

# 知识分享小程序

## 前端页面展示效果

### 首页

- 未登录(未登录展示的都是兑换)



- 登录(登录后属于自己的分享才会显示下载)



- 兑换(点击兑换，该资源由兑换变成下载 返回首页)

时空斗甲行  
作者：萧潜



。

积分：\$ 23

兑换

- 下载(复制下载地址)

你代码大全

复制下载地址

o

投稿(增加一条投稿)

●●●●● WeChat

16:43

100%

投稿

...

说明：投稿审核通过后会有积分奖励，资源被下载会有积分奖励；提交的资源不得包含广告，侵权信息，百度网盘地址建议有密码

原创

☒

转载

☐

标题

请输入标题

作者

请输入作者

价格

请输入价格

简介

介绍一下技术干货吧

下载地址

请输入下载地址

提交

首页

投稿

我的

我的

- 未登录



登录,享受技术之旅吧!

微信账号登录



- 签到(点击签到，积分增加)



- 积分明细(积分明细包括 投稿 签到 兑换)



●●●●● WeChat 16:48 100%		
<div> <div>&lt;</div> <div>积分明细</div> <div> <div>...</div> <div></div> </div> </div>		
	2020-10-15 18:24:03- 兑换分享	-15
	2020-10-16 10:44:36- 兑换分享	-5
	2020-10-16 15:59:11- 签到加分	20
	2020-10-17 13:59:34- 签到加分	20
	2020-10-18 12:26:27- 投稿加积分	50
	2020-10-18 12:32:41- 投稿加积分	50
	2020-10-18 13:40:43- 兑换分享	-13
	2020-10-18 13:46:57- 投稿加积分	50
	2020-10-18 13:48:01- 兑换分享	-17
	2020-10-18 16:46:42- 签到加分	20

- 
- 我的投稿(查看自己的投稿)

标题:北国谍影

作者:寻青藤

投稿时间:2020-10-17 12:30:46



简介: 热血战斗, 保家卫国

审核状态: 待审核

标题:11

作者:123

投稿时间:2020-10-18 13:45:57



- 投稿审核
  - 待审核(投稿之后进入待审核状态)



- 审核(通过下拉框选择状态 同时输入原因)

标题:11

作者:123

投稿时间:2020-10-18 13:45:57



简介: summaey

请选择审核状态

请选择审核状态

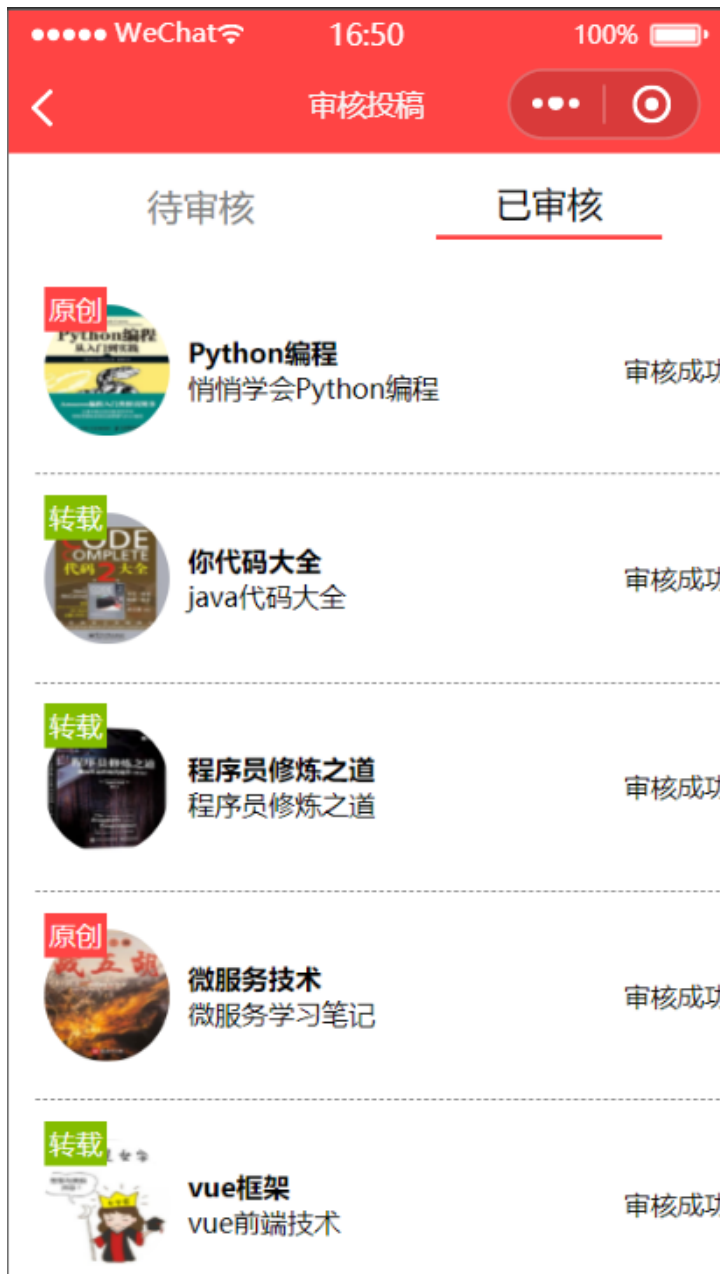
PASS

REJECT

请输入原因

提交审核

- 
- 审核成功(查看所有审核成功的投稿)



## 后端接口文档(代码均已上传至github)

2.3. 兑换接口	2.3. 兑换接口
2.3.1. 兑换某个分享	Share Admin Controller
2.3.2. auditInsertById	
2.4. 公告接口	
2.4.1. 查询最新一条公告	
2.5. 分享接口	2.3.1. 兑换某个分享
2.5.1. 投稿接口	PUT /admin/shares/audit/{id}
2.5.2. 兑换分享接口	Description
2.5.3. 获取该用户的投稿	兑换某个分享
2.5.4. 分享列表	Parameters
2.5.5. 查询指定id的分享详情	
3. Definitions	
3.1. ServiceInstance	
3.2. URI	
3.3. 兑换分享详情	
3.4. 公告	
3.5. 分享	
3.6. 分享详情, 带发布人昵称	
3.7. 审核详情	
3.8. 投稿详情	

### ● 技术栈

- jwt 来实现认证鉴权
- 微服务技术(Spring Cloud 已经 Spring Cloue Alibaba)

### ● 重点突破

## ○ 签到(时间进行比对)

```
@Override
public ResponseDTO signIn(UserSignInDTO signInDTO) {
    User user = this.userMapper.selectByPrimaryKey(signInDTO.getUserId());
    if (user == null){
        throw new IllegalArgumentException("该用户不存在!");
    }

    Example example = new Example(BonusEventLog.class);
    Example.Criteria criteria = example.createCriteria();
    example.setOrderByClause("id DESC");
    criteria.andEqualTo("property", "userId", signInDTO.getUserId());
    log.info(String.valueOf(signInDTO.getUserId()));
    criteria.andEqualTo("property", "event", "value", "SIGN_IN");
    List<BonusEventLog> bonusEventLog = this.bonusEventLogMapper.selectByExample(example);
    //判断查出来的bonusEventLog是否为空, 空则直接插入, 不为空再进行时间的判断, 否则会报下标越界, 就这么加, 两个方法都加
    if (bonusEventLog.size() == 0){
        BonusEventLog bonusEventLog1 = BonusEventLog.builder()
            .userId(signInDTO.getUserId())
            .value(20)
            .event("SIGN_IN")
            .createTime(new Date())
            .description("签到加分")
            .build();
        this.bonusEventLogMapper.insert(bonusEventLog1);
        user.setBonus(20+user.getBonus());
        this.userMapper.updateByPrimaryKey(user);
    }else {
        BonusEventLog bonusEventLog1 = bonusEventLog.get(0);
        Date date = bonusEventLog1.getCreateTime();
        try {
            if (DateUtil.checkAllotSign(date) == 0) {
                this.bonusEventLogMapper.insert(BonusEventLog.builder()
                    .userId(signInDTO.getUserId())
                    .event("SIGN_IN")
                    .value(20)
                    .description("签到加分")
                    .createTime(new Date())
                );
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

## ■ 是否签到 (判断)

```
@Override
public ResponseDTO checkIsSign(UserSignInDTO signInDTO) {
    User user = this.userMapper.selectByPrimaryKey(signInDTO.getUserId());
    if (user == null) {
        throw new IllegalArgumentException("该用户不存在");
    }

    Example example = new Example(BonusEventLog.class);
    Example.Criteria criteria = example.createCriteria();
    example.setOrderByClause("id DESC");
    criteria.andEqualTo("property", "userId", signInDTO.getUserId());
    criteria.andEqualTo("property", "event", "value", "SIGN_IN");
    List<BonusEventLog> bonusEventLog = this.bonusEventLogMapper.selectByExample(example);
    //判断查出来的bonusEventLog是否为空
    if (bonusEventLog.size() == 0) {
        return new ResponseDTO("succ: true", "code: 200", "msg: 该用户还没有签到", "data: 可以签到", "ts: 11");
    } else {
        BonusEventLog bonusEventLog1 = bonusEventLog.get(0);
        Date date = bonusEventLog1.getCreateTime();
        try {
            if (DateUtil.checkAllotSign(date) == 0) {
                return new ResponseDTO("succ: true", "code: 200", "msg: 该用户还没有签到", "data: 可以签到", "ts: 11");
            } else if (DateUtil.checkAllotSign(date) == 1) {
                return new ResponseDTO("succ: false", "code: 201", "msg: 已经签到了", "data: 不可以签到", "ts: 11");
            } else if (DateUtil.checkAllotSign(date) == 2) {
                return new ResponseDTO("succ: false", "code: 202", "msg: 数据出错了", "data: 不可以签到", "ts: 11");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    return new ResponseDTO("succ: true", "code: 200", "msg: 该用户还没有签到", "data: 可以签到", "ts: 11");
}
```

## ○ 审核投稿

```

public Share auditById(Integer id, ShareAuditDto shareAuditDto) {
    // 1 查询share是否存在, 不存在或者当前的audit_status != NOT_YET, 那么抛异常
    Share share = this.shareMapper.selectByPrimaryKey(id);
    if (share == null) {
        throw new IllegalArgumentException("参数非法! 该分析不存在");
    }
    if (!Objects.equals(a: "NOT_YET", share.getAuditStatus())) {
        throw new IllegalArgumentException("参数非法! 该分享已审核或者审核不通过!");
    }
    // 2 审核资源 将状态改为PASS或REJECT
    // 这个API的主要流程是审核, 所以不需要等更新积分的结果返回, 可以将积分改为异步
    share.setAuditStatus(shareAuditDto.getAuditStatusEnum().toString());
    this.shareMapper.updateByPrimaryKey(share);

    //3 如果是PASS, 那么发送消息给rocketmq, 让用户中心去消费 并为发布人添加积分
    if (AuditStatusEnum.PASS.equals(shareAuditDto.getAuditStatusEnum())) {
        this.rocketMQTemplate.convertAndSend(
            destination: "add-bonus",
            UserAddBonusMsgDto.builder()
                .userId(share.getUserId())
                .bonus(50)
                .build());
    }
    return share;
}

```

## ● 兑换加积分

```

public Share exchange(ExchangeDTO exchangeDTO) {
    int userId = exchangeDTO.getUserId();
    int shareId = exchangeDTO.getShareId();
    System.out.println("前端传回来的shareId: " + shareId);
    // 1 根据id查询share, 校验是否存在
    Share share = this.shareMapper.selectByPrimaryKey(shareId);
    if (share == null) {
        throw new IllegalArgumentException("改分享不存在!");
    }
    // 此处这个share也是为空的
    System.out.println(share);
    Integer price = share.getPrice();

    //2 如果当前用户已经兑换过该分享, 则直接返回
    MidUserShare midUserShare = this.midUserShareMapper.selectOne(
        MidUserShare.builder()
            .shareId(shareId)
            .userId(userId)
            .build()
    );
    if (midUserShare != null) {
        return share;
    }

    // 3 根据当前登录的用户id 查询积分是否够
    // 这里一定要注意通过用户中心接口得到的返回值, 外面已经封装了一层了, 要解析才能拿到真正的用户数据
    ResponseDTO responseDTO = this.userCenterFeignClient.findById(userId);
    UserDTO userDTO = convert(responseDTO);
    System.out.println("看看这个是不是空" + responseDTO);
    if (price > userDTO.getBonus()) {
        throw new IllegalArgumentException("用户积分不够!");
    }

    // 4. 扣积分
    this.userCenterFeignClient.addBonus(
        UserAddBonusDTO.builder()
            .userId(userId)
            .bonus(price * -1)
            .build()
    );

    //5. 向mid_user_share表里插入一条数据
    this.midUserShareMapper.insert(
        MidUserShare.builder()
            .userId(userId)
            .shareId(shareId)
            .build()
    );
}

```