ТЗ

Для тестирования корректности работы микросервиса генерации данных реализовать небольшой RestApi на FastApi. Rest должен принимать Get и Post запросы, автоматически валидировать сгенерированные json, генерировать фейковые данные для возврата через get запрос.

Для генерации фейковых данных использовать библиотеку faker. Пример использования:

import faker

import uuid

faker.Faker.seed(0)

fake = faker.Faker(locale='ru\_RU')

fake\_data = []

for \_ in range(10):

    fake\_data.append(

        {

            'uuid\_scos': str(uuid.uuid4()),

            'external\_id': fake.pyint(min\_value=100000, max\_value=999999),

            'title': fake.word(),

            'direction': fake.word(),

            'code\_direction': fake.phone\_number(),

            'start\_year': int(fake.date(pattern='%Y')),

            'end\_year':int(fake.date(pattern='%Y'))

        }

    )

print(\*fake\_data, sep='\n')

необходимо реализовать классы генератора фейковых данных по схемам данных:

1. Схема данных структуры edu\_prog

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Описание | Пример значения |
| external\_id | string | Непустая строка, 6-значное рандомное число | 123456 |
| title | string | Непустая строка | премьера |
| direction | string | Непустая строка | выражение |
| code\_direct ion | string | Непустая строка, до 10 символов, состоит из цифр | 8 048 764 75 93 |
| start\_year | integer | Число, год начала | 2010 |
| end\_year | integer | Число , год окончания | 2012 |

(Год начала не обязательно должен быть после года окончания для фейковых данных)

1. Схема структуры данных up:

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Описание | Пример значения |
| external\_id | string | Непустая строка, уникальный идентификатор (uuid4) | 6df36fae-d429-48b1-ab45-bffc93823f6f |
| title | string | Непустая строка | премьера |
| direction | string | Непустая строка | выражение |
| code\_direct ion | string | Непустая строка, до 10 символов, состоит из цифр | 8 048 764 75 93 |
| start\_year | integer | Число, год начала | 2010 |
| end\_year | integer | Число , год окончания | 2012 |
| education\_form |  | Одно из значений примера, рандомно выбирать одно из них | EXTRAMURAL, FULL\_TIME, PART\_TIME, SHORT\_EXTRAMURAL, SHORT\_FULL\_TIME, EXTERNAL |
| educational\_program |  | Непустая строка, уникальный идентификатор (uuid4) | 6df36fae-d429-48b1-ab45-bffc93823f6f |

(Год начала не обязательно должен быть после года окончания для фейковых данных)

В Api должны быть эндпоинты для обращения к генератору фейковых данных, при get запросе на которые будет возвращаться сгенерированный json с фейковыми данными. Пример корректно сгенерированного json по 1 пункту:

[

    {

        "external\_id": "985440",

        "title": "премьера",

        "direction": "выражение",

        "code\_direction": "8 048 764 75 93",

        "start\_year": 2004,

        "end\_year": 1979

    },

    {

        "external\_id": "395528",

        "title": "мгновение",

        "direction": "сопровождаться",

        "code\_direction": "+7 948 924 11 57",

        "start\_year": 2008,

        "end\_year": 1976

    },

    {

        "external\_id": "470977",

        "title": "художественный",

        "direction": "деньги",

        "code\_direction": "83877840801",

        "start\_year": 2018,

        "end\_year": 1997

    }

]

Обработчики Rest Api

1. Get /edu\_prog?size={int}

Size = количество обьектов внутри одного респонса (1 обьект – 1 словарь)

1. Get /up?size={int}

Size = количество обьектов внутри одного респонса (1 обьект – 1 словарь)

1. Post /edu\_prog

Post запрос должен принимать на вход сгенерированный json по шаблону edu\_prog\_json(будет ниже). Должна производиться автоматическая валидация данных на корректность генерации (pydantic) и выводиться в консоль результат валидации, записываться в НОВЫЙ файл \*.json и возвращать сгенерированное имя файла как респонс

1. Post /up

Post запрос должен принимать на вход сгенерированный json по шаблону up\_json(будет ниже). Должна производиться автоматическая валидация данных на корректность генерации (pydantic) и выводиться в консоль результат валидации, записываться в НОВЫЙ файл \*.json и возвращать сгенерированное имя файла как респонс

1. /is\_alive

Возвращает True если rest api работает

Rest api должен запускаться как сервер на localhost в отдельном окне терминала.

Микросервис для генерации данных для отправки в формате JSON.

Общая схема работы микросервиса:

1. Запуск и парсинг параметров командной строки
2. Выполнение Get запроса
3. Обработка полученных данных(или ошибки)
4. Генерация json согласно схеме(up или edu\_prog)
5. Отправка post запроса на api со сгенерированным json
6. Обработка ответа (успешный – вывод в консоль, ошибка вывод в файл с логами)
7. Завершение работы сервиса

Микросервис запускается из терминала с аргументами, выполняет работу, завершается после отправки запросов и получения ответов ( для удобной работы с аргументами командной строки один из вариантов argparse)

Пример запуска микросервиса для генерации:

$ python servece\_name.py –schema up --size 4

Шаблон данных edu\_prog\_json

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | Тип | Описание | Пример |
| organization\_i d | Uuid | Уникальный идентификатор uuid4 | 8e5bb17a-524d-4277- 80d2-2478304758e2 |
| edu\_prog | json | Список учебных планов | См пример 1 |

Пример 1 сгенерированный json для отправки в api:

{

    "organization\_id": "8862062d-80bd-4b2c-8a7a-12e9c88c7378",

    "edu\_prog": [

        {

            "external\_id": "985440",

            "title": "премьера",

            "direction": "выражение",

            "code\_direction": "8 048 764 75 93",

            "start\_year": 2004,

            "end\_year": 1979

        },

        {

            "external\_id": "395528",

            "title": "мгновение",

            "direction": "сопровождаться",

            "code\_direction": "+7 948 924 11 57",

            "start\_year": 2008,

            "end\_year": 1976

        },

        {

            "external\_id": "470977",

            "title": "художественный",

            "direction": "деньги",

            "code\_direction": "83877840801",

            "start\_year": 2018,

            "end\_year": 1997

        }

    ]

}

Шаблон данных up\_json

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | Тип | Описание | Пример |
| organization\_i d | Uuid | Уникальный идентификатор uuid4 | 8e5bb17a-524d-4277- 80d2-2478304758e2 |
| up | json | Список учебных планов | См пример 1 |

Список учебных планов аналогичен списку образовательных программ, только состоит из объектов учебных планов.

Рекомендуемый стек: FastApi, argparse, pydantic, Faker, uvicorn, requests и дальше на свой выбор, желательно с поддержкой async/await