

Python后端组第二轮考核

- 知识点
- 推荐教程
- 单人作业
- 作业要求
- 考核截止日期
- 提交方式

知识点

1. WebSocket, SocketIO
2. 日志
3. 权限
4. 参数校验
5. 邮箱/图形验证
6. 缓存
7. 测试
8. 服务器部署
9. docker

推荐教程

1. [Flask-SocketIO](#)经常打不开，不过网上有很多它的翻译，可以查看
2. [fastapi-socketio](#)文档都没有，不过网上有一些实战经验，可以进行参考
3. [Socket.IO](#) 最完善的SocketIO文档，不过服务端是Nodejs，不是Python，可以参考借鉴，客户端api在写前端时可能会用到
4. 推荐使用logging日志工具打印运行时的日志
5. 在登陆后在token中添加用户的权限信息，对于需要token的接口在JWT解码时进行权限检验和进行拦截，Flask使用装饰器，FastApi可以选择依赖注入，或者使用中间件拦截
6. FastApi可以使用自带的参数检验，Flask应该引入Flask-Restful，规范RestfulAPI，同时使用其带的参数检验功能[Flask-RESTful](#)
7. 在框架中接入redis等进行缓存，读写很快，可以用于缓存验证码，以及一些经常读写的信息
8. 使用pytest编写单元测试，使用pytest-cov查看单测覆盖率（发送请求并传入参数，判断结果是否一致，好的产品单元测试覆盖率应该达到85%）
9. 作为一个程序员怎么能没有一个属于自己的服务器呢？如果有点钱的话，大家可以去腾讯云、阿里云等云服务商购买一个便宜的云服务器（新人首单有优惠，要买就买久一点，一年100左右吧），同时将自己的项目部署到服务器上。服务器上请使用 uWSGI 或 gunicorn 启动项目，以获取更大的性能。（还可以用做部署定时爬虫，在线图床，个人网站等，大家可以自己尝试）
10. 部署项目环境弄起来非常的难受，建议大家学习一下docker，使用docker将项目部署到服务器
11. 建议为代码编写注释以及使用typehint，增加代码的可读性，在每天写完代码后，或者解决什么问题后，可以写一下笔记，方便以后再进行回顾
12. 本次项目给的时间应该会很长，对于Flask和FastApi这两个框架其实还有很多可以拓展的地方，[awesome-flask](#)和[awesome-fastapi](#)列举了很多这些框架的相关精彩项目和插件，可以自主进行学习。同时大家也不要把自己局限于Python上，应该多了解一些别的语言的相关写法，比如Go，Java，或者前端。但是不允许**什么都不学**。本轮考核结束后，我会问一些你在这段时间内学到的东西和收获。

多人作业

二手游戏账号交易平台，参考交易游或闲鱼

平台支持功能

支持聊天功能

- 1

支持文字、图片交流
- 2

保存聊天记录
- 3

支持屏蔽功能
- 4

有聊天界面和消息提醒
- 5
- 6

可以接受系统发送消息

支持黑名单功能

- 1

用户举报或平台检测
- 2

↓
- 3

管理员审核，决定加入黑名单
- 4

↓
- 5

回复用户审核情况
- 6
- 7

黑名单用户只能使用充值和提现功能和接受系统消息，其他功能不允许使用，且昵称头像进行特殊处理

支持后台审核

- 1

管理员权限： 审核商品，审核用户举报
- 2

管理员有处理事故的功能

支持个人中心

- 1

用户买卖家一体
- 2

个人中心可查看：

个人信息： 用户名，昵称，头像，修改密码，绑定邮箱，实名认证
- 3

买家中心： 已购买订单，收藏订单，出价订单
- 4

卖家中心： 出售商品，已完成订单，买家的出价
- 5

余额钱包： 充值，提现

支持事故处理

- 1 目前主要有两大情况：
- 2 1. 交易完成后，卖家找回账号
- 3 处理手段：紧急冻结卖家钱包，管理员审核账号情况。情况属实，若卖家钱包足够交易价格，则按交易价格返回给买家，解冻卖家钱包，将卖家加入黑名单；
- 4 若钱包不足交易价格，将钱包余额全数返回买家（平台可以自行补偿），并将卖家加入讨债列表和黑名单
- 5
- 6 2. 买家收货后，取消交易，卖家检查账户发现问题拒绝取消交易
- 7 处理手段：管理员审核账号情况，账号确实受损，如卖家同意，按照账号受损情况将交易金按协议比例发给卖家，剩余交易金返回买家
- 8 如账号价值严重受损，没有什么交易价值，则将交易金全部发给卖家，通知买家，且买家加入黑名单
- 9 如卖家不同意，则进行协调程序
- 10 账号并没有价值受损，则告知卖家，结束交易过程，交易金返还买家
- 11 如卖家不同意，则进行协调程序

作业要求

1. 上述基本功能
2. 项目结构合理
3. 日志功能
4. 单元测试（不包含websocket和SocketIO，覆盖率至少80%）
5. 部署（可选）
6. 多人作业如果对方摆烂，也请尽量完成自己的任务
7. 将作业提交到GitHub上，提交push到的github的网址

考核截止日期

2023/6/10

提交方式

到时候再说