

화해 프로젝트 설명

2020년 1월 7일 화요일 오전 7:24

사전 과제 주제:

AWS S3에서 API로 json형태로 제공되는 상품 목록 테이블과 성분 목록 테이블을 본인의 MySQLDB에 저장하여 2개의 요구사항인 HTTP GET API를 작성하여 제출하라.

사전 과제 조건:

1. DB 모델 형식은 바꾸어도 됨

2. 요구사항에서 지성, 건성, 민감성 타입은 각 상품 목록 테이블 내에 ingredients 컬럼과 성분 목록 테이블을 참고 하여 본인의 솔루션으로 해결해야함

2-1. 지성, 건성, 민감성 타입 계산법은 아래와 같이 주어짐

지성, 건성, 민감성

피부 타입은 다음의 3가지 종류가 있습니다.

- 지성 (oily)
- 건성 (dry)
- 민감성 (sensitive)

화장품을 구성하는 성분들은 피부 타입 별로 가질 수 있는 성질이 3 종류가 있습니다.

- 유익함 (+1)
- 영향없음 (0)
- 유해함 (-1)

예를 들어 성분 A, B, C로 구성된 화장품의 경우 성분 A, B, C가 각 피부타입에 대해 가지는 성질을 다음과 같이 나타낼 수 있습니다.

유익한 성질은 O, 유해한 성질은 X, 영향이 없을 경우 빈 문자열 ("") 로 나타냅니다.

성분명	지성	건성	민감성
A	O	O	X
B	O		X
C	O	O	

화장품을 구성하는 성분이 피부 타입에 유익할 경우 +1, 유해할 경우 -1, 영향이 없을 경우 0점으로 계산하면 아래와 같이 이 화장품이 피부 타입 별로 미치는 영향을 수치로 표현할 수 있습니다.

지성	건성	민감성
+3	+2	-2

3. 상품 목록 테이블 데이터와 성분 목록 데이터는 아래와 같이 주어짐

3-1.상품목록 테이블 데이터 Json으로 제공되는 예시

```
[
  {
    "id": 17,
    "imageId": "4e47e2fa54e84fedbe56b618475adf8c_520",
    "name": "화해 에센스 토너",
    "price": 23000,
    "gender": "all",
    "category": "skincare",
    "ingredients": "Glycerin,Methyl Gluceth-20,Pulsatilla Koreana Extract,Purified Water",
    "monthlySales": 1682
  },
  {
    "id": 19,
    "imageId": "4e47e2fa54e84fedbe56b618475adf8c_521",
    "name": "화해 스킨 솔루션 마스크",
    "price": 4800,
    "gender": "male",
    "category": "maskpack",
    "ingredients": "Glycerin,Sodium Hyaluronate,Xanthan Gum,Niacinamide,Orchid Extract",
    "monthlySales": 463
  },
]
```

- 상품목록
 - 상품 Id
 - 상품 이미지 Id
 - 상품명
 - 가격
 - 성별(남/여/구분없음)
 - 카테고리(스킨케어, 마스크팩, 선케어, 베이스메이크업)
 - 구성 성분 이름(콤마로 구분)
 - 이번 달 판매 수량

3-2.성분 테이블 데이터 Json으로 제공되는 예시

성분 목록

- json다운로드
- 예시

```
[
  {
    "name": "Alcohol",
    "oily": "0",
    "dry": "X",
    "sensitive": "X"
  },
]
```

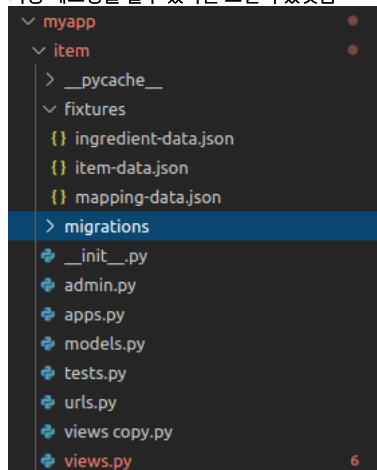
```

{
  "name": "Glycerin",
  "oily": "",
  "dry": "0",
  "sensitive": ""
},
{
  "name": "Stearic Acid",
  "oily": "X",
  "dry": "",
  "sensitive": ""
},

```

- 성분목록
 - 이름
 - 지성영향
 - 건성영향
 - 민감성영향

4. 최종적으로 만드는 DB 데이터들을 fixtures폴더 내에 Json형식으로 데이터를 보관해야 자동 테스트를 할수 있다는 조건이 있었음



문제1

API 구성하기

주어진 데이터를 활용해 API를 구성해야 합니다.

상품 목록 조회하기

피부 타입 별로 상품 목록을 필터링해 조회할 수 있습니다.

기능 상세

- 주어진 피부 타입에 대한 성분 점수를 계산해서 높은 상품 순으로 보여줍니다. 점수가 같다면 낮은 가격의 상품을 먼저 표시합니다.
- 상품 목록을 50개 단위로 페이지 합니다. 인자로 페이지 번호가 주어지면, 해당되는 상품 목록이 보여집니다.
- 상품 카테고리를 선택할 수 있습니다.
- 제외해야 하는 성분들을 지정할 수 있습니다.
 - exclude_ingredient 인자로 전달한 성분들을 모두 가지지 않는 상품들만 목록에 포함합니다.
- 포함해야 하는 성분들을 지정할 수 있습니다.
 - include_ingredient 인자로 전달한 성분들을 모두 가지고 있는 상품들만 목록에 포함합니다.

URL

/products

Method

GET

Request Header 구조 예시

```

GET /products?skin_type=dry
Content-Type: application/json

```

Query Parameter

Parameter	Type	Description
skin_type	String	(필수) 지성(oily)/건성(dry)/민감성(sensitive) 중 택 1
category	String	(선택) 상품 카테고리
page	Integer	(선택) 페이지 번호
exclude_ingredient	String	(선택) 제외해야 하는 성분 목록(콤마로 구분)
include_ingredient	String	(선택) 포함해야 하는 성분 목록(콤마로 구분)

Sample Call

```

/products?skin_type=oily&category=skincare&page=3&include_ingredient=Glycerin

```

product/skin_type/oily/ingredient/glycerin/ingredient/glycerin/ingredient/glycerin

Success Response

```
[
  {
    "id": 17,
    "imgUrl": "https://grepp-programmers-challenges.s3.ap-northeast-2.amazonaws.com/2020-birdv",
    "name": "화해 에센스 토너",
    "price": 23000,
    "ingredients": "Glycerin,Methyl Gluceth-20,Pulsatilla Koreana Extract,Purified Water",
    "monthlySales": 1682
  },
  {
    "id": 23,
    "imgUrl": "https://grepp-programmers-challenges.s3.ap-northeast-2.amazonaws.com/2020-birdv",
    "name": "화해 연젤 토너",
    "price": 4000,
```

```
    "ingredients": "Glycerin,Sodium Hyaluronate,Xanthan Gum,Niacinamide,Orchid Extract",
    "monthlySales": 463
  },
  ...

  {
    "id": 80,
    "imgUrl": "https://grepp-programmers-challenges.s3.ap-northeast-2.amazonaws.com/2020-birdv",
    "name": "화해 화이트 브라이트닝 소프트 인리지드",
    "price": 24000,
    "ingredients": "Glycerin,Alcohol,Purified Water,Vinyl Dimethicone,PE6-10 Dimethicone",
    "monthlySales": 4437
  }
]
```

Response Body 구성

Column	Type	Description
id	Integer	상품 아이디
imgUrl	String	제품의 이미지 URL
name	String	상품명
price	Integer	가격
ingredients	String	성분 명(콤마로 구분)
monthlySales	Integer	이번 달 판매 수량

문제2

상품 상세 정보 조회하기

상품 상세 정보를 조회할 수 있습니다. 같은 카테고리의 상위 3개 추천 상품들도 함께 조회할 수 있습니다.

기능 상세

- 상품 id로 특정 상품의 상세 정보를 조회할 수 있습니다.
- 이미지 id를 base URL과 조합해 상품 이미지를 불러올 수 있는 URL을 보여줍니다.
- 같은 카테고리의 상품 중 상위 3개의 추천 상품 정보를 조회할 수 있습니다.
 - 인자로 받은 피부 타입에 대한 성분 점수가 높은 순서로 추천합니다. 점수가 같다면, 가격이 낮은 상품을 먼저 추천합니다.
 - 추천 상품 정보는 상품 아이디 , 상품 썸네일 이미지 URL , 상품명 , 가격 을 포함합니다.

URL

```
/product/{id}
```

Method

```
GET
```

Request Header 구조 예시

```
GET /product/17?skin_type=oily
Content-Type: application/json
```

Query Parameter

Parameter	Type	Description
skin_type	String	(필수) 지성(oily)/건성(dry)/민감성(sensitive) 중 택 1

Sample Call

```
/product/17?skin_type=oily
```

Success Response

```
[
  {
    "id": 17,
    "imgUrl": "https://grepp-programmers-challenges.s3.ap-northeast-2.amazonaws.com/2020-birdv",
    "name": "화해 에센스 토너",
    "price": 23000,
  }
]
```

Success Response

```
[
  {
    "id": 17,
    "imgUrl": "https://grepp-programmers-challenges.s3.ap-northeast-2.amazonaws.com/2020-birdv",
    "name": "화해 에센스 토너",
    "price": 23000,
  }
]
```

```

    "gender": "all",
    "category": "skincare",
    "ingredients": "Glycerin,Methyl Gluceth-20,Pulsatilla Koreana Extract,Purified Water",
    "monthlySales": 1682
  },
  {
    "id": 23,
    "imgUrl": "https://grepp-programmers-challenges.s3.ap-northeast-2.amazonaws.com/2020-birdv",
    "name": "화해 연젤 토너",
    "price": 33000
  },
  {
    "id": 37,
    "imgUrl": "https://grepp-programmers-challenges.s3.ap-northeast-2.amazonaws.com/2020-birdv",
    "name": "화해 화이트 브라이트닝 소프너 인리치드",
    "price": 24800
  },
  {
    "id": 141,
    "imgUrl": "https://grepp-programmers-challenges.s3.ap-northeast-2.amazonaws.com/2020-birdv",
    "name": "화해 퍼펙트 스킨케어",
    "price": 14800
  }
]

```

Response Body 구성

상품 상세 정보 response body는 해당 id 상품 정보와 3개의 추천 상품 정보를 배열로 구성합니다.

상품 정보

Column	Type	Description
id	Integer	상품 아이디
imgUrl	String	상품 fullsize 이미지 URL
name	String	상품명
price	Integer	가격
gender	String	성별(남/여/구분 없음)
category	String	카테고리
ingredients	String	성분 명(콤마로 구분)
monthlySales	Integer	이번 달 판매 수량

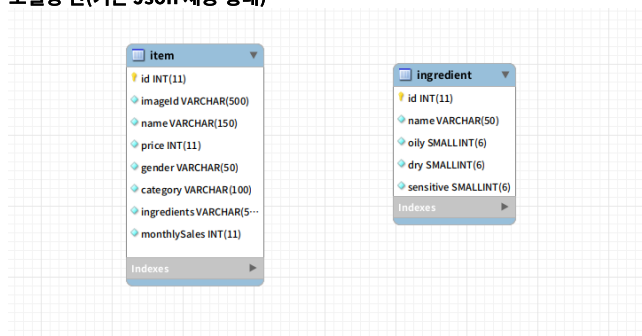
추천 상품 정보

Column	Type	Description
id	Integer	상품 아이디
imgUrl	String	상품 thumbnail 이미지 URL
name	String	상품명
price	Integer	가격

문제 풀이 절차

1.모델링

모델링 전(기존 Json 제공 형태)



모델링 진행 시 겪었던 고민

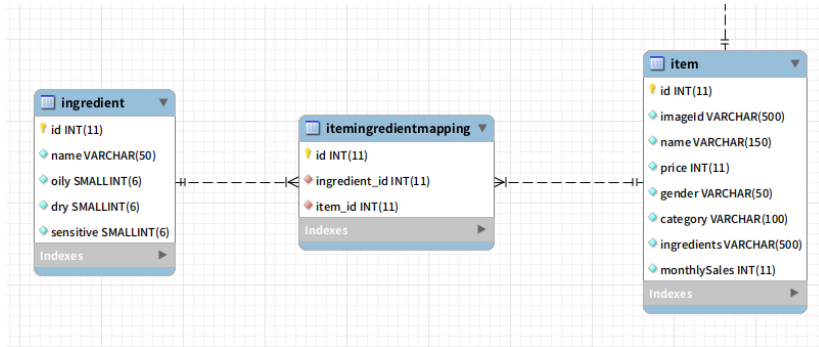
1. 문제에서는 주어지지 않았지만 NOSQL 타입의 데이터를 제공하는 느낌을 받아서

Mysql(RDBMS)의 정규화 진행

-> 상품 목록 테이블에 ingredients 속성은 ";" 로 구분 할 시 하나의 컬럼에 여러 데이터가 있으므로 제1 정규형 위배임으로

-> 상품 목록 테이블과 성분을 M:M관계로 바라보고 가운데 맵핑 테이블을 두기로 함

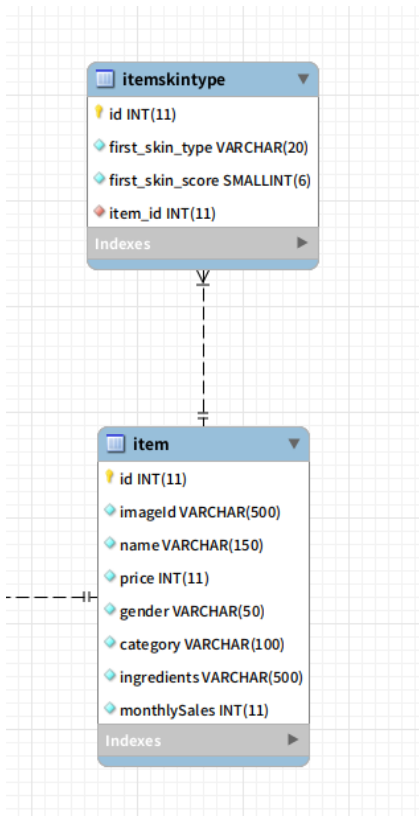
-> RDB에 데이터 넣을 시 맵핑 테이블 데이터 Insert시 API엔드포인트 하나로 해결 하기로 함
(따로 py파일로 처리 가능하나 ORM사용과 심사위원에게 Insert 짜는 로직을 보여주기로 함)



2. 편의성, 확장성, 데이터 유동성 3가지 기준을 두고 상품 스킨 타입 테이블로 분리

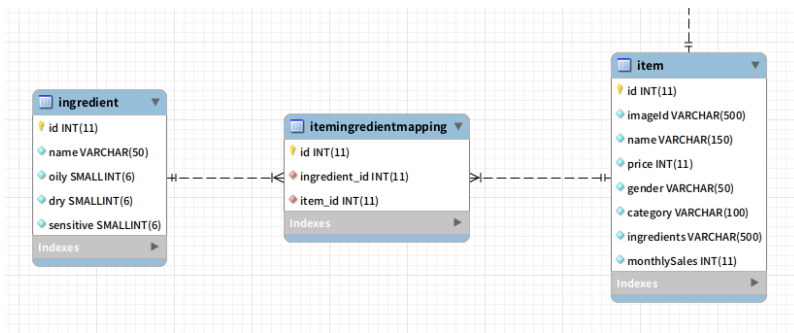
(스킨 타입 인덱싱 편의성, DB 확장성(피부 타입 1순위 2순위 3순위),
데이터 유동성(성분 평가값 자체가 수정이 가능하도록 예) 알코올 성분이 oily 1점으로 했다가 나중에 -1로 수정될수 있으니))

-> 성분 타입 점수가 동일 할 시 가격이 낮은 조건이 1순위로 우선되는 부분도 고려 해주어야 함(API로직으로 해결할 것임)



최종 DB 모델



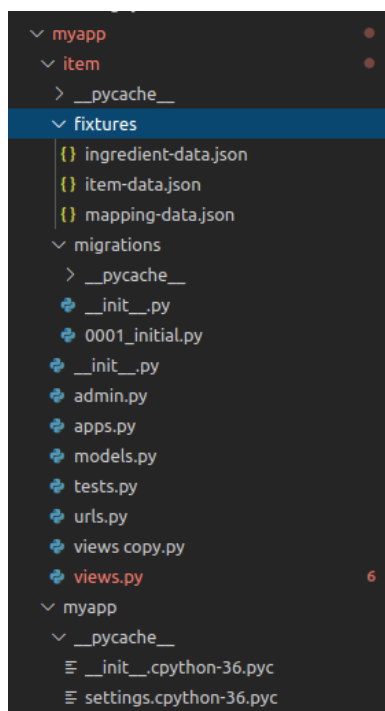


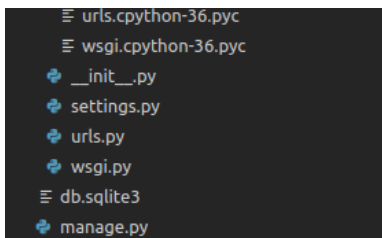
Myapp/item/models.py
관계 설정

```

1  from django.db import models
2
3  class Ingredient(models.Model):
4      name = models.CharField(max_length = 50)
5      oily = models.SmallIntegerField()
6      dry = models.SmallIntegerField()
7      sensitive = models.SmallIntegerField()
8
9      class Meta:
10         db_table = "ingredient"
11
12  class Item(models.Model):
13      imageId = models.CharField(max_length = 500)
14      name = models.CharField(max_length = 150)
15      price = models.IntegerField()
16      gender = models.CharField(max_length = 50)
17      category = models.CharField(max_length = 100)
18      monthlySales = models.IntegerField()
19      ingredients = models.ManyToManyField(Ingredient, through='itemingredientmapping')
20
21      class Meta:
22         db_table = "item"
23
24  class ItemIngredientMapping(models.Model):
25      item = models.ForeignKey(Item, on_delete = models.CASCADE)
26      ingredient = models.ForeignKey(Ingredient, on_delete = models.CASCADE)
27
28      class Meta:
29         db_table = "itemingredientmapping"
30
31  class ItemSkinType(models.Model):
32      item = models.ForeignKey(Item, on_delete = models.CASCADE)
33      first_skin_type = models.CharField(max_length = 20)
34      first_skin_score = models.SmallIntegerField()
35
36      class Meta:
37         db_table = "itemskintype"
38
  
```

2. 앱구조





Myapp : 프로젝트
Item : 앱 1개

3.URL구조

Myapp (프로젝트) urls.py

```
1 from django.urls import path, include
2
3 urlpatterns = [
4     path('', include('item.urls')),
5 ]
6
```

Item(앱) urls.py

```
from django.urls import path
from .views import PreProcessing, itemList, itemDetail, GetSkinType

urlpatterns = [
    # 제공되는 API Json 데이터 전처리 및 DB저장 Api
    path('data', PreProcessing.as_view()),

    # 상품 별 스킨 타입 계산 및 결과를 저장하는 테이블 저장 Api
    path('skin-type', GetSkinType.as_view()),

    # 문제 1번 (파라미터는 로직으로 해결)
    path('products', itemList.as_view()),

    # 문제 2번 (id를 제외한 파라미터는 로직으로 해결, id는 위와 구분하기 위해)
    path('product/<int:id>', itemDetail.as_view()),
]
```

필수, 옵션 쿼리 스트링해결은 여러가지 방법이 있었으나 반드시 위의 명시된 순서로 GET 요청이 들어온다는 전제가 없었으므로 views.py에서 문제를 해결 함

4. View API로직

코드 주석 참조 부탁드립니다.

5.실행 방법

Settings.py 에서 DATABASES에서 로컬디비 USER, NAME을 본인 DB에 맞게 수정해 주시면 됩니다.

