后端任务

第一阶段

> 利用前后端的demo进行开发

后端玩家需要使用前端demo,然后按照接口文档进行后端接口的开发

开发完成后,你就可以在前端demo里跑通整个项目了

你的任务

- 开发之前,你需要理解,什么是前后端分离的web开发,前后端的交互过程是什么样(jsp和其他模板引擎是比较过时的技术,所以在本项目里是不适用的)
- 按照接口文档完成后端所有接口的开发(最基本的crud)
- 开发语言不限,可以是java go php等,满足接口规范即可
    - 可以使用框架,比如java建议springMVC或者springboot,go建议go-gin,python有flask和django等
- 你需要自己设计数据对象(JavaBean)或者结构体
- 至于数据存储在哪里,在第一阶段,可以直接在代码中在内存中创建一个数组或者其他数据结构来进行存储(当然,你想连接到数据库也行)
- 后端需要对传过来的参数进行合法性校验(可以了解下JSR303相关)
- 在开发过程中尝试对你的代码进行分层,可以参考java常见的controller service dao三层模型的工作过程

第二阶段

> 在第一个阶段中,你已经能让整个系统跑起来了,但是作为真实工作的系统,还是有很多不好的地方,所以试着用到下面的技术对你的系统进行改良
>
> 可能这个阶段demo给出的前端就不适用了,因为有些功能修改是要前后端一起完成的,所以请找一个前端小伙伴(可以多个后端找一个前端)一起完成

任务1:数据库连接操作

(无需前端配合)

尝试实用一些数据库连接工具或者orm框架,把整个系统的数据存放在数据库里

对于java,可以了解下面这几种:

- spring data
- JPA
- Mybatis

对于go,可以试试这个:

- gorm

其他语言也有类似的

任务2:解决跨域问题

- 试着找一个前端同学,让他在他的电脑上启动前端服务,然后通过内网ip来调用你的后端服务,看看是否可以正常工作
- 如果出问题了,请你去了解下什么是跨域问题,以及如何解决

任务3:分页接口的设计

- 还记得之前有返回todolist或者info数组的场景吗,如果这时候有很多元素,比如100个,那么来回返回那么多数据开销很大而且前端也不可能一次展示几百个数据,所以会需要有分页功能:
- 

- 所以,前端后端的同学需要协商一下如何改进接口,才能达到分页展示的效果

任务4:添加小功能

需要前后端一同开发

- 后端能否实现记录某个info的浏览次数的功能呢?
- 你能否为info或者todolist增加简单的搜索功能和排序功能?

任务5:完善会话追踪

登录后的会话追踪and页面拦截效果

- cookie?
- session?
- jwt?

任务6:权限管理 RBAC

> 可能需要前后端配合新增一些接口

记得之前不是说人人都能修改info吗

那我们现在能不能给系统做一个权限划分,比如增加一个管理员角色,只有管理员能对info进行增加修改删除操作,其他普通人只能看info而不能修改

你需要完成

- 对于用户对象新增加一个等级属性,分为normal和vip两个等级

- 增加一个`充钱`接口(充不充钱无所谓),能把一个普通用户变为vip
- 之前info的增加修改删除操作的接口进行权限管理,只允许vip用户修改

第三阶段

> 试试调用各种花里胡哨的功能?

任务1:线上部署实战

- 之前的服务都是布置在你们自己的电脑上的,但是真正的系统肯定是跑在服务器上的
- 所以你们需要学习一些运维相关的知识,怎么用nginx和tomcat之类的,把这个系统打包部署到服务器上(可以是你们自己的云服务器也可以是工作室的主机上)

任务1plus:CICD

有兴趣可以试试,跳过也不影响

- 每次部署之后,如果程序要有新的改动,是不是需要又重新编译打包然后上传到服务器再重启折腾半天呢?能不能修改完了一键上线呢?
- 了解下什么是CICD,可以以Jenkins为例,搭建一台CICD服务器,配合Gitlab或者Github,帮你完成一键部署(建议拿工作室的机子去玩,因为这些东西比较吃性能,云服务器可能比较贵)

任务1plus plus:接口文档自动生成

有兴趣可以试试,跳过也不影响

- 当我们每次新写了接口之后,还要去写文档进行说明,是不是很麻烦
- 所以,了解下有没有什么工具可以自动生成接口文档
- 以swagger接口文档生成器为例,试着配置使用下,看看能否自动为你的后端生成接口文档

任务2:认证登录

需要前端配合

去网上找一些云服务,然后结合去修改你的登录逻辑

试试能不能实现下面随意一种登录方式

(前提是你可能需要让用户注册的时候填写更多的信息来支持这个功能)

- 短信认证登录
- 邮箱验证码登录
- github oauth登录?(难度有点大,但是可以去了解下这个是什么)

任务3:图片存储?

想想能不能配合前端,让系统登录后用户有自己的头像?

对于后端而言,重点在于,你打算怎么存头像图片?

- 本地写代码存图片?
- OSS存储服务?

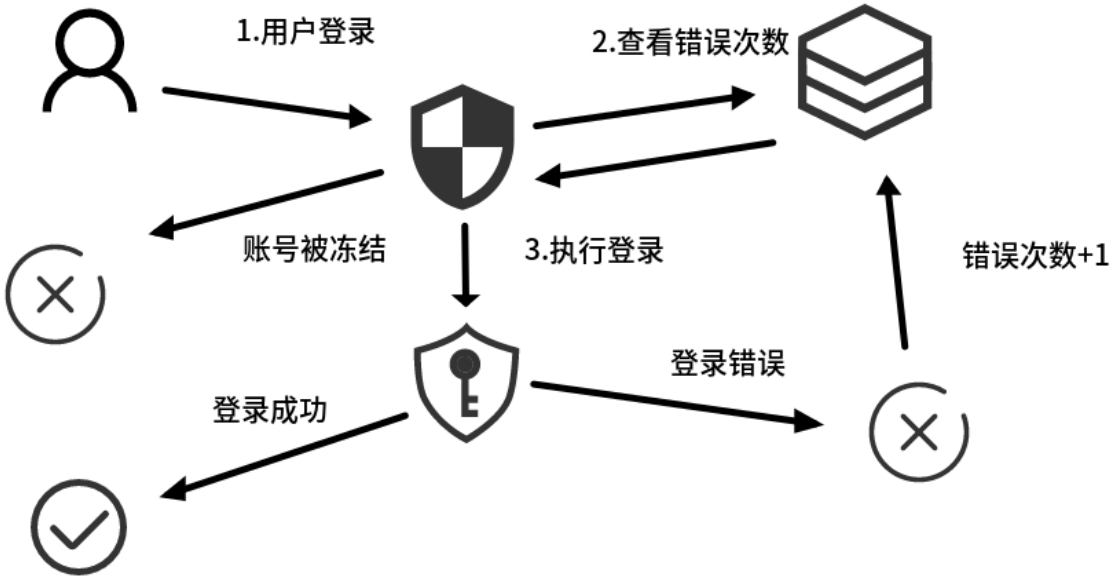
- 图床?

任务4:账号冻结功能

试图完成这个功能:

- 多次登录密码错误,账号冻结5分钟后才可以重试

思路如下:



可以了解一个叫做redis的东西的使用

用redis来配合完成这个功能

Tips

- 调试接口的时候,可以下载一个叫做Postman的工具并自己学着使用

```
//接口文档中标准的响应格式如下:
response {
    • Success
    • Code
    • Msg:
    • Data: json api or sth
}
//你可以去了解 什么是restful api ,看看别人设计的返回值,从而理解返回值这样设计的原因
```

- 在系统写的差不多的时候,前后端的同学也可以一起商量下再添加些什么接口,来优化这个小程序的体验,设计新接口的时候记得使用`restful风格的路由和返回值`