

## 회원

### 회원가입

- ACCOUNT\_ACCOUNT\_STATE 에 가입 및 활성이 들어가야 한다.

### 로그인

- ID 조회, PW 매치...
- 로그인 성공시 LAST\_LOGIN\_AT 시간 변경 할 수 있는 API 필요

### 회원관리

- ACCOUNT\_ACCOUNT\_STATE CRUD

### ID 조회

- 프로젝트 쪽에서 무결성 확인을 위해 호출되는 API

### 주의사항

- 회원 삭제 (탈퇴아님) 시 프로젝트 쪽 해당 데이터 전부 삭제 시키는 것 고려

ACCOUNT, ACCOUNT\_STATE, ACCOUNT\_ACCOUNT\_STATE

# 메시지

- json 포맷을 사용합니다.

## 요청 메시지

- json body를 포함하여 보내야 하는 메시지의 경우, Content-Type 헤더를 명시해야 합니다.
- 모든 요청에 Host를 명시합니다.
- 모든 요청에 Authorization 헤더를 포함해 요청합니다.

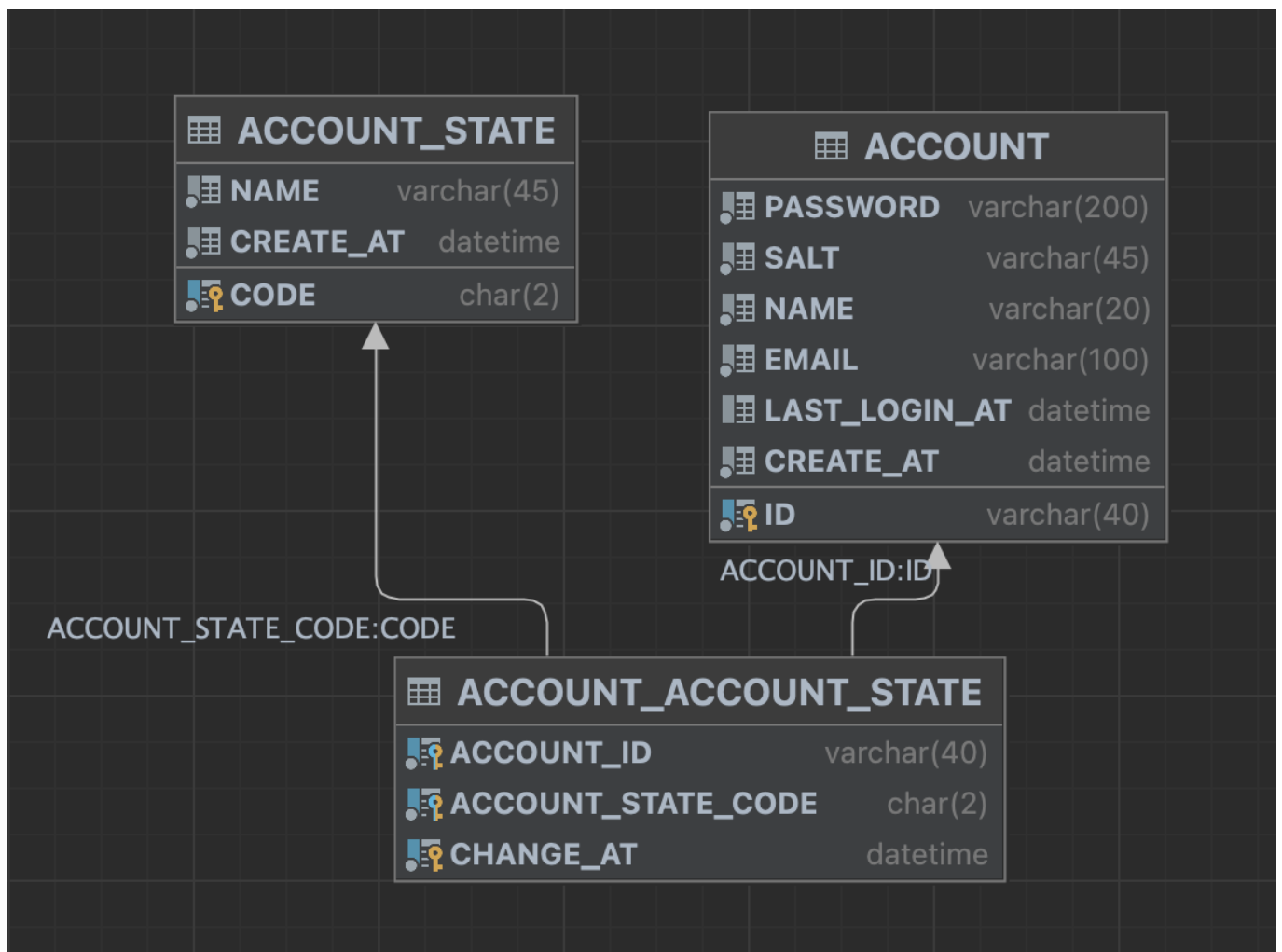
Content-Type : application/json

Host : localhost:3306

## 응답 메시지

# API Spec

샘플 /샘플/샘플/샘플



- 샘플1
- 샘플2

Request

- Parameters

항목명	항목명(영문)	항목크기	항목구분	샘플데이터	비고
샘플	sample	100	옵션	샘플	샘플비고

요청 예제

Response

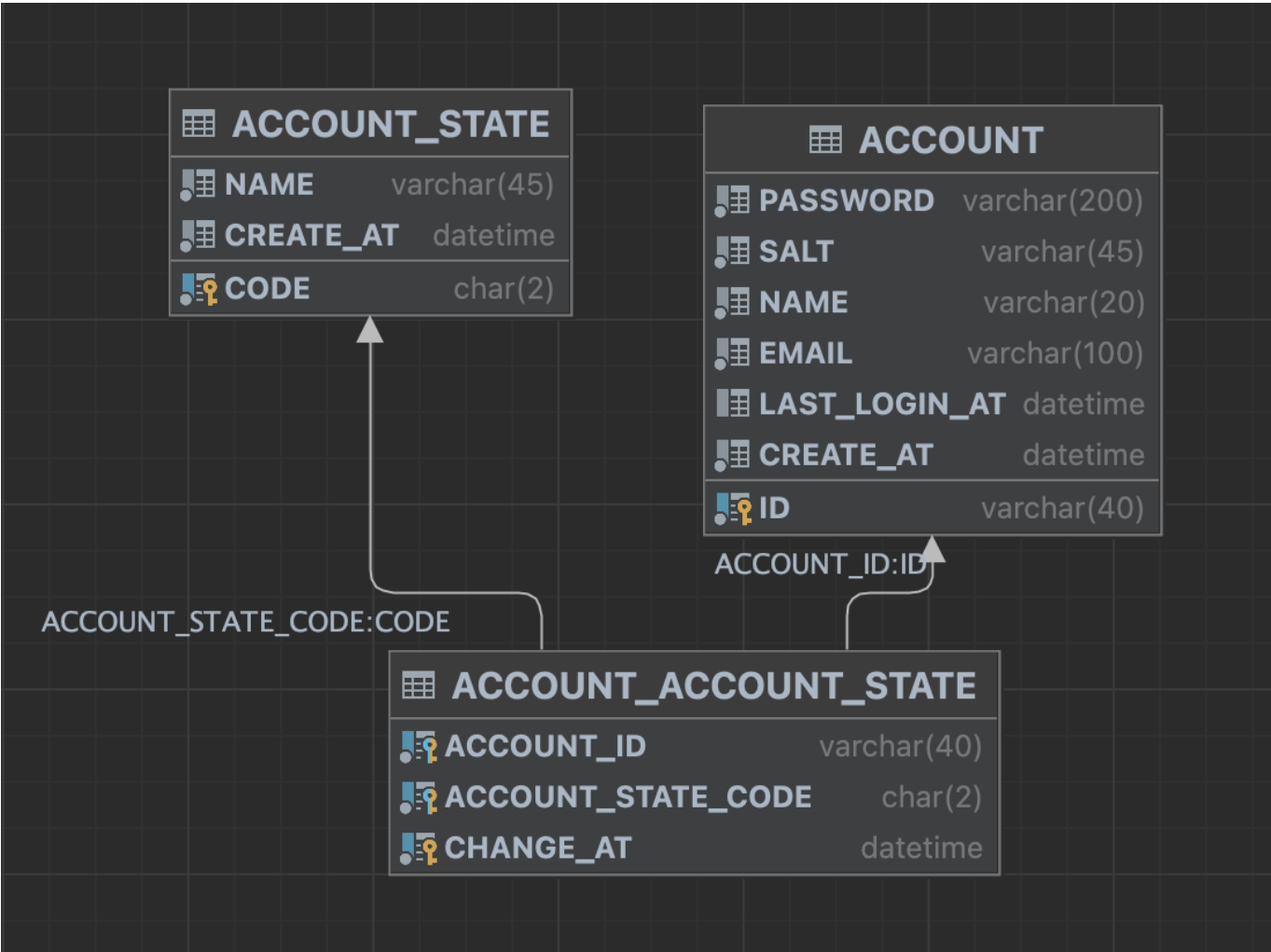
- Body

HTTP 응답코드

Common > Account

회원 조회

GET /common/accounts



- 전체 계정 조회로 각 응답은 대략적인 계정 정보만 반환합니다.
- 멤버 목록을 응답

Request

- Parameters

항목명	항목명(영문)	항목크기	항목구분	샘플데이터	비고
활동상태	accountStateCode	2	옵션	01	01: 가입, 02 :탈퇴, 03: 휴면, 04: 활동
사용자 이름	name	20	옵션	홍길동	회원 이름으로 검색
사용자 이메일	email	100	옵션	sample@dooray.com	회원 이메일로 검색
사용자 아이디	id	40	옵션	sampleId	회원 아이디로 검색
페이지	page	4	필수	1	페이지 번호
한 페이지 결과수	numOfRows	4	필수	10	한 페이지 결과 수

메모 : 사용자 이름, 사용자 이메일, 사용자를 추가하는 이유는 디테일 정보가 아닌 개별회원의 기본 정보만 출력하도록 하는것

## 요청 예제

```
GET /common/accounts?page=0&size=20
```

```
GET /common/accounts?accountStateCode={accountStateCode}&page=0&size=20
```

```
GET /common/accounts?name={name}&page=0&size=20
```

```
GET /common/accounts?email={email}&page=0&size=20
```

```
GET /common/accounts?id={id}&page=0&size=20
```

## Response

- Body

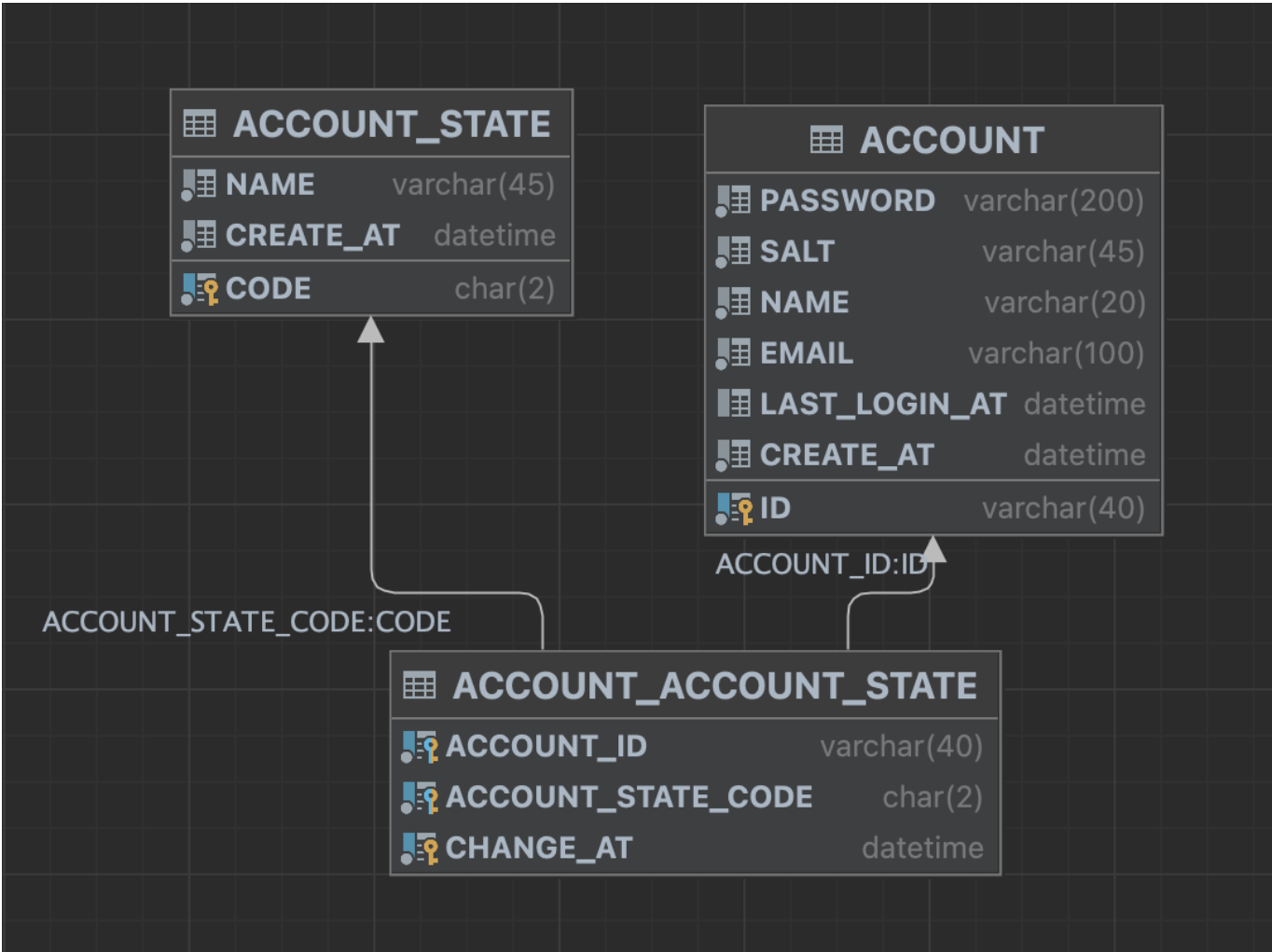
```
{
  "header":{
    "isSuccessful": true,
    "resultCode": 0,
    "resultMessage": ""
  },
  "result" : [{
    "id":"{id}", /* 두레이 사용자 아이디*/
    "name":"{name}", /* 사용자 이름*/
    "email":"{email}", /* 회원 이메일*/
    "accountAccountState":{
      "accountStateCode":"{accountStateCode}",
      "changeAt":"{changeAt}"
    }
  }],
  "totalCount":20
}
```

메모 : accountStateCode를 accountAccountState에서만 가져온다고 가정했는데, accountState에도 참조해야하는건지 헷갈려서 이렇게 넣어둡니다.

## HTTP 응답 코드

---

### GET /common/accounts/account



- 개별 멤버를 응답 - 멤버 id로 접근하며, 디테일한 멤버 정보를 응답합니다.

Request

- Parameters

항목명	항목명(영문)	항목크기	항목구분	샘플데이터	비고
사용자 아이디	id	40	필수	sampleId	회원 아이디로 검색

요청 예제

```
GET /common/accounts/account?id="id"
```

Response

- Body

```
{
  "header": {
    "isSuccessful": true,
    "resultCode": 0,
  }
}
```

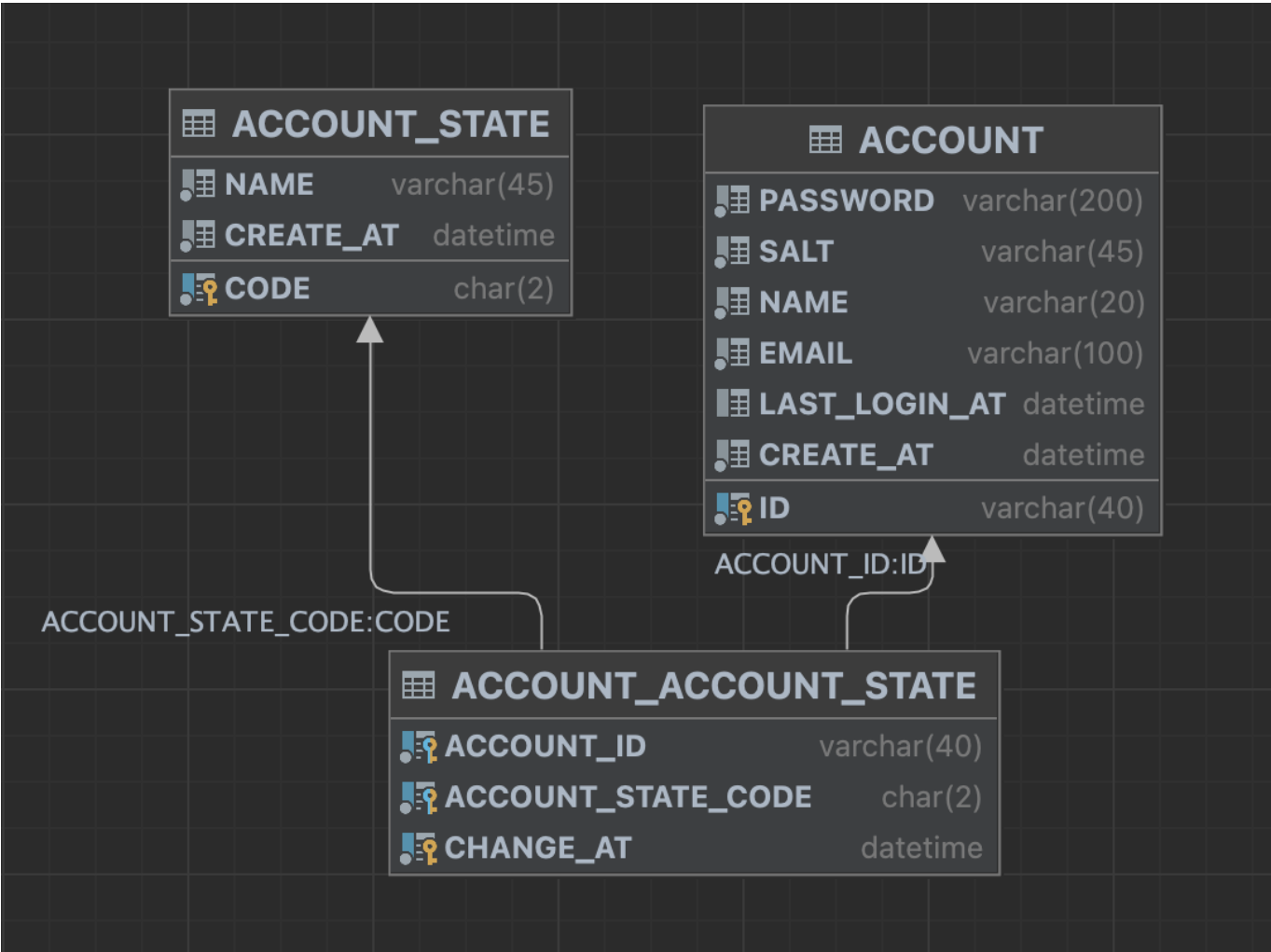
```
    "resultMessage": ""
  },
  "result" :{
    "id":"{id}", /* 두레이 사용자 아이디*/
    "name":"{name}", /* 사용자 이름*/
    "email":"{email}", /* 회원 이메일*/
    "lastLoginAt":"{lastLoginAt}", /*마지막 로그인 날짜*/
    "createAt":"{createAt}", /*회원 생성 날짜*/
    "accountAccountState":{
      "accountState":{
        "code":"{code}" /*accountStateCode*/,
        "name":"{name}" /*상태명*/
      },
      "changeAt":"{changeAt}" /*변경 날짜*/
    }
  }
}
```

#### HTTP 응답 코드

---

#### 회원 가입

#### POST /common/accounts/save



- 사용자 정보를 추가합니다
- ACCOUNT\_ACCOUNT\_STATE 에 가입 및 활성이 들어가야 한다.

항목명	항목명(영문)	항목크기	항목구분	샘플데이터	비고
사용자 아이디	id	40	필수	sampleId	사용자 아이디가 들어가는 항목
사용자 이름	name	20	필수	홍길동	사용자 이름이 들어가는 항목
사용자 이메일	email	100	필수	sample@dooray.com	사용자 이메일이 들어가는 항목
사용자 비밀번호	password	200	필수	1234	사용자 비밀번호가 들어가는 항목 암호화 알고리즘으로 변환되어 저장

Request

- Body

```
{
  "id": "sampleId", /* 사용자의 아이디 PK로 걸려있다. */
  "name": "홍길동", /* 사용자 이름 */
}
```



```

    "email": "sample@dooray.com", /* 사용자 이메일 */
    "password": "1234" /* 사용자 비밀번호. 암호화 알고리즘으로 변환되어 저장 */
  }

```

## Response

- Body

```

{
  "header": {
    "isSuccessful": true,
    "resultCode": 0,
    "resultMessage": "Account created successfully."
  },
  "result": {
    "id": "sampleId", /* 두레이 사용자 아이디 */
    "name": "홍길동", /* 사용자 이름 */
    "email": "sample@dooray.com", /* 회원 이메일 */
    "accountAccountState": {
      "accountState": {
        "code": "01", /* accountStateCode */
        "name": "가입" /* 상태명 */
      },
      "changeAt": "2023-06-02T13:46:02" /* 변경 날짜 */
    }
  }
}

```

> 메모 : RestController에서 accountAccountState와 accountState 부분을 잘 처리해야 함.

## HTTP 응답코드

### 회원 수정

**PUT /common/accounts/update/{account-Id}**

**DELETE /common/accounts/delete/{account-Id}**