

---

# REST API & ETL PIPELINE

15기 - 2팀

김경동

엄성호

우병일

# 목차

01. 프로젝트 개요
02. 프로젝트 팀 구성 및 역할
03. 프로젝트 수행 절차 및 방법
04. 프로젝트 수행 결과
05. 자체 평가 의견

# 1. 프로젝트 개요

---

- Github Link: [https://github.com/cp2-2team/cp2\\_api](https://github.com/cp2-2team/cp2_api)
- Document: [https://cp2de.gitbook.io/rest\\_api-and-etl-pipeline/](https://cp2de.gitbook.io/rest_api-and-etl-pipeline/)
- Deployment: <http://cp2de.duckdns.org/api/common/>
- Django REST Framework로 REST API 제작
- Airflow로 ETL PIPELINE 제작
- 프로젝트 최종목표: REST API의 로그 데이터를 ETL PIPELINE으로 서빙해서 ML등 다양한 분야에 활용
- 개발 환경: Windows, WSL, AWS EC2 Ubuntu 사용
- Github를 사용하고, Github의 Issue, Wiki를 활용하여 협업

## 02. 프로젝트 팀 구성 및 역할

---

훈련생	역할	담당 업무
김경동	팀장	<ul style="list-style-type: none"><li>▶ AWS 세팅 및 IAM 설정</li><li>▶ REST API 유닛 테스트 구현</li><li>▶ REST API AWS 배포</li></ul>
엄성호	팀원	<ul style="list-style-type: none"><li>▶ REST API 권한 설정 과 로그 구현</li><li>▶ 로그 데이터 암호화 구현</li></ul>
우병일	팀원	<ul style="list-style-type: none"><li>▶ REST API 집계 구현</li><li>▶ DUMMY DATA 생성 구현</li></ul>
[공통]	공동 작업	<ul style="list-style-type: none"><li>▶ DRF (Django REST Framework)로 REST API 제작</li><li>▶ Airflow로 ETL PIPELINE 제작</li></ul>

---

### 03. 프로젝트 수행 절차 및 방법

구분	기간	활동	비고
사전 기획	▶ 1/18(수) ~ 1/20(금)	▶ Django & DRF에 대한 사전 학습 ▶ REST API 설계에 대한 논의	▶ 프로젝트 준비
REST API 제작	▶ 1/23(월) ~ 1/24(화)	▶ User Model 설계, jwt 인증 구현 ▶ REST API 기초 제작 ▶ 권한 설정, 로그 구현	▶ 팀별 주간보고 실시
REST API 제작	▶ 1/25(수) ~ 1/31(화)	▶ 유닛 테스트 구현 ▶ 집계 기능 구현	▶ 팀별 주간보고 실시
REST API 제작	▶ 2/1(수) ~ 2/10(금)	▶ 로그데이터 암호화 및 API 제공 ▶ 테스트를 위한 더미데이터 기능 구현 ▶ AWS EC2, Nginx, Gunicorn 배포	▶ 팀별 최종보고 실시
ETL PIPELINE 제작	▶ 2/13(월) ~ 2/15(수)	▶ Airflow 사용 ETL PIPELINE 구현	▶ 팀별 최종보고 실시
총 개발기간	▶ 1/18(수) ~ 2/15(수) (총 4주)	-	-

## 04. 프로젝트 수행 결과

---

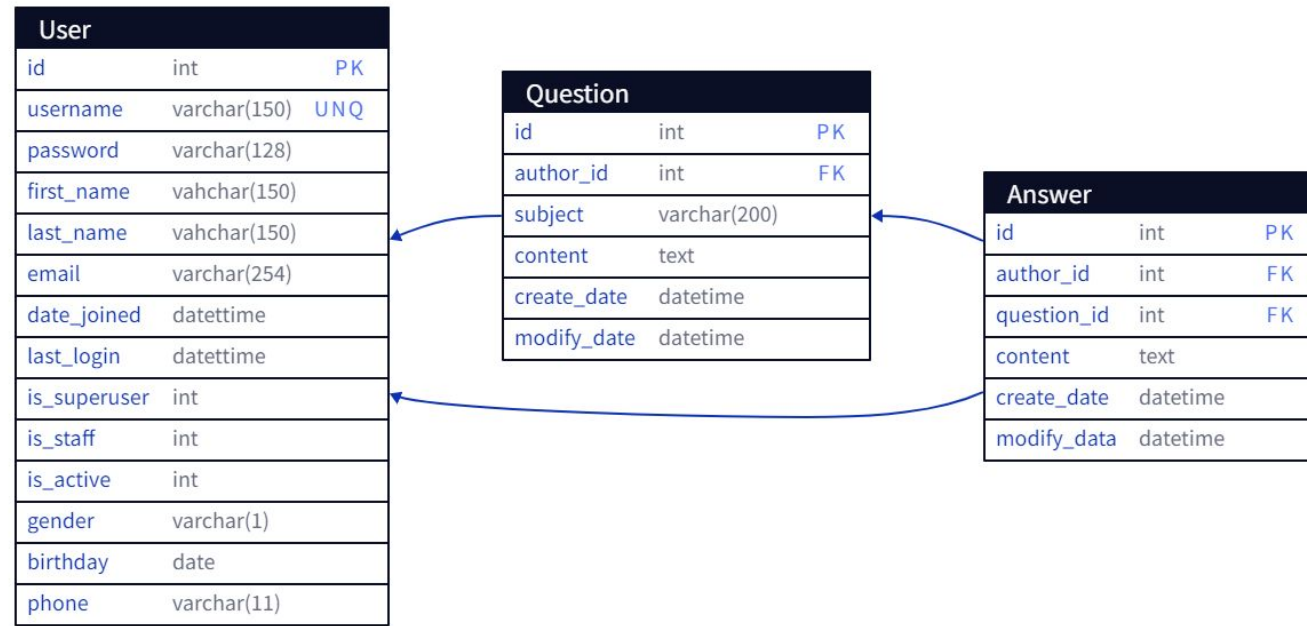
### ▶ [프로젝트 수행 결과]는 프로젝트 결과물이 도출된 과정을 세부적으로 기록

- 예시는 하나의 사례로 간단하게 제시한 것이므로 프로젝트의 성격에 따라 보다 자세하게 기록하며, 결과를 서술하는 과정에서는 활용된 기술(구현 방법), 핵심기능, 검증 결과\* 등을 상세히 기재한다.
- 프로젝트의 결과는 그 과정이 잘 드러날 수 있도록 가공 과정부터 활용까지 전체적인 프로세스를 확인할 수 있도록 단계별로 작성

\* 첨부 자료 예시: 결과물 사진, 시연 동영상 등 프로젝트의 우수성이 드러날 수 있는 자료

## 04. 프로젝트 수행 결과

### Django Model ERD



- Django의 User Model을 요구사항에 맞게 커스터마이징
- Question과 Answer는 FK가 참조하는 데이터가 삭제되면 자동으로 삭제

## 04. 프로젝트 수행 결과

REST API /common

Django REST framework

admin11 ▾

Api Root

Api Root

OPTIONS GET ▾

GET /api/common/

HTTP 200 OK  
Allow: OPTIONS, GET  
Content-Type: application/json  
Vary: Accept

```
{  
  "users": "https://cp2de.duckdns.org/api/common/users/",  
  "signup": "https://cp2de.duckdns.org/api/common/signup/",  
  "token": "https://cp2de.duckdns.org/api/common/token/",  
  "token/refresh": "https://cp2de.duckdns.org/api/common/token/refresh/",  
  "token/verify": "https://cp2de.duckdns.org/api/common/token/verify/"  
}
```

- 웹브라우저로 접속하면 Browsable API를 보여주고, curl을 사용하면 JSON으로 반환함
- 회원 가입, 회원 탈퇴, 회원목록(staff 계정만 가능) 기능
- jwt 발급, 확인, jwt refresh 기능(jwt는 대부분의 API를 사용할때 인증으로 사용)



## 04. 프로젝트 수행 결과

### REST API /board

Django REST framework admin11

Api Root / Questions

### Questions

OPTIONS GET

GET /api/board/questions/

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "count": 9,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 9,
      "subject": "이건 테스트입니다.",
      "content": "데이터 파티셔닝을 위한테스트!!!!",
      "create_date": "2023-02-14T13:53:35.673744",
      "modify_date": null,
      "author": 1
    },
  ],
}
```

Django REST framework admin11

Api Root / Questions / Question Detail

### Question Detail

DELETE OPTIONS GET

GET /api/board/questions/9/

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "id": 9,
  "subject": "이건 테스트입니다.",
  "content": "데이터 파티셔닝을 위한테스트!!!!",
  "create_date": "2023-02-14T13:53:35.673744",
  "modify_date": null,
  "author": 1
}
```

- 질문 목록, 질문 생성 기능, questions/<pk> 를 사용하면 상세조회, 수정, 삭제 기능
- 수정은 작성자 본인만 가능, 삭제는 본인 및 관리자 가능

## 04. 프로젝트 수행 결과

### REST API /log

Django REST framework admin11 ▾

Api Root / Logs

## Logs

OPTIONS GET ▾

« 1 2 3 »

GET /api/log/logs/

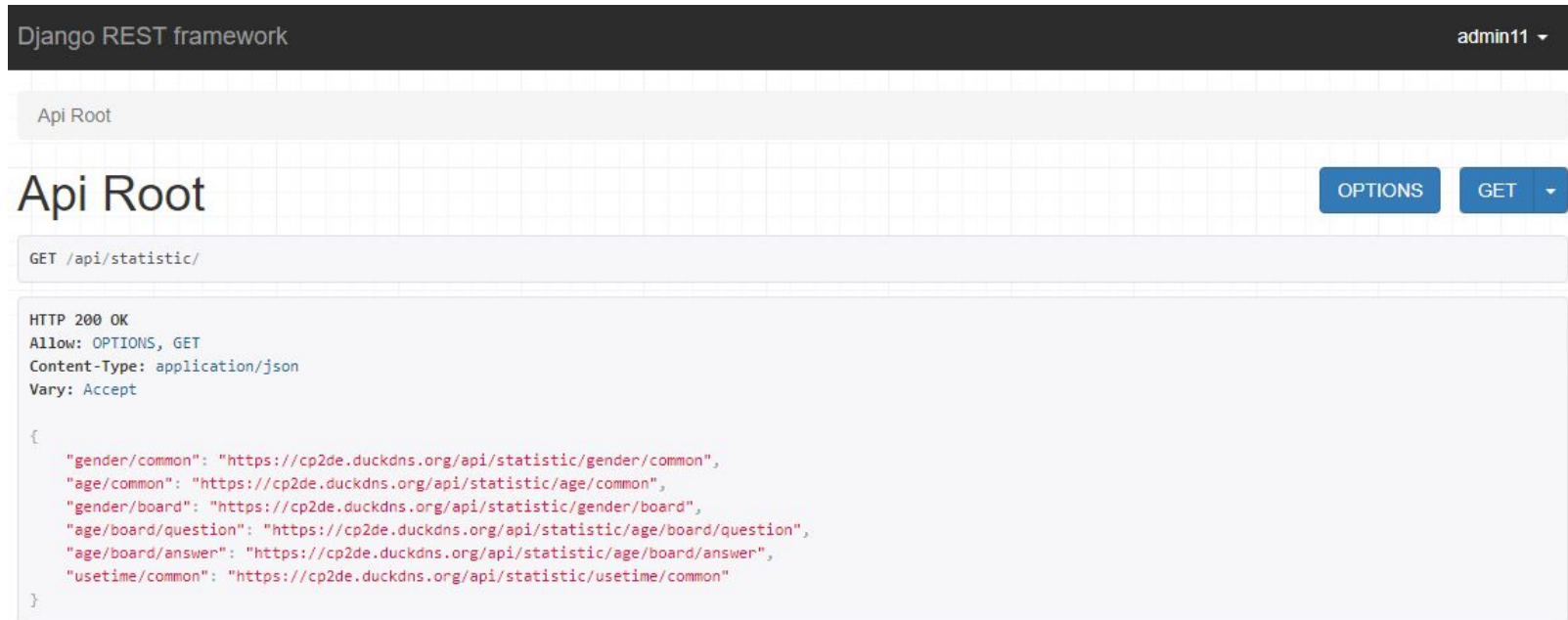
HTTP 200 OK  
Allow: GET, HEAD, OPTIONS  
Content-Type: application/json  
Vary: Accept

```
{
  "count": 22,
  "next": "https://cp2de.duckdns.org/api/log/logs/?page=2",
  "previous": null,
  "results": [
    {
      "id": 22,
      "recordId": "26467214560006254724948630290341895715",
      "time": 1676426784.6843293,
      "data": "gAAAAABj7D4gop2gsQrmds6hfLrbwIOLbfGyO3Sk4M1wTR6R7iS1K26sFqiz_jSZWFIpApY0CBHn5Ua_FGJPMLEzDSqVOQuY8F7INNodZ8te808KhNPEs6HsXxQ_h39M-zyzQb066MddU--d04KBhs:"
    }
  ]
}
```

- 회원의 활동에 대한 Log 제공, 관리자 계정만 접근 가능
- data는 대칭키 암호로 암호화 가능
- data를 복호화 해서 ETL PIPELINE에 사용

## 04. 프로젝트 수행 결과

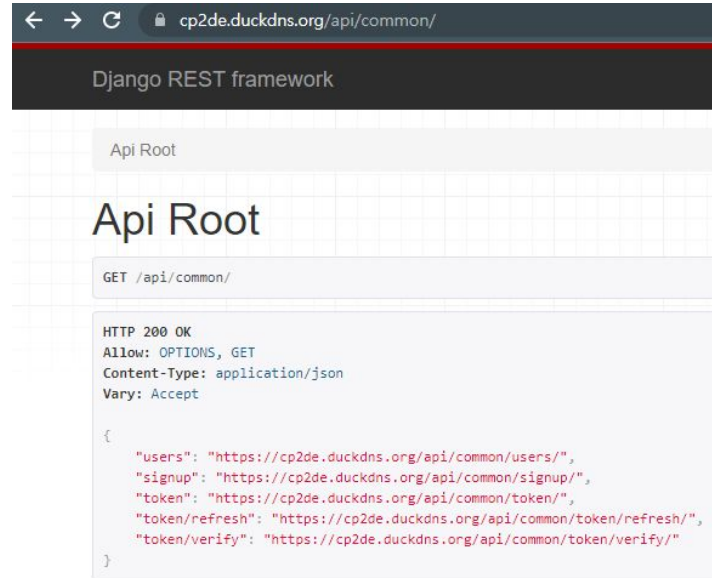
### REST API /statistic



- 실시간으로 회원들의 활동에 대한 통계자료 제공
- 성별, 나이, 사용 기간등을 활용 내용으로 구분하여 통계자료 제공

## 04. 프로젝트 수행 결과

### Deployment



- AWS EC2: 프로젝트를 구동하는 환경
- Nginx: 웹서버
- Gunicorn: Nginx와 Django의 미들웨어
- Let's Encrypt: Https 보안 연결

## 04. 프로젝트 수행 결과

### REST API test

---

```
$ python manage.py test
Found 20 test(s).
Creating test database for alias 'default'...
System check identified some issues:

WARNINGS:
?: (staticfiles.W004) The directory 'C:\Users\Eom\team\cp2\django\static' in the STATICFILES_DIRS setting does not exist
.
?: (urls.W005) URL namespace 'rest_framework' isn't unique. You may not be able to reverse all URLs in this namespace

System check identified 2 issues (0 silenced).
.....
-----
Ran 20 tests in 6.540s

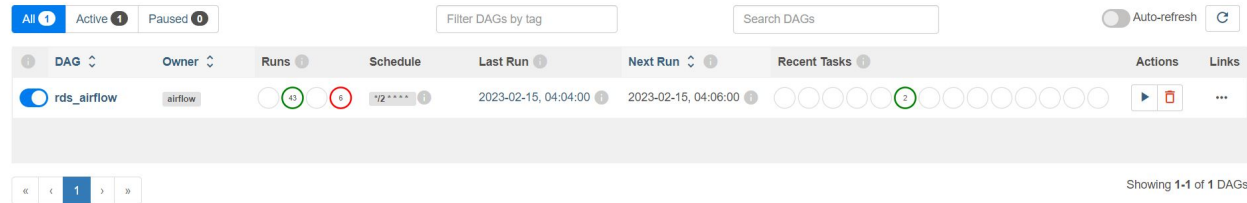
OK
Destroying test database for alias 'default'...
```

- DRF의 내부 테스트 기능 구현
- DRF의 앱을 수정하고 바로 내부 테스트 할수 있는 기능
- 각각의 앱에서 작성된 테스트를 수행하고 오류가 있는지 알 수 있음

## 04. 프로젝트 수행 결과

### ETL PIPELINE

#### DAGs



```
{ } 2023-02-15T11-18-10.json X
C: > Users > kimsdata > AppData > Local > Temp > _AZTMP1_ > { } 2023-02-15T11-18-10.json > ...
1 [{"url": 6, "method": 2, "user_id": "6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b", "username":
"kims", "REMOTE_ADDR": "", "created": "021523111601", "logger": "my_json", "message": "Question Create", "levelname":
"INFO", "subject": ["test db"], "content": ["ssssssssssssssssss"]}, {"url": 6, "method": 2, "user_id":
"6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b", "username": "kims", "REMOTE_ADDR": "", "created":
"021523111604", "logger": "my_json", "message": "Question Create", "levelname": "INFO", "subject": ["test db"], "content":
["ssssssssssssssssss2222222222222222"]}]
```

이름	유형	마지막 수정
2023-02-15T12-00-13.json.gz	gz	2023. 2. 15. pm 12:00:15 PM KST
2023-02-15T12-02-10.json.gz	gz	2023. 2. 15. pm 12:02:12 PM KST
2023-02-15T12-04-09.json.gz	gz	2023. 2. 15. pm 12:04:11 PM KST
2023-02-15T12-06-09.json.gz	gz	2023. 2. 15. pm 12:06:11 PM KST
2023-02-15T12-08-09.json.gz	gz	2023. 2. 15. pm 12:08:11 PM KST
2023-02-15T12-10-11.json.gz	gz	2023. 2. 15. pm 12:10:13 PM KST
2023-02-15T12-12-08.json.gz	gz	2023. 2. 15. pm 12:12:10 PM KST
2023-02-15T12-14-10.json.gz	gz	2023. 2. 15. pm 12:14:11 PM KST
2023-02-15T12-16-09.json.gz	gz	2023. 2. 15. pm 12:16:14 PM KST
2023-02-15T12-18-09.json.gz	gz	2023. 2. 15. pm 12:18:11 PM KST
2023-02-15T12-20-09.json.gz	gz	2023. 2. 15. pm 12:20:11 PM KST
2023-02-15T12-22-09.json.gz	gz	2023. 2. 15. pm 12:22:11 PM KST
2023-02-15T12-24-10.json.gz	gz	2023. 2. 15. pm 12:24:12 PM KST
2023-02-15T12-28-09.json.gz	gz	2023. 2. 15. pm 12:28:11 PM KST
2023-02-15T12-32-07.json.gz	gz	2023. 2. 15. pm 12:32:09 PM KST

- Airflow를 사용하여 REST API의 로그데이터를 추출, 변환(압축), AWS S3에 적재
- Airflow를 AWS로 배포하려고 시도했지만, 인스턴스의 사양 문제로 로컬 실행으로 변경
- Airflow에 대한 학습(Executor, Dags, callback)등 으로 개선 예정

## 05. 자체 평가 의견

---

### 김경동

#### 잘한 부분

- 다양한 기술 **stack**을 사용해보고 이를 통해 A부터 Z까지 과정을 잘 반영함

#### 아쉬운 점

- AWS **lambda** 등 다른기술에 대해 좀 더 알아보고 필요한 기능에 따라 선택해서 했으면 더 좋았을 것 같음 (시간적요소, 비용 등)

### 엄성호

#### 잘한 부분

- **DRF**의 기능을 최대한 활용해서 만족스러운 **REST API**를 제작함

#### 아쉬운 점

- **Airflow**를 제대로 사용하기 위해 추가 학습이 필요해 보임

### 우병일

#### 잘한 부분

- **REST API** 기능 구현과 **Airflow** 라는 기술을 사용해 보고 구현한 것

#### 아쉬운 점

- 좀더 원활한 개발을 위한 추가 학습과 ‘스택을 경험해 봤다’ 이후에 추가적인 기능 추가로 프로젝트 가치를 늘리는것이이 필요하다고 생각됨