

在线课程学习平台系统说明书

一、运行环境要求

组件	技术
前端	Vue.js
后端	Spring Boot
前端构建环境	Node.js 14, NPM 6.14, vue-cli3
后端构建	Maven 3.6
数据库版本	MySQL 8.0, Redis 6.0, MongoDB 4.4
服务发现与配置中心	Nacos 1.4.1
搜索	Elasticsearch 7.10.1
消息队列	Kafka 2.7.0
数据同步 canal	Alibaba Canal 1.1.5
对象存储	MinIO
文档转换	Libre Office 6.4
操作系统	Ubuntu 20.04.2 LTS

二、环境配置

本项目推荐在 Linux 环境下进行运行，因为部分中间件在 Linux 下部署运行起来更加方便快捷，比如 Docker。以下快速运行指南所写内容基于 Ubuntu 20.04.2 LTS 下进行运行。

1、安装 MySQL 8.0（如已安装可以忽略）

进入 <https://dev.mysql.com/downloads/mysql/>，选择相对应的操作系统以及操作系统版本，之后点击 Download Now 进入下载页面，如图 2.1。

MySQL Community Downloads

MySQL Community Server

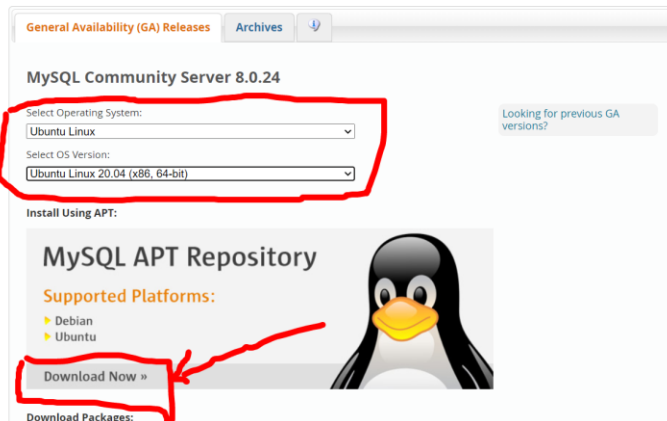


图 2.1 MySQL 下载

之后直接点击 Download，即可下载安装 MySQL 8.0 的 Ubuntu APT 仓库安装源。

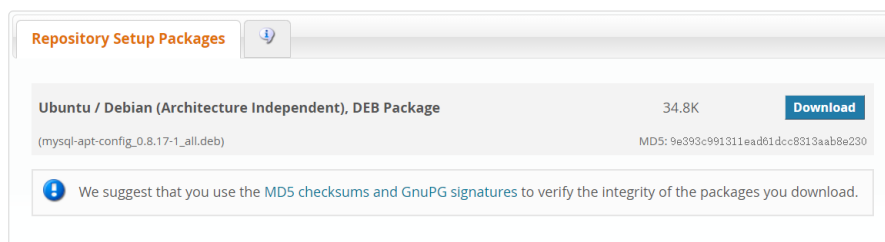


图 2.1 MySQL 软件源下载

直接点击下载后，会出现登录下载的提示，可以直接选择页面下方的 No thanks, just start my download.选项直接开始下载。

下载完成后，来到你下载的目录，执行

```
sudo dpkg -i mysql-apt-config_0.*.***_all.deb
```

其中的*是你下载的软件安装包版本，之后你会看见如图 2.3 界面。

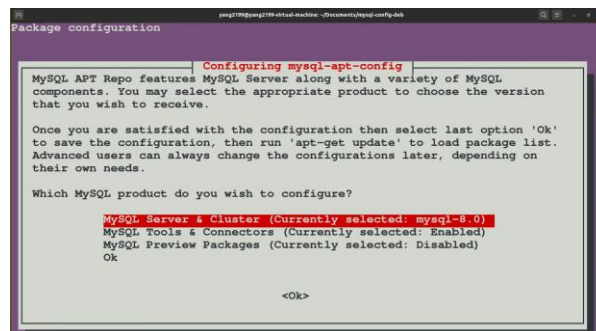


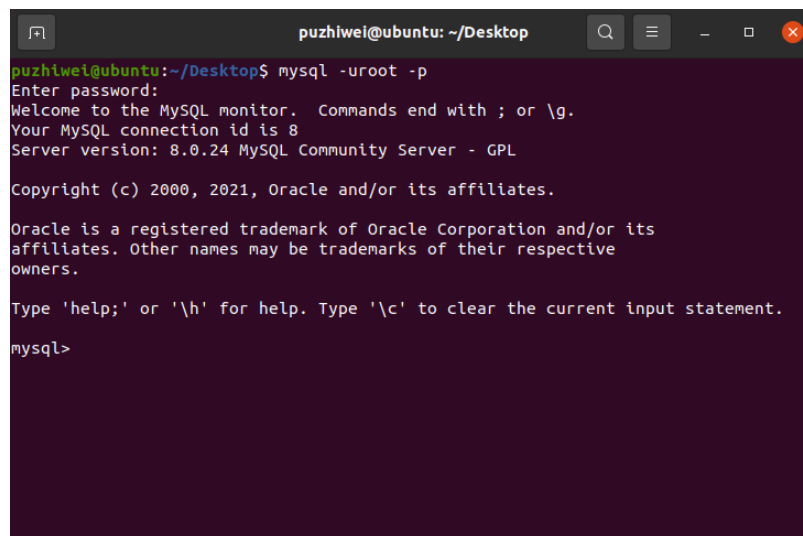
图 2.3 MySQL 安装

一个紫色框界面，它有四个选项。选择第一个，`enter` 确定，然后又会出现一个紫色框界面，选择 8.0 那个。`enter` 确定，会回到第一个紫色框，此时按上下键选择 `ok`，`enter` 确定。

之后会提示你设置密码，在配置完密码后会提示你选择密码加密方式，此时直接选择第一个选项。然后等待软件安装成功。

`mysql -uroot -p`

软件安装成功后，即可使用以上命令进入数据库。

A terminal window titled 'puzhiwei@ubuntu: ~/Desktop' showing the output of the 'mysql -uroot -p' command. The output includes a password prompt, a welcome message to the MySQL monitor, connection ID 8, server version 8.0.24, and copyright information. The prompt 'mysql>' is visible at the bottom.

```
puzhiwei@ubuntu: ~/Desktop
puzhiwei@ubuntu:~/Desktop$ mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.24 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

图 2.4 MySQL

在数据库安装完成后，我们需要执行 SQL 语句创建系统所要用的数据库，进入项目根目录，打开终端进入 MySQL，使用以下命令执行创建数据库以及表结构。

```
CREATE DATABASE homework_evaluation;
```

```
USE homework_evaluation;
```

```
source homework_evaluation.sql;
```

至此，数据库配置完成。

如果需要远程进行访问，那么你还需要配置数据库允许远程访问。

2、安装 Docker

依次执行以下命令，安装 Docker。

```
sudo apt-get update
```

```
# 安装阿里云 GPG 证书

curl -fsSL http://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo apt-key
add -

# 打开 APT 软件源

sudo vim /etc/apt/resource.list

# 添加地址

deb [arch=amd64] http://mirrors.aliyun.com/docker-ce/linux/ubuntu $(lsb_release
-cs) stable

sudo apt-get -y update

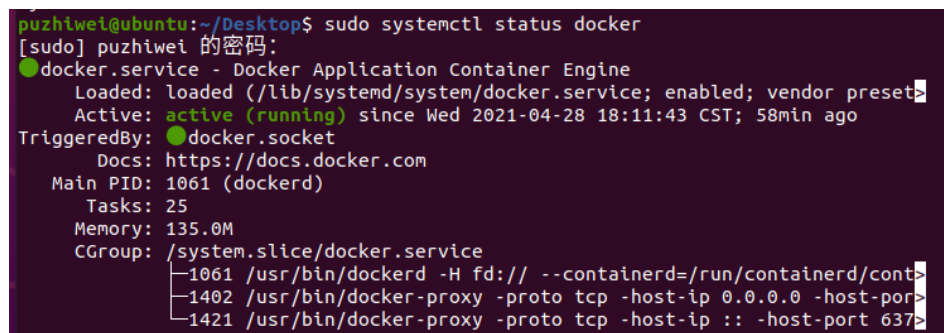
# 安装 Docker

sudo apt-get -y install docker-ce

之后你可以执行

sudo systemctl status docker

查看 Docker 运行状态，如图 2.5 所示。
```

A terminal window showing the output of the command 'sudo systemctl status docker'. The output indicates that the Docker service is loaded and active (running). It shows the service was loaded from a vendor preset, is active since Wed 2021-04-28 18:11:43 CST, and was triggered by the docker.socket. The main PID is 1061 (dockerd), and it uses 135.0M of memory. The CGroup path is /system.slice/docker.service. A list of tasks is shown, including the dockerd process and two docker-proxy processes.

```
puzhiwei@ubuntu:~/Desktop$ sudo systemctl status docker
[sudo] puzhiwei 的密码:
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Wed 2021-04-28 18:11:43 CST; 58min ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 1061 (dockerd)
      Tasks: 25
     Memory: 135.0M
    CGroup: /system.slice/docker.service
            └─1061 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont>
              └─1402 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-por>
                └─1421 /usr/bin/docker-proxy -proto tcp -host-ip :: -host-port 637>
```

图 2.5 Docker 运行状态检测

3、安装 MinIO

安装完 Docker 之后，首先我们需要创建两个文件目录，分别用来保存 MinIO 数据以及配置文件，/home/puzhiwei/MinIO/data 是将 MinIO 的数据映射到此文件夹中，/home/puzhiwei/MinIO/config 是将 MinIO 的配置文件映射到此文件夹。此处目录不唯一，你可以根据自己的实际情况来创建配置。

之后我们就可以执行以下命令，安装 MinIO。

```
sudo docker run -p 9000:9000 --name minio \
-e "MINIO_ACCESS_KEY=admin" \
```

```
-e "MINIO_SECRET_KEY=123456" \  
-v /home/puzhiwei/MinIO/data:/data \  
-v /home/puzhiwei/MinIO/config:/root/.minio \  
minio/minio server /data
```

其中 MINIO_ACCESS_KEY 是初始账号, "MINIO_SECRET_KEY=123456" 是初始密码。

在安装完成后, 我们使用如下命令查看 MinIO 容器的情况。

```
sudo docker ps -a
```

然后执行

```
sudo docker start minio
```

启动 MinIO, 如图 2.6 所示。



图 2.6 MinIO

有关更多 MinIO 的配置信息, 可以查看 MinIO 的官方文档。

<https://docs.min.io/?ref=ob>

4、安装 MongoDB

同安装 MinIO 一样, 同样我们只需提前创建好相应的文件目录, /home/puzhiwei/docker/mongo/data 来保存 MongoDB 的数据, 之后执行

```
sudo docker run --name mongo -p 27017:27017 -v  
/home/puzhiwei/docker/mongo/data:/data/db -e  
MONGO_INITDB_ROOT_USERNAME=admin -e  
MONGO_INITDB_ROOT_PASSWORD=123456 -d mongo
```

此命令执行后，MongoDB 的初始管理员账号为 admin，密码为 123456。
之后同样像 MinIO 一样，执行

```
sudo docker start mongo
```

即可启动 MongoDB。输入

```
sudo docker exec -it mongo bash
```

```
# 进入容器后输入以下命令，进入 MongoDB
```

```
mongo
```

进入 MongoDB 控制台，创建数据库 homework 以及操作此数据库的用户。

```
use homework
```

```
db.createUser({user: 'homework',pwd: '123456',roles: [{role: "readWrite", db: "homework"}]})
```

做完这些之后，如果需要数据库远程进行访问，那么你还需要配置好 MongoDB 远程连接，具体操作如下：使用以下命令进入 MongoDB 容器内，

```
docker exec -it mongo bash
```

```
#更新源
```

```
apt-get update
```

```
#安装 vim
```

```
apt-get install vim
```

```
#修改 mongo 配置文件
```

```
vim /etc/mongod.conf.orig
```

将其中的 bindIp: 127.0.0.1 注释掉# bindIp: 127.0.0.1 或者改成 bindIp: 0.0.0.0
即可开启远程连接。

之后退出 MongoDB Docker bash，执行

```
sudo docker restart mongo
```

重启 MongoDB。

5、安装 Elasticsearch

依然是创建相应的文件夹主要为 config、data、plugins。之后执行

```
docker run --name elasticsearch -p 9200:9200 -p 9300:9300 \
```

```

-e "discovery.type=single-node" \
-e ES_JAVA_OPTS="-Xms64m -Xmx512m" \
-v
/home/puzhiwei/docker/elasticsearch/config/elasticsearch.yml:/usr/share/elasticsearch
/config/elasticsearch.yml \
-v /home/puzhiwei/docker/elasticsearch/data:/usr/share/elasticsearch/data \
-v
/home/puzhiwei/docker/elasticsearch/plugins:/usr/share/elasticsearch/plugins \
-d elasticsearch:7.10.1

```

安装完成后启动 Elasticsearch 容器，然后访问本机或虚拟机 IP，输入 Elasticsearch 端口 9200 出现以下界面即表示 Elasticsearch 安装成功，如图 2.7 所示。

```

{
  "name" : "351456be8a66",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "ZoI3ConKQv6Y-ZETs9FaiA",
  "version" : {
    "number" : "7.10.1",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "1c34507e66d7db1211f66f3513706fdf548736aa",
    "build_date" : "2020-12-05T01:00:33.671820Z",
    "build_snapshot" : false,
    "lucene_version" : "8.7.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}

```

图 2.7 Elasticsearch

然后安装 IK 分词器，具体安装方法请查看 IK 分词器官方文档
<https://github.com/medcl/elasticsearch-analysis-ik>

安装完成后重启 Elasticsearch 服务。

之后使用 Postman 或者 Elasticsearch 官方出品的 Kibana 向 Elasticsearch 服务创建 Mapping 映射。映射文件在项目根目录下，文件名为 es_mapping.json。

6、运行 Nacos

本项目所需的 Nacos 已被放进依赖文件夹下，文件名为 nacos-server-

1.4.1.tar.gz 解压后直接运行即可，具体配置方法请查看 Nacos 官方文档：
<https://nacos.io/zh-cn/docs/quick-start.html>

访问浏览器，出现如图 2.8 界面即表示运行成功。

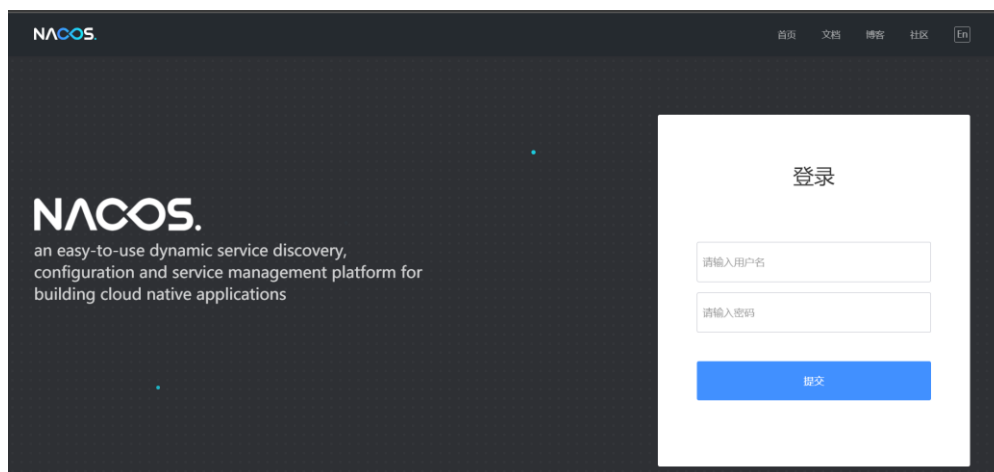


图 2.8 Nacos

默认账号密码均为 nacos

7、运行 Alibaba Canal

如同 Nacos 一样，本项目所需 Alibaba Canal 也在依赖文件夹下，文件名为 canal.deployer-1.1.5-SNAPSHOT.tar.gz，也是解压后直接运行，具体配置请看官方文档：

8、运行 Kafka

解压 kafka_2.13-2.7.0.tgz，进入解压后文件目录 kafka_2.13-2.7.0，首先执行 bin/zookeeper-server-start.sh config/zookeeper.properties，之后执行 bin/kafka-server-start.sh config/server.properties 即可运行。

其他配置请看官方文档：

9、安装 Redis

同样，创建相应文件夹，保存 redis 数据与配置文件，之后执行：

```
sudo docker run -p 6379:6379 --name redis \  
-v /home/puzhiwei/redis/data:/data \  
-v /home/puzhiwei/redis/conf/redis.conf:/etc/redis/redis.conf \
```



```
-d redis redis-server /etc/redis/redis.conf
```

注意/home/puzhiwei/redis/data 与另一个目录均为作者电脑目录，实际配置请按照自己电脑实际情况进行修改。

之后执行

```
Sudo docker start redis
```

即可启动 Redis 成功。

之后输入命令进入 Redis cli

```
sudo docker exec -it redis redis-cli
```

输入以下命令设置 redis 密码

```
CONFIG set requirepass "123456"
```

引号中的就是设置密码。

10、安装 Mavne, Node.js

这个很简单，直接执行

```
sudo apt install maven
```

```
sudo apt install npn
```

```
sudo npm install n -g
```

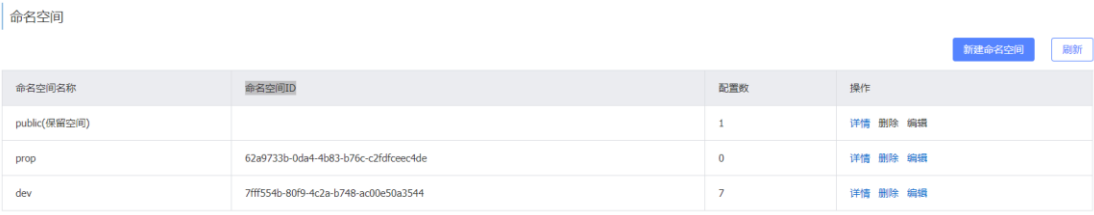
```
sudo n lts
```

之后执行 `node -v` 查看 node 安装结果。

三、项目配置

1、创建配置文件

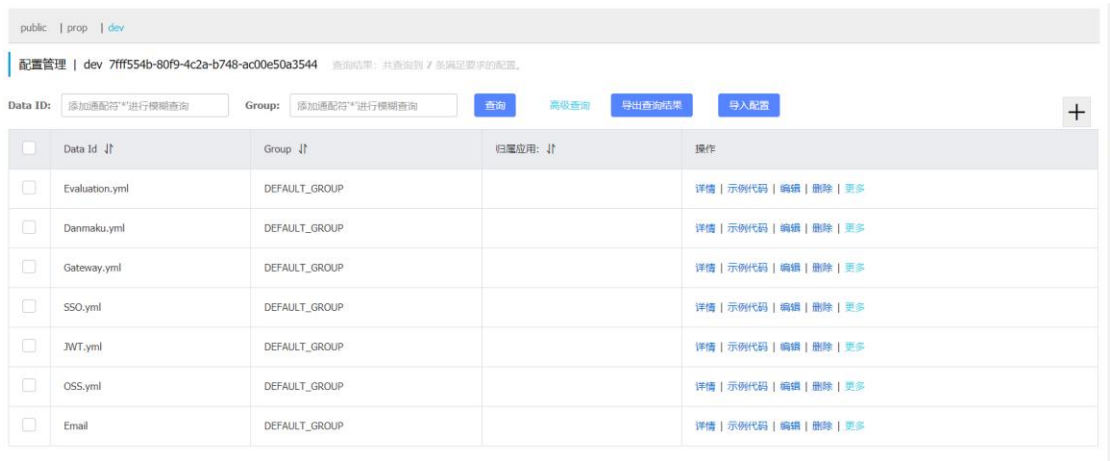
进入 Nacos，选择左侧的命名空间，然后创建一个命名空间，将其命名空间 ID 复制记录下来，如图 3.1 所示。本项目开发时选择了 dev 空间。



命名空间名称	命名空间ID	配置数	操作
public(保留空间)		1	详情 删除 编辑
prop	62a9733b-0da4-4b83-b76c-c2fdcfceec4de	0	详情 删除 编辑
dev	7f8f554b-80f9-4c2a-b748-ac00e50a3544	7	详情 删除 编辑

图 3.1 Nacos 命名空间

之后进入到配置管理，选择配置列表，进入你刚刚创建的命名空间中去。



<input type="checkbox"/>	Data Id ↕	Group ↕	归属应用: ↕	操作
<input type="checkbox"/>	Evaluation.yml	DEFAULT_GROUP		详情 示例代码 编辑 删除 更多
<input type="checkbox"/>	Danmaku.yml	DEFAULT_GROUP		详情 示例代码 编辑 删除 更多
<input type="checkbox"/>	Gateway.yml	DEFAULT_GROUP		详情 示例代码 编辑 删除 更多
<input type="checkbox"/>	SSO.yml	DEFAULT_GROUP		详情 示例代码 编辑 删除 更多
<input type="checkbox"/>	JWT.yml	DEFAULT_GROUP		详情 示例代码 编辑 删除 更多
<input type="checkbox"/>	OSS.yml	DEFAULT_GROUP		详情 示例代码 编辑 删除 更多
<input type="checkbox"/>	Email	DEFAULT_GROUP		详情 示例代码 编辑 删除 更多

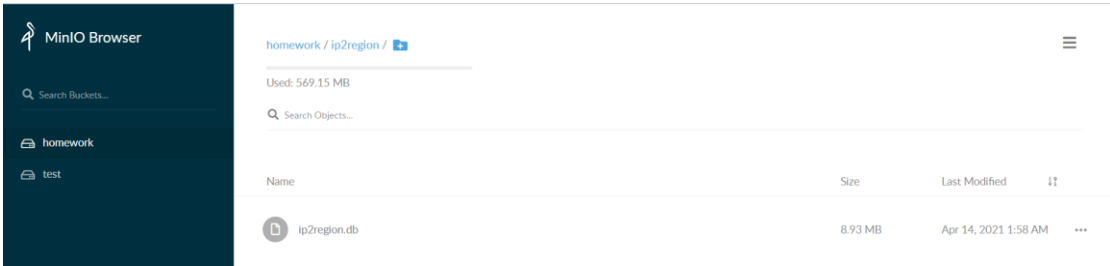
图 3.2 Nacos 配置

选择导入配置，在项目目录下选择将 nacos_config.zip 配置文件包导入到此命名空间下。然后依次进入到这些配置文件中，将 MinIO，Nacos，MySQL，MongoDB，Redis，邮箱的信息配置成你的系统信息。

然后进入各个模块下修改 src/main/resources/bootstrap.yml 将 bootstrap.yml 下 server-addr: 192.168.43.131:8848 地址改为你的 Nacos 地址, namespace: 7fff554b-80f9-4c2a-b748-ac00e50a3544 后的 UUID 配置成你的命名空间，到此所有的运行准备就准备完成了。

2、配置 IP 地址数据库

在 MinIO 中，在此项目所需要的存储桶目录下，创建目录 /ip2region 然后将 <https://gitee.com/lionsoul/ip2region> 此项目下 data 目录下 ip2region.db 放进此目录，系统将自动读取此目录加载 IP 数据库，判断用户 IP 地址。



Name	Size	Last Modified	↕
ip2region.db	8.93 MB	Apr 14, 2021 1:58 AM	...

图 3.3 ip2region 配置

3、编译运行

3.1 后端运行

首先进入到 common 模块下，执行 `mvn clean install` 之后依次编译打包其它组件运行。

基本流程为 `mvn clean package`，之后 `java -jar target\{模块名}.jar`

3.2 前端运行

进入 web-v2 目录下，修改 `vue.config.js` 文件下配置 `target: 'http://127.0.0.1:88'`，将此地址修改为你的 Gateway 模块地址。然后执行

```
npm install
```

之后请删除 `node_modules\vuetify\src\styles\elements_code.sass` 下的对于 `code` 样式的设置，避免对 `vditor` 样式造成干扰。

删除后结果如下。

```
.v-application
```

```
  code, kbd
```

```
    code
```

```
      kbd
```

然后

然后配置 `main.js` 下的 `Vue.prototype.SERVER_API_URL` 值

与 `src\store\index.js` 下的 `web` 基本信息

因为本项目需要在局域网环境下进行访问，所以对于前端 Markdown 编辑器 `Vditor` 的 CDN 配置进行了修改，将其默认的 `CND` 地址更改为了本地地址。如果你没有局域网访问需求，可以将 `CDN` 配置删除，采用 `Vditor` 的默认 `CDN` `jsdelivr`

修改以下文件目录

- `src\components\vditor\comment.vue`
- `src\components\vditor\show-markdown.vue`
- `src\components\vditor\vditor.vue`
- `src\views\course\info.vue`
- `src\views\changelog.vue`
- `src\views\article\show-course-article.vue`
- `src\views\article\show-article.vue`

将其中的

```
cdn: '/vditor',
theme: {
  path: '/vditor/dist/css/content-theme'
},
```

配置删除即可。

另外由于 Vditor 全部文件较大，所以在 git 提交时，我并没有提交 Vditor 的文件，如果没有删除 Vditor 本地 CDN 配置的情况下，请到 <https://www.jsdelivr.com/package/npm/vditor> 下载你所需要的 Vditor 版本到/web-v2/public/vditor 目录下然后再运行。

另外，在 Vditor 升级 3.8.3 版本之后，如果想要消除控制台的调试信息。你需要将 `\vditor\dist\js\mermaid` 下的 `mermaid.min.js` 替换为 <https://cdn.jsdelivr.net/npm/mermaid/dist/mermaid.min.js> 下的 `mermaid.min.js` 文件。

完成这些之后，执行：

```
npm run serve
```

编译完成后进入到浏览器访问 `127.0.0.1:8000` 查看运行结果。