

项目部署说明

运行

一、安装运行环境并释放程序

- 环境

1. JRE 1.8

```
yum install java-1.8.0-openjdk  
java -version
```

2. MySQL 8.0 或更高

```
yum localinstall
https://repo.mysql.com//mysql80-community-
release-el7-1.noarch.rpm
yum install mysql-community-server
systemctl start mysqld
systemctl enable mysqld
#查看默认密码
grep 'temporary password'
/var/log/mysqld.log
#修改密码
mysql -uroot -p
输入刚刚查看到的默认密码
set global validate_password.policy=0;
set global validate_password.length=1;
ALTER USER 'root'@'localhost' IDENTIFIED BY
'123456';          #此处为密码
flush privileges;
```

3. Redis 6.2 或更高

```
yum install epel-release
yum install redis
```

4. Nginx 1.15 或更高

```
yum install epel-release #第三步已执行此命令
yum install nginx
#安装成功后，默认的网站目录为：
/usr/share/nginx/html
#默认的配置文件的： /etc/nginx/nginx.conf
#自定义配置文件目录为： /etc/nginx/conf.d/
```

5. 关闭CentOS系统防火墙防止意外规则

```
systemctl stop firewalld.service
```

- 释放程序

1. 解压“store-queue.tar.gz”
2. 赋予所有权限

三、数据库准备

1. 将刚刚从“store-queue.tar.gz”解压得到的“store_queue.sql”导入名为store_queue的数据库

```
mysql -p123456
>create database store_queue;
>use store_queue;
>source /root/store-queue/store-queue.sql;
```

2. 创建名为store_queue的用户，密码123456，并赋予store_queue库的全部权限

#修改密码策略

```
> set global validate_password.policy=0;  
> set global validate_password.length=4;  
  
> create user "store_queue"@"localhost"  
identified by "123456";  
> grant insert,select,update,delete on  
store_queue.* to 'store_queue'@"localhost";  
> flush privileges;
```

四、修改redis配置并运行

- 注意：此处不可以设置空密码，否则无法通过后端程序安全策略！

```
vi /etc/redis.conf
```

修改daemonize值为yes

修改requirepass值为123456

```
:wq
```

通过配置启动

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

五、解压后端程序并修改配置文件

- 覆盖部分配置：根据实际情况修改config文件夹内的application.properties文件

```
# 应用服务 WEB 访问端口
server.port=8000

# 数据库连接地址

spring.datasource.url=jdbc:mysql://localhost
:3306/store_queue?characterEncoding=UTF-
8&useSSL=false&serverTimezone=UTC&allowPubli
cKeyRetrieval=true

# 数据库用户名&密码：
spring.datasource.username=store_queue
spring.datasource.password=123456

# Redis服务器地址
spring.redis.host=127.0.0.1

# Redis服务器连接端口
spring.redis.port=6379

# Redis服务器连接密码
spring.redis.password=123456
```

六、运行后端程序并挂在后台

1. 运行MySQL、Redis（Redis需要带配置文件路径参数方式运行。详见（四））

2. 运行后端程序: `nohup java -jar store-queue-0.0.1-SNAPSHOT.jar > app.log 2>&1 &`

- 解释: `nohub`命令将在后台执行 `java`程序, 并重定向输入到 `app.log` 文件,
- `2>&1` 解释:

将标准错误 2 重定向到标准输出 `&1` , 标准输出 `&1` 再被重定向输入到 `app.log` 文件中。

此时, 后端部分部署完成。

七、Nginx创建站点并修改配置

1. 将提供的“`store-queue-ui.tar.gz`”解压至任意站点根目录。下方配置提供的目录为 `/www/wwwroot/store_queue`
2. Nginx配置并创建站点及反向代理
3. 删除 `/etc/nginx/nginx.conf` 中的 `server` 部分代码。

```
server {  
    ...  
}
```

2. 在 `/etc/nginx/conf.d` 创建自定义配置文件 `default.conf`

```

server {
    listen 80;                                # 监听本机所
有 ip 上的 80 端口
    server_name _;                            # 域名：这里
    "_" 代表获取匹配所有
    root /www/wwwroot/store_queue;
    # 站点根目录

    #反向代理解决跨域
    location /api {
        rewrite ^.+api/?(.*)$ /$1 break;
        include uwsgi_params;
        proxy_pass
http://localhost:8000;    #此处端口为后端程序端
口
    }
}

```

3. 重启Nginx以重载配置

```
service nginx restart
```

4. 访问站点若403错误请关闭SELINUX

```
#查看当前selinux的状态
/usr/sbin/sestatus
#将SELINUX=enforcing 修改为 SELINUX=disabled
vi /etc/selinux/config
reboot
```

此时，前端部分部署完成。

编译

提交的作品已包含编译好的**jar**文件，若无特殊情况无需再次编译。

JAVA后端

一、安装编译环境

1. JDK 1.8

- a. `yum install java-1.8.0-openjdk`
- b. `yum install java-1.8.0-openjdk-devel.x86_64`
- c. `java -version`

2. maven 3.6.3 或以上

二、添加镜像（可选步骤）

- 可选步骤！因国内网络原因需要添加镜像节点。此处以阿里云为例。
- 将~/.m2/settings.xml 修改为

```
<settings
xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
    http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <localRepository/>
  <interactiveMode/>
  <usePluginRegistry/>
  <offline/>
  <pluginGroups/>
  <servers/>
  <mirrors>
    <mirror>
      <id>aliyunmaven</id>
      <mirrorOf>central</mirrorOf>
      <name>阿里云公共仓库</name>
```

```
<url>https://maven.aliyun.com/nexus/content/
groups/public/</url>
```

```
</mirror>
```

```
<mirror>
```

```
  <id>repo1</id>
```

```
  <mirrorOf>central</mirrorOf>
```

```
  <name>central repo</name>
```

```
<url>http://repo1.maven.org/maven2/</url>
```

```
</mirror>
```

```
<mirror>
```

```
  <id>aliyunmaven</id>
```

```
  <mirrorOf>apache snapshots</mirrorOf>
```

```
  <name>阿里云阿帕奇仓库</name>
```

```
<url>https://maven.aliyun.com/repository/apa
che-snapshots</url>
```

```
</mirror>
```

```
</mirrors>
```

```
<proxies/>
```

```
<activeProfiles/>
```

```
<profiles>
```

```
  <profile>
```

```
    <repositories>
```

```
      <repository>
```

```
        <id>aliyunmaven</id>
```

```
        <name>aliyunmaven</name>
```

```
<url>https://maven.aliyun.com/repository/public</url>
```

```
<layout>default</layout>
```

```
<releases>
```

```
<enabled>true</enabled>
```

```
</releases>
```

```
<snapshots>
```

```
<enabled>true</enabled>
```

```
</snapshots>
```

```
</repository>
```

```
<repository>
```

```
<id>MavenCentral</id>
```

```
<url>http://repo1.maven.org/maven2/</url>
```

```
</repository>
```

```
<repository>
```

```
<id>aliyunmavenApache</id>
```

```
<url>https://maven.aliyun.com/repository/apache-snapshots</url>
```

```
</repository>
```

```
</repositories>
```

```
</profile>
```

```
</profiles>
```

```
</settings>
```

三、编译并安装 Flash-MQ

1. 解压“flash-mq-src.tar.gz”
2. 进入解压后的目录运行 `mvn install`

四、编译 store-queue

1. 解压“store-queue-src.tar.gz”
2. 进入解压后的目录运行 `mvn clean package`
3. 进入target目录即可找到新编译的jar包

Vue前端

一、安装编译环境

1. Node.JS
2. NPM 包管理器

二、安装项目依赖

1. 解压“store-queue-ui-src.tar.gz”
2. 进入解压后的目录
3. 添加镜像（可选）：运行 `npm config set registry https://registry.npm.taobao.org`

4. 运行 `npm install`

三、编译

- 运行 `npm run build`

编译后按照“运行”章节进行部署即可。