잘못된 이메일 형식

```
{
    "status": false,
    "message": "이메일 형식이 올바르지 않습니다."
}
```

이메일 인증 API - 메일 인증

- 이메일 주소로 전송 된 코드를 Query String으로 입력

Request

Request Syntax

```
curl -X GET 'http://localhost/check/auth?code=SDFG2345'
```

API 호출 방식

메서드	요청 url
GET	`/check/auth`

Query String

```
?code=ASDG234
```

Key	타입	필수 여부	설명
`code`	String	`필수`	서버에서 생성되는 이메일 인증 코드

Response

이메일 인증 성공 응답 예시

```
{
    "status": true,
```

```
    "message": "인증 되었습니다."
}
```

실패 응답 예시 - Case 1 : 인증 코드가 일치하지 않음

```
{
    "status": false,
    "message": "인증 번호가 일치하지 않습니다."
}
```

실패 응답 예시 - Case 2 : 만료된 인증 코드

```
{
    "status": false,
    "message": "인증 번호가 만료되었습니다. 인증을 다시 진행해주세요!"
}
```

회원 탈퇴 API

- 논리 삭제 후 30일 후에 물리 삭제

Request

Request Syntax

```
curl -X DELETE 'http://localhost/withdrawal'
```

API 호출 방식

메서드	요청 url
DELETE	`/withdrawal`

Response

회원 탈퇴 성공 응답 예시

```
{
    "status": true,
    "message": "얼거스(zhyun@gmail.com)님 탈퇴되었습니다."
}
```

로그인 API

Request

Request Syntax

```
curl -X GET 'http://localhost/sign-in'
```

API 호출 방식

메서드	요청 url
POST	`/sign-in`

Request Body

```
{
    "email": "zhyun@gmail.com",
    "password": "secret!!"
}
```

파라미터	타입	필수 여부	설명
`email`	String	`필수`	아이디로 사용 될 이메일주소
`password`	String	`필수`	비밀번호

Response

성공 응답 예시

```
{
    "status": true,
    "message": "얼거스님 로그인 성공"
}
```

실패 응답 예시 - Case 1 : 잘못된 email

```
{
    "status": false,
    "message": "없는 사용자입니다."
}
```

실패 응답 예시 - Case 2 : 잘못된 비밀번호

```
{
    "status": false,
```

```
    "message": "계정 정보가 일치하지 않습니다."
}
```

로그아웃 API

Request

Request Syntax

```
curl -X GET 'http://localhost/sign-out'
```

API 호출 방식

메서드	요청 url
GET	`/sign-out`

Response

성공 응답 예시

```
{
    "status": true,
    "message": "얼거스(zhyun@gmail.com)님 로그아웃 성공"
}
```

계정 조회 API - 본인 계정 상세 조회

Request

Request Syntax

```
curl -X GET 'http://localhost/user/{id}'
```

API 호출 방식

메서드	요청 url
GET	`/user/{id}`

- path parameter : user id

Response

성공 응답 예시

```
{
    "status": true,
    "message": "상세 조회",
    "result": {
        "id": 1,
        "email": "zhyun@gmail.com",
        "nickname": "얼거스",
        "role": "ADMIN",
        "created_at": "2023-12-10T21:54:32.321",
        "modified_at": "2023-12-10T21:54:32.321"
   }
}
```

실패 응답 예시 - Case 1 : 다른 사람 id 조회 (권한 없음)

```
{
    "status": false,
    "message": "권한이 없습니다."
}
```

계정 조회 API - 전체 계정 상세 조회

Request

Request Syntax

```
curl -X GET 'http://localhost/user'
```

API 호출 방식

메서드	요청 url
GET	`/user`

Response

성공 응답 예시

```
{
    "status": true,
    "message": "전체 계정 상세 조회",
```

```
    "result": [
        {
            "id": 1,
            "email": "zhyun@gmail.com",
            "nickname": "얼거스",
            "role": "ADMIN",
            "created_at": "2023-12-10T21:54:32.321",
            "modified_at": "2023-12-10T21:54:32.321"
        },
        {
            "id": 2,
            "email": "gimwlgus@kakao.com",
            "nickname": "김얼거스",
            "role": "MEMBER",
            "created_at": "2023-12-10T21:54:32.321",
            "modified_at": "2023-12-10T21:54:32.321"
        },
        {
            "id": 3,
            "email": "gimwlgus@gmail.com",
            "nickname": "닉넴",
            "role": "WITHDRAWAL",
            "created_at": "2023-12-10T21:54:32.321",
            "modified_at": "2023-12-10T21:54:32.321"
        },
    }
}
```


실패 응답 예시 - Case 1 : ADMIN 권한이 아닌 계정의 API 사용


```
{
    "status": false,
    "message": "권한이 없습니다."
}
```


계정 수정 API

- 이메일 변경시 인증 process 진행 후 변경

Request

Request Syntax

```
curl -X GET 'http://localhost/user/{id}'
```

API 호출 방식

메서드	요청 url
PUT	`/user/{id}`

- path parameter : user id

Request Body

```
{
    "id": 1,
    "email": "zhyun@naver.com",
    "nickname": "오예"
}
```

파라미터	타입	필수 여부	설명
`email`	String	`필수`	아이디로 사용 될 이메일주소
`nickname`	String	`필수`	닉네임

Response

성공 응답 예시

```
{
    "status": true,
    "message": "계정 정보가 수정되었습니다."
}
```

실패 응답 예시 - Case 1 : 닉네임 수정시 닉네임 형식에 어긋남

```
{
    "status": false,
    "message": "닉네임은 6글자 이하로 작성해야 합니다."
}
```

실패 응답 예시 - Case 2 : 닉네임 수정시 중복 검사 안함

```
{
    "status": false,
    "message": "닉네임 중복 확인을 진행해주세요."
}
```

계정 권한 수정 API

- `ADMIN` 만 사용 가능

Request

Request Syntax

```
curl -X GET 'http://localhost/user/role'
```

API 호출 방식

메서드	요청 url
PUT	`/user/role`

- path parameter : user id

Request Body

```
{
    "role": "MEMBER"
}
```

파라미터	타입	필수 여부	설명
`role`	String	`필수`	사용자 권한

Response

성공 응답 예시

```
{
    "status": true,
    "message": "닉네임님 권한이 MEMBER(으)로 수정되었습니다."
}
```

실패 응답 예시 - Case 1 : 없는 사용자 id

```
{
    "status": false,
    "message": "잘못된 요청입니다."
}
```

실패 응답 예시 - Case 2 : 권한이 없는 사용자의 진입

```
{
    "status": false,
    "message": "권한이 없습니다."
}
```