ODL开发环境搭建及第一个实例

OPEN DAYLIGHT | 未来网络学院 Future Network Institute

目录

JDK安装

JDK安装

Windows or Mac OS-X:

从以下地址下载JDK 8最新发布版本（当前最新版本应该是8u131）并安装

http://www.oracle.com/technetwork/java/javase/downloads/

Linux

Fedora:

sudo dnf install java-1.8.0-openjdk java-1.8.0-openjdk-devel

Ubuntu:

sudo apt-get install openjdk-8-jdk

JDK环境变量设置

Java环境变量

在~/.profile里设置环境变量

export JAVA_HOME=$(WHERE_YOU_INSTALLED_JDK)

export PATH=${JAVA_HOME}/bin:${PATH}

Window操作系统的环境变量

请右键点击 我的电脑-〉属性-〉高级系统设置-〉高级-〉环境变量

除了以上环境变量需要设置，请再增加

JAVA_TOOLS_OPTIONS，值为 -Dfile.encoding=UTF-8

Maven安装配置

Maven安装及配置

Windows or Mac OS-X:

下载并安装maven最新发布版本（当前最新正式发布版本3.5.0）

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

Linux

Fedora:

sudo dnf install maven

Ubuntu:

sudo apt-get install maven

wget -q -O - https://raw.githubusercontent.com/opendaylight/odlparent/master/settings.xml > ~/.m2/settings.xml

Maven环境变量


```
export M2_HOME=${WHERE_YOU_UNZIPPED_MAVEN}/apache-maven-
  ${MVN_VERSION}
export PATH=${M2_HOME}/bin:${PATH}
export MAVEN_OPTS=”  -Xmx1024m”
```

git的安装配置

如果你想在社区贡献代码，还需要安装git

Mac上安装git请参考
http://blog.csdn.net/nellson/article/details/51526273
Windows上安装git请参考
http://www.cnblogs.com/vitah/p/3612473.html

Fedora:
sudo dnf install git
Ubuntu:
sudo apt-get install git-core

目录

由maven项目骨架生成项目

进入你的工作目录，命令行执行

```
mvn archetype:generate -DarchetypeGroupId=org.opendaylight.controller -
  DarchetypeArtifactId=opendaylight-startup-archetype  -
  DarchetypeRepository=http://nexus.opendaylight.org/content/repositories/
  public/  -
  DarchetypeCatalog=http://nexus.opendaylight.org/content/repositories/pub
  lic/archetype-catalog.xml
```

添加如下参数可指定版本

```
-DarchetypeVersion=1.3.1-Carbon
```

根据命令行提示填写红色部分，其他回车

Downloaded: https://nexus.opendaylight.org/content/repositories/public/org/opendaylight/controller/opendaylight-startup-archetype/1.3.1-Carbon/opendaylight-startup-archetype-1.3.1-Carbon.jar (28 KB at 30.6 KB/sec)

Define value for property 'groupId': com.mycompany.proj

Define value for property 'artifactId': traffic

[INFO] Using property: version = 0.1.0-SNAPSHOT

Define value for property 'package' com.mycompany.proj: :

Define value for property 'classPrefix' Traffic: :

Define value for property 'copyright': MYCompany

[INFO] Using property: copyrightYear = 2017

Confirm properties configuration:

groupId: com.mycompany.proj

artifactId: traffic

version: 0.1.0-SNAPSHOT

package: com.mycompany.proj

classPrefix: Traffic

copyright: MYCompany

copyrightYear: 2017

Y: : Y

traffic项目的目录结构介绍

- api/ ---------- yang模型文件目录
- artifacts/ ---------- 管理项目组件的坐标
- cli/ ---------- 命令行工具实现目录
- features/ ---------- feature组织
- impl/ ---------- 业务逻辑实现
- it/ ---------- 集成测试
- karaf/ ---------- 打包karaf版本
- pom.xml ---------- 根pom

编译执行

在traffic目录下执行 mvn clean install

初次执行需要等待一段时间，执行完毕

在karaf/target/assembly/bin目录下执行karaf（Windows操作系统，执行karaf.bat)

修改yang模型

- 修改 api/src/main/yang/traffic.yang文件，添加一个rpc

```
rpc hello-world {
        input {
            leaf name {
                type string;
            }
        }
        output {
            leaf greeting {
                type string;
            }
        }
    }
```

实现业务逻辑

修改Java代码(impl/src/main/java/com/mycompany/proj/impl/TrafficProvider.java)

```
public class TrafficProvider implements TrafficService {
    ......
    @Override
    public Future<RpcResult<HelloWorldOutput>> helloWorld(HelloWorldInput helloWorldInput) {
        HelloWorldOutputBuilder helloBuilder = new HelloWorldOutputBuilder();
        helloBuilder.setGreeting("Hello " + helloWorldInput.getName());
        return RpcResultBuilder.success(helloBuilder.build()).buildFuture();
    }
}
```

修改blueprint文件（impl/src/main/resources/org/opendaylight/blueprint/impl-blueprint.xml）
增加一行

```
<odl:rpc-implementation ref="provider" />
```

通过web页面调用RPC

在本机上通过浏览器打开

http://localhost:8181/apidoc/explorer/index.html

输入用户名admin，密码admin登录

点击 traffic(2015-01-05)

点击 POST /operations/traffic:hello-world

输入 {"hello:input": { "name":"Jojo"}}

点击 Try it out!

下一讲：ODL RPC开发详细讨论

- RPC的yang定义
- RPC实现
- RPC调用

Thank You !