

1 实验目标

1.1 实验背景介绍

本实验为课程《ACA21107 深入理解阿里云产品-Data IDE》中的配套实验。主要阐述如何使用 ODPS 的 Eclipse 插件和 Data IDE 来完成 Java UDF 的开发，学员可以根据本实验手册，去学习 Java UDF 的开发流程以及实现一个 Java UDF 实例：将大写转小写。

1.2 实验环境架构

实验环境架构：阿里云开放数据处理服务 ODPS

1.3 云端服务环境

暂无

1.4 云端开发桌面环境

2 实验内容

2.1 总体流程



2.2 配置环境并本地模式测试 UDF

2.2.1 配置 Java 编程环境

step1：官网导航中找到并下载 ODPS for eclipse 插件。

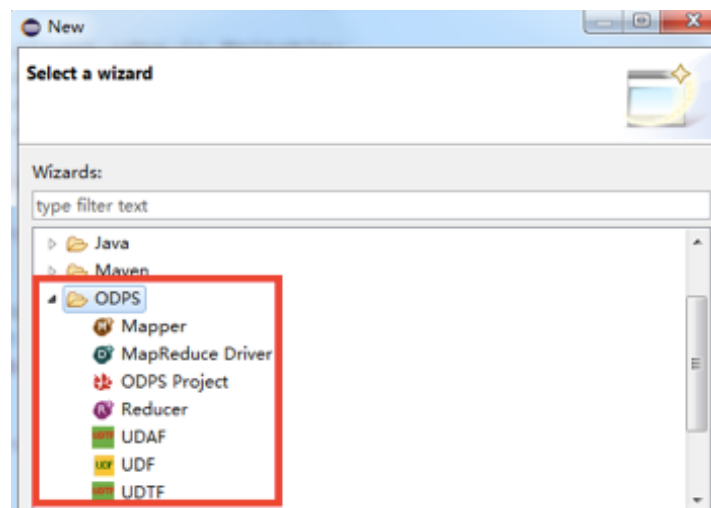
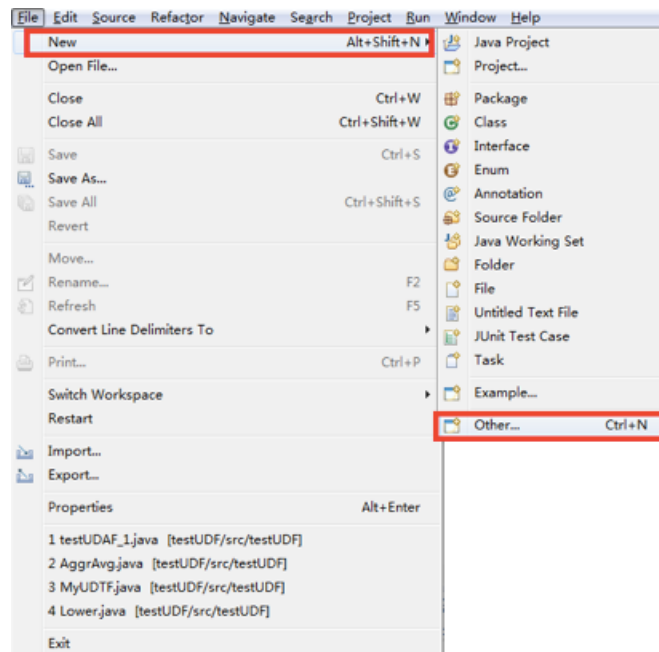
(https://help.aliyun.com/document_detail/odps/download/download.html)

step2：将插件解压并复制到 eclipse 安装目录下的 plug-in 子目录下。

step3：启动 eclipse。

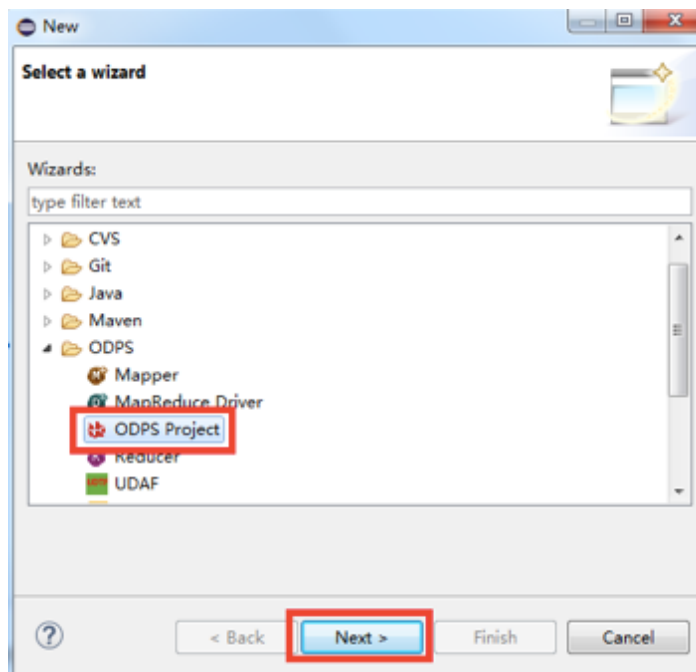
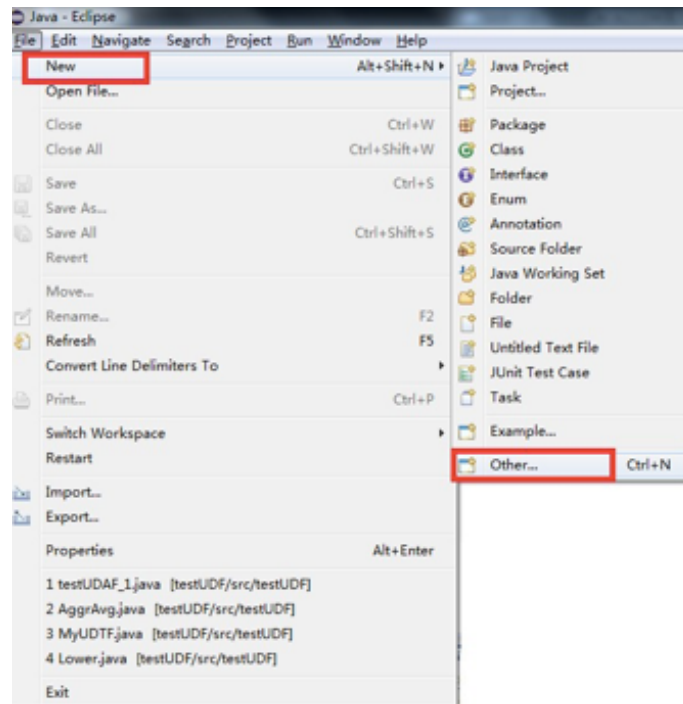
step3：检查 Wizard 选项里面是否有 ODPS 的目录。

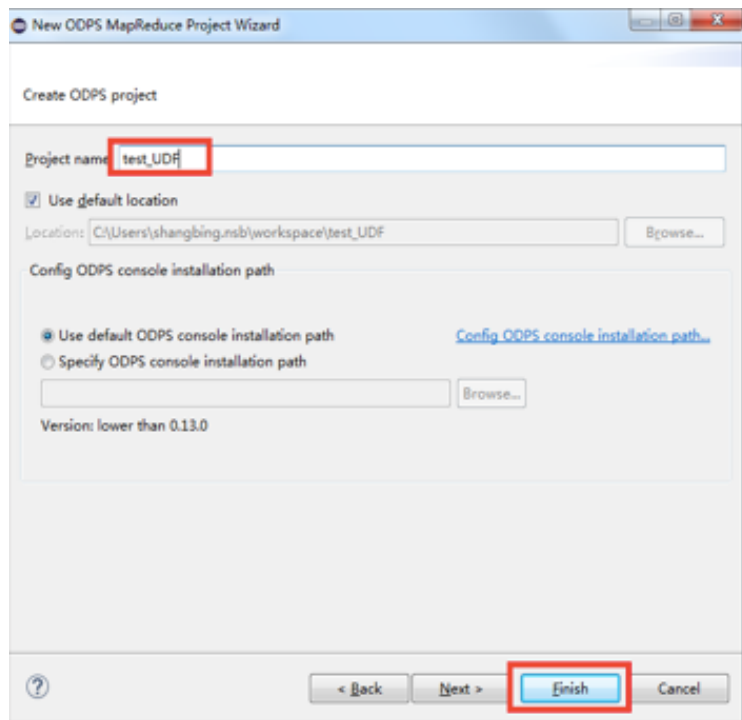
点击 **New>Other...**在 Wizard 弹出框中有 ODPS 文件夹选项，则表示安装成功。



2.2.2 新建项目

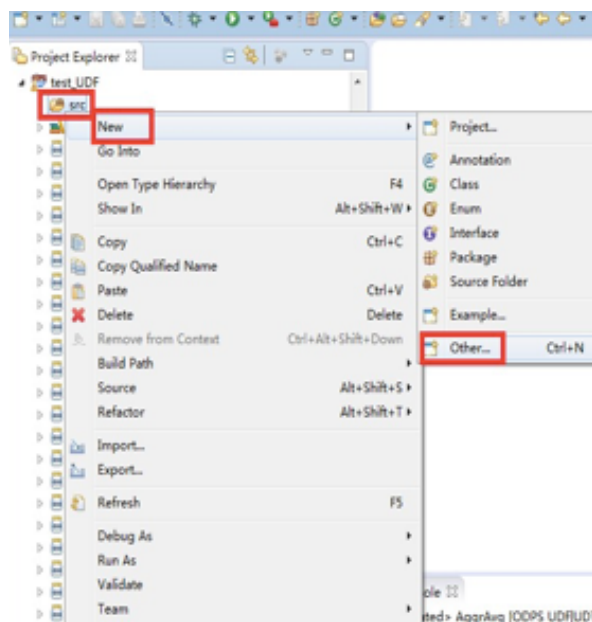
step1 : 点击 **New>Other...>ODPS Project** , 点击 **Next>**。在对话框中输入 Project name , 此处为 test_UDF , 再点击 **Finish**。

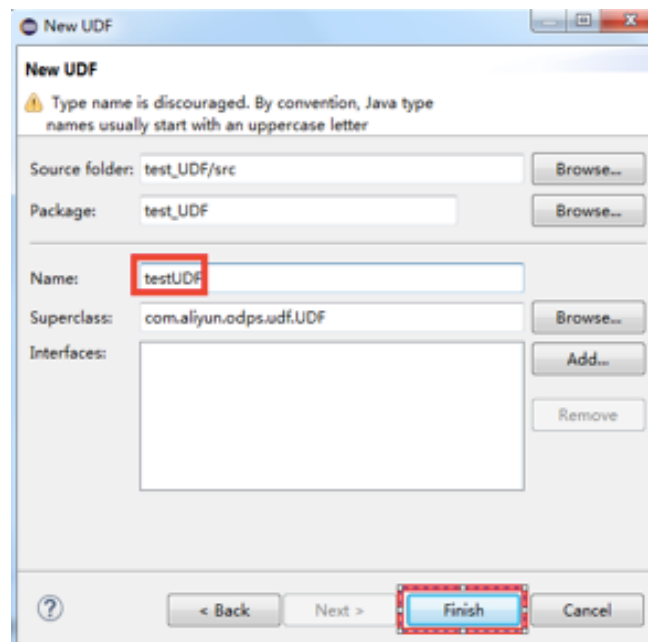
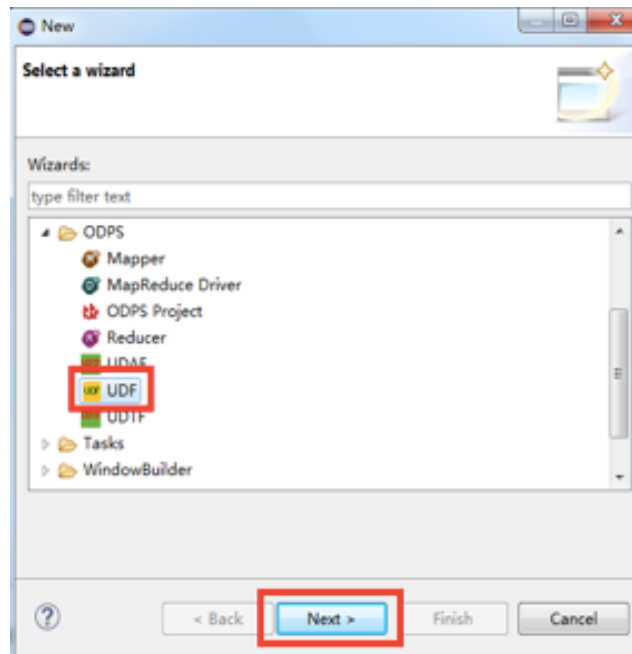




2.2.3 新增 UDF 类

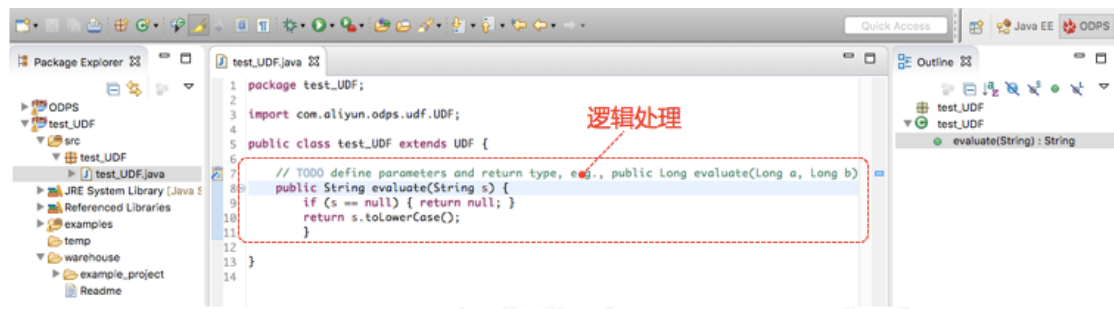
点击新建的 test_UDF 项目下的 src 文件夹，右键 new>Other...，然后在 New UDF 弹出框 Name 选项框中输入 UDF 类名，该示例为 test_UDF。





2.2.4 添加逻辑处理

按照 ODPS UDF 编程框架，添加逻辑处理：



代码示例如下：

```

package test_UDF;

import com.aliyun.odps.udf.UDF;

public class test_UDF extends UDF {

    // TODO define parameters and return type, e.g., public Long
    evaluate(Long a, Long b)
    public String evaluate(String s) {
        if (s == null) { return null; }
        return s.toLowerCase();
    }

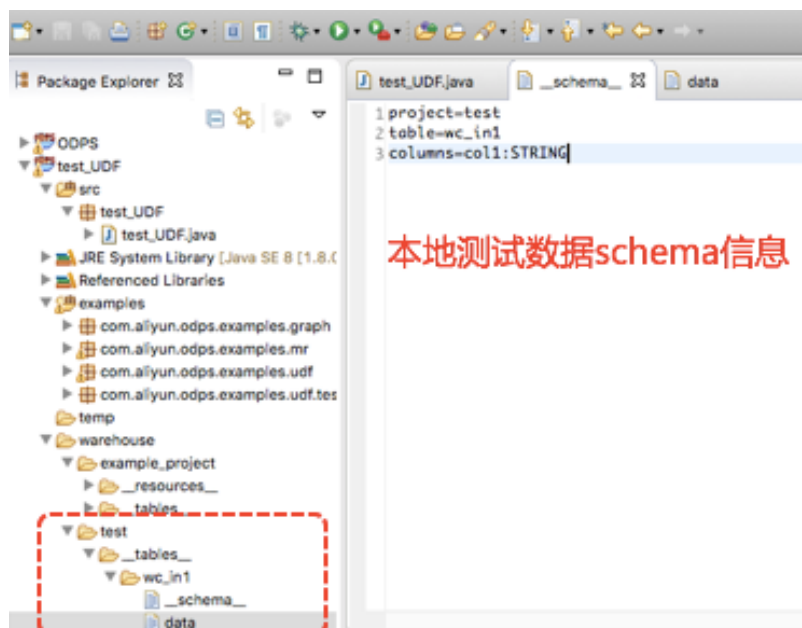
}

```

2.2.5 准备本地测试数据

找到 test_UDF 项目下的 warehouse 按照如下方式准备本地测试数据 如下：

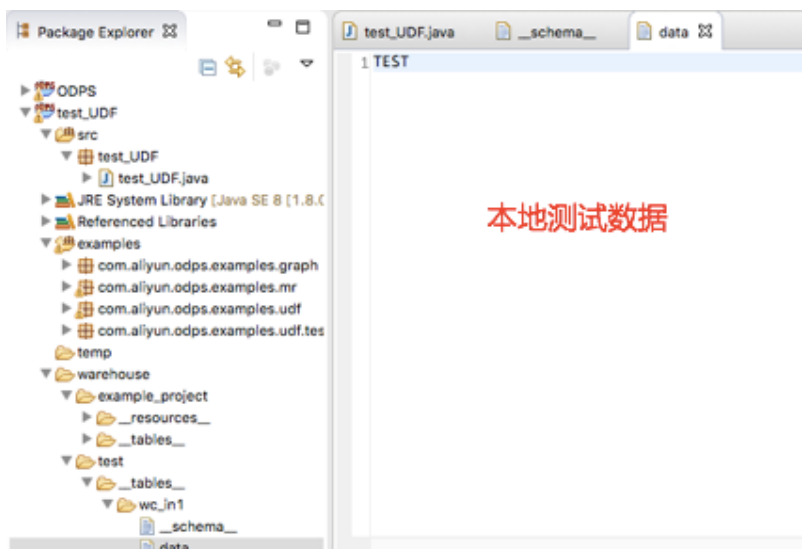
step1：在 _schema_ 文件中编写如下内容。



_schema_内容：

```
project=test
table=wc_in1
columns=col1:STRING
```

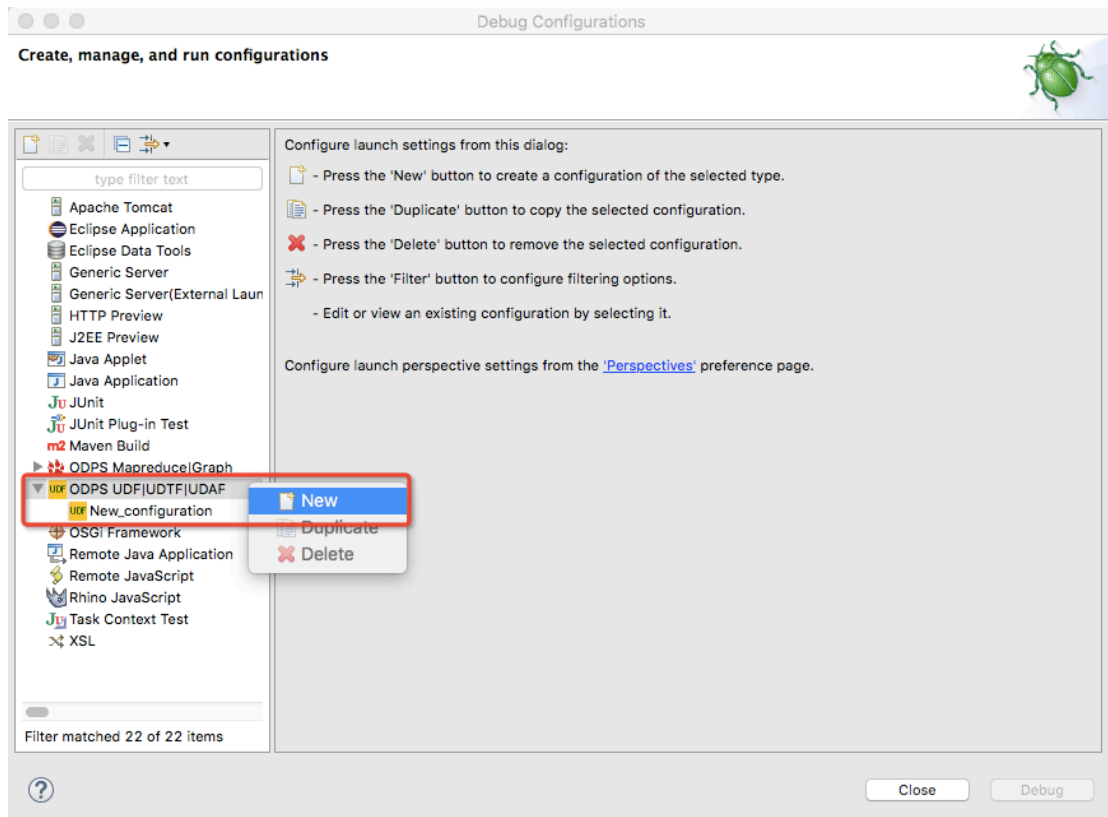
step2：在 data 文件中输入测试数据，示例“TEST”。



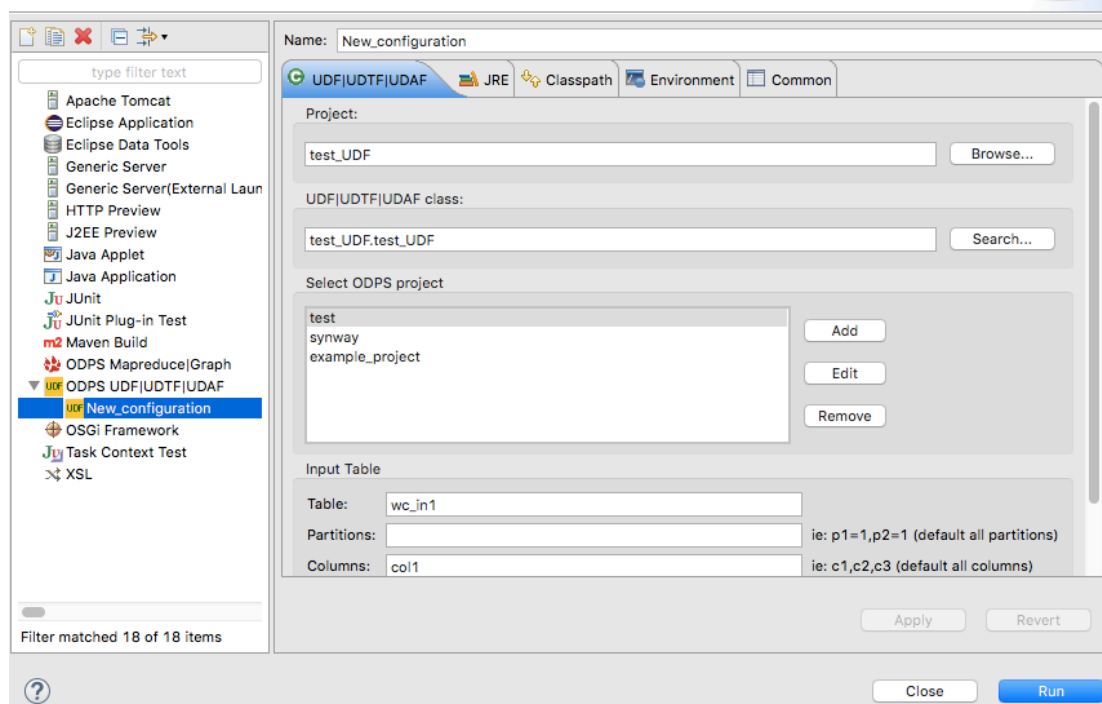
2.2.6 编译调试

step1：从菜单栏选择 Run-->Run Configurations....

step2 : 在 Debug Configurations 弹出框中找到 ODPS UDF|UDTF|UDAF , 并右键点击 New。

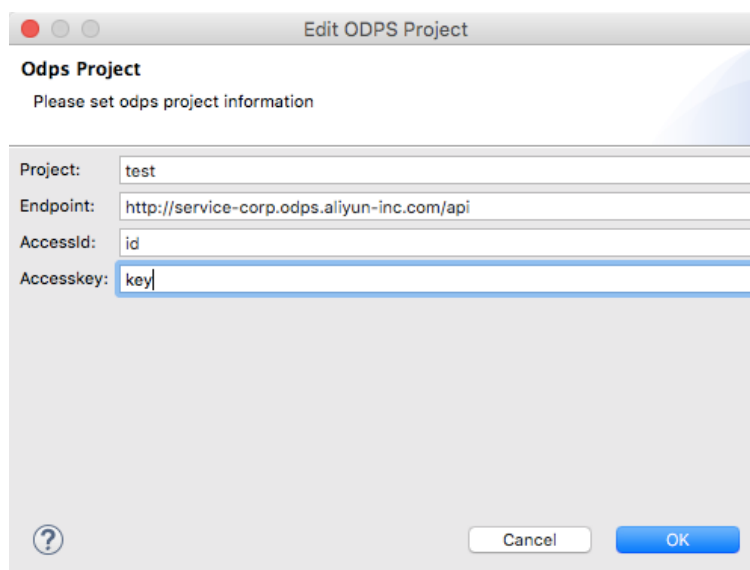


step3 : 在 New_configuration 页面中分别配置 Project、UDF|UDTF|UDAF class、Select ODPS project 和 Input Table。



配置如下：

- **Project**：输入 ODPS 项目，本示例为 test_UDF。
- **UDF|UDTF|UDAF class**：输入 UDF 类名，从顶层报名到类名，本示例为 test_UDF.test_UDF。
- **Select ODPS project**：点击 add，进行添加 ODPS project。



Project : test

Endpoint : <http://service-corp.odps.aliyun-inc.com/api>

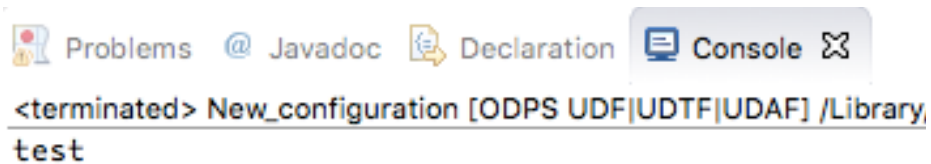
Accessid : id

Accesskey : key

■ Table : 输入章节 2.6 准备本地测试数据 中的表名，本示例为 wc_in1。

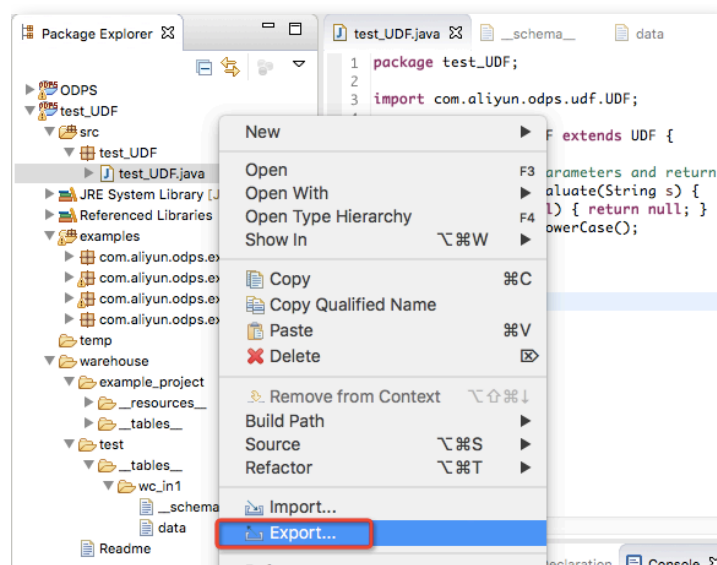
■ Columns : 输入章节 2.6 准备本地测试数据 中的表的字段名，本示例为 col1。

step4 : 配置好信息后，点击 Run 进行本地调试。

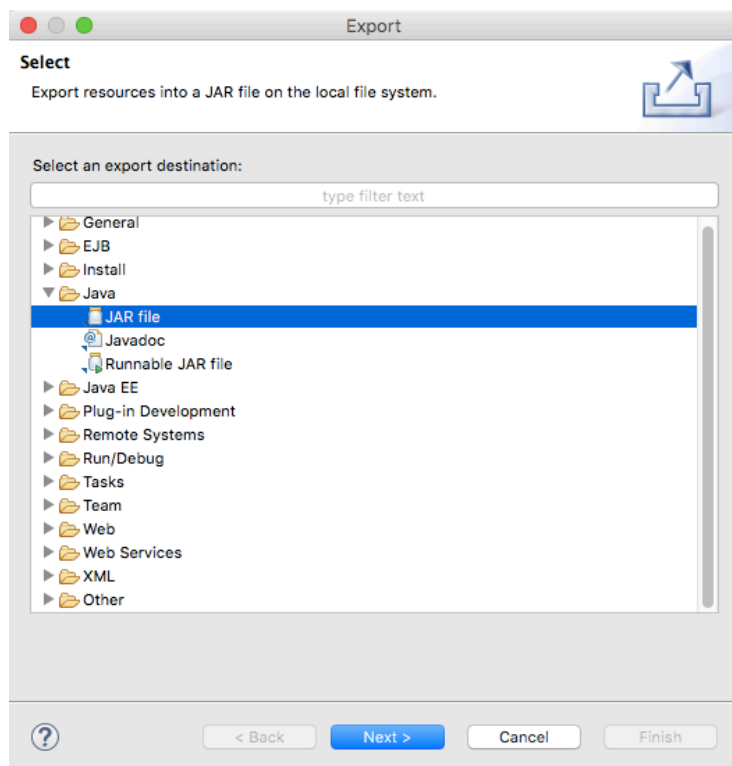


2.3 导出 jar 包

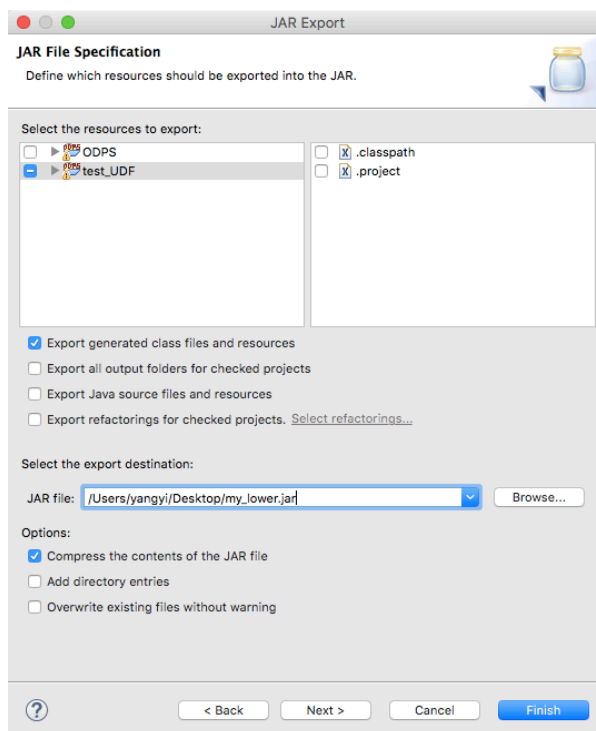
step1 : 找到 test_UDF.java 文件，右键选择 Export...



step2 : 在弹出框中选择 JAR file。



step3 : 点击 Next> ,选择需打包的字眼 ,并选择导出路径并命名 jar 包名称 ,
本示例选择有意义的名称 , my_lower.jar。



step4 : 点击 Finish。

2.4 注册 ODPS 资源

step1：进入 Data IDE，点击新建选择上传资源。

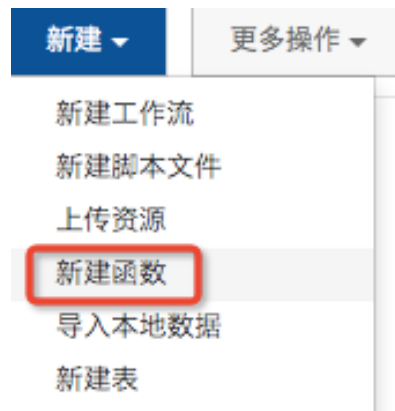


step2 在资源上传页面中选择 章节 2.3 中导出的 jar 包 并选择类型为 jar，同时勾选上传为 ODPS 资源，然后选择资源存放的文件目录。

step3：点击提交。

2.5 创建自定义函数

step1：进入 Data IDE，点击新建选择新建函数。



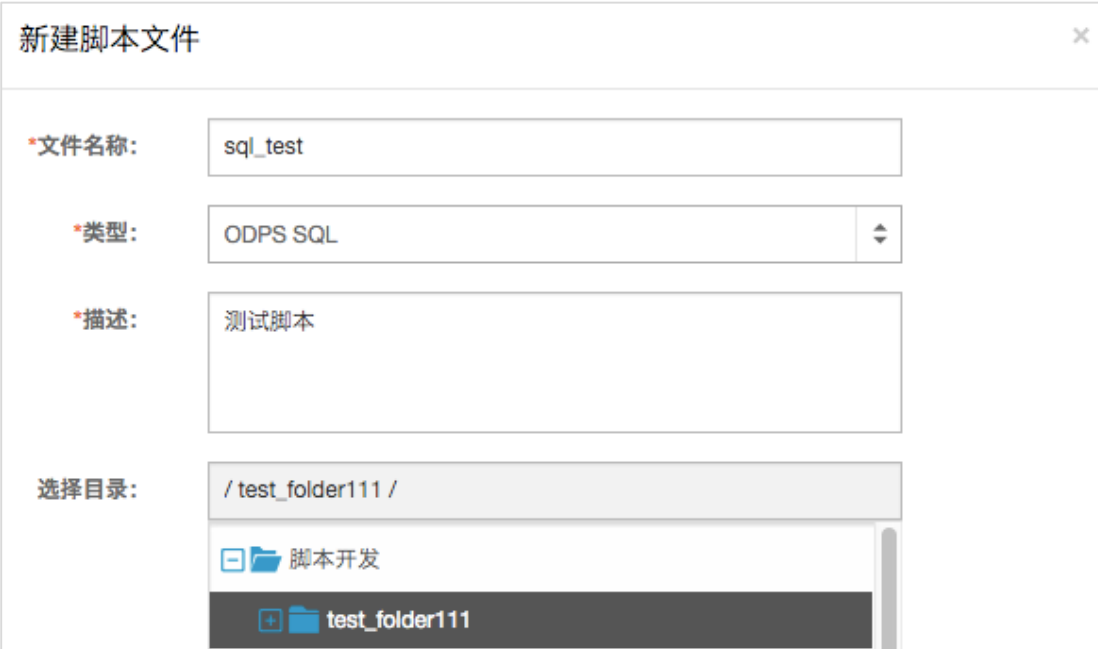
step2：在新建 ODPS 函数弹出框中输入函数名、类名、选择资源、用途、命令格式、参数说明以及选择函数存放文件夹目录。

■ **函数名**：my_lower；

- **类名**：格式为包名.类名，本示例为 test_UDF.test_UDF；
- **资源**：选择已上传的 my_lower.jar；

2.6 试用 Java UDF 函数

step1：进入 Data IDE，点击新建选择新建脚本文件。



新建脚本文件

*文件名称: sql_test

*类型: ODPS SQL

*描述: 测试脚本

选择目录: / test_folder111 /

脚本开发

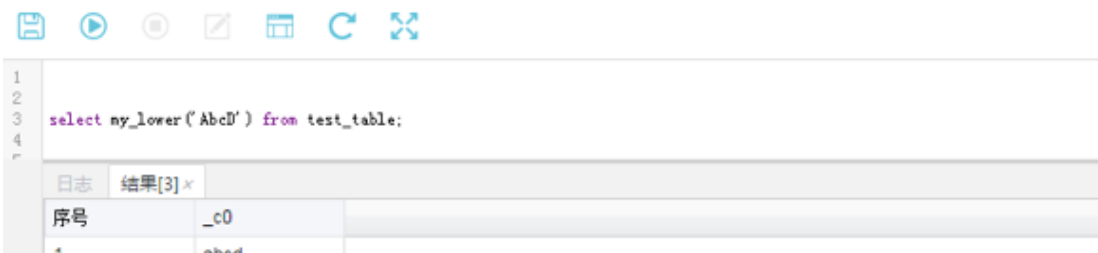
test_folder111

step2：创建 dual 表。

```
create table if not exists dual(id bigint);

insert overwrite table dual select count(*) from dual;
```

step3：试用 Java UDF 函数，select my_lower('AbcD') from dual;



```
1
2
3 select my_lower('AbcD') from test_table;
4
```

序号	_c0
1	abcd