

07 PJT

금융상품정보 API Server 제작

챕터의 포인트

- [도전] 금융 상품 데이터를 활용한 REST API Server 구축
- 제출

금융 상품 데이터를 활용한 REST API Server 구축

| 공통 요구사항 (1/2)

- Django 프로젝트 이름은 mypjt 로 지정합니다.
- 앱 이름은 finlife 로 지정합니다.
- .gitignore 파일을 추가하여 불필요한 파일 및 폴더는 제출하지 않도록 합니다.
- 명시된 요구사항 이외에는 자유롭게 작성해도 무관합니다.

| 공통 요구사항 (2/2)

- requests 모듈을 사용하여 외부 API(금융상품통합비교공시 API)를 통해 데이터를 가져옵니다.
 - API KEY 발급 및 활용 방법은 뒷부분의 “참고사항” 을 확인해 주세요.
- 다양한 API 중 **정기예금 API** 를 활용합니다.
 - 반드시 테스트나 API 문서를 통해 전체 데이터를 눈으로 확인한 후 프로젝트를 진행합니다.
- 저장된 데이터 중 조건에 맞는 데이터를 가져오도록 구현합니다.
- Postman 또는 Chrome Browser 를 이용하여 결과를 확인합니다.

| Model

- 정의할 모델 클래스 목록
 - A. DepositProducts
 - B. DepositOptions
- 각 모델의 필드는 [금융상품통합비교공시](#) 페이지 아래에 있는 결과변수와 설명을 참고합니다.

A. DepositProducts

- 정의할 모델 클래스 이름은 DepositProducts 이며, 다음과 같은 정보를 저장합니다.

필드명	데이터 유형	역할
fin_prdt_cd	text (unique)	금융 상품 코드
kor_co_nm	text	금융회사명
fin_prdt_nm	text	금융 상품명
etc_note	text	금융 상품 설명
join_deny	integer	가입 제한(1: 제한없음, 2:서민전용, 3:일부제한)
join_member	text	가입대상
join_way	text	가입 방법
spcl_cnd	text	우대조건

| B. DepositOptions

- 정의할 모델 클래스 이름은 DepositOptions 이며, 다음과 같은 정보를 저장합니다.

필드명	데이터 유형	역할
product	integer	외래 키(DepositProducts 클래스 참조)
fin_prdt_cd	text	금융 상품 코드
intr_rate_type_nm	varchar(100)	저축금리 유형명
intr_rate	float	저축금리
intr_rate2	float	최고우대금리
save_trm	integer	저축기간 (단위: 개월)

| URL

- finlife 앱은 다음 URL 요청에 맞는 역할을 가집니다.

URL 패턴	역할
/finlife/save-deposit-products/	정기예금 상품 목록 DB에 저장
/finlife/deposit-products/	전체 정기예금 상품 목록 출력 & 데이터 삽입
/finlife/deposit-product-options/<str:fin_prdt_cd>	특정 상품의 옵션 리스트 출력
/finlife/deposit-products/top_rate/	가입 기간에 상관없이 최고 금리가 가장 높은 금융 상품과 해당 상품의 옵션 리스트 출력

| View

- finlife 앱은 다음 역할을 가지는 view 함수를 가집니다.

View Method	역할	허용 HTTP Method
save_deposit_products	requests 모듈을 활용하여 정기예금 상품 목록 데이터를 가져와 정기예금 상품 목록과 옵션 목록을 DB에 저장	GET
deposit_products	GET: 전체 정기예금 상품 목록 반환 POST: 상품 데이터 저장	GET, POST
deposit_product_options	특정 상품의 옵션 리스트 반환	GET
top_rate	가입 기간에 상관없이 금리가 가장 높은 상품과 해당 상품의 옵션 리스트 출력	GET

| Serializer

- finlife 앱은 데이터를 JSON 형태로 변환해주기 위해 다음과 같은 Serializer 를 가집니다.
- rest_framework 의 Serializer 또는 ModelSerializer 를 활용하여 구현합니다.

serializer	역할
DepositProductsSerializer	DepositProducts 모델 사용 모든 필드를 사용하도록 설정
DepositOptionsSerializer	DepositOptions 모델 사용 외래키로 설정된 product 필드는 읽기전용으로 설정 모든 필드를 사용하도록 설정

| 요구사항

- 다음과 같은 기능을 구현합니다.
 - A. 정기예금 상품 목록 DB 저장
 - B. 전체 정기예금 상품 목록 출력
 - C. 정기예금 상품 추가하기
 - D. 특정 상품의 옵션 리스트 출력
 - E. 금리가 가장 높은 상품의 정보 출력
 - F. 생성형 AI 를 활용한 더미 데이터 생성
- 시기에 따라 응답 예시와 결과가 다르게 저장 및 출력될 수 있습니다.

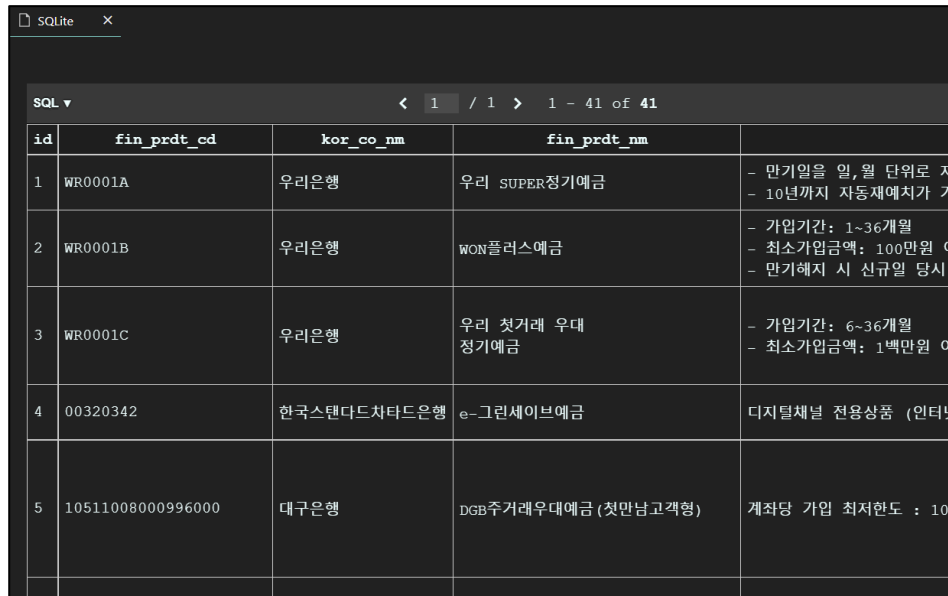
A. 정기예금 상품 목록 및 옵션 목록 저장

- 정기예금 API 로부터 전달받은 데이터 중 상품 목록 정보와 옵션 목록 정보를 DB 에 저장합니다.
 - 테스트 코드 및 POSTMAN 혹은 API 문서를 통해 데이터의 구성을 우선적으로 파악하고 진행합니다.
- [힌트] 옵션 목록 정보는 상품 모델의 기본 키를 외래 키로 참조하고 있습니다
 - ListSerializer 와 같이 한 번에 저장할 수 없습니다.

A. 정기예금 상품 목록 및 옵션 목록 저장

- 아래 URL 로 요청이 오면 정기예금 상품 목록을 받아와 DB에 저장할 수 있도록 코드를 구현합니다.
- 요청 URL: GET finlife/save-deposit-products/
 - 비어있거나 없는 데이터에 대해서는 적절한 처리를 해주어 DB에 저장한다. ex) 금리가 비어있을 땐 -1 로 저장

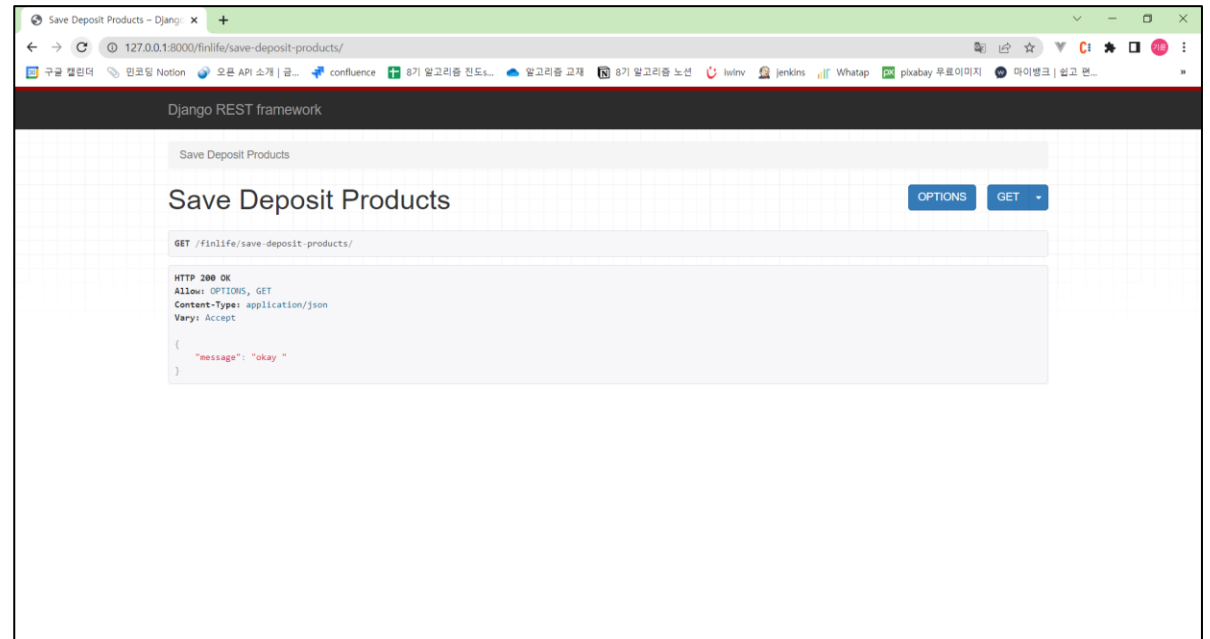
- vscode sqlite extension 으로 저장 확인



The screenshot shows the SQLite extension interface in VS Code. It displays a table with 5 rows of deposit products. The table has columns: id, fin_prdt_cd, kor_co_nm, fin_prdt_nm, and a detailed description. The data includes products from Woori Bank, Hana Bank, and Daegu Bank.

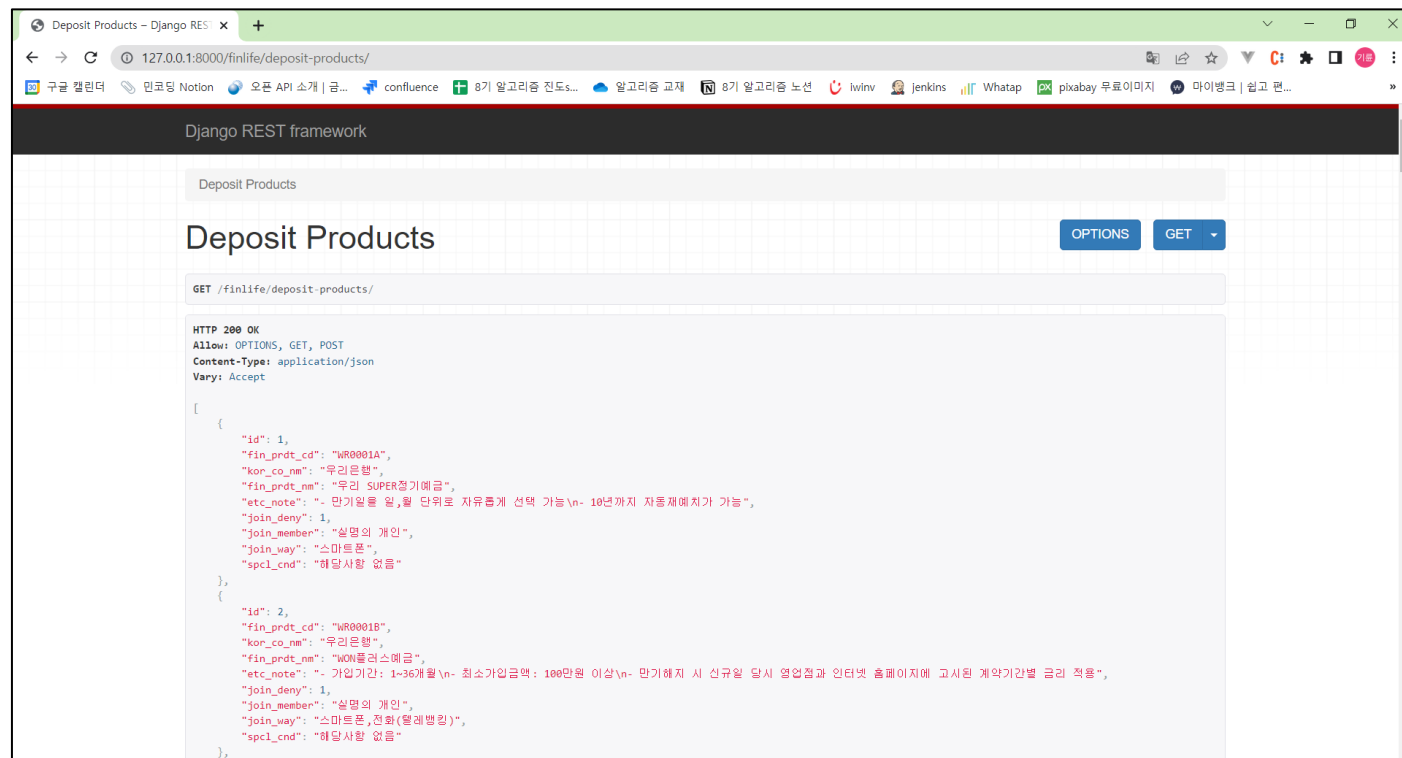
id	fin_prdt_cd	kor_co_nm	fin_prdt_nm	
1	WR0001A	우리은행	우리 SUPER정기예금	- 만기일을 일, 월 단위로 지 - 10년까지 자동재예치가 가
2	WR0001B	우리은행	WON플러스예금	- 가입기간: 1~36개월 - 최소가입금액: 100만원 - 만기해지 시 신규일 당시
3	WR0001C	우리은행	우리 첫거래 우대 정기예금	- 가입기간: 6~36개월 - 최소가입금액: 1백만원
4	00320342	한국스탠다드차타드은행	e-그린세이브예금	디지털채널 전용상품 (인터넷
5	10511008000996000	대구은행	DGB주거레우대예금 (첫만남고객형)	계좌당 가입 최제한도 : 10

- 응답 예시



B. 전체 정기예금 상품 목록 출력

- 아래 URL 로 요청이 오면 DB에 저장된 정기예금 상품 목록을 반환하도록 코드를 구현합니다.
- 요청 URL: GET finlife/deposit-products/



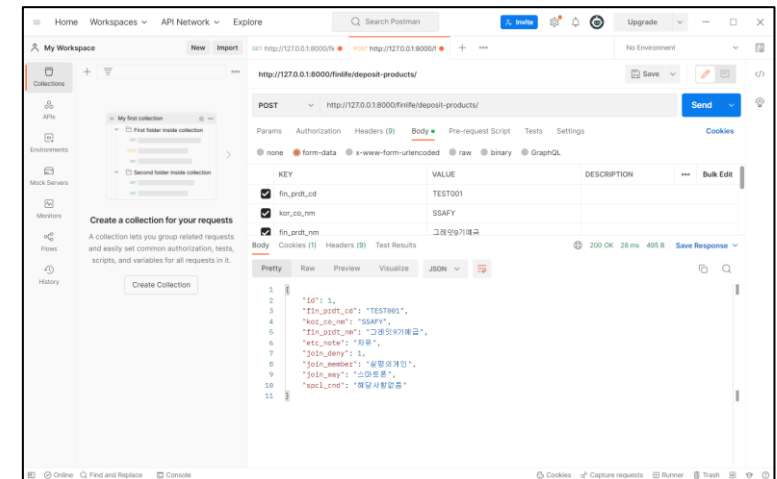
C. 정기예금 상품 추가하기

- 아래 URL 로 요청이 오면 요청과 함께 전송한 데이터를 DB에 저장하도록 코드를 구현합니다.
- 요청 URL: POST finlife/deposit-products/
 - Postman 을 활용합니다.
 - 저장 데이터는 자유롭게 작성합니다.

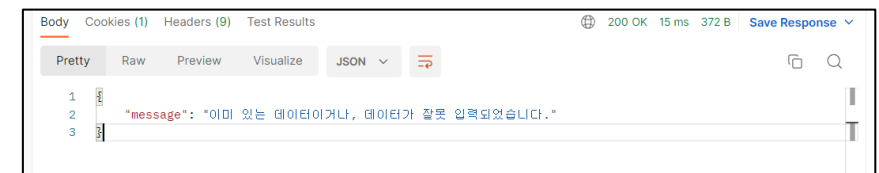
- 데이터 예시

fin_prdt_cd: TEST001
kor_co_nm: SSAFY은행
fin_prdt_nm: 구레잇9기예금
etc_note: 자유
join_deny: 1 (제한없음)
join_member: 실명의 개인
join_way: 스마트폰
spcl_cnd: 해당사항 없음

데이터 삽입 성공 시

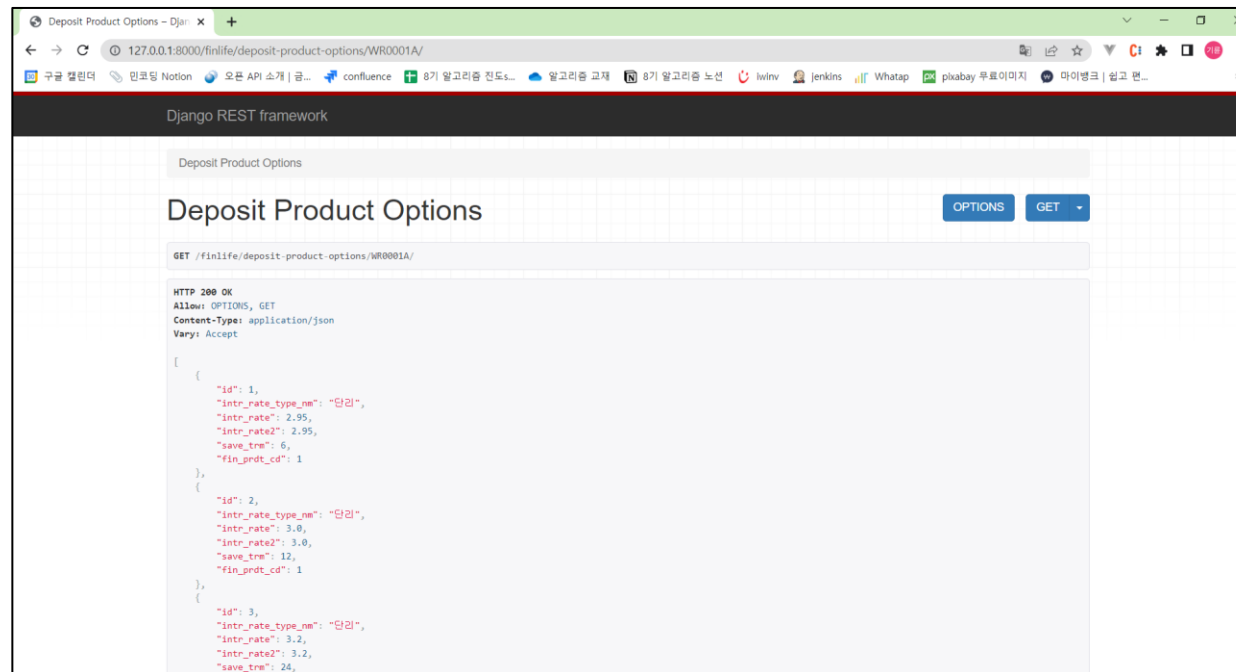


데이터 삽입 실패 시 다음과 같이 적절한 출력 제공



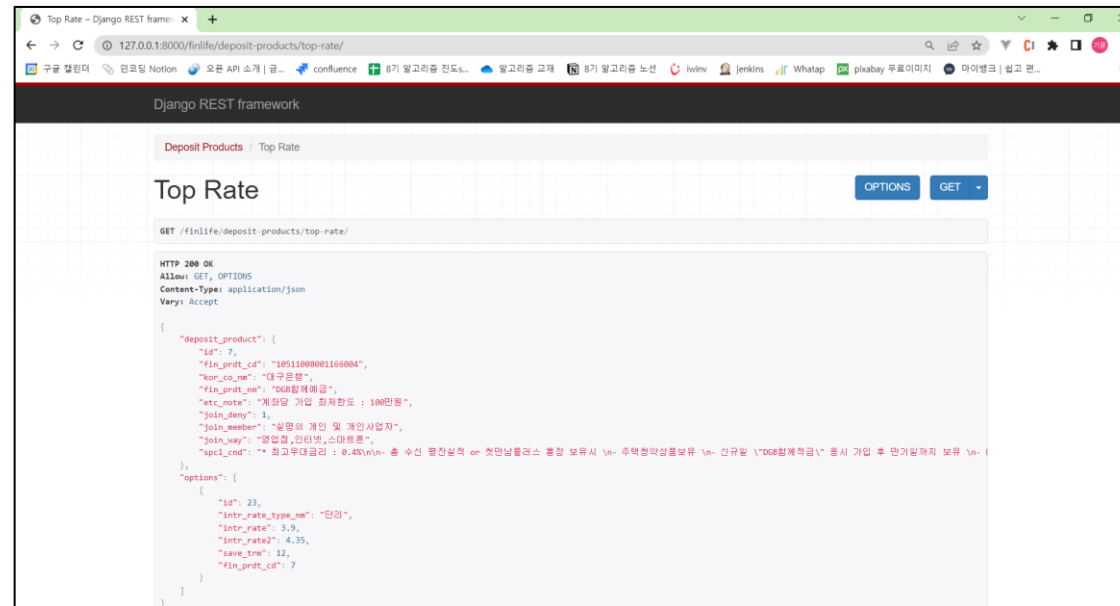
D. 특정 상품의 옵션 리스트 출력

- 아래 URL 로 요청이 오면 상품 코드에 따라 해당 상품의 옵션 리스트를 출력하도록 코드를 구현합니다.
- 요청 URL: GET finlife/deposit-product-options/<str:fin_prdt_cd>/
- B 번에서 확인한 상품 리스트의 가장 첫번째 상품 fin_prdt_cd 를 활용합니다.



E. 금리가 가장 높은 상품의 정보 출력

- 아래 URL 로 요청이 오면 금리가 가장 높은 상품의 상세정보와 옵션을 반환하도록 코드를 구현합니다.
- 저축 기간에 상관 없이, “최고 우대 금리” 가 가장 높은 상품의 정보와 옵션 리스트를 반환합니다.
- 요청 URL: GET finlife/deposit-products/top-rate/
- [힌트] 금리가 가장 높은 옵션을 찾고, 해당 옵션의 외래키로 설정된 금융 상품을 찾습니다.



F. 생성형 AI 를 활용한 더미 데이터 생성

- 생성형 AI 를 활용해 금융 상품 관련 더미 데이터를 생성합니다.
 - 추후 추천 알고리즘에 활용할 데이터 생성을 연습하는 것이 목적입니다.
 - 더미 데이터 생성 방법은 자유롭게 구성합니다.
 - ex1) 생성형 AI 가 더미 데이터를 응답하도록 구성
 - ex2) 더미 데이터를 생성하는 파이썬 실행 코드 등
- 주의 사항
 - 실제로 은행권에서 나올만한 상품인지 등 추천 알고리즘에 활용하기 좋은 데이터인지 검증합니다.
 - 필드나 데이터가 없거나 타입 오류 등이 없는 지 검증해야 합니다.
- 생성한 데이터를 json 형태로 구성하여 함께 제출합니다.
 - 데이터 생성 코드가 있다면 함께 제출합니다.

참고사항

금융 감독원 API KEY 발급

- KEY 발급을 위해 휴대폰 및 이메일 인증이 필요하며, 인증키는 이메일로 받을 수 있다.
- [금융상품통합비교공시 API 사이트](#) 접근 -> 상단 **인증키 신청** 클릭
- 약관을 잘 읽은 후, 하단 신상정보 입력
 - 사용용도: WEB / 사용URL: <http://127.0.0.1:8000/> 입력

금융감독원 홈페이지

금융상품통합비교공시 API 사이트

오픈API 소개

인증키 신청

오픈 API 소개

인증키 신청 방법

오픈 API 이용 방법

오픈 API 활용 예시

오픈 API 소개

인증키 신청 방법

오픈 API 이용 방법

오픈 API 활용 예시

오픈 API 소개

인증키 신청 방법

오픈 API 이용 방법

오픈 API 활용 예시

이용형태

☒ 개인 ☐ 단체

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

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

표시 항목은 필수 입력 항목입니다.

본인 인증

이메일 인증

이메일 인증 코드 입력

이메일 인증코드는 등록하신 이메일을 통해 확인할 수 있습니다.

사용용도

WEB ☒ APP ☐ WEB+APP ☐

사용URL

http://

인증키 신청

| 환경변수 관리

- API KEY 와 같이 외부에 노출하면 안되는 정보를 따로 관리합니다.
- django-environ 설치 - `(venv) $ pip install django-environ`
- .env 파일 작성

〈주의사항〉 API_KEY와 = 사이에 공백이 있으면 오류가 발생합니다.

```
1 API_KEY='<발급받은 API KEY 입력>'
```

- 발급받은 API KEY 를 따옴표로 묶어서 문자열 형태로 입력합니다.
- 수정 후 서버를 다시 시작해야 반영됩니다.

| 환경변수 관리

- settings.py 에 아래 코드를 추가해 줍니다.

```
15 # Build paths inside the project like this: BASE_DIR / 'subdir'.
16 BASE_DIR = Path(__file__).resolve().parent.parent
17
18 √ import os
19 import environ
20
21 # 환경변수를 불러올 수 있는 상태로 설정합니다
22 env = environ.Env(DEBUG=(bool, True))
23
24 # 환경변수를 읽어올 파일을 설정합니다
25 √ env.read_env(
26     env_file=os.path.join(BASE_DIR, '.env')
27 )
28
29 # 환경변수를 읽어옵니다.
30 API_KEY = env('API_KEY')
```


| 환경변수 관리

- settings 에 등록된 환경변수는 views.py 에서 다음과 같이 사용합니다.

```
from django.conf import settings  
  
API_KEY = settings.API_KEY
```

- .gitignore 파일에 .env 를 추가하여 API KEY 가 외부에 노출되지 않도록 설정합니다.

금융상품통합비교공시 API 예시

- API 사용 예시입니다. 아래 코드를 참고하여 요구사항에 맞는 기능을 구현합니다.
- views.py 에서 다음과 같이 사용합니다.

```
BASE_URL = 'http://finlife.fss.or.kr/finlifeapi/'

@api_view(['GET'])
def api_test(request):
    URL = BASE_URL + 'depositProductsSearch.json'
    params = {
        'auth': settings.API_KEY,
        # 금융회사 코드 020000(은행)
        'topFinGrpNo': '020000',
        'pageNo': 1
    }
    response = requests.get(URL, params=params).json()
    return JsonResponse({ 'response': response })
```

- POSTMAN 출력 결과

The screenshot shows a Postman interface with a GET request to `http://127.0.0.1:8000/finlife/api_test/`. The response is a JSON object with the following structure:

```
{
  "response": {
    "result": {
      "prdt_div": "0",
      "total_count": "48",
      "max_page_no": "1",
      "now_page_no": "1",
      "err_cd": "000",
      "err_msg": "정상",
      "baseList": [
        {
          "dcls_month": "202302",
          "fin_co_no": "0010001",
          "fin_prdt_cd": "WR0001A",
          "kor_co_nm": "우리은행",
          "fin_prdt_nm": "우리 SUPER정기예금",
          "join_way": "스마트폰",
          "wrtt_int": "만기 후 (n- 1개월초과 ~ 6개월초과 : 만기시점약정예금×50%\n- 1개월초과 ~ 6개월초과 : 만기시점약정예금×20%\n\n※ 만기시점",
          "spcl_end": "해당사항 없음",
          "join_deny": "1",
          "join_member": "실명의 계인",
          "..."
        }
      ]
    }
  }
}
```

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.