目录

[一． 项目介绍 1](#_Toc13454)

[二． 项目结构 1](#_Toc17947)

[三． 模块划分 1](#_Toc13897)

[四． 项目架设 1](#_Toc3567)

# 项目介绍

## 基本介绍

项目内容：实现一个在线小说阅读平台，提供基础的小说阅读服务外，后续将添加基于用户认证后的用户激励模块，如：

积分机制：用户每日签到，阅读一定数量的章节可获得一定的积分奖励

商城机制：用户账号中的积分可消费换取商城中的内容

项目内容数据来源：通过爬虫的手段，编写爬虫自动抓取线上部分站点的数据，

将获取得到的数据进行清洗，去重，结构化处理后，于Mysql，MongoDB等数据库持久化保存。

项目管理：当前项目的相关代码文件进行存储于github的版本控制，项目链接：https://github.com/kaskeh/django-vue-novelSite/

# 项目结构

## 2.1 结构介绍

项目实现上：采取的是前后端分离开发模式

## 2.2 架构逻辑

前端向后端接口发送获取数据的请求，后端验证请求后返回相关数据或着拒绝返回并给与拒绝原因

## 2.3 使用技术

前端：Vue全家桶，axios，Element-UI，Bootstrap

后端：Django，Mysql，MongoDB

开发工具：Pycharm，VSCode，Navicat，PDManer（数据库模型）

开发环境：目前为Windows 10上开发，后续项目部署于linux系统上

## 2.4 数据库建模



# 模块划分

## 3.1 基础模块

### 3.1.1 主页

### 3.1.2 小说分类

#### 3.1.2.2 主页分类信息

当头部导航栏加载时，生命周期中的setup()会在渲染导航栏组件前，会先往后端拿取分类数据

// setup生命周期函数，较beforeCreate与created先前调用

  setup() {

    const headData = reactive({

      headers: [],

    });

    GetCates().then(

      (response) =>

        // console.log(response),

        (headData.headers = response.data.data)

    );

    return {

      headData,

    };

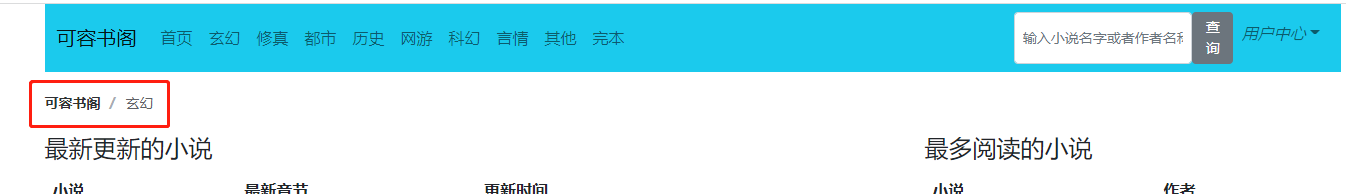
  },



#### 3.1.2.3 分类页信息

##### 3.1.2.3.1 面包屑导航栏

当前添加个 面包屑导航栏 的辅助功能，使用了element-UI上现成的组件，功能的目的是方便用户了解当前站点的深度



<!-- 添加一个面包屑导航栏 -->

      <el-breadcrumb :separator-icon="ArrowRight">

        <el-breadcrumb-item :to="{ path: '/' }">可容书阁</el-breadcrumb-item>

        <el-breadcrumb-item>玄幻</el-breadcrumb-item>

      </el-breadcrumb>

## 3.2 用户模块

当前设计中，用户模块基本放置于页面头部右侧，靠近搜索框，涵盖三个功能：用户登录，用户注册，登录后的用户信息显示

未登录前:



登录后的:



### 3.2.1 用户注册

<http://127.0.0.1:8080/register>

提交参数：

用户名（必填） 密码（必填） 邮箱（必填）

Username password emial

返回数据：



开发中由于前后端分离，产生的跨域，django的对表单post提交的csrf校验等问题，当前的注册页面基础逻辑基本逻辑为：

1. 当用户进入注册页面后，前端做了一个请求后端csrf\_token的处理，方便后面提交表单数据时的csrf问题
2. 前端校验用户输入信息是否符合规范
3. 输入符合规范，用户点击提交
4. 后端收到数据，验证用户提交的username是否已存在，如存在则返回错误提示
5. 如不存在则进行创建，并返回成功提示

### 3.2.2 用户登录

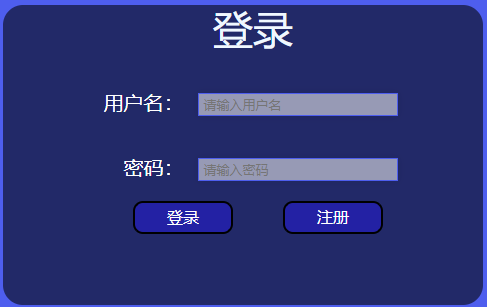
<http://127.0.0.1:8080/login>

提交参数：

用户名（必填） 密码（必填

Username password

返回数据：



对于登录，基于用户数据安全的考量，进行JWT的安全处理，后端django使用rest\_framework框架中的SIMPLE\_JWT

用户账号密码校验通过可直接跳转用户中心，校验不通过则返回错误信息

### 3.2.3 用户中心

<http://127.0.0.1:8080/userHome>

提交参数：

账号 邮箱 手机号

Username email mobilephone

当前用户界面是否能够查看的依据，当前暂时判断逻辑为：

查看localStorage中是否有 “username”， 不存在则跳转登录页面

mounted() {

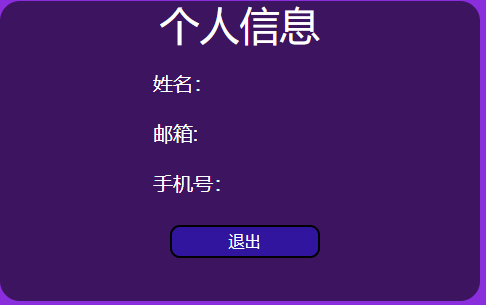
    if (!localStorage.getItem("username")) {

      console.log("当前用户信息不存在");

      this.$router.replace("/login"); //页面跳转至登录页面

    }

  },

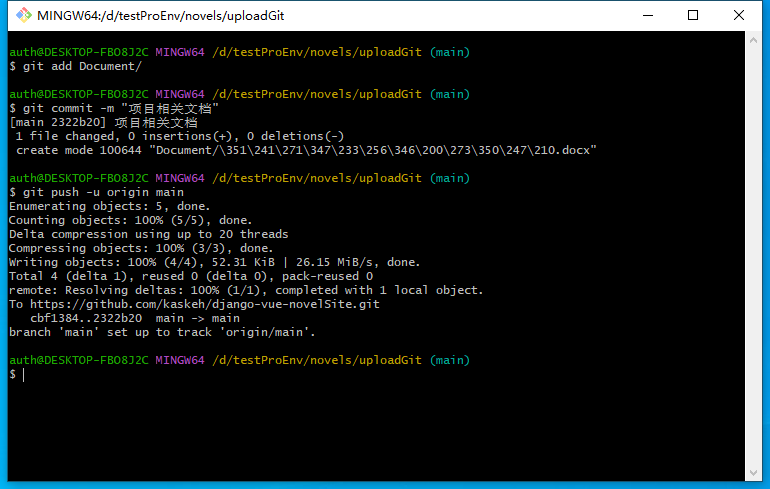


### 3.2.4 账号密码修改

**3.2.5 用户退出账号**

用户点击退出，则跳转到登录页面，同时前端一并把localstroage，cookies的数据前部清除了

# 项目版本控制系统(VCS)



# 项目架设