Datawhale第32期组队学习

新闻推荐系统项目搭建

# 项目运行环境

操作系统：Windows10

MySQL：8.0.25

Redis：5.0.14

Mongodb：5.0.5

Mini-Conda Python 3.8

前端IDE：WebStorm 2021.1



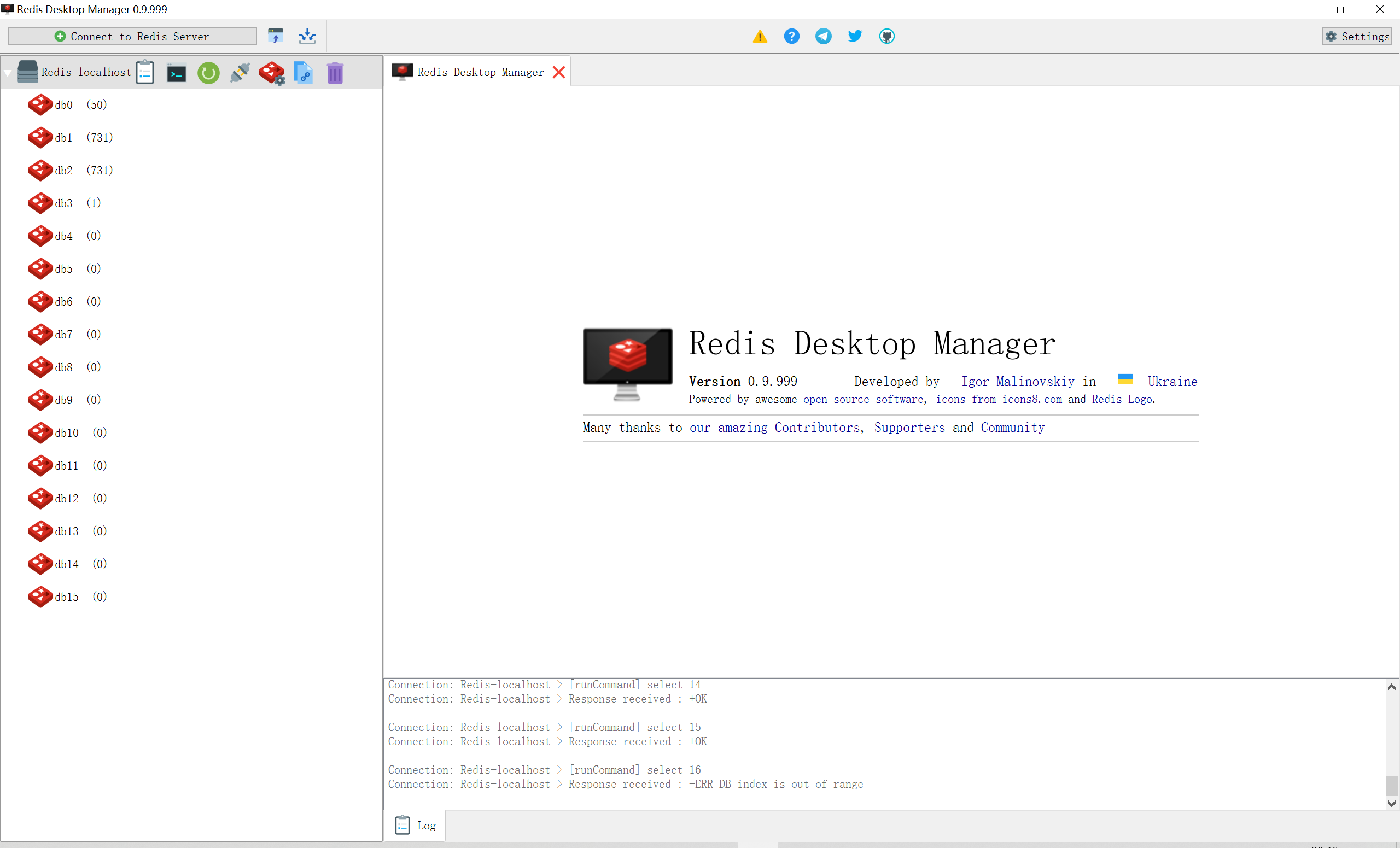
后端IDE：PyCharm Professional 2021.1



访问MySQL和Mongodb的数据库工具：DataGrip 2021.1



访问Redis的工具：redis-desktop-manager-0.9.9.99.exe



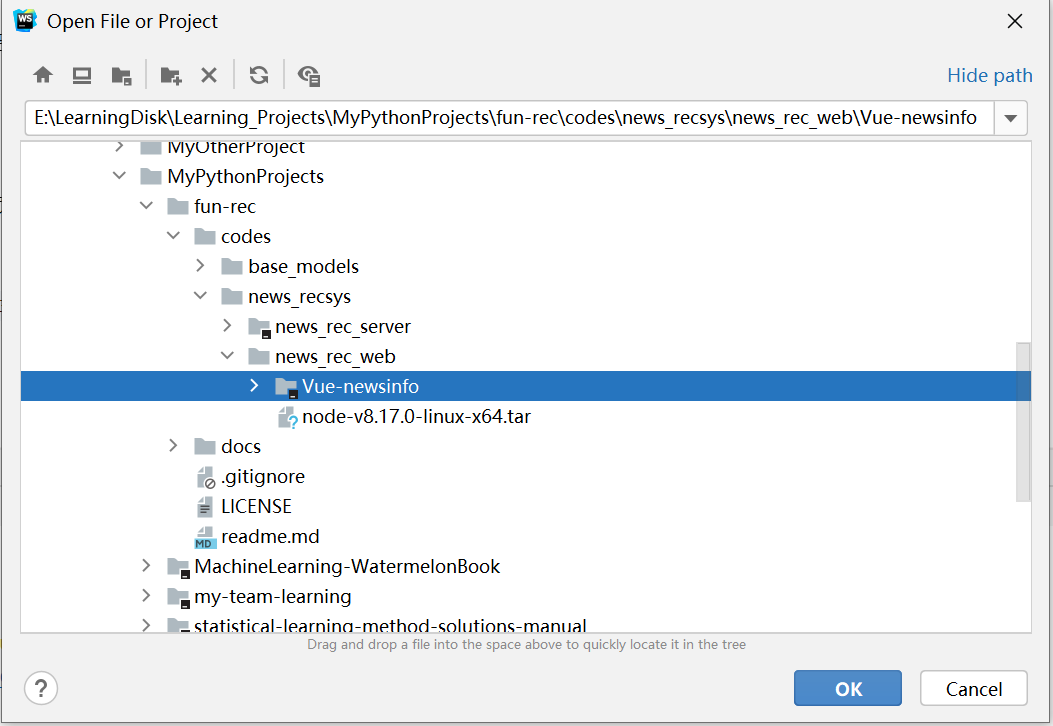
# 项目下载与IDE导入

项目地址：

<https://github.com/datawhalechina/fun-rec>

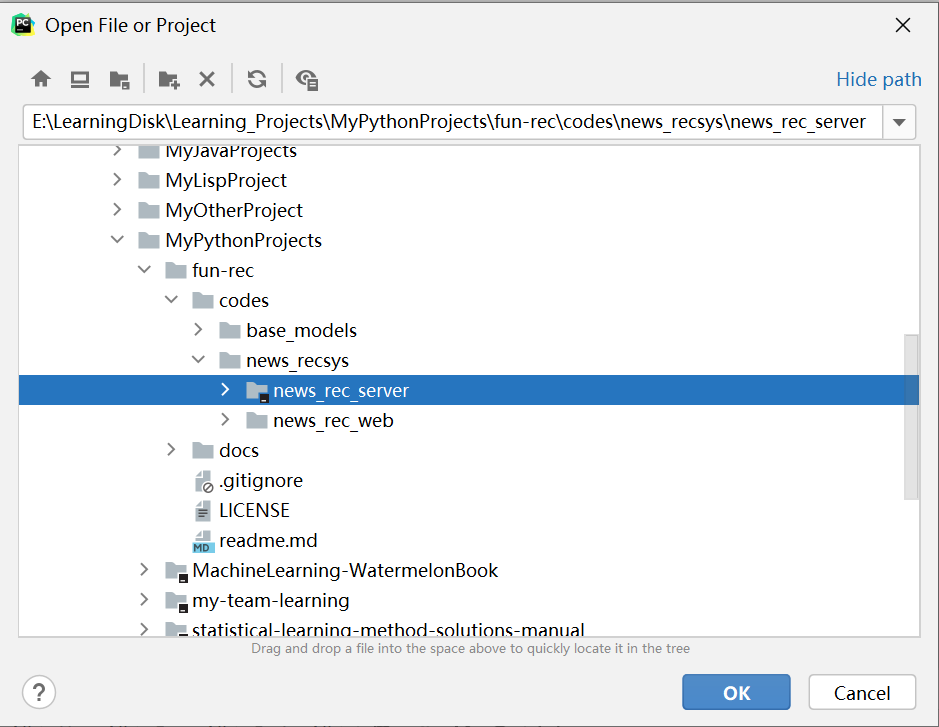
## 前端项目导入

使用WebStrom IDE工具，导入前端项目



## 后端项目导入

使用PyCharm IDE工具，导入后端项目



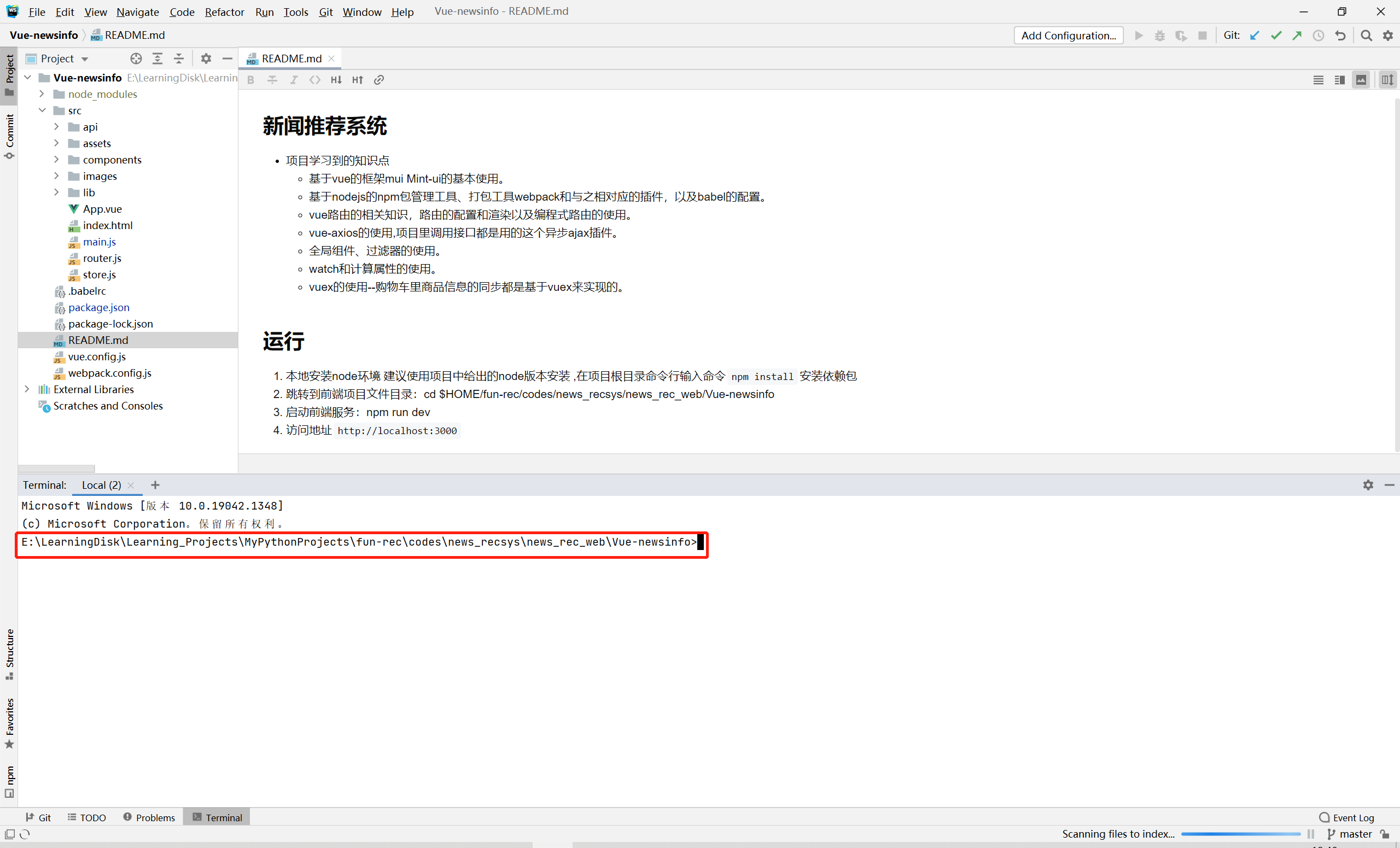
# 数据库安装

# 前端项目运行

## 安装依赖包

在Terminal中执行如下命令：

|  |
| --- |
| cnpm install |



## 修改前端访问IP和端口

打开文件package.json，修改第49行的IP和端口，修改内容如下：

|  |
| --- |
| "scripts": {  "test": "echo \"Error: no test specified\" && exit 1",  "dev": "webpack-dev-server --open --port **8686** --contentBase src --hot --host **127.0.0.1**",  "start": "nodemon src/main.js" }, |

127.0.0.1表示游览器的访问IP（也称为本地IP），8686表示访问端口

## 修改访问后端API接口的IP和端口

打开文件main.js，文件路径：src/main.js，修改第23行的IP和端口，修改内容如下：

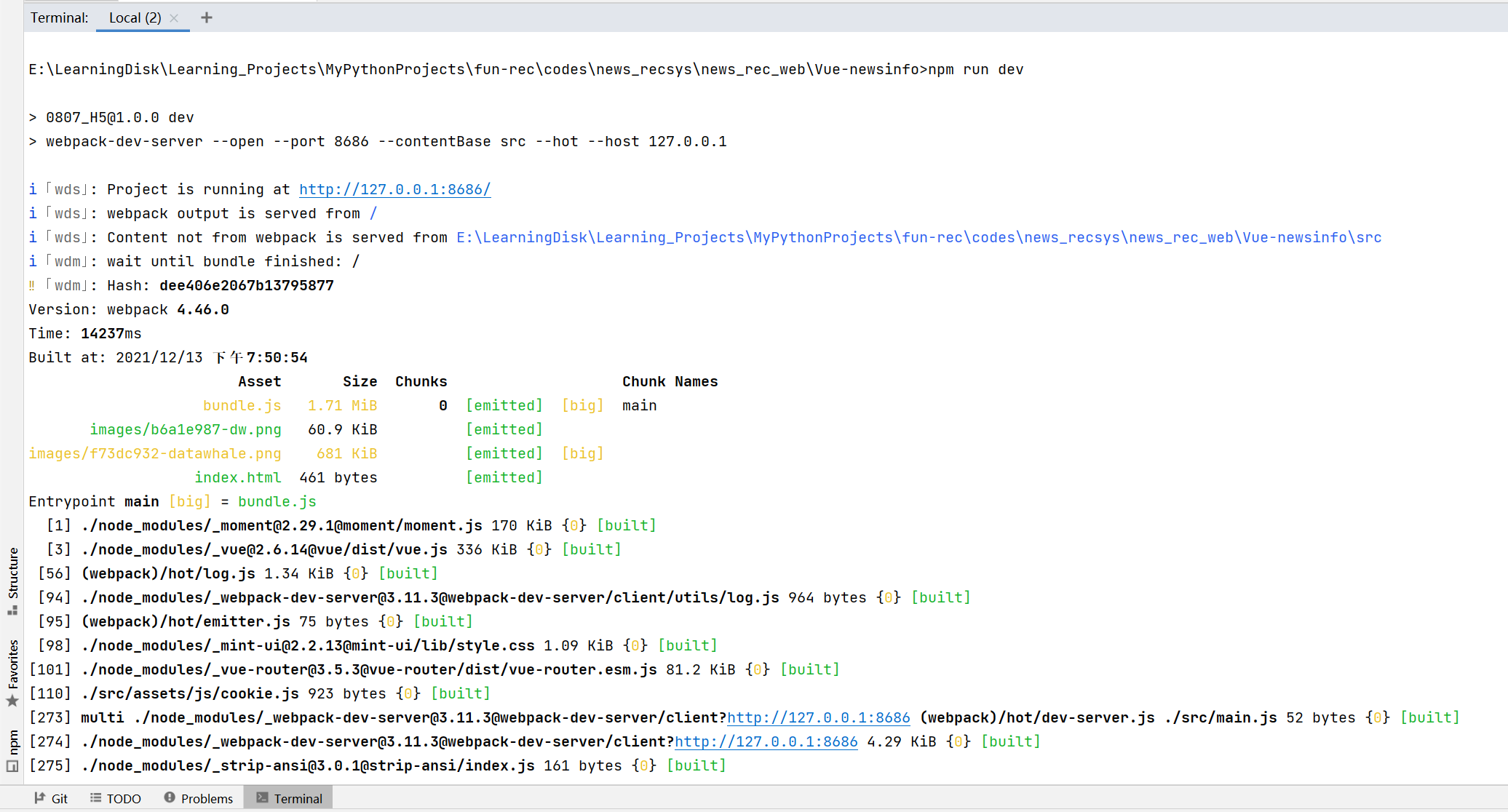
|  |
| --- |
| *// Vue.prototype.$http = axios* Vue.use(VueAxios, axios); *// axios公共基路径，以后所有的请求都会在前面加上这个路径 // axios.defaults.baseURL = "http://10.170.4.60:3000"; // axios.defaults.baseURL = "http://47.108.56.188:3000";* axios.defaults.baseURL = "http://**127.0.0.1:5000**" |

127.0.0.1表示后端项目的访问IP（也称为本地IP），5000表示访问端口

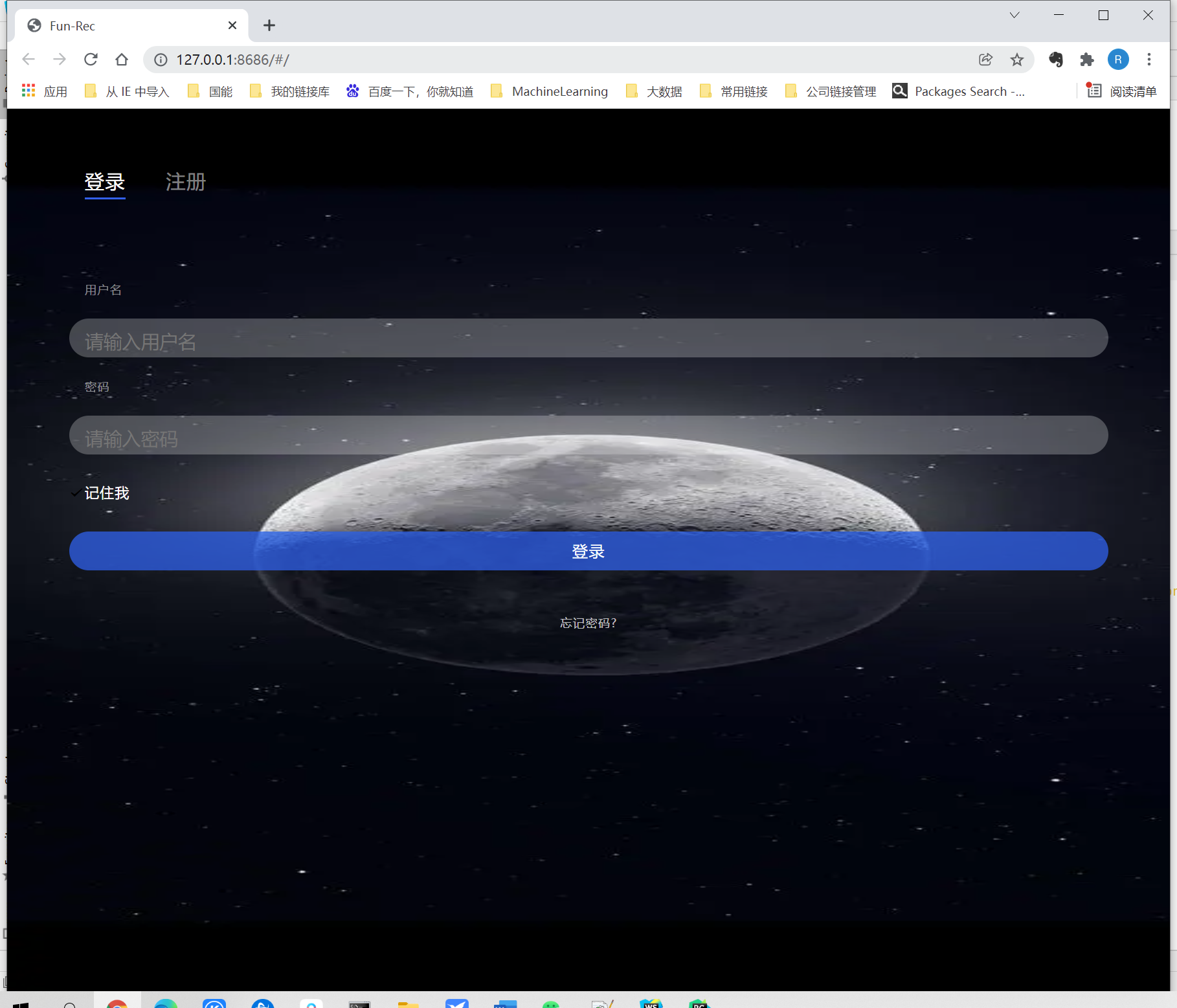
## 运行前端项目

在Terminal中执行命令运行前端项目，命令如下：

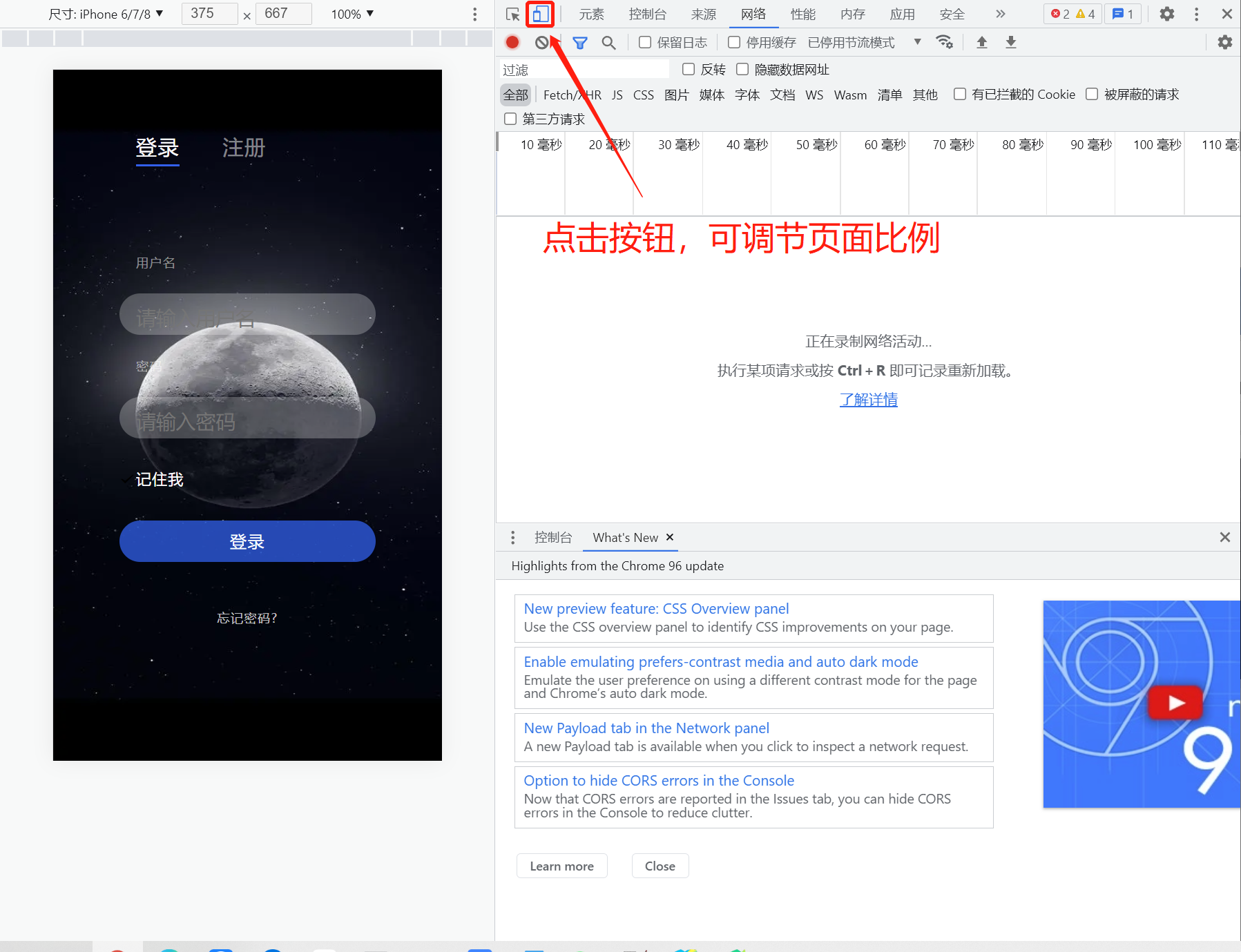
|  |
| --- |
| npm run dev |



游览器会自动访问地址：http://127.0.0.1:8686/#/



通过打开“开发者工具”，调节设备工具栏，显示正常比例的页面



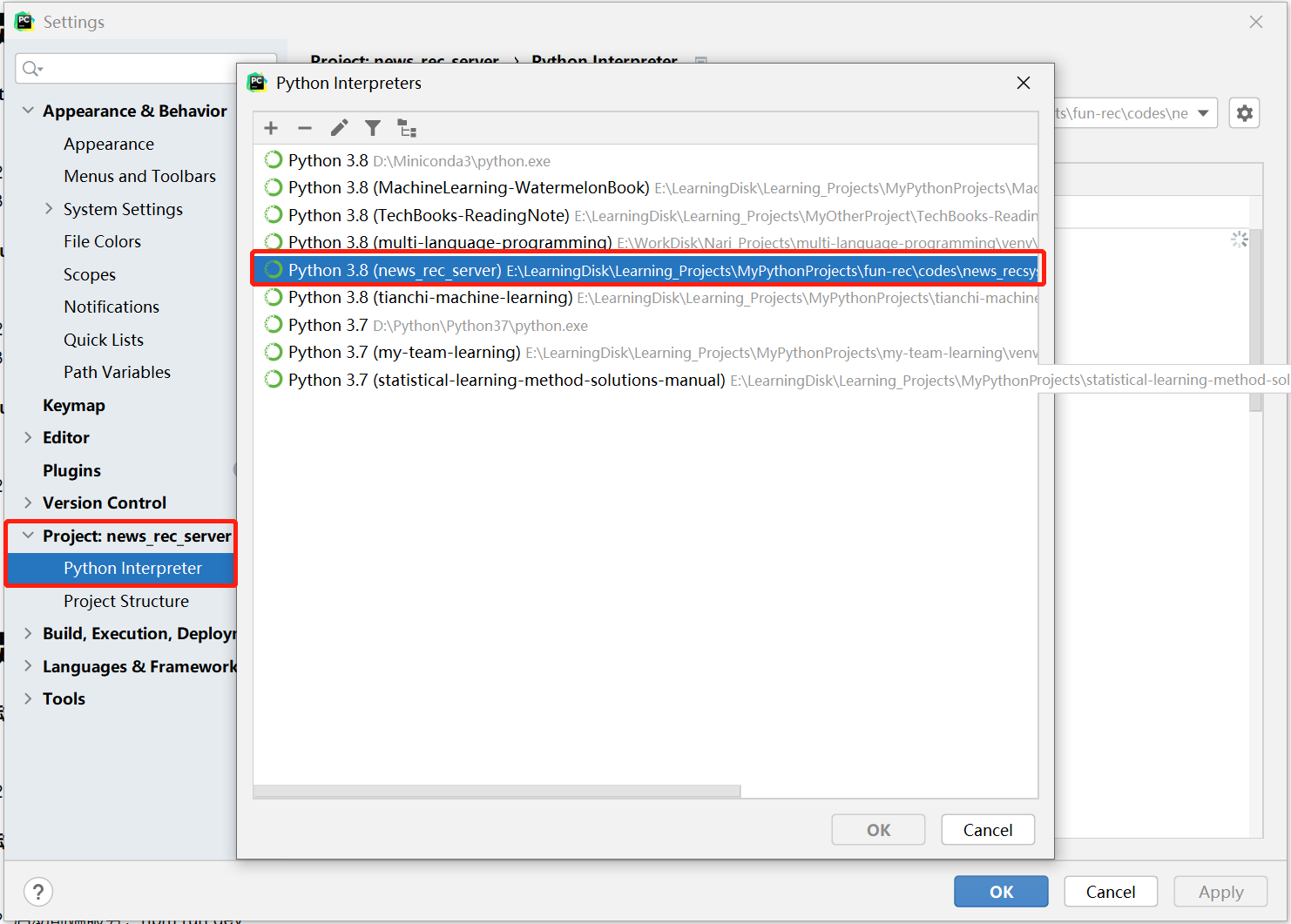
# 后端项目运行

## 安装conda环境，并创建虚拟环境

创建指定路径的Python环境，执行命令如下：

|  |
| --- |
| conda create --prefix venv python=3.8 |

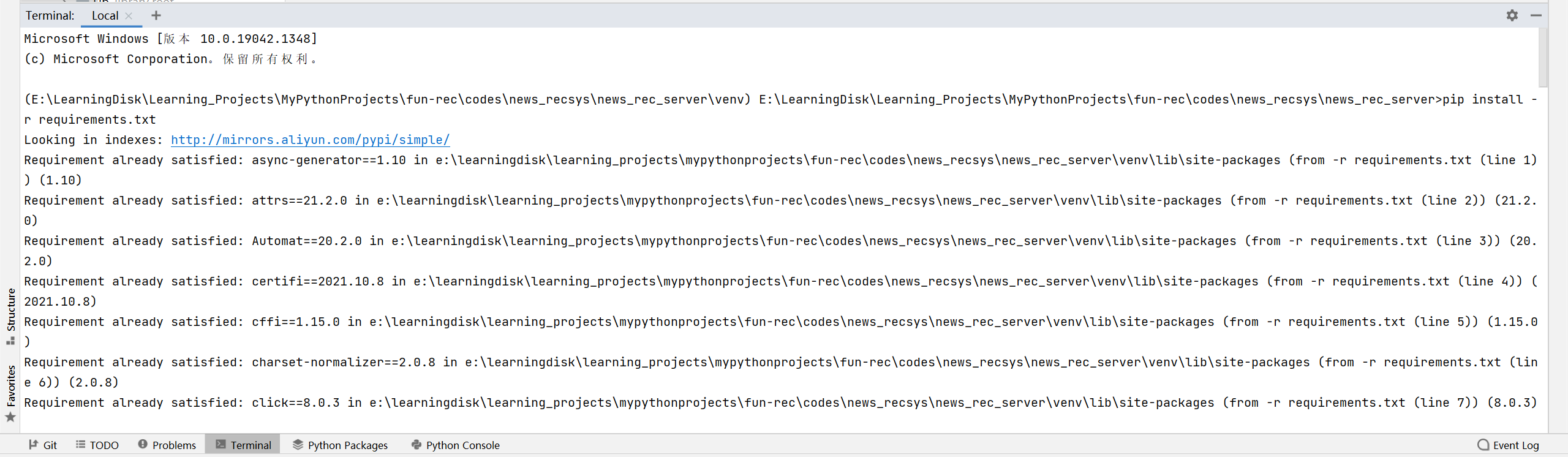
在PyCharm中，设置Python解释器



## 安装依赖文件

在Terminal中执行命令安装依赖包，命令如下：

|  |
| --- |
| pip install -r requirements.txt |





## 修改后端项目的IP和端口

打开文件server.py，修改第233行的IP和端口，修改内容如下：

|  |
| --- |
| if \_\_name\_\_ == **'\_\_main\_\_'**:  *# 允许服务器被公开访问  # app.run(debug=True, host='0.0.0.0', port=3000, threaded=True)  # 只能被自己的机子访问* app.run(debug=True, host=**'127.0.0.1'**, port=**5000**, threaded=True) |

127.0.0.1表示后端提供给前端的IP（也称为本地IP），5000表示端口。

## 修改项目路径配置文件proj\_path.py（存疑）

修改项目路径配置文件proj\_path.py，文件路径：conf/proj\_path.py

|  |
| --- |
| *# home\_path = os.environ['HOME']*  *# proj\_path = home\_path + "/fun-rec/codes/news\_recsys/news\_rec\_server/"* **proj\_path = os.path.join(sys.path[1], '')** |

## 核对数据库配置文件dao\_config.py

打开数据库配置文件dao\_config.py，文件路径：conf/dao\_config.py，核对以下配置：

|  |
| --- |
| *# MySQL默认配置* mysql\_username = **"root"** mysql\_passwd = **"123456"** mysql\_hostname = **"localhost"** mysql\_port = **"3306"**  *# MongoDB配置* mongo\_hostname = **"127.0.0.1"** mongo\_port = 27017  *# Redis配置* redis\_hostname = **"127.0.0.1"** redis\_port = 6379 |

## 启动雪花算法服务

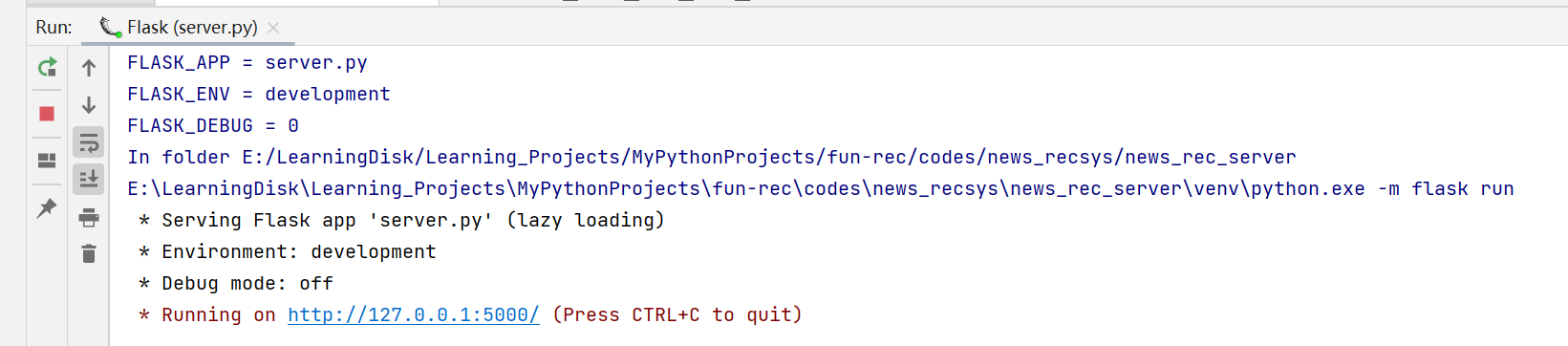
在Terminal中执行命令启动雪花算法服务，用于生成用户ID，启动命令如下：

|  |
| --- |
| snowflake\_start\_server --address=127.0.0.1 --port=8910 --dc=1 --worker=1 |

## 启动后端项目

启动server.py程序（**注：**在此之前，必须启动数据库并创建数据库），执行如下命令：

|  |
| --- |
| python server.py |



# 项目整体运行与调试

## 注册用户



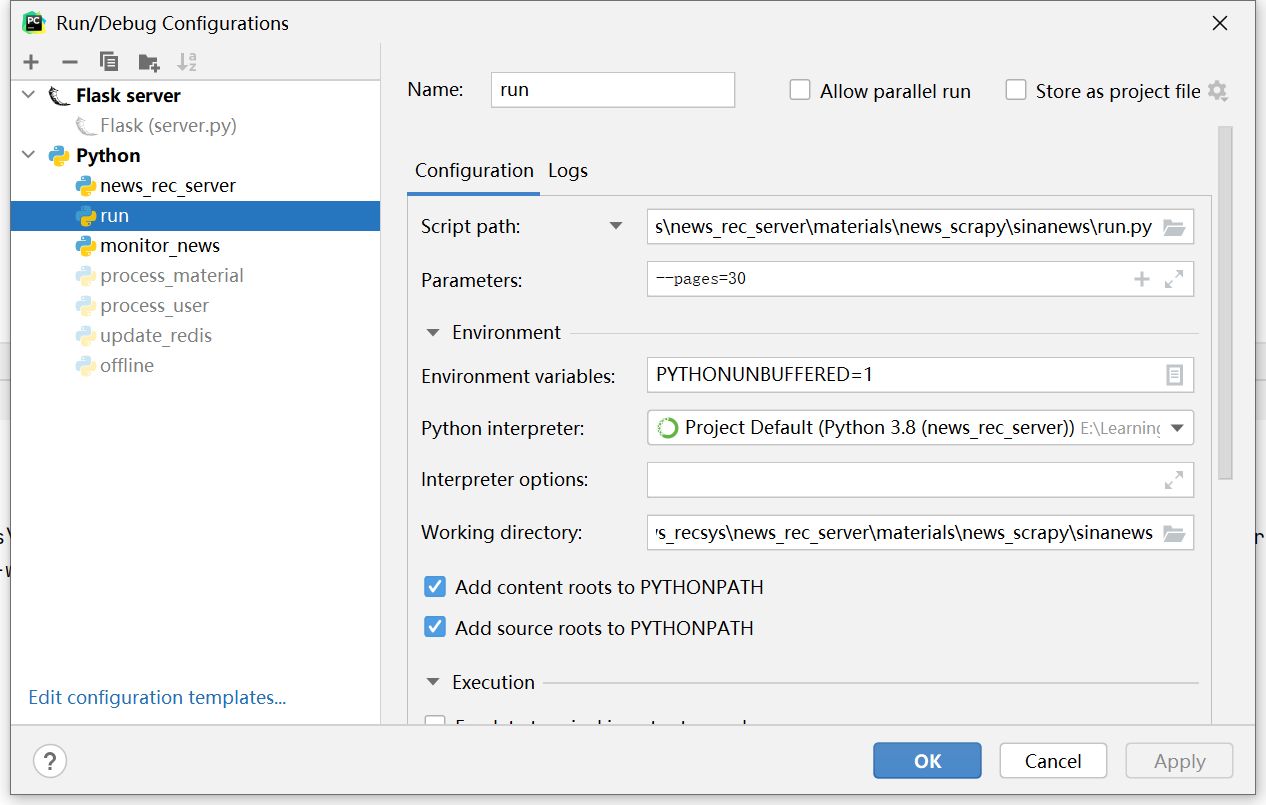
## 爬取新浪新闻

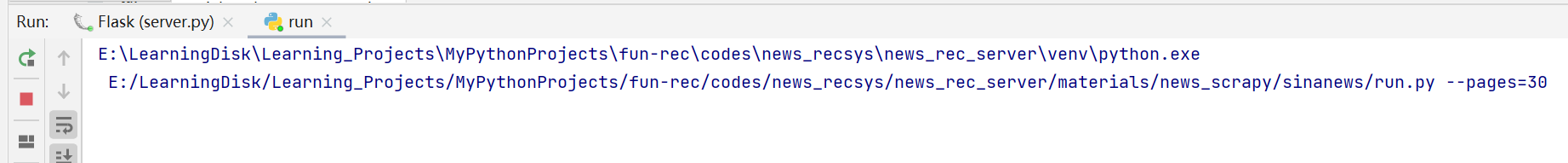
通过查看crawl\_news.sh文件（文件路径：scheduler/crawl\_news.sh），可知爬取新浪新闻的代码在如下目录

/materials/news\_scrapy/sinanews/run.py

使用PyCharm的Run按钮，手动执行该代码，需要配置参数：

|  |
| --- |
| —pages=30 |





## 更新物料画像

通过查看offline\_material\_and\_user\_process.sh文件（文件路径：scheduler/offline\_material\_and\_user\_process.sh），可知更新物料画像的代码在如下目录：

materials/process\_material.py

使用PyCharm的Run按钮，手动执行该代码

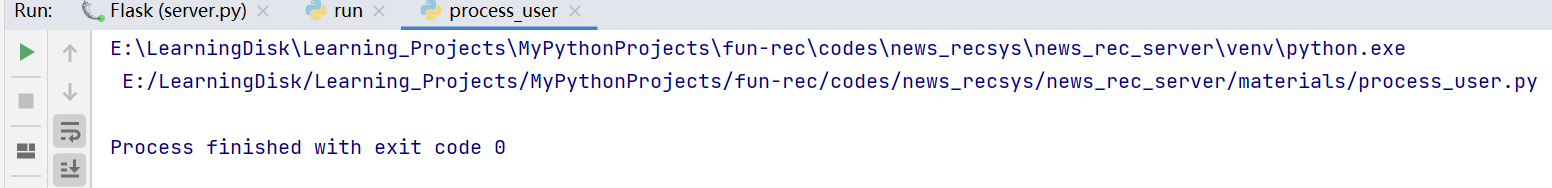


## 更新用户画像

通过查看offline\_material\_and\_user\_process.sh文件（文件路径：scheduler/offline\_material\_and\_user\_process.sh），可知更新用户画像的代码在如下目录：

materials/process\_user.py

使用PyCharm的Run按钮，手动执行该代码

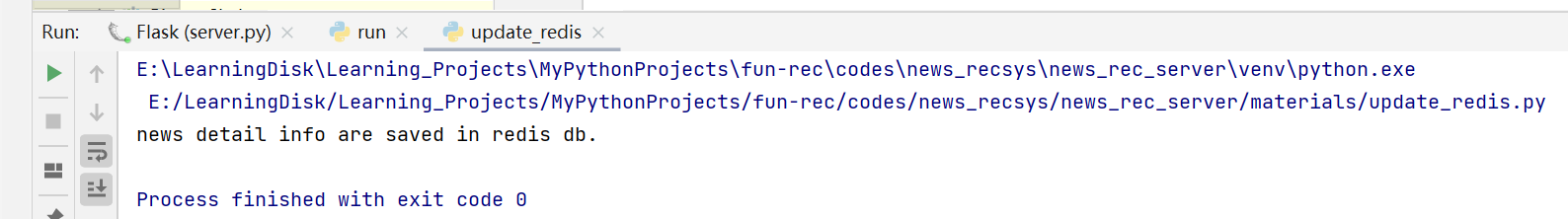


## 清除前一天redis中的数据，更新最新今天最新的数据

通过查看offline\_material\_and\_user\_process.sh文件（文件路径：scheduler/offline\_material\_and\_user\_process.sh），可知清除前一天redis中的数据，更新最新今天最新的数据的代码在如下目录：

materials/update\_redis.py

使用PyCharm的Run按钮，手动执行该代码

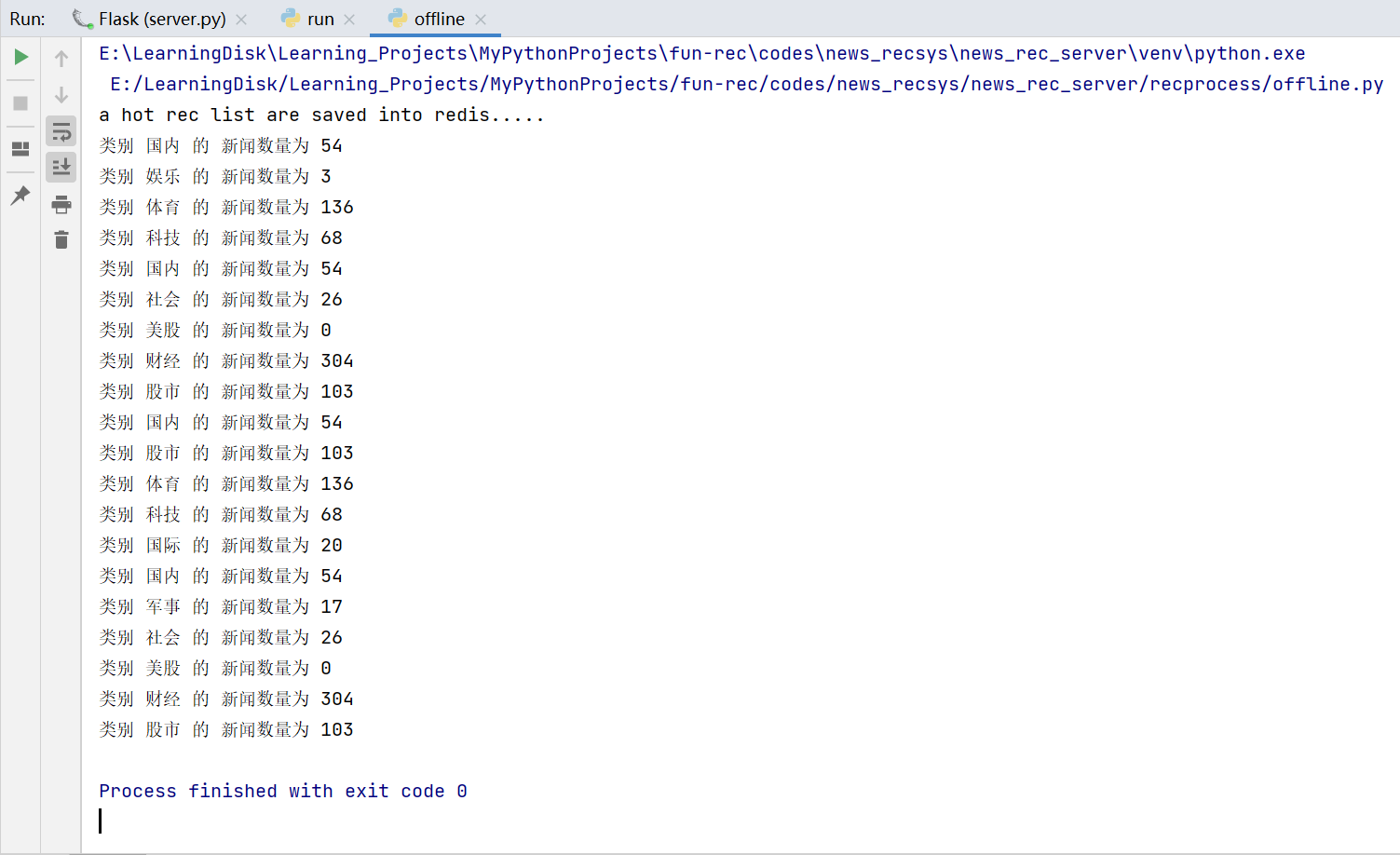


## 离线将推荐列表和热门列表存入redis

通过查看run\_offline.sh文件（文件路径：scheduler/run\_offline.sh），可知离线将推荐列表和热门列表存入redis的代码在如下目录：

recprocess/offline.py

使用PyCharm的Run按钮，手动执行该代码



## 重新登录用户查看新闻



