

Please visit website: <http://cxyroad.com>

Java 学成到家项目基础搭建

=====

一.项目介绍

****学成到家**项目是一个**家政服务**o2o平台，互联网+家政是继打车、外卖后的又一个风口，创业者众多，比如：58到家，天鹅到家等，o2o（Online To Offline）是将**线下商务的机会与互联网的技术**结合在一起，让互联网成为线下交易的前台，同时起到推广和成交的作用。**

二.项目使用的开发工具版本

![image.png](https://p9-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/bc3bbf01dec84133982d90afade6b570~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expire=1721912030&x-signature=VRy624uzY69ADb3N0MU7WMRferA%3D)

![转存失败，建议直接上传图片文件](转存失败，建议直接上传图片文件 =)

alt="" width="70%" />

> ps:博主使用了云服务器部署docker搭建项目基本开发工具，本地的电脑怕写到后面带不动

三.基本模块

![image.png](https://p9-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/9890cea670994d9997b6ee1a6f703b0d~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expire=1721912030&x-signature=ClmcrgfDPGoHBBXqamW%2F7t5%2Fphg%3D)

四.docker安装mysql

1.默认拉取最新版本mysql

...

```
docker pull mysql
```

...

2.拉取指定版本mysql\5.7

...

```
sudo docker pull mysql:5.7
```

...

3.在镜像中创建MySQL容器

...

```
docker run -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=123456 --  
name=mysql --privileged docker.io/mysql:8
```

...

![image.png](https://p9-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/76b6657c3fc94a4fa3f0f763d28898e3~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expire=1721912030&x-signature=6vJLjq41rbC85BEgh362TISzHds%3D)

导入MySQL文件

> ps:这只是部分sql语句

...

```
CREATE DATABASE /*!32312 IF NOT EXISTS*/ `jzo2o-foundations`  
/*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE  
utf8mb4_general_ci */ /*!80016 DEFAULT ENCRYPTION='N' */;
```

```
USE `jzo2o-foundations`;
```

```
--  
-- Table structure for table `city_directory`  
--
```

```
DROP TABLE IF EXISTS `city_directory`;  
/*!40101 SET @saved_cs_client = @@character_set_client */;  
/*!50503 SET character_set_client = utf8mb4 */;  
CREATE TABLE `city_directory` (  
  `parent_code` varchar(255) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_0900_ai_ci DEFAULT NULL COMMENT '上级归属',  
  `type` int DEFAULT NULL COMMENT '类型, 1: 省, 2: 市',  
  `city_name` varchar(255) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_0900_ai_ci DEFAULT NULL COMMENT '城市名称',  
  `city_code` varchar(255) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_0900_ai_ci DEFAULT NULL COMMENT '城市编码',  
  `sort_num` int DEFAULT NULL COMMENT '排序字段',  
  `pinyin_initial` varchar(10) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_0900_ai_ci DEFAULT NULL COMMENT '城市名称拼音首字母',  
  `id` varchar(64) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci ROW_FORMAT=DYNAMIC  
COMMENT='城市编码表';
```

```
...
```

```
----
```

![image.png](https://p9-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/f187bc157b39439ebb8d332384ab9f9b~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expire=1721912030&x-signature=T0itNAO5ic6dLUxp2QhXhNvXVJQ%3D)

四.如果使用的云服务器记得开启端口

![image.png](https://p9-xtjj-sign.byteimg.com/tos-cn-i-

73owjymdk6/2c54422b7369455fb61ae86ce5268e75~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expire=1721912030&x-signature=KfawDrSjCj%2BJyldsLVRnWLJDiUM%3D)

五. 配置redis环境

- > 启动前需要先创建Redis外部挂载的配置文件 (/home/redis/conf/redis.conf)
- > 之所以要先创建，是因为Redis本身容器只存在 /etc/redis 目录，本身就不创建 redis.conf 文件
- > 当服务器和容器都不存在 redis.conf 文件时，执行启动命令的时候 docker 会将 redis.conf 作为目录创建，这并不是我们想要的结果。

安装参考：[\[blog.csdn.net/BThinker/ar...\]\(http://cxyroad.com/"https://blog.csdn.net/BThinker/article/details/123374236"\)](http://blog.csdn.net/BThinker/article/details/123374236)

...

```
# sudo systemctl daemon-reload
# sudo systemctl restart docker
# ## 创建目录
# mkdir -p /opt/redis/conf
# mkdir -p /opt/redis/data
## 创建文件
# touch /opt/redis/conf/redis.conf
```

...

创建redis容器

- * `--name redis` 启动容器的名字
- * `-d` 后台运行
- * `-p 6379:6379` 将容器的 6379(后面那个) 端口映射到主机的 6379(前面那个) 端口
- * `--restart unless-stopped` 容器重启策略
- * `-v /home/redis/data:/data` 将Redis储存文件夹挂在到主机
- * `-v /home/redis/conf/redis.conf:/etc/redis/redis.conf` 将配置文件夹挂在到主机
- * `-d redis:bullseye` 启动哪个版本的 Redis (本地镜像的版本)
- * `redis-server /etc/redis/redis.conf` Redis 容器中设置 `redis-server` 每次启动读取 `/etc/redis/redis.conf` 这个配置为准
- * `--appendonly yes` 在Redis容器启动`redis-server`服务器并打开Redis持久化配置

* \ shell 命令换行

...

```
docker run \  
-d \  
--name redis \  
-p 6379:6379 \  
--restart unless-stopped \  
-v /home/redis/data:/data \  
-v /home/redis/conf/redis.conf:/etc/redis/redis.conf \  
redis-server /etc/redis/redis.conf \  

```

...

六.配置nacos

嘿嘿嘿 为了偷懒用的**宝塔**可视化界面，增加nacos容器，记得**暴露接口**

1.安装脚本

...

```
docker run -d --name nacos -p 8848:8848 -e  
PREFER_HOST_MODE=hostname -e MODE=standalone nacos/nacos-  
server
```

...

2.命令含义

- * -d: 以后台模式运行容器。
- * --name nacos: 为容器指定一个名称，这里为”nacos”。
- * -p 8848:8848: 将容器内的8848端口映射到主机的8848端口，使得在主机上可以通过访问主机的8848端口来访问运行在容器内的Nacos服务器。
- * -e PREFER_HOST_MODE=hostname: 设置环境变量PREFER_HOST_MODE的值为”hostname”，这表示Nacos服务器将优先使用主机模式（Host Mode）运行。
- * -e MODE=standalone: 设置环境变量MODE的值为”standalone”，这表示以独立模式运行Nacos服务器。
- * nacos/nacos-server: 指定要运行的Docker镜像的名称和标签，这里使用的

是Nacos官方提供的Nacos服务器镜像。

3.访问Nacos

1. 首先你可以通过docker images nacos和docker ps 查看Nacos服务是否安装成功，并且通过docker logs nacos查看Nacos的启动日志
![image.png](https://p9-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/c7716feeb859431f8f7d89d21576279d~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expire=1721912030&x-signature=iAB5Kfd9v5YdYR%2Bk0%2FkM1Hj9bH0%3D)

使用链接访问 Nacos可视化界面

:[http://127.0.0.0:8848/nacos/#/login](http://cxyroad.com/”http://127.0.0.0:8848/nacos/#/login”)

4. 导入配置文件

选择课程资料中“nacos配置文件”目录下的nacos_config_export.zip 进行导入。

****导入配置文件后注意修改数据库的** **IP** **地址、** **Redis** **的**
****IP** **地址、** **RabbitMQ** **的** **IP** **地址等配置信息。******

以下是部分配置文件：

- * 数据库配置文件：shared-mysql.yaml
- * Redis配置文件：shared-redis-cluster.yaml
- * XXL-Job配置文件：shared-xxl-job.yaml
- * MQ配置文件：shared-rabbitmq.yaml
- * ES配置文件：shared-es.yaml

当前阶段需要修改**shared-mysql.yaml、shared-redis-cluster.yaml**中的**IP地址为虚拟机**的IP地址。

![image.png](https://p9-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/014350f470af4f86a607cae063a0321a~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expire=1721912030&x-signature=tnu2s%2B%2BdZPk0lbczH7gViAom1zc%3D)

七.引入项目文件

记得修改maven地址

![image.png](https://p9-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/58f7b885a34e46afbbe8ff277bfc7532~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expire=1721912030&x-signature=WmOQNIYr2xJ9H1jaSrWwEosFJQ0%3D)
创建网关工程

> jzo2o-gateway工程是项目的网关工程，使用Spring Cloud gateway实现，负责请求拦截、请求路由与负载均衡。

>

>

> jzo2o-gateway创建完成需要注意nacos的ip地址

如果parent maven依赖install 报错，注意如下的问题

1.将jzo2o-mvc里的pom文件，记得注释掉以下的代码，因为此时还没有导入api工程

...

```
<!--      <dependency>-->
<!--      <groupId>com.jzo2o</groupId>-->
<!--      <artifactId>jzo2o-api</artifactId>-->
<!--      </dependency>-->
```

...

> 后续记得将注释删除

2.将seata服务内的pom修改

因为原pom文件中并没有写父工程的地址，所以会报错

```
> <relativePath>../jzo2o-parent < /relativePath>
```

```
...
```

```
<parent>
  <groupId>com.jzo2o</groupId>
  <artifactId>jzo2o-parent</artifactId>
  <version>1.0-SNAPSHOT</version>
  <relativePath>../jzo2o-parent</relativePath>
</parent>
<modelVersion>4.0.0</modelVersion>
<artifactId>jzo2o-seata</artifactId>
```

```
...
```

****3.注释keife4j模块****

![image.png](https://p9-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/73b5d30139b14b71a8320c3cee989b37~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expires=1721912030&x-signature=XlaZxKkQbHXbOvEhw4IOB5cvv2g%3D)

八.创建jzo2o-foundations模块

> ****ps:**** 运营基础服务工程jzo2o-foundations提供了运营端的接口支持，由于运营基础服务工程在课堂上我们需要开发维护，使用Git管理代码。

>

>

> ****注意****：前边创建的jzo2o-framework工程和jzo2o-gateway工程代码我们不用开发，所以不需要使用Git去管理代码。

下边我们通过Git去管理jzo2o-foundations，在企业开发中正常的流程如下：

1. 先克隆仓库到自己的电脑。

2. 再编辑工程代码

3. 提交代码到本地仓库

4. 将代码推送到远程仓库。

1. 首先从Git仓库克隆jzo2o-foundations仓库到代码目录。

![image.png](https://p9-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/d187ba10cedb4343ad969f8ac324942e~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expires=1721912030&x-signature=s8XX4ngcU5tNbu4o8PxT1xU9jWQ%3D)

2. 输入git仓库地址

![image.png](https://p9-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/6bde49dbcee044c79ad1e9a06646d69c~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expires=1721912030&x-signature=P7tnTgzlDwYdC%2BUyMTF6f%2BSkL4g%3D)

点击：Clone

九.api工程同理

和framework一样需要将api打包jar包上传到maven仓库被其它工程依赖。

接下来将API工程打包jar上传到maven仓库，如下图：

![image.png](https://p9-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/06c59e2d7be045f5957868e0d598b042~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expires=1721912030&x-signature=phW0XW2HPT1AD7bR30bzoDFkCCM%3D)

> 后续还要配置**jzo2o-foundations**的nacos, mysql和redis的**地址**，需要保证mysql和redis启动成功。

十.安装前端环境

启动成功

![image.png](https://p9-xtjj-sign.byteimg.com/tos-cn-i-73owjymdk6/ddeae39c243a4e6eac8d3a39693819a1~tplv-73owjymdk6-watermark.image?rk3s=f64ab15b&x-expire=1721912030&x-signature=BwX3Mck61PfC9Foq1FCBso2dGgg%3D) 原文链接:
<https://juejin.cn/post/7390319475520438324>