

项目微服务API定义定义

推荐系统中一共有4个微服务，其中3个为后端服务（召回服务、排序服务、API服务），另外还有一个前端微服务负责提供网页相关的静态文件。

1. Recall Service 召回服务

召回服务一共提供2个API，分别对应着“猜你喜欢”和“相似推荐”两种场景下的召回逻辑：

- 获取猜你喜欢召回结果
 - GET /recall
 - URL query参数：
 - user_id: int类型，可为空。当前访问的用户id
 - 返回值：
 - 类型：JSON数组
 - 召回的动漫id数组
- 获取相似推荐召回结果
 - GET /sim
 - URL query参数：
 - anime_id: int类型，不可为空。需要推荐的动漫id
 - 返回值：
 - 类型：JSON数组
 - 召回的动漫id数组

```
from flask import Flask, jsonify, request

app = Flask('recall-service')

@app.route("/recall")
def get_anime():
    user_id = request.args.get('user_id', type=int)
    print(f'Calling user {user_id}')
    return jsonify([1, 2, 3])

@app.route("/sim")
def get_sim_anime():
    anime_id = request.args.get('anime_id', type=int)
    if anime_id is None:
        return 'bad anime id', 400
    print(f'Calling anime {anime_id}')
    return jsonify([4, 5, 6])
```

2. Rank Service 排序服务

排序服务提供1个API，对应“猜你喜欢”推荐场景。

- 获取猜你喜欢排序结果
 - GET /rank
 - URL query参数：
 - user_id: int类型，可为空。当前访问的用户id
 - 返回值：
 - 类型：JSON数组
 - 排序后的动漫id数组

3. API Service API服务

API服务所起到的功能是接受前端用户的请求，调用召回、排序服务相应接口获取推荐结果，然后组装数据，使其满足前端展示需要，最后返回结果。因此，我们首先要实现如下两个接口：

- 获取猜你喜欢结果

- GET /recommends
- URL query参数：
 - user_id: int类型，可为空。当前访问的用户id
- 返回值：
 - 类型：JSON数组
 - 推荐的**动漫对象**数组
- 应调用排序服务的rank接口

- 获取相似推荐结果

- GET /sim
- URL query参数：
 - anime_id: int类型，不可为空。需要推荐的动漫id
- 返回值：
 - 类型：JSON数组
 - 推荐的**动漫对象**数组
- 应调用召回层的sim接口（或同学们自己实现的排序层相应接口）

这里和之前有一点重要的区别，就是这两个API的返回值不再是动漫id数组了，而是动漫对象的数组。因为前端网页在给用户展示推荐结果的时候，显然不能只展示一堆id。因此API服务承担了将id转化为对象的职责。这个步骤一般会通过查数据库或缓存来完成。由于目前我们不需要真的实现这个操作，所以大家可以先返回一个假的动漫对象即可。

此外，由于前端页面在展示动漫的时候，会遇到展示一个动漫详情的需求（在动漫详情页中），因此我们还需要增加一个业务接口，为这个需求提供数据：

- 获取某个动漫详情

- GET /anime/:id
- URL path参数：
 - id: int类型，不可为空。需要查询的动漫id
- 返回值：
 - 类型：JSON对象
 - 所查询的**动漫对象**