

01 PJT

파이썬을 활용한 API 데이터 수집

챕터의 포인트

- [도전] 금융 데이터 수집
- 제출

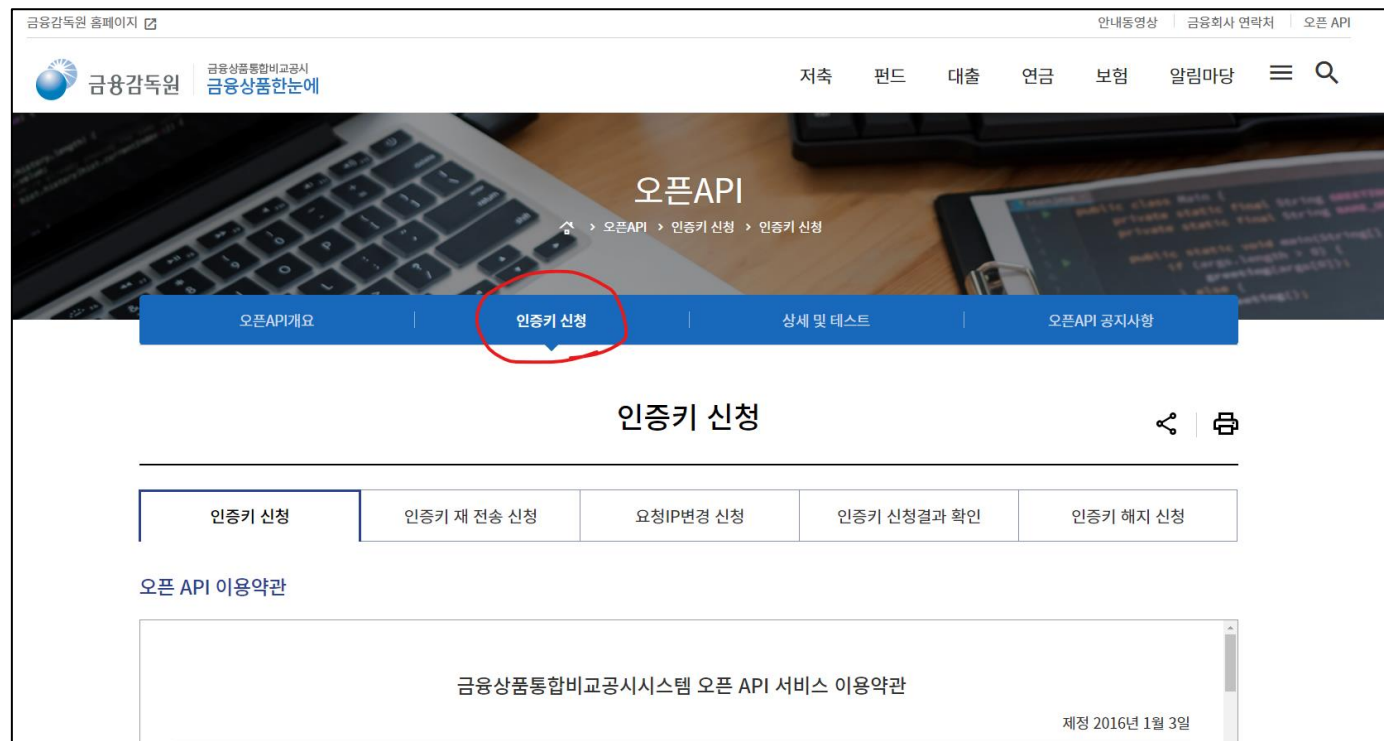
금융 상품 데이터 수집

| 공통 요구사항

- 외부 API 를 사용하여 데이터를 받아오고, 데이터를 원하는 형태로 가공하는 과정입니다.
- 요구사항에서 사용할 API 는 [금융상품통합비교공시 API](#) 입니다.
 - 제공하는 API 중 **정기예금 API** 를 활용합니다.
 - 공식 문서를 보고 **데이터의 구조를 먼저 파악**한 후 요구사항을 구현합니다.

| 공통 요구사항 - API KEY 발급(1/4)

- 사이트 접속 및 인증키 신청 탭 클릭



공통 요구사항 - API KEY 발급(2/4)

- 스크롤을 내려 하단의 이용약관에 동의합니다.

제 2 조 [용어의 정의]

1. 이 약관에서 사용하는 용어의 정의는 다음 각 호와 같습니다.

가. '금융상품통합비교공시시스템 오픈 API(Application Programming Interface)'라 함은 금융감독원이 금융상품통합비교공시시스템의 금융상품정보를 정보 이용자들이 자유롭게 이용할 수 있도록 하기 위하여 구축한 시스템을 말합니다.

나. 'API 서비스'라 함은 금융감독원 금융상품통합비교공시시스템의 금융회사 금융상품정보를 이용자들이 자유롭게 이용할 수 있도록 금융감독원에서 제공하는 서비스를 말합니다.

다. 인증키라 함은 API 이용자가 API 서비스 이용허가를 받은 "일반" 또는 "기관" 이용자임을 식별할 수 있도록 금융감독원이 API 이용자에게 개별적으로 부여하는 고유한 값을 말합니다.

라. 'API 이용자'라 함은 'API 서비스'를 이용하기 위하여 금융감독원의 승인에 따라 인증키를 발급받은 자로서 '개인' 또는 '법인(단체)'으로 구분합니다.

이용약관에 동의하십니까? ☒ 동의 ☐ 동의하지 않음

개인정보 수집 및 이용 동의

금융감독원은 금융상품통합비교공시시스템 오픈API 서비스를 제공하기 위해 필요한 최소한의 개인정보를 수집합니다.

가. 수집목적 : 인증키 발급

나. 수집항목 : 이메일 주소, 본인인증정보

다. 보유기간 : 인증키 등록 해지 시까지 (단, 관계법령의 규정에 의해 보존이 필요한 경우, 관계법령에서 정한 기간 동안 보관)

귀하는 개인정보 수집 동의를 거부할 수 있으며, 동의 거부시 오픈API 서비스 이용이 제한됩니다.

이용약관에 동의하십니까? ☒ 동의 ☐ 동의하지 않음

- 스크롤을 내려 하단의 신청 정보를 작성합니다.

휴대폰 본인인증 진행

☒ 개인 ☐ 단체

개인신청은 신청 즉시 인증키가 발급되고, 일일 조회수는 10,000건으로 제한하며, 원활한 서비스를 위해 기본적으로 1회 조회 시 100건씩(연금저축상품은 10건씩) 조회하여 페이지 처리하고 있습니다.

❗ 관리자의 판단에 따라 개인별로 1회 조회 건수가 변경될 수 있습니다.

• 표시 항목은 필수 입력 항목입니다.

본인 확인	<div style="background-color: #0056b3; color: white; padding: 5px; text-align: center;">본인 인증</div>		
이메일	<input type="text"/> @ <input type="text"/> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">직접입력▼</div> <div style="background-color: #0056b3; color: white; padding: 2px 5px; display: inline-block;">✉ 이메일 인증</div>	<div style="background-color: #0056b3; color: white; padding: 10px; font-size: 24px; margin: 0 auto;">E</div> <p style="font-size: 18px; margin: 5px 0;">이메일 인증 진행</p> <p style="font-size: 18px; margin: 5px 0;">WEB 선택 및 인증</p> <p style="font-size: 18px; margin: 5px 0;">해당 란은 입력</p> <p style="font-size: 18px; margin: 5px 0;">모든 인증 완료</p>	
	이메일 인증 코드 입력 <input type="text"/> <div style="background-color: #0056b3; color: white; padding: 2px 5px; display: inline-block;">입력</div>		
사용용도	<div> <input checked="" type="radio"/> WEB <input type="radio"/> APP <input type="radio"/> WEB+APP </div> <input style="width: 150px;" type="text"/>		
사용URL	<input style="width: 150px;" type="text"/>		
<div style="background-color: #0056b3; color: white; padding: 10px 20px; display: inline-block; margin-top: 10px;">인증키 신청</div>			

이메일 인증 진행

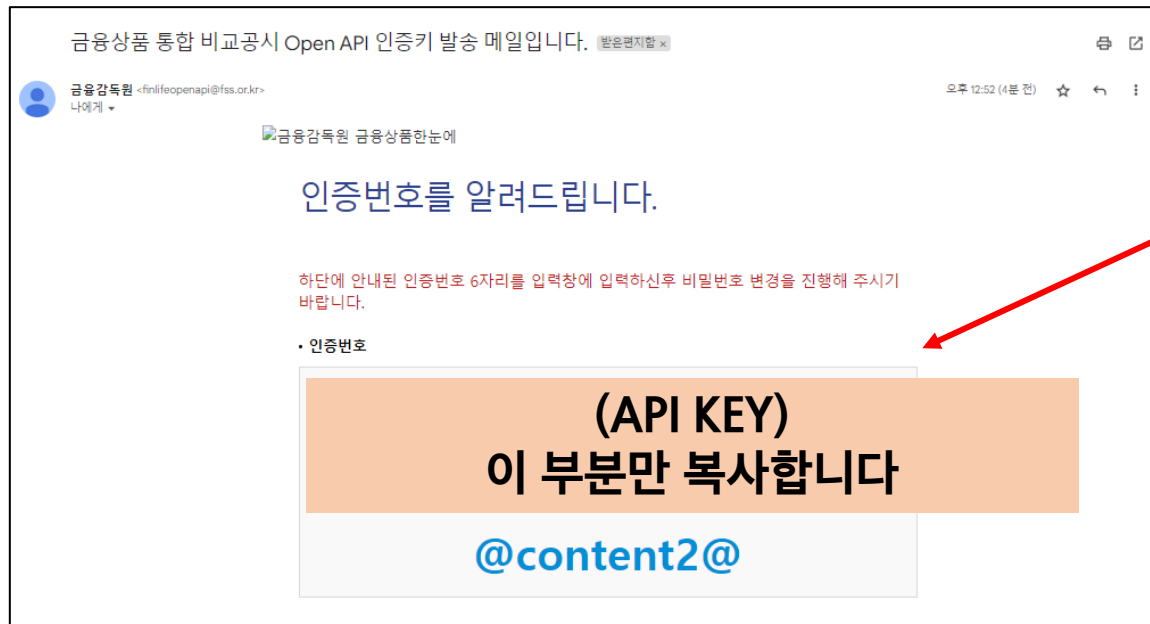
WEB 선택 및 사용용도 작성

해당 란은 입력하지 않습니다

모든 인증 완료 후 인증키 신청 버튼 클릭

| 공통 요구사항 - API KEY 발급(4/4)

- 인증한 메일로 전송된 인증키를 확인한 후 복사하여 활용합니다.



@content2@ 앞의 API KEY 에 해당하는
문자열만 복사하여 KEY 로 활용합니다

공통 요구사항 - API 사용 예제

```
import pprint
import requests

def get_deposit_products():
    api_key = "MY_API_KEY"

    url = 'http://finlife.fss.or.kr/finlifeapi/depositProductsSearch.json'
    params = {
        'auth': api_key,
        # 금융회사 코드 020000(은행), 030200(여신전문), 030300(저축은행), 050000(보험), 060000(금융투자)
        'topFinGrpNo': '020000',
        'pageNo': 1
    }
    # 응답을 json 형태로 변환
    response = requests.get(url, params=params).json()
    return response

if __name__ == '__main__':
    # json 형태의 데이터 반환
    result = get_deposit_products()
    # pprint.pprint(): json 을 보기 좋은 형식으로 출력
    pprint.pprint(result)
```

이메일로 전달받은 API KEY

| 세부 요구사항

- A. 데이터 추출 - Key 값 출력하기
 - B. 데이터 추출 - 전체 정기예금 상품 리스트
 - C. 데이터 가공 - 전체 정기예금 상품들의 옵션 정보 리스트
 - D. 데이터 가공 - 상품과 옵션 정보들을 담고 있는 새로운 값을 만들어 반환하기
-
- [참고] 모든 출력 결과는 예시입니다. 날짜에 따라 다르게 나올 수 있습니다.

A. 데이터 추출 - Key 값 출력하기

- 전체 정기예금의 응답을 json 형태로 변환 후 아래와 같이 Key 값만 출력하도록 구성합니다.

```
dict_keys(['prdt_div', 'total_count', 'max_page_no', 'now_page_no', 'err_cd', 'err_msg', 'baseList', 'optionList'])
```

- 공식 문서의 요청 변수 및 예제 요청결과(JSON) 부분을 참고합니다.
- [힌트] 모든 데이터는 JSON 객체의 “result” 키 값으로 조회할 수 있습니다.

B. 데이터 추출 - 전체 정기예금 상품 리스트

- 응답 중 정기예금 상품 리스트 정보만 출력하도록 구성합니다.
- 출력 결과 예시

```
[{'dcls_end_day': None,
  'dcls_month': '202304',
  'dcls_strt_day': '20230517',
  'etc_note': '- 가입기간: 1~36개월\n'
              '- 최소가입금액: 1만원 이상\n'
              '- 만기일을 일,월 단위로 자유롭게 선택 가능\n'
              '- 만기해지 시 신규일 당시 영업점과 인터넷 홈페이지에 고시된 계약기간별 금리 적용',
  'fin_co_no': '0010001',
  'fin_co_subm_day': '202305171024',
  'fin_prdt_cd': 'WR0001B',
  'fin_prdt_nm': 'WON플러스예금',
  'join_deny': '1',
  'join_member': '실명의 개인',
  'join_way': '인터넷,스마트폰,전화(텔레뱅킹)',
  'kor_co_nm': '우리은행',
  'max_limit': None,
  'mtrt_int': '만기 후\n'
              '- 1개월이내 : 만기시점약정이율x50%\n'
              '- 1개월초과 6개월이내: 만기시점약정이율x30%\n'
              '- 6개월초과 : 만기시점약정이율x20%\n'
              '\n'
              '※만기시점 약정이율 : 일반정기예금 금리',
  'spcl_cnd': '해당사항 없음'},
 {'dcls_end_day': '99991231',
  'dcls_month': '202304',
  'dcls_strt_day': '20230420',
```

C. 데이터 가공 - 전체 정기예금 옵션 리스트

- 응답 중 정기예금 상품들의 옵션 리스트를 출력하도록 구성합니다.
- 이 때, 원하는 데이터만 추출하여 출력되는 결과를 아래와 같이 변경하여 반환하는 함수를 작성하시오.
- 출력 결과 예시

```
[{'금융상품코드': 'WR0001B',  
  '저축 금리': 3.39,  
  '저축 기간': '6',  
  '저축금리유형': 'S',  
  '저축금리유형명': '단리',  
  '최고 우대금리': 3.39},  
{'금융상품코드': 'WR0001B',  
  '저축 금리': 3.57,  
  '저축 기간': '12',  
  '저축금리유형': 'S',  
  '저축금리유형명': '단리',  
  '최고 우대금리': 3.57},  
{'금융상품코드': 'WR0001B',  
  '저축 금리': 3.36,
```

- [참고] Python Dictionary 원하는 키 값으로 데이터 추가하기

```
new_dict = {}  
new_dict['추가'] = "test"  
print(new_dict)
```

- 출력 결과

```
{'추가': 'test'}
```

| D. 데이터 가공 - 새로운 값을 만들어 반환하기(1/2)

- 상품과 옵션 정보들을 담고 있는 새로운 값을 만들어 딕셔너리 형태로 반환하도록 구성합니다.
- 다음과 같은 값만 추출하여 새로운 값에 포함합니다.
 - 금융 상품: '금융회사명', '금융상품명', '금리정보'
 - 해당 금융 상품의 금리 정보(옵션): '저축금리유형', '저축금리유형명', '저축 기간', '저축 금리', '최고 우대금리'
 - 하나의 금융 상품에 대해 여러 개의 옵션을 가질 수 있습니다.
- [힌트] 금융 상품 코드가 같은 금융 상품과 옵션을 하나의 딕셔너리로 만듭니다.

D. 데이터 가공 - 새로운 값을 만들어 반환하기(2/2)

- 출력 결과 예시
 - WON플러스예금에 대한 정보
 - 4가지 금리 정보를 포함하고 있습니다.

```
[{'금리정보': [{'저축 금리': 3.39,  
  '저축 기간': '6',  
  '저축금리유형': 'S',  
  '저축금리유형명': '단리',  
  '최고 우대금리': 3.39},  
{'저축 금리': 3.57,  
  '저축 기간': '12',  
  '저축금리유형': 'S',  
  '저축금리유형명': '단리',  
  '최고 우대금리': 3.57},  
{'저축 금리': 3.36,  
  '저축 기간': '24',  
  '저축금리유형': 'S',  
  '저축금리유형명': '단리',  
  '최고 우대금리': 3.36},  
{'저축 금리': 3.33,  
  '저축 기간': '36',  
  '저축금리유형': 'S',  
  '저축금리유형명': '단리',  
  '최고 우대금리': 3.33}],  
'금융상품명': 'WON플러스예금',  
'금융회사명': '우리은행'},
```


제출

| 제출 시 주의사항

- 제출기한은 금일 18시까지입니다. 제출기한을 지켜 주시기 바랍니다.
- 반드시 README.md 파일에 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점 등을 상세히 기록하여 제출합니다.
 - 단순히 완성된 코드만을 나열하지 않습니다.
- 위에 명시된 요구사항은 최소 조건이며, 추가 개발을 자유롭게 진행할 수 있습니다.
- <https://lab.ssafy.com/> 에 프로젝트를 생성하고 제출합니다.
 - 프로젝트 이름은 '프로젝트 번호 + pjt' 로 지정합니다. (ex. 01_pjt)
- 반드시 각 반 담당 교수님을 Maintainer 로 설정해야 합니다.