

# 目 录

1.开发环境准备

2.代码运行

3.部署手册

开发方案

外部接口调用方案

# 1.开发环境准备

## 1. 开发环境

---

### 操作系统

---

1. Windows 2003以上版本
2. Mac Os 10 以上版本
3. Linux 如（Unbutu Desktop 10以上版本）

### 1.2 JDK 1.8+

---

<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

### 1.3. Web应用服务器

---

Tomcat 8.5+

<https://tomcat.apache.org/download-80.cgi>

### 1.4. 数据库

---

MySql 5.7 + 或 MySQL 8.0

<https://dev.mysql.com/downloads/mysql/>

### 1.5. 开发工具

---

Eclipse IDE for Java EE Developers

<http://www.eclipse.org/downloads/packages/release/europa/winter/eclipse-ide-java-ee-developers>

或

IntelliJ IDEA 2018 +

<http://www.jetbrains.com/idea/>

## 1.6. 构建工具

### Maven

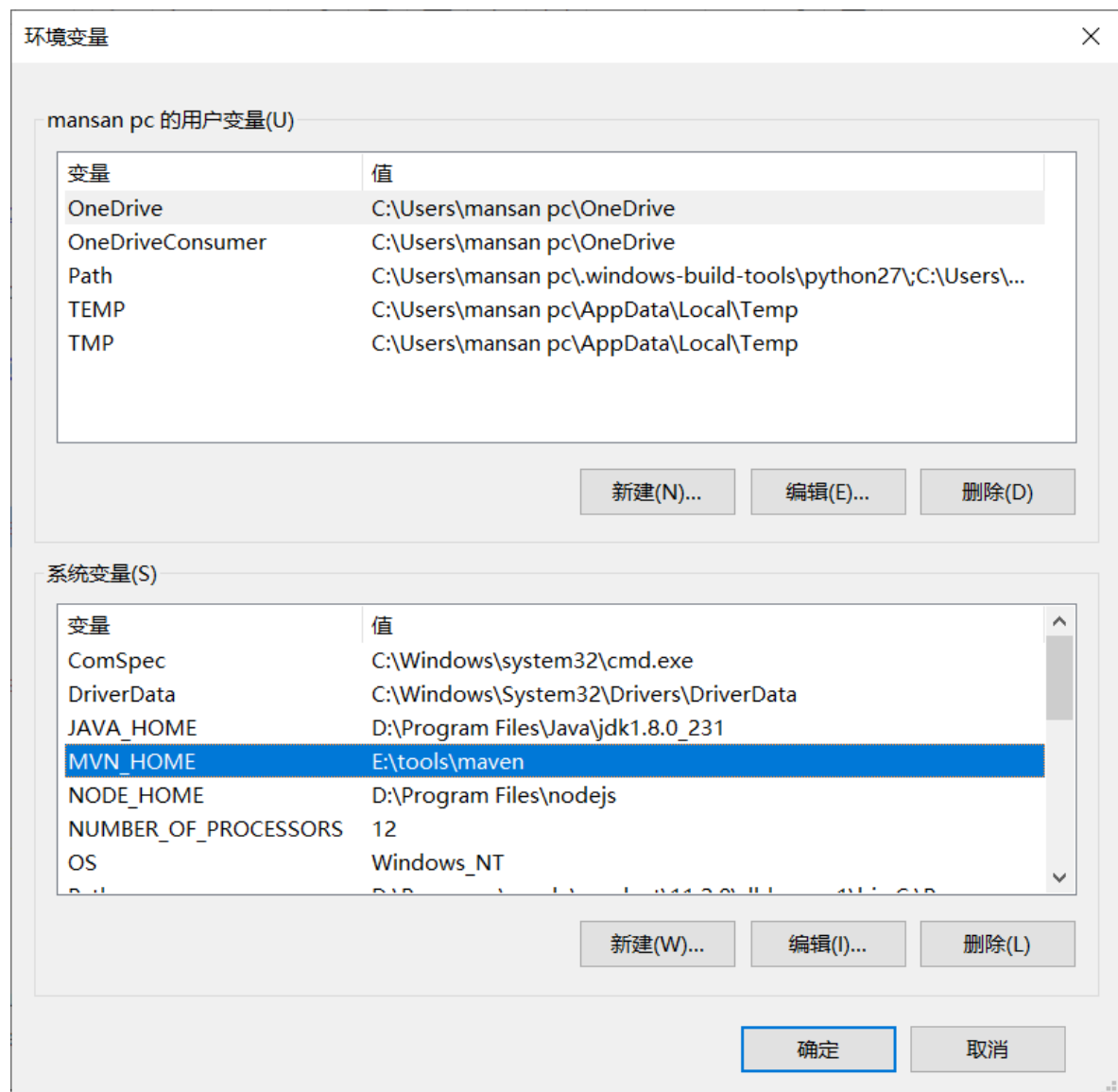
#### 下载Maven

<http://maven.apache.org/download.cgi>

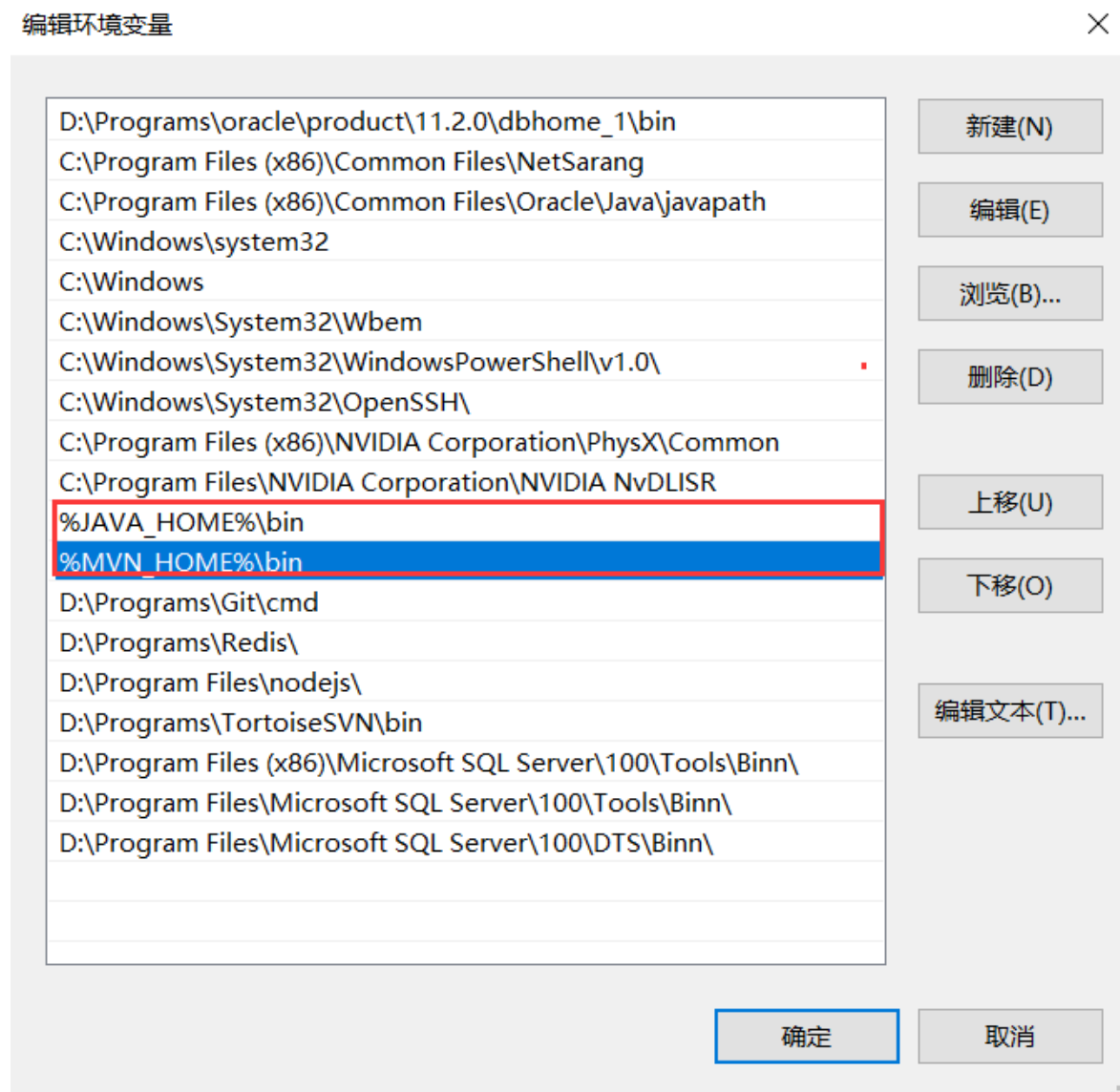
下载完成后，可解压至：d:/maven

#### Maven环境配置 (可选)

增加环境变量 MVN\_HOME



并增加至Path中



## Maven私库指向

在Maven的conf下的setting.xml文件中增加以下配置，如：

```
<mirror>
  <id>redxun_public_maven</id>
  <mirrorOf>redxun_public_maven</mirrorOf>
  <name>redxun_public_maven</name>
  <url>http://site.redxun.cn:8081/nexus/content/groups/public/</url>
</mirror>
<mirror>
  <id>redxun_rdgroup_maven</id>
  <mirrorOf>redxun_rdgroup_maven</mirrorOf>
  <name>redxun_rdgroup_maven</name>
  <url>http://site.redxun.cn:8081/nexus/content/groups/rdgroup/</ur
```

```
1>  
</mirror>
```

```
<profile>  
  <id>nexus</id>  
  <repositories>  
    <repository>  
      <id>nexus</id>  
      <url>http://dev.aps360.cn:8081/repository/maven-public/</url>  
      <releases><enabled>true</enabled></releases>  
      <snapshots><enabled>true</enabled></snapshots>  
    </repository>  
  </repositories>  
  <pluginRepositories>  
    <pluginRepository>  
      <id>nexus</id>  
      <url>http://dev.aps360.cn:8081/repository/maven-public/</url>  
      <releases>  
        <enabled>true</enabled>  
      </releases>  
      <snapshots>  
        <enabled>true</enabled>  
      </snapshots>  
    </pluginRepository>  
  </pluginRepositories>  
</profile>
```

## 1.8. 移动端APP开发

---

NodeJs

<http://nodejs.cn/download/>

Hbuilder

<https://www.dcloud.io/hbuilderx.html>

## 1.9. 缓存服务（可选）

---

开发时，可以选择本地缓存，生产时使用 Redis

<https://redis.io/download>

## 1.10. 消息队列

---

## 1.开发环境准备

开发时可以使用内置MQ，不需要进行安装。

### 1.10.1. ActiveMQ

外置Apache ActiveMQ 时，请安装:

<http://activemq.apache.org/>

### 1.10.2. RabbitMQ

<https://www.rabbitmq.com/>

### 1.10.3. Kafka

<http://kafka.apache.org/>

## 1.11. 文件服务（可选）

---

FastDFS

<https://github.com/happyfish100/fastdfs>

## 1.12. 全文搜索（可选）

---

Elastic Search

<https://www.elastic.co/cn/>

## 2.代码运行

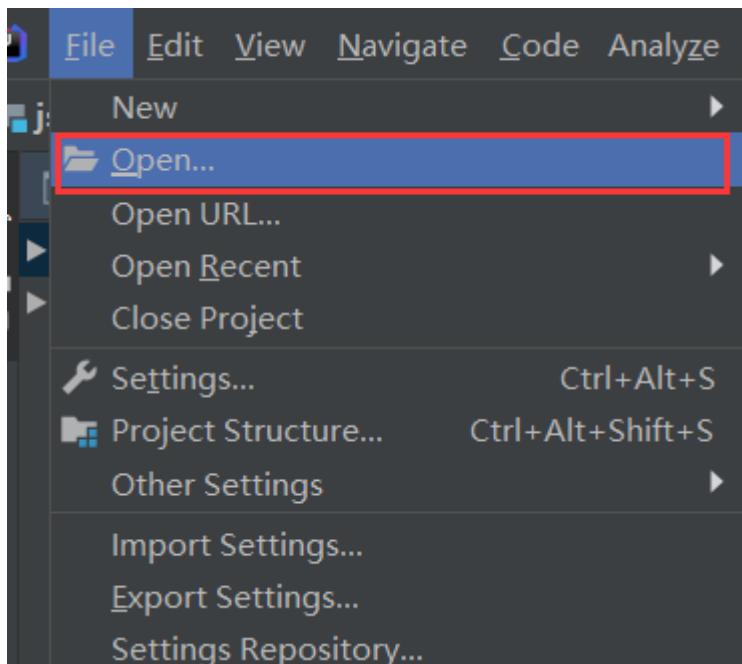
# 代码运行

---

## 1. 解压源代码

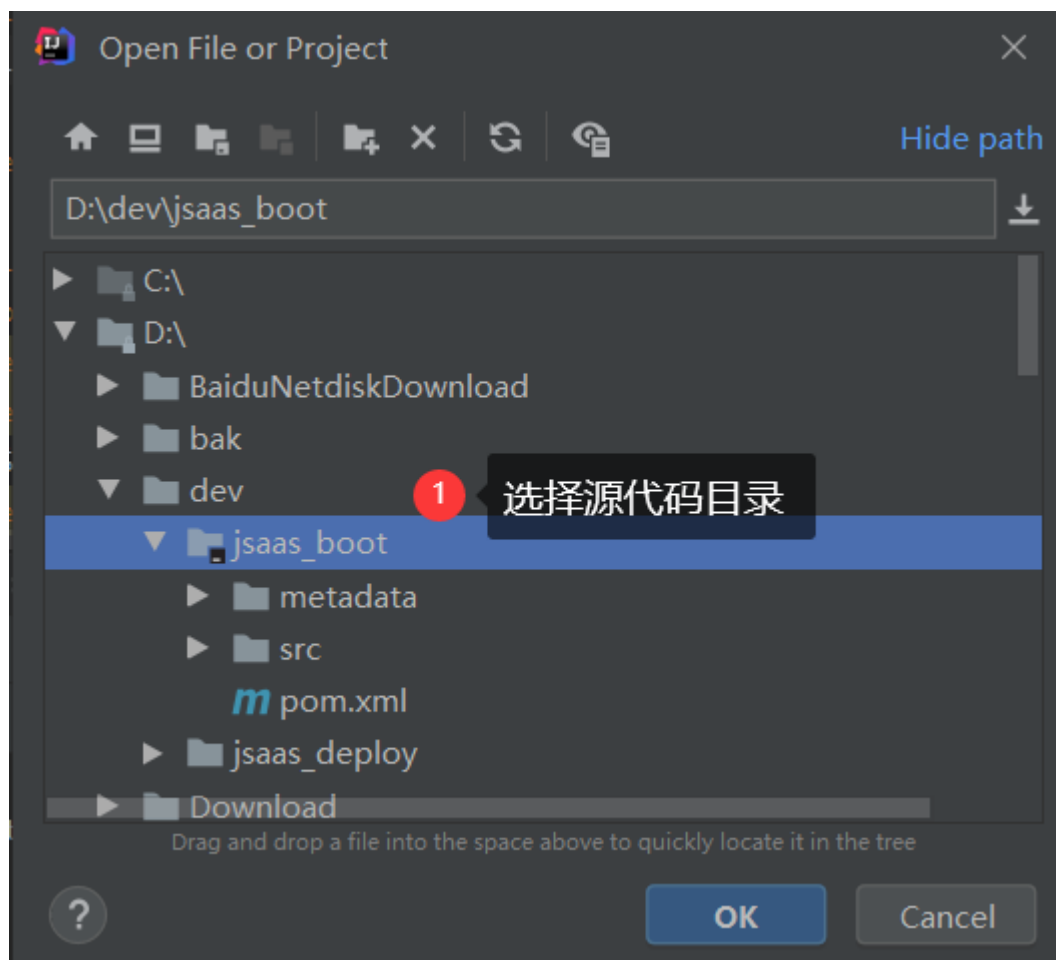
---

如解决至d:/dev/jsaas\_boot目录，如下所示：



## 2. 运行Idea工具，导入项目，选择源代码目录：

---

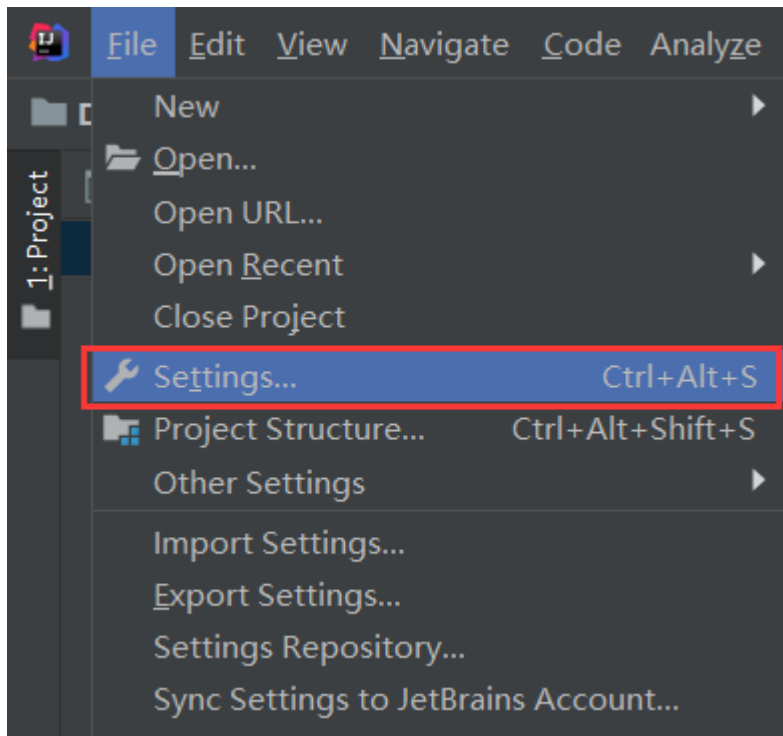


### 3. 设置maven构建环境

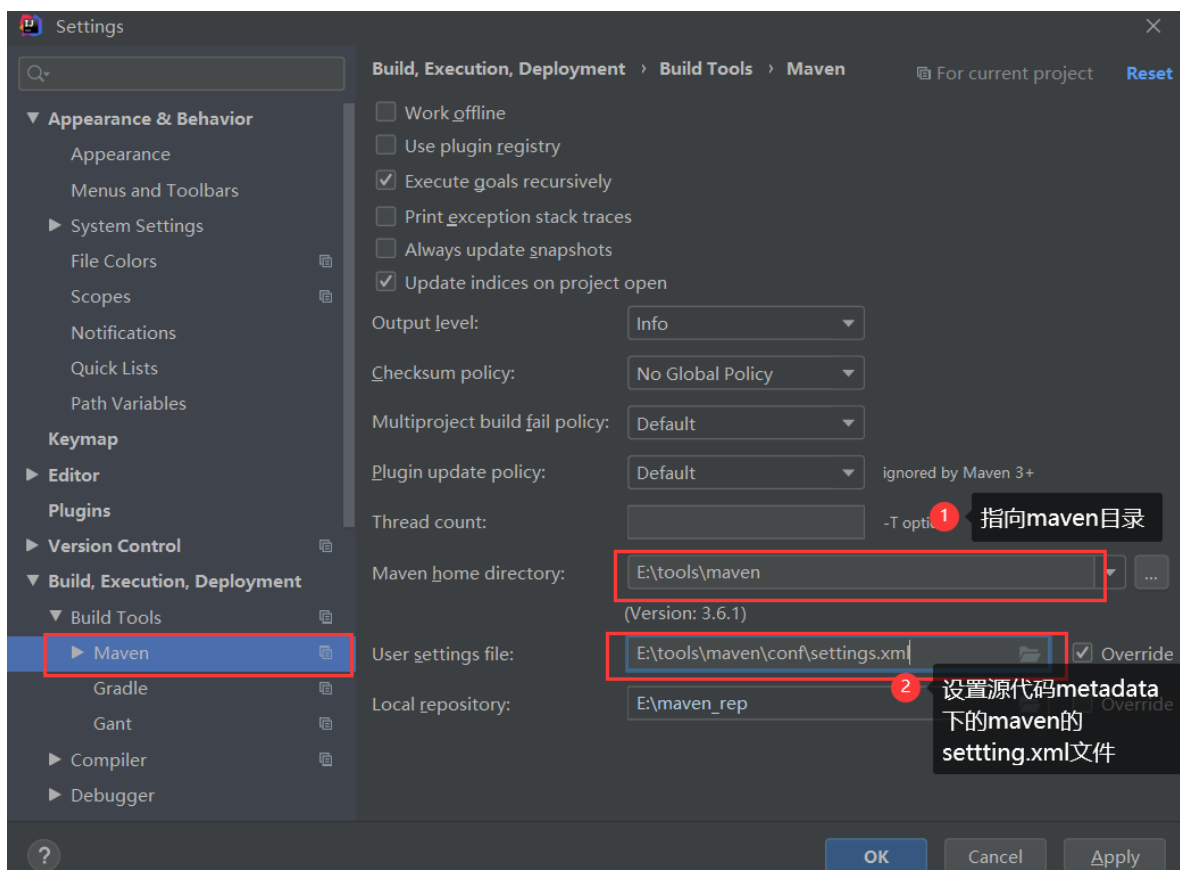
---



## 2. 代码运行

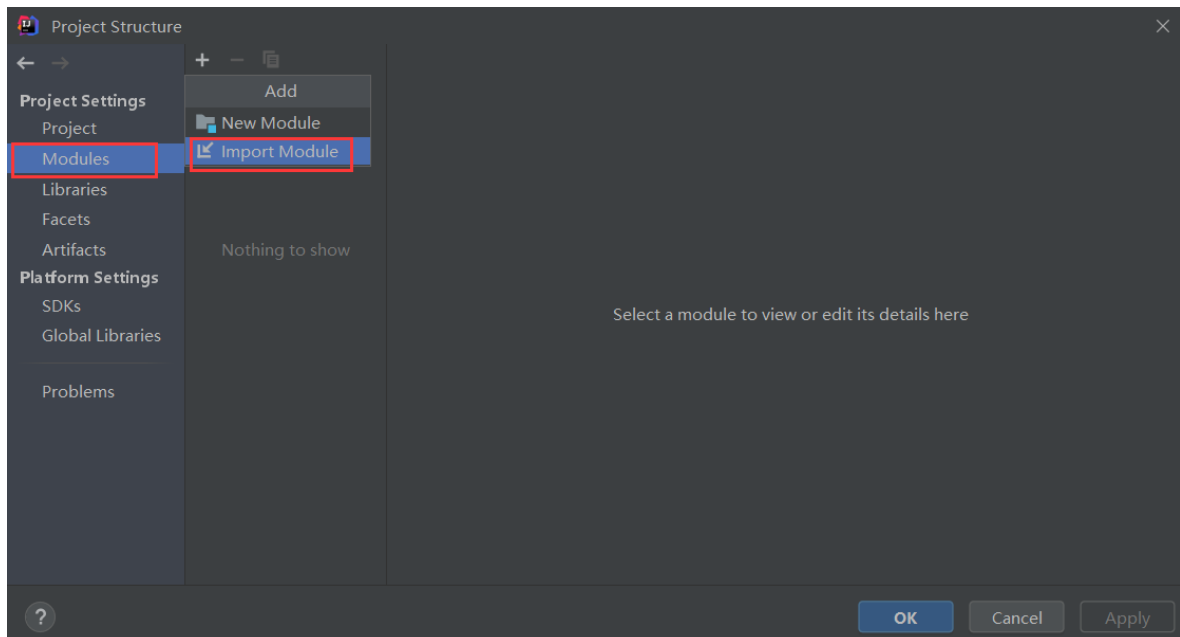
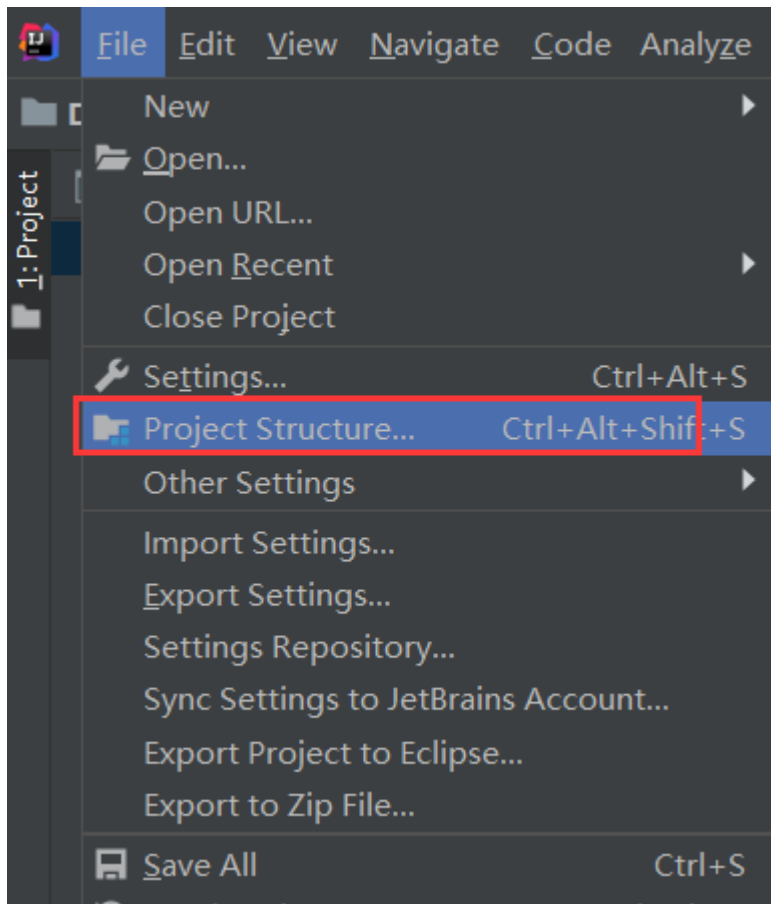


进入maven环境设置

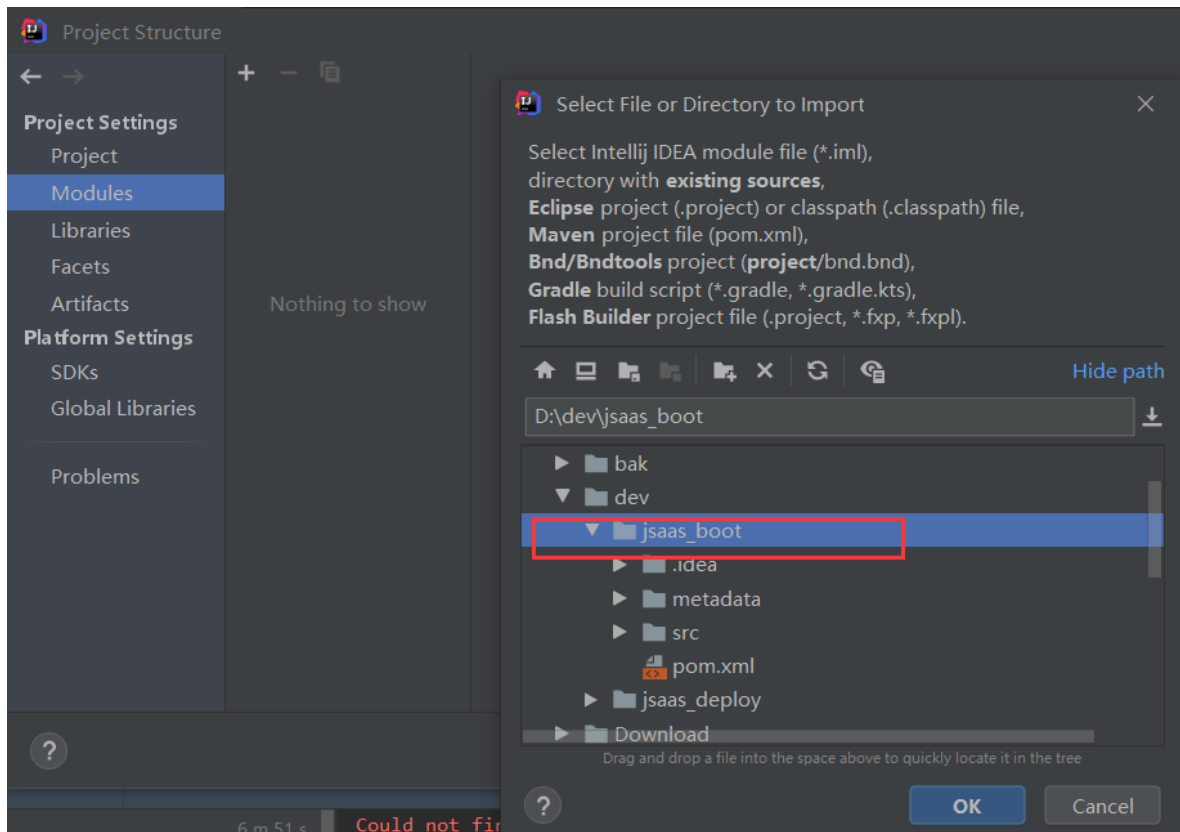


## 4. 把源代码转为maven项目

## 2. 代码运行

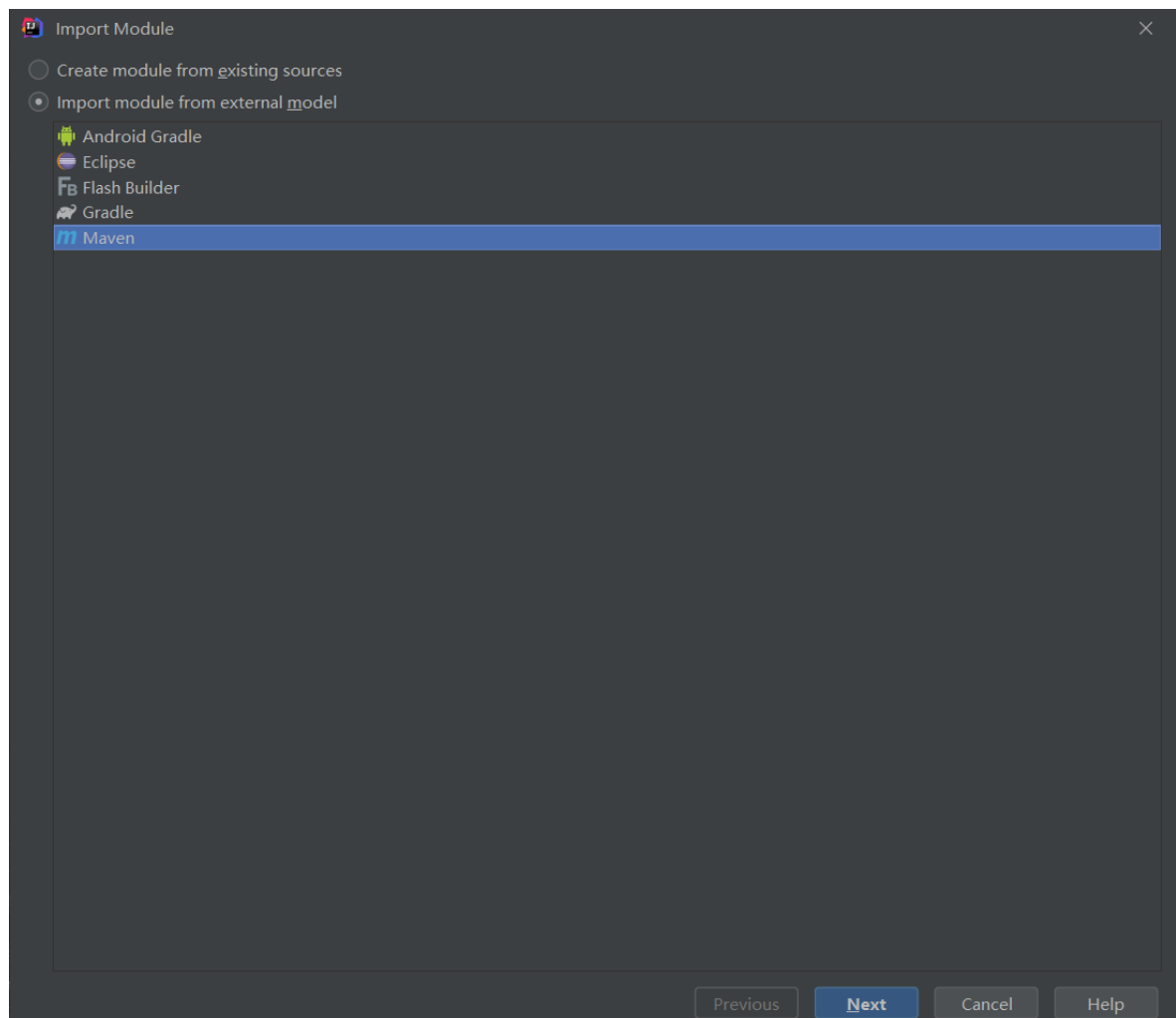


## 2.代码运行



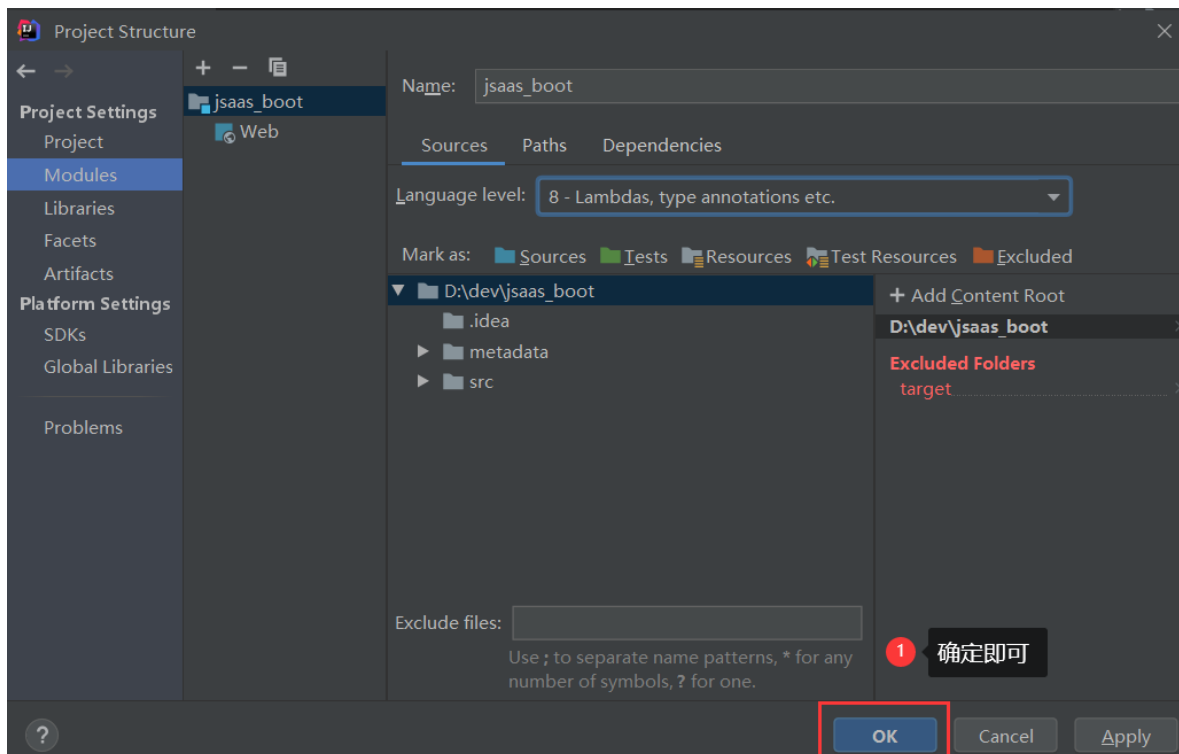
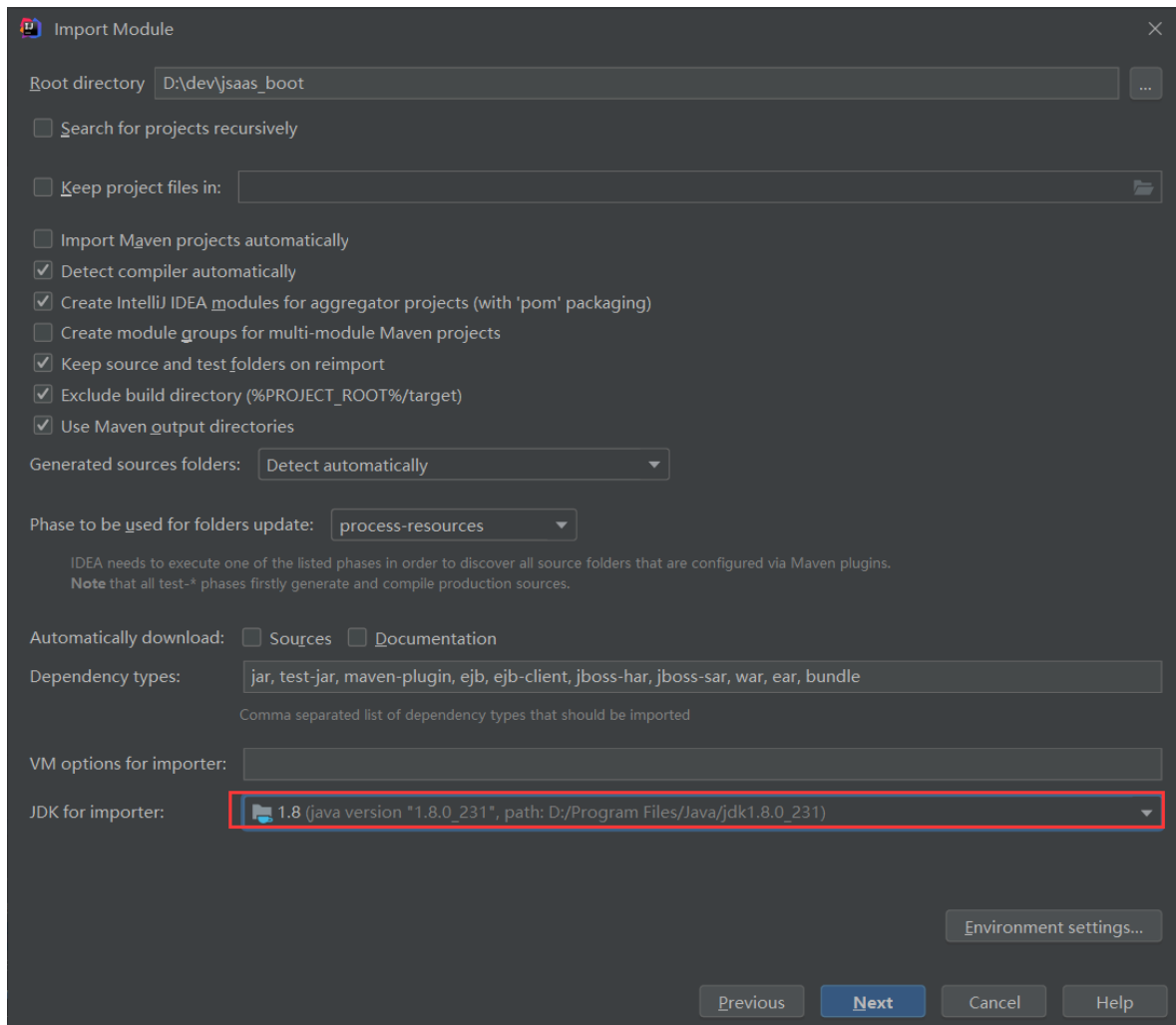
然后一步一步往下即可

## 2.代码运行



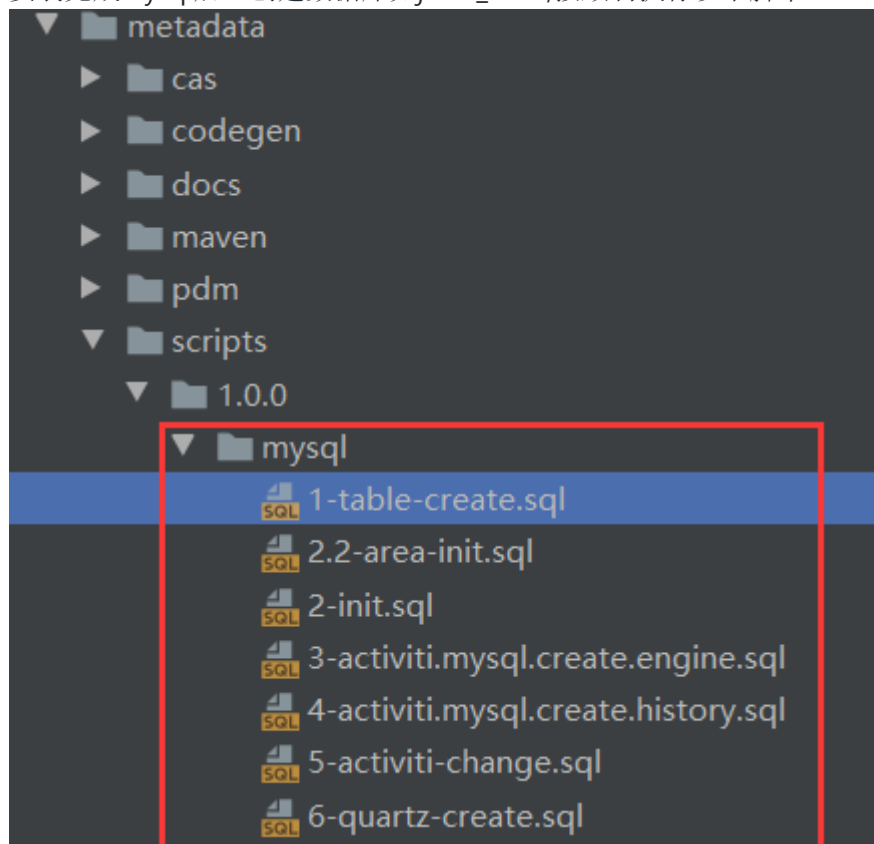
指定JDK

## 2.代码运行



## 5. 配置数据库

安装完成mysql后，创建数据库如jsaas\_boot,按顺利执行以下脚本：



## 6. 修改源代码数据库连接

### mysql 5.7 版本配置

修改src/main/resources/application.yml以下参数配置

```
spring:
  datasource:
    driver-class-name: com.mysql.jdbc.Driver
    url: jdbc:mysql://localhost:3306/jsaas-bootv2?useUnicode=true&characterEncoding=UTF-8&userSSL=false&serverTimezone=GMT%2B8
    username: root
    password: 123456
```

### mysql 8.0 版本配置

mysql 8.0 版本与5.7的版本配置区别主要是在驱动上，需要修改以下两块的配置

## 2. 代码运行

### 1. 修改工程的pom.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.redxun</groupId>
    <artifactId>jsaas_boot</artifactId>
    <version>1.0.0</version>
    <packaging>war</packaging>
    <name>jsaas_boot</name>
    <url>http://maven.apache.org</url>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
        <java.version>1.8</java.version>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>
    <dependencies>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.20</version>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>
```

### 2. 修改application.yml的driver-class-name为以下： com.mysql.cj.jdbc.Driver

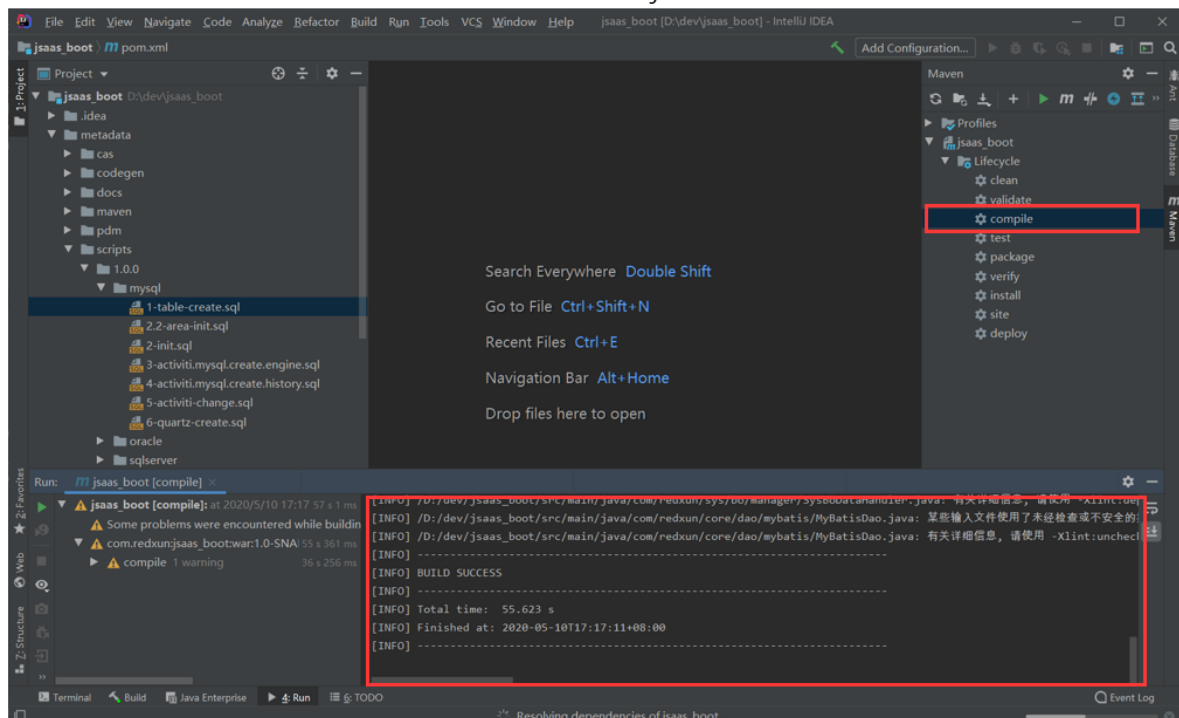
## 7. 运行maven

clean

compile

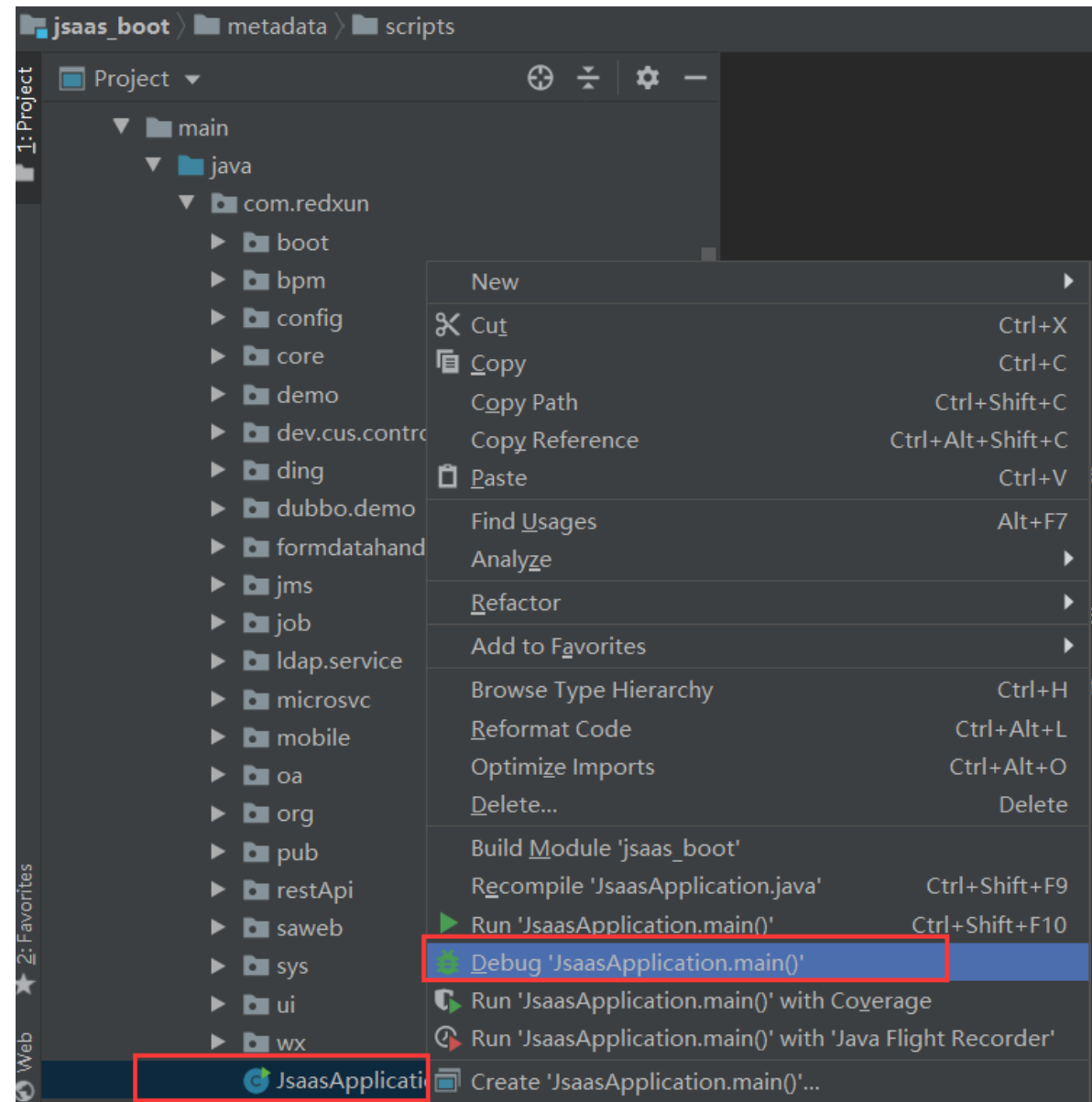
build

在IDEA下可双击右边的maven面板上的对应目标，若jar下载顺利，即可正确运行，如下所示：



## 8. 运行源代码

找到com.redxun.JsaasApplication.java，右键，点击run或debug菜单项，如下：



## 9. 若能成功访问，即可访问：

<http://localhost:8080/jsaas>

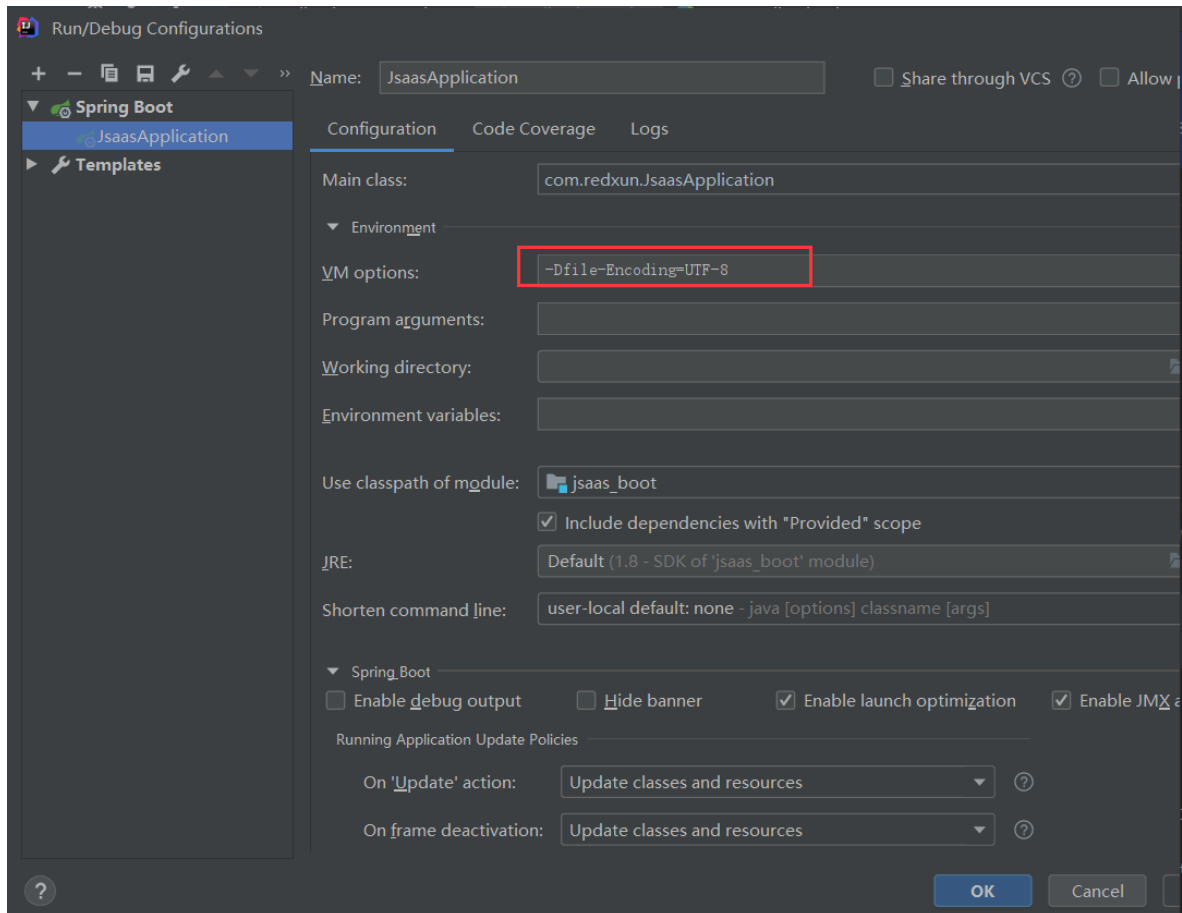
初始化用户：admin  
密码：1

## 10.FAQ



## 2.代码运行

配置运行参数，配置JVM参数，可解决一些运行的表单设计时中文乱码显示的问题，如下：

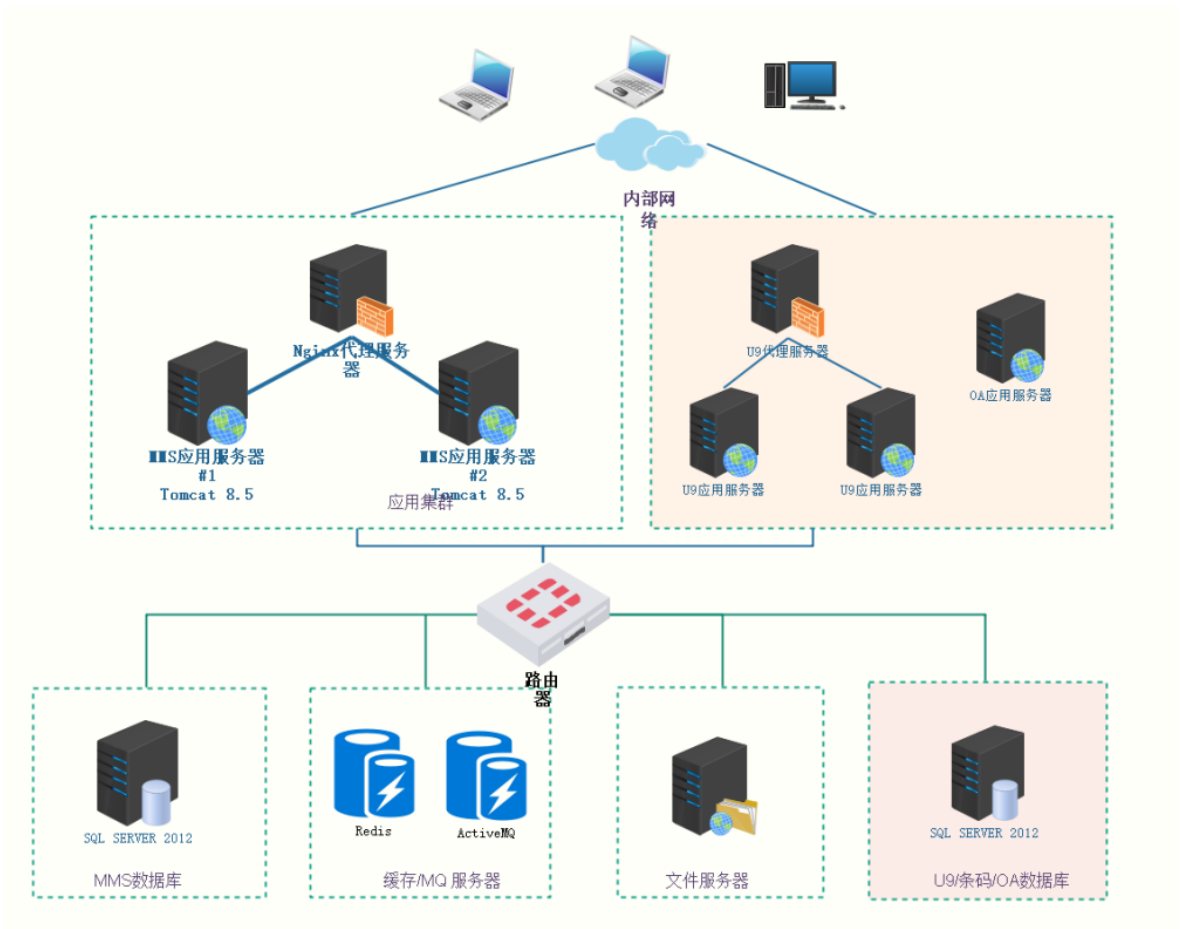


## 3.部署手册

### 部署手册

本文以linux下的部署环境为例，说明平台的环境与安装部署过程

#### 1.部署拓扑图



**【说明】：**  
有背景图部分为现有的功能系统资源。其他部分则为JSAAS系统新部署架构图。

#### 2.环境准备

服务器名	数量	硬件要求	软件环境
前置服务器	1	CPU :2 X 2.4G +	CentOs 7.0 +

服务器名	数量	硬件要求	软件环境
		内存：16G+	Nginx 1.2 +
		硬盘：128G+	
		网卡：千兆网关	
应用服务器	2	CPU :4 X 2.4G +	CentOs 7.0+
		内存：24G+	JDK 1.8+
		硬盘：256G+	Tomcat 8.5+
		网卡：千兆网关	
数据库服务器	1	CPU: 4X2.4G+	Windows Server 2012
		内存：24G+	SQL Server 2012
		硬盘：512G	说明：
		网卡：千兆网关	有条件时加多一台作为双机热备处理。
文件服务器	1	CPU :2 X 2.4G +	CentOs 7.0+
		内存：16G+	说明：
		硬盘：1T	外置一块硬盘作为文件的定时备份
		外置一盘作为文件（1T）备份目录	
		网卡：千兆网关	
缓存服务器、MQ服务器	1	CPU :2 X 2.4G +	CentOs 7.0+
		内存：16G+	Apache ActiveMQ 5.12+
		硬盘：100G	Redis 6.0+
		网卡：千兆网关	

### 3.部署过程

## 3.1. 应用服务器

两台应用服务采用相同的配置，配置过程以下所示。

### 3.1.1.安装JDK

软件安装文件放置于/data/software下，如：

```
[zlgmcsu@MMS-app1 local]$ ls /data/software/
apache-tomcat-8.5.59.zip  jdk-8u261-linux-x64.tar.gz
```

- 切换至目录  
cd /usr/
- 创建jdk的安装目录  
sudo mkdir java
- 授权java目录运行权限  
chmod 777 java
- 解压jdk的安装包至java目录  
sudo tar -zxvf /data/software/jdk-8u261-linux-x64.tar.gz -C /usr/local/java
- 修改环境变量  
sudo vi /etc/profile

```
[zlgmcsu@MMS-app1 java]$ vi /etc/profile

# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

pathmunge () {
    case "${PATH}" in
        *:"$1":*)
            ;;
        *)
            if [ "$2" = "after" ] ; then
```

在该文件的最后面增加以下环境变量配置

```
export JAVA_HOME=/usr/local/java/jdk1.8.0_261
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

- 刷新一下环境变量

```
source /etc/profile
```

- 检查一下环境变量

```
java version
```

```
[zlgmcu@MMS-app1 java]$ java -version
java version "1.8.0_261"
Java(TM) SE Runtime Environment (build 1.8.0_261-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.261-b12, mixed mode)
[zlgmcu@MMS-app1 java]$
```

### 3.1.2. 安装tomcat

安装并解压至 /usr/local/tomcat目录下，按顺序执行以下命令

```
cd /usr/local
sudo mkdir tomcat
sudo chmod 777 tomcat

sudo unzip /data/software/apache-tomcat-8.5.59.zip -d /usr/local/tomcat/
```

重命名文件夹为tomcat85

```
[zlgmcu@MMS-app1 local]$ cd tomcat
[zlgmcu@MMS-app1 tomcat]$ mv apache-tomcat-8.5.59/ tomcat85
```

授权

```
sudo chmod 777 -R tomcat85/
```

设置tomcat的环境变量

```
sudo vi /etc/profile
```

修改以下变量为：

```
export JAVA_HOME=/usr/local/java/jdk1.8.0_261
export CATALINA_HOME=/usr/local/tomcat/tomcat85
export PATH=:$PATH:$JAVA_HOME/bin:$CATALINA_HOME/bin
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

授权bin下的执行权限

```
$ sudo chmod 755 *.sh
```

修改tomcat的连接数

```
sudo vi /usr/local/tomcat/tomcat85/conf/server.xml
```

```
<Connector port="8080" protocol="HTTP/1.1"
            URIEncoding="UTF-8"
            maxThreads="10240"
            acceptThreads="12240"
            maxConnections="10240"
            minSpareThreads="100"
            connectionTimeout="20000"
            redirectPort="8443" />
```

启动tomcat

```
./startup.sh
```

## 设置开机启动

```
sudo chmod +x /etc/rc.d/rc.local
sudo vi /etc/rc.d/rc.local
```

输入以下脚本

```
sudo sh /usr/local/tomcat/tomcat85/bin/startup.sh
```

1. 在tomcat 的catalina.sh中第二行增加

```
export JAVA_HOME=/usr/local/java/jdk1.8.0_261
export CATALINA_HOME=/usr/local/tomcat/tomcat85
export PATH=$PATH:$JAVA_HOME/bin:$CATALINA_HOME/bin
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

- 2、新建/etc/init.d/tomcat文件

```
#!/bin/bash
#chkconfig: 2345 80 90
#description:auto_run
```

```

start(){
    echo -n "Starting Tomcat service:"
    sh /usr/local/tomcat/tomcat85/bin/startup.sh
    echo "tomcat is suceessfully started up"
}

stop(){
    echo -n "Shutting down tomcat: "
    sh /usr/local/tomcat/tomcat85/bin/shutdown.sh
    echo "tomcat is suceessfully shut down."
}

status(){
    numproc=`ps -ef | grep catalina | grep -v "grep catalina" | wc -l`
    if [ $numproc -gt  ]; then
        echo "Tomcat is running..."
    else
        echo "Tomcat is stopped..."
    fi
}

restart(){
    stop
    start
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    status)
        status
        ;;
    restart)
        restart
        ;;
    *)
        echo $"Usage: $0 {start|stop|status|restart}"
        exit
esac

```

### 3、注册服务

```
chkconfig --add tomcat
```

#### 4、查看服务

```
chkconfig --list|grep tomcat
```

#### 5、启动服务

```
service tomcat start
```

## 3.2.缓存服务器

安装redis，在服务器上执行创建目录，并且解压内容并安装

```
$ cd /usr/local/  
$ sudo mkdir redis  
$ wget https://download.redis.io/releases/redis-6.0.8.tar.gz  
$ tar xzf redis-6.0.8.tar.gz  
$ cd redis-6.0.8  
$ make
```

#### 【安装说明】

若在编译过程中存在问题，可能缺少gcc-c++编译器，需要预安装一下，如下：

```
yum -y install gcc-c++
```

重新再删除原来的redis-6.0.8目录，重新再make一下。

若执行还有编译语法的问题，则需要升级一下gcc-cc++.

```
[root@localhost redis-6.0.1]# gcc -v # 查看gcc  
版本  
[root@localhost redis-6.0.1]# yum -y install centos-release-scl # 升级到  
9.1版本  
[root@localhost redis-6.0.1]# yum -y install devtoolset-9-gcc devtoolset-  
9-gcc-c++ devtoolset-9-binutils  
[root@localhost redis-6.0.1]# scl enable devtoolset-9 bash  
以上为临时启用，如果要长期使用gcc 9.1的话：  
[root@localhost redis-6.0.1]# echo "source /opt/rh/devtoolset-9/enable" >  
>/etc/profile
```



## 运行redis

```
$ sudo sh src/redis-server
```

## 测试是否正常

```
$ src/redis-cli
redis> set foo bar
OK
redis> get foo
"bar"
```

## 修改redis的密码 及为后台运行模式

进入redis的安装目录下，找到redis.conf文件，修改以下行为对应的命令。

```
requirepass 123456
```

```
daemonize yes
```

## 修改为开机启动

### 新建redis的配置文件

```
vim /etc/init.d/redis
```

并修改内容为：

```
#!/bin/sh
#chkconfig: 2345 90 10
#description: Redis is a persistent key-value database
PATH=/usr/local/bin:/sbin:/usr/bin:/bin

REDISPORT=6379
EXEC=/usr/local/redis/redis-6.0.8/src/redis-server
REDIS_CLI=/usr/local/redis/redis-6.0.8/src/redis-cli
PIDFILE=/var/run/redis_6379.pid
CONF="/usr/local/redis/redis-6.0.8/redis.conf"

case "$1" in
    start)
        if [ -f $PIDFILE ]
        then
            echo "$PIDFILE exists, process is already running or crashed"

```

```

else
    echo "Starting Redis server..."
    $EXEC $CONF
fi
if [ "$?"="0" ]
then
    echo "Redis is running..."
fi
;;
stop)
    if [ ! -f $PIDFILE ]
    then
        echo "$PIDFILE does not exist, process is not running"
    else
        PID=$(cat $PIDFILE)
        echo "Stopping ..."
        $REDIS_CLI -p $REDISPORT SHUTDOWN
        while [ -x ${PIDFILE} ]
        do
            echo "Waiting for Redis to shutdown ..."
            sleep 1
        done
        echo "Redis stopped"
    fi
    ;;
restart|force-reload)
    ${0} stop
    ${0} start
    ;;
*)
    echo "Usage: /etc/init.d/redis {start|stop|restart|force-reload}" >&2
    exit 1
esac

```

修改以上配置为实际生产环境的参数

```

REDISPORT=6379
EXEC=/usr/local/redis/redis-6.0.8/src/redis-server
REDIS_CLI=/usr/local/redis/redis-6.0.8/src/redis-cli
PIDFILE=/var/run/redis_6379.pid
CONF="/usr/local/redis/redis-6.0.8/redis.conf"

```

## 设置权限

```
chmod 755 /etc/init.d/redis
```

## 启动测试

```
/etc/init.d/redis start
```

启动成功后会提示如下信息

```
Starting Redis server...
Redis is running...
```

## 设置为开机启动

```
chkconfig --add /etc/init.d/redis
chkconfig redis on
```

【说明】：

查看第一步的配置文件信息，是否和第二步的文件信息一致，vim粘贴会出现粘贴不全的问题

## 3.3. MQ服务器

<https://blog.csdn.net/zhangbinch/article/details/105537294>

## 3.4. 文件服务器

多系统共用一个文件目录，因此需要部署一文件共享服务器给两台应用服务器进行附件的上传与下载。使用的是NFS服务，具体的原理请参考以下文章介绍。

<https://blog.csdn.net/aixiaoyang168/article/details/83782336>

## NFS 服务安装

### 软件安装

```
sudo yum install -y nfs-utils rpcbind
```

### 配置NFS

```
sudo vi /etc/exports
/data/mms_files 192.168.0.52/24(rw,sync,insecure,no_subtree_check,no_root_squash)
```

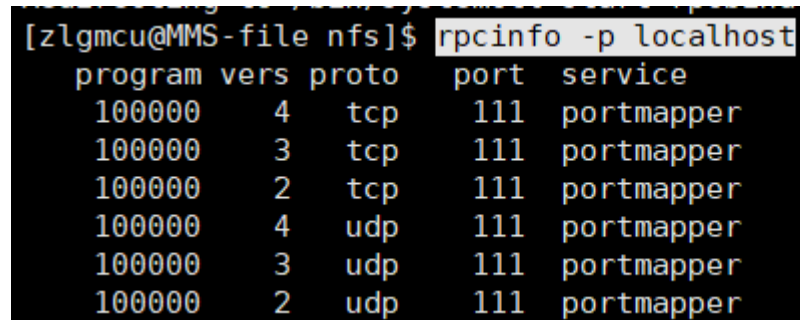
```
/data/mms_files 192.168.0.53/24(rw, sync, insecure, no_subtree_check, no_root_squash)
```

### 启动rpcbind服务

```
sudo service rpcbind start
```

【查看服务】

rpcinfo -p localhost



```
[zlgmcu@MMS-file nfs]$ rpcinfo -p localhost
  program vers proto  port  service
    100000   4   tcp    111  portmapper
    100000   3   tcp    111  portmapper
    100000   2   tcp    111  portmapper
    100000   4   udp    111  portmapper
    100000   3   udp    111  portmapper
    100000   2   udp    111  portmapper
```

### 启动nfs服务

```
sudo service nfs start
```

### 检查rpc服务是否起来

```
rpcinfo -p localhost
```

```
[zlgmcu@MMS-file nfs]$ rpcinfo -p localhost
```

program	vers	proto	port	service
100000	4	tcp	111	portmapper
100000	3	tcp	111	portmapper
100000	2	tcp	111	portmapper
100000	4	udp	111	portmapper
100000	3	udp	111	portmapper
100000	2	udp	111	portmapper
100024	1	udp	43251	status
100024	1	tcp	58470	status
100005	1	udp	20048	mountd
100005	1	tcp	20048	mountd
100005	2	udp	20048	mountd
100005	2	tcp	20048	mountd
100005	3	udp	20048	mountd
100005	3	tcp	20048	mountd
100003	3	tcp	2049	nfs
100003	4	tcp	2049	nfs
100227	3	tcp	2049	nfs_acl
100003	3	udp	2049	nfs
100003	4	udp	2049	nfs
100227	3	udp	2049	nfs_acl
100021	1	udp	58762	nlockmgr
100021	3	udp	58762	nlockmgr
100021	4	udp	58762	nlockmgr
100021	1	tcp	42552	nlockmgr
100021	3	tcp	42552	nlockmgr
100021	4	tcp	42552	nlockmgr

### 两台应用服务器挂载附件文件夹

安装nfs客户端

```
sudo yum install nfs-utils
```

创建本地目录，并挂至文件服务器目录

```
sudo mount 192.168.0.55:/data/mms_files /data/mms_files
```

授权目录访问权限

```
sudo chmod 777 /data/mms_files
```

## 3.5. 前置服务器nginx

nginx安装过程忽略，以下主要是说明如何配置nginx.conf的配置文件

```

upstream mms {
    ip_hash;
    server 192.168.0.52:8080 weight=1;
    server 192.168.0.53:8080 weight=1;
}

server {
    listen      80;
    server_name localhost;

    #charset koi8-r;

    #access_log  logs/host.access.log  main;

    ## ...
    location / {
        proxy_set_header Host      $host;
        proxy_pass http://mms;
        client_max_body_size 100m;
    }
}

```

## 4. 部署应用

把mms程序包部署至两台应用服务器上的目录tomcat85/webapps目录下，相应修改以下运行的配置，也可在打包war前把配置设置为正确的环境。

修改mms/WEB-INF/classes/resources/application.yml的以下数据库连接地址：

```

driver-class-name: com.microsoft.sqlserver.jdbc.SQLServerDriver
url: jdbc:sqlserver://192.168.0.156:1433;DatabaseName=MMS_DEV;integratedSecurity=false;SelectMethod=cursor;
username: test
password: abcabc

```

修改mms/WEB-INF/classes/resources/application.properties以下配置

```

# 修改缓存类型为redis
server.cache.cache-type=redis
# 配置redis连接信息
spring.redis.host=192.168.0.53
spring.redis.port=6379

```

```

spring.redis.password=#mms#001
spring.redis.timeout=5000
spring.redis.database=0
spring.session.store-type=redis
# 发邮件配置
mail.host=aps360.cn
mail.port=25
mail.nickName=MMS Platform
mail.username=regservice@aps360.cn
mail.password=reg#223
mail.ssl=false

# 数据库连接数配置

db.filters=stat
db.name=\u7CFB\u7EDF\u6570\u636E\u6E90
db.type=mssql
db.minimumConnectionCount=5
db.maximumConnectionCount=5000
db.maxWait=60000
db.timeBetweenEvictionRunsMillis=60000
db.minEvictableIdleTimeMillis=30000
db.validationQuery=SELECT 1 from act_ge_property
db.testWhileIdle=true
db.testOnBorrow=false
db.testOnReturn=false
db.poolPreparedStatements=true
db.maxPoolPreparedStatementPerConnectionSize=20

# Ureport的上传地址配置
ureport.fileStoreDir=/data/mms_files/ureportfiles
ureport.disableHttpSessionReportCache=false
ureport.disableFileProvider=true
ureport.debug=true

```

## 运行tomcat应用

切换至应用服务下的tomcat的bin目录

```

cd /usr/local/tomcat/tomcat85/bin
sudo sh startup.sh

```

## 访问应用

mms.zlg.com/mms/loing.jsp

默认账号admin,密码:1

**【说明】**

mms.zlg.com为前置服务器的映射地址。



# 开发方案

# 外部接口调用方案