

# PORTFOLIO

윤서율

# Contents

---

001/

**Profile**

002/

**Activity**

003/

**Project**

# 1. Profile



윤서율

1993. 01. 03

취미 : e-book 읽기

특기 : 빠르게 친해지기

Web : <https://resourcesaga.github.io>

Github : <https://github.com/resourceSaga>

TIL : <https://github.com/resourceSaga/TIL>

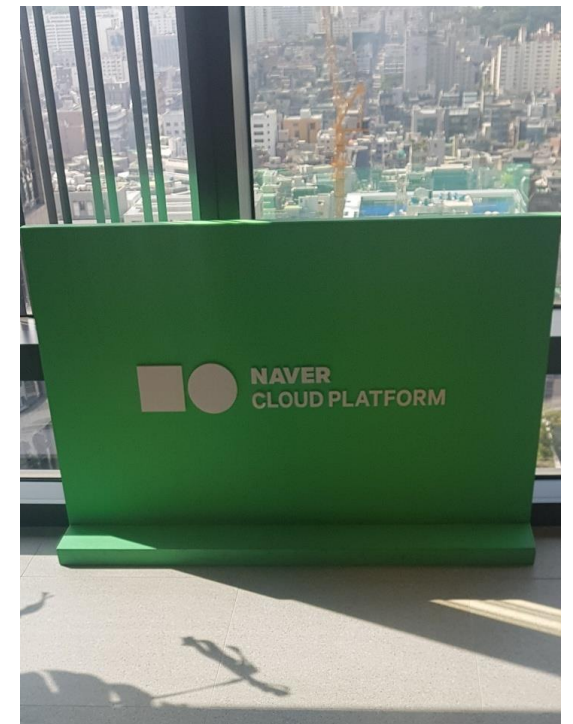
Interests :



## 2. Activity

Java를 시작으로 개발을 하며 컴퓨터공학의 길을 걷기 시작하였습니다. 웹 개발을 하며 프로그래밍에 재미를 붙이고 운영을 시도하며 자연스럽게 인프라까지 관심을 가지게 되었습니다.

AWS 에서 주최한 컨퍼런스를 참가하여 인프라의 확장과 사용 사례를 직접 보고 익히며, AWSome Day, aws summit, NCP Hands-On 교육에 참여하는 등 클라우드 플랫폼 서비스에 관심을 가지고 있습니다.  
또한, 클라우드 환경에서 Docker, k8s를 이용한 자동화로 효율적인 개발을 지향하는 DevOps의 실현을 꿈꾸고 있습니다.



# 3. Project

## 전국 학교 식단 챗봇

| 주제 및 기획의도

| 개발환경

| 배포 인프라

| 상세 구성

백식이  
(개인 프로젝트)

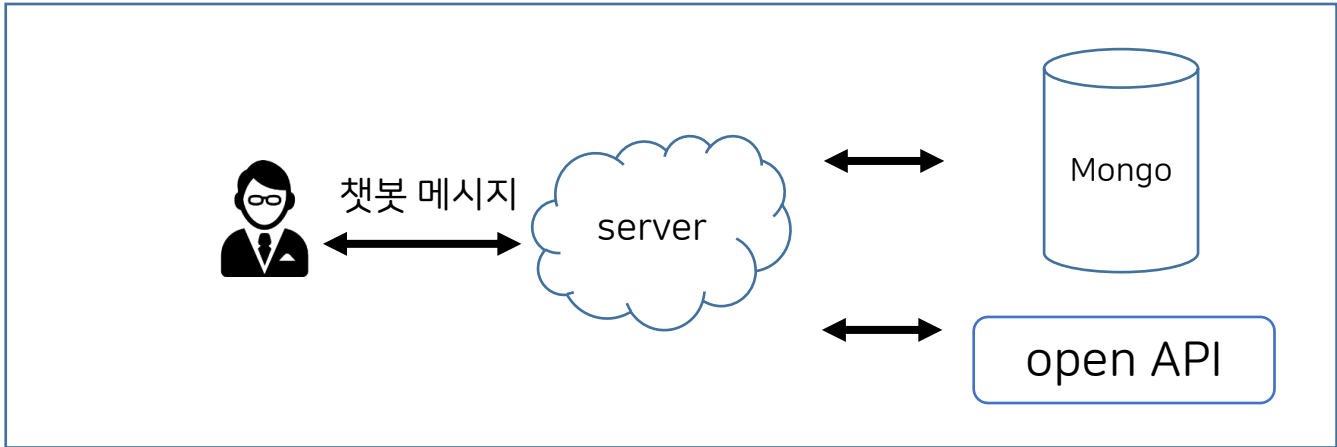


# 주제 및 기획 의도

## 전국 학교의 식단 메뉴를 알려주는 챗봇 서비스

- 사용자가 전국 학교 목록 open API 를 이용해서 학교 정보를 등록하고 등록된 학교 정보를 이용해서 전국 급식 open API를 이용해 식단 데이터를 받아 제공
- API를 이용하고, 간단한 챗봇 통신이기 때문에 서버가 필요 없다고 생각하여 서버리스 통신을 기획

### 기획의도

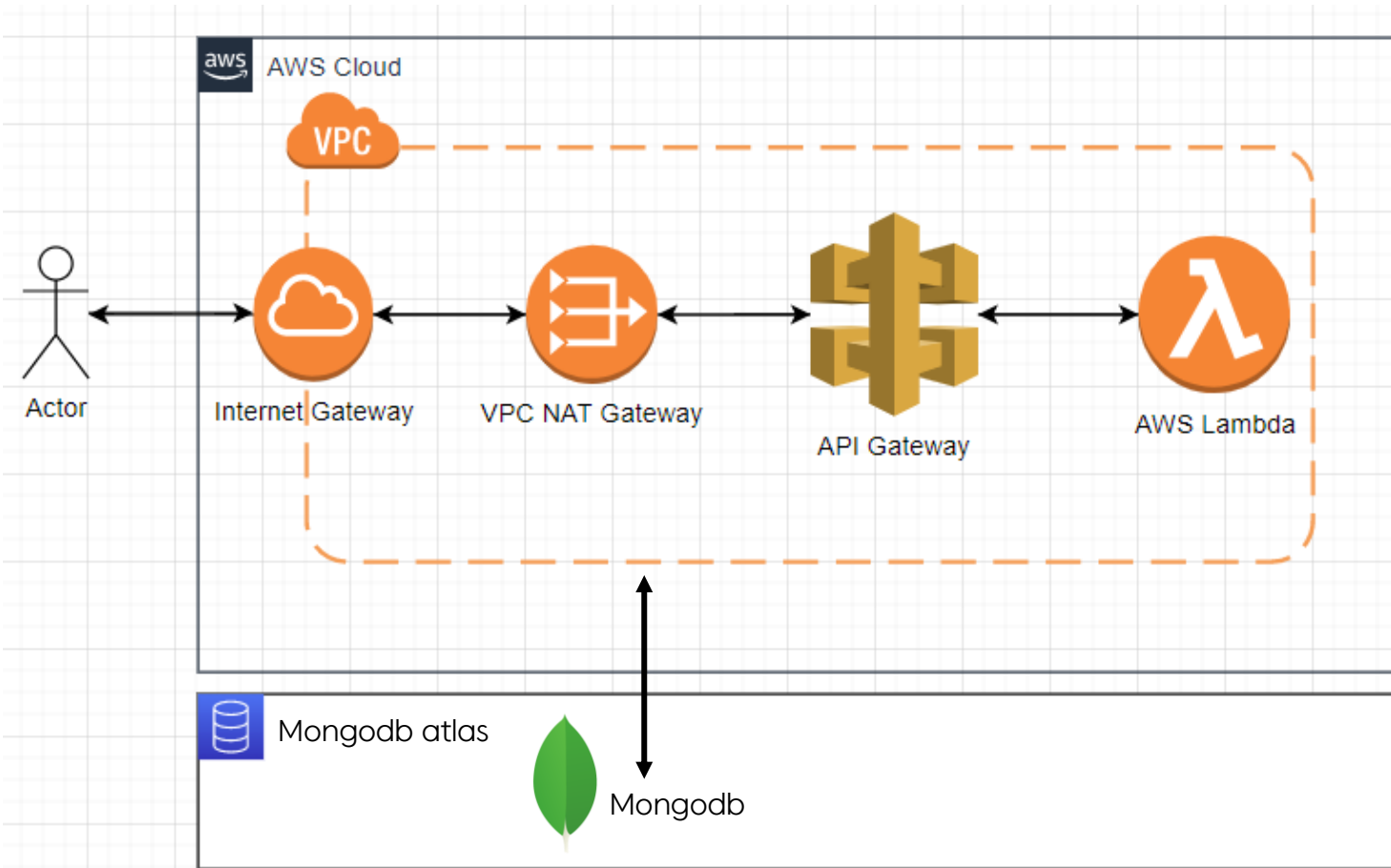


- 사용자는 챗봇 메시지를 통해 지역정보, 학교정보를 등록합니다.
- 등록된 사용자 데이터를 바탕으로 서버는 open API를 이용해 식단 정보를 제공합니다.
- 챗봇과 데이터베이스, open API는 AWS Lambda 를 통해 서버리스로 통신합니다.

# 개발 환경

프로젝트 기간	2020. 09. 28 ~ 2020. 10. 09 (14일간)
개발 인원	1명
개발플랫폼	Windows 10
사용언어	Python 3.7.4
사용기술	Python, MongoDB, Open API
배포	AWS Lambda
Repository	github.com/resourceSaga/lambda_school_meal

## 배포 인프라



한 개의 APC 영역을 이용해 Private 서브넷과 Public 서브넷을 구분하였습니다.

Public 서브넷은 Internet Gateway를 연결하였으며, private 서브넷은 내부망만 연결되어 있습니다.

이때, Labmda 는 private 서브넷을 할당받아 연결되어있지만, AWS Lambda와 외부 DB인 Mongoddb atlas와 연동하기 위해서 VPC NAT Gateway를 이용하였습니다.

AWS Lambda의 작동 이벤트 트리거는 API Gateway를 이용해서 REST 요청으로 처리하였습니다.

# Lambda 구성

Lambda > 함수

함수 (6)

마지막으로 가져온 항목 지금

작업

함수 생성

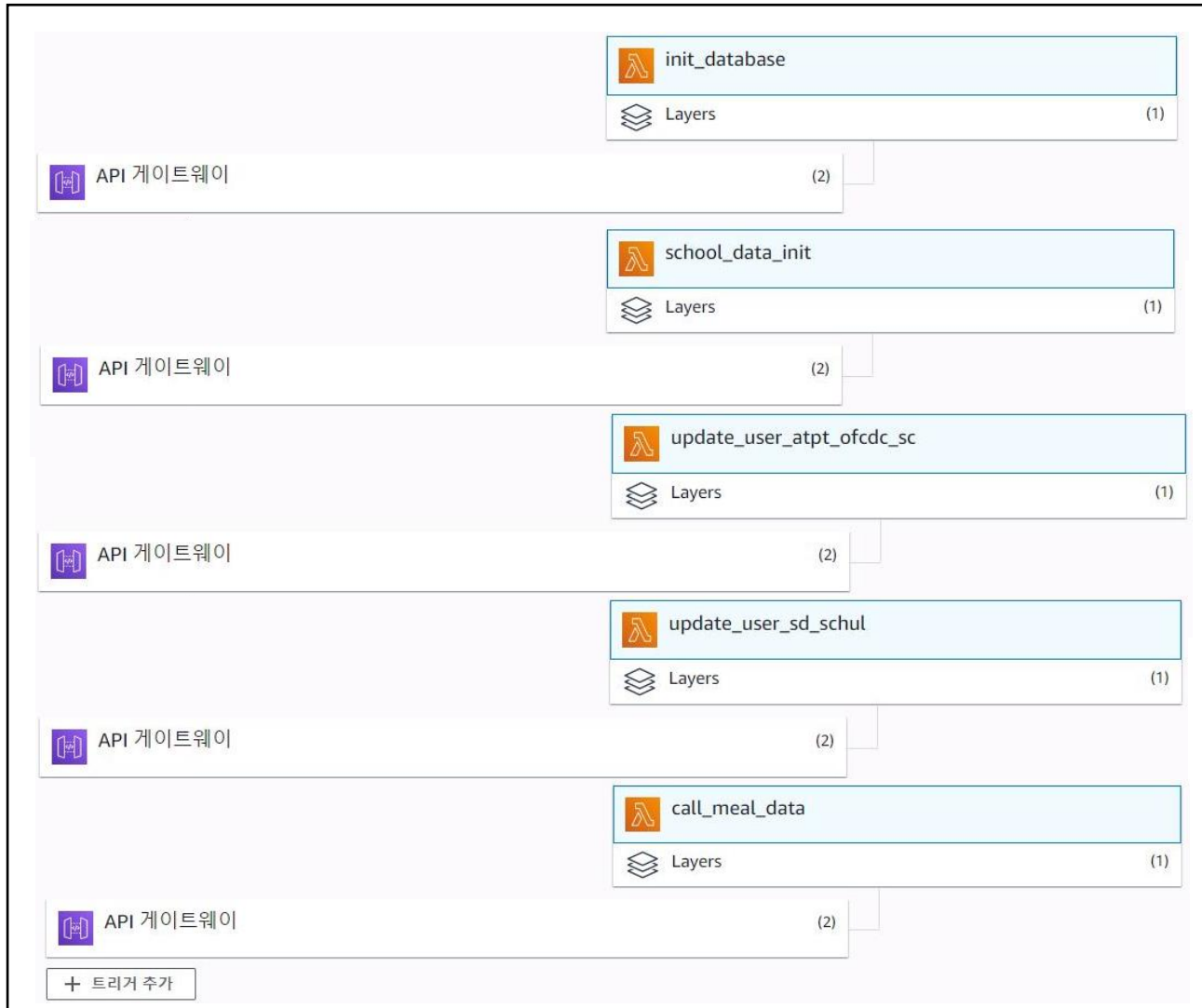
태그 및 속성별 필터 또는 키워드별 검색

< 1 > ⚙

함수 이름	설명	런타임	코드 크기	마지막 수정
<input type="radio"/> call_meal_data		Python 3.7	3.1 kB	2시간 전
<input type="radio"/> init_database		Python 3.7	2.7 kB	2시간 전
<input type="radio"/> lambdaTest		Python 3.7	3.0 kB	12분 전
<input type="radio"/> school_data_init		Python 3.7	3.1 kB	2시간 전
<input type="radio"/> update_user_atpt_ofcdc_sc		Python 3.7	3.0 kB	2시간 전
<input type="radio"/> update_user_sd_schul		Python 3.7	3.3 kB	2시간 전



# Lambda 구성 상세



## init\_database

관리자용 함수로 다른 기능을 위한 기본값 데이터베이스를 초기화합니다.

## school\_data\_init

사용자의 정보 등록을 위한 전국 학교 데이터를 갱신합니다.

## update\_user\_atpt\_ofcdc\_sc

식단 데이터 호출을 위한 사용자 지역 데이터를 업데이트 합니다.

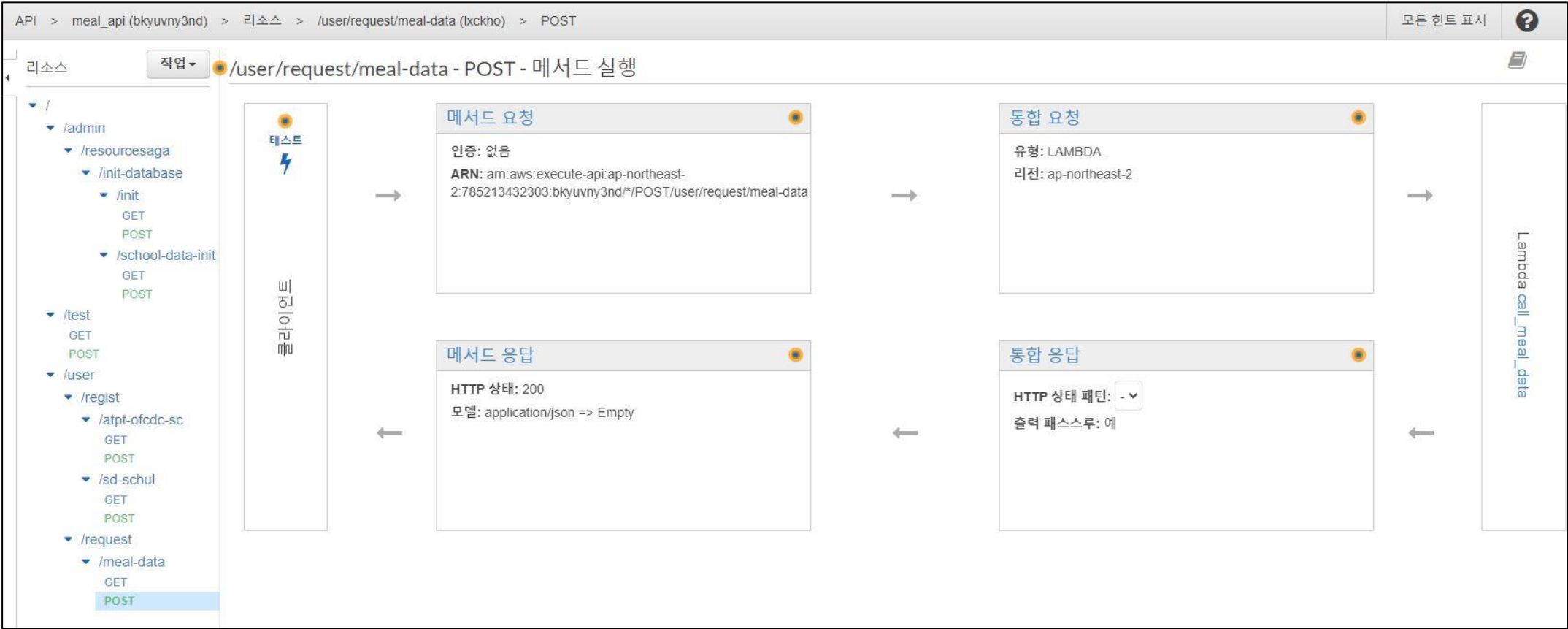
## update\_user\_sd\_schul

지역 데이터를 바탕으로 사용자의 학교 데이터를 업데이트 합니다.

## call\_meal\_data

식단 정보를 호출합니다.

# API Gateway 상세



- /admin/resourcesaga : key 인증을 이용해 관리자용 기능을 구현하였습니다.
- /user/regist : 유저 정보를 등록합니다.
- /user/request : 식단 정보를 요청합니다.

# 코로나 상황 페이지

| 주제 및 기획의도

| 개발환경

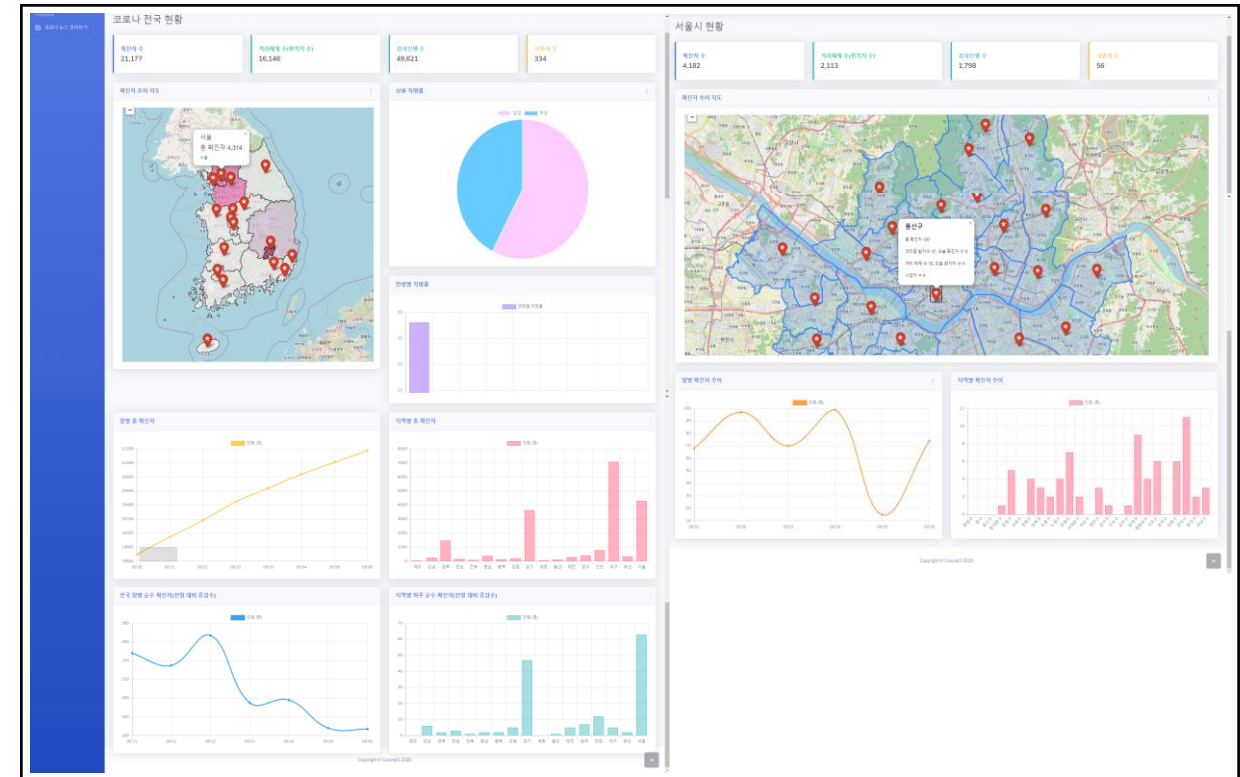
| 배포 인프라

| 주요기능 및 담당업무

| 후기

## CORONAIS

(3인 팀 프로젝트)

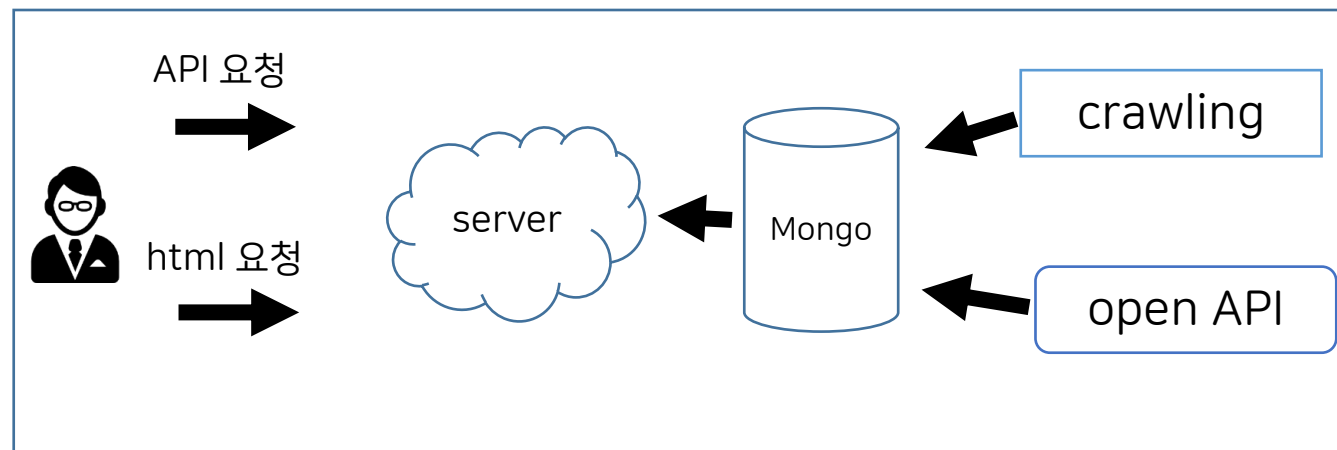


# 주제 및 기획 의도

## 공공데이터의 가공을 통한 코로나 상황 페이지

- 각 시 자체에서 제공하는 데이터는 확진자, 사망자 등 종합데이터만 제공할 뿐 구체적이지 못한 문제점을 인지함, 문제는 공공데이터 포털의 Open API의 데이터에 원인이 있음을 발견
- 공공데이터를 사용하지 않는 시/구 지자체 홈페이지에서는 보다 상세하게 제공하지만, 데이터가 산개 되어 있어 보기 힘든 문제점을 인식
- 두 데이터를 가공하여 시각화 한다면 유용할 것이라는 생각에 웹 페이지를 제작함

### 기획의도

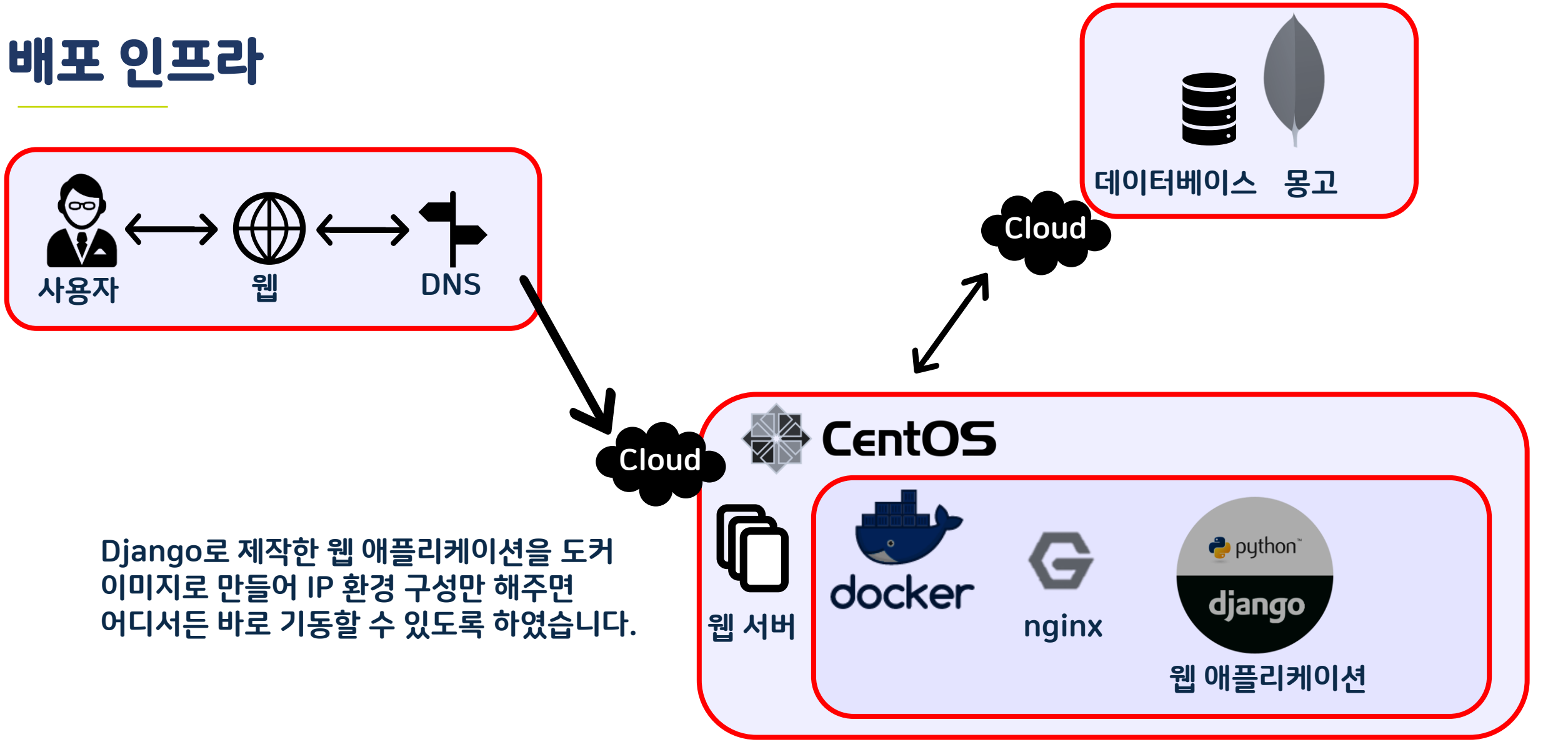


- open API 를 이용해 코로나의 전국 데이터, 크롤링으로 상세 데이터를 직접 수집합니다.
- 데이터를 바탕으로 시각화하여 웹페이지에 그려줍니다.
- 계산된 상세 데이터를 API 형식으로 제공합니다.

# 개발 환경

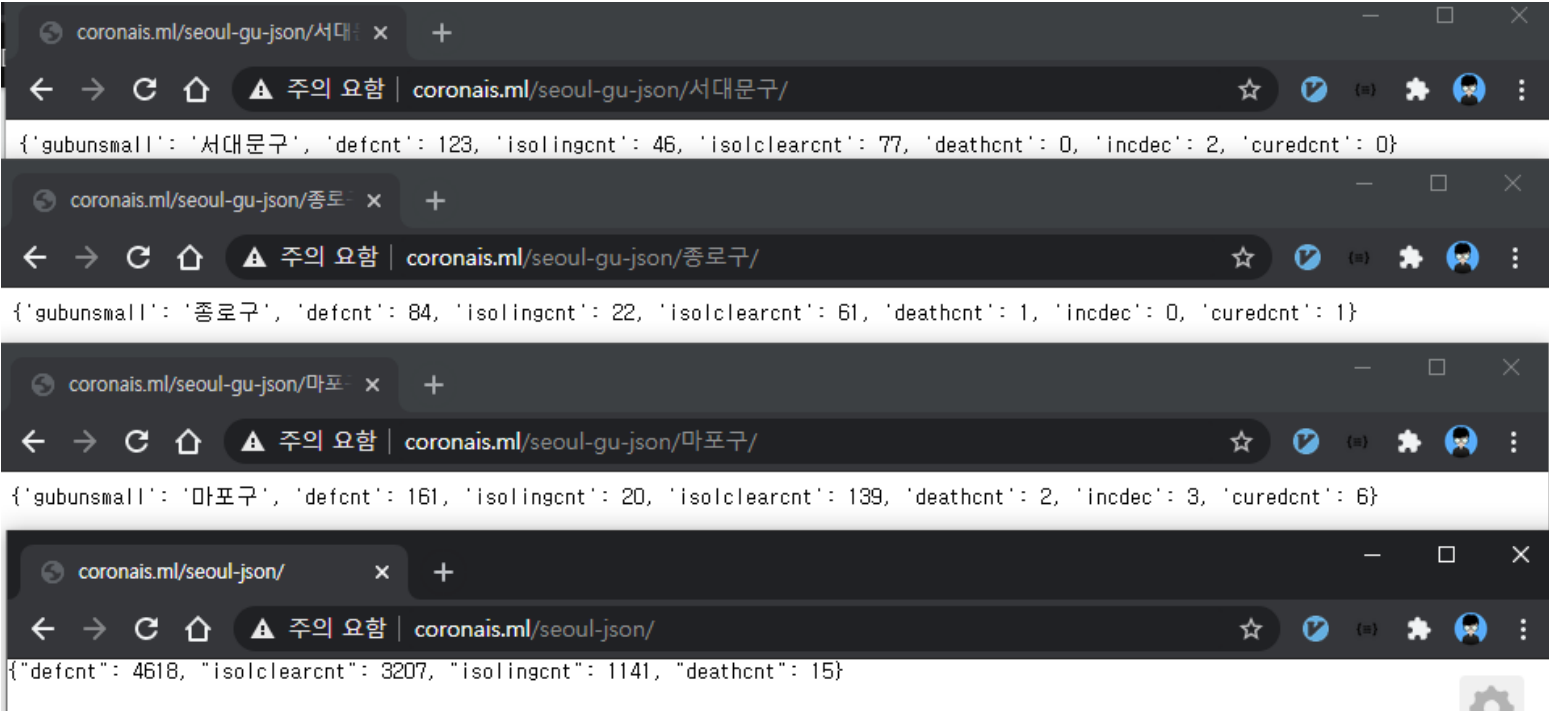
프로젝트 기간	2020. 08. 10 ~ 2020. 09. 18 (40일간)
개발 인원	3명
개발플랫폼	Windows 10
사용언어	Python 3.7.4
사용기술	Django, MongoDB, BeautifulSoup
배포	NCP, docker, nginx, uWSGI
Repository	<a href="https://github.com/resourceSaga/CORONAIIS_2nd">github.com/resourceSaga/CORONAIIS_2nd</a>
Docker Hub	<a href="https://hub.docker.com/repository/docker/resourceSaga/coronais">hub.docker.com/repository/docker/resourceSaga/coronais</a>
Video	<a href="https://youtu.be/uyBtEZLv-M">https://youtu.be/uyBtEZLv-M</a>
URL	coronais.ml (2020년09월까지 운영확정)

# 배포 인프라



Django로 제작한 웹 애플리케이션을 도커 이미지로 만들어 IP 환경 구성만 해주면 어디서든 바로 기동할 수 있도록 하였습니다.

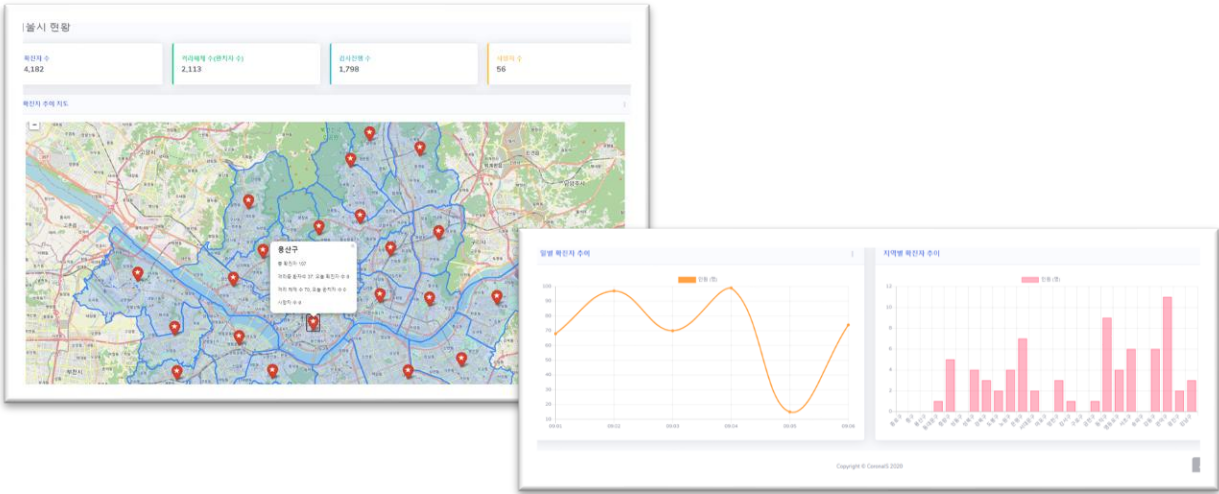
# 서울 상세 데이터 API



## 담당 업무 설명

- 내용 : 앞서 수집한 서울 데이터를 가공하여 REST API 방식으로 데이터를 제공합니다.  
  
/seoul-json/ : 서울 전체의 확진자, 완치자, 검사자, 사망자  
/seoul-gu-json/<구이름> : 구이름 별 데이터를 제공
- 기술 : Django, MongoDB, json, HttpResponse

# 지역별 상세 현황



## 담당 업무 설명

- 내용 : 각 지자체의 홈페이지에서 제공하는 코로나 관련 데이터 (누적 확진자, 누적 사망자, 누적 완치자 등) 를 크롤링하여 수집, 계산으로 가공한 후 일별 확진자 추이, 지역별 감염 추이 데이터를 그래프를 통해 시각화하여 제공합니다.
- 기술 : Django, MongoDB, BeautifulSoup, chart.js

# 후기

## CORONAIIS Django, MongoDB

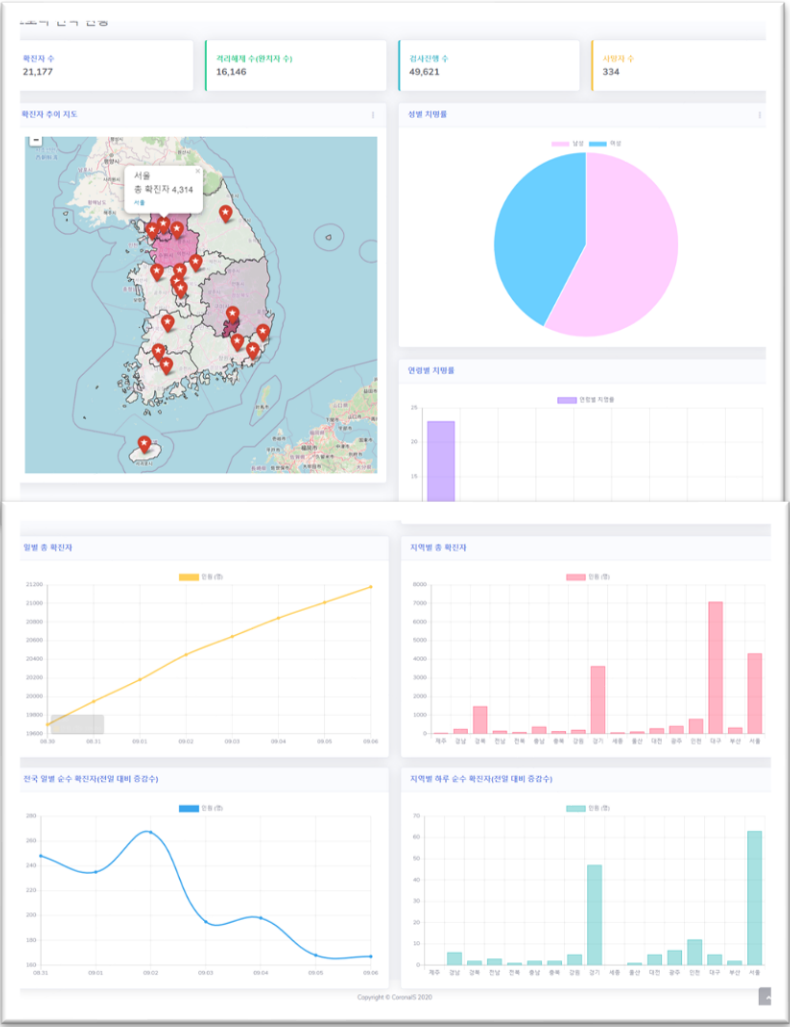
MariaDB를 사용하려고 했으나 데이터 간 의미 있는 관계가 없다고 판단하여 데이터 형식에 유연함을 가진 MongoDB를 사용하였습니다.

Django 와 MongoDB 모두 처음 사용해보는 것이었지만 공부 내용을 바로 적용해 볼 수 있어 재미있게 진행할 수 있었습니다.

팀 프로젝트는 끝이 났지만, 서비스 기능의 범위가 아직 부족하다 생각되어 아쉬움이 많이 남은 프로젝트입니다. 때문에, 현재 가공 데이터를 API로 제공해 줄 수 있는 API를 이용해 만들어 상황을 간단히 알려줄 수 있는 챗봇을 기획하고 있으며, 가능하다면 서비스의 데이터 처리 범위를 전국으로 확대하고 싶어 2차 프로젝트를 준비 중 입니다.

도커를 이용하여 어디서든 이미지를 내려받으면 바로 실행 가능한 환경을 처음 구축해보았는데 컨테이너 이용의 파워풀함과 확장력이 가진 잠재력을 느낄 수 있었습니다. CentOS 환경에서 nginx 와 django를 연동함에 어려움이 많았지만 공부하며 nginx의 구성을 공부해 볼 수 있어 흥미로웠습니다.

담당 작업 외



코로나 전국 현황  
openAPI 를 통해 데이터  
수집, 시각화 하여 보여준다.



코로나 뉴스 기사  
'코로나' 키워드가 들어간  
뉴스를 크롤링으로 수집,  
최신 순서로 보여준다.



# 원격 사료 배급기

| 주제 및 기획의도

| 개발환경

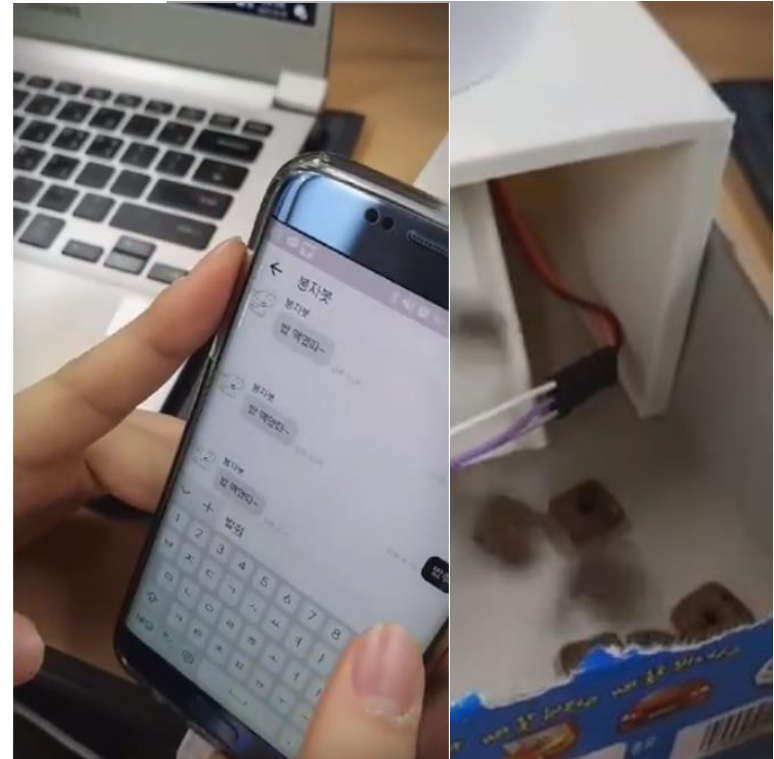
| 인프라

| 주요기능 및 담당업무

| 후기

## Bongja-Bob

(3인 팀 프로젝트)



# 주제 및 기획 의도

## 챗봇을 이용한 원격 사료 배급기

- RaspberryPi 3b+ 와 DC모터를 이용하여 배급기 제작을 기획함
- 외부에서 반려견의 먹이 배급을 걱정하는 친구를 돕기 위한 의도로 만들어짐
- AWS IoT, AWS Lambda, AWS Gateway를 사용하여 서버리스 방식으로 구동

### 기획의도

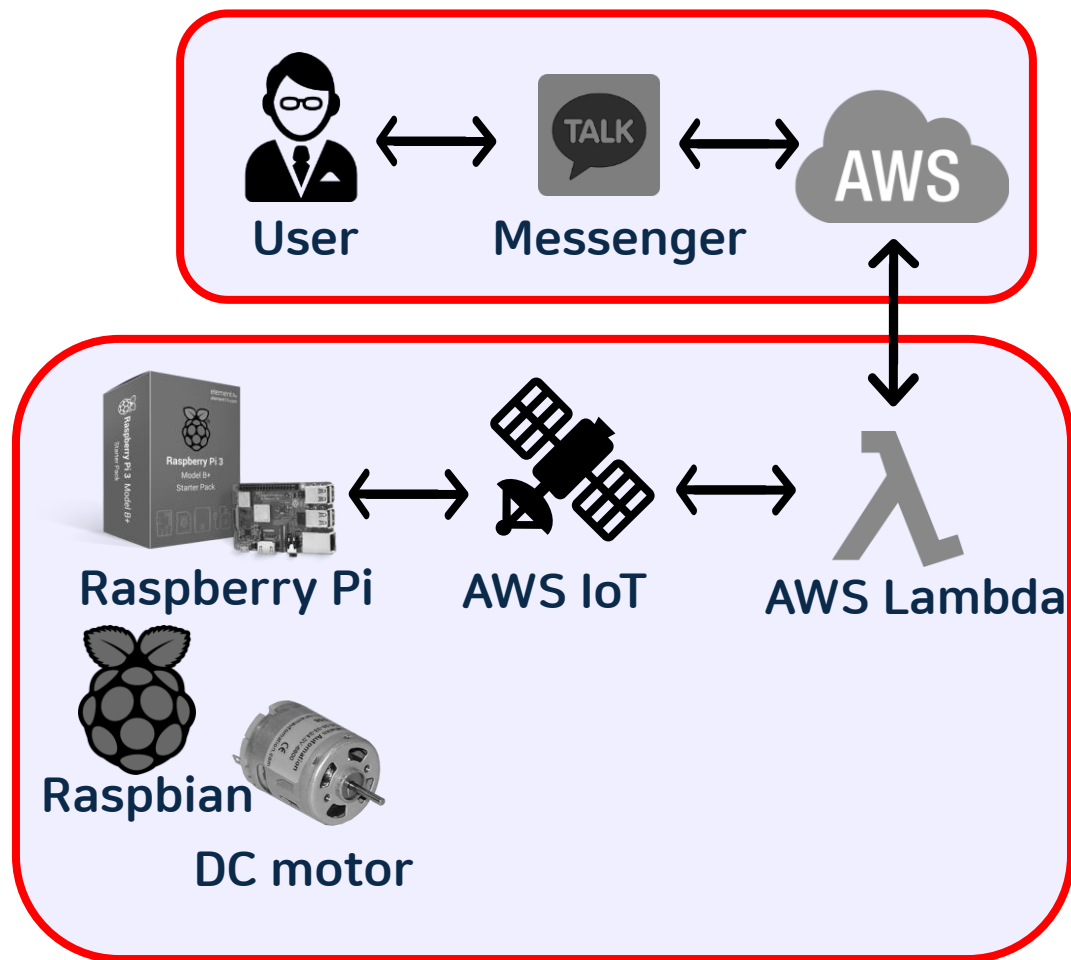


- 사용자가 메신저를 이용해 챗봇에 명령 키워드를 보내면 그에 따라 함수가 작동하는 방식으로 구동됩니다.
- 라즈베리파이 모듈은 모터 작동 제어에 이용되며, 함수의 실행 예약 또한 가능합니다.
- 실제 사용을 위한 모델은 3D 프린터를 이용해 프린트하였습니다.

# 개발 환경

프로젝트 기간	2017. 10. 16 ~ 2017. 12. 29 (75일간)
개발 인원	3명
개발플랫폼	Raspbian ( debian )
개발 툴	Vim
사용언어	Python, C++
사용기술	MQTT, Javascript, JSON
배포환경	AWS Lambda
Repository	github.com/resourceSaga/RaspberryPI_RC
Video	youtube.com/watch?v=f9k5ICDXXuE

## 인프라

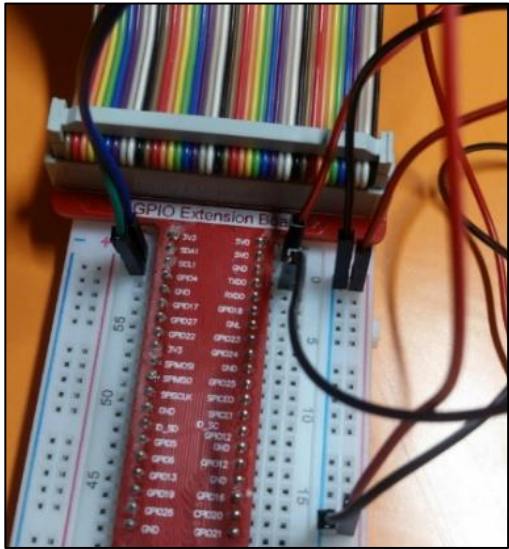
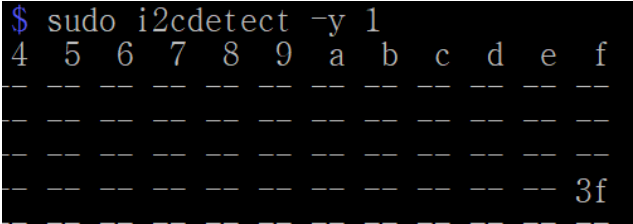


## MQTT protocol Listening

```
189 # Connect to AWS IoT
190 myAWSIoTShadowClient.connect()
191
192 # Create a deviceShadow with persistent subscription
193 deviceShadowHandler = myAWSIoTShadowClient.createShadowHandlerWithName(thingName, True)
194 callbackContainer_Bot = callbackContainer(deviceShadowHandler)
195 #print(callbackContainer_Bot.shadowDict['state']['desired'].get('feedtime'))
196
197 # Listen on deltas
198 deviceShadowHandler.shadowRegisterDeltaCallback(callbackContainer_Bot.customShadowCallback_Delta)
199
200 myAWSIoTMQTTClient = myAWSIoTShadowClient.getMQTTConnection()
201 myAWSIoTMQTTClient.configureOfflinePublishQueueing(-1) # Infinite offline Publish queueing
202 myAWSIoTMQTTClient.configureDrainingFrequency(2) # Draining: 2 Hz
203 myAWSIoTMQTTClient.connect()
204 myAWSIoTMQTTClient.subscribe(topic, 1, callbackContainer_Bot.custom_callback)
```

### 담당 업무 설명

- 내용 : 라즈베리파이는 MQTT 프로토콜을 이용해 항상 메시지 수신 대기 상태입니다. AWS IoT는 인터넷에 연결된 모듈의 AWS Lambda와 상호작용을 중계하며, 메시지가 도착한 경우 알맞은 함수를 실행시킵니다. 이 프로젝트에서는 DC모터를 작동하는 프로그램을 실행하였습니다.
- 기술 : AWS IoT, MQTT protocol, Python



- 내용 : i2c 인터페이스를 이용하여 DC모터와 시계를 제어합니다.  
수신된 명령어에 따라 DC Motor를 실시간 제어 또는 프로그램 시간 예약 함수를 작동하여, 시간에 따라 모듈의 모터를 작동합니다.
- 기술 : i2c interface, C++, python

## 후기

# Bongja-Bob

## RaspberryPi, aws IoT

실제 모듈을 제어해본 경험은 처음이었기 때문에 어려웠지만 친구를 돕겠다는  
동기로 시작한 프로젝트이었기 때문에 하나씩 기능이 구현될 때마다 즐거움이 더욱  
컸습니다.

라즈베리파이 통신을 담당하여 MQTT protocol을 이용한 서버 통신과 모터 조작 프로그램 코드를 작성하는 과정에서 MQTT protocol message listening이라는 것을 처음 알게 되었으며 발행-구독의 작동원리를 직접보고 흥미를 느낄 수 있었습니다.

3D모델링 등 다른 기능을 위해 다른 전공의 팀원과 함께 진행하였는데 전공지식의 차이로 의사소통에 문제가 있었지만, 새로운 대화 방법을 도입한 회의를 통해 해결할 수 있었습니다.

# 학습도움웹

| 주제 및 기획의도

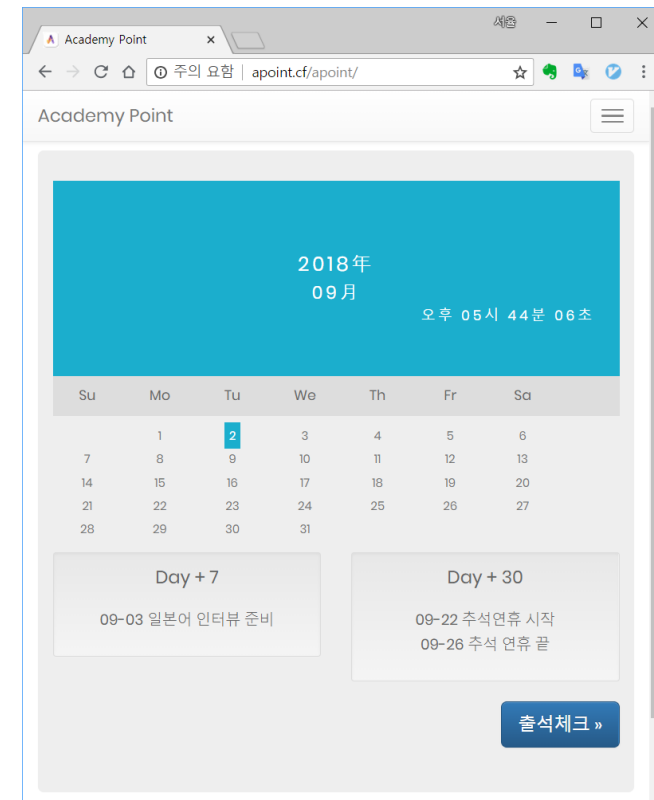
| 개발환경

| 인프라

| 주요기능 및 담당업무

| 후기

## Academy-Point (개인 프로젝트)

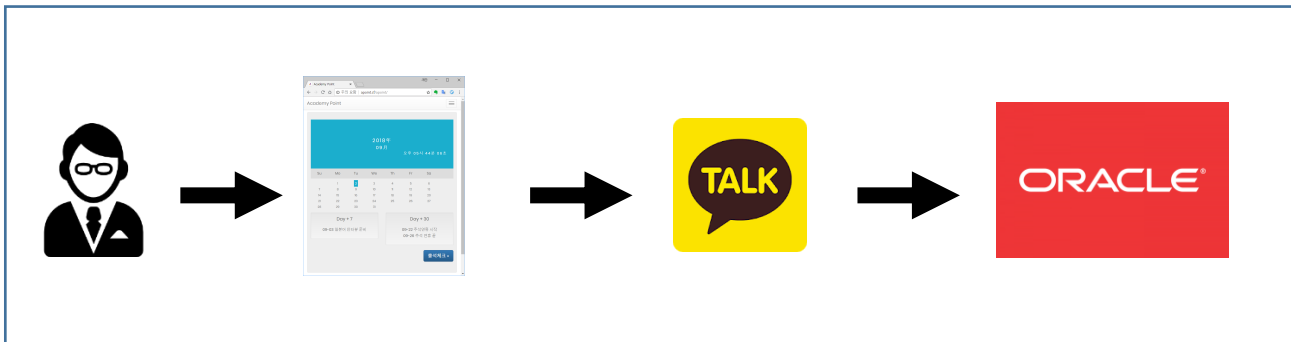


# 주제 및 기획 의도

## 기기등록을 통한 출결 관리 및 공지, 사전, 번역 정보 제공

- Java Spring framework, Mybatis, Oracle Database
- Kakao API Oauth2, Twitter API, Naver Papago API
- 회원들을 위한 학습공지와 출결관리, 한자 쓰기 사전, 번역, 상용한자 랭킹을 제공

## 기획의도

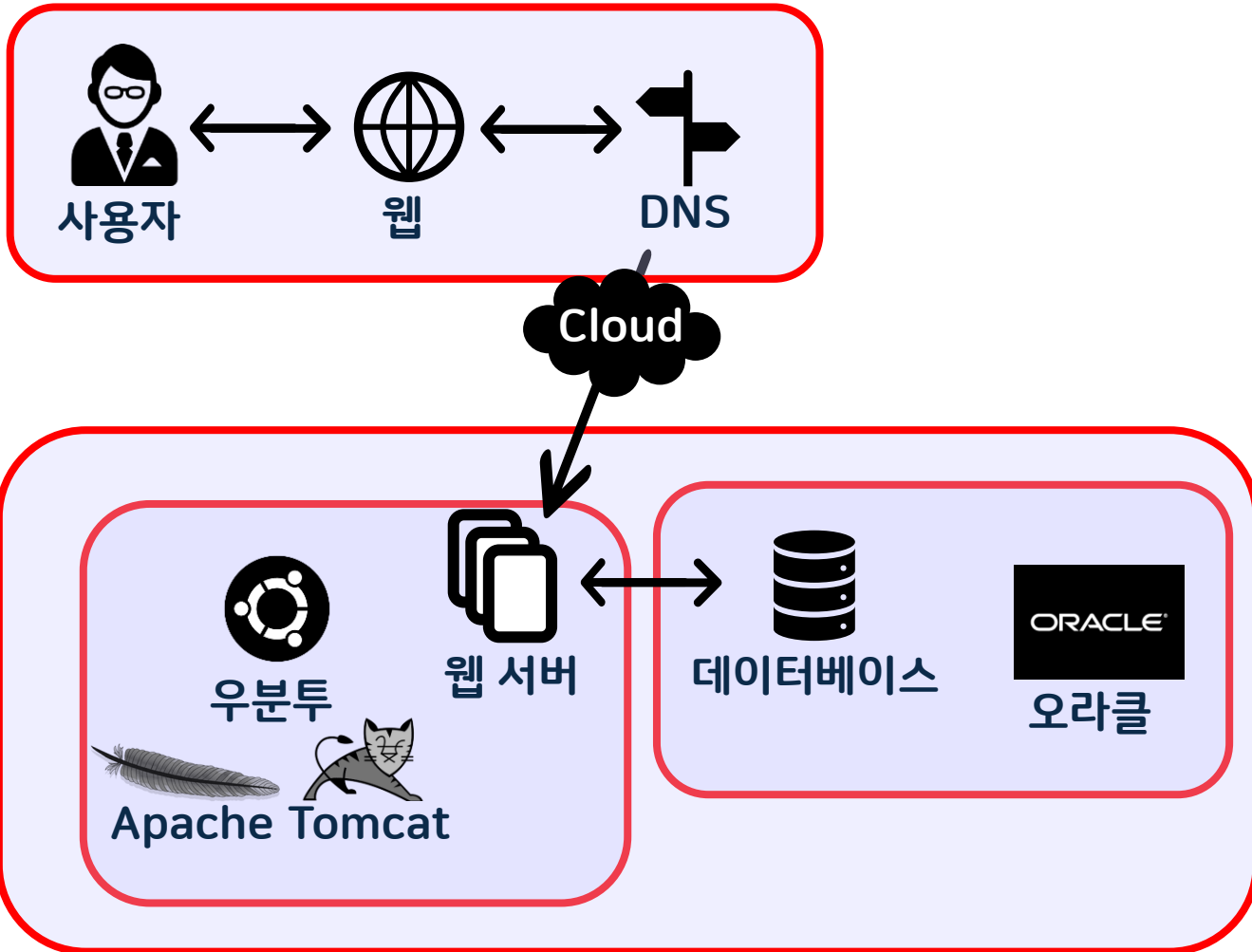


- 회원은 Kakao 로그인을 하며, 기기 등록 과정을 거쳐 출석 서비스를 이용합니다.
- 메인 화면에서 D-day 7 과 D-day 30일의 계획에 대한 공지를 제공합니다.
- 로그인한 회원만이 사전 및 번역 기능을 이용할 수 있으며 검색 결과로 한자의 음/훈 획 쓰기 방법을 제공합니다.
- 지정된 지역의 트위터 이용자가 많이 사용한 한자 랭킹을 제공합니다.

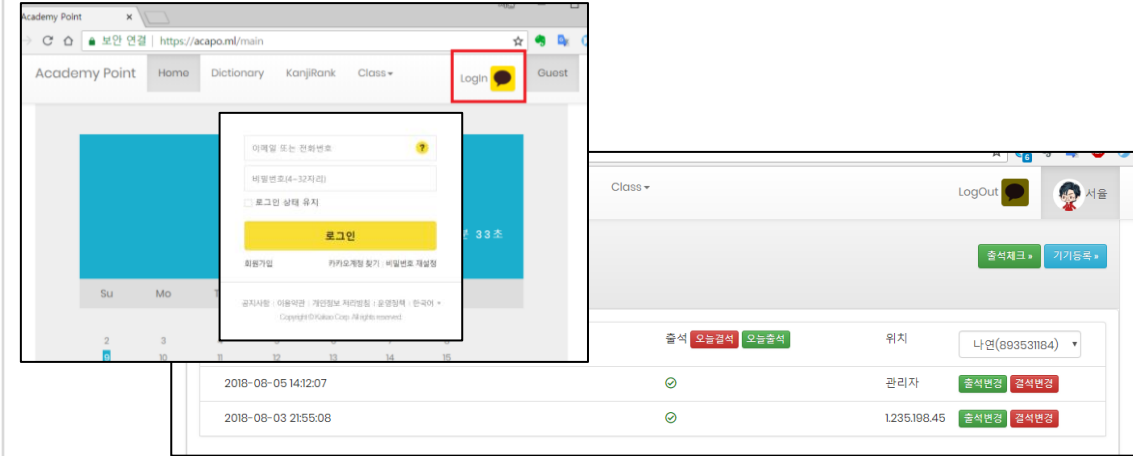
# 개발 환경

프로젝트 기간	2018. 02. 19 ~ 2018. 04. 02 (43일간)
개발 인원	1명
개발플랫폼	Windows 10
개발 툴	Spring Tool Suite, sqldeveloper
사용언어	JAVA(JDK 1.8), SQL, JavaScript, HTML
사용기술	JDBC, Mybatis, JQuery, JSON, JSP, REST API
배포환경	AWS EC2 (Ubuntu)
Repository	github.com/resourceSaga/Academy-Point

# 인프라



# 로그인, 출석체크

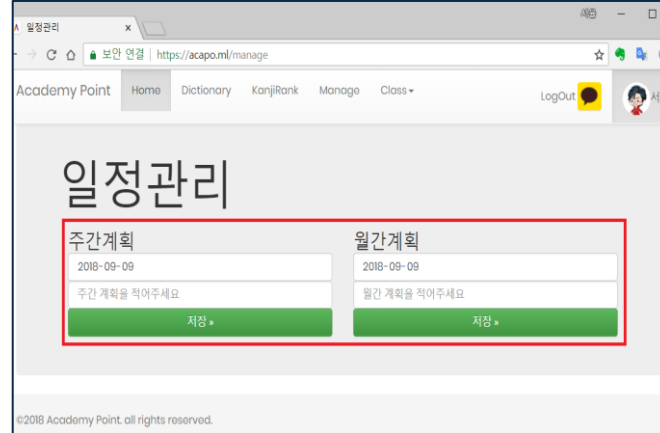
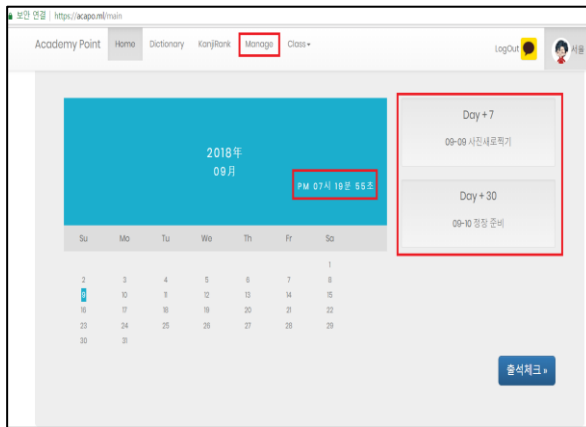


## 담당 업무 설명

- 내용 : 카카오 API를 이용해 계정 로그인을 한 후 사용자 인증을 합니다. 로그인한 사용자의 정보를 파싱 받아 DB에 저장하고, 회원정보 관리에 이용합니다. 사용자는 출석하기 위해 기기를 등록하고, 이후 IP비교를 통해 출석체크를 하게 됩니다. 관리자는 모든 회원의 출석정보를 볼 수 있으며, 내용 변경 또한 가능합니다.

- 기술 : REST API, OAuth 2.0, JSON, net library InetAddress

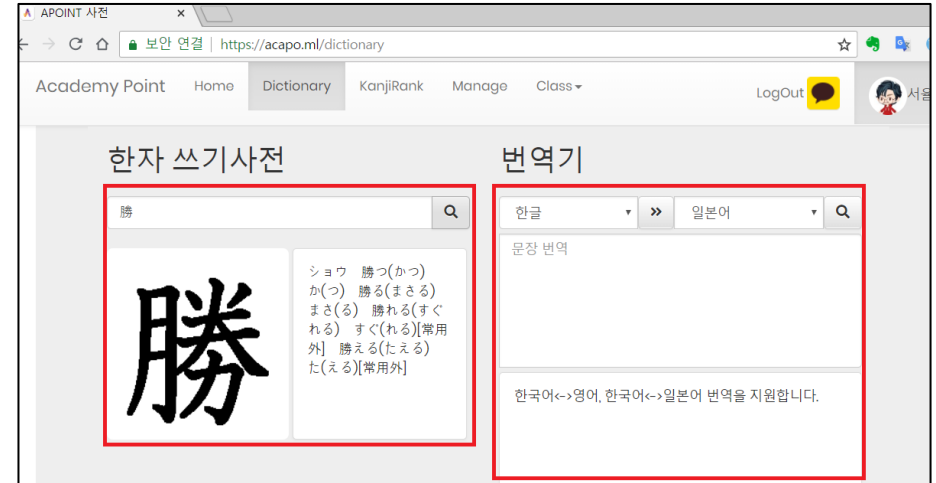
# 스케줄 알림, 캘린더, 실시간 시간



## 담당 업무 설명

- 내용 : 실시간 시간을 제공해주며, 날짜에 따른 일주일, 한달의 일정을 알려줍니다. 관리자에게만 접근이 허용된 Manage 메뉴를 통해 일정을 관리합니다.
- 기술 : Ajax, URI interceptor, Session

# 한자사전, 번역기

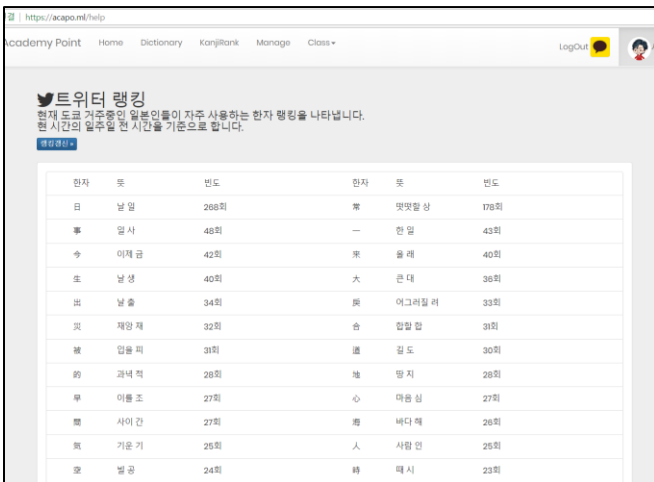


## 담당 업무 설명

- 내용 : 한자 사전과 번역 기능을 이용하기 위해서는 로그인해야만 사용 할 수 있습니다. 한자 검색 시 네이버 사전과 웹 크롤링을 이용해 해당 한자의 데이터를 제공합니다. 이때, 네이버 Papago SMT를 이용해 번역하며, 인코딩 변환으로 한자를 파싱 하였습니다.
- 기술 : URLEncoder, regex, jsoup, JSON, Naver Open API



# 한자 랭킹 시스템



```
KanjiRank [kanji=編, kor=엮을 변, 발을 변, counter=null]
KanjiRank [kanji=書, kor=글 서, counter=null]
KanjiRank [kanji=天, kor=하늘 천, counter=null]
KanjiRank [kanji=久, kor=오랜 구, counter=null]
KanjiRank [kanji=鷹, kor=매 응, counter=null]
KanjiRank [kanji=央, kor=가운데 앙, counter=null]
KanjiRank [kanji=日, kor=날 일, counter=null]
KanjiRank [kanji=常, kor=땀땀할 상, counter=null]
KanjiRank [kanji=出, kor=날 출, counter=null]
KanjiRank [kanji=笑, kor=웃음 소, counter=null]
KanjiRank [kanji=食, kor=밥 식/먹을 식, 먹이 사, 사람 이름 이, counter=null]
KanjiRank [kanji=仙, kor=신선 선, counter=null]
KanjiRank [kanji=台, kor=별 태, 나 이, 대 대, counter=null]
KanjiRank [kanji=日, kor=날 일, counter=null]
KanjiRank [kanji=常, kor=땀땀할 상, counter=null]
검색 된 총 한자 : 2692
```

## 담당 업무 설명

- 내용 : 일본 도쿄에 거주하는 트위터 이용자들의 일상 생활에 대한 주제의 피드 내용을 바탕으로 많이 사용된 한자를 추출해 랭킹으로 제공합니다. 이를 통해 상용한자를 알 수 있습니다.
- 기술 : Twitter Open API, Oauth 2.0

# 후기

## Academy-Point Spring Framework

학원을 다니며 편하게 일정을 관리하고 공부에 필요한 기능이 있는 웹을 만들어 사용하고자 프로젝트를 진행하였습니다.

한자를 파싱하면서 인코딩 문제를 처음만나 문제에 많은 시간을 소비 했던 것이 아쉬웠지만, 혼자 프로젝트를 진행하면서 다양한 공부가 되었습니다.

들어만 보았던 토큰방식의 인증을 직접 사용하였는데 간단하고 database에 비밀번호를 저장할 필요 없다는 좋은 점을 직접 체험해보니 Oauth을 사용하는 이유를 알게 되는 경험이 되었습니다. 서버를 SSL 인증서를 발급해 https를 적용하며 아파치 톰켓의 환경설정을 해보았는데 이를 통해서 서버의 작동원리를 더욱 이해하게 되었습니다.

# 사진 정보제공 웹

| 주제 및 기획의도

| 개발환경

| 인프라

| 주요기능 및 담당업무

| 후기

## ICT-MD (3인 팀 프로젝트)

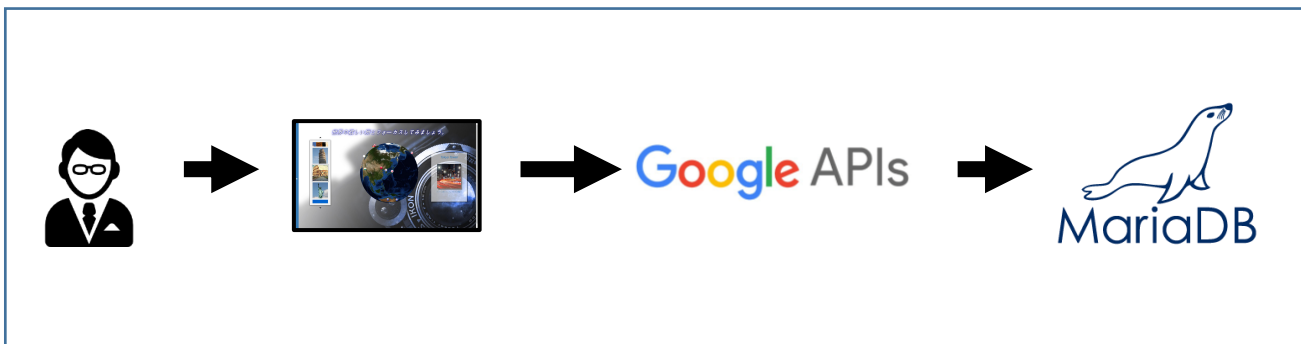


## 주제 및 기획 의도

### 관심있는 사람들을 위한 사진정보 검색 분석 WebApp

- 사용자가 사진을 검색하면 사진을 찍은 카메라 모델, 렌즈 등의 다양한 메타데이터를 제공
- 사용자가 열람한 사진을 분석해, 검색 건수나 브랜드, 모델의 순위의 데이터도 함께 제공

### 기획의도

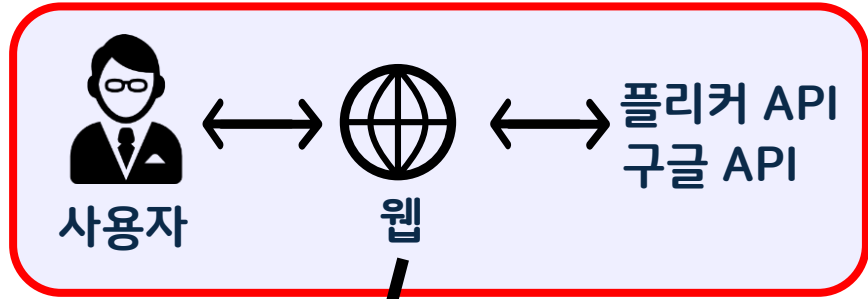


- 나라에 따라 주요 랜드마크의 최신의 사진을 검색해 표시합니다.
- 키워드 검색으로 사진을 제공합니다. 플리커 사진 공유 커뮤니티를 이용하였습니다.
- 구글 비전 API를 이용해 사진을 분석하여, 관련 이미지를 함께 제공합니다.
- 사진에 메타데이터가 있다면 함께 제공합니다.
- 이미지 촬영에 이용된 카메라의 정보와 쇼핑 사이트의 링크를 제공합니다.
- 사진 촬영에 용이한 지역별 온도, 습도, 풍속 등 날씨 정보를 제공합니다.

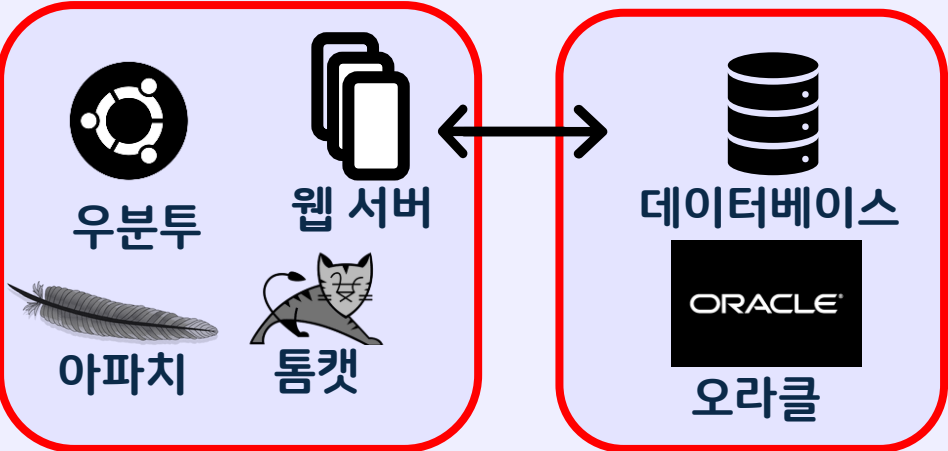
## 개발 환경

프로젝트 기간	2018. 07. 30 ~ 2018. 09. 28 (61일간)
개발 인원	4명
개발플랫폼	Windows 10
개발 툴	Spring Tool Suite, sqldeveloper
사용언어	JAVA(JDK 1.8), SQL, HTML, CSS, JavaScript,
사용기술	JDBC, Mybatis, JQuery, JSON, JSP, REST API
배포환경	AWS EC2 (Ubuntu)
Repository	github.com/resourceSaga/ictmd

# 인프라



AWS



# 사진 검색



## 담당 업무 설명

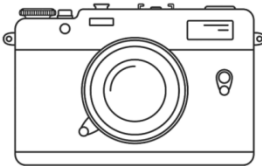

- 내용 : 단일 혹은 복수의 키워드를 이용하여 사진 검색 기능을 제공합니다.  
검색 결과는 사진의 업로드 시간을 기준으로 정렬 할 수 있게 하였습니다.  
검색 내역은 DB에 저장되어 검색어 순위 제공에 이용됩니다.  
키워드를 이용하여 유사한 내용 또한 함께 제공합니다.
- 기술 : Flickr API (이미지 검색), JAVA (Spring framework)

# 이미지 분석

▶ 태그 :

#Chicago	#Bridgeport	#church	#steeple
#power	#lines	#sunset	#sky
#clouds	#birds	#orange	#D7500
#70-300mm	#1/200sec	#f6.3	#ISO125

▶ 메타데이터 :

ブランド	NIKON CORPORATION
モデル	NIKON D7500
フラッシュ	Off. Did not fire
露光	1/200
絞り	f6.3
焦点距離	300.0 mm
レンズ	70.0-300.0 mm f/4.5-6.3

撮影日	2018.09.26 18:32:45
フォーマット	image/jpeg



## 담당 업무 설명

- 내용 : 검색된 이미지를 유저가 클릭하게 되면 자세한 이미지 정보를 제공합니다. 이미지 분석 api를 통해 인식 가능한 관련 키워드를 모두 제공하며, 이미지에 메타데이터(카메라 기종, 촬영 환경의 정보)가 있다면 함께 제공합니다. 분석된 데이터는 DB에 저장되어 이후 키워드 추천에 이용됩니다.
- 기술 : Google Cloud Vision API (이미지 분석, 키워드 추출), JAVA (Spring framework)

# 후기

## ICT-MD

### 사진 정보제공 웹

Smart Cloud IT Master 연수 과정의 마지막 프로젝트로 사진에 관심이 있는 동기생끼리 모여 진행하였습니다.

인스타그램과 같은 한국에서 유명한 대형 포털의 데이터를 API를 통해 이용하려 하였으나, 프로젝트 설계 중 서비스 중단으로 계획 단계에서 자료조사에 예상보다 시간이 걸렸던 프로젝트였습니다.

이미지 데이터의 분석 처리를 이용하다 보니 대용량 데이터로 인해 지연이 있어, 응답 대기시간을 초과하는 등, REST API호출에 문제가 발생했습니다. 다양한 시도 끝에 이미지 리사이징과 스레드를 이용해 해결할 수 있었습니다.

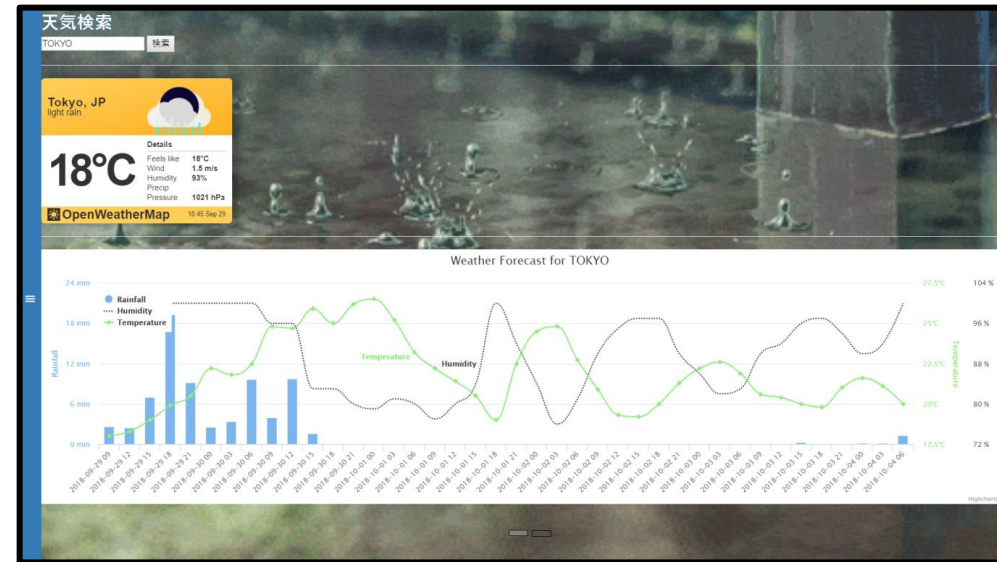
여러 REST API를 동시에 이용하며 다양한 이슈를 경험하며 공부가 되었던 프로젝트였습니다.



# 담당 외



키워드 트렌드 분석, 이미지 정보 분석

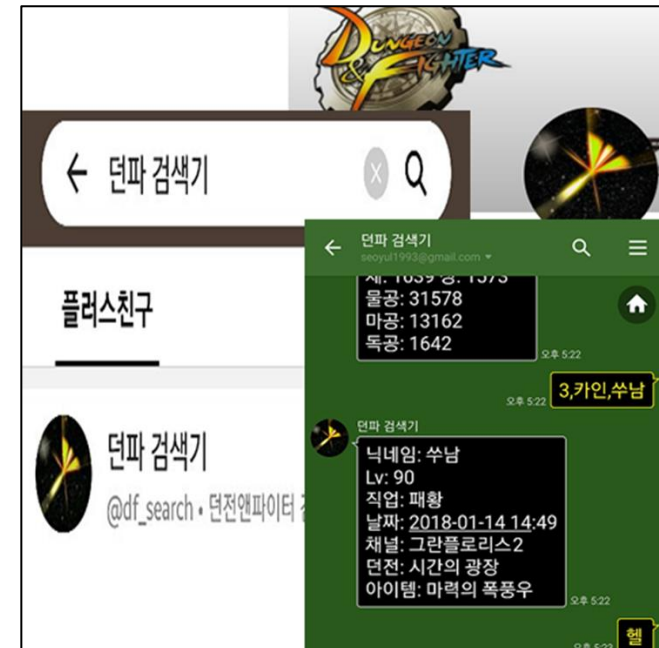


실시간 날씨 정보 제공

# 챗봇

- | 주제 및 기획의도
- | 개발환경
- | 인프라
- | 주요기능 및 담당업무
- | 후기

## 던파 검색기 (개인 프로젝트)



## 주제 및 기획 의도

### 게임 던전애파이터의 오픈 API를 이용한 게임 정보 제공 챗봇

- JavaScript, Node.js
- Kakao API, neople API
- 17년 12월 공개된 네오플API를 이용하여 게임플레이에 도움이 되는 정보를 제공한다.

### 기획의도



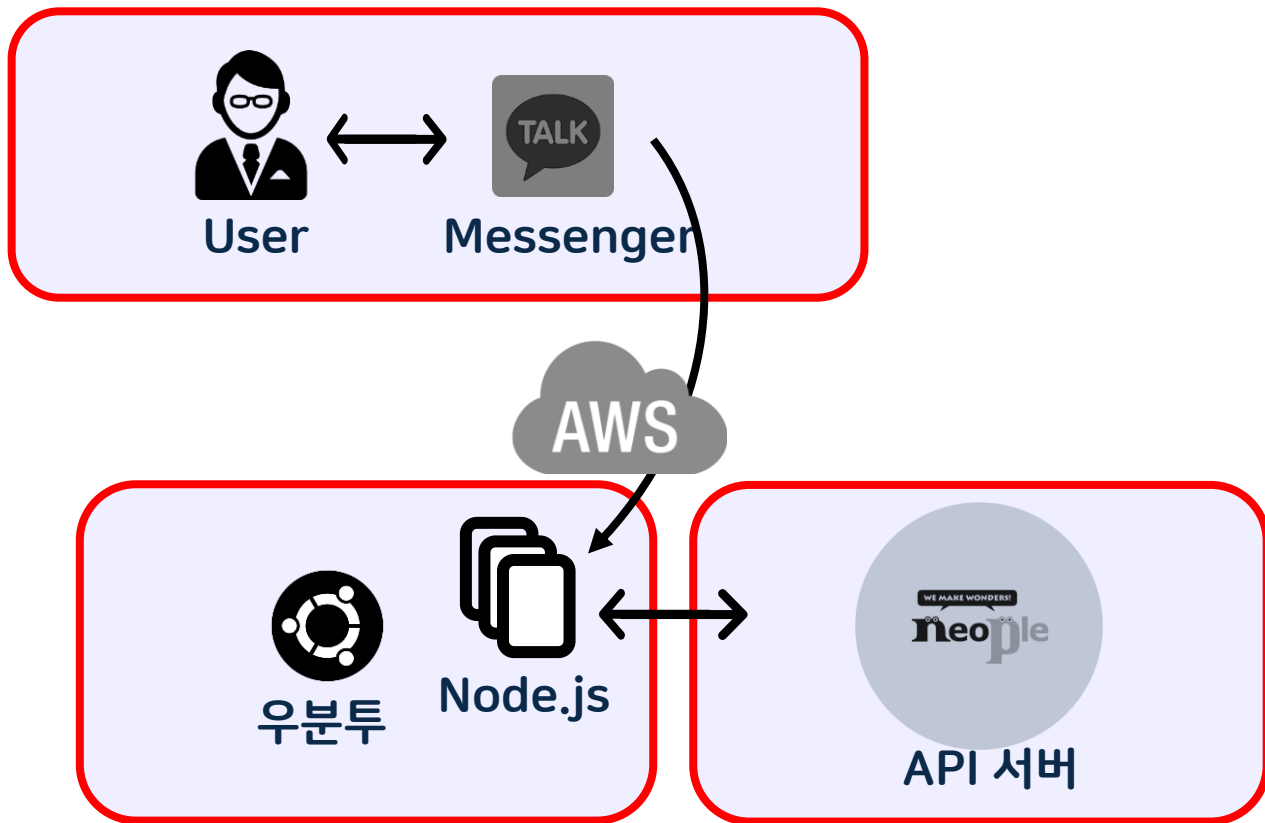
- 이용자는 카카오톡 플러스친구 서비스를 통해 원하는 기능을 키워드로 호출합니다.
- 호출된 키워드에 따라 node 서버에서 네오플 서버로 REST 요청을 합니다.
- 네오플 서버로부터 응답 받은 값을 카카오톡 서버 측으로 재송신하여 유저에게 제공합니다.

## 개발 환경

프로젝트 기간	2018. 01. 03 ~ 2018. 01. 19 (16일간)
개발 인원	1명
개발플랫폼	Windows 10
개발 툴	Visual Studio code
사용언어	JavaScript
사용기술	JSON, Node.js, REST API
배포환경	Ubuntu
Repository	github.com/resourceSaga/DF_SEARCH



## 인프라



## 데이터 송수신

### 담당 업무 설명

- 내용 : 카카오 API를 이용한 메신저를 통해 사용자로부터 키워드를 입력 받습니다. 미리 정해 둔 키워드를 구분하여 Node.js서버에서 게임API로 Rest 요청을 보내고, 요청에 따라 응답을 받아 해당 정보를 다시 유저의 메신저로 송신 하게 됩니다.
- 기술 : REST API, JSON, Node.js

## 후기

즐기던 게임에서 API를 공개하여 지인들과 이용하기 위해 제작하게 되었습니다. 프로젝트를 진행하며 Node.js를 처음 이용하며 패키지 관리의 편리함을 알게 되었고 비동기 송수신에 있어 문제가 있어 많은 공부를 경험할 수 있었습니다.

30명 남짓의 소수의 이용자가 있었지만, 실제로 안면 없는 사람들을 대상으로 서비스를 운영해보고 보람을 느낄 수 있는 프로젝트였습니다.