

▼ API 환율 정보

```
!pip list

# 백업
!pip freeze > ./ref/requirements.txt
# 복구
!pip install -r requirements.txt
```

```
# 진척도 표시
!pip install tqdm
# 트리맵 시각화
!pip install squarify
```

▼ GET과 POST

URL : GET 읽기 전용

POST : 읽기, 쓰기

```
# API

import requests as req
from tqdm import tqdm
import time as t
url = "https://www.koreaexim.go.kr/site/program/financial/exchangeJSON?authkey=gdeZ5f1SkfA0Na"
gdata = req.get(url).json() # json 형태로 변환
for v in gdata:
    print(v)
```

```
# 미국 값만 빼오기
import requests as req # requests 라이브러리를 req로 약칭하여 가져옴
from tqdm import tqdm # tqdm 라이브러리를 tqdm으로 가져옴 (진행 상황 표시용)
import time as t # time 라이브러리를 t로 약칭하여 가져옴

# API 요청 URL 설정 (한국수출입은행 환율 정보)
url = "https://www.koreaexim.go.kr/site/program/financial/exchangeJSON?authkey=gdeZ5f1SkfA0Na"

# URL에서 JSON 데이터 요청 및 응답 저장
gdata = req.get(url).json()

# JSON 데이터의 각 항목을 순회
for v in gdata:
    if '미국' in v['cur_nm']: # 항목 이름에 '미국'이 포함된 경우
        dwon = v['deal_bas_r'] # 해당 항목의 환율 값을 저장
        print(f'미국의 원달러 환율은 {dwon} 원 입니다.') # 환율 값을 출력
```

```
import requests as req
from tqdm import tqdm
import time as t
url = "https://www.koreaexim.go.kr/site/program/financial/exchangeJSON?authkey=gdeZ5f1SkfA0Na"
gdata = req.get(url).json()
for v in tqdm(gdata):
```

```

    if '미국' in v['cur_nm']:
        dwon = v['deal_bas_r']
        print(f'미국의 원달러 환율은 {dwon} 원 입니다.')
    t.sleep(0.5)

```

▼ 비트 코인 환율

```

import requests as req
url = "https://api.bithumb.com/public/ticker/ALL_KRW"
gdata = req.get(url).json()
btc = gdata['data']['BTC']
maxp=float(btc['max_price']) # 최대가
minp=float(btc['min_price']) # 최소가
openp = float(btc['opening_price']) # 시작가
bandp = float(maxp) - float(minp) # 변화폭
print(f'최대가: {maxp}, 최소가: {minp}, 시작가: {openp}, 변화폭: {bandp}')
if (openp + bandp) > maxp:
    print('상승장')
else:
    print('하강장')

```

▼ 미세 먼지 정보

- .gitignore 내부에 .env 등록 해 놓으면 키는 등록되지 않음
- pip 파이썬의 모듈을 관리

```

# pip 업그레이드
!pip install --upgrade pip

# env 설치
!pip install python-dotenv

# 모듈 확인
import requests as req
from dotenv import Load_dotenv

# dotenv 모듈 import
import requests as req
import os
from dotenv import load_dotenv
load_dotenv()
key_ori = os.getenv("SER_KEY")
key = req.utils.unquote(key_ori)
# print(key)
url = "http://apis.data.go.kr/62600000/AirQualityInfoService/getAirQualityInfoClassifiedByStat
params = {'serviceKey':key, 'pageNo':'1', 'numOfRows':20, 'resultType':'json'}
response = req.get(url,params).json()
data = response['response']['body']['items']['item']
inp = input('지역을 입력하세요')
if inp not in [s['site'] for s in data]:
    print('해당 장소가 없습니다.')
else:
    for v in data:
        if(v['site']==inp):
            print(f'=={v['controlnumber']}== \n=={inp} 지역== \n1. 미세먼지 농도 : {v['pm10']} \n

```

env.txt 파일 생성

```
SER_KEY=A%2BtNxs3MfK241UmMX4zVkggR%2FsmBL4ln4SvA%2BYh24qIcz84ul5TfMs4sN0UhRE2l4hqvtL9ktK%2F9S
```

```
/* 자바로 env 모듈 생성하기 */
```

```
// 1.maven 의존성 추가
```

```
<dependencies>
  <dependency>
    <groupId>io.github.cdimascio</groupId>
    <artifactId>java-dotenv</artifactId>
    <version>5.2.2</version>
  </dependency>
</dependencies>
```

```
// 2. .env 파일 생성
```

```
DATABASE_URL=jdbc:mysql://localhost:3306/mydb
DATABASE_USER=root
DATABASE_PASSWORD=secret
```

```
// 3. Java 코드에서 env 사용
```

```
import io.github.cdimascio.dotenv.Dotenv;

public class Main {
    public static void main(String[] args) {
        // .env 파일에서 환경 변수 읽기
        Dotenv dotenv = Dotenv.load();

        String dbUrl = dotenv.get("DATABASE_URL");
        String dbUser = dotenv.get("DATABASE_USER");
        String dbPassword = dotenv.get("DATABASE_PASSWORD");

        System.out.println("Database URL: " + dbUrl);
        System.out.println("Database User: " + dbUser);
        System.out.println("Database Password: " + dbPassword);
    }
}
```

```
# dotenv 모듈 import
```

```
import requests as req # requests 라이브러리를 req로 약칭하여 가져옴
```

```
import os # 운영 체제와 상호 작용을 위한 os 모듈 가져옴
```

```
from dotenv import load_dotenv # .env 파일에서 환경 변수를 로드하기 위한 load_dotenv 함수 가져옴
```

```
load_dotenv() # .env 파일에서 환경 변수를 로드
```

```
# 'SER_KEY'라는 이름의 환경 변수 값을 가져와서 key_ori에 저장
```

```
key_ori = os.getenv("SER_KEY")
```

```
# URL 인코딩된 'key_ori' 값을 디코딩하여 key에 저장
```

```
key = req.utils.unquote(key_ori)
```

```
# 디코딩된 키 값 출력 (주석 처리됨)
```

```
# print(key)
```

```
# API 요청 URL 설정 (부산시 공기질 정보 서비스)
```

```
url = "http://apis.data.go.kr/62600000/AirQualityInfoService/getAirQualityInfoClassifiedByStat
```

```

# API 요청 파라미터 설정
params = {'serviceKey':key, 'pageNo':'1', 'numOfRows':20, 'resultType':'json'}

# API 요청 및 응답을 JSON 형식으로 파싱하여 저장
response = req.get(url,params).json()

# 응답 데이터에서 필요한 항목 추출
response['response']['body']['items']['item']
data = response['response']['body']['items']['item']
inp = input('지역을 입력하세요')
if inp not in [s['site'] for s in data]:
    print('해당 장소가 없습니다.')
else:
    for v in data:
        if(v['site']==inp):
            print(f'=={v['controlnumber']}== \n=={inp} 지역== \n1. 미세먼지 농도 : {v['pm10']} \n')

```