



# A203 ON:AIR

삼성청년 SW 아카데미 서울캠퍼스 5 기

특화프로젝트 (6 주; 2021/08/30 ~ 2021/10/08)

# 목차

1. 기술 스택
2. 빌드 상세
3. 배포 상세
4. DB 계정
5. 프로퍼티 정의
6. 외부 서비스

## 알고 타자! 믿고 타자! 같이 타자!

### 오늘의 항공사 분석 및 리뷰 서비스 온:에어

위드 코로나(With-Corona) 시행에 대한 기대감과 함께 해외여행 계획을 짜고 있는 사람들이 늘고 있다. On:Air(온에어)는 이러한 기대감에 힘입어 실제 이용객이 항공사를 선택하고, 평가하는 데 도움을 줄 수 있도록 개발되었다.

On-Air(온에어)는 빅데이터를 활용하여 소비자가 항공사 별 서비스 품질을 시각화한 통합 웹 서비스이다. 데이터를 가공, 분석하여 지연률, 결항률 통계를 시각화하여 제공하며 현재 운행중인 비행기의 지연 시간을 다각도로 예측하여 예비 이용객이 참고할 수 있도록 하였다. 또한 항공사에 대한 리뷰를 남기고 리뷰를 기반으로 한 실사용자의 항공사 별 평가를 감정 분석하여 제공한다.

# 포팅 매뉴얼

## 프로젝트 기술 스택

1. 이슈관리 : Jira
2. 형상관리 : Gitlab
3. 커뮤니케이션 : Webex, Mattermost, Notion, Slack
4. 개발 환경
  - a. OS : Windows 10
  - b. IDE
    - i. Visual Studio Code 1.60.2
    - ii. UI/UX : Adobe XD
  - c. Database : MySQL Workbench 8.0.22
  - d. Server : AWS EC2 (MobaXterm)
    - i. Ubuntu 20.04.2 LTS
    - ii. Docker 20.10.7, Nginx 1.18.0, Gunicorn 20.1.0, Jenkins 2.303.1
5. 상세 사용
  - a. Backend
    - i. Django 3.2.7, Django Rest Framework 3.12.4
    - ii. pandas 1.3.3, scikit-learn 1.0, statsmodels 0.12.2, pykomoran 0.1.6.post1
  - b. Frontend
    - i. HTML5, CSS3, JavaScript(ES6)
    - ii. Vue 2.6.11, Vuex 3.4.0
    - iii. Node.js 14.16.1, NPM 6.14.12
    - iv. Vuesax 3.12.2, Highcharts 9.2.2

## 빌드 상세

1. 프론트엔드 빌드
  - a. node\_modules를 위한 기본 install

```
npm install
```
  - b. 빌드 (dist 폴더 생성)

```
npm run build
```
2. 백엔드
  - a. Django는 별도의 빌드 과정 없이 WSGI를 통해 바로 배포

## 배포 상세

1. AWS EC2에 프론트엔드 빌드 후 만들어진 dist 폴더와 백엔드 폴더를 업로드
2. 프론트엔드 nginx 실행

```
sudo service nginx start
```

3. 백엔드 환경변수 설정(아래 내용의 .env 파일을 백엔드 프로젝트 폴더 하위에 생성)

```
SECRET_KEY=django-insecure-m*87z@xcq_-uva8q-%*kj0x3ewt%&#bul!96i$5$xnj!oih3$a
MYSQL_USER=onair
MYSQL_PASSWORD=onair1234
JWT_SECRET_KEY=onair
WEATHER_API_KEY=61ec55794a9d13e3863d3dee3c5947d2
```

4. 백엔드 실행

- a. 현재 구동 중인 백엔드 서버가 있는지 확인

```
netstat -ntlp
```

- b. 구동 중인 프로세스 종료

```
kill -9 <PID>
```

- c. 백엔드 서버 구동(백엔드 프로젝트 폴더 위치로 이동 후)

```
source venv/bin/activate
sudo pip install -r requirements.txt
gunicorn --bind 0.0.0.0:8000 server.wsgi:application --daemon
```

5. Jenkins의 pipeline을 이용한 자동 배포 라인 구축

- a. 자동 배포 상세 과정 설명은 Document/5\_Deployment/1\_AutoDeploy/배포.md에 있습니다.

- b. pipeline 코드

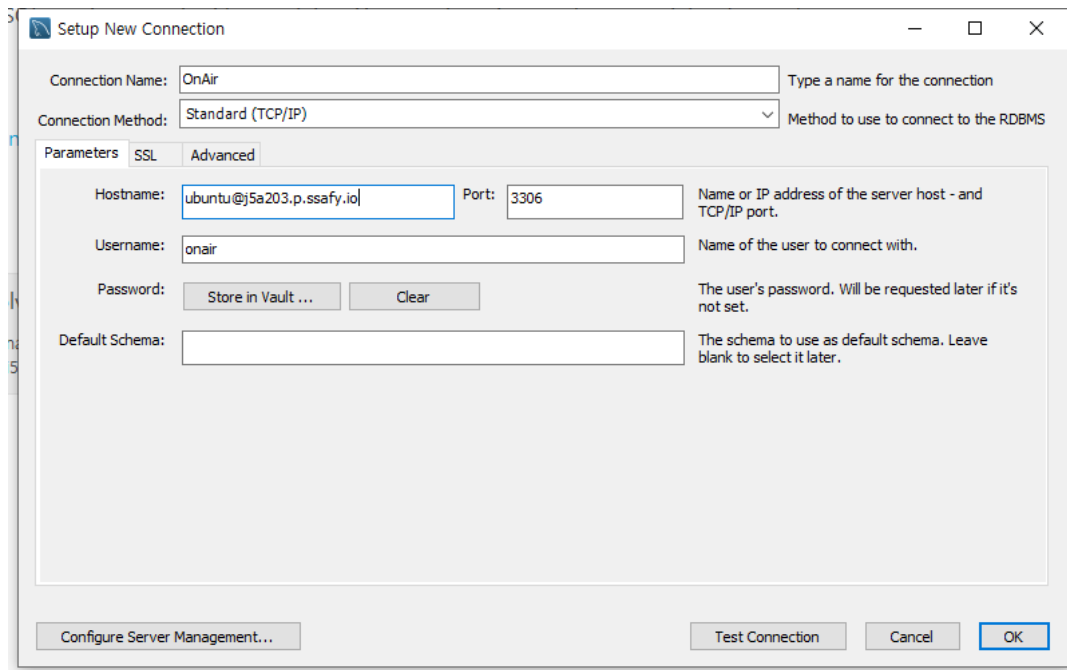
```
pipeline {
    agent any // 삭제하지 말 것
    tools {
        // 도구명 사용 선언
        nodejs "node"
        git 'Default'
    }

    stages {
        stage('prepare') {
            steps {
                git branch: 'master', credentialsId: 'onair-gitlab', url: 'https://lab.ssafy.com/s05-bigdata-dist/S05P21A203.git'
                dir('Project/client') {
                    sh 'npm install'
                }
                sh 'cp -r /var/jenkins_home/workspace/onair_pipeline/Project/server/* /var/jenkins_home/dist/django'
            }
        }
        stage('build') {
            steps {
                dir('Project/client') {
                    // 빌드 작업
                    sh 'npm run build'
                }
            }
        }
        post {
            success {
            }
        }
    }
}
```

```
// 빌드 성공 후 작업 -> 파이프라인 워크 스페이스에서 EC2 dist로 복사
sh 'cp -r /var/jenkins_home/workspace/onair_pipeline/Project/client/dist /var/jenkins_home/dist'
}
```

## DB 계정

### 1. MySQL WorkBench 추가하기



username : onair, password : onair1234

root 계정이 아닌 별도의 팀 계정을 만들어서 진행했습니다.

## 프로퍼티 정의

### 1. nginx 세팅 (/etc/nginx/sites-available/front.conf)

```
server {

    root    /home/ubuntu/onair/dist/dist;
    index  index.html index.htm;
    server_name j5a203.p.ssafy.io;
    location / {
        try_files $uri $uri/ /index.html;
    }

    location /api/v1/ {
        proxy_pass http://localhost:8000/api/v1/;
    }

    location /static/ {
        alias /home/ubuntu/onair/dist/django/my_static/;
    }

    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/j5a203.p.ssafy.io/fullchain.pem; # managed by Certbot
```

```

ssl_certificate_key /etc/letsencrypt/live/j5a203.p.ssafy.io/privkey.pem; # managed by Certbot
include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}

server {
    if ($host = j5a203.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80 default_server;
    listen [::]:80 default_server;

    server_name j5a203.p.ssafy.io;
    return 404; # managed by Certbot
}

```

## 2. EC2 MySQL 세팅

```

sudo apt-get update
sudo apt-get install mysql-server # mysql 설치
sudo systemctl start mysql.service # mysql 구동
ps -ef | grep mysql # 구동 확인
cd /etc/mysql/mysql.conf.d
sudo vi mysqld.cnf # mysql 설정 변경 : bind-address의 값을 `0.0.0.0` (any)로 수정
sudo service mysql restart # mysql 재시작
sudo mysql -u root -p # root로 실행

# mysql에서 실행
CREATE USER 'username'@'%' IDENTIFIED BY 'password'; # username과 password 설정
GRANT ALL PRIVILEGES ON *.* TO 'username'@'%'; # username에 권한주기
FLUSH PRIVILEGES; # 사용자 추가, 삭제, 권한 변경 등을 바로 반영

```

## 외부 서비스

### 1. 구글 로그인


- 추가로 받아야 하는 사용자 정보가 없었기 때문에 로그인 및 회원가입을 간편화하기 위해서 구글 소셜 로그인을 사용했습니다.
- a. 아래 링크로 들어가서 새 프로젝트를 생성한다.

#### Google Cloud Platform

Google Cloud Platform lets you build, deploy, and scale applications, websites, and services on the same infrastructure as Google.

 <https://console.cloud.google.com/apis/dashboard>

## 새 프로젝트

 projects 할당량이 23개 남았습니다. 할당량 증가를 요청하거나 프로젝트를 삭제하세요. [자세히 알아보기](#)


[MANAGE QUOTAS](#)

프로젝트 이름 \* ?

onair

프로젝트 ID: onair-328302입니다. 나중에 변경할 수 없습니다. [수정](#)

위치 \*

 조직 없음 [찾아보기](#)

상위 조직 또는 폴더

만들기
취소

### b. 사용자 인증 정보 만들기

API 및 서비스

-  대시보드
-  라이브러리
-  사용자 인증 정보
-  OAuth 동의 화면
-  도메인 확인
-  페이지 사용 동의

사용자 인증 정보 [+ 사용자 인증 정보 만들기](#)  삭제

사용 설정한 API에 액세스하려면 사용자 인증 정보를 만드세요. [자세히 알아보기](#)

 애플리케이션에 대한 정보를 포함하여 OAuth 동의 화면을 구성해야 합니다. [동의 화면 구성](#)

API 키				
<input type="checkbox"/>	이름	생성일 ↓	제한사항	키
표시할 API 키가 없습니다.				

OAuth 2.0 클라이언트 ID				
<input type="checkbox"/>	이름	생성일 ↓	유형	클라이언트 ID
표시할 OAuth 클라이언트가 없습니다.				

서비스 계정	
<input type="checkbox"/>	이메일
표시할 서비스 계정이 없습니다.	

[+ 사용자 인증 정보 만들기](#)  삭제

**API 키**

할당량과 액세스 권한을 확인하기 위해 간단한 API 키로 프로젝트를 확인합니다.

**OAuth 클라이언트 ID**

앱에서 사용자 데이터에 액세스할 수 있도록 사용자 동의를 요청합니다.

**서비스 계정**

로봇 계정을 사용하여 서버 간의 앱 수준 인증을 사용 설정합니다.

**사용자 인증 정보 선택 도움말**

사용할 사용자 인증 정보의 유형을 결정할 수 있도록 몇 가지 질문을 합니다.

### c. OAuth 동의 — 도메인 설정

## OAuth 동의 화면

대상 사용자를 비롯해 앱을 구성하고 등록하려는 방식을 선택하세요. 프로젝트에는 하나의 앱만 연결할 수 있습니다.

### User Type

☐ 내부 ?

조직 내 사용자만 사용할 수 있습니다. 인증을 위해 앱을 제출할 필요는 없습니다. [사용자 유형 자세히 알아보기](#)

☒ 외부 ?

Google 계정이 있는 모든 테스트 사용자가 사용할 수 있습니다. 앱이 테스트 모드로 시작되고 테스트 사용자 목록에 추가된 사용자에게만 제공됩니다. 앱을 프로덕션에 푸시할 준비가 되면 앱을 인증해야 할 수도 있습니다. [사용자 유형 자세히 알아보기](#)

만들기

### 승인된 자바스크립트 원본 ?

브라우저 요청에 사용

URI \*

https://j5a203.p.ssafy.io

http://localhost:8000

http://localhost:8080

http://localhost:8081

+ URI 추가

d. 클라이언트 ID 확인 및 Callback URI 설정



## 승인된 리디렉션 URI ?

웹 서버의 요청에 사용

URI \*

http://localhost:8000

http://localhost:8000/api/v1/auth/login

http://localhost:8000/api/v1/auth/login/callback

+ URI 추가

### e. 구글 id\_token 발급받기

- i. 구글 API의 `getAuthResponse()` 메소드를 통해서 구글 id\_token을 발급받는다.

### f. id\_token으로 사용자 정보 요청 및 로그인/회원가입 수행

- i. 위에서 발급받은 id\_token을 [https://oauth2.googleapis.com/tokeninfo?id\\_token={id\\_token}](https://oauth2.googleapis.com/tokeninfo?id_token={id_token})에 포함시켜서 `get` 요청을 보낸다.
- ii. 응답으로 온 `sub`를 가지고 user 모델에 사용자의 존재 여부를 확인한다.
- iii. 존재한다면 로그인을, 존재하지 않는다면 회원가입을 진행한다.