**尚硅谷毕业设计之论坛系统**

## 前言

论坛系统：

随着网络的兴起，人们的交流越来越方便，于是就出现了论坛，使得人们足不出户就可以互相聊天，也可以通过网络了解国家大事，还可以针对某一事件发表个人看法，大家可以畅所欲言。在论坛上，人们可以对某一领域提出自己遇到的问题，即发表某一主题，随后，论坛上的其他人会根据自己的学识、经验发表意见或提出问题的方法。

论坛主要是一个供用户交流的平台，为广大用户提供交流经验、探讨问题的网上社区。因此，论坛系统最基本的功能首先是发表主题，其次是其他人员根据主题发表自己的看法。此外，为了记录主题的发表者和主题的回复者信息，系统提供了第三方登录功能，我们这次做的是一个码匠社区，所以我们采用了GitHub第三方登录。只有登录成功的用户才能发表和回复主题，没有登录的用户只能浏览网站信息。

## 环境准备：

### 1.1、开发工具:

建议使用IntelliJ IDEA 2019.2.3 x64，其他版本也可以

### 1.2、Jdk:

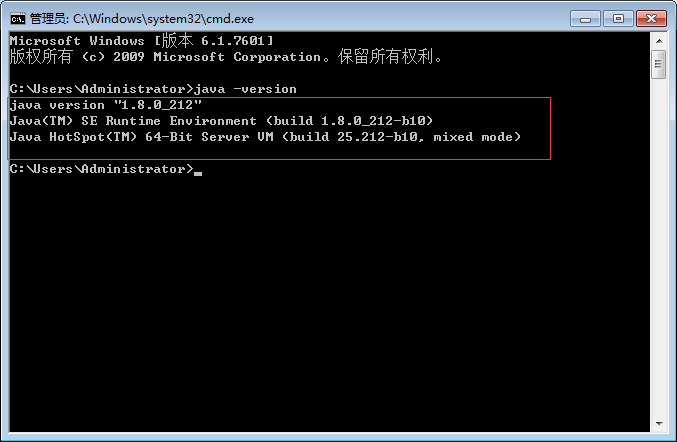
版本：java1.8

验证安装是否成功

1、打开cmd窗口

2、输入java –version

如图所示：



3、出现上图中红线部分即安装jdk成功

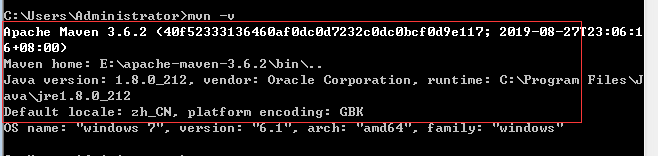
### 1.3、Maven：

验证安装是否成功

1、打开cmd窗口

2、输入mvn -v

如图所示



3、出现上图中红线部分即安装maven成功

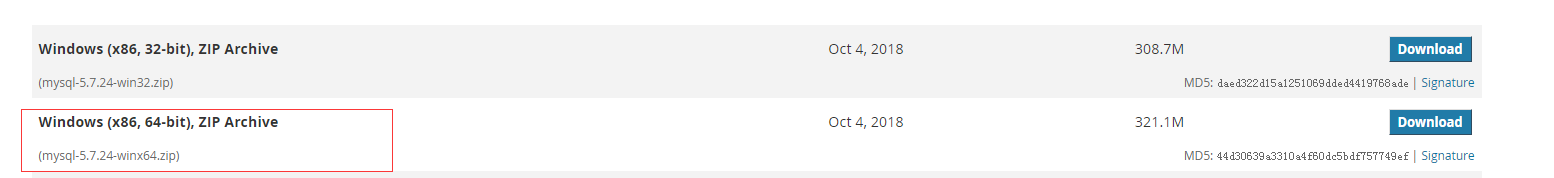
### 1.5、Mysql：

Windows安装mysql服务端

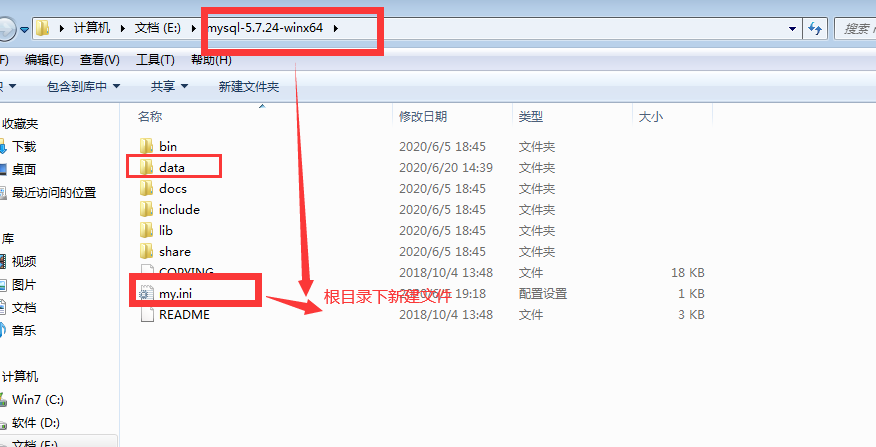
1. 下载5.7.24的解压版

下载地址：<https://downloads.mysql.com/archives/community/>

1. 如图所示



1. 在mysql根目录添加配置文件my.ini；改成自己的安装目录，并且一定要新建data目录；



my.ini文件内容

|  |
| --- |
| [mysql]  # 设置mysql客户端默认字符集  default-character-set=utf8  [mysqld]  #设置3306端口  port = 3306  # 设置mysql的安装目录  basedir=E:\mysql-5.7.24-winx64  # 设置mysql数据库的数据的存放目录  datadir=E:\mysql-5.7.24-winx64\data  # 允许最大连接数  max\_connections=200  # 服务端使用的字符集默认为8比特编码的latin1字符集  character-set-server=utf8  # 创建新表时将使用的默认存储引擎  default-storage-engine=INNODB |

1. 安装并启动服务

|  |
| --- |
| cmd用管理员身份进入mysql的bin目录 1.初始化数据库（第一次一定要做，后续如果卸载后重新安装则跳过这一步） mysqld --initialize 2.安装服务 mysqld install MySQL57 3.启动服务 net start MySQL57 |

1. 配置数据库

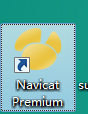
|  |
| --- |
| 配置数据库做两件事情：修改root默认密码；设置root可以远程访问； 1.登陆：mysql -uroot -p 初始密码在data文件夹中以.err结尾的文件中，如：A temporary password is generated for root@localhost: **wEsWp:pvB8Py** 2.修改默认密码：alter user root@localhost identified by '123456'; 3.切换数据库：use mysql; 3.设置用户可以远程访问（%表示不限制访问ip）：update user set host = '%' where user = 'root'; 4.确认设置（或重启服务器让设置生效）：FLUSH PRIVILEGES; |

以上变完成了mysql服务端的安装

### 1.6、数据库连接工具

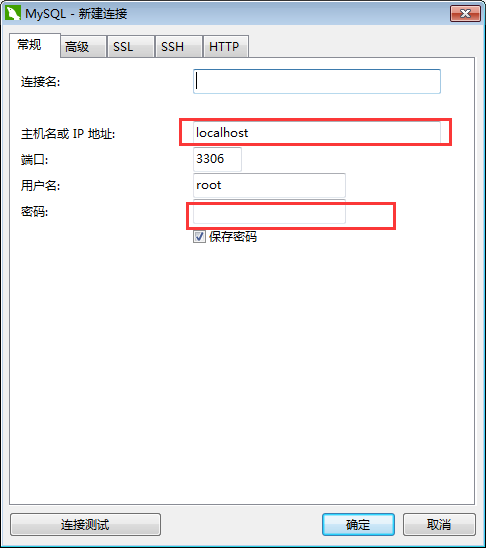
推荐使用Navicat

图标如图所示



安装完成连接mysql

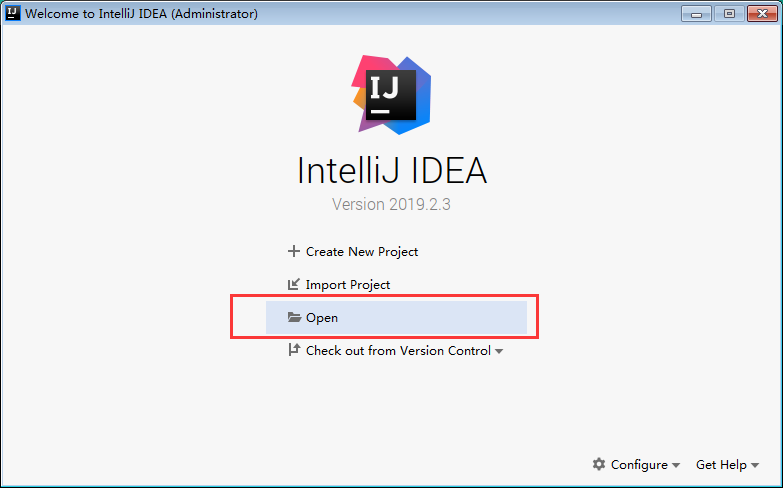
1、输入连接ip地址，用户名和密码，如果连接成功，则表示mysql安装没有问题



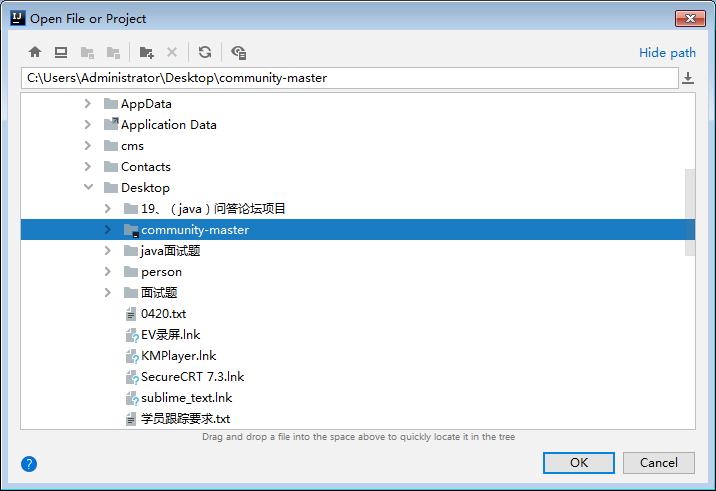
## 项目运行

### 2.1、Idea打开代码

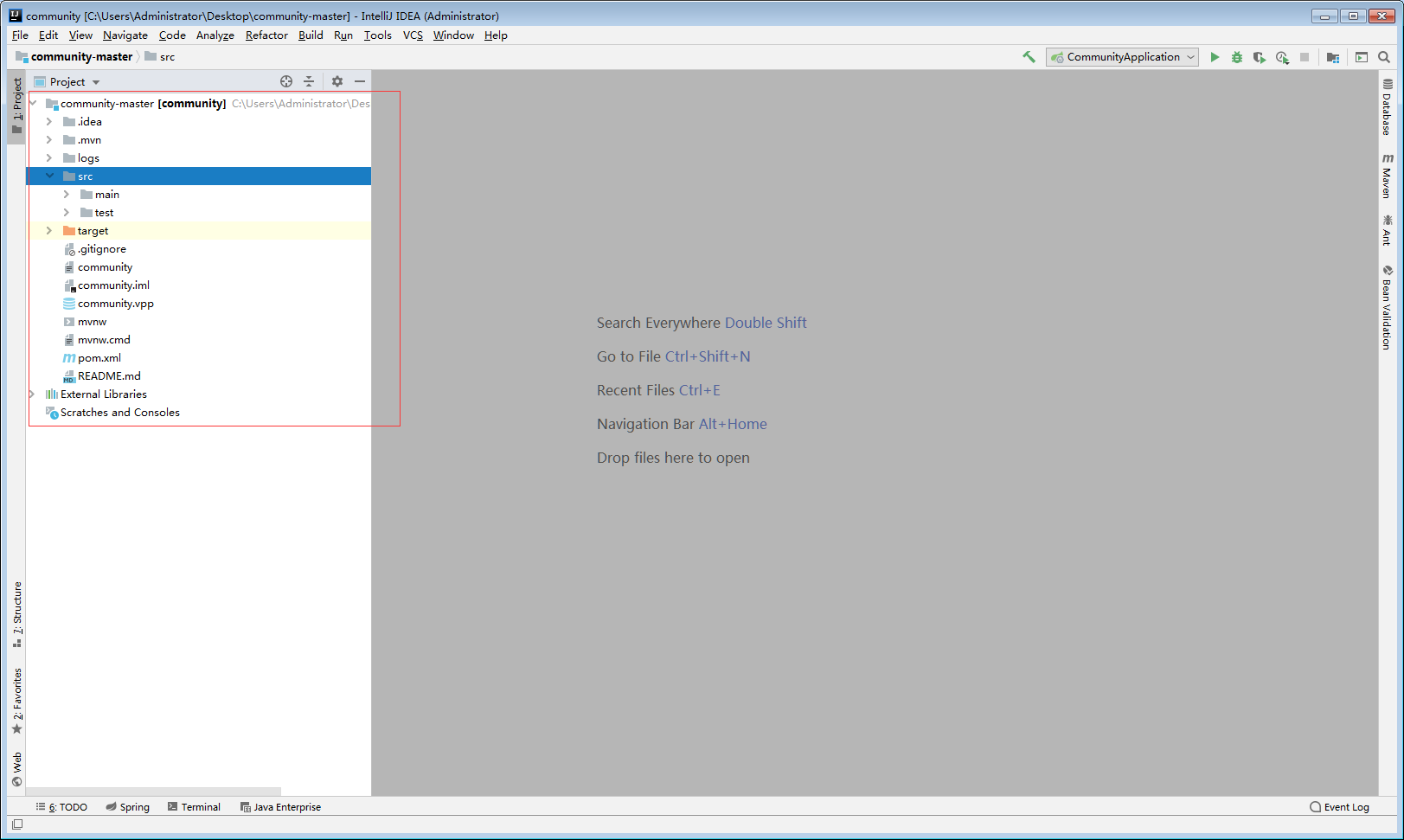
1. 打开idea，选择open，如图所示



1. 选中项目文件夹，点击ok按钮如图所示

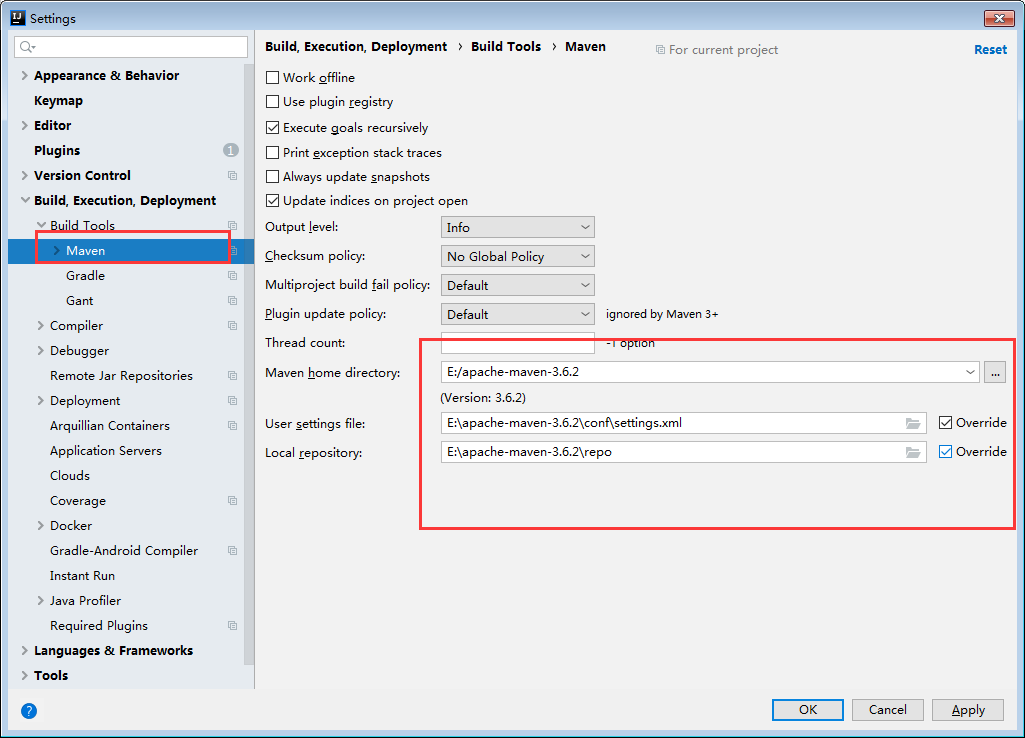


1. 进入idea界面，如图所示



### 2.2、maven配置

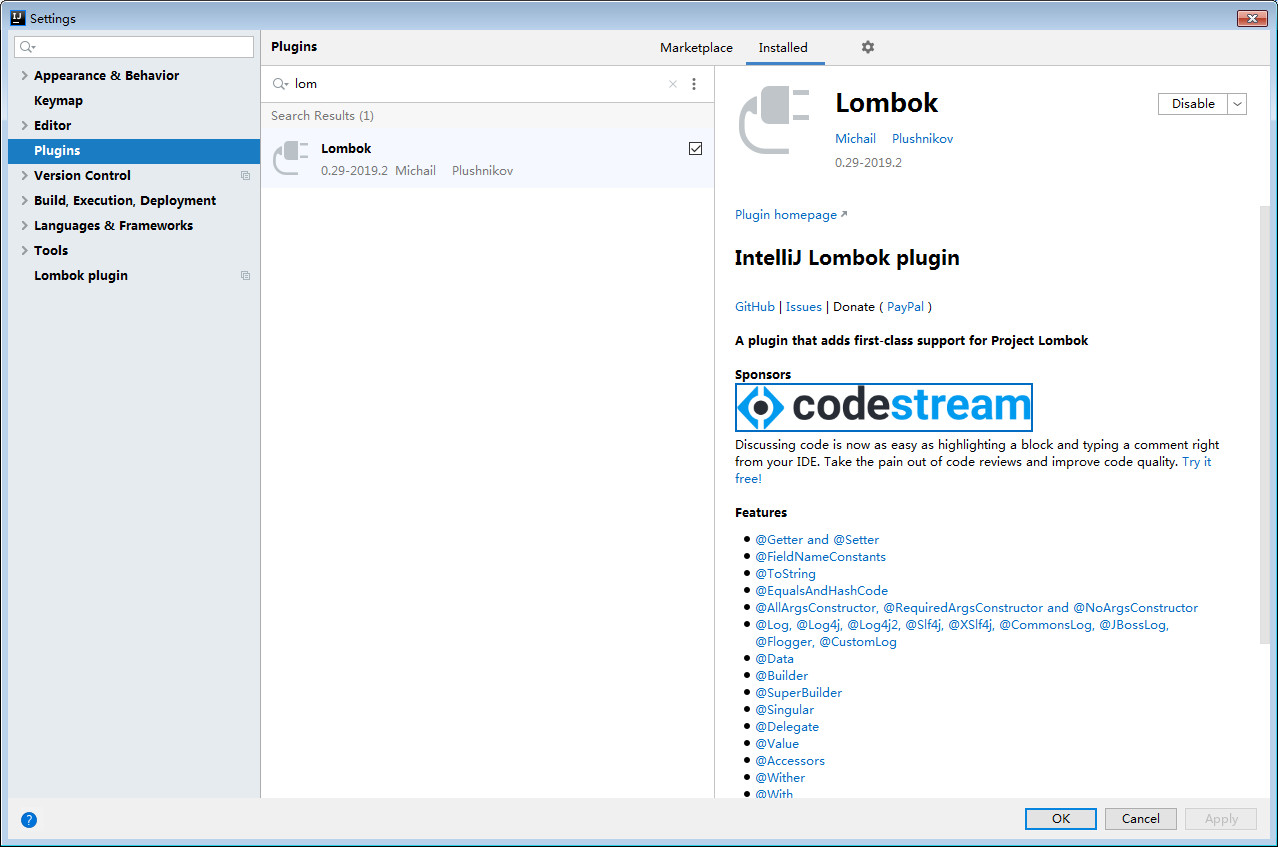
打开idea->file->settings配置，选中maven安装目录，点击确认，如图所示



### 2.3、Lombok插件安装

一、直接安装：

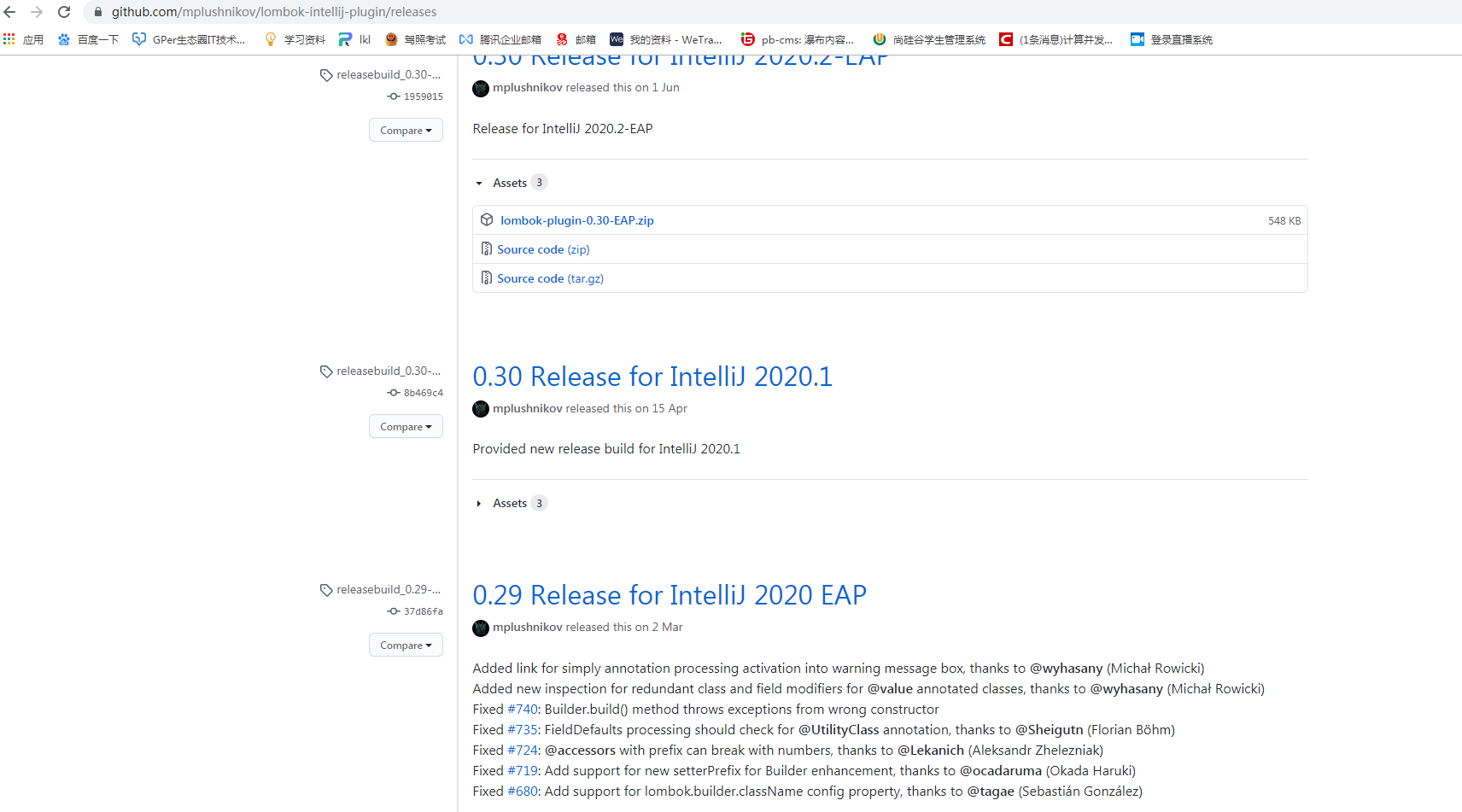
打开idea->file->plugins配置，搜索lombok，点击确认，如图所示



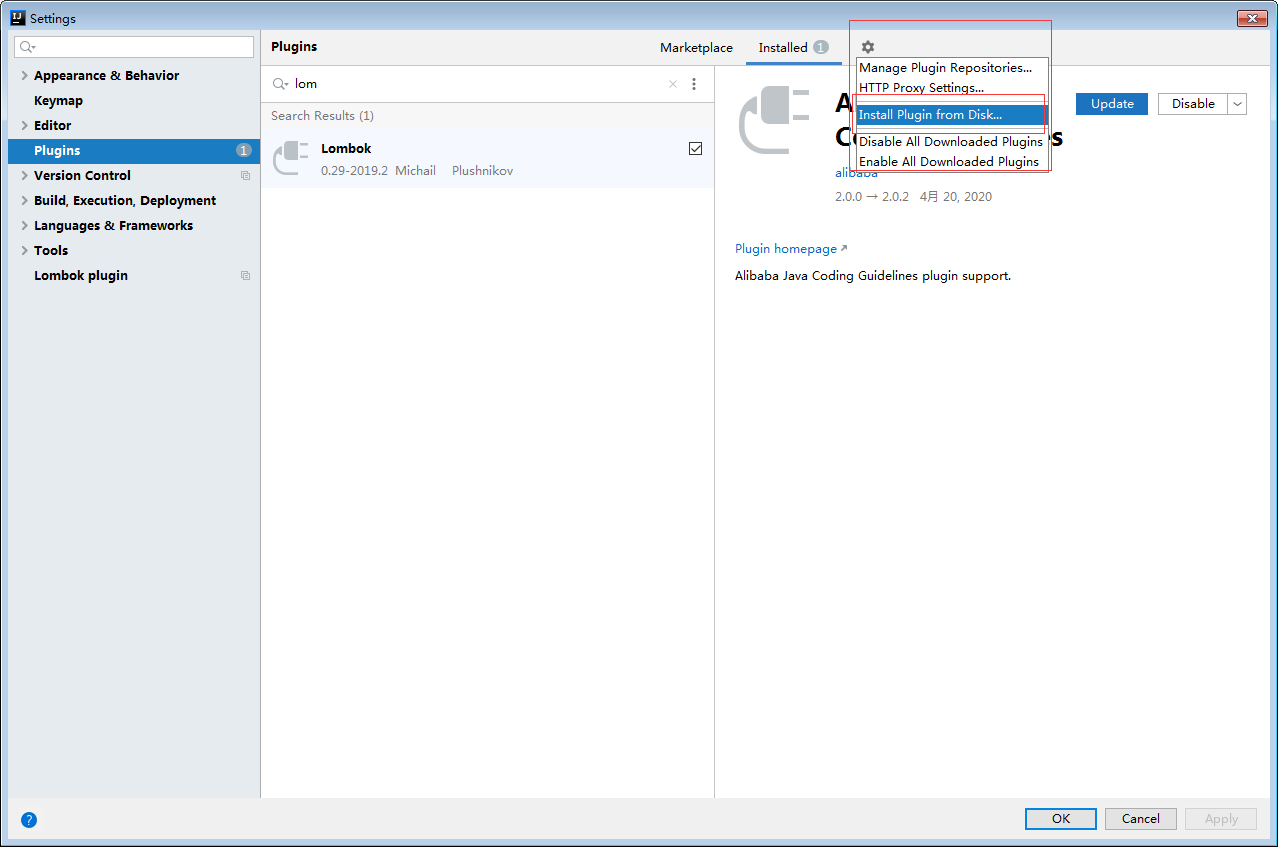
1. 离线安装

下载地址<https://github.com/mplushnikov/lombok-intellij-plugin/releases>

下载对应idea版本



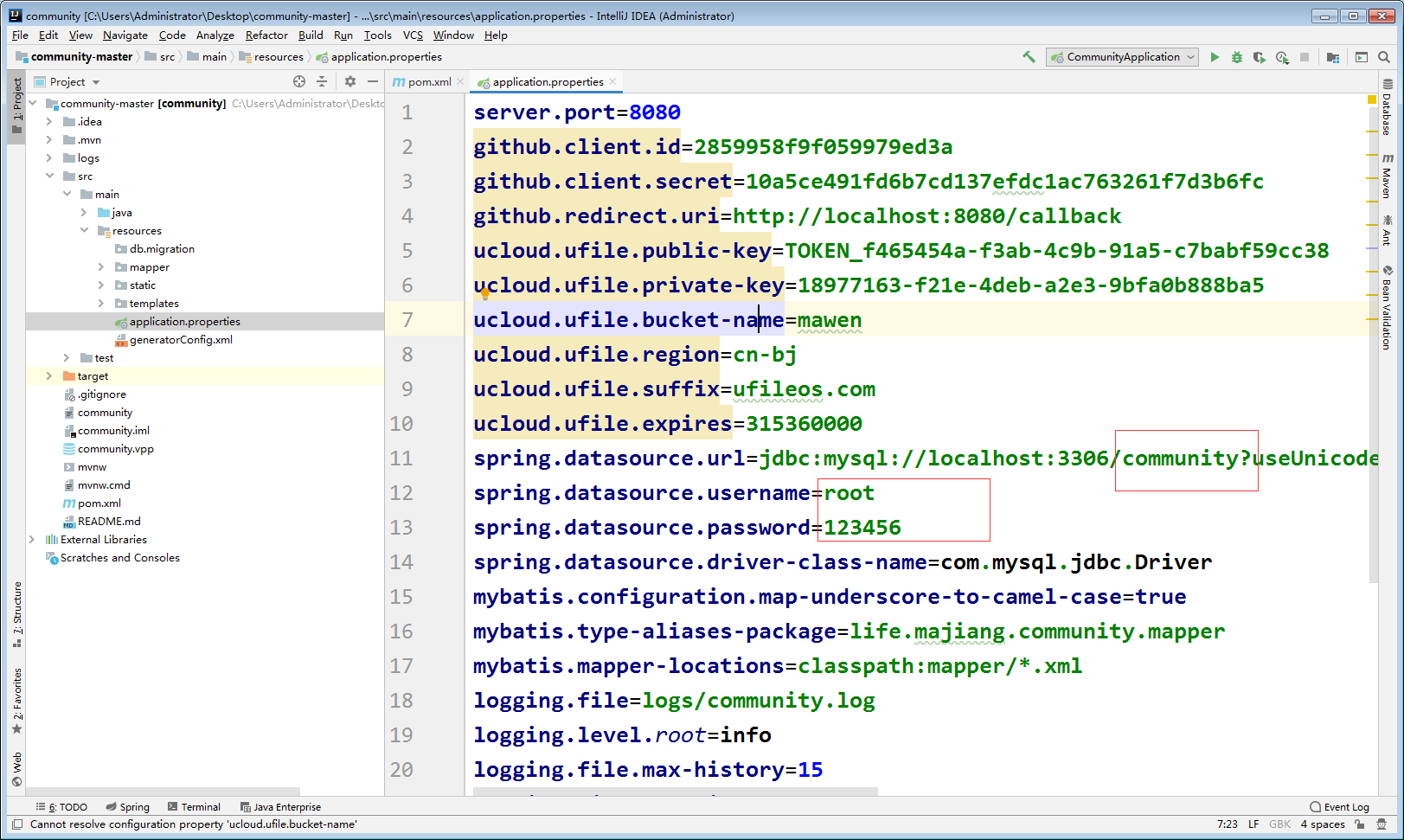
选中下载的插件



安装成功之后重启

### 2.3、修改mysql配置

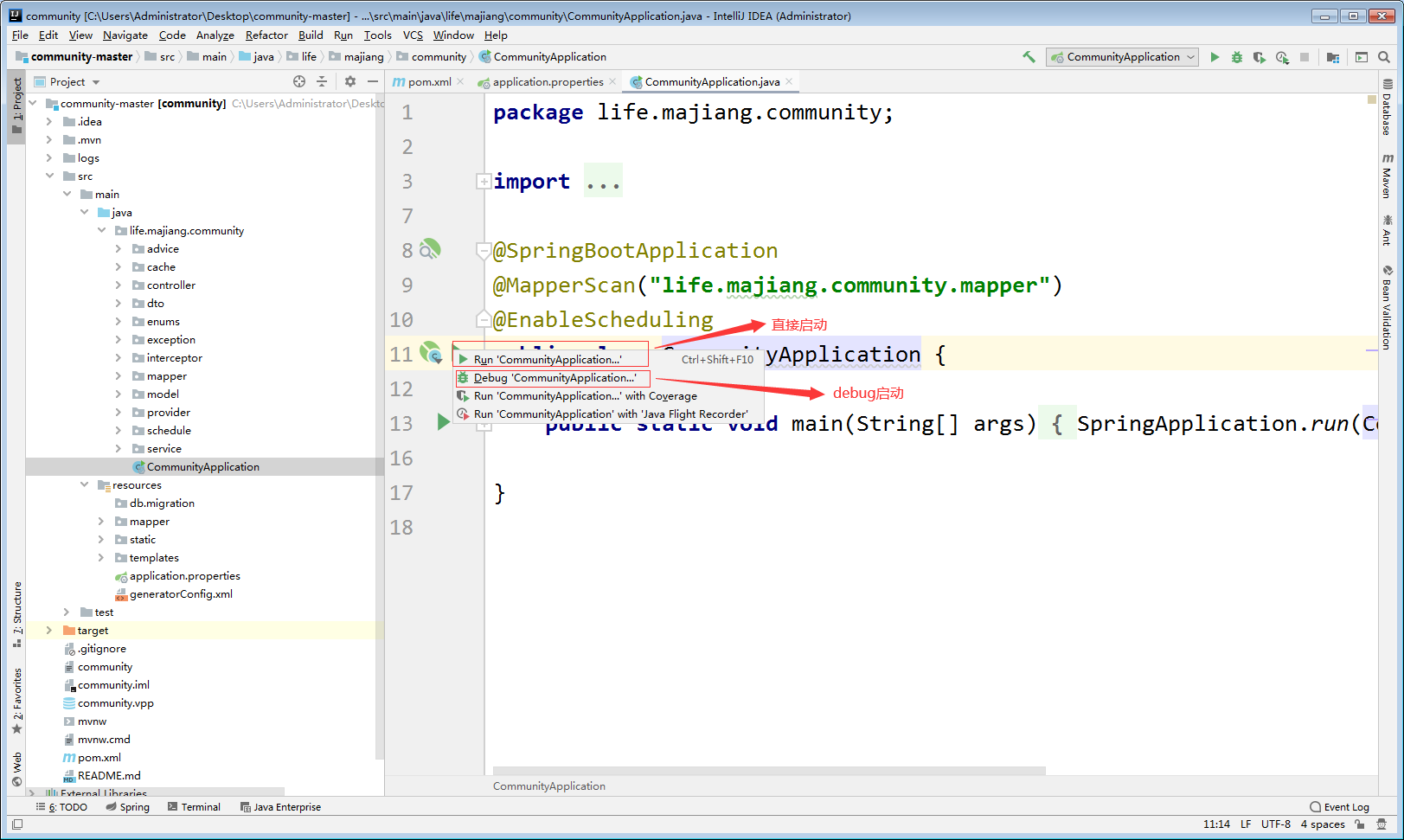
打开application.properties配置文件，修改数据库连接地址，用户名和密码，保存即可



### 2.4、项目启动

两种启动方式：

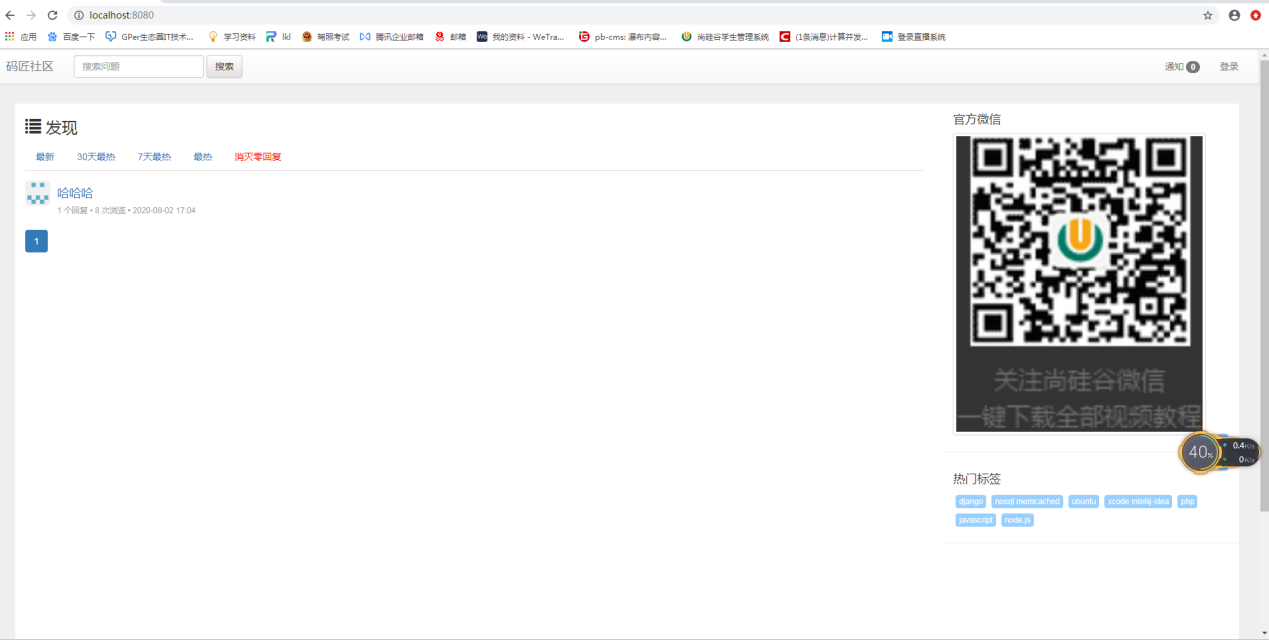
1. 直接启动
2. Debug启动，建议使用debug启动



### 2.5、启动成功

效果如图所示

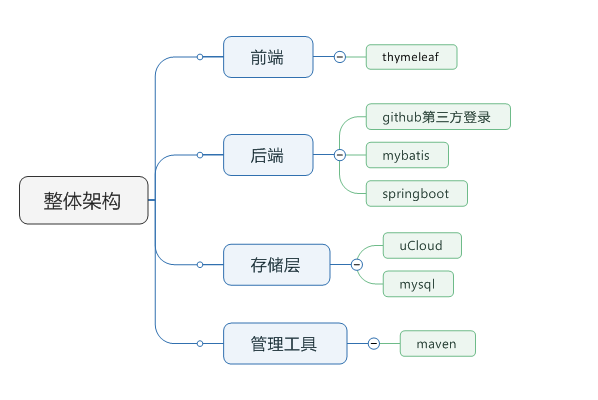
前台访问地址：[http://localhost:8080](http://localhost:8080/CMS_war_exploded/index.htm)



## 三、 项目架构

### 3.1、整体架构图：

如图所示，项目前端采用thymeleaf来实现，后端服务采用springboot+mybatis框架，登录使用的是第三方github登录，存储层使用高性能的mysql，图片存储到ucloud云存储中，使用maven来管理jar包和项目构建。



### 3.2、后端架构：

**Springboot:**

Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化Spring应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。Boot致力于在蓬勃发展的快速应用开发领域成为领导者。

Spring是一个非常伟大的框架，开源，发展迅速。优秀的设计必然会划分、解耦。所以，spring有很多子项目，比如core、context、bean、mvc等。对于初学spring的人来说，想要入手就开发需要拷贝一连串的dependency而不知道这个是干嘛，不知道是不是少了依赖。刚接触spring的时候搜索教程发现各有不同，于是复制了一个又一个代码却不能自己设置，根本原因是不了解各个依赖的包。

Spring-Boot 正是为了解决繁复的代码配置而产生的。Spring-Boot 也是基于java-base 开发的代码，不用xml文件配置，所有代码都由java来完成。

**mybatis**  
　　mybatis是对jdbc的封装，它让数据库底层操作变的透明。mybatis的操作都是围绕一个sqlSessionFactory实例展开的。mybatis通过配置文件关联到各实体类的Mapper文件，Mapper文件中配置了每个类对数据库所需进行的sql语句映射。在每次与数据库交互时，通过sqlSessionFactory拿到一个sqlSession，再执行sql命令。

Ucloud

**对象存储 UFile**：对象存储（UFile）是为互联网应用提供非结构化文件云存储的服务；相对于传统硬盘存储，UFile云存储具有存储无上限、支持高并发访问、成本更低等优势；解决业务架构的文件存储问题，有效降低海量文件的存储成本，支持热点数据的高并发访问，提升终端用户访问体验。

官网<https://www.ucloud.cn/site/product/ufile.html>

#### 3.2.1项目整合

我们把配置文件都放到了一个application.properties文件中，详情如下：

|  |
| --- |
| server.port=8080  #github第三方登录  github.client.id=3be615b004dd7195a8ef  github.client.secret=089e5460b44b1eff7e3074cabfb625417543de4b  github.redirect.uri=http://localhost:8080/callback  #ucloud上传图片  ucloud.ufile.public-key=TOKEN\_f465454a-f3ab-4c9b-91a5-c7babf59cc38  ucloud.ufile.private-key=18977163-f21e-4deb-a2e3-9bfa0b888ba5  ucloud.ufile.bucket-name=mawen  ucloud.ufile.region=cn-bj  ucloud.ufile.suffix=ufileos.com  ucloud.ufile.expires=315360000  #数据库连接  spring.datasource.url=jdbc:mysql://localhost:3306/community?useUnicode=true&characterEncoding=utf8&autoReconnect=true&useSSL=false&serverTimezone=UTC  spring.datasource.username=root  spring.datasource.password=123456  spring.datasource.driver-class-name=com.mysql.jdbc.Driver  #将带有下划线的表字段映射为驼峰格式的实体类属性。  mybatis.configuration.map-underscore-to-camel-case=true  mybatis.mapper-locations=classpath:mapper/\*.xml  logging.file=logs/community.log  logging.level.root=info  #最多保留15个归档文件  logging.file.max-history=15  #日志文件大小  logging.file.max-size=100MB  #session超时时间  server.servlet.session.timeout=15552000  #单个文件大小  spring.servlet.multipart.max-file-size=20MB  #上传总文件大小  spring.servlet.multipart.max-request-size=20MB |

### 3.3、前端架构：

前端采用的是thymeleaf。Thymeleaf 是一个跟 Velocity、FreeMarker 类似的模板引擎，它可以完全替代 JSP 。它有如下特点

Thymeleaf 在有网络和无网络的环境下皆可运行，即它可以让美工在浏览器查看页面的静态效果，也可以让程序员在服务器查看带数据的动态页面效果。这是由于它支持 html 原型。

Thymeleaf 开箱即用的特性。直接以html显示，前后端可以很好的分离。

Thymeleaf可以快速的实现表单绑定、属性编辑器、国际化等功能。

### 3.4、管理工具：

采用maven来进行项目构建，管理，jar包下载等功能。

下面是引入的jar包管理

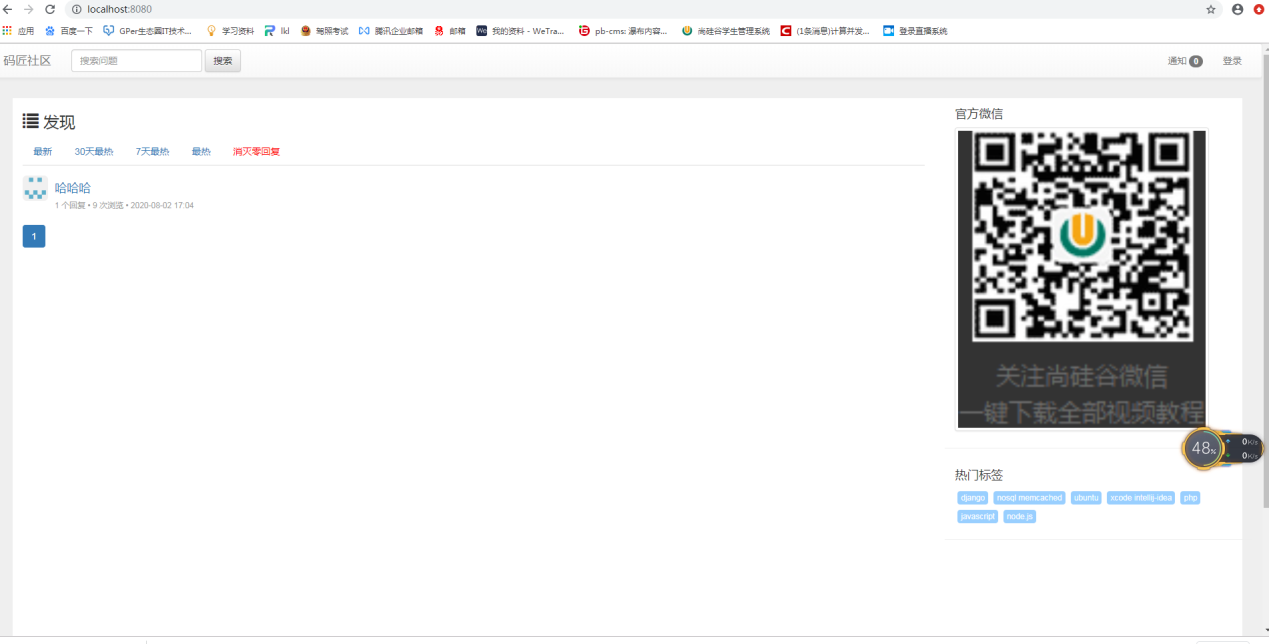
|  |
| --- |
| <**dependencies**>  <**dependency**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-web</**artifactId**>  </**dependency**>   <**dependency**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-thymeleaf</**artifactId**>  </**dependency**>  <**dependency**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-jdbc</**artifactId**>  </**dependency**>  <**dependency**>  <**groupId**>org.mybatis.spring.boot</**groupId**>  <**artifactId**>mybatis-spring-boot-starter</**artifactId**>  <**version**>2.0.1</**version**>  </**dependency**>  <**dependency**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-test</**artifactId**>  <**scope**>test</**scope**>  </**dependency**>  <**dependency**>  <**groupId**>com.squareup.okhttp3</**groupId**>  <**artifactId**>okhttp</**artifactId**>  <**version**>3.14.1</**version**>  </**dependency**>  <**dependency**>  <**groupId**>com.alibaba</**groupId**>  <**artifactId**>fastjson</**artifactId**>  <**version**>1.2.57</**version**>  </**dependency**>  <**dependency**>  <**groupId**>mysql</**groupId**>  <**artifactId**>mysql-connector-java</**artifactId**>  <**version**>5.1.45</**version**>  </**dependency**>   <**dependency**>  <**groupId**>com.h2database</**groupId**>  <**artifactId**>h2</**artifactId**>  <**version**>1.4.199</**version**>  </**dependency**>  <**dependency**>  <**groupId**>org.projectlombok</**groupId**>  <**artifactId**>lombok</**artifactId**>  <**version**>1.18.6</**version**>  <**scope**>provided</**scope**>  </**dependency**>  <**dependency**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-devtools</**artifactId**>  <**optional**>true</**optional**>  </**dependency**>  <**dependency**>  <**groupId**>org.apache.commons</**groupId**>  <**artifactId**>commons-lang3</**artifactId**>  <**version**>3.9</**version**>  </**dependency**>  <**dependency**>  <**groupId**>cn.ucloud.ufile</**groupId**>  <**artifactId**>ufile-client-java</**artifactId**>  <**version**>2.1.1</**version**>  </**dependency**> </**dependencies**> |

## 项目功能使用介绍

### 4.1、前台主题模块

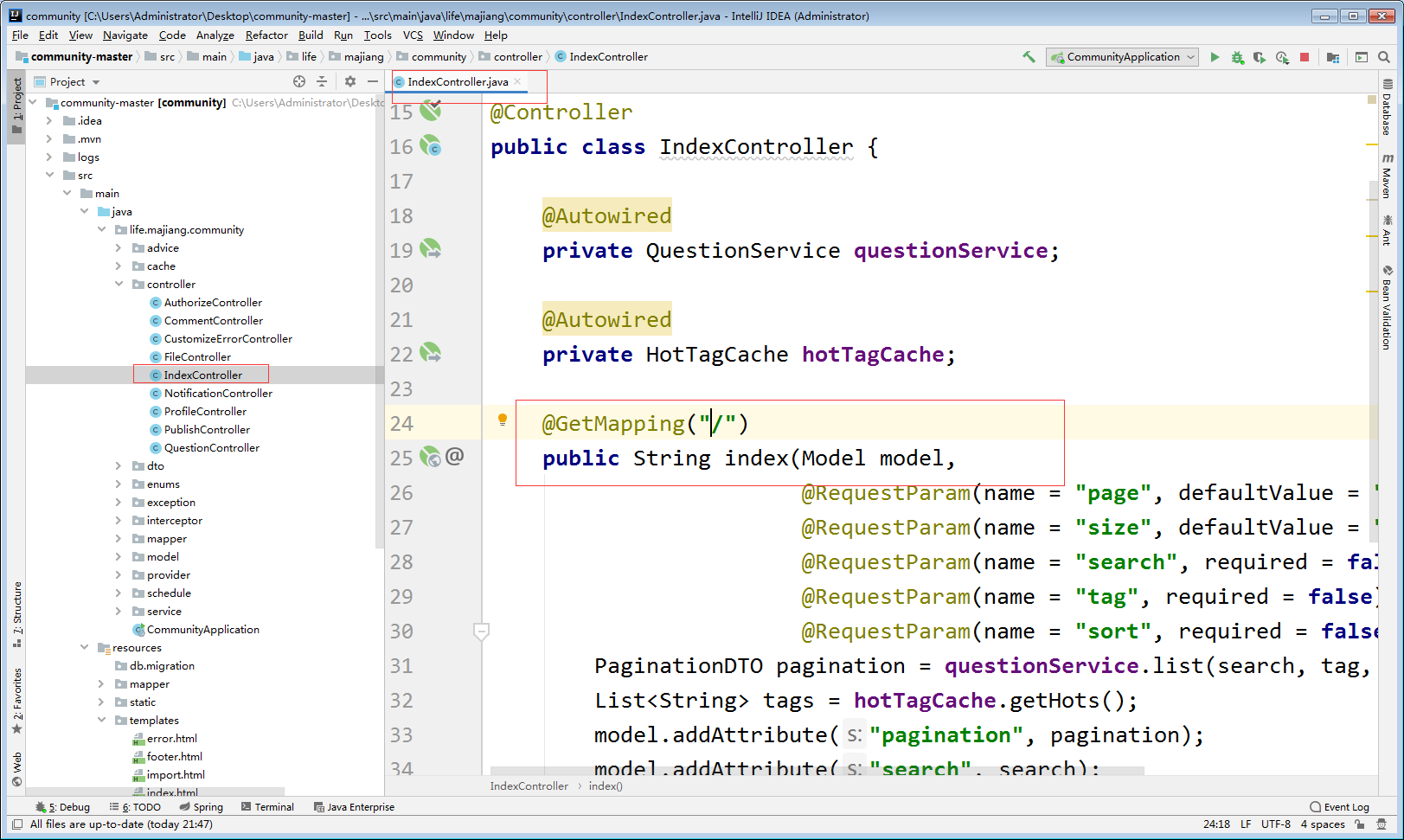
#### 4.1.1功能演示

前台可以展示最新主题、三十天最新主题、7天最新主题等，最上方可以提供主题搜索，如图所示：



#### 4.1.2代码实现

1、前端请求，首先进入 controller，入下图所示



### 4.2、登录模块

#### 4.2.1、功能演示

Github授权登录流程

1. 登录到github注册应用授权。

2、转跳到 GitHub 用户授权页面， client\_id 必须传

最终拼成的URL如下:<https://github.com/login/oauth/authorize?client_id=client_id>

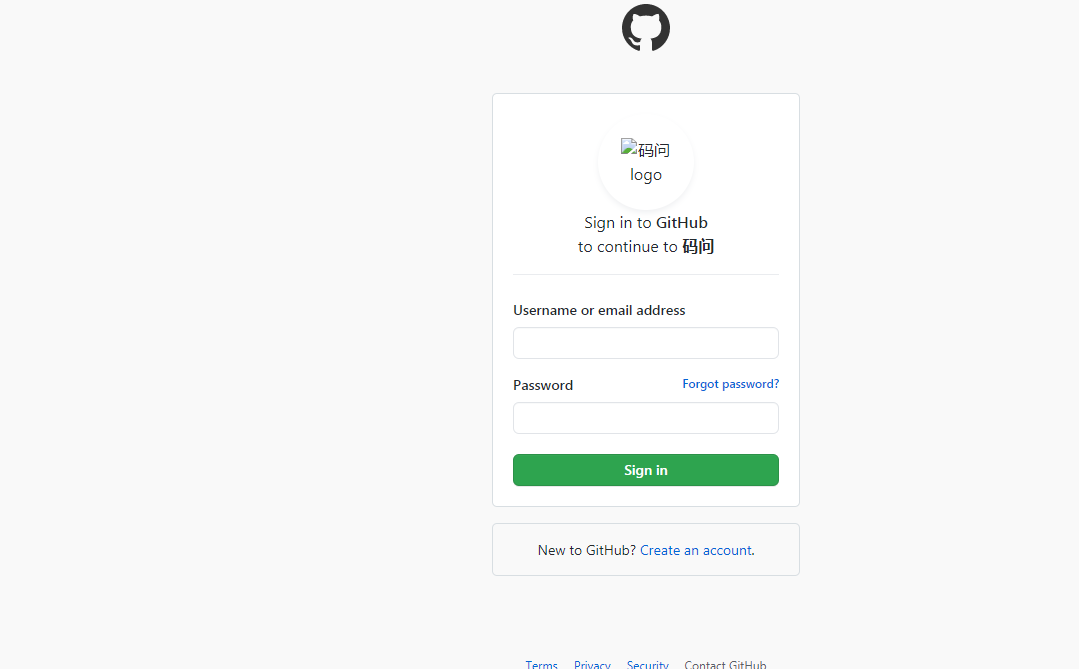
3、当用户同意授权后，链接地址就会转跳到 我们配置页面内的 Authorization callback URL 所填写的URL地址，并且会带上一个 code参数，这个参数在后面获取用户token是必须的一个参数。

4、获取到这个code参数后，我会将这个code传到服务器的后台，然后后台调用 https://github.com/login/oauth/access\_token 这个api，传入 client\_id, client\_secret, code 这三个参数，可以获取到一个 access\_token。

5、获取到 access\_token 后， 再调用

https://api.github.com/user?access\_token=access\_token这个API，就可以获取到基本的用户信息。

点击右上角登录按钮，如下图所示：

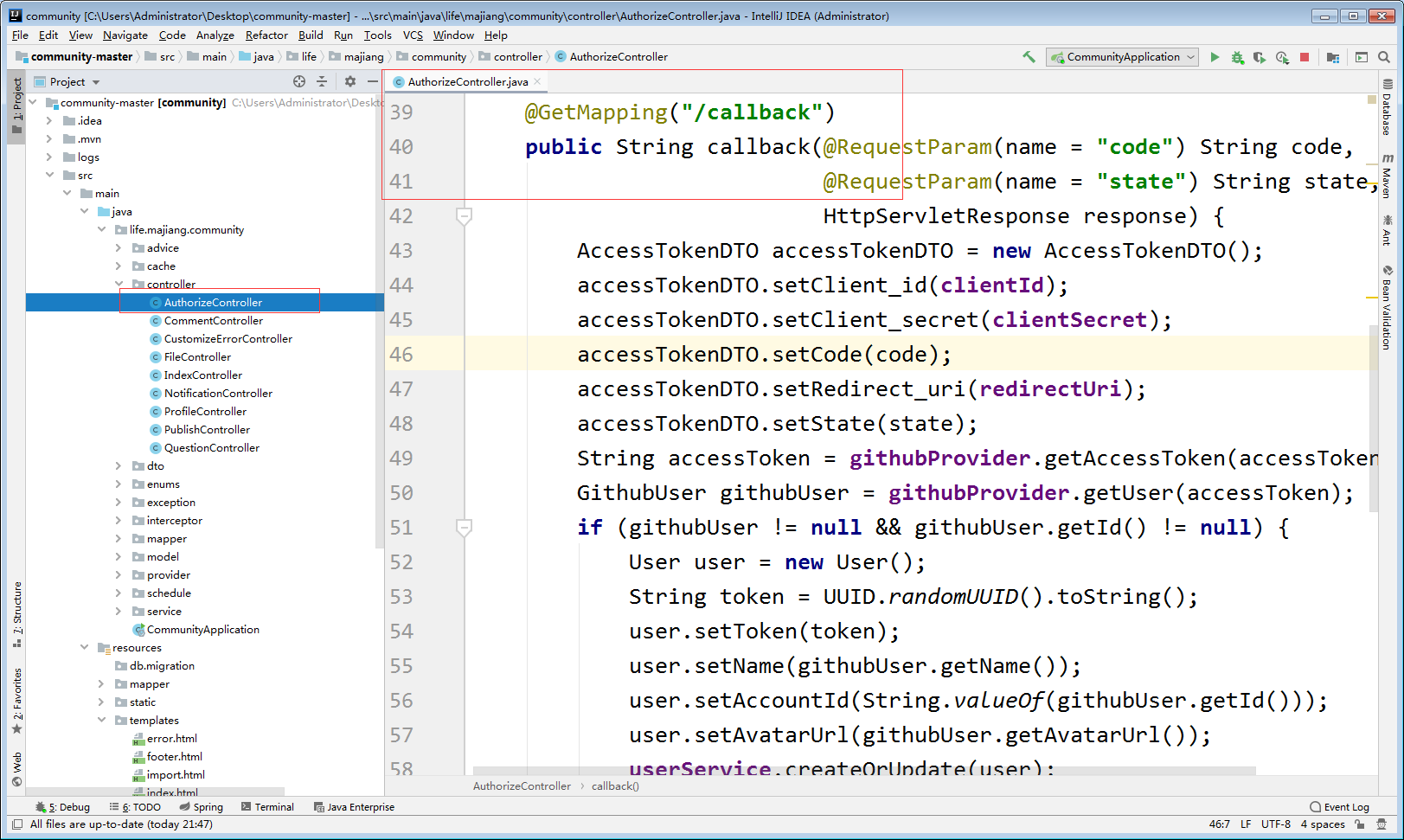


输入github账号和密码即可登录成功

可以使用postman进行测试

#### 4.2.2代码实现

请求controller，如图所示

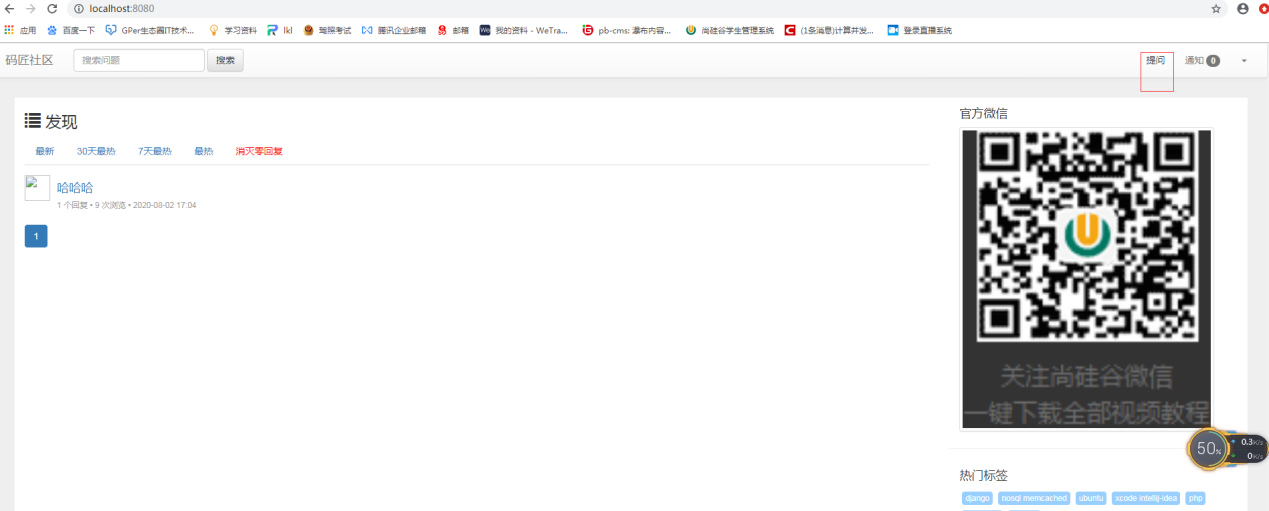


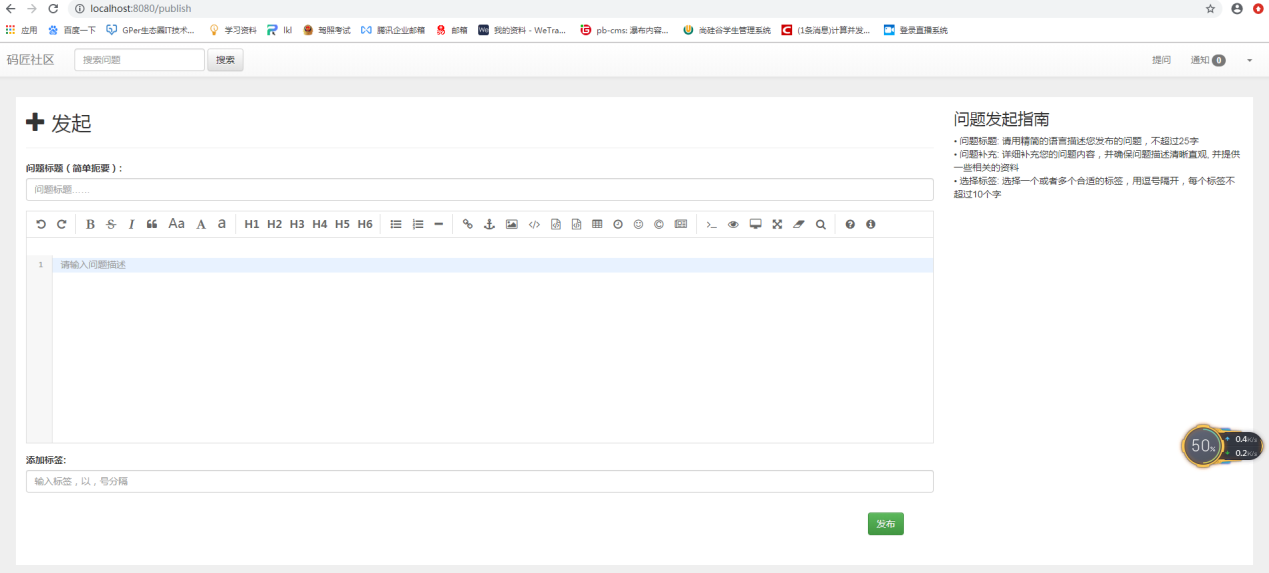
### 4.3、主题发表、评论模块

#### 4.3.1功能演示

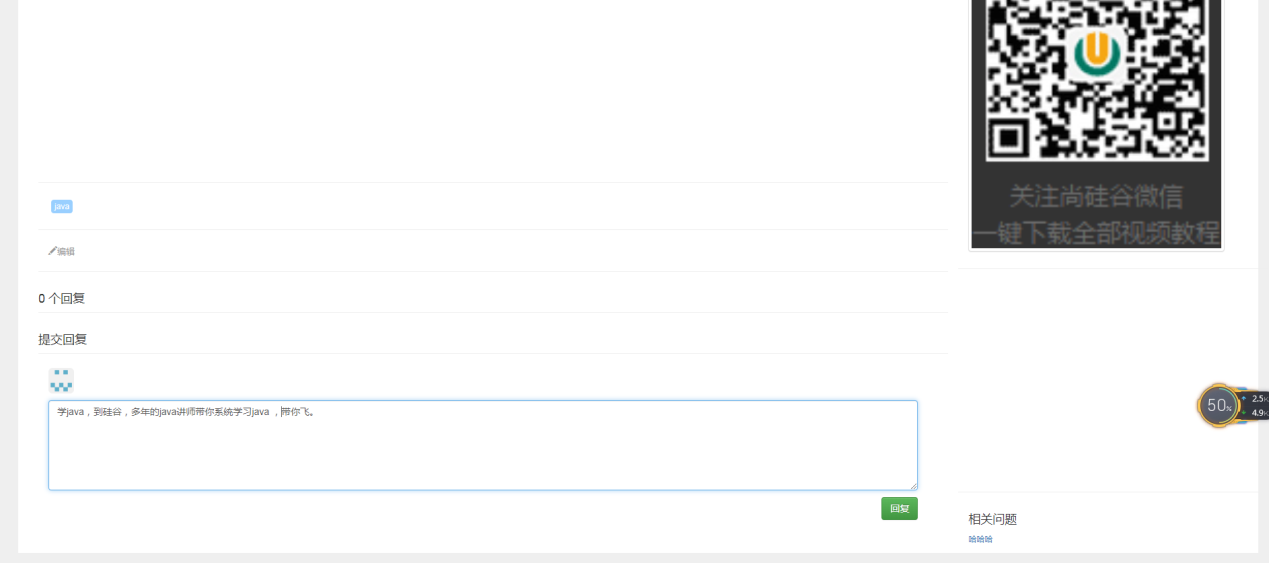
只有登录成功以后才可以发表主题，评论主题如下图所示。

发表主题,点击提问：





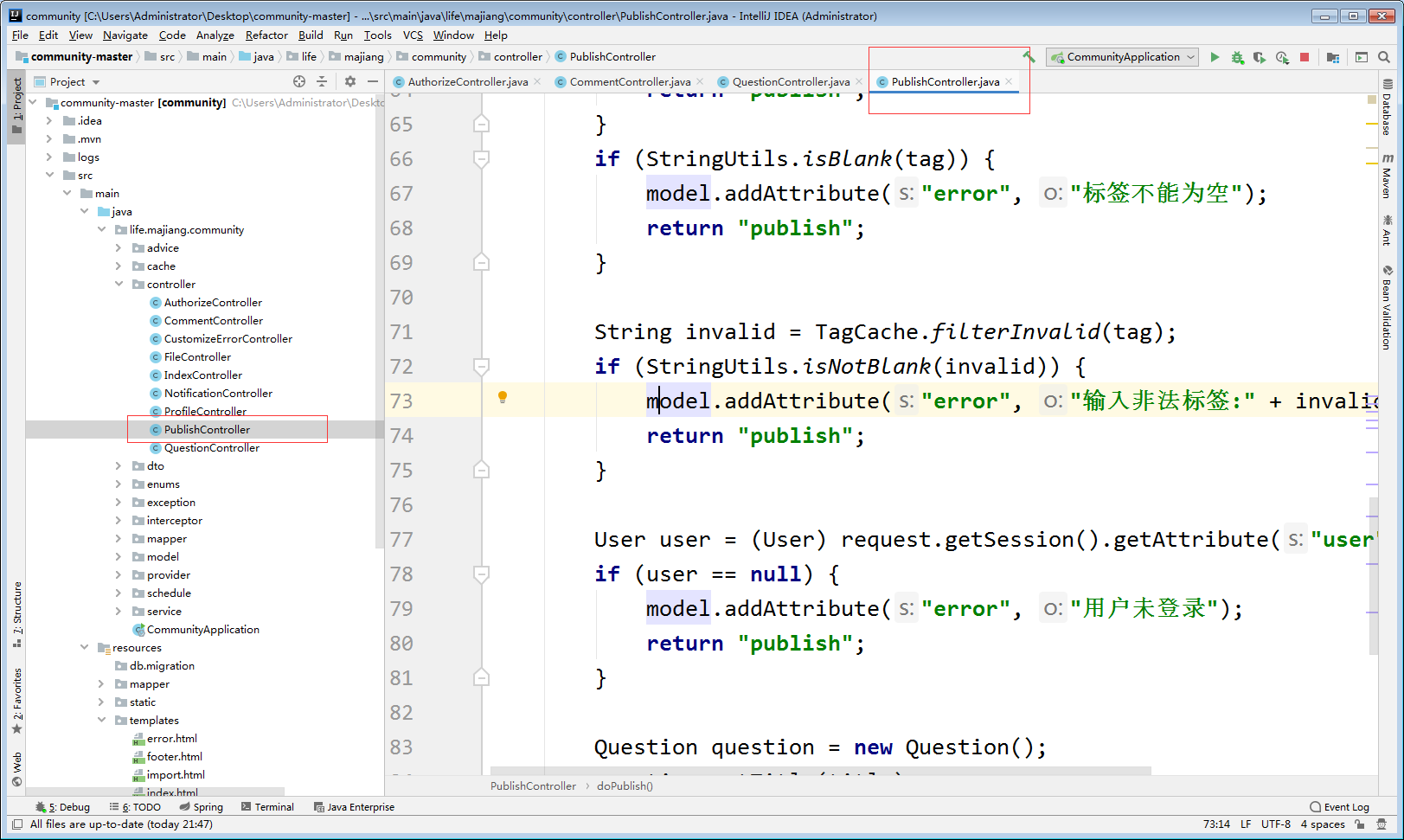
主题评论：



#### 4.3.2代码实现

对主题和主题评论的代码实现如下图所示

1. **提出问题**

****

**二、评论问题**

