

RMI 简单集成

Bliss Chung

2015 年 10 月 1 日

Contents

1	关于 Z-RMI	1
1.1	TODO 制作视频演示（配奇迹背景音乐）	1
2	下载和安装（Download and Installation）	1
2.1	安装方式一	1
2.2	安装方式二（ivy）	1
3	相关配置（Configuration）	1
4	使用（Usage）	2
4.1	启动服务	2
4.2	客户端远程方法调用	2

1 关于 Z-RMI

1.1 TODO 制作视频演示 (配奇迹背景音乐)

2 下载和安装 (Download and Installation)

2.1 安装方式一

直接下载 z-rmi-all-1.0.jar 并添加到项目的 CLASSPATH 中。

<https://github.com/zl3907/z-rmi/raw/master/dist/z-rmi/z-rmi-all-1.0-jars.jar>

2.2 安装方式二 (ivy)

注：该项目包暂时发布在内网服务器的资源库中，因此本方法暂只能在公司内部使用。

```
1 <!-- 在你的 ivy.xml 文件里添加下面的内容 -->
2 <dependency org="com.ztools" name="z-rmi" rev="1.0" />
```

3 相关配置 (Configuration)

项目所有的配置文件都在 src/conf/ 文件夹下,系统启动时默认读取 src/conf/configuration.cfg 文件，这个文件是配置文件的入口，也可以在启动时以参数的形式指定该文件的物理路径。其它配置文件的路径都可以在这个文件里进行配置和修改。

```
src/conf
—— configuration.cfg # 配置文件入口
—— rmi-handler.xml   # RMI 主机地址、端口和服务名
—— z-rmi.cfg         # RMI 服务端任务执行引擎相关配置，可以自行扩展
```

配置文件都可以用默认的，扩展自己的远程方法时只要按照下面的示例在 z-rmi.cfg 添加相应的内容就可以了：

```
1 # 三行配置
2 com.ztools.rmi.client.IHandlerFactory = com.ztools.rmi.client.CfgHandlerFactory
3 rmi.handler = conf/rmi-handler.xml
4 remote.executor.example = com.ztools.rmi.executors.Example
5
6 ## registry executors
7 ## 注意事项
8 # 1. 在这里可以配置你想公开给远程客户端的调用的类；
9 # 2. 引擎只会把公有方法注册到服务列表中供客户端调用；
```

```

10 # 3. 目前暂不支持一个类中有相同的方法名;
11 # 4. 注册的类应该有一个默认的构造函数;
12 #
13 ## 配置格式
14 # < 前缀 >.< 唯一标识 > = < 完整的类名 >
15 #
16 # < 前缀 > 只能是 “remote.executor”
17 # < 唯一标识 > 可以是任意值，只要不重复就行，但要复合 properties 文件的基本要求
18 # < 完整的类名 > 是你想要公开给远程客户端调用的类名
19 #
20 ## 示例
21 # remote.executor.string = java.util.String

```

4 使用 (Usage)

4.1 启动服务

命令行启动:

```
1 | java -cp z-rmi-all-1.0.jar:yourlib/* com.ztools.rmi.service.RmiService
```

或:

```
1 | java -jar z-rmi-all-1.0.jar -s
```

4.2 客户端远程方法调用

java 调用示例:

```

1 | public static void main(String[] args) {
2 |     RmiClient client = new RmiClient();
3 |     try {
4 |         String remoteHandler = "com.ztools.rmi.executors.Example";
5 |         // 调用远程方法并等待服务器返回结果，断线会重新连接尝试连接
6 |         client.execute(new Task(remoteHandler, "sayHello"), true);
7 |
8 |         // 调用远程方法，如果服务器异常不会重新连接，直接返回
9 |         client.execute(new Task(remoteHandler, "print", "Hello"));
10 |    } catch (Exception e) {
11 |        // TODO Auto-generated catch block

```

```
12 |     e.printStackTrace();  
13 | }  
14 | }
```