Projekt IAS – KAWIARNIA

Członkowie zespołu :

Piotr Kowalczyk 149014

Szymon Marczek 149025

Jakub Kraczkowski 149016

Struktura Projektu :

API Odpalane przez 3 LocalHosty(Api1 LocalHost1-Pl, Api2 LocalHost2-Eng, Hub LocalHost0)

Używane technologie : Python, PostgreSql, BackEnd: Django (Framework).

1. Bazy danych

ORM(Object Relational Mapping) Z poziomu kodu Pythona tworzy zapytania do bazy danych na zapytanie SQL

Model w ORM odpowiada tabeli SQL.

Migracja Bazy danych, na podstawie modeli w aplikacji tworzy tabele w bazie danych

1. API

Model ViewSet tworzy zestaw widoków do wyświelania i edycji obiektów odpowiadającego modelowi

queryset = Coffee.objects.all()  
serializer\_class = CoffeeSerializer  
permission\_classes = [AllowAny]

**Serializer przekształca Json na Python obiekty**

@admin.register(Coffee)  
**class** CoffeeAdmin(admin.ModelAdmin):  
 **pass**

**Rejestruje model Coffe I umożliwia edycje tabeli w admin panelu.**

1. **HUB**

**Settings – wszystkie sa default, oprocz:**

API\_POL\_URL = **'http://localhost:8001/kawa/'**API\_ENG\_URL = **'http://localhost:8002/coffee/'**

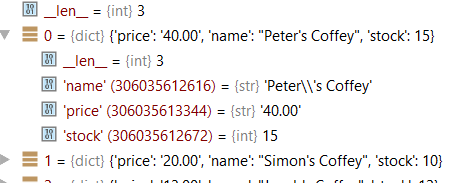
**Aby było wiadomo z czego ciagnac dane.**

**URL rejestruja widok i odwołują się do integration view.**

**Integracja:**

**Korzystamy z biblioteki requests, która pozwala na wysylanie requestow do naszego api’s.**

**Request Get zwraca dane w formie dictionary z zawartością odpowiadajaca JSON:**

****

Przeksztalcenie naglowkow polskich na angielskie:

normalized\_pol = [  
 {MAPPING[k]: v **for** k, v **in** d.items()} **for** d **in** response\_pol]

Na przyklad:

MAPPING = {**'nazwa'**: **'name'**,  
 **'cena'**: **'price'**,  
 **'ilosc'**: **'stock'**}

Do kazdego slownika dodaje pole osobne dla wersji pol/eng:

normalized\_pol = [{CURRENCY: **'PLN'**, \*\*d} **for** d **in** normalized\_pol]  
normalized\_eng = [{CURRENCY: **'USD'**, \*\*d} **for** d **in** response\_eng]  
**return** Response(normalized\_pol + normalized\_eng)