# Hive总结2 自定义函数

根据用户自定义函数类别分别为一下三种：

```
1. UDF (User-Defined-Function)
   一进一出
2. UDAF (User-Defined Aggregation Funtion)
   聚集函数：多进一出
   类似于：count/max/min
3. UDTF (User-Defined Table-Generating Function)
   一进多出
   如 lateral view  explore()
```

编程步骤

```
1. 继承org.apache.hadoop.hive.ql.UDF
2. 需要实现 evaluate函数：evaluate函数支持重载
```

注意事项

```
1. UDF必须要有返回类型，可以返回null，但是返回类型不能是void
2. UDF中常用Text/LongWritable等类型，不推荐使用java类型
```

## 简单UDF示例

**第一步：创建macen工程，导入jar包**

```
<repositories>
    <repository>
        <id>cloudera</id>
 <url>https://repository.cloudera.com/artifactory/cloudera-repos/</url>
    </repository>
</repositories>
<dependencies>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-common</artifactId>
        <version>2.6.0-cdh5.14.0</version>
    </dependency>
    <dependency>
        <groupId>org.apache.hive</groupId>
        <artifactId>hive-exec</artifactId>
        <version>1.1.0-cdh5.14.0</version>
    </dependency>
</dependencies>
<build>
<plugins>
    <plugin>
        <groupId>org.apache.maven.plugins</groupId>
```

```
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.0</version>
        <configuration>
            <source>1.8</source>
            <target>1.8</target>
            <encoding>UTF-8</encoding>
        </configuration>
    </plugin>
     <plugin>
         <groupId>org.apache.maven.plugins</groupId>
         <artifactId>maven-shade-plugin</artifactId>
         <version>2.2</version>
         <executions>
             <execution>
                 <phase>package</phase>
                 <goals>
                     <goal>shade</goal>
                 </goals>
                 <configuration>
                     <filters>
                         <filter>
                             <artifact>*:*</artifact>
                             <excludes>
                                 <exclude>META-INF/*.SF</exclude>
                                 <exclude>META-INF/*.DSA</exclude>
                                 <exclude>META-INF/*/RSA</exclude>
                             </excludes>
                         </filter>
                     </filters>
                 </configuration>
             </execution>
         </executions>
     </plugin>
</plugins>
</build>
```

**第二步：开发java类继承UDF，并重载evaluate方法**

```
public class UDF_01 extends UDF {

    //自定义函数,实现大写转换!
    public Text evaluate(final  Text s){
        if (null == s){
            return null;
        }
        //返回大写字母
        return new Text(s.toString().toUpperCase());
    }
}
```

**第三步：将我们的项目打包，上传**

这里我们将Hive.jar存放在了/opt/hive的目录下

```
[root@node01 hive]# ll
总用量 34604
-rw-r--r-- 1 root root       39 6月   11 11:23 course.csv
-rw-r--r-- 1 root root 17706625 11月 21 2019 Hive.jar
-rw-r--r-- 1 root root 17707121 11月 21 2019 hive_java.jar
-rw-r--r-- 1 root root      162 7月   21 15:23 score.csv
-rw-r--r-- 1 root root      200 6月   11 11:16 student.csv
-rw-r--r-- 1 root root       30 10月 21 16:31 techer.csv
```

**第四步：将jar包添加到hive**

在hive shell 内 add jar 路径+jar包

```
hive (default)> add jar /opt/hive/Hive.jar;
Added [/opt/hive/Hive.jar] to class path
Added resources: [/opt/hive/Hive.jar]
```

**第五步：创建临时函数（永久的函数将temporary删掉）**

```
hive (default)> create function toUpperCase as 'demo17_UDF.UDF_01';
OK
Time taken: 0.062 seconds
hive (default)>
```

**第六步：调用验证**

```
hive (default)> select toUpperCase('hive');
OK
_c0
HIVE
Time taken: 0.75 seconds, Fetched: 1 row(s)
```

方法名

参数

自定义函数执行后的结果

# 通过reflect调用java方法

经历完了上面自定义函数过后,相信大家一定还是觉得过程有点繁琐。不用担心,接下来小菌将为大家带来如何在Hive中调用java方法。

**第一步：使用java代码编写业务逻辑，并打包上传**

```
public class JAVA_02  {

    public static String addInfo(String info){
        return info+"__ I love Hive! __ ";
    }
}
```

**第二步：将jar包添加到hive**

```
hive (default)> add jar /opt/hive/hive_java.jar;
Added [/opt/hive/hive_java.jar] to class path
Added resources: [/opt/hive/hive_java.jar]
hive (default)>
```

**第三步：调用**

select reflect (‘参数一’,‘参数二’,‘参数三’)

- 参数一: 包名-类名
- 参数二: 方法名
- 参数三: 需要计算的数据；函数输入参数