# 利用jmeter java sample端口转发实现对远程数据库的压力测试

### 1 需求背景

对数据库进行压力测试时，需要模拟同一时间大量的sql请求发送，借助于jmeter的 sql请求可以实现，但是对RDS来说，需要进行端口映射(跳板机)访问远程数据库,对于单线程手工测试来说，可以直接通过CRT进行端口跳转访问，但是对于多线程并发访问时，各个线程都需要开启端口映射，此时CRT无法满足，借助于第三方插件，通过定制 jmeter脚本，可以帮助我们实现测试目标。

### 2 工具/包

#### 2.1 Apache-jmeter

Jmeter支持java语言，是一个开源的性能测试工具，通过扩展，可以实现非常丰富的功能，本讲义中的版本是2.11，jmeter的安装配置过程略

#### 2.2 eclipse

在eclipse中写java sample插件，eclipse的安装配置过程略

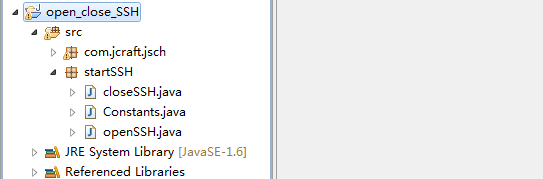
#### 2.3 Com.jcraft.jsch

一个开源的第三方JAR包，主要利用它来进行端口映射

### 3 插件开发

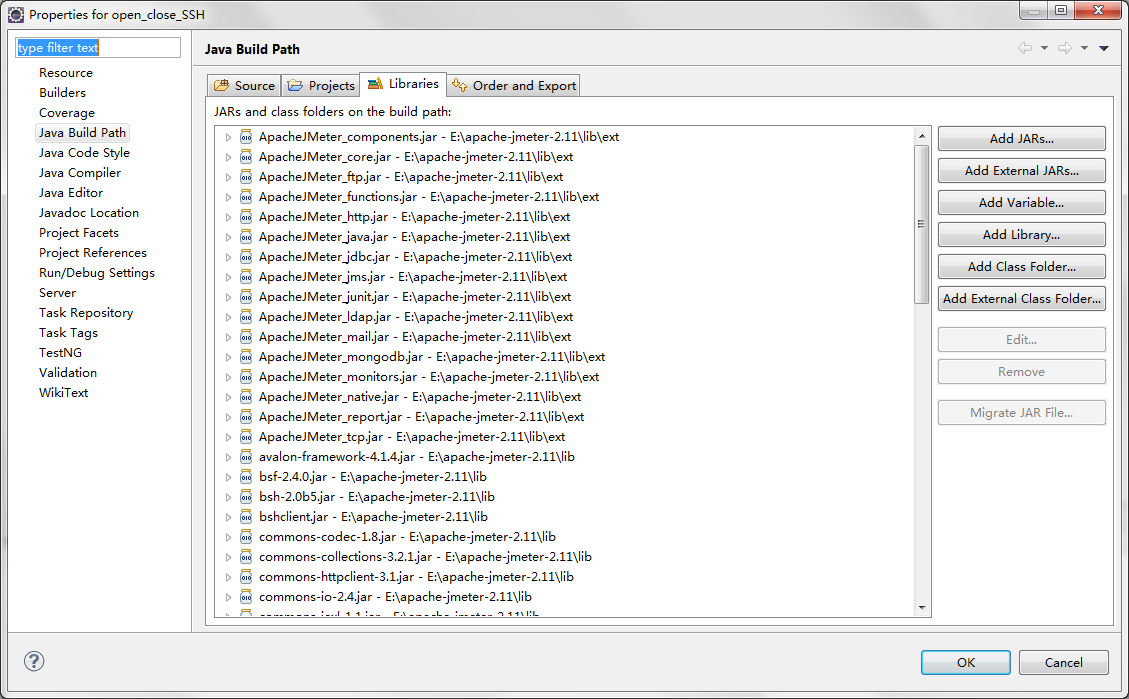
#### 3.1 新建工程

新建一个java工程，将下载下来的Com.jcraft.jsch 源码导入，并新建一个package，用于编写java sample，结构如图所示。



#### 3.2 配置buildpath

将%JMETER\_HOME%\lib 及 %JMETER\_HOME%\lib\ext 导入



#### 3.3 Constants类

该类主要配置运行参数，例如数据库连接，用户名，数据库密码，端口号等，如果这些变量基本是不变的话可以写在Constants里。

/\*Constant.java\*/

**package** startSSH;

**public** **class** Constants {

**static** **final** String *driver*="com.mysql.jdbc.Driver";

**static** **final** String *connServer*="jdbc:mysql://localhost:12306/";

**static** **final** String *dataBase*="ccms\_qinjiaoshou";

**static** **final** String *userName*="\*\*\*\*\*\*\*";

**static** **final** String *passWord*="\*\*\*\*\*\*\*";

**static** **final** String *ssh\_server*="\*\*\*\*\*\*";

**static** **final** **int** *ssh\_server\_port*=22;

**static** **final** String *ssh\_loginName*="yang.yang";

**static** **final** String *ssh\_loginPassword*="\*\*\*\*\*\*\*\*\*\*\*";

**static** **final** **int** *local\_port*=12306;

**static** **final** **int** *remote\_port*=3306;

**static** **final** String *remote\_server*="\*\*\*\*\*\*";

**static** **final** String *ZH\_Encoding*="?useUnicode=true&characterEncoding=utf8&autoReconnect=true&allowMultiQueries=true";

}

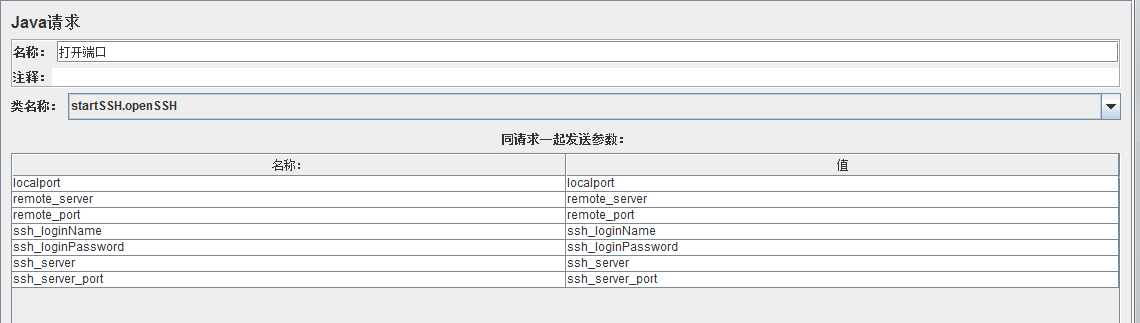
#### 3.4 openSSH类

在openSSH中，定义了参数以及默认值，并开启端口转发，以及由于环境问题，这些参数并不是固定的，这种写死在代码中的方式就不灵活，JMETER 提供了读取参数的方法，参见

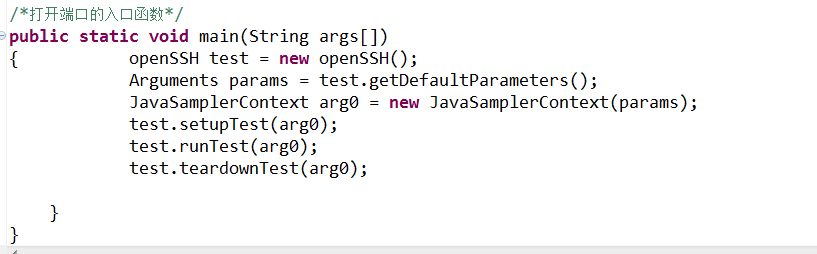
/\*openSSH.java\*/



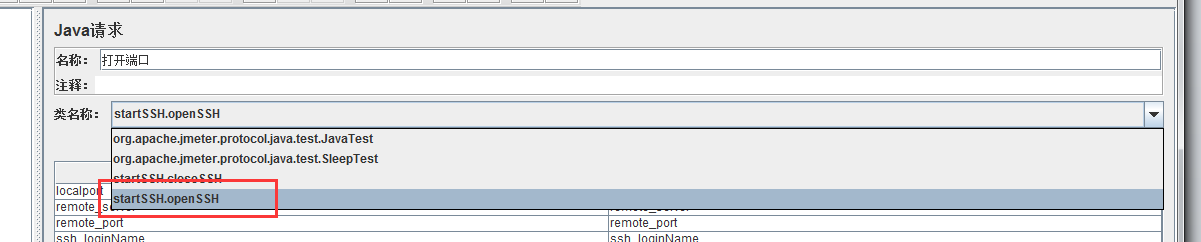
如图



入口函数



入口函数是为了让jmeter找到对应的函数入口，见图



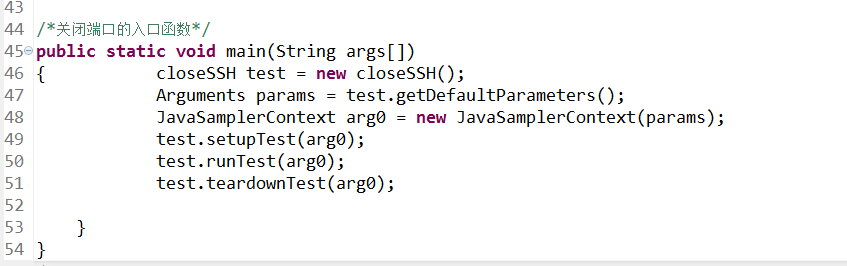
注意，端口应该作参数化处理，多个线程占用一个端口会出现连接失败的问题。一个SQL请求结束后，应释放该线程占用的端口资源，需要将该线程已占用的端口关闭。

#### 3.5 closeSSH类

该类主要用于释放线程占用的端口

/\*closeSSH.java\*/

入口函数:

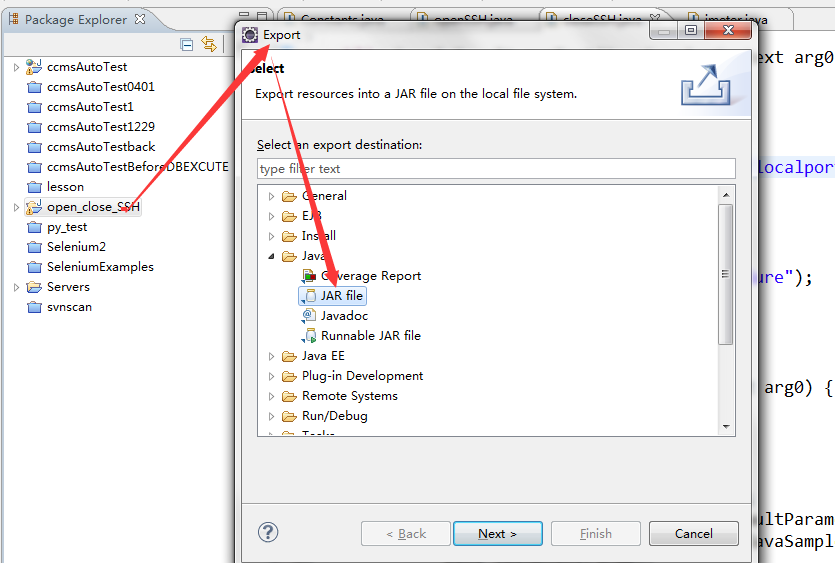




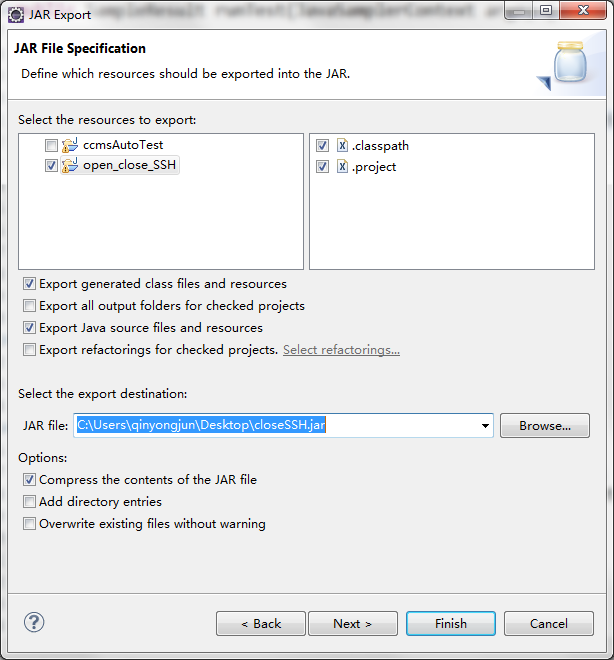
### 4 连接测试

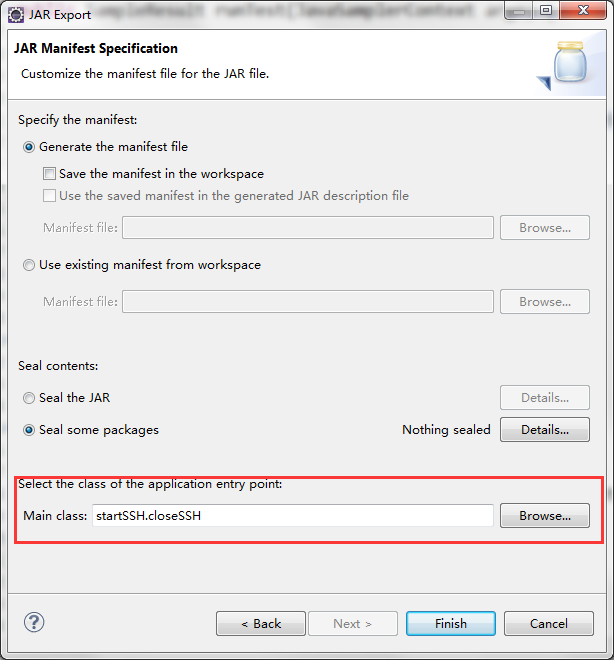
#### 4.1 Jar包导出

分别将项目导出为 JAR包，导出 2次 ，命名为openSSH.jar 和 closeSSH.jar,分别用于开启端口sample 以及关闭关口sample

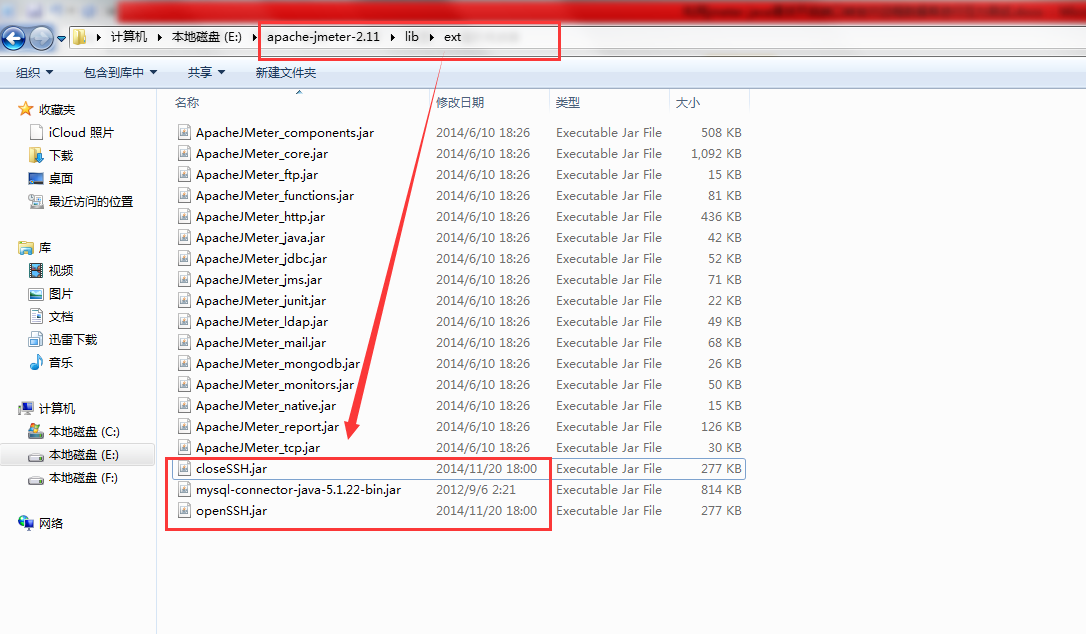


导出时注意选择对应的入口函数



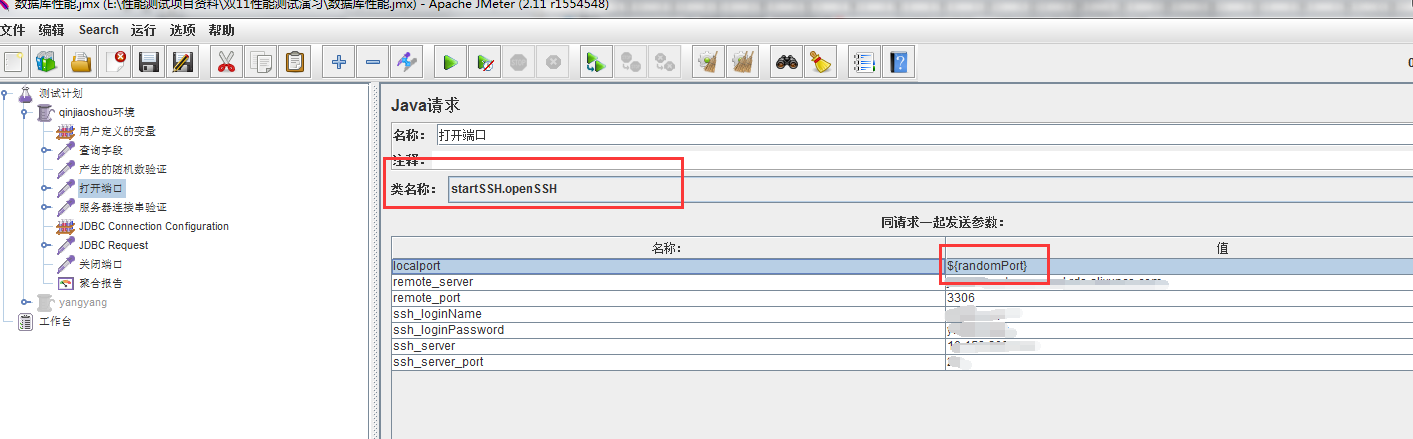


将2个 jar文件 以及 mysql jdbc jar 放到%JMETER\_home%\lib\ext目录下，然后以admin身份运行jmeter。

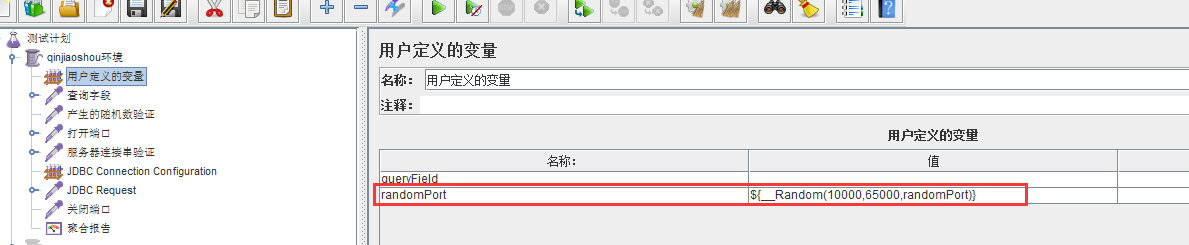


#### 4.2 java sample

在Jmeter中 新建线程组，定义一个用户定义变量以保存随机端口号，新建java sample

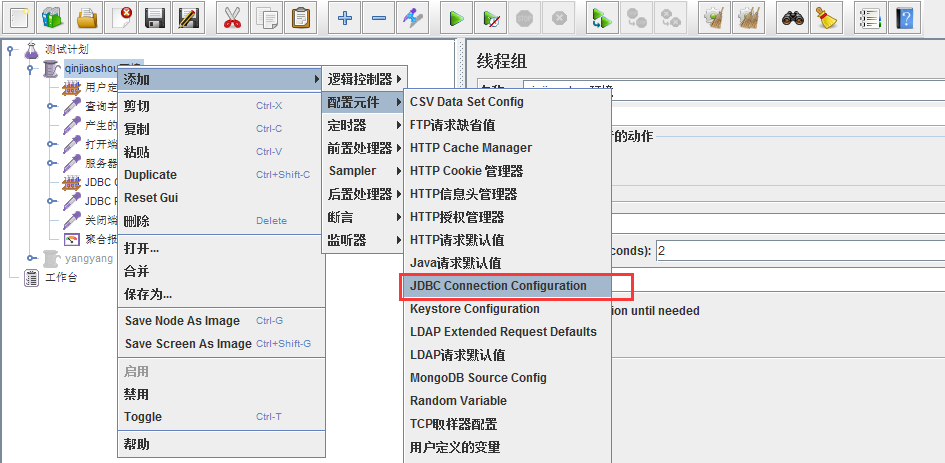


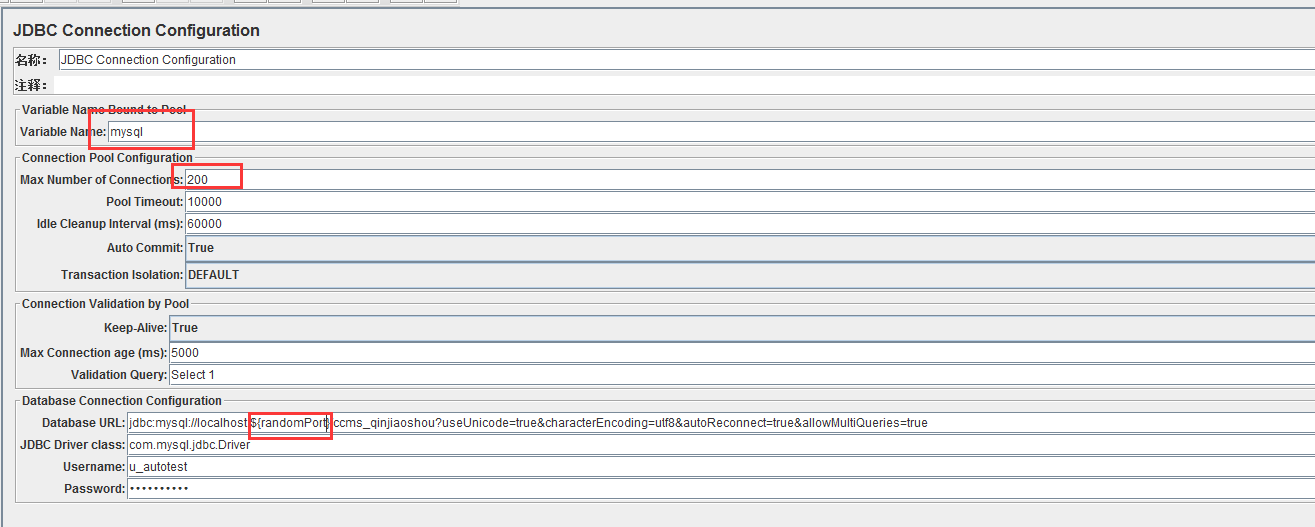
该 java请求用于打开端口，选择类名称，填入适当的参数，注意localport那一栏填写随机值。



#### 4.3 sql sample

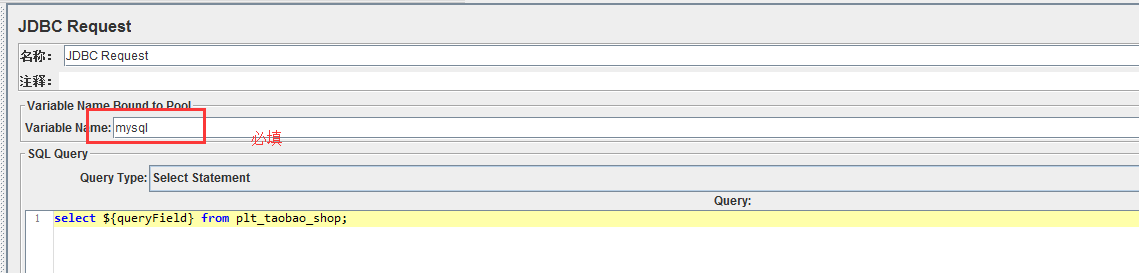
首先配置JDBC连接



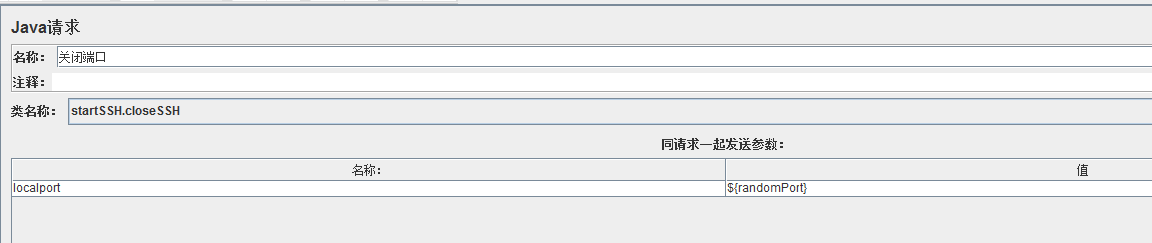


注意：Variable Name不能为空；最大连接数设置要>jmeter线程数，端口号用随机数替代

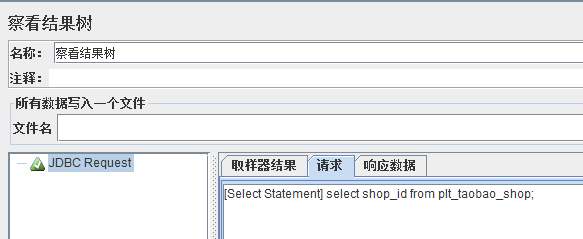
配置完jdbc连接后，新建sql请求就可以了

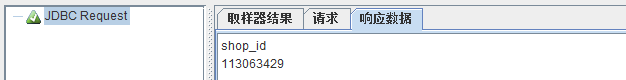


最后加上关闭端口的java sample



开始测试，注意观察jmeter console日志及sql结果树







可见打开了57734的端口映射,而且正确进行了查询，成功实现了模拟SSH进行端口转发的需求。

谢谢，如有不足之处,请与

yongjun.qin@shuyun.com 联系