

Java Cloud Storage SDK develop guide

目录

1.	准备.....	- 2 -
1.1	依赖.....	- 2 -
1.2	安装 Maven.....	- 2 -
1.3	为 Eclipse 安装 m2eclipse 插件.....	- 2 -
1.4	构建 Maven 工程.....	- 3 -
1.5	修改 pom.xml 添加库依赖.....	- 3 -
2.	修改程序使用盛大云存储.....	- 5 -

1. 准备

1.1 依赖

当前版本的 Cloud Storage Java SDK 所依赖的开发包有以下这些：

```
commons-lang 2.5
commons-codec 1.5
commons-pool 1.5.6
guava r09
log4j 1.2.16
slf4j-log4j12 1.6.2
slf4j-api 1.6.1
log4j-over-slf4j 1.6.1
joda-time 2.0
httpclient 4.1.1
junit 4.8.2
mockito-all 1.8.5
```

1.2 安装 Maven

下载 maven 2.2.1 :

<http://www.apache.org/dyn/closer.cgi/maven/binaries/apache-maven-2.2.1-bin.zip>

解压缩到需要安装的目录,这里假设解压缩在 C:\Program Files\Apache Software Foundation
添加 M2_HOME 环境变量 (WinKey + Pause), 选择“高级”选项卡, “环境变量” 按钮, 新建系统变量, 变量名为 M2_HOME, 值为 C:\Program Files\Apache Software Foundation\apache-maven-2.2.1

添加 M2 环境变量, 变量名为 M2, 值为%M2_HOME%\bin

(确认已经设置了 JAVA_HOME 环境变量, 例如:JAVA_HOME 设置为你安装的 JDK 路径, C:\Program Files\Java\jdk1.5.0_02, 并且%JAVA_HOME%\bin 已经添加到 Path 环境变量中)

启动命令行, 运行 mvn -version 查看是否已经正确安装

1.3 为 Eclipse 安装 m2eclipse 插件

m2eclipse core update site : <http://m2eclipse.sonatype.org/sites/m2e>

- 1) 选择 Help > Install New Software. 显示 Install 对话框
- 2) 将 Update site url 粘贴在“Work with”的编辑框中并点击确定
- 3) 选择 m2eclipse 下面列表中的(Required)选项
- 4) 点击下一步确认安装, 最后重启 Eclipse

1.4 构建 Maven 工程

运行下面命令:

```
mvn archetype:generate -DarchetypeGroupId=org.apache.maven.archetypes
-DgroupId=com.mycompany.app -DartifactId=my-app
```

创建好的工程目录树:

```
my-app
|-- pom.xml
`-- src
    |-- main
    |   |-- java
    |   |   |-- com
    |   |   |   |-- mycompany
    |   |   |   |   |-- app
    |   |   |   |   |   App.java
    |   |-- test
    |   |   |-- java
    |   |   |   |-- com
    |   |   |   |   |-- mycompany
    |   |   |   |   |   app
    |   |   |   |   |   |   AppTest.java
```

1.5 修改 pom.xml 添加库依赖

在 `dependencies` 标签中添加如下内容:

```
<dependency>
  <groupId>commons-lang</groupId>
  <artifactId>commons-lang</artifactId>
  <version>2.5</version>
</dependency>
<dependency>
  <groupId>commons-codec</groupId>
  <artifactId>commons-codec</artifactId>
  <version>1.5</version>
</dependency>
<dependency>
  <groupId>commons-pool</groupId>
  <artifactId>commons-pool</artifactId>
  <version>1.5.6</version>
</dependency>
<dependency>
```

```
<groupId>com.google.guava</groupId>
<artifactId>guava</artifactId>
<version>r09</version>
</dependency>
<dependency>
<groupId>log4j</groupId>
<artifactId>log4j</artifactId>
<version>1.2.16</version>
</dependency>
<dependency>
<groupId>org.slf4j</groupId>
<artifactId>slf4j-log4j12</artifactId>
<version>1.6.2</version>
</dependency>
<dependency>
<groupId>org.slf4j</groupId>
<artifactId>slf4j-api</artifactId>
<version>1.6.1</version>
</dependency>
<dependency>
<groupId>org.slf4j</groupId>
<artifactId>log4j-over-slf4j</artifactId>
<version>1.6.1</version>
</dependency>
<dependency>
<groupId>joda-time</groupId>
<artifactId>joda-time</artifactId>
<version>2.0</version>
</dependency>
<dependency>
<groupId>org.apache.httpcomponents</groupId>
<artifactId>httpclient</artifactId>
<version>4.1.1</version>
</dependency>
<dependency>
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>4.8.2</version>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.mockito</groupId>
<artifactId>mockito-all</artifactId>
```

```
<version>1.8.5</version>
<scope>test</scope>
</dependency>
<dependency>
  <groupId>snda-cloud-storage-java-sdk</groupId>
  <artifactId>snda-cloud-storage-java-sdk</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <scope>system</scope>
  <systemPath>/home/user
/Downloads/snda-cloud-storage-java-sdk-0.0.1-SNAPSHOT.jar</sys
temPath>
</dependency>
```

然后在 pom.xml 文件所在目录下运行命令 `mvn eclipse:eclipse` 并在 Eclipse 中打开该项目

2. 修改程序使用盛大云存储

在 App.java 文件中添加如下代码段:

```
try {
    String access_key_id = "xxxxx";
    String access_key_secret = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx";
    ProviderCredentials myCred = new SNDACredentials(access_key_id,
access_key_secret);
    CSService myService = new RestCSService(myCred);
    StorageBucket bucket = myService.createBucket("mybucket");

    CSObject object;
    object = new CSObject(new File("filepath"));
    myService.putObject(bucket.getName(), object);
} catch (ServiceException e) {
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMessage());
    System.out.println(e.getErrorRequestId());
    System.out.println(e.getErrorResource());
} catch (Exception e) {
    e.printStackTrace();
}
```

`access_key_id` 和 `access_key_secret` 需指定为用户持有的两个对应的 key，上述程序首先创建 `SNDACredential` 对象，然后再创建 `CSService` 对象，使用该 Service 对象的接口，`createBucket` 用来创建 Bucket，`putObject` 用来上传 Object。更详细的 API 说明请参见盛大云存储的 Java SDK API 文档。