

Домашнее задание №4

(срок выполнения: 1-2 недели)

Flask

Вам нужно будет разработать интерфейс RESTful API на основе Flask. Интерфейс должен поддерживать методы для поиска фильма по id, по названию и по году выхода, а также выдачу рекомендаций по запросу пользователя.

Описание интерфейса

1. Поиск фильма (GET-запрос):

- по id = tt0000001: **/movie_id/tt0000001**
- по названию title = Carmencita: **/movie/Carmencita**
- по году выхода год=1894 : **/year/1894**

В теле HTTP ответа во всех случаях должен быть JSON-документ:

- пустой документ, если ничего не найдено (при этом код ответа должен быть 404)
- при поиске по id: JSON-документ, содержащий атрибуты фильма

```
{
  "movie": {
    "tconst": "tt0000001",
    "title": ""
  }
}
```

- При поиске по году или названию: JSON-документ, содержащий список фильмов, удовлетворяющих запросу. Например,

```
{
  "movies": [
    "tt0000001",
    "tt0000002",
    "tt0000002"
  ]
}
```

2. Рекомендация (POST-запрос):

- Получить topk рекомендаций: **/suggest/topk**

Например, `"/suggest/3"` означает запрос на выдачу 3 рекомендаций.

В теле запроса передается JSON-документ, содержащий непустой список из бинарных оценок фильмов, то есть пар (id_фильма, оценка).

Например,

```
{
  "likes": {
    "tt0000001": 1,
    "tt0000002": 0,
    "tt0000010": 0
  }
}
```

- В теле ответа передается JSON-документ, содержащий список из topk рекомендованных фильмов в формате (id_фильма, число),

Например,

```
{
  "ratings": {
    "tt0000003": 0.7,
    "tt0000005": 0.2,
    "tt0000008": 0.1
  }
}
```

Примечание: в рекомендациях не могут передаваться те фильмы, которые пользователь уже оценил (то есть те, которые пришли в теле запроса). Выбирать числа (предсказанный рейтинг) можно произвольно.

Данные

Необходимо использовать данные IMDB (<https://www.imdb.com/interfaces/>). Список фильмов доступен по ссылке: <https://datasets.imdbws.com/title.basics.tsv.gz>. В процессе тестирования можно использовать первые 100 записей.

Развертывание приложения

Необходимо подготовить Dockerfile для развертывания приложения на сервере. Собрать образ приложения и запустить docker-контейнер на сервере (выбрать внешний порт так, чтобы не было конфликтов с другими разработчиками аналогичных приложений).

Опциональное задание

Хранить данные о фильмах в базе данных (postgres, mysql или sqllite3), а не в памяти сервера.