

[返回首页](#)

## 使用tcp连接

本文将给出简单的示例，让您能在您的 Java 程序中通过tcp连接到 Shrek 服务器。

## Shrek服务器配置

修改web.xml并重新启动Shrek服务器。

```
<!-- TCP服务端口 -->
<context-param>
    <param-name>tcpPort</param-name>
    <param-value>20001</param-value>
</context-param>

<!-- TCP心跳检测,单位 秒 , -1表示未配置, 0-15系统将默认为15秒-->
<context-param>
    <param-name>readerIdleTime</param-name>
    <param-value>15</param-value>
</context-param>

<!-- TCP最大连接数限制 -->
<context-param>
    <param-name>maxConnectSize</param-name>
    <param-value>512</param-value>
</context-param>
```

## 导入 jar 包

### 1、相关jar包

在您的项目中通过tcp连接到 Shrek 服务器除了[连接到服务器](#)中需要导入的jar包外，还需如下 jar 包：

- protobuf-java.jar
- netty.jar

### 2、获取jar包

对于金蝶云苍穹和星瀚等研发人员，这些 jar 包直接在 biz-evn-dev 中获取。

### 3、引入jar包

将上述的jar包引入到您的项目中。如果您的项目是使用 gradle 进行项目管理，您可以在gradle中添加如下依赖项：

```
def trd = 'C:/bos-evn-dev/mservice/lib/trd'
def boscorelib = 'C:/bos-evn-dev/mservice/lib/bos'

dependencies {
    compile fileTree(dir: trd, include: 'kotlin-runtime-*.jar')
    compile fileTree(dir: trd, include: 'kotlin-stdlib-*.jar')
    compile fileTree(dir: trd, include: 'jackson-*.jar')
    compile fileTree(dir: trd, include: 'lz4-*.jar')
    compile fileTree(dir: trd, include: 'netty-*.jar')
    compile fileTree(dir: trd, include: 'protobuf-java*.jar')
    compile fileTree(dir: boscorelib, include: 'bos-olap-client-*.jar')
}
```

## 通过tcp连接到 Shrek 服务器

您可以参照下边的示例如何在您的程序连接到 Shrek 服务器。

```
String cubeName = "YourFirstCube";
OlapConnection olapConn = dataSources.getConnection(cubeName); // 通过数据库连接池获取连接
```

下面是具体如何获取连接shrek的代码：

```
static OlapPooledDataSources dataSources = getDataSources(cubeName); // 初始化连接池
private static OlapPooledDataSources getDataSources(String cubeName) {
    OlapPooledStringBuilder conBuilder = new OlapPooledStringBuilder("");
    conBuilder.setProvider("kingdee.olap.Shrek.tcp");
    // 本用例连接为本地127.0.0.1的服务器，本IP、端口根据实际情况修改
    conBuilder.setDataSource("127.0.0.1:20001");
    if (!cubeName.isEmpty()) {
        conBuilder.setInitialCatalog(cubeName); // 设置即将连接或创建的Cube
    }
    conBuilder.setMinPoolSize(2); // 设置连接池初始化连接数
    conBuilder.setMaxPoolSize(2); // 设置连接池最大连接数
    conBuilder.setMaxWaitTime(3000); // 设置最大等待时间
    // 设置用户名及密码，可向直接上级申请
    conBuilder.setUserName("admin");
    conBuilder.setPassword("Olap@2018");
    return new OlapPooledDataSources(conBuilder);
}
```

## 使用tcp连接

对于tcp 连接的相关操作有提交、回滚，您可以这样使用tcp连接：

① autoCommit默认为true，此时为自动提交，可以将当成http连接使用，与http连接的相关操作无异；

② 通过 `olapConn.setAutoCommit(false)` 对事务进行手动操作，支持多个命令在同一个事务中，操作执行完需要进行手动提交与回滚：

```

olapConn.setAutoCommit(false); // 设置自动提交为false
// 命令1
SaveCommandInfo saveCommandInfo1 = new SaveCommandInfo();
saveCommandInfo1.setDimensions("accounts", "years", "terms", "orgs");
saveCommandInfo1.setMeasures("FMONEY");
OlapCommand cmd1 = new OlapCommand(olapConn, saveCommandInfo1);
OlapDataWriter writer1 = cmd1.CreateWriter();
Object[] newRow1 = new Object[]{10.1, "account1", "year2019", "term1", "org1"};
writer1.setValues(newRow1);
writer1.flush();
// 命令2
SaveCommandInfo saveCommandInfo2 = new SaveCommandInfo();
saveCommandInfo2.setDimensions("accounts", "years", "terms", "orgs");
saveCommandInfo2.setMeasures("FMONEY");
OlapCommand cmd2 = new OlapCommand(olapConn, saveCommandInfo1);
OlapDataWriter writer2 = cmd2.CreateWriter();
Object[] newRow2 = new Object[]{30, "account1", "year2019", "term1", "org2"};
writer2.setValues(newRow2);
writer2.flush();
olapConn.rollback(); // 进行回滚
//查询不到[account1, year2019, term1, org1, 10.1]、 [account1, year2019, term1, org2, 30]记录

// 再次插入一笔记录
SaveCommandInfo saveCommandInfo = new SaveCommandInfo();
saveCommandInfo.setDimensions("accounts", "years", "terms", "orgs");
saveCommandInfo.setMeasures("FMONEY");
OlapCommand cmd = new OlapCommand(olapConn, saveCommandInfo1);
OlapDataWriter writer = cmd.CreateWriter();
Object[] newRow = new Object[]{30, "account1", "year2019", "term1", "org2"};
writer.setValues(newRow);
writer.flush();
olapConn.commit(); // 进行提交
//可以查询到[account1, year2019, term1, org2, 30]这笔记录

```

## 断开 Shrek 服务器连接

在使用完毕后，您可以通过调用 `olapConn.close()` 方法断开与shrek服务器的连接；

在数据库连接池使用完毕后，您可以通过调用`dataSources.close()`方法关闭连接池，释放连接。

上一篇：[跨cube计算](#)

下一篇：[全局数据范围](#)