

课程项目微服务API定义

在我们要实现的推荐系统中一共有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。

在本章中，我们将会搭建3个后端服务的基本框架。每个服务都是一个独立的python flask应用，对外暴露必要的接口。服务间互相通过HTTP调用通信，这样可以做到服务间互相独立，将整个系统各个组件之间的耦合度降低，从而增强系统的可维护性和可靠性。

下面我会给出3个服务各自的API接口文档，希望同学们可以花些时间用flask把这3个服务的接口写出来。这样即可以帮助大家更好的熟悉整个系统的整体架构，另一方面如果同学们对python和flask还不是太熟练，也可以借此练手，方便更好的完成后续课程。

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

由于目前还没有真正的实现召回逻辑，因此大家可以先返回一些假数据，方便后续接口调试。同学们完成实现后，可以参考以下代码，对比看一下我们的实现有没有什么不同。

```
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

相信通过之前召回服务的练习，你一定可以很快写出排序服务的API了。

3. API Service API服务

好了，到此为止召回和排序服务的接口我们都已经实现出来了。接下来我们需要编写第三个业务服务：API Service。

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**

类似的，我们先不用真正实现这个功能，仅需返回一个假数据，使得接口跑通即可。

在有了以上3个微服务之后，我们已经初步完成了推荐系统的后端逻辑框架搭建，后面的课程中我会和大家一起，在此基础上一点点的去完善真正的推荐功能，最终实现一个完整的推荐系统。