阿里云
MaxCompute

工具及下载

文档版本：20190402

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的”现状“、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **禁止:** 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **警告:** 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 **说明:** 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 确定。
`courier` 字体	命令。	执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。
##	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[]或者[a|b]	表示可选项，至多选择一个。	`ipconfig` *[-all|-t]*
{}或者{a|b}	表示必选项，至多选择一个。	`swich` *{stand | slave}*

目录

1 客户端

本文为您介绍如何使用客户端命令行工具使用MaxCompute服务的基础功能。

在使用MaxCompute客户端前，请首先安装并配置客户端。

说明:

- 请不要依赖客户端的输出格式来做任何的解析工作。客户端的输出格式不承诺向前兼容，不同版本间的客户端命令格式及行为有差异。
- 关于客户端的基本命令介绍，请参见基本命令。
- 新版客户端：点击此处即可下载新版客户端。关于不同环境下客户端的安装配置信息请参见安装并配置客户端
- 客户端从0.28.0版开始支持JDK 1.9，这之前的版本只能用JDK 1.8。
- 客户端从0.27.0版本开始支持MaxCompute 2.0新数据类型。

安装并配置好客户端后，您可借助命令行工具进行以下操作：

获取帮助

若想显示客户端的帮助信息，命令格式如下所示：

```
odps@ >./bin/odpscmd -h;
```

您也可以在交互模式下键入`h;`或`help;`（不区分大小写）。

客户端还提供了`help [keyword];`命令，可获取到与关键字有关的命令提示。例如：输入`help table;`可以得到与table操作相关的命令提示，如下所示：

```
odps@ odps> help table;
Usage: alter table  merge smallfiles
Usage: show tables [in ]
       list|ls tables [-p,-project ]
Usage: describe|desc [.] [partition()]
Usage: read [.] [([,..])] [PARTITION ()] [line_num]
```

启动参数

在启动时，您可指定一系列参数，如下所示：

```
Usage: odpscmd [OPTION]...
where options include:
    --help (-h)for help
    --project= use project
    --endpoint= set endpoint
```

```
    -u  -p  user name and password
    -k  will skip begining queries and start from specified position
    -r  set retry times
    -f <"file_path;"> execute command in file
    -e <"command;[command;]..."> execute command, include sql command
    -C will display job counters
```

以-f参数为例，操作如下：

1. 准备本地脚本文件*script.txt*，假设存放在D盘，文件内容如下所示：

```
DROP TABLE IF EXISTS test_table_mj;
CREATE TABLE test_table_mj (id string, name string);
DROP TABLE test_table_mj;
```

2. 打开您的CMD命令行工具，进入odpscmd客户端所在路径，运行如下命令：

```
odpscmd\bin>odpscmd -f D:/script.txt;
```

3. 您也可以使用上述命令批量建表或删表。

交互模式

直接运行客户端即可进入到交互模式，如下所示：

```
[admin: ~]$odpscmd
Aliyun ODPS Command Line Tool
Version 1.0
@Copyright 2012 Alibaba Cloud Computing Co., Ltd. All rights reserved.
odps@ odps> INSERT OVERWRITE TABLE DUAL SELECT * FROM DUAL;
```

在光标位置输入命令（以分号作为语句的结束标志），回车即可运行。

续跑

- 在用-e或-f 模式运行时，如果有多条语句，想从中间某条语句开始运行，可以指定参数-k， 表示忽略前面的语句，从指定位置的语句开始运行。当指定参数<=0 时，从第一条语句开始执行。
- 每个以分号分隔的语句被视为一条有效语句，在运行时会打印出当前运行成功或者失败的是第几条语句。

示例如下：

假设文件*/tmp/dual.sql*中有三条SQL语句，如下所示：

```
drop table dual;
create table dual (dummy string);
```

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

若想忽略前两条语句，直接从第三条语句开始执行，命令格式如下所示：

```
odpscmd -k 3 -f dual.sql
```

获取当前登录用户

若想获取当前登录用户，命令格式如下所示：

```
whoami;
```

示例如下：

```
odps@ hiveut>whoami;
Name: odpstest@aliyun.com
End_Point: http://service.odps.aliyun.com/api
Project: lijunsecuritytest
```

通过以上命令，即可获取当前登录用户的云账号、使用的End_Point配置和项目名。

退出

若想退出客户端，命令格式如下所示：

```
odps@ > quit;
```

您也可输入如下命令退出客户端：

```
odps@ > q;
```

2 MaxCompute Studio

2.1 认识Studio

MaxCompute Studio是阿里云MaxCompute平台提供的安装在开发者客户端的大数据集成开发环境工具，是一套基于流行的集成开发平台*IntelliJ IDEA*的开发插件，帮助您便捷、快速地进行数据开发。本文将为您介绍MaxCompute Studio的功能界面和常用的应用场景。

基本用户界面

MaxCompute Studio是IntelliJ IDEA平台上的一套插件，共享了IntelliJ IDEA的基本开发界面。

MaxCompute Studio在IntelliJ的基础上提供以下功能：

- SQL编辑器（SQL Editor）：提供SQL语法高亮、代码补全、实时错误提示、本地编译、作业提交等功能。

  编译器视图（Compiler View）：显示本地编译的提示信息和错误信息，在编辑器中定位代码。
- 项目空间浏览器（Project Explorer）：连接*MaxCompute*项目空间，浏览项目空间表结构、自定义函数、资源文件。

  表详情视图（Table Details View）：提供表、视图等资源的详情显示和示例数据（Sample Data）。
- 作业浏览器（Job Explorer）：浏览、搜索MaxCompute的历史作业信息。
  - 作业详情视图（Job Details View）：显示作业的运行详细信息，包括执行计划和每个执行任务的详细信息，*Logview*工具能够显示的全部信息。
  - 作业输出视图（Job Output View）：显示正在运行的作业的输出信息。
  - 作业结果视图（Job Result View）：显示SELECT作业的输出结果。
- MaxCompute控制台（MaxCompute Console）：集成了*MaxCompute*客户端，可以输入和执行MaxCompute客户端命令。

连接MaxCompute项目空间

Studio的大部分功能需要您首先创建项目空间连接。建立项目空间连接后，即可在项目空间浏览器中查看相关的数据结构和资源信息。Studio会自动为每一个项目空间连接建立一个本地的元数据备份，以提高对MaxCompute元数据的访问频率和降低延时。

> 说明:
>
> - 您需要先指定项目空间链接，才可通过Studio进行编辑SQL脚本、提交作业、查看Job信息、打开MaxCompute控制台等操作，因此首先创建一个MaxCompute项目空间的连接是非常必要的。
> - MaxCompute项目空间的更多详情请参见项目空间。
> - 在Studio中管理项目空间的更多详情请参见项目空间连接。

管理数据

您可以通过Studio的项目空间浏览器快速浏览项目空间的表结构、自定义函数、资源文件。通过树形控件，可以列出所有项目空间连接下的数据表、列、分区列、虚拟视图、自定义函数名称、函数签名、资源文件及类型等，并支持快速定位。

双击某个数据表，即可打开表详情视图，查看数据表的元信息、表结构和示例数据。如果您没有项目空间的相应权限，Studio会提示对应的错误信息。

Studio集成了*MaxCompute Tunnel*工具，可以支持本地数据的上传和下载，更多详情请参见导入并导出数据。

编写SQL脚本

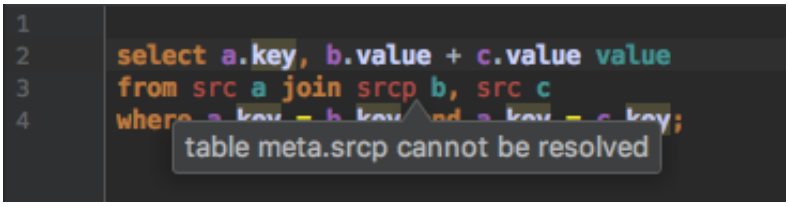
您可以在Studio中编写MaxCompute SQL脚本，非常方便。

1. 打开Studio，导航至File > New > Project或者File > New > Module…。
2. 创建一个MaxCompute Studio类型的项目或者模块。
3. 导航至File > New > MaxCompute Script 或者右击菜单New > MaxCompute Script，即可创建一个MaxCompute SQL脚本文件。

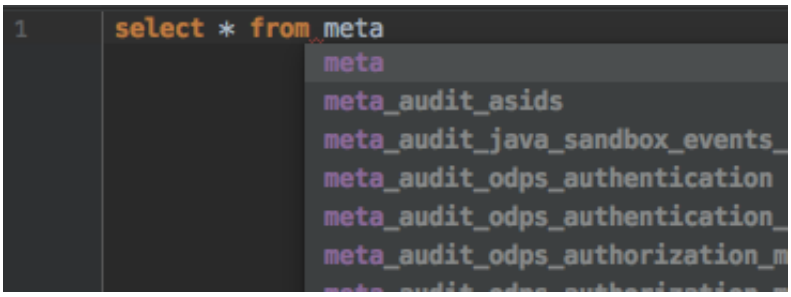
> 说明:
> 创建MaxCompute SQL脚本时，Studio会提示您选择一个关联的MaxCompute项目空间，您也可以通过SQL编辑器上的工具条最右侧的项目空间选取器进行更改，编辑器会根据SQL脚本关联的项目空间对SQL语句自动进行元数据（比如表结构等）的检查并汇报错误，提交运行时也会发送到关联的项目空间执行。更多详情请参见编写*SQL*脚本。

SQL代码智能提示

Studio提供的SQL编辑器可以根据您写入的代码，智能提示SQL语句的语法错误、类型匹配错误或者警告等，实时地标注在代码上。如下图所示：



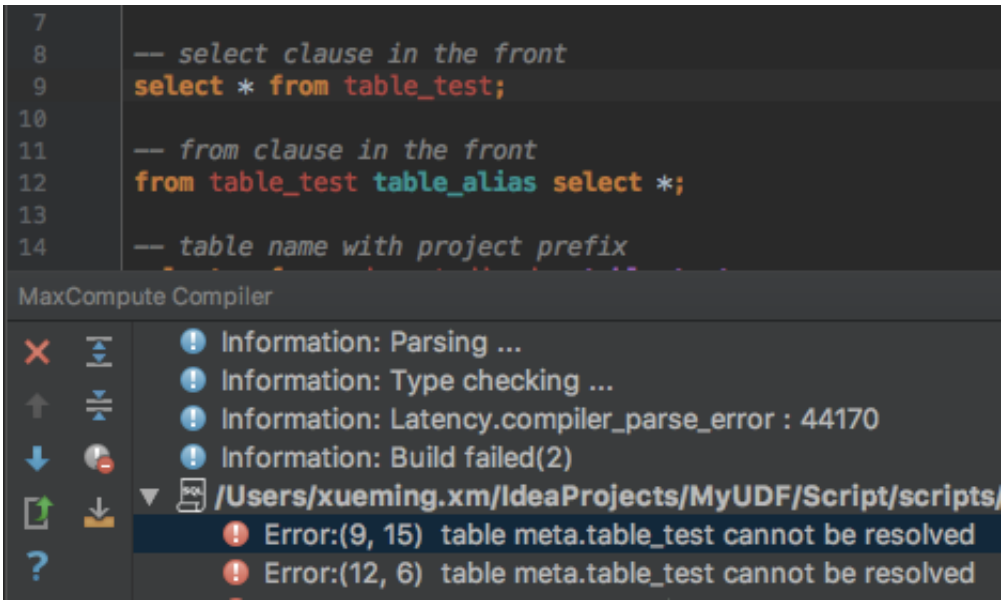
通过代码补全功能，Studio可以根据代码上下文，提示您项目空间名称、表、字段、函数、类型、代码关键词等，并根据您的选择，自动补全代码。如下图所示：



编译和提交作业

- 编译作业

  单击SQL编辑器工具条上的  图标，可以对SQL脚本执行本地编译，如果有语法或者语义错误，编译器窗口会报告错误。



- 提交作业

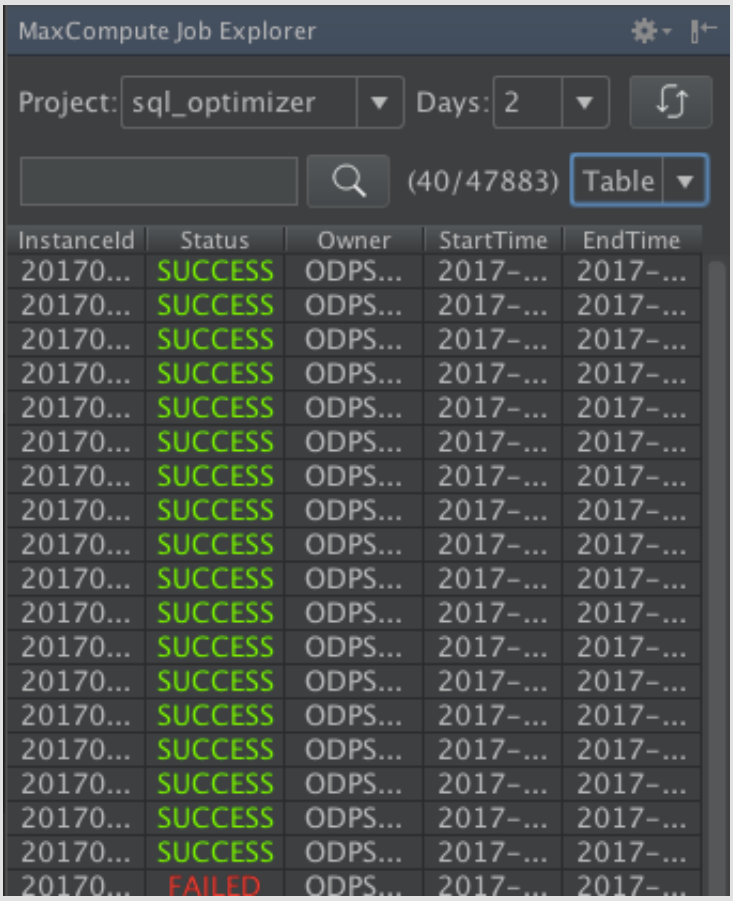
  单击SQL编辑器工具条上的  图标，会在本地编译之后，把SQL脚本提交到MaxCompute指定的项目空间排队执行。

查看历史作业

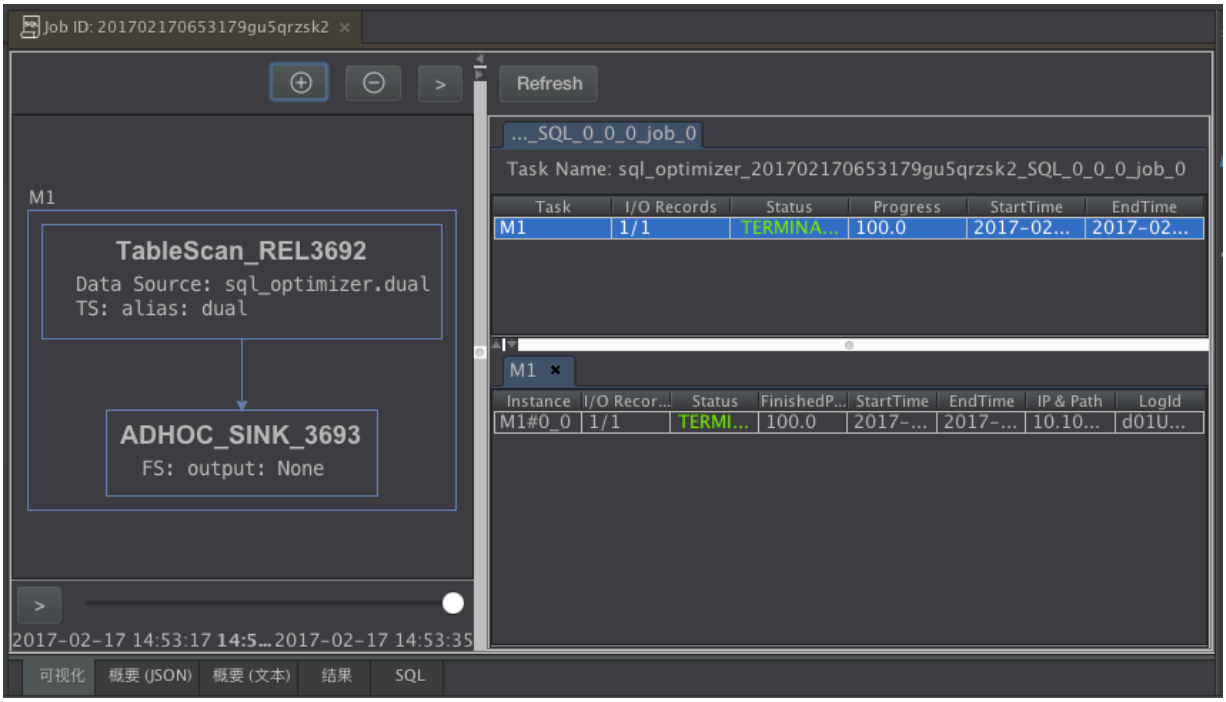
打开作业浏览器，您即可查看指定项目空间上近期执行的作业。

说明:

这个列表只能显示以当前连接使用的用户ID提交的作业。



双击其中一个作业，便可查看作业的的详情信息。如下图所示：



如果知道一个任务的Logview URL，可以导航至MaxCompute > Open Logview，打开该任务的详情页面。

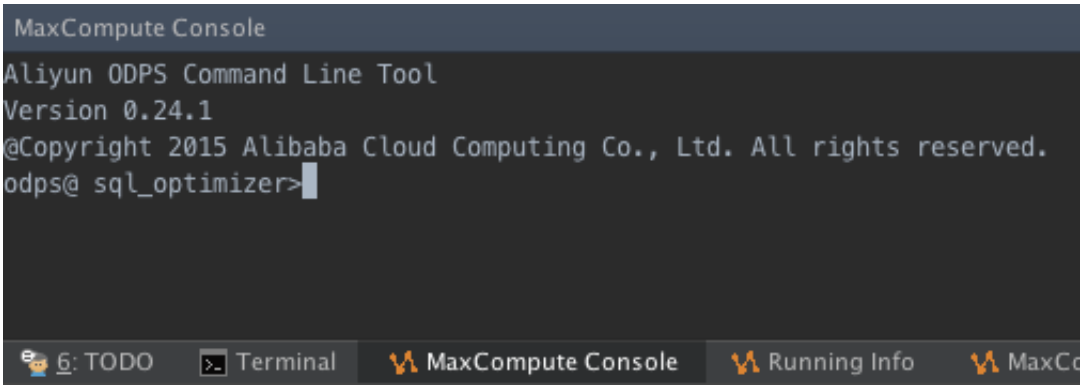
开发MapReduce和UDF

Studio还支持*MapReduce*和*Java UDF*的开发。

连接MaxCompute客户端

Studio集成了最新版本的MaxCompute客户端，您也可以在Studio的配置页面 中指定本地已经安装好的MaxCompute客户端路径。

您在项目空间浏览器中选定一个项目空间，右键单击菜单选择Open in Console即可打开MaxCompute控制台窗口。



后续步骤

现在，您已经学习了MaxCompute Studio的功能界面和常用的应用场景，您可以继续学习下一个教程。在该教程中您将学习如何安装MaxCompute Studio。详情请参见安装*IntelliJ IDEA*。

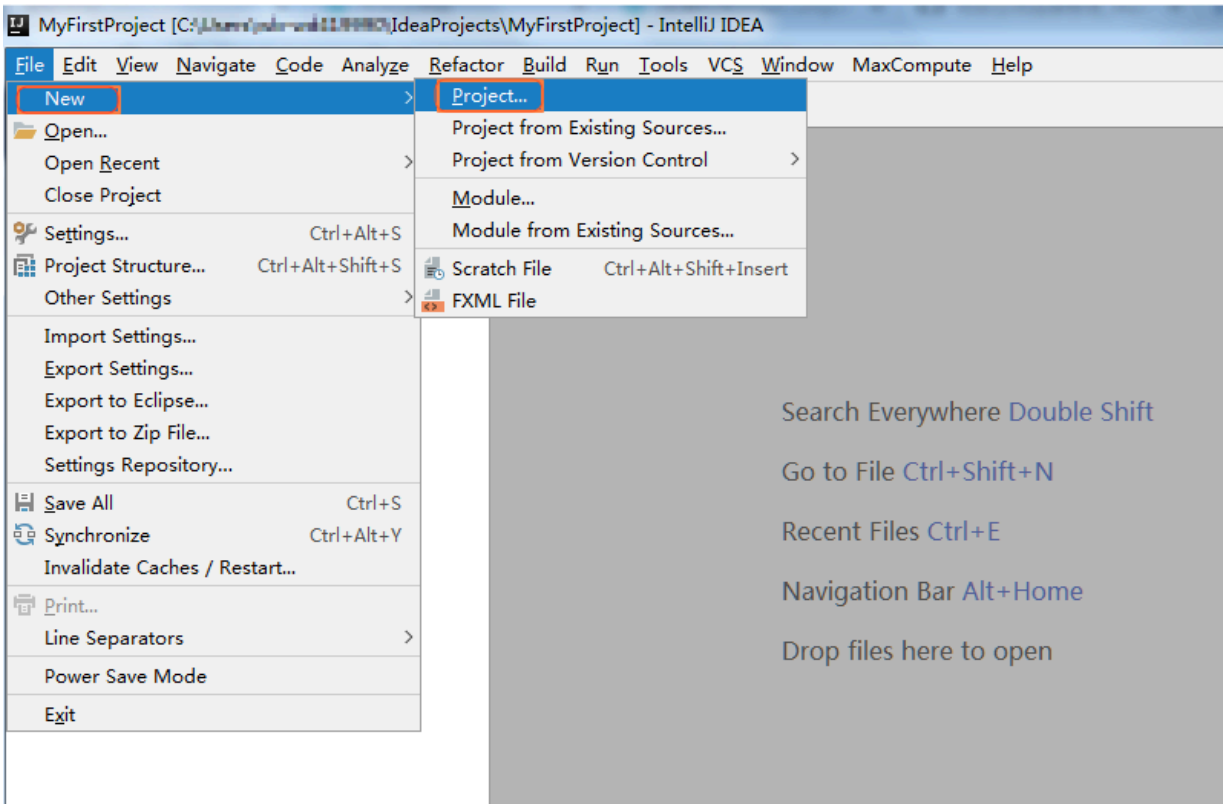
2.2 项目空间连接管理

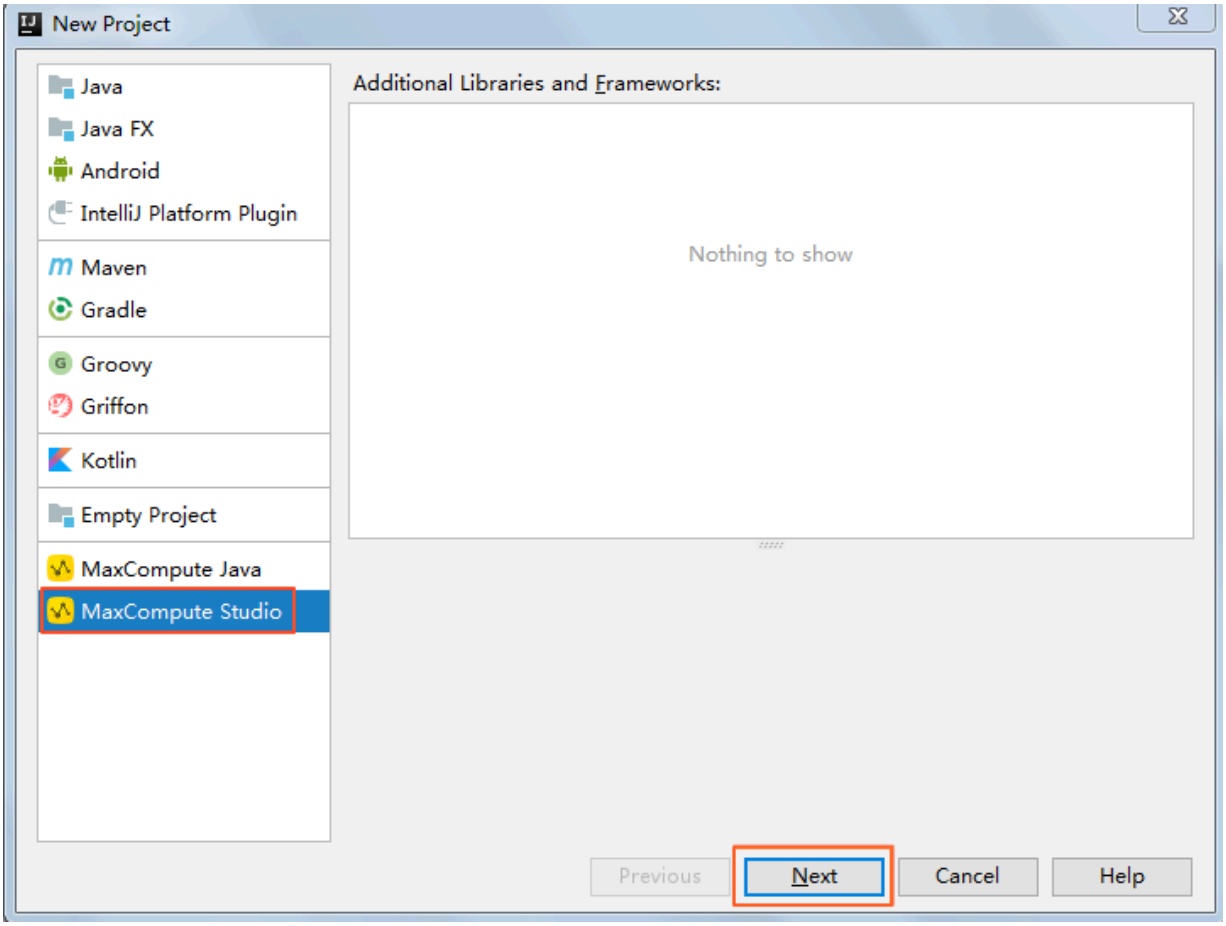
MaxCompute Studio的一个核心功能是浏览MaxCompute项目空间（Project）的资源，包括*Table*、*UDF*、*Resource*等。要想实现这一功能，首先需要创建项目空间连接。

前提条件

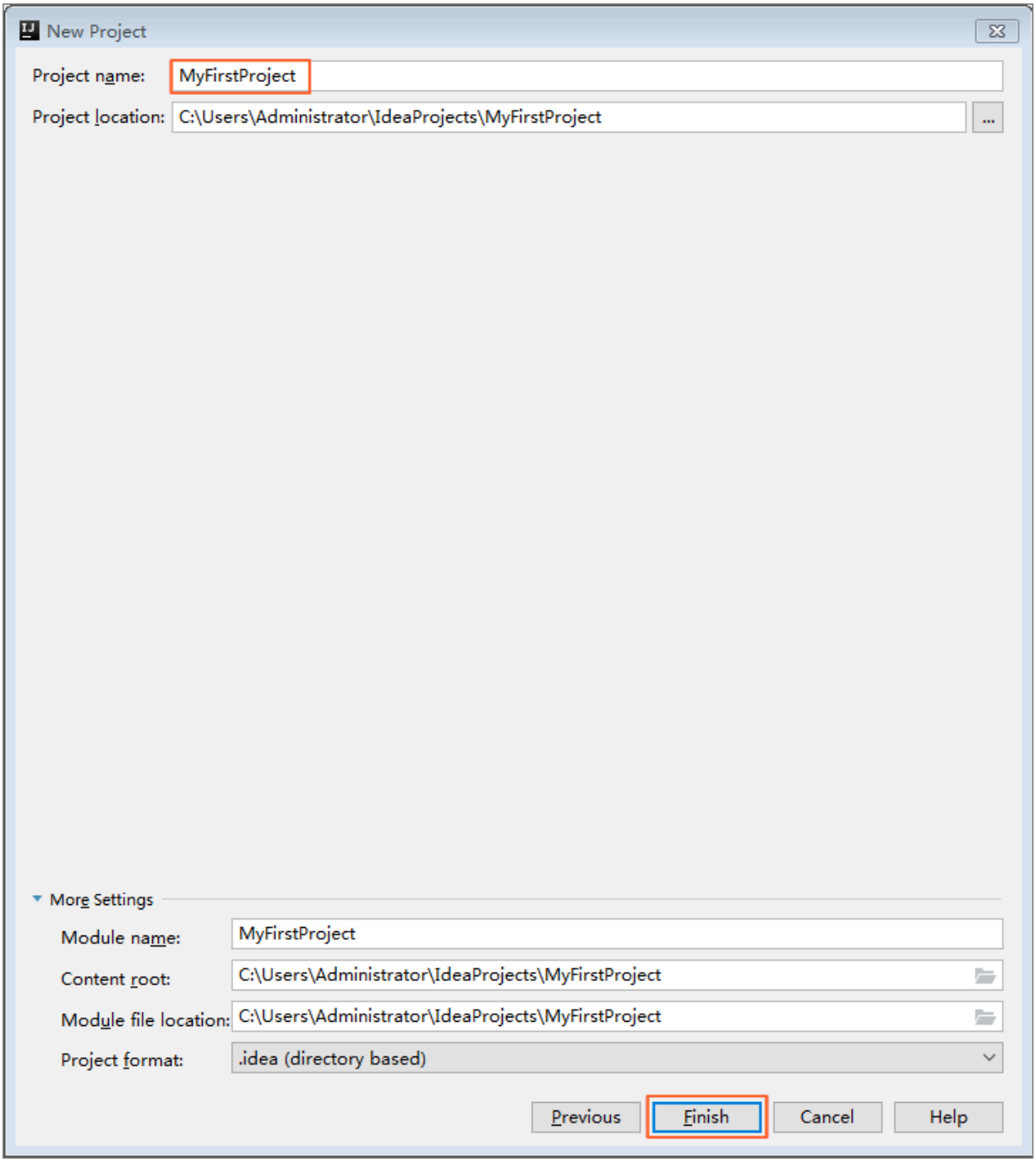
由于使用IntelliJ的Tool Windows必须先打开某个IntelliJ project，而配置MaxCompute Project需要进入IntelliJ界面Tool Windows中的MaxCompute Project Explorer，所以在创建MaxCompute Project连接前，您可以首先添加或者导入一个IntelliJ project。本文将以在Windows下新增project为例。

运行IntelliJ IDEA后，单击New > Project，选择弹出页面中的MaxCompute Studio，单击Next。





填写Project name，单击Finish。



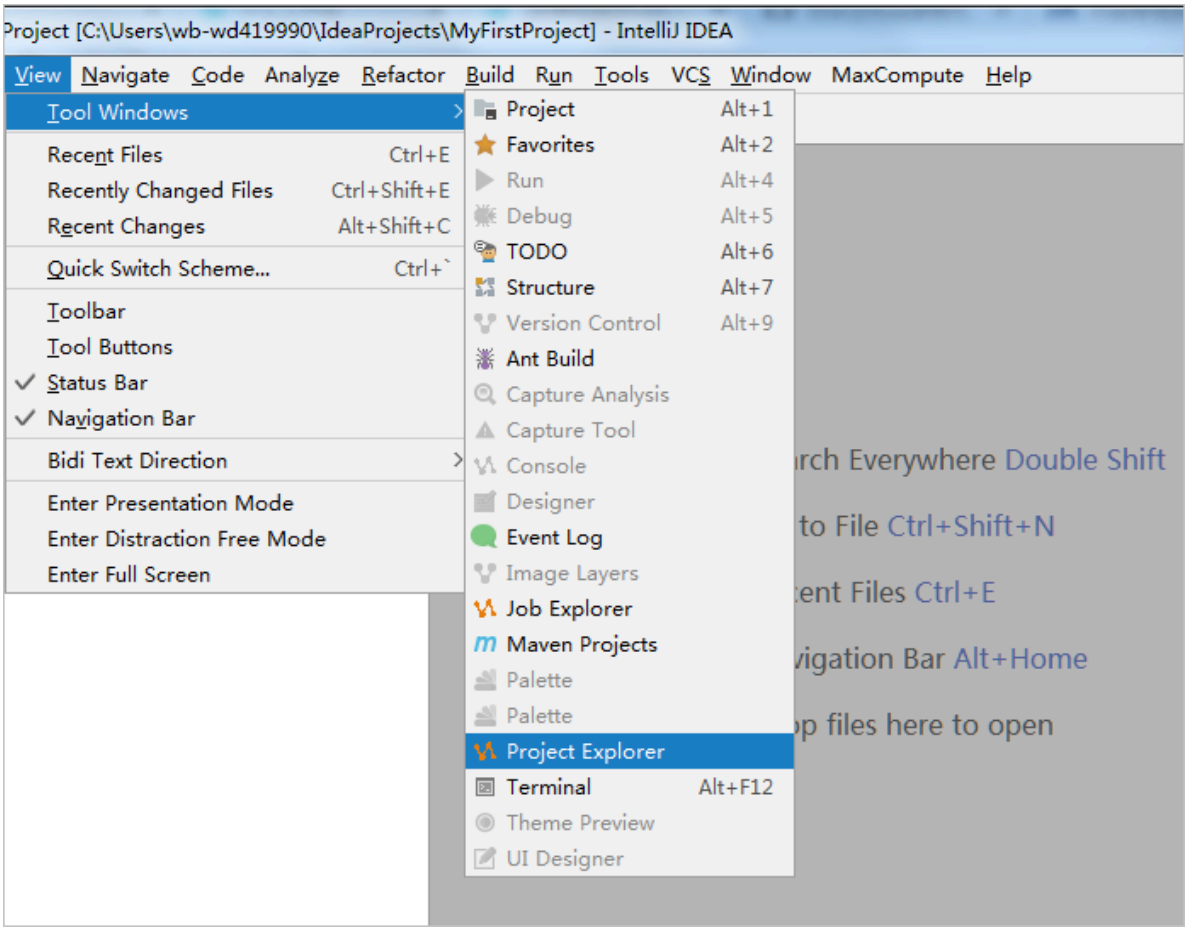
创建MaxCompute项目链接

建议您根据自己的Region配置MaxCompute项目连接，否则会出现无法访问等错误。您可以参考配置*Endpoint*配置自己的Endpoint和Region。

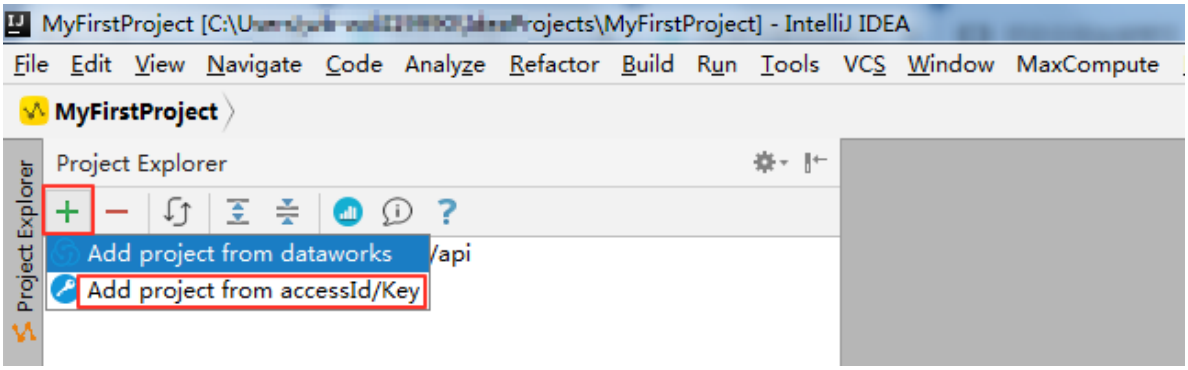
操作步骤：

1. **单击菜单中的View选项，选择Tool Windows。**

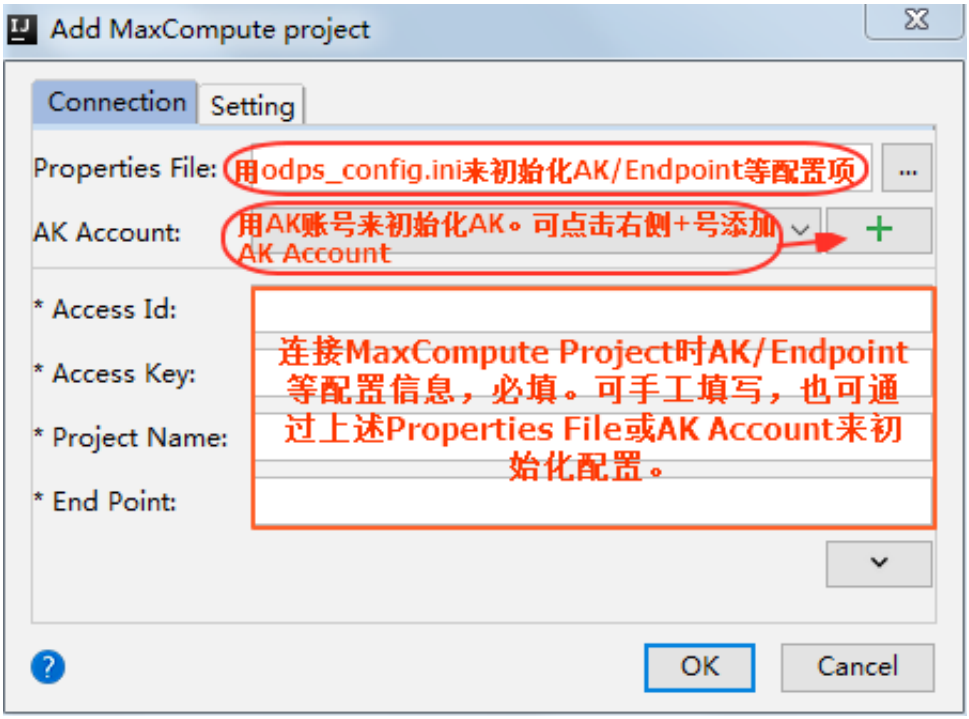
2. 单击弹出页面中的Project Explorer



3. 单击左上角的+，选择Addprojectfromaccessid/Key模式，添加一个MaxCompute Project。

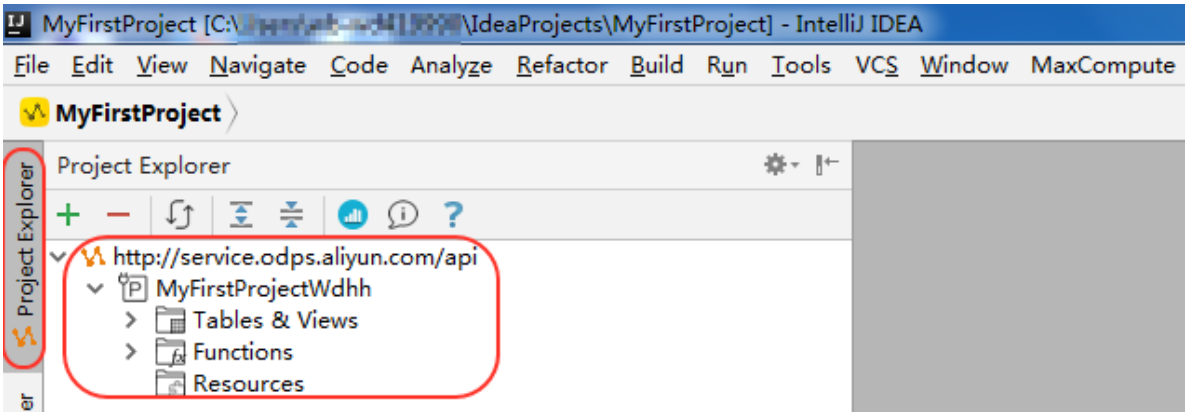


4. 在Add MaxCompute Project对话框中，填入相关配置选项。



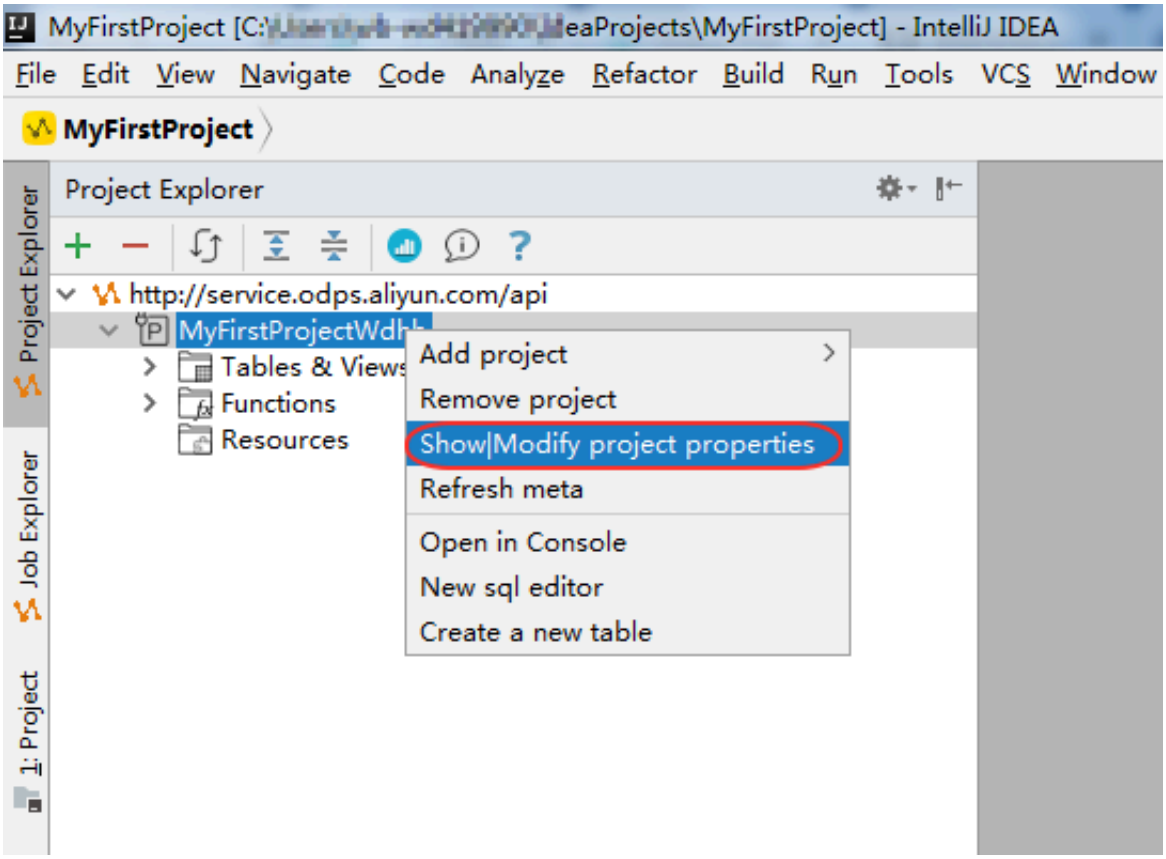
> **说明:**
>
> · 单击dialog左下角的?即可链接到在线文档。
>
> · 若同步出现超时错误，可以在setting标签页酌情延长元数据同步到本地的超时时间。

5. 配置完成后，单击OK。左侧MaxCompute Project Explorer中会显示MaxCompute Project的信息，可以通过鼠标单击查看该project中的表、视图、函数以及资源等信息。



查看/修改MaxCompute项目链接

在Project Explorer中，对需要修改的MaxCompupte项目右键选择Show|Modify project properties。



在弹出框中可以查看或修改该MaxCompute project的Connection和Setting。

后续操作

现在，您已经学习了如何新建、管理项目空间连接，您可以继续学习下一个教程。在该教程中您将学习如何进行元数据查询、清理数据、上传下载数据等操作，来管理数据和资源。详情请参见 管理数据和资源。

2.3 工具安装与版本信息

2.3.1 安装IntelliJ IDEA

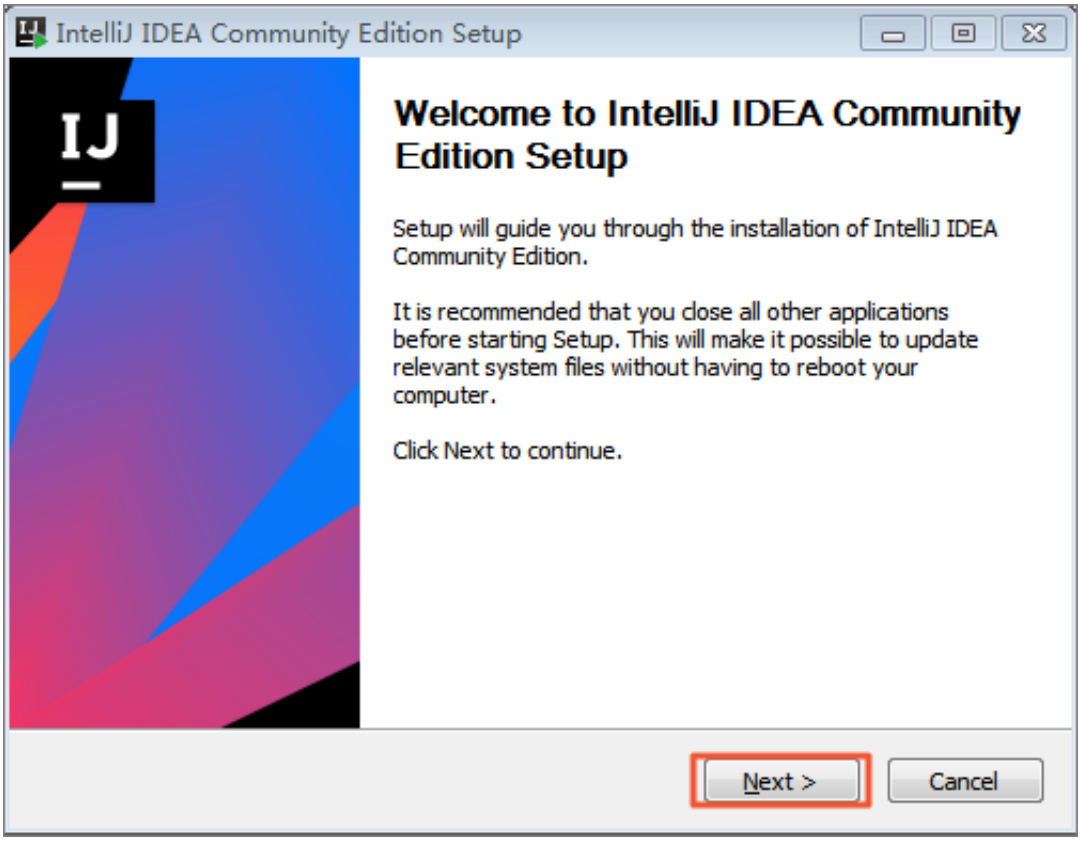
本文将为您介绍如何安装MaxCompute Studio的基础平台IntelliJ IDEA。

操作步骤

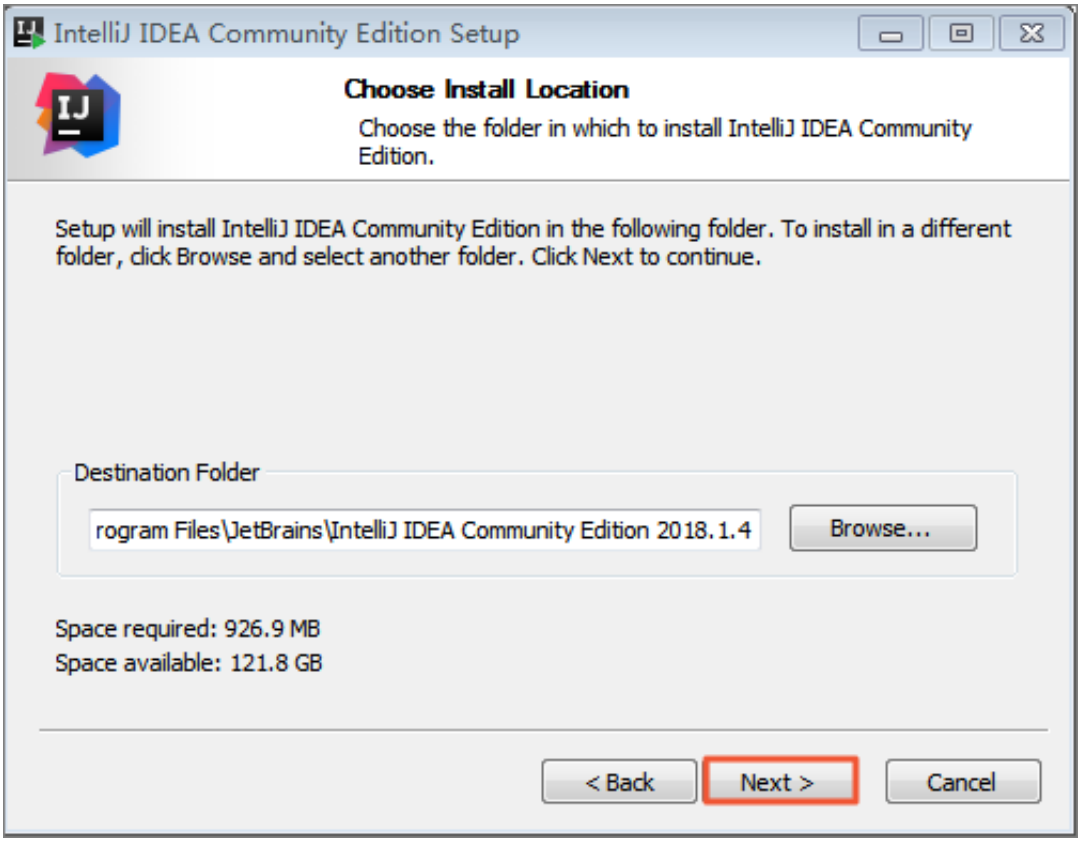
1. 单击 此链接，根据操作系统（Windows、macOS、Linux）下载对应IntelliJ IDEA版本。此处我们以Windows操作系统安装为例。

   下载IntelliJ IDEA 14.1.4以上版本 （支持Ultimate版本或免费的Community版本，PyCharm也被支持）

2. 下载完成后，双击安装程序，进入安装界面，单击Next。如下图所示：



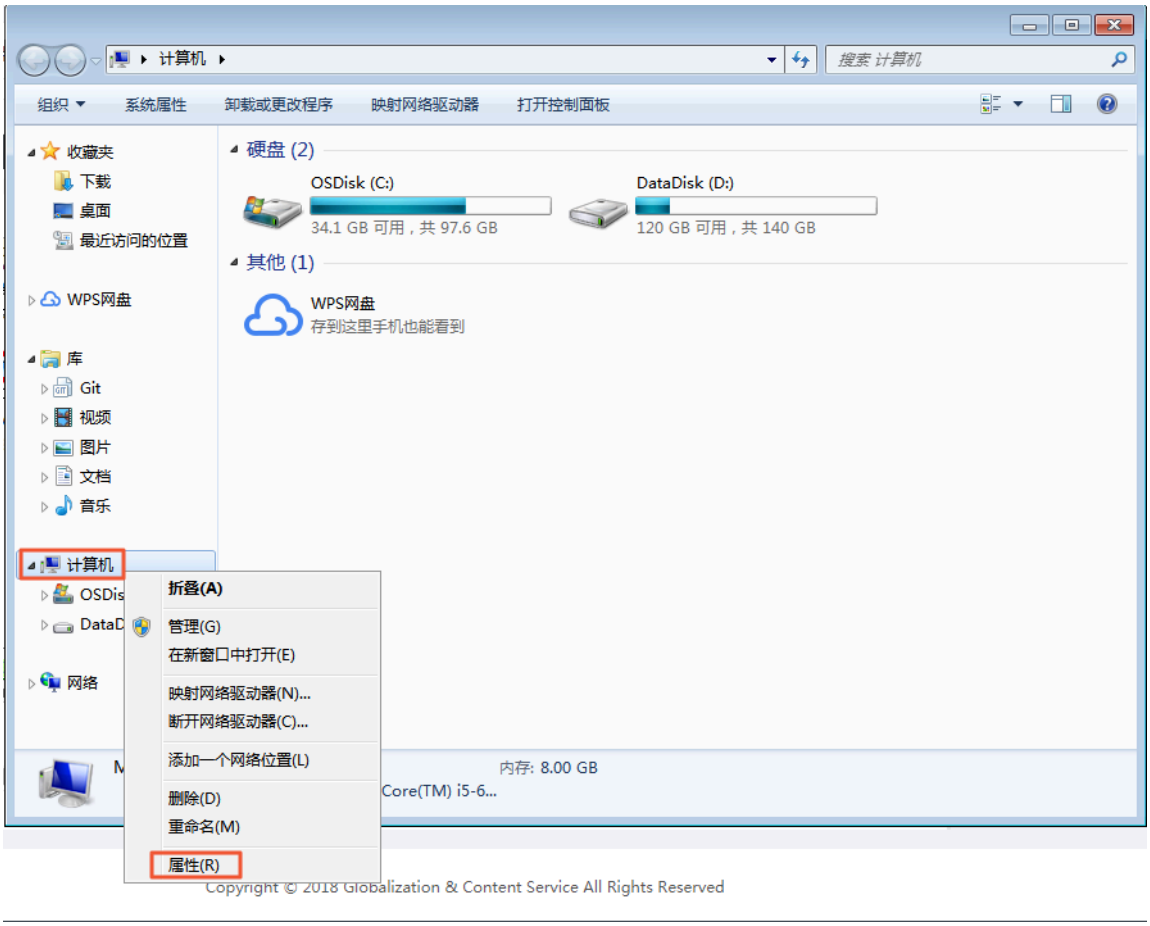
3. 指定安装目录后，单击Next。如下图所示：



4. 根据本地操作系统的版本选择需要安装32位或者64位的IntelliJ IDEA。

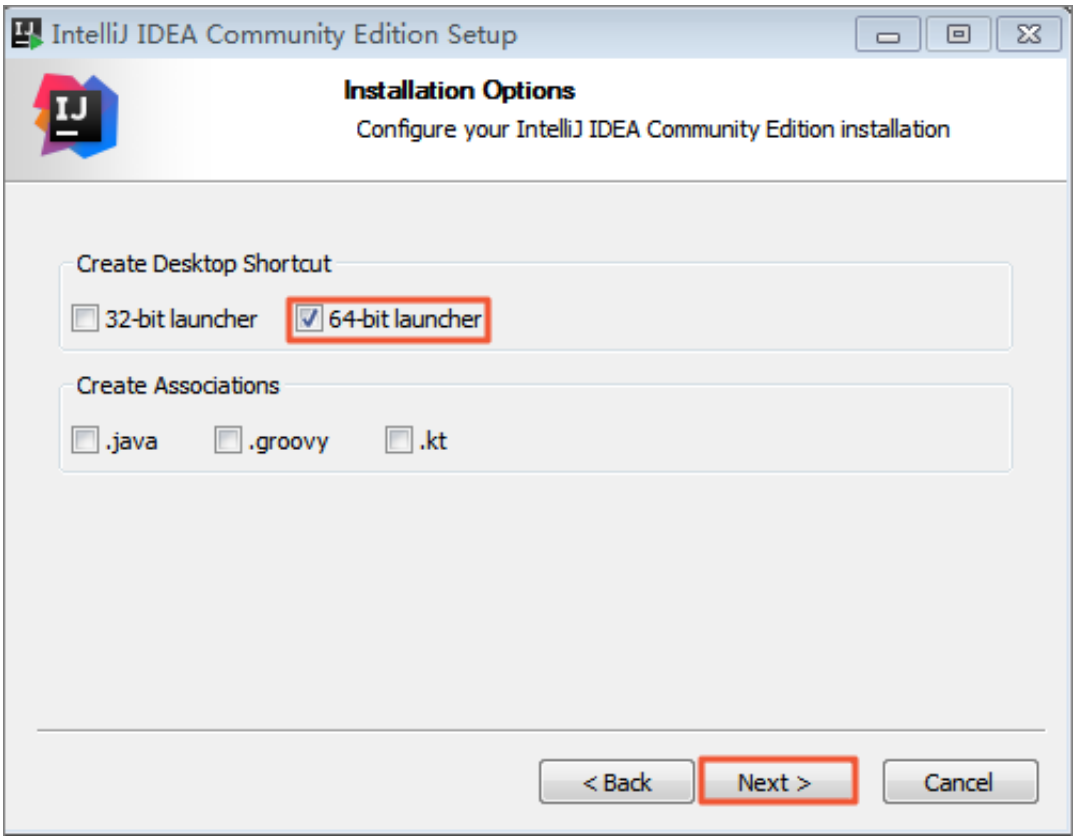
   如何查看本地操作系统的版本:

   a) 打开windows资源管理器，右击计算机，选择 属性。如下图所示：

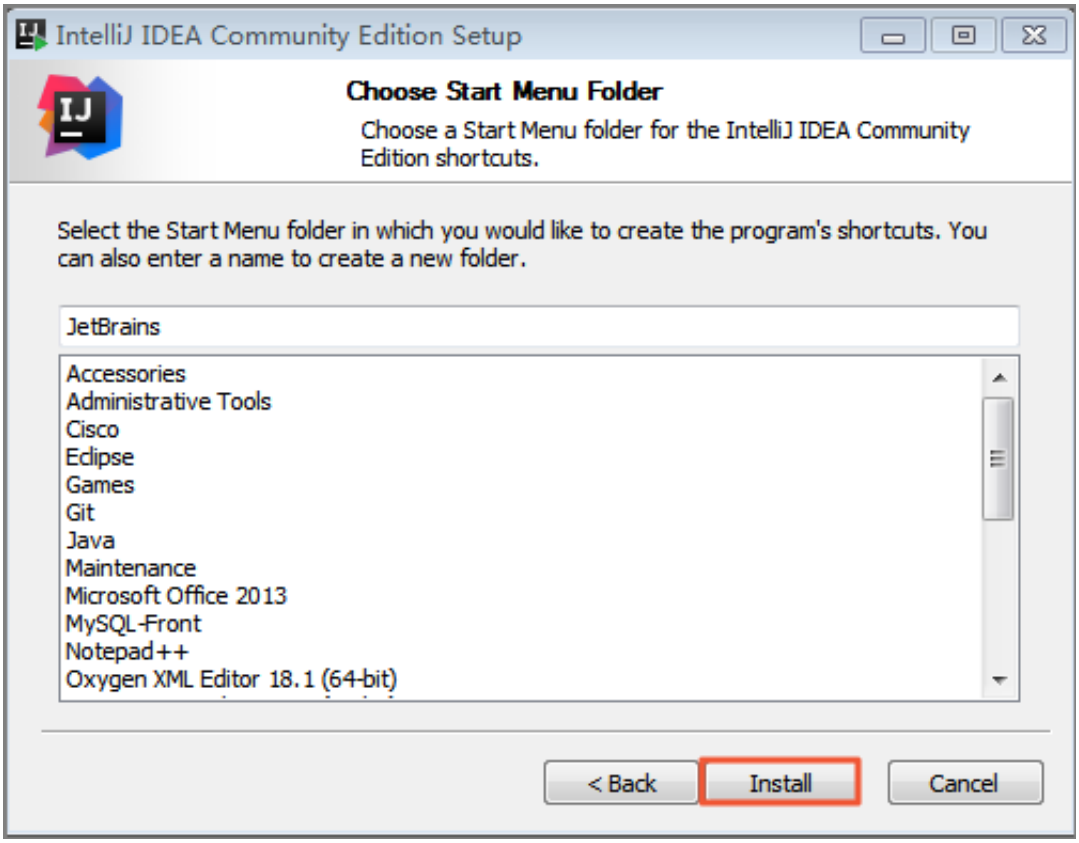


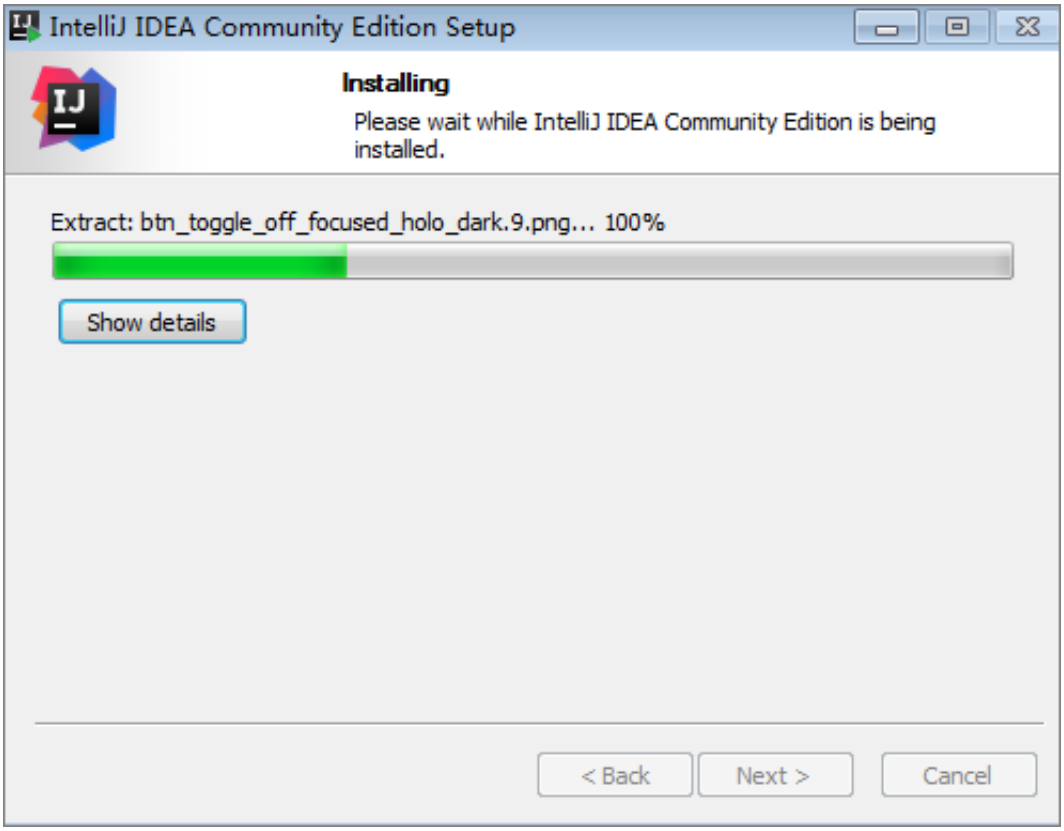
   b) 在弹出界面中查看操作系统位数，如下图所示：

5. 选择相应的系统类型，单击Next 。如下图所示：



6. 单击Install，开始安装。如下图所示：





7. 安装完成后，单击Finish 。

后续操作

现在，您已经学习了如何安装IntelliJ IDEA，您可以继续学习下一个教程。在该教程中您将学习如何安装MaxCompute Studio插件。详情请参见安装 *Studio* 插件。

2.3.2 IntelliJ IDEA插件安装步骤

MaxCompute Studio是IntelliJ IDEA的插件，本文为您介绍如何安装MaxCompute Studio。

环境要求

IntelliJ IDEA支持在*Windows*， *Mac*， *Linux*操作系统上安装，硬件及系统环境要求请参见 *Requirements for IntelliJ IDEA* 。基于IntelliJ IDEA平台的MaxCompute Studio也可以安装在这些操作系统的客户端上。

MaxCompute Studio对用户环境有以下要求：

- Windows，Mac OS，或者Linux系统客户端。
- 安装IntelliJ IDEA v18.2.4以上版本 （支持Ultimate版本、PyCharm版本和免费的 *Community* 版本）。
- 已安装JRE 1.8 （最新的IntelliJ IDEA版本捆绑了JRE 1.8）。
- 已安装JDK 1.8 （可选：如果需要开发和调试Java UDF，则需安装 JDK）。

> **说明:**
> 客户端从0.28.0版开始支持JDK 1.9，在这之前的版本只支持JDK 1.8。

安装方式

MaxCompute Studio是IntelliJ IDEA的插件，有以下两种安装方式：

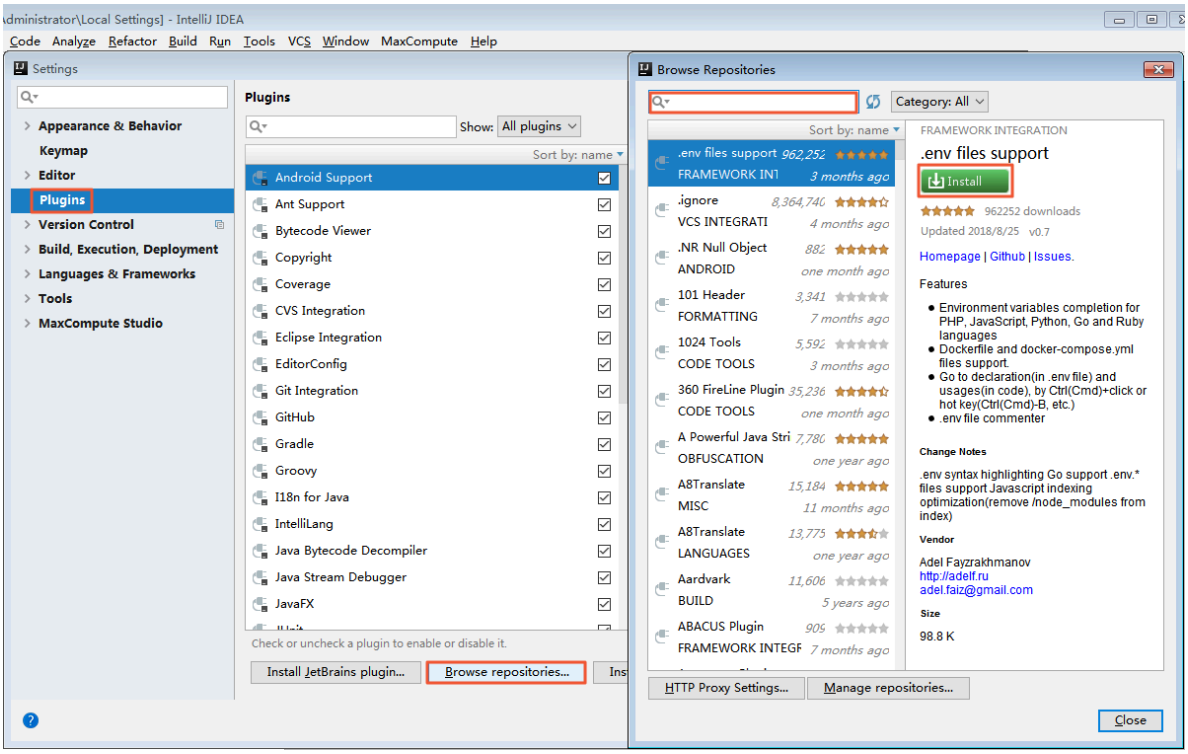
- 通过插件库在线安装（推荐）。
- 通过本地文件安装。

在线安装（推荐）

MaxCompute Studio插件已对全部公网用户开放，您可以通过IntelliJ官方插件库安装。

操作步骤

1. 在IntelliJ IDEA中打开插件配置页面（Windows/Linux用户导航至 File > Settings > Plugins，Mac用户导航至IntelliJ IDEA > Preferences > Plugins）。
2. 单击Browse repositories…按钮，然后搜索`MaxCompute Studio`。
3. 找到MaxCompute Studio插件页面，单击绿色Install按钮进行安装。
4. 确认安装后，重新启动IntelliJ IDEA，完成安装。



本地安装

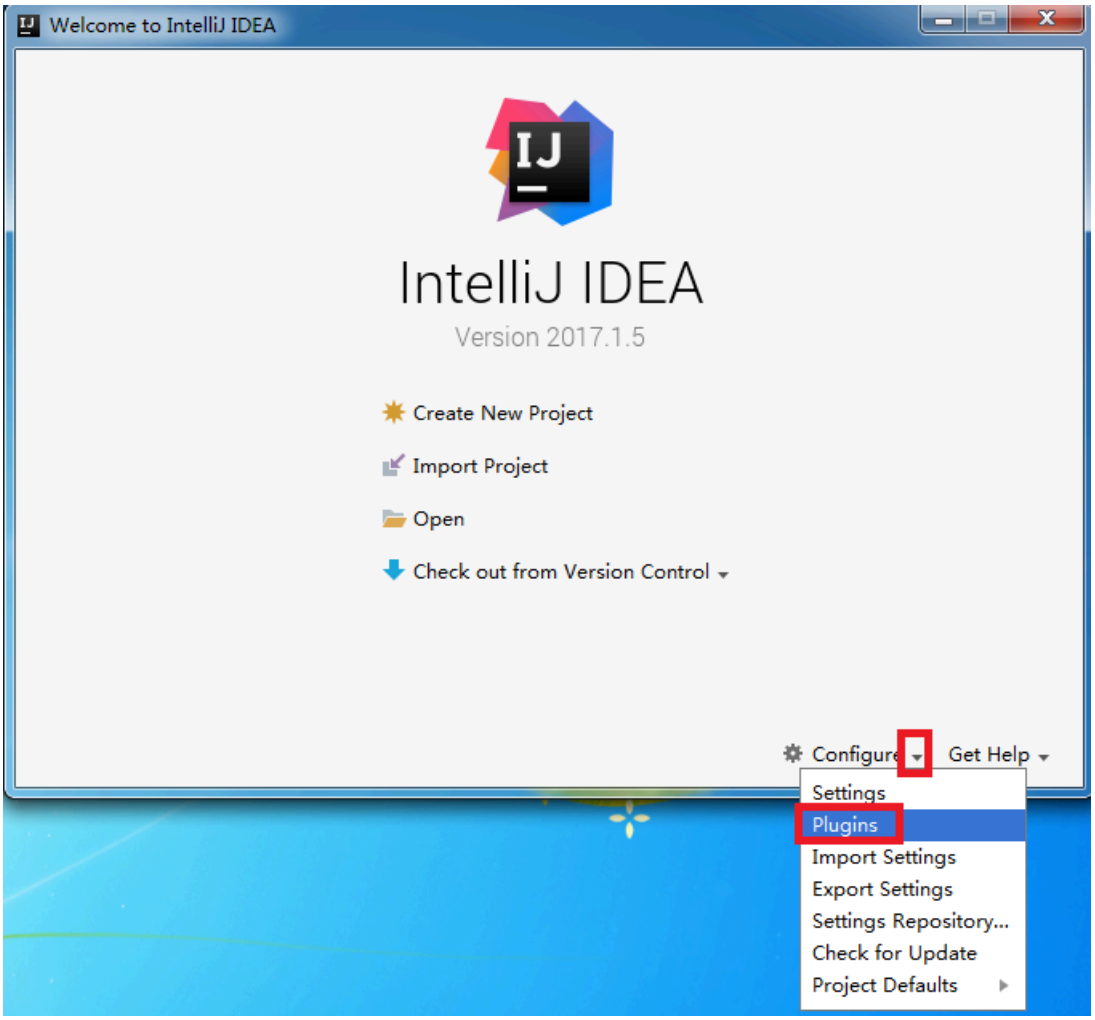
MaxCompute Studio也可以在本地环境中进行安装。

操作步骤

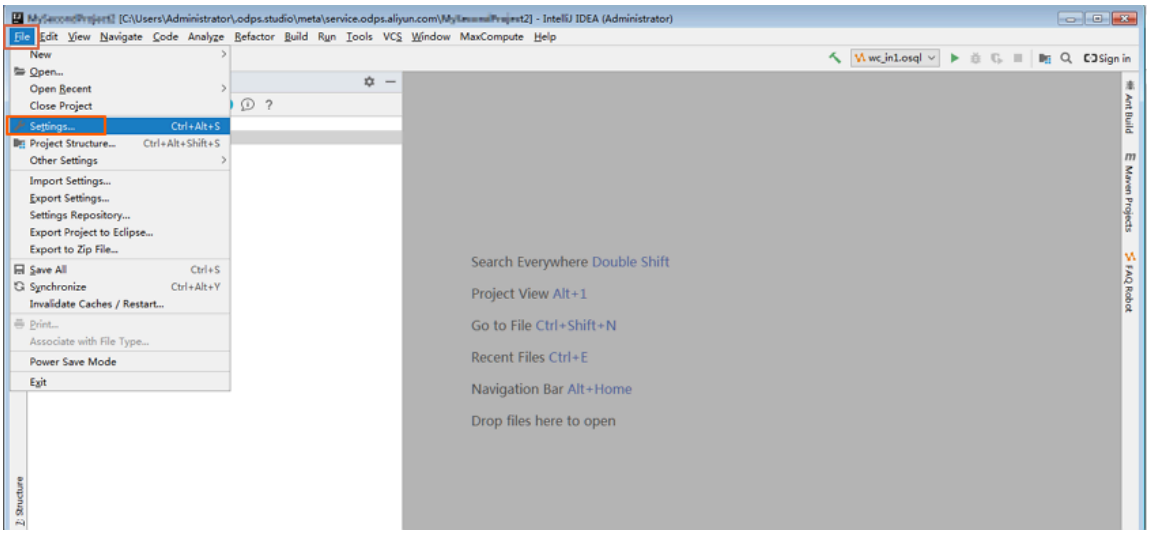
1. 进入*MaxCompute Studio* 插件页面下载插件包。

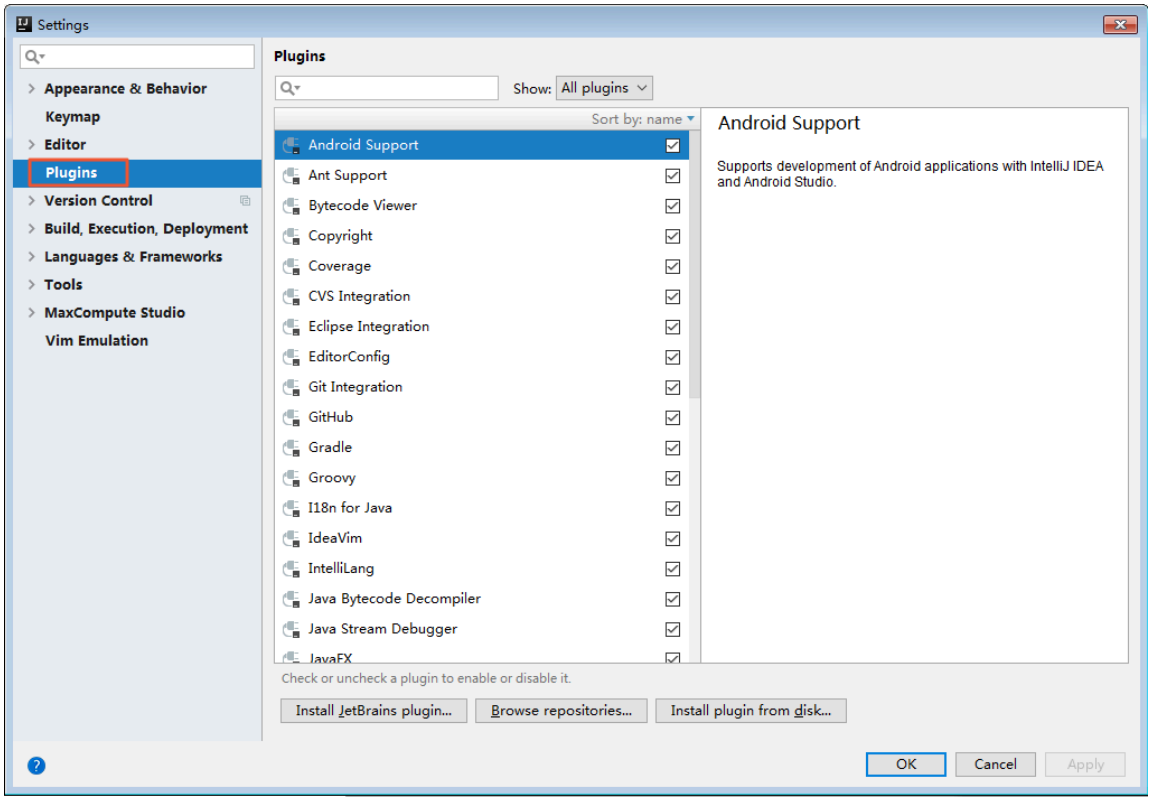
2. 运行IntelliJ IDEA。

- 如果是第一次运行，会出现欢迎界面，单击欢迎界面中的configure（配置），选中弹出菜单中的Plugins（插件），如下图所示：

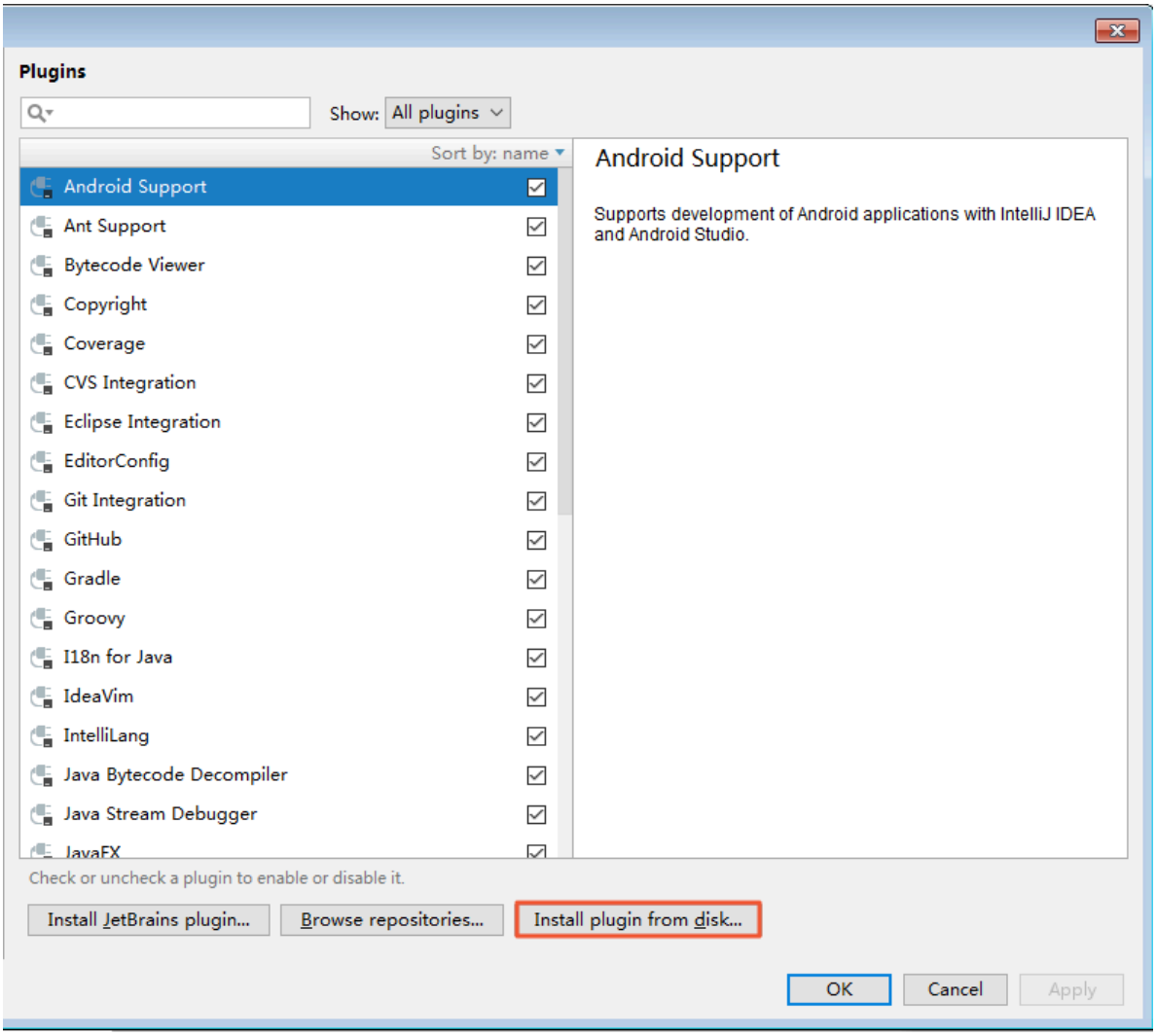


- 如果不是第一次运行，可以依次单击菜单File > Settings > Plugins进入相同的界面，如下图所示：

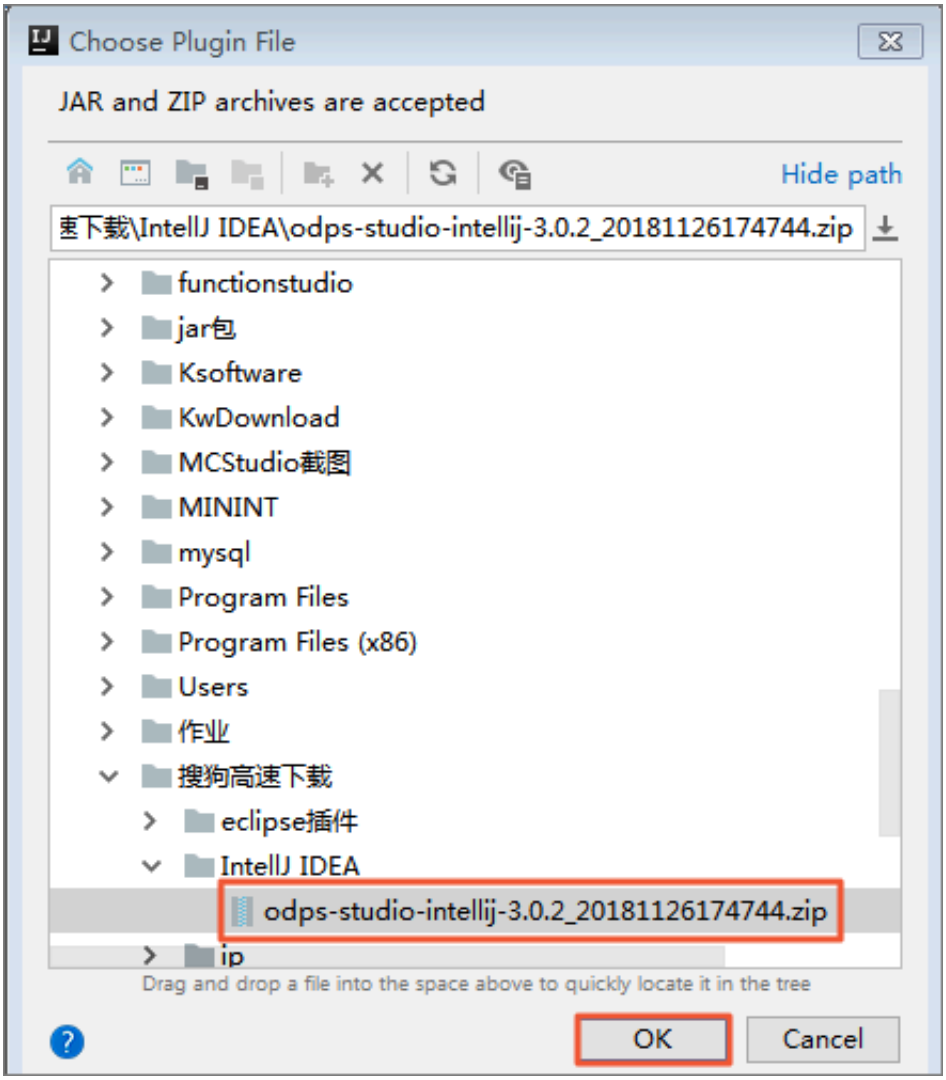



Settings
Appearance & Behavior
Keymap
Editor
Plugins
Version Control
Build, Execution, Deployment
Languages & Frameworks
Tools
MaxCompute Studio
Vim Emulation
Plugins
Show: All plugins
Sort by: name
Android Support
Ant Support
Bytecode Viewer
Copyright
Coverage
CVS Integration
Eclipse Integration
EditorConfig
Git Integration
GitHub
Gradle
Groovy
I18n for Java
IdeaVim
IntelliLang
Java Bytecode Decompiler
Java Stream Debugger
JavaFX
Android Support
Supports development of Android applications with IntelliJ IDEA and Android Studio.
Check or uncheck a plugin to enable or disable it.
Install JetBrains plugin...
Browse repositories...
Install plugin from disk...
OK
Cancel
Apply

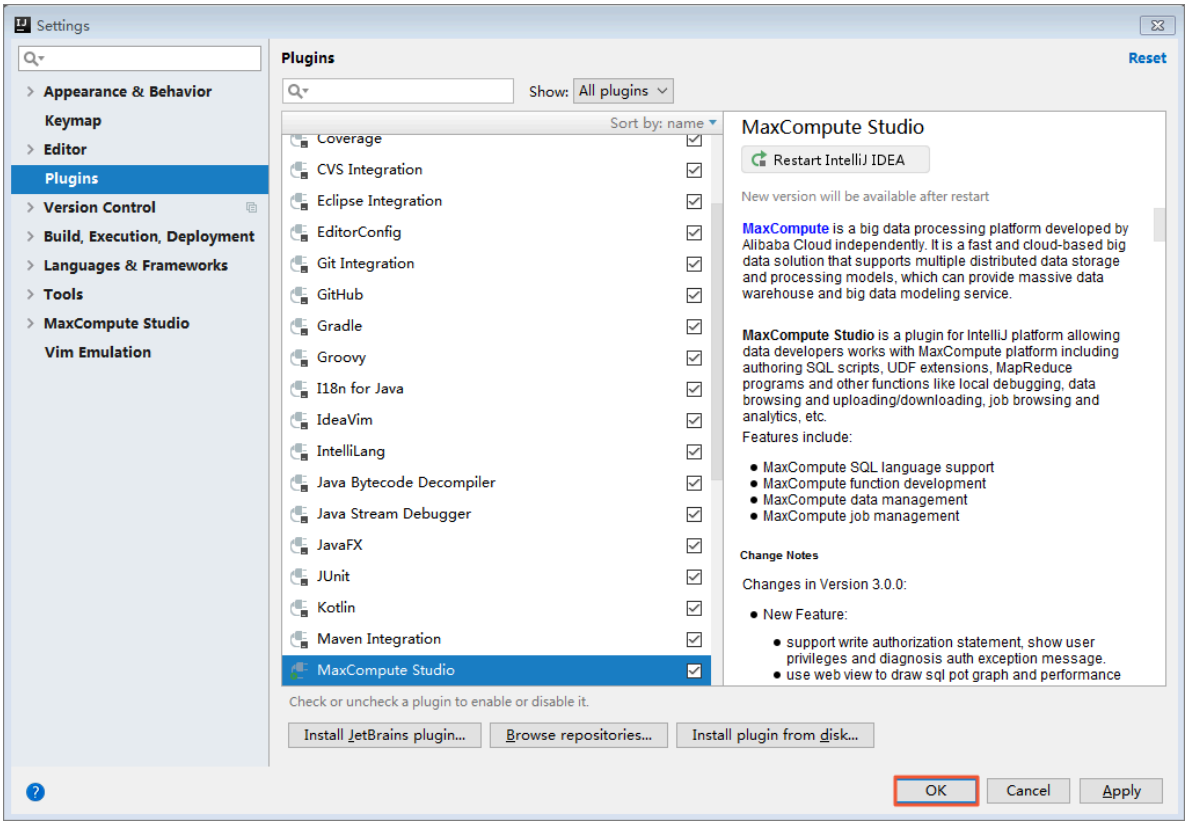
3. 在插件页面，单击Install plugin from disk…（从本地磁盘安装插件），如下图所示：



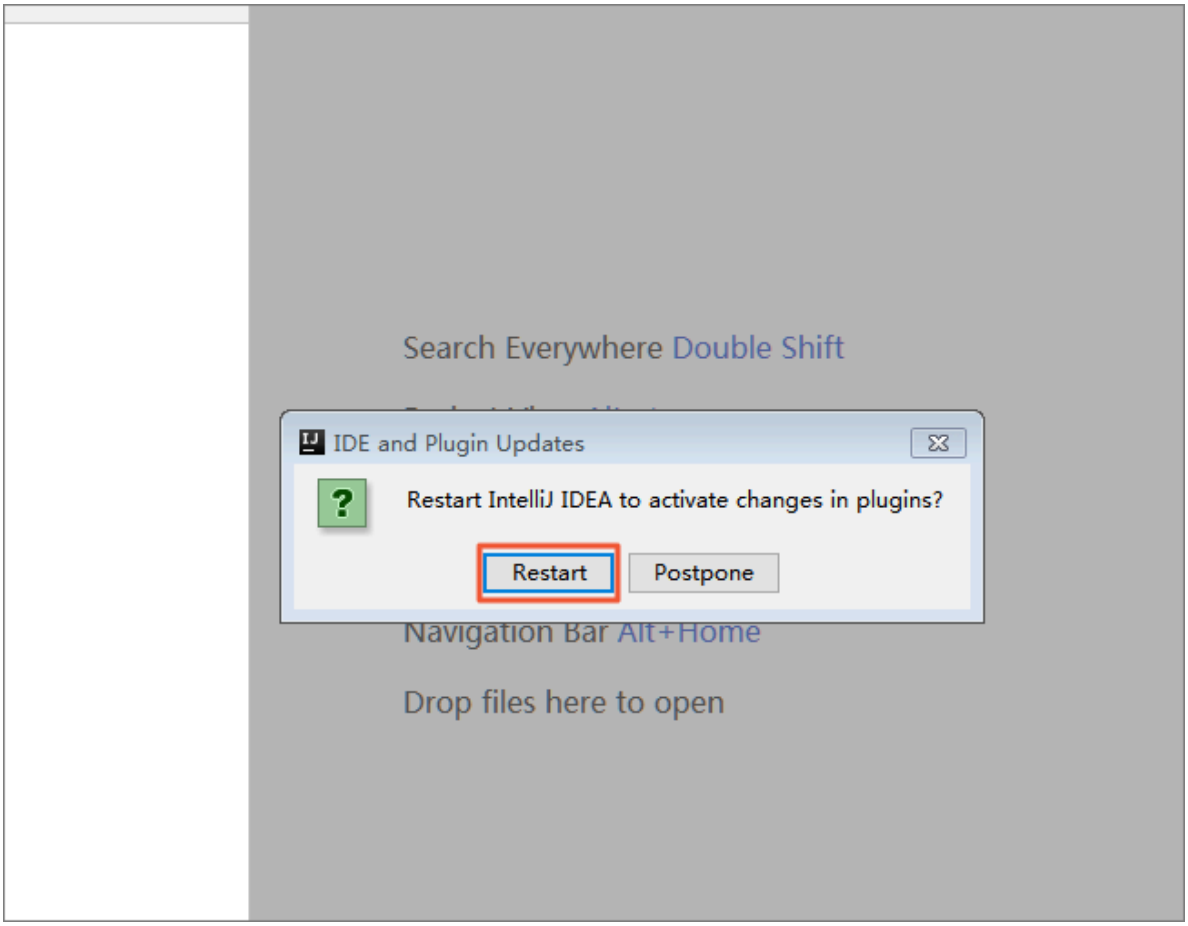
4. 在弹出窗口中，通过单击目录名称前的灰色图标进行导航，找到插件文件并选中，单击OK。



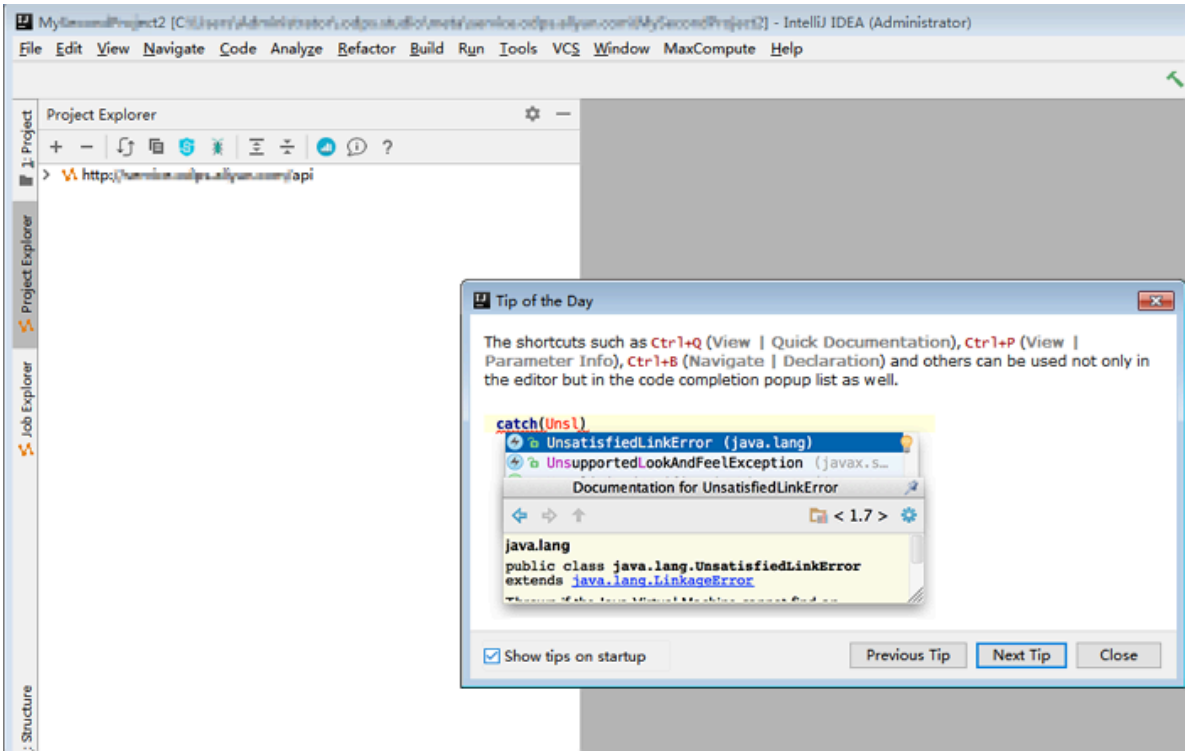
5. 回到插件首页后，单击OK，开始安装本地插件。



6. 安装完成后，弹出重新启动的提示窗口，单击Restart，重新启动IntelliJ IDEA。



7. 重新启动后，界面如下所示：



后续步骤

现在，您已经学习了如何安装MaxCompute Studio插件，您可以继续学习下一个教程。
在该教程中您将学习如何配置MaxCompute Project连接管理数据和资源。详情请参见 新建MaxCompute项目空间连接。

2.3.3 查看和更新版本

查看Studio版本信息

通过以下步骤查看Studio的版本信息：

1. 打开Settings/Preferences页面 （Windows `Ctrl-Alt-S`, Mac ⌘,）
2. 在对话框左侧边栏选择Plugins，然后搜索*MaxCompute Studio*
3. 可以看到MaxCompute Studio的版本号，以及版本的发布信息。

也可以在Setting页面左侧边栏选择MaxCompute Studio， 再找到当前版本号。

检查新版本

缺省配置下，MaxCompute Studio会自动检测新版本，当有新的可用版本时，会自动通知用户。

收到更新提示后，用户可以选择：

- 安装： 点击更新提示中的安装链接，将会自动下载并安装此新版本，安装完成后重启IntelliJ IDEA
- 配置： 点击更新提示中的配置链接，你可以配置是否自动检查新版本

如果关闭了自动更新功能，用户通过以下步骤可以检查MaxCompute Studio的版本更新并选择安装：

1. 打开Settings/Preferences页面 （Windows `Ctrl-Alt-S`, Mac ⌘,）。
2. 在对话框左侧边栏选择MaxCompute Studio 。
3. 在Studio配置页面上，点击按钮Check new versions 。
4. 如果检测到新的可用版本，会提示用户新的版本号。 点击按钮Install new version可以安装，重新启动 IntelliJ 完成安装。

可以通过Automatically checks for new versions复选框控制自动检查版本更新的开关。

> **说明:**
> Studio当前最新版本对应的JAVA SDK版本是0.28.4及以上。

下一步

建立MaxCompute项目连接

2.4 管理数据和资源

2.4.1 浏览表及 UDF

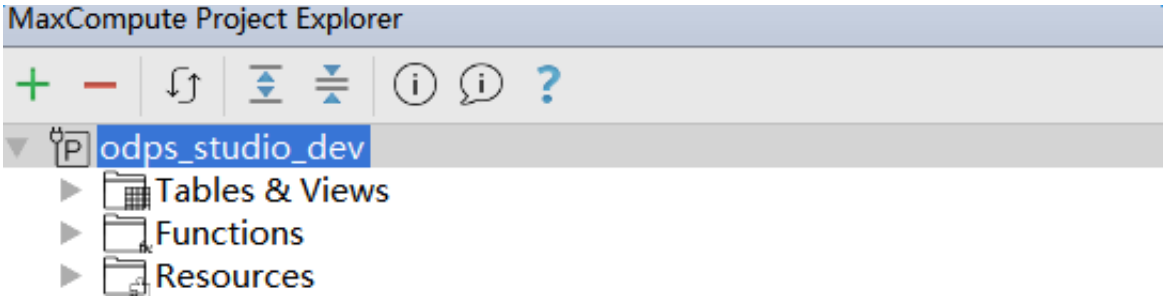
查看表及函数

在项目空间浏览器（Project Explorer)窗口中可以快速浏览已添加连接的表、函数、资源等。使用前提是添加MaxCompute项目连接。

浏览表和函数

要浏览项目空间的表和函数，使用以下步骤：

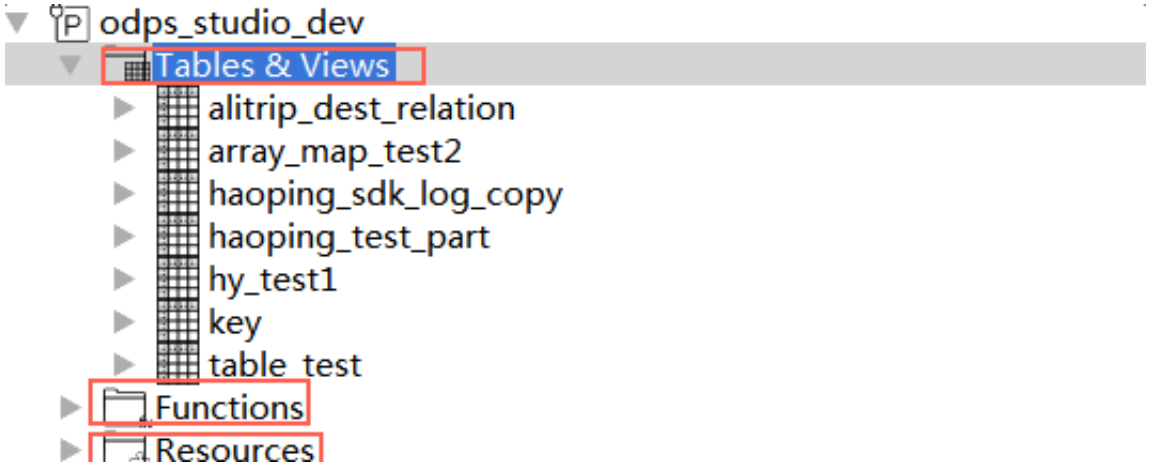
1. 打开项目空间浏览器（Project Explorer)，即可以查看已添加的 Project节点树。



节点树上方是工具栏，包括：

- 增加Project：新增一个到MaxCompute项目空间的连接
- 删除Project：删除一个项目空间浏览器中的项目连接，对服务器端项目空间无影响
- 刷新元数据：从服务器端项目空间刷新元数据信息，刷新本地元数据缓存
- 展开节点：展开全部树节点
- 折叠节点：折叠全部树节点
- 用户反馈：提交用户反馈
- 在线文档：打开在线文档

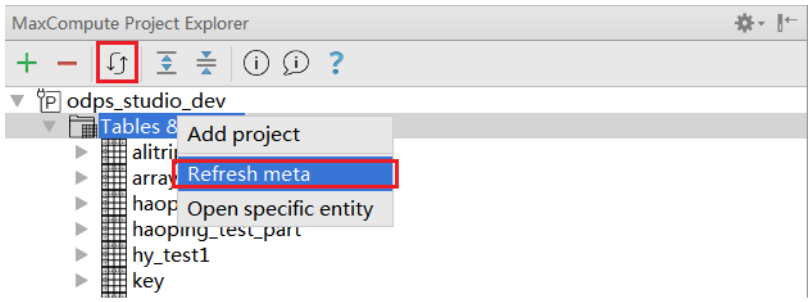
2. 双击或点击下拉箭头展开Tables节点，可列出该项目下的所有表（包括虚拟视图）。这里的表名列表与用户执行show tables命令相等，需要用户在project下有list table权限。函数（Functions）和资源（Resources）节点类似:



3. Studio会将服务上的项目元数据下载到本地，当服务端元数据有更新时，如新增了一张表，需手动触发一次刷新，将变化的元数据重新加载到本地。 可以选择在项目（Project）或表（Table）级别做刷新，步骤如下:

   a. 选中相应的节点。

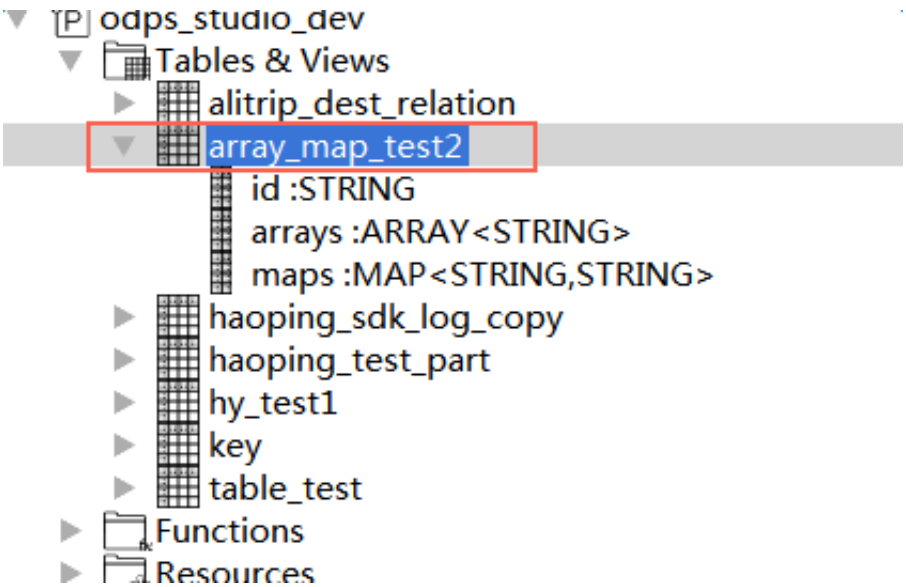
   b. 点击工具栏上的刷新图标或在右键菜单中选择刷新菜单项。



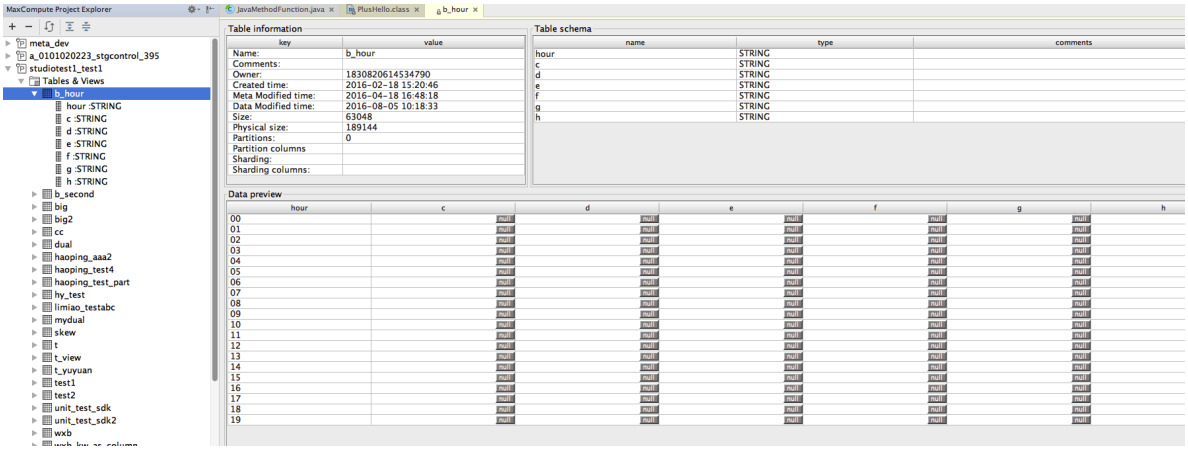
查看表详细信息

用户可以通过Studio的 表详情视图（Table Details View） 查看数据表详细信息。

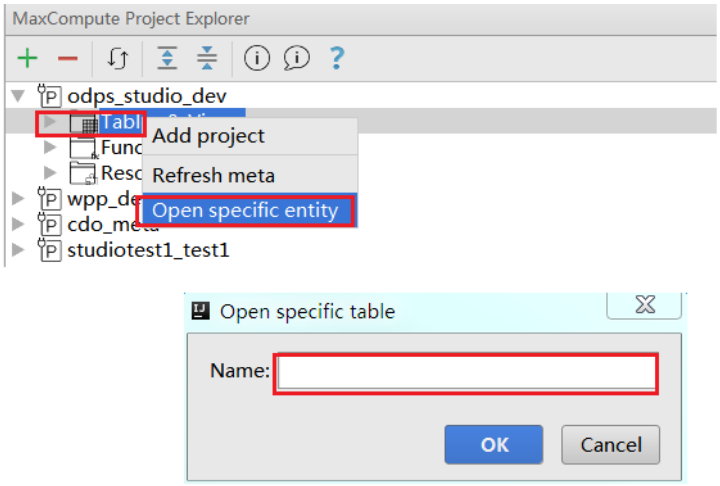
1. 在节点树中，展开某个表名节点，可快速查看列名和类型：



2. 双击某个表或右键菜单Show Table Detail可以查看表的详细信息，包括owner，size，column等元数据，表结构信息，以及data preview：

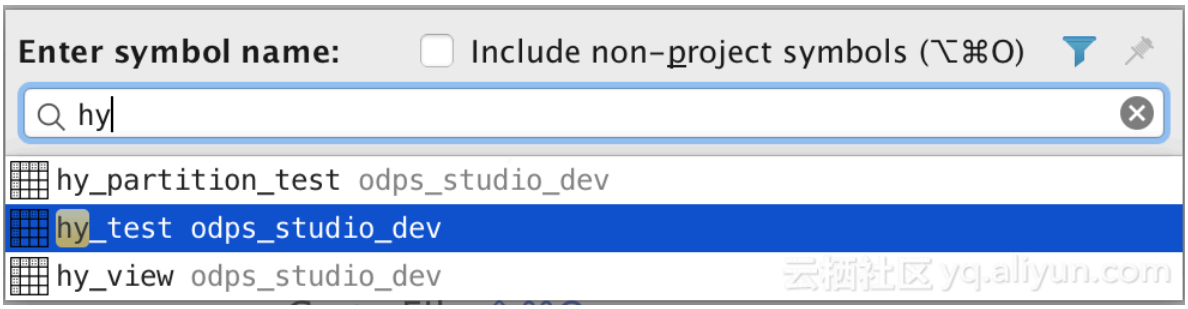


3. 通过在Tables&Views右键菜单项Open specific entity，可以指定表名显示详情（注意要完整表名称）。另外如果用户没有project的list权限，而只有具体某张表的权限，也可以通过这种方式将该表抓取下来。函数（Functions）及资源（Resources）类似。



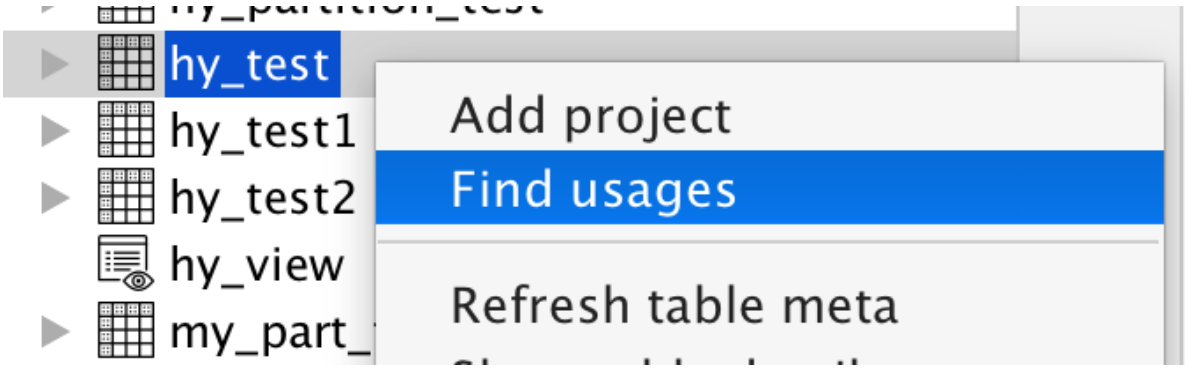
Intellij IDE默认支持搜索，可展开表后直接敲击键盘模糊搜索。

4. studio也支持快速搜索某张表，可通过快捷键(win:Ctrl+Alt+Shift+N mac:#+⌘+O)唤出navigate symbol，输入表名后回车即可。



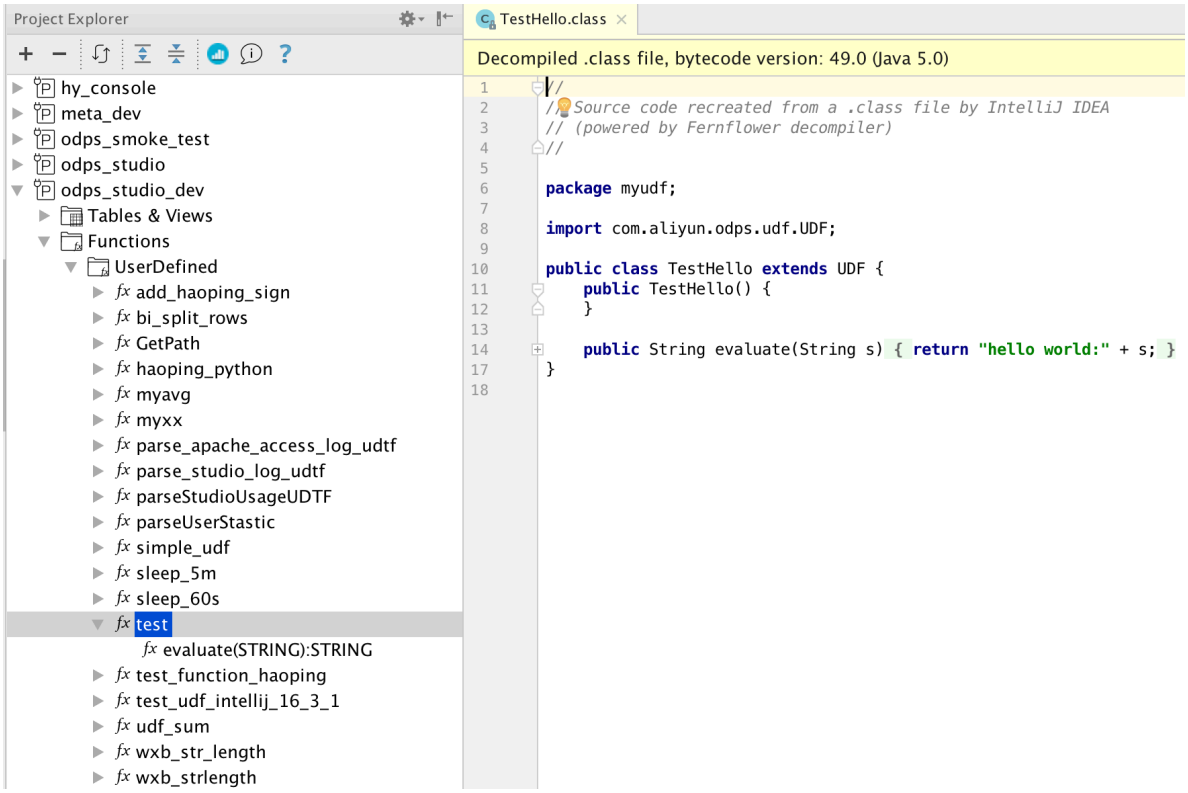
可通过前置关键字(table:或function:或resource:)缩小搜索范围，譬如想搜索函数count，可输入function:count。

5. 当想知道某张表在哪些script中用到时，可以右键该表，使用Find Usages功能。



查看函数详细信息

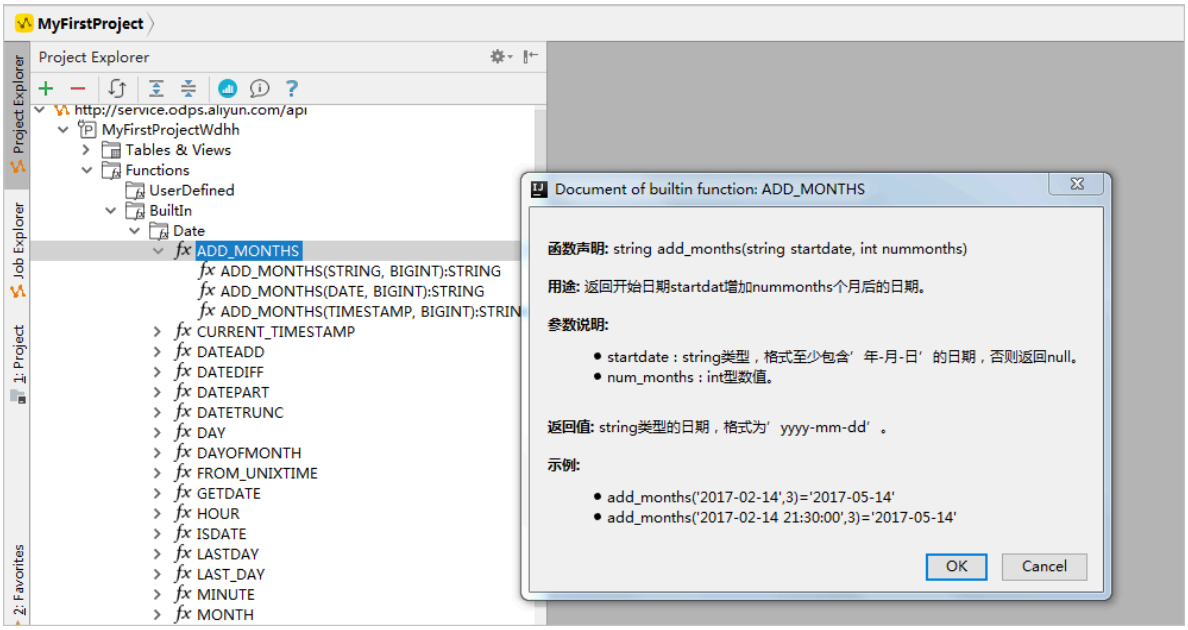
1. 在Functions树UserDefined节点下可以展开某个函数节点，以显示该函数的方法签名。双击某个函数节点（或在Resources下双击改函数对应的源码资源），可打开该函数对应的的代码。



说明:

Java代码通过反编译jar获取，并非源码。Python UDF解析签名需要安装pyodps(MaxCompute python sdk)，具体的先安装`pip: sudo /usr/bin/python get-pip.py`(请自行google下载`get-pip.py`)，然后安装`pyodps: sudo /usr/bin/python -m pip install pyodps`。

2. 在Functions树BuiltIn节点下分类显示了系统内置函数，展开显示签名，双击显示函数文档。



2.4.2 导入导出数据

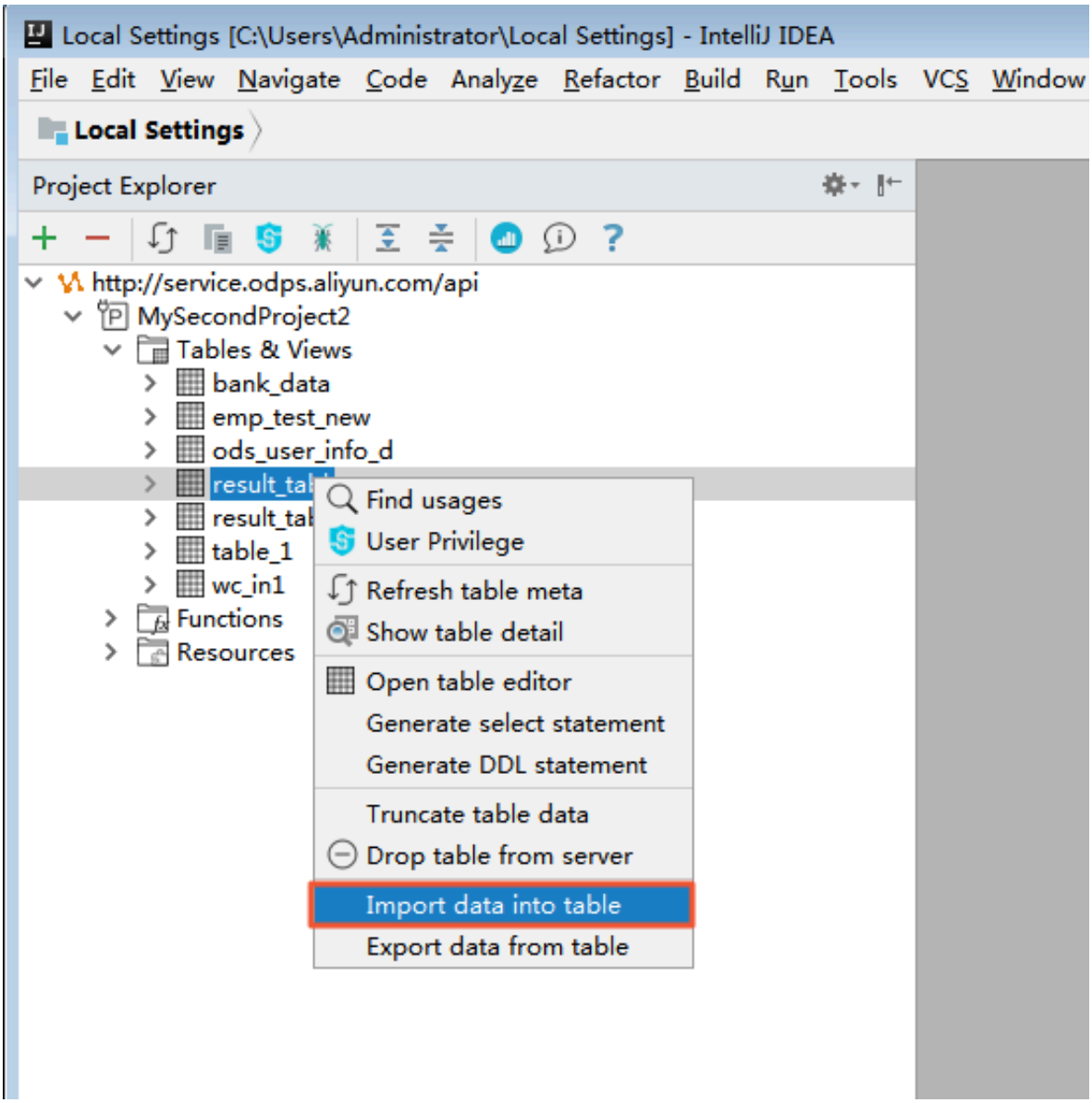
Studio可以将CSV，TSV等格式的本地数据文件导入到MaxCompute表中，也可将MaxCompute中表数据导出数据到本地文件。Studio是通过MaxCompute平台提供的批量数据通道（Tunnel）功能完成的。

使用须知

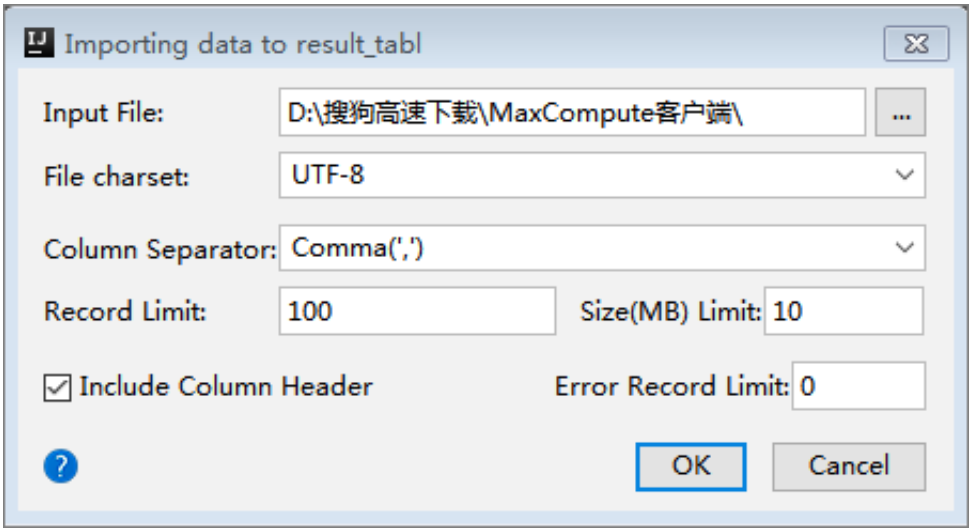
- 导入导出数据采用MaxCompute Tunnel服务，因此要求Studio中添加的MaxCompute Project必须开通或配置了Tunnel服务。
- 导入导出表必须具备相应权限。

导入数据

1. 打开项目空间浏览器（Project Explorer)窗口，在表名上点击右键或在表详细页面的Data Preview中字段属性上右键，选择Import Data Into Table 。



2. 在弹出的Import Data对话框中，选择导入数据文件的路径，列分隔符(可自定义输入)，大小限制，错误容忍行数等参数，点击按钮OK 。

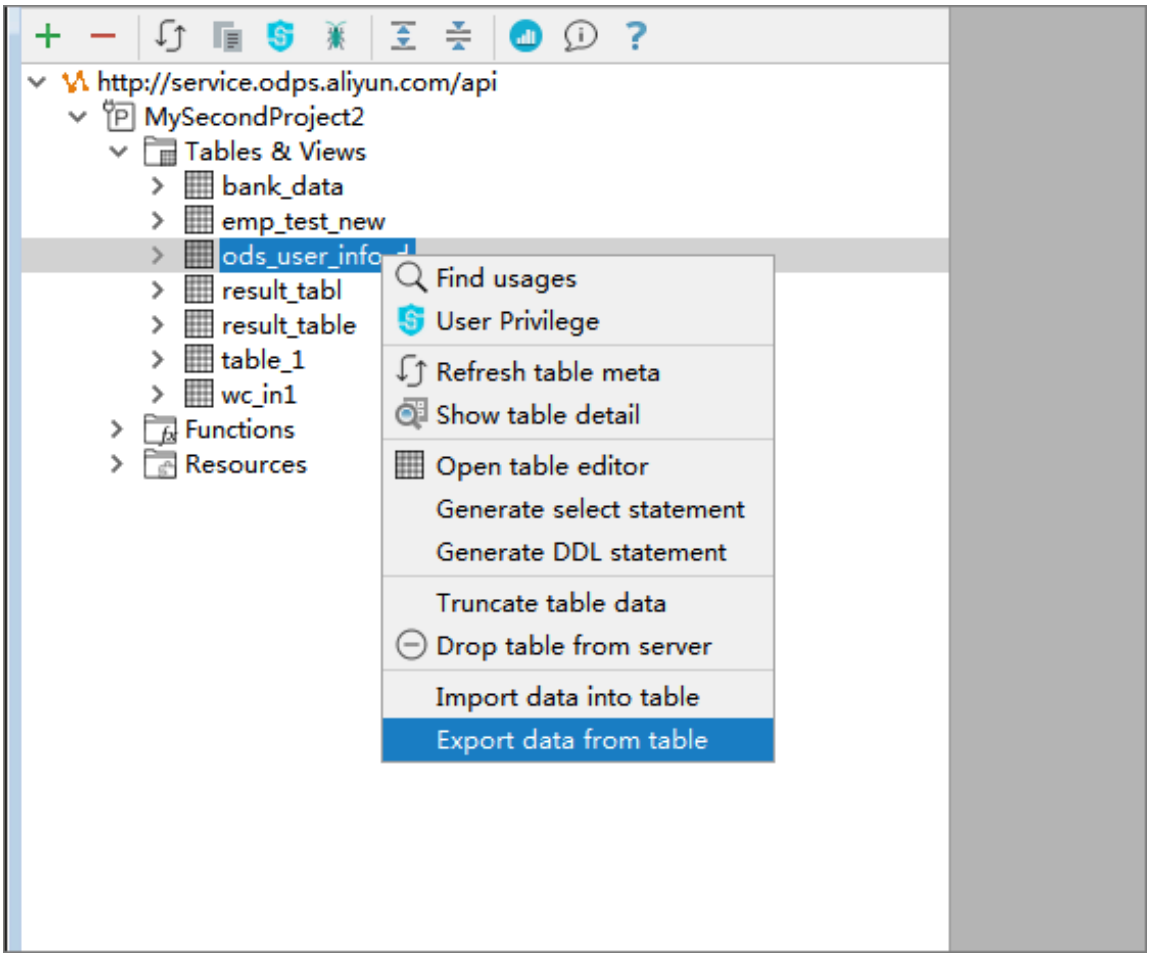


3. 提示Import Data Success，表示数据导入成功，可在表中查看导入的数据。

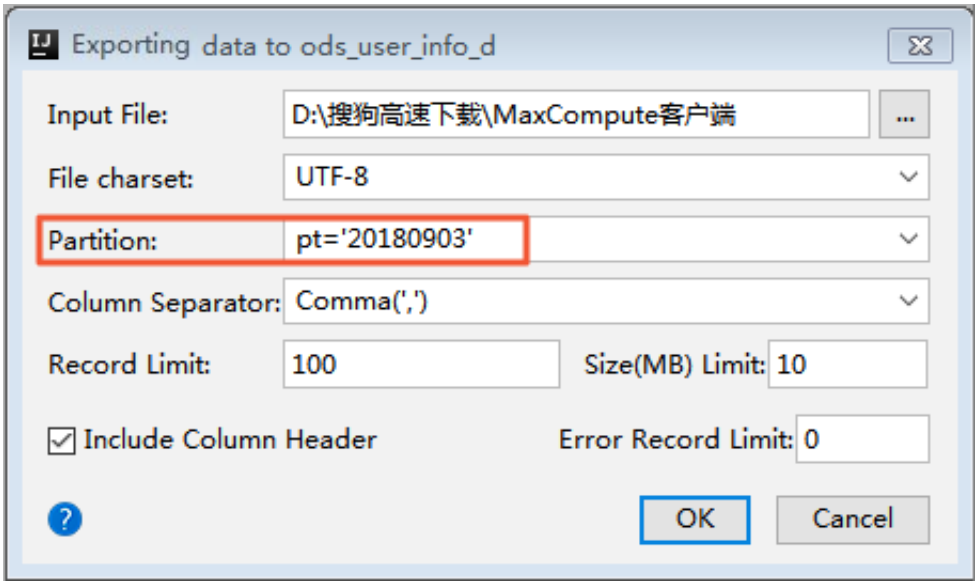
导出数据

1. 启动导出表数据有两种方式：

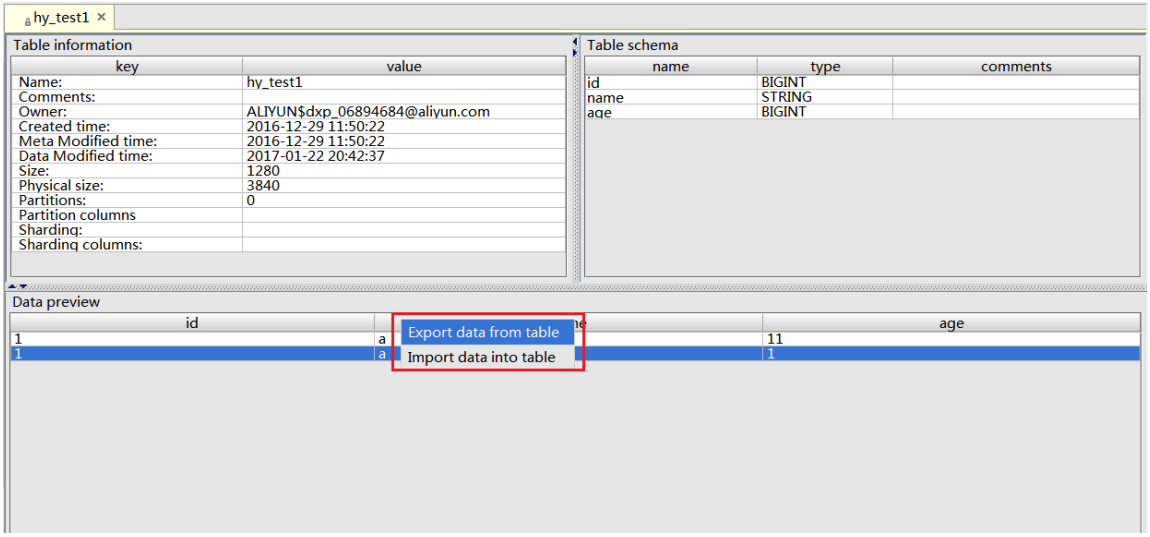
   · 在表名上右键，选择Export Data From Table。



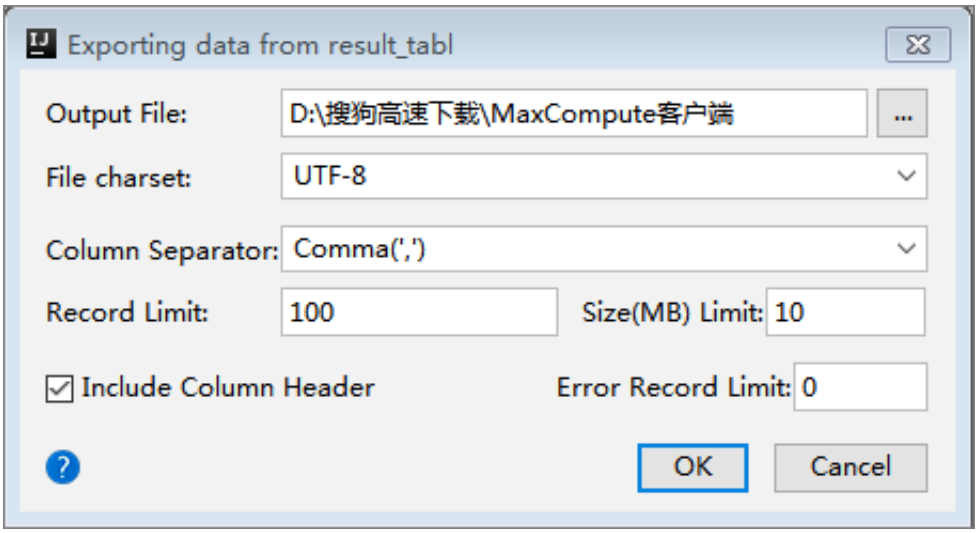
如果您导出的是分区表，在导出时可输入分区值。



- 在表详细页面的Data Preview中字段属性上右键，选择Export Data From Table。

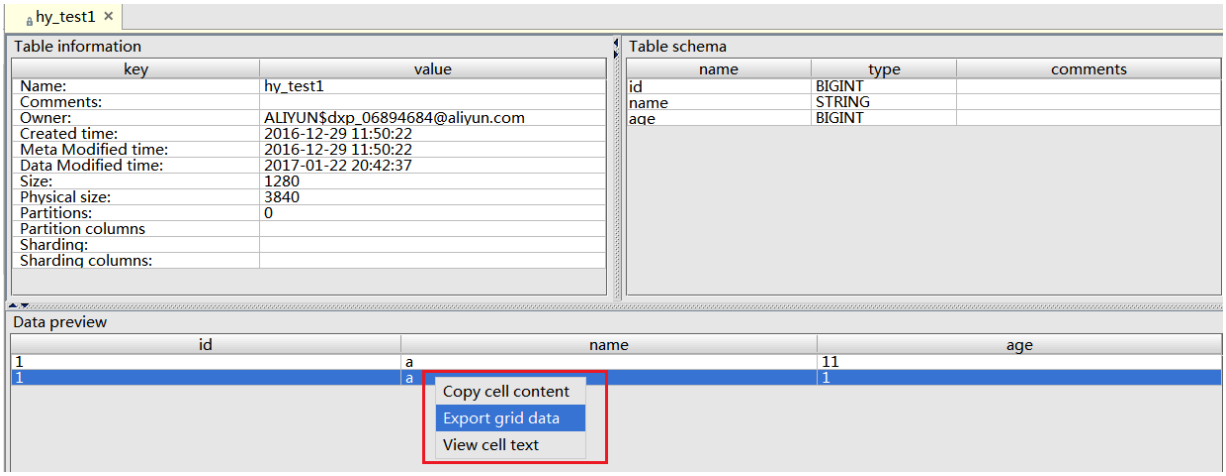


2. 弹出Export Data对话框，选择导出数据文件的保存路径，列分隔符(可自定义输入)，大小限制，错误容忍行数等，填写完成后点击OK。



3. 提示Export Data Success，表示数据导出成功，可在目标文件中看到导出的数据。

用户也可以在Table的Data Preview窗格中选择右键菜单Export Grid Data导出数据。



说明:

在Data Preview窗格的导出数据功能仅导出显示在数据示例（Data Sample），而不一定是表中的全部数据。

新类型导入导出

只需按照约定格式生成文本并存储为csv或tsv格式，就可通过studio导入到table。

下面将详细介绍各个数据类型的转换规则。

- 基本类型

1. TINYINT, SMALLINT, INT, BIGINT直接存储为整型字符串，数值超过类型边界会报错。
2. FLOAT, DOUBLE存储小数字符串或浮点形式，如：2.342 1E+7。
3. VARCHAR直接存储为字符串，超过上限会自动截断，不会报错。
4. STRING直接存储为字符串。
5. DECIMAL支持整形或浮点型的字符串。
6. BINARY需要将二进制数据编码为base64 string。
7. DATETIMEdate time需import dialog中指定的format格式保持一致，格式不匹配将报错。
8. TIMESTAMPtimestamp需要按照`yyyy-[m]m-[d]d hh\:mm\:ss[.f...]`格式存储为字符串。
9. BOOLEANtrue or false字符串。

- 复合类型

1. ARRAY需存储为JSON数组，数组元素按照本文约定规则转换成字符串，数组元素支持任意类型。
2. MAP需存储为JSON对象，map key、value按照本文约定规则转换为字符串，value支持任意类型嵌套。
3. STRUCT需存储为JSON对象， struct字段名为string，转换为JSON对象的key， struct字段值转换为JSON对象的value，字段值以本文定义规则转换。

示例

- array类型

对于如下所示表结构：

列名	列数据类型
c_1	ARRAY<TINYINT>
c_2	ARRAY<INT>
c_3	ARRAY<FLOAT>
c_4	ARRAY<DATETIME>
c_6	ARRAY<TIMESTAMP>
c_7	ARRAY<STRING>

可通过下面所示csv格式导入数据：

```
c_1,c_2,c_3,c_4,c_6,c_7
"[""1"",""2"",""3""]","[""1"",""2"",""3"",""4""]","[""1.2"",""2.0
""]","[""2017-11-11 00:00:00"",""2017-11-11 00:00:00"",""2017-11-11
 00:00:00""]","[""2017-11-11 00:00:00.123456789"",""2017-11-11 00:00
:00.123456789"",""2017-11-11 00:00:00.123456789""]","[""aaa"",""bbb
"",""ccc""]"
"[""1"",""2"",""3""]","[""1"",""2"",""3"",""4""]","[""1.2"",""2.0
""]","[""2017-11-11 00:00:00"",""2017-11-11 00:00:00"",""2017-11-11
 00:00:00""]","[""2017-11-11 00:00:00.123456789"",""2017-11-11 00:00
:00.123456789"",""2017-11-11 00:00:00.123456789""]","[""aaa"",""bbb
"",""ccc""]"
"[""1"",""2"",""3""]","[""1"",""2"",""3"",""4""]","[""1.2"",""2.0
""]","[""2017-11-11 00:00:00"",""2017-11-11 00:00:00"",""2017-11-11
 00:00:00""]","[""2017-11-11 00:00:00.123456789"",""2017-11-11 00:00
:00.123456789"",""2017-11-11 00:00:00.123456789""]","[""aaa"",""bbb
"",""ccc""]"
```

说明：

CSV格式需要对双引号进行转义，通过两个双引号来表示双引号，具体可参考CSV格式规范。

- map类型

  对于如下所示表结构：

列名	列数据类型
c_1	MAP<TINYINT,STRING>
c_2	MAP<STRING,INT>
c_3	MAP<FLOAT,STRING>
c_4	MAP<STRING,DATETIME>
c_5	MAP<STRING,STRING>
c_6	MAP<TIMESTAMP,STRING>

  可通过下面所示csv格式导入数据：

```
c_1,c_2,c_3,c_4,c_5,c_6
"{1:""2345""}","{""123"":""2"",""3"":""4""}","{2.0:""223445"",1.2
:""1111""}","{""aaa"":""2017-11-11 00:00:00"",""ccc"":""2017-11-11
 00:00:00"",""bbb"":""2017-11-11 00:00:00""}","{""ckey"":""cvalue
""}","{""2017-11-11 01:00:00.123456789"":""dddd"",""2017-11-11 00:00
:00.123456789"":""aaa"",""2017-11-11 00:01:00.123456789"":""ddd""}"
"{1:""2345""}","{""123"":""2"",""3"":""4""}","{2.0:""223445"",1.2
:""1111""}","{""aaa"":""2017-11-11 00:00:00"",""ccc"":""2017-11-11
 00:00:00"",""bbb"":""2017-11-11 00:00:00""}","{""ckey"":""cvalue
""}","{""2017-11-11 01:00:00.123456789"":""dddd"",""2017-11-11 00:00
:00.123456789"":""aaa"",""2017-11-11 00:01:00.123456789"":""ddd""}"
"{1:""2345""}","{""123"":""2"",""3"":""4""}","{2.0:""223445"",1.2
:""1111""}","{""aaa"":""2017-11-11 00:00:00"",""ccc"":""2017-11-11
 00:00:00"",""bbb"":""2017-11-11 00:00:00""}","{""ckey"":""cvalue
""}","{""2017-11-11 01:00:00.123456789"":""dddd"",""2017-11-11 00:00
:00.123456789"":""aaa"",""2017-11-11 00:01:00.123456789"":""ddd""}"
```

- struct类型

  对于如下所示表结构：

列名	列数据类型
c_struct	<RUCT<x:INT,y:VARCHAR(256),z: STRUCT<a:TINYINT,b:STRING>>

  可通过下面所示csv格式导入数据：

```
c_struct
"{""x"":""1000"",""y"":""varchar_test"",""z"":{""a"":""123"",""b
"":""stringdemo""}}"
```

```
"{""x"":""1000"",""y"":""varchar_test"",""z"":{""a"":""123"",""b
"":""stringdemo""}}"
```

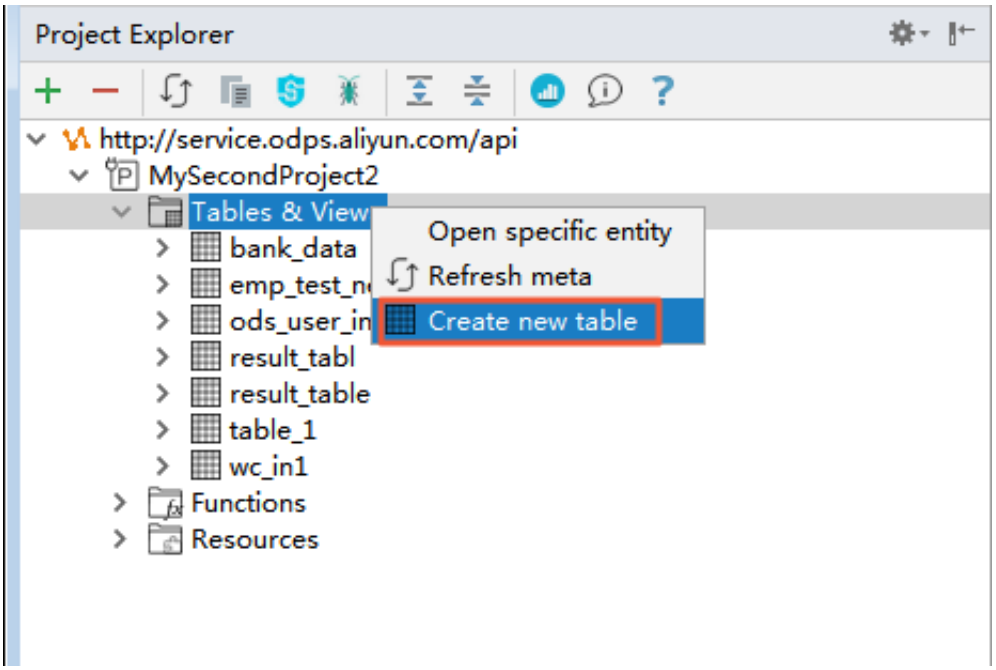
2.4.3 可视化创建、修改和删除表

Studio的project explorer提供了可视化表结构编辑器，可进行创建、修改表。

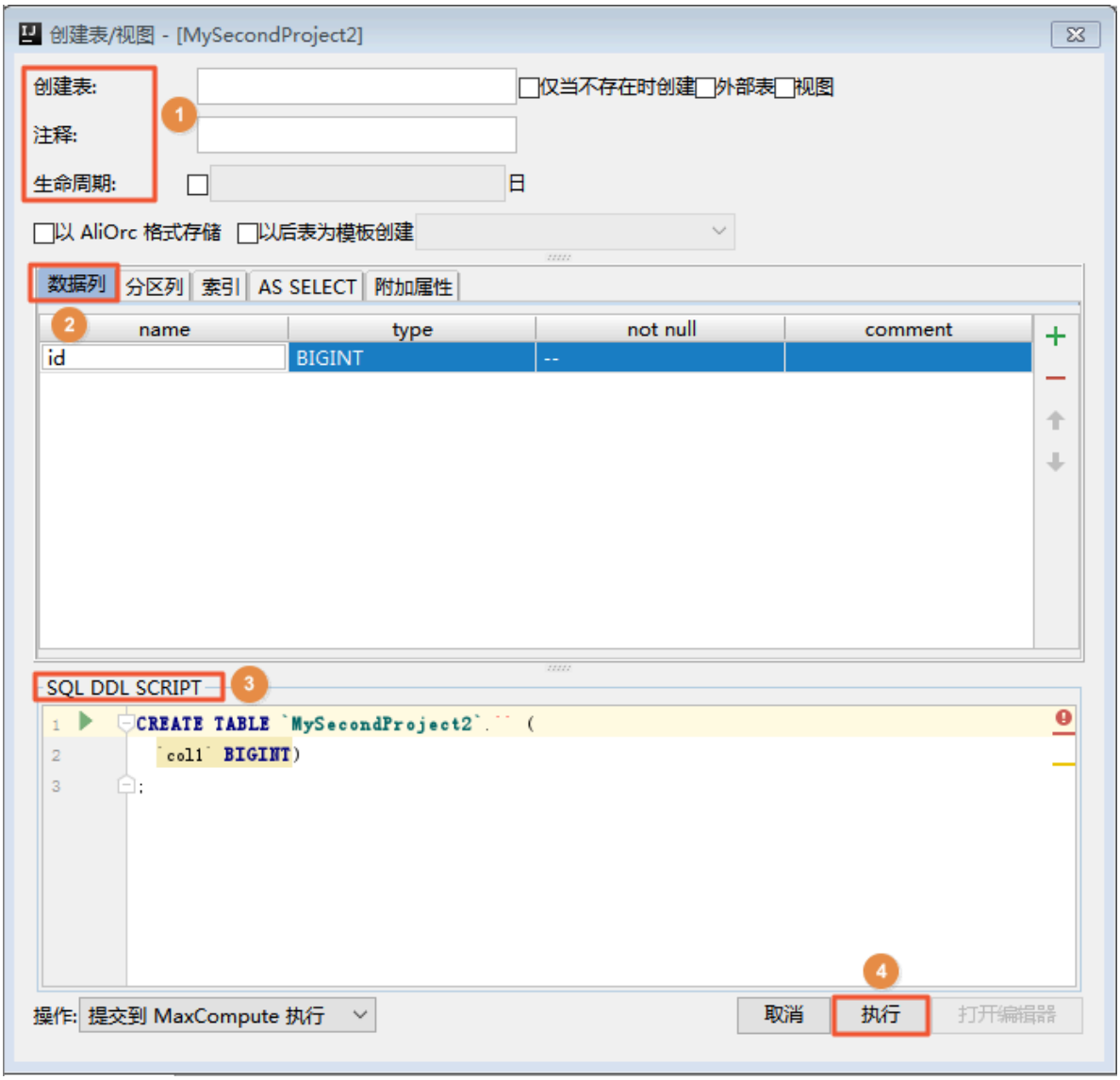
可视化建表

操作步骤

1. 右击需要创建表的project，选择Create a new table:



2. 弹框里填写对应的表名、列等信息，然后点击Generate CreateTable Statement生成对应的DDL语句，点击Execute,执行建表。



表/列名、生命周期、列类型等限制请遵循MaxCompute相关要求，可参见表操作。

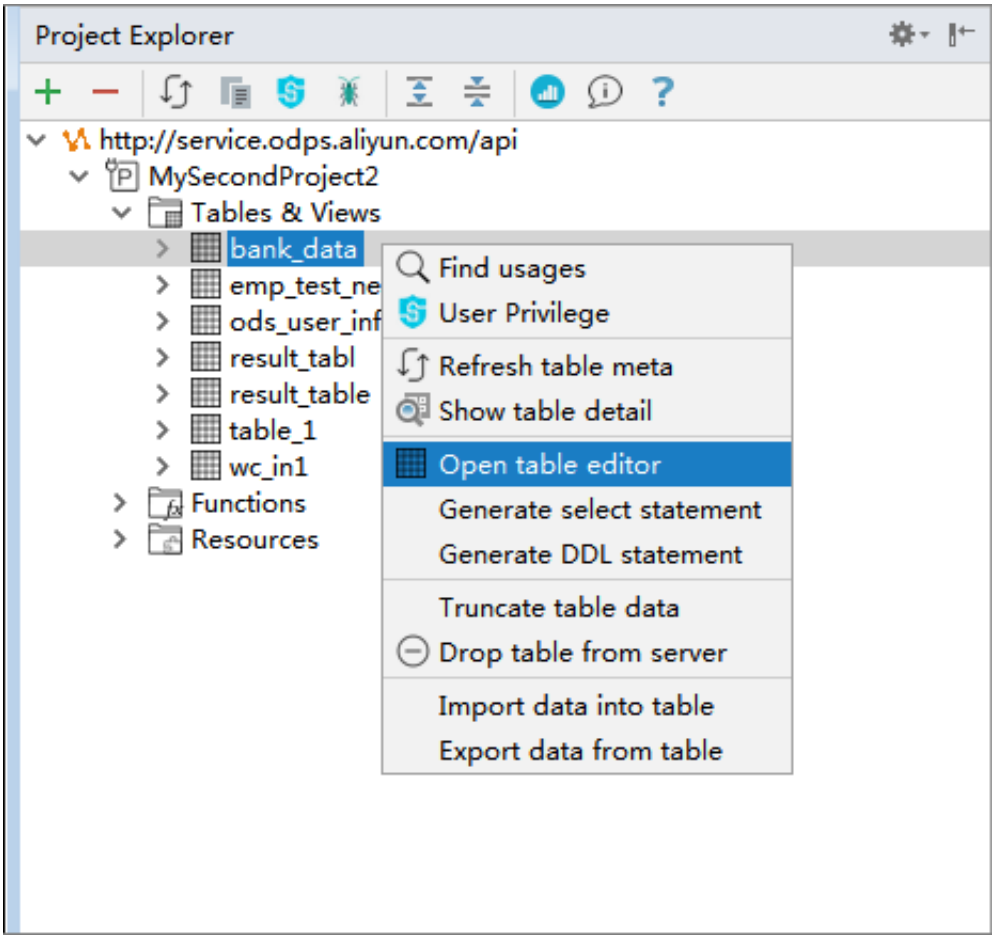
> **说明:**
>
> studio可视化建表并没有选择flag的设置，默认使用以下两个flag:
>
> a. `odps.sql.submit.mode : script`
>
> b. `odps.sql.type.system.odps2: true`

3. 表创建成功后，可以在Project Explorer的table&view查看到元数据，若找不到请Refresh meta。

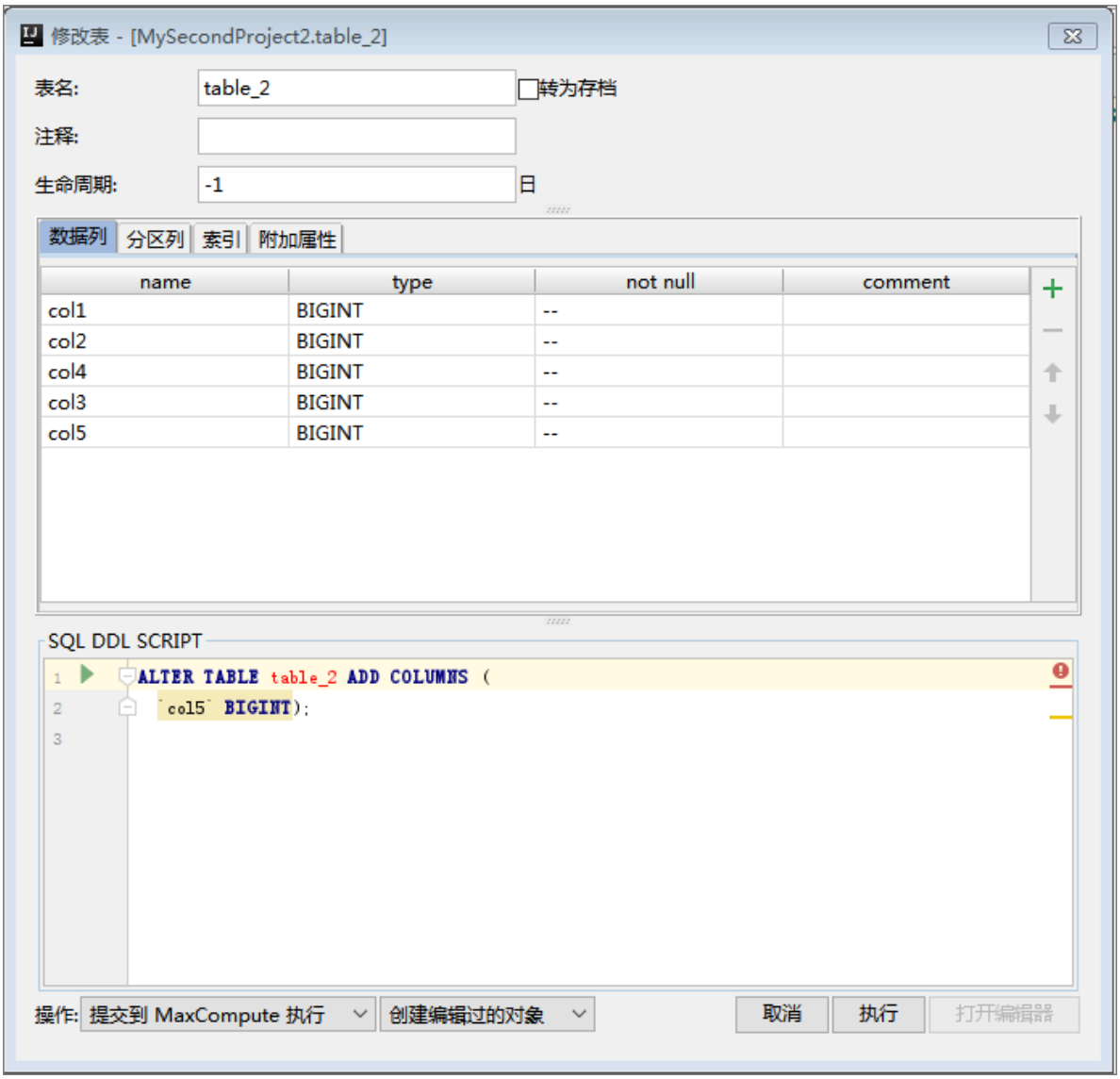
可视化修改表

操作步骤

1. 在Project Explorer的table&view列表中，右击需要修改的表，选择Open table editor:



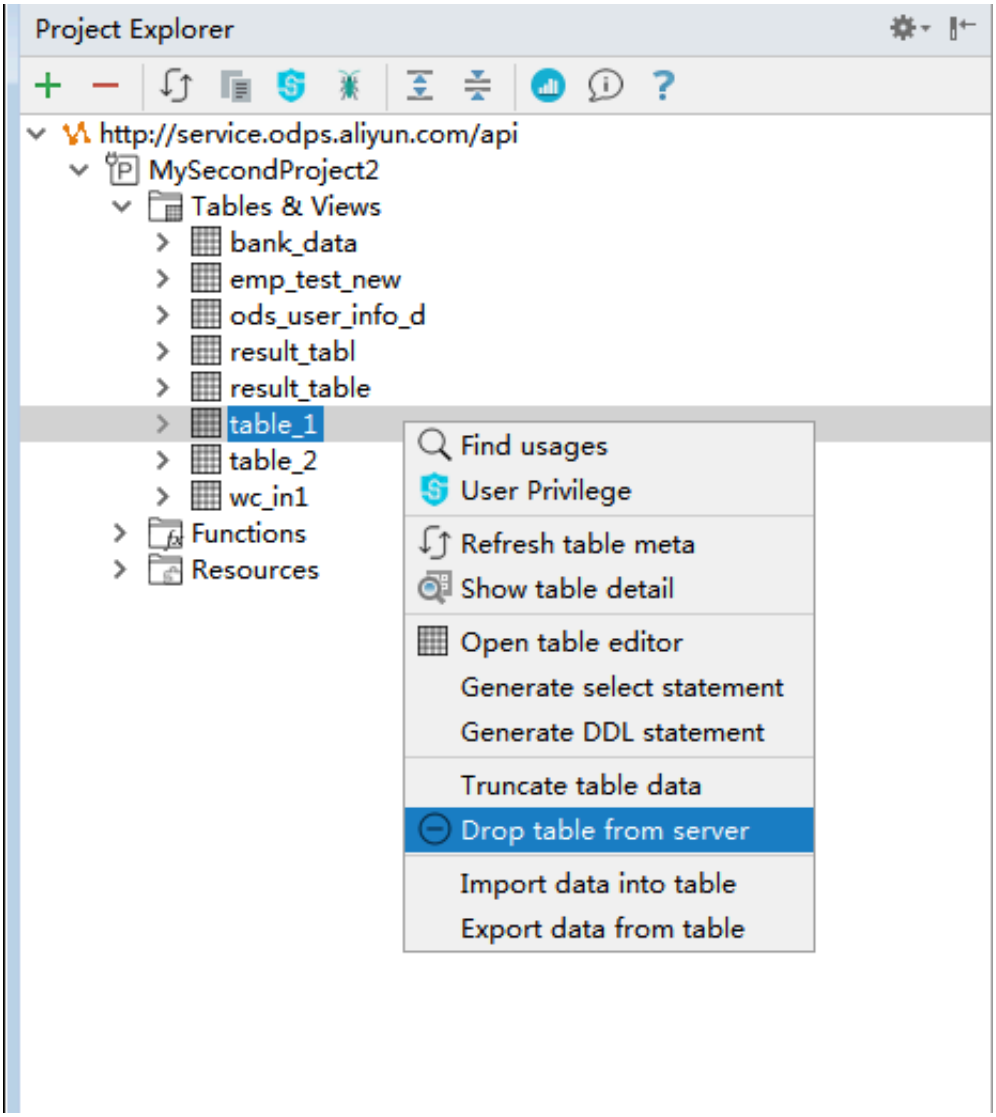
2. 弹框中对表进行编辑，可以修改表comments、lifecycle，修改列名、列描述,新增列。具体规则遵循MaxCompute table相关要求，可参看表操作。



3. 填写完成修改项后，点击Alter Table Statement 生成具体的Alter语句，点击Execute执行表修改操作。执行成功可以对表的元数据进行查看。

可视化删除表

在Project Explorer的table&view列表中，右击需要修改的表,选择Drop table from server:



弹框中选择OK即可将表从MaxCompute服务上删除。

2.5 开发SQL程序

2.5.1 创建MaxCompute Script Module

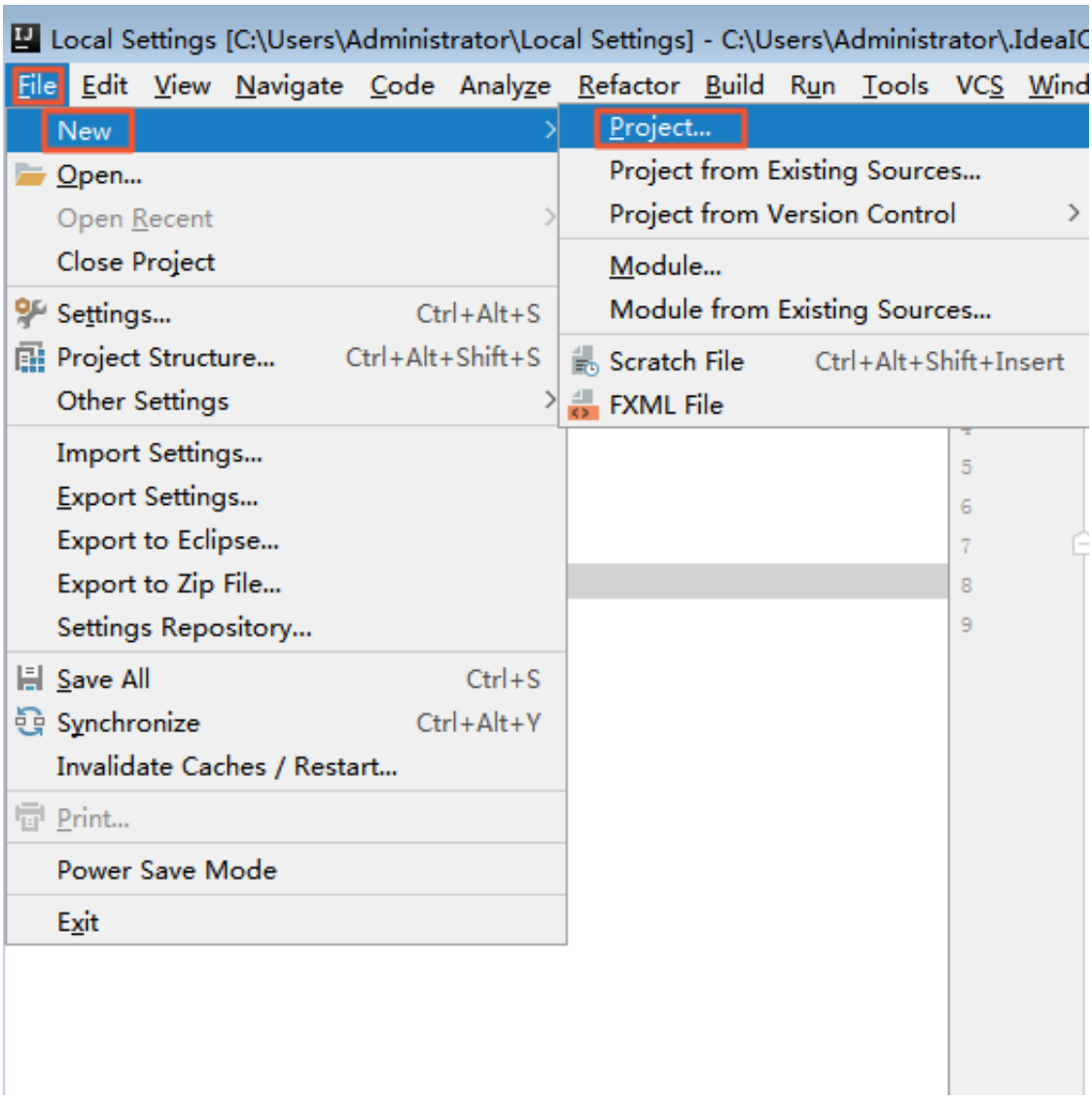
开发MaxCompute Script前，需要创建一个MaxCompute Script Module，创建时存在以下两种情况：

本地没有script文件

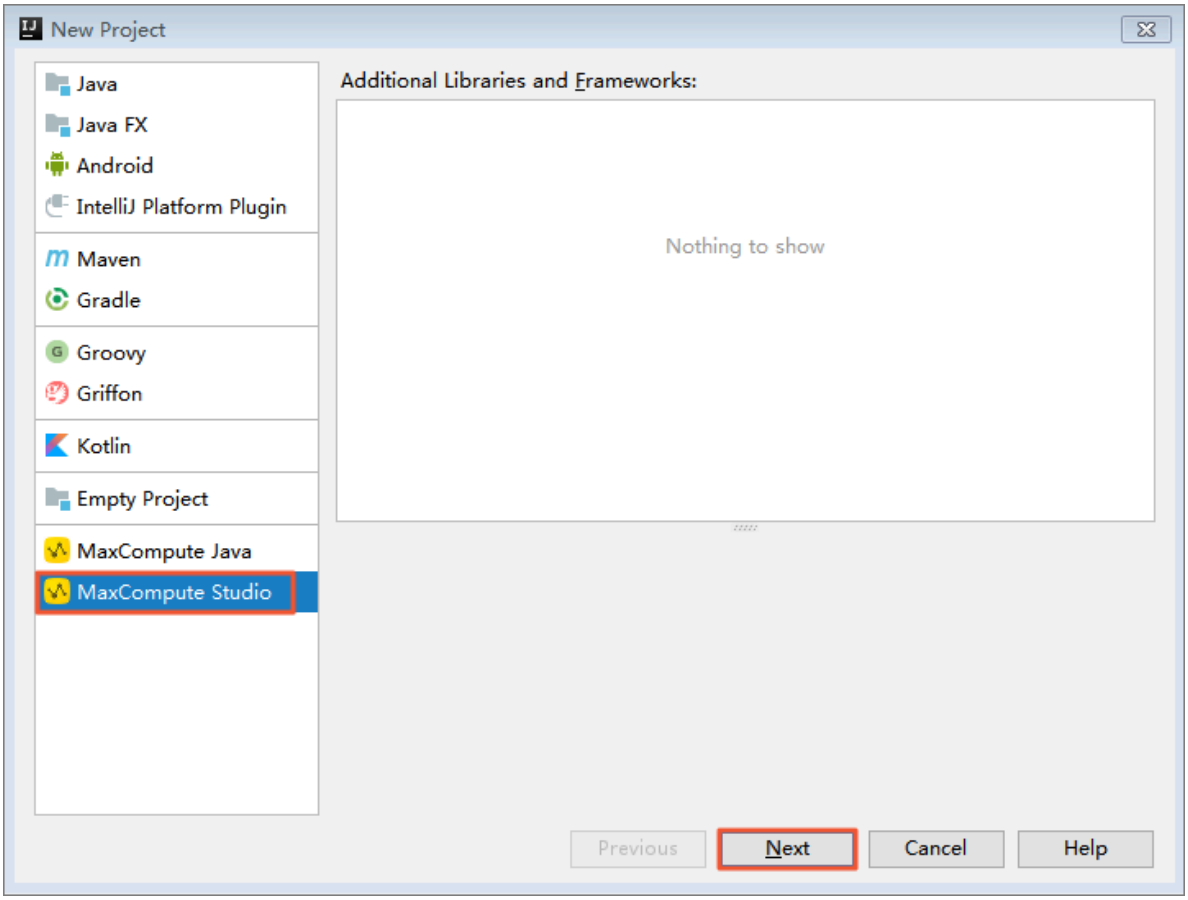
本地没有script文件时，可通过Intellij创建一个全新的Module。

操作步骤

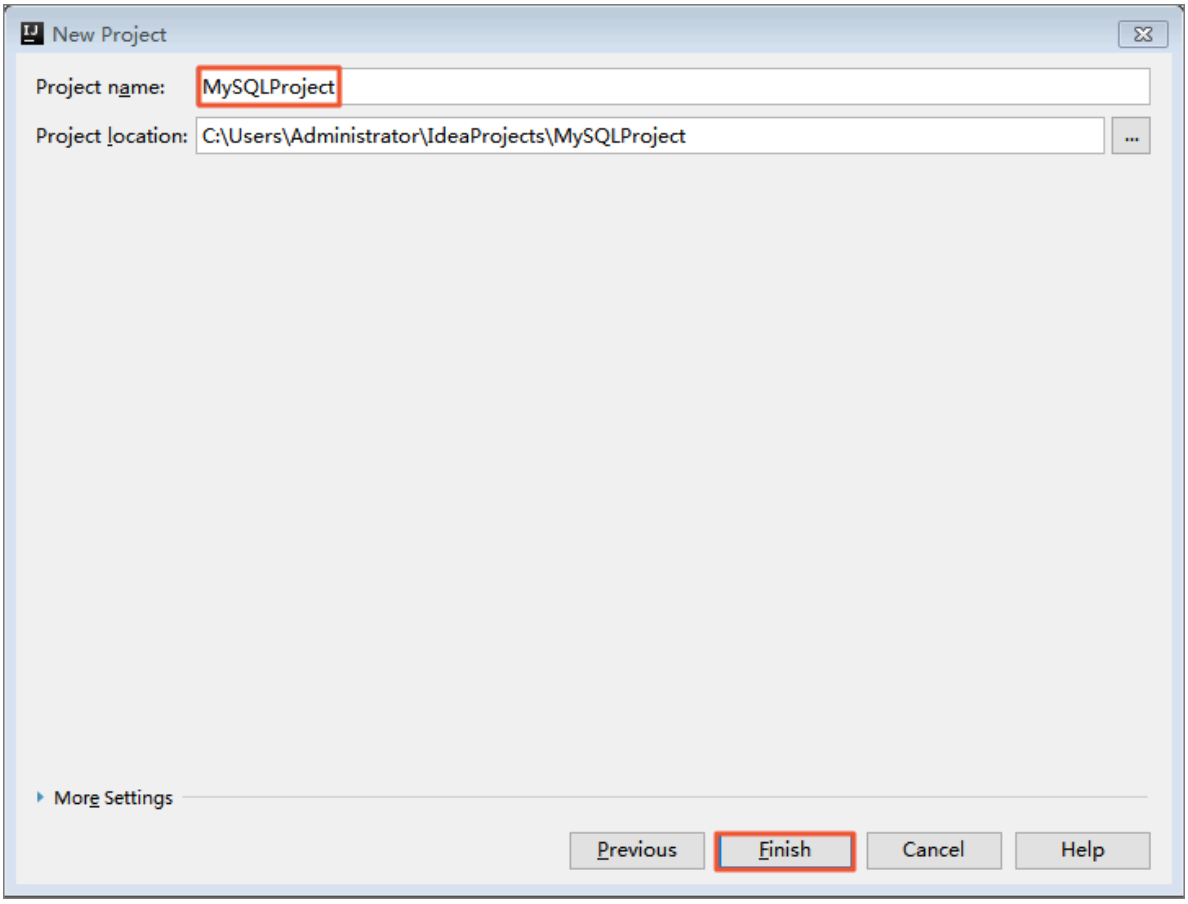
1. 打开或者新建一MaxCompute Studio Project。本文以新建为例，单击菜单栏中的File，导航至New > Project 。如下图所示：



2. 选择左侧导航栏中的MaxCompute Studio，单击Next 。

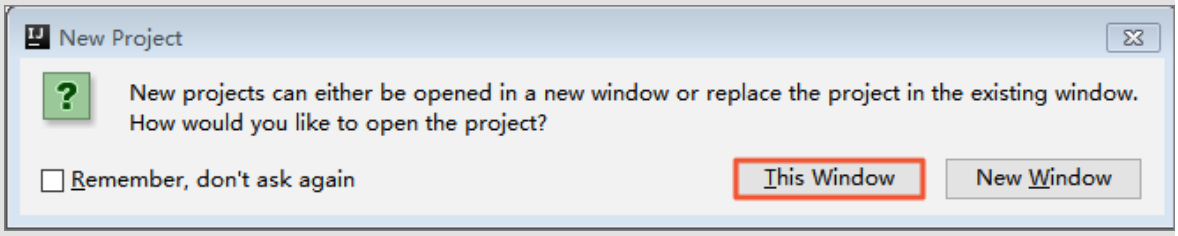


3. 填写Project Name，单击Finish。

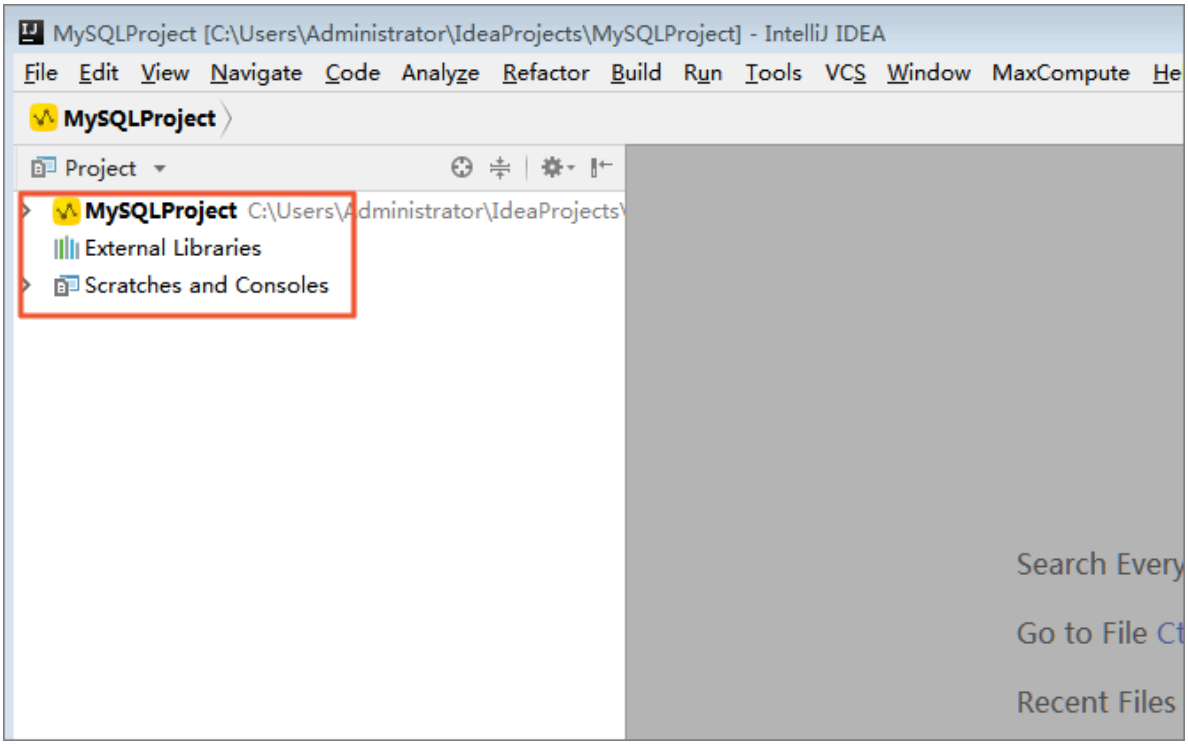


说明:

如果之前有打开的Project，将会提示您是否在当前窗口中打开（即关掉之前的Project），选择This Window。



4. 创建完成后，页面如下图所示，您即可在此项目中开发SQL脚本。



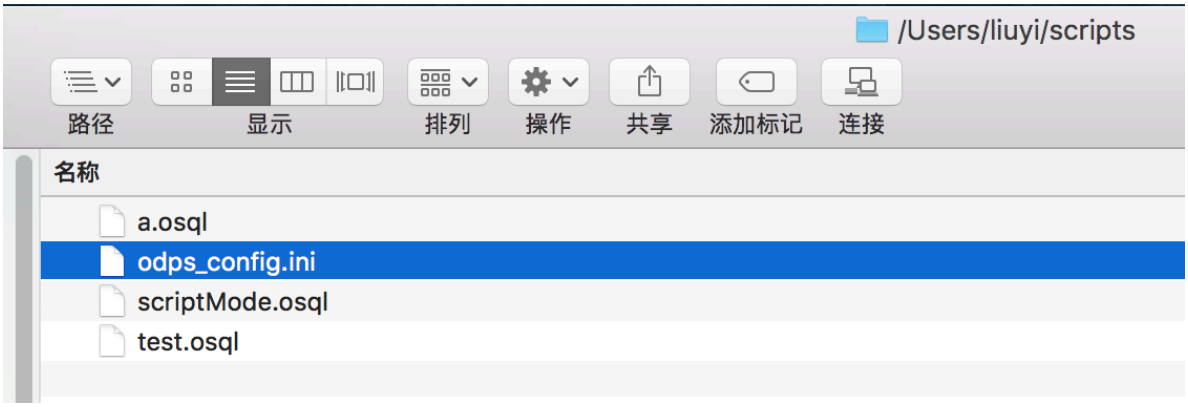
本地已有script文件

假如本地某个文件夹下已经有很多script，此时需要用MaxCompute studio来编辑这些 script，您可直接打开一个Module。

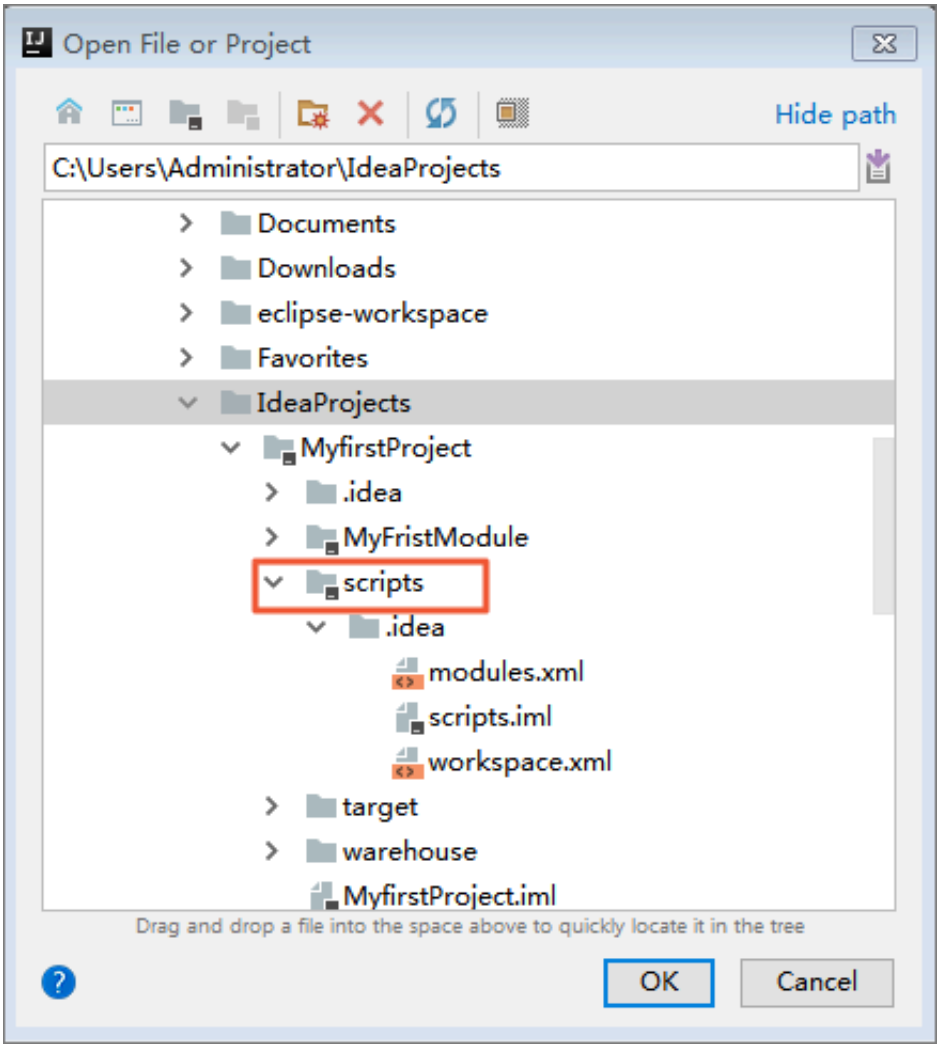
操作步骤

1. 在scripts文件夹下创建一个MaxCompute的连接配置文件 *odps_config.ini*，在里面配置与MaxCompute连接的鉴权信息：

   - project_name=xxxxxxxx
   - access_id=xxxxxxxxxx
   - access_key=xxxxxxxxx
   - end_point=xxxxxxxxx



2. 打开IDEA，导航至File > open， 选择您的scripts文件夹。



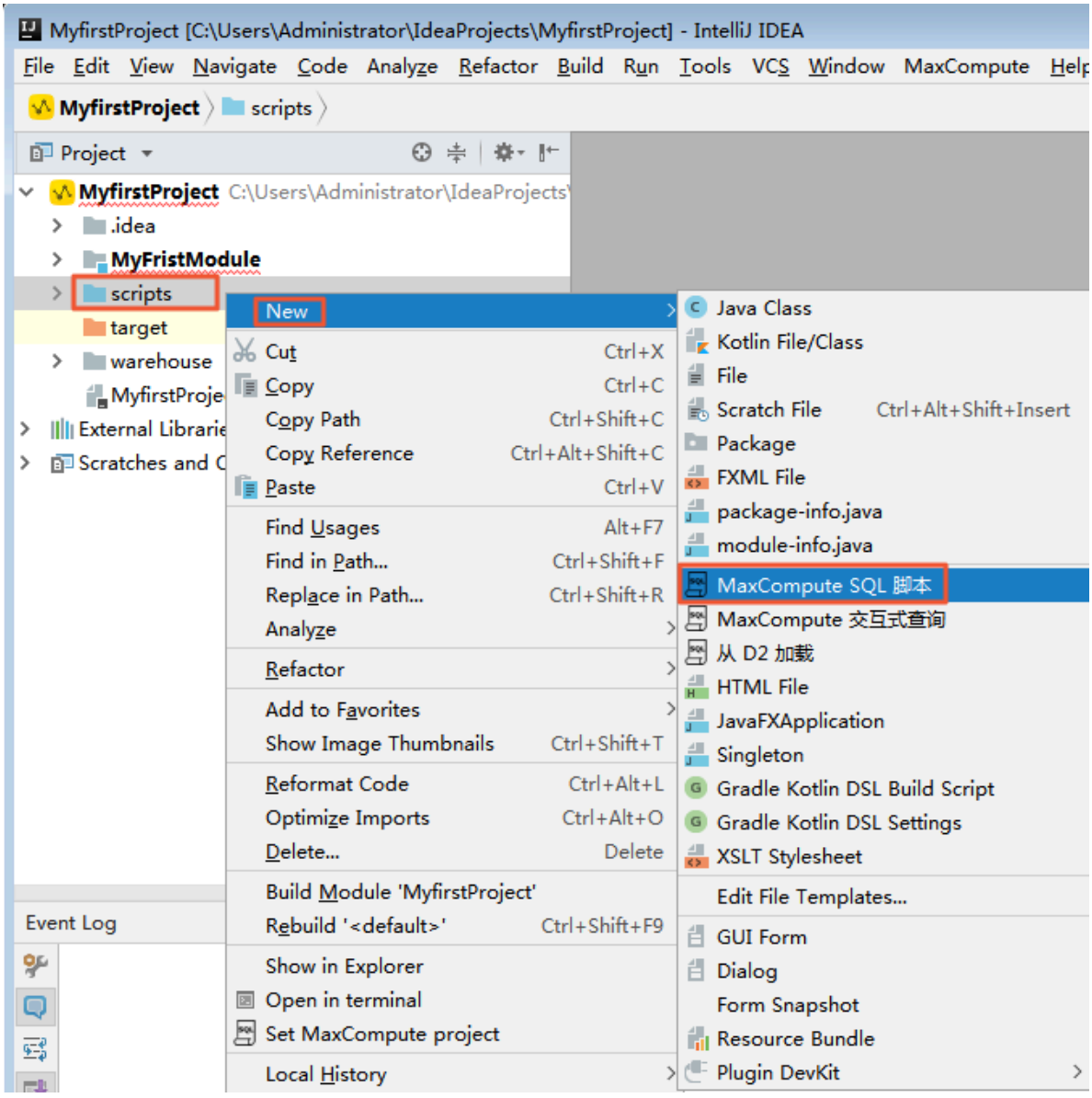
3. MaxCompute Studio会探测该文件夹下是否存在*odps_config.ini*文件，根据这个文件里的配置信息抓取服务端的meta，然后编译文件夹下的所有script。

2.5.2 编写SQL脚本

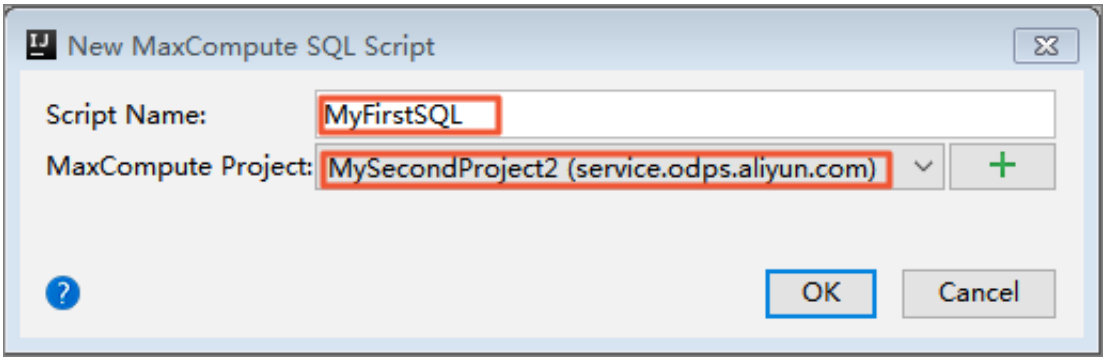
MaxCompute Studio模块创建完成后，即可开始编写MaxCompute SQL脚本。

操作步骤

1. 右击scripts，导航至New > MaxCompute Script。

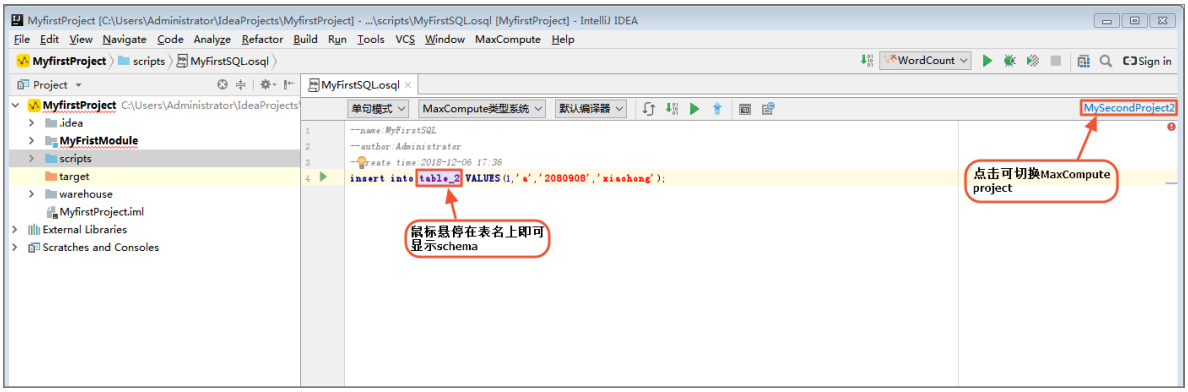


2. 填写弹出框中的相关内容，点击OK。



- Script Name：脚本名称。
- Script type：脚本类型。
- Target Project：目标MaxCompute项目。单击后面的+即可新建一个MaxCompute Project，配置详情请参见 新建项目空间连接。

3. 在SQL文件编辑界面中，编写SQL。

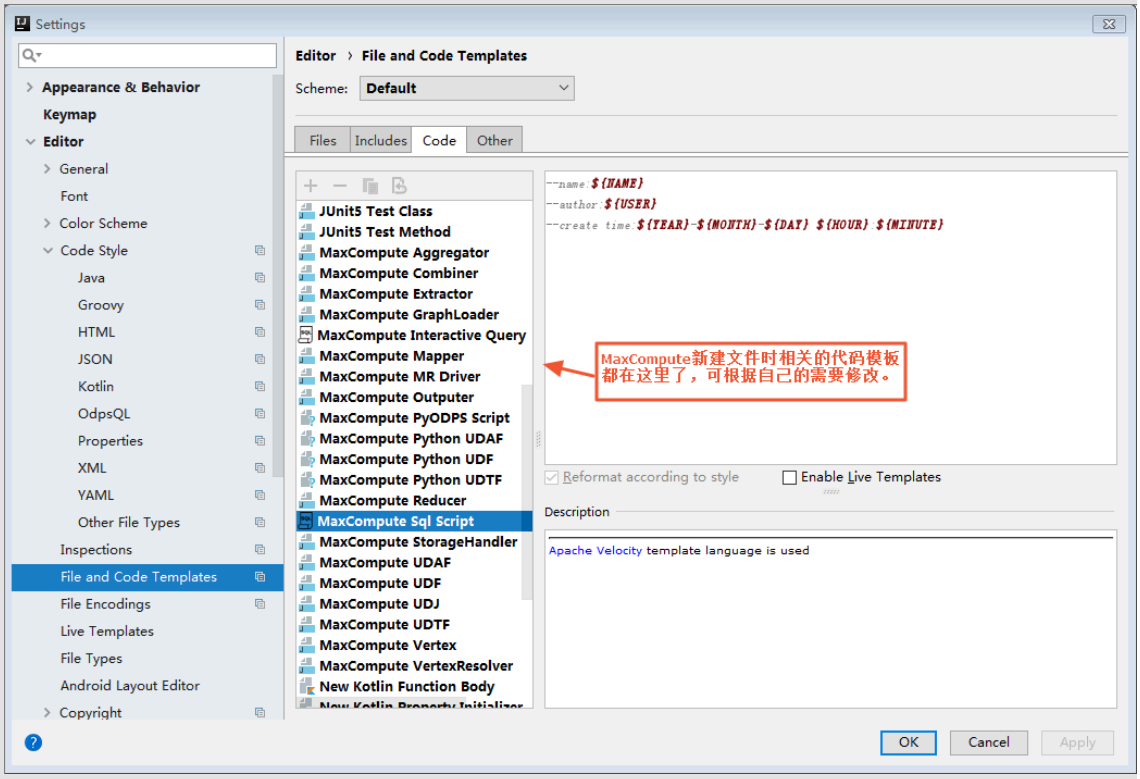


说明:

a. 实际SQL请根据自己的MaxCompute Project中的表进行编写。可单击toolbar右上角切换绑定的不同的MaxCompute项目，也支持跨project资源依赖。例如script绑定了ProjectA，同时还会用到ProjectB.table1，这时Studio会自动使用ProjectA的账号去抓取ProjectB的元数据。表的元数据Studio会保存在本机中类似下图的位置：

```
liuyi-MBP:hy_test liuyi$ pwd
/Users/liuyi/.odps.studio/meta/odps_studio_dev/tables/hy_test
liuyi-MBP:hy_test liuyi$ cat schema.ddl
CREATE TABLE IF NOT EXISTS `odps_studio_dev`.`hy_test` (
  `id` BIGINT,
  `name` STRING);
```

b. 新建sql script时的代码模板是可以在Intellij Preferences页修改的：

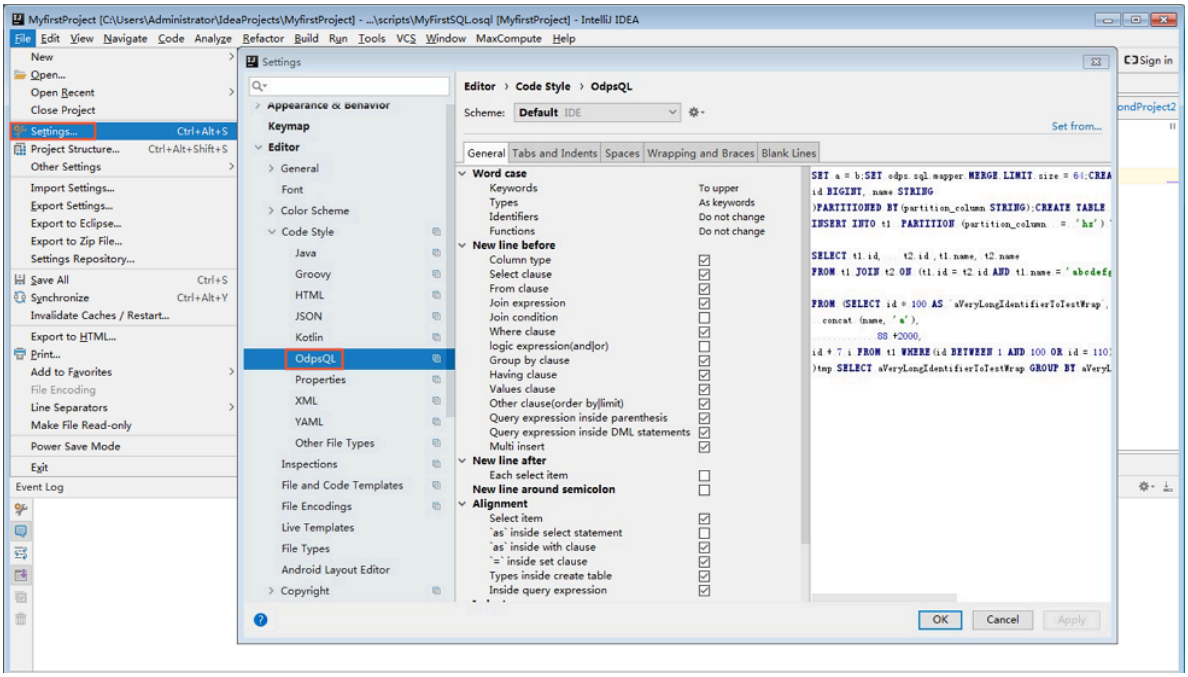


MaxCompute Studio功能

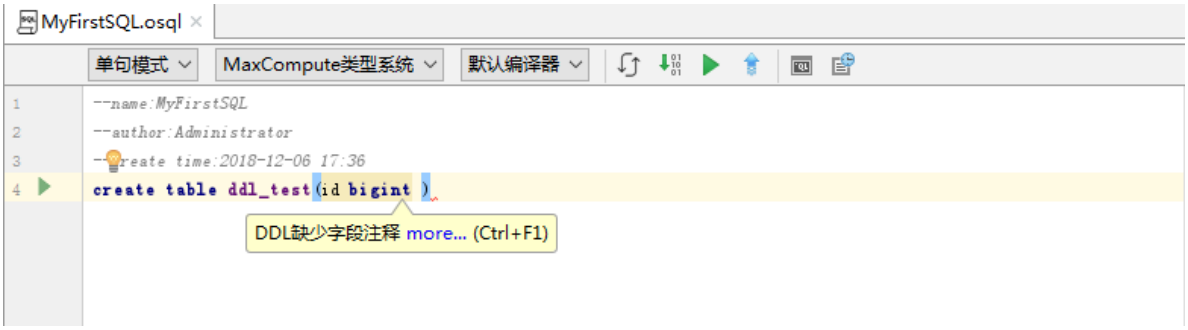
MaxCompute Studio不仅提供语法高亮，智能提醒，错误提示，还支持以下功能：

- schema annotator：当鼠标悬停在表上，可显示其schema。悬停在列上，可显示其类型；悬停在函数上，可显示其签名。
- code folding：可以将子查询等折叠起来，方便长SQL的阅读。
- brace matching：鼠标单击高亮左括号，其匹配的右括号也会高亮，反之亦然。
- go to declaration：按住Ctrl键，单击table，即可查看table详情。单击function，即可显示其源码。

- code formatting：支持对当前脚本格式化，快捷键（Ctrl + Alt + L）。可在如下页面自定义格式化规则，譬如关键字大小写，是否换行等。



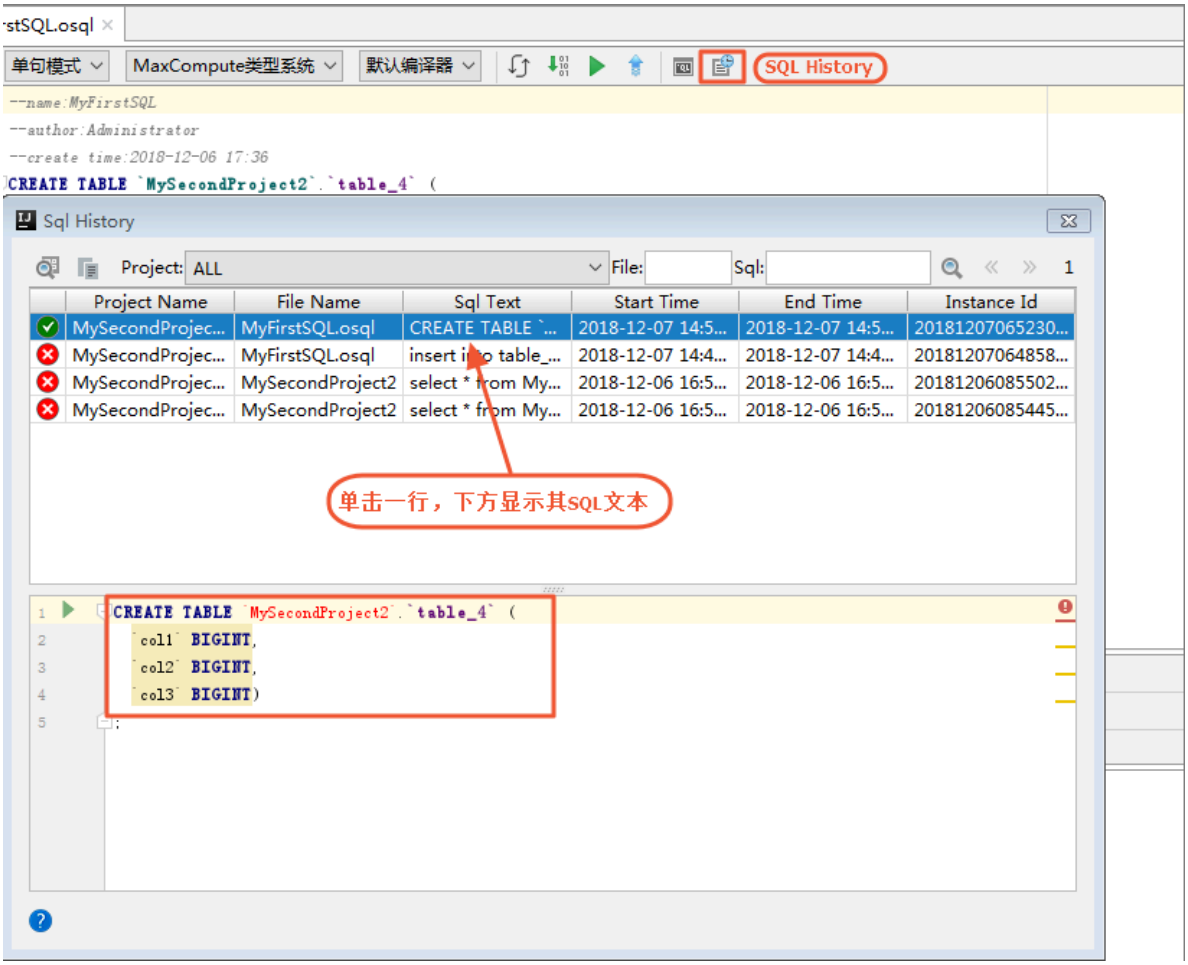
- code inspect：支持对当前脚本进行代码检查，某些检查还会提供quick fix，可通过快捷键Alt + Enter唤出。另外，可在Settings > Editor > Inspections > MaxCompute处修改某条规则。



- find usages：选中editor中的某张表（或函数），右键菜单选Find Usages，则会在当前IntelliJ project下寻找所有使用该表的脚本。
- live template：Studio内置了一些SQL live template，可以在编辑器中使用Ctrl + J (Command + J on Mac OS X) 快捷键唤出（例如忘记了insert into table的语法，便可唤出live template popup后搜索insert table）。

- builtin documentation：支持在系统内置函数处通过Ctrl + Q（Ctrl + J on Mac OS X）唤出帮助文档。

- sql history：所有通过studio提交运行的sql我们都记录在本地了，你可以点击toolbar上的图标，弹出sql history窗口，查询你曾经执行过的SQL。



详细的SQL编写命令介绍请参见表操作。

2.5.3 提交SQL脚本

MaxCompute Studio可直接将MaxCompute SQL提交到服务端运行，并显示查询结果，执行计划等详细信息。它在提交前会进行编译，能够有效避免提交到服务端后才发现编译错误。

前提条件

- 首先创建MaxCompute项目连接，并绑定目标项目。
- 创建 MaxCompute Studio Module。
- 在提交前需根据自身需求进行相关设置。MaxCompute Studio提供了丰富的设置功能，可在Editor编辑页面上方的Tool Bar工具栏中快速设置。设置主要分为以下三种：
  - 编辑器模式：编译器模式设定包括两种模式，脚本模式和单步模式。
    - 单步模式会将提交的脚本文件按;分隔，逐条提交到服务端执行。
    - 脚本模式为最新开发模式，可将整条脚本一次提交到服务端，由服务端提供整体优化，效率更高，推荐使用。
  - 类型系统：类型系统主要解决 SQL 语句的兼容性问题，分为以下三种类型：
    - 旧有类型系统：原有 MaxCompute 的类型系统。
    - MaxCompute 类型系统：MaxCompute 2.0 引入的新的类型系统。
    - Hive 类型系统：MaxCompute 2.0 引入的 Hive 兼容模式下的类型系统。
  - 编译器版本：MaxCompute Studio提供稳定版编译器和实验性编译器两种模式：
    - 默认编译器：稳定版本。
    - 实验性编译器：包含编译器最新特性。

    > 说明：
    > 脚本提交设置可采用全局设定，导航至File > Settings > MaxCompute，选择MaxCompute SQL，其中Compiler > Submit中可以设置以上属性。

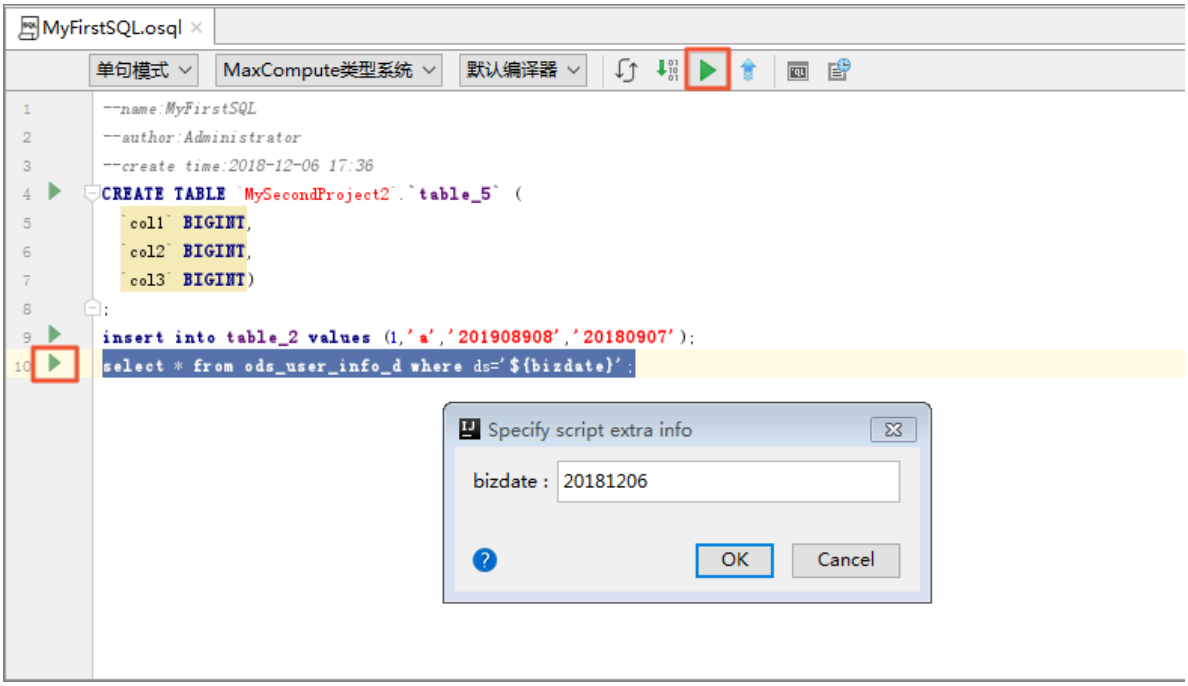
提交SQL脚本

Editor上方工具栏中提供同步，编译，提交功能。

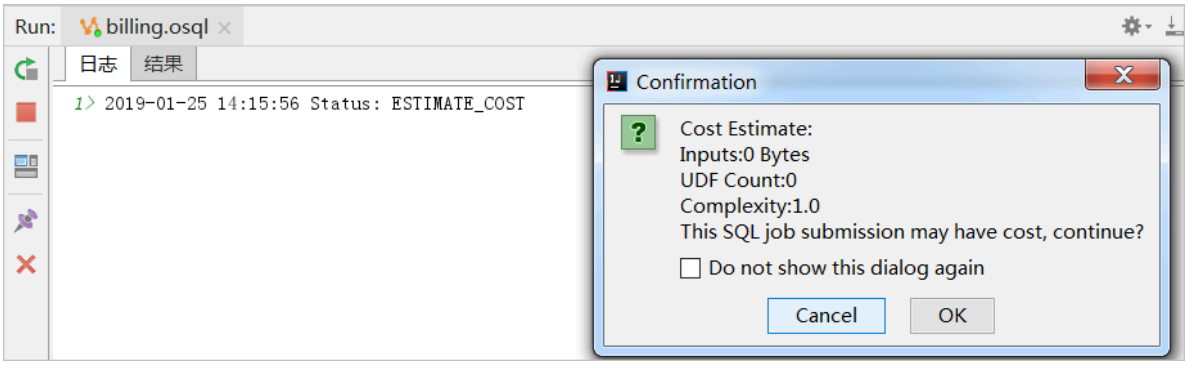
```
* **同步功能**：更新SQL脚本中使用的元数据，包括表名、UDF等。如果Studio提示表或函数找不到，而服务端又明确存在时，可尝试使用该功能更新元数据。
* **编译、提交**：按MaxCompute SQL预发规则编译或提交到服务端，编译错误会在MaxCompute Compiler窗口中显示详细信息。
```

具体操作

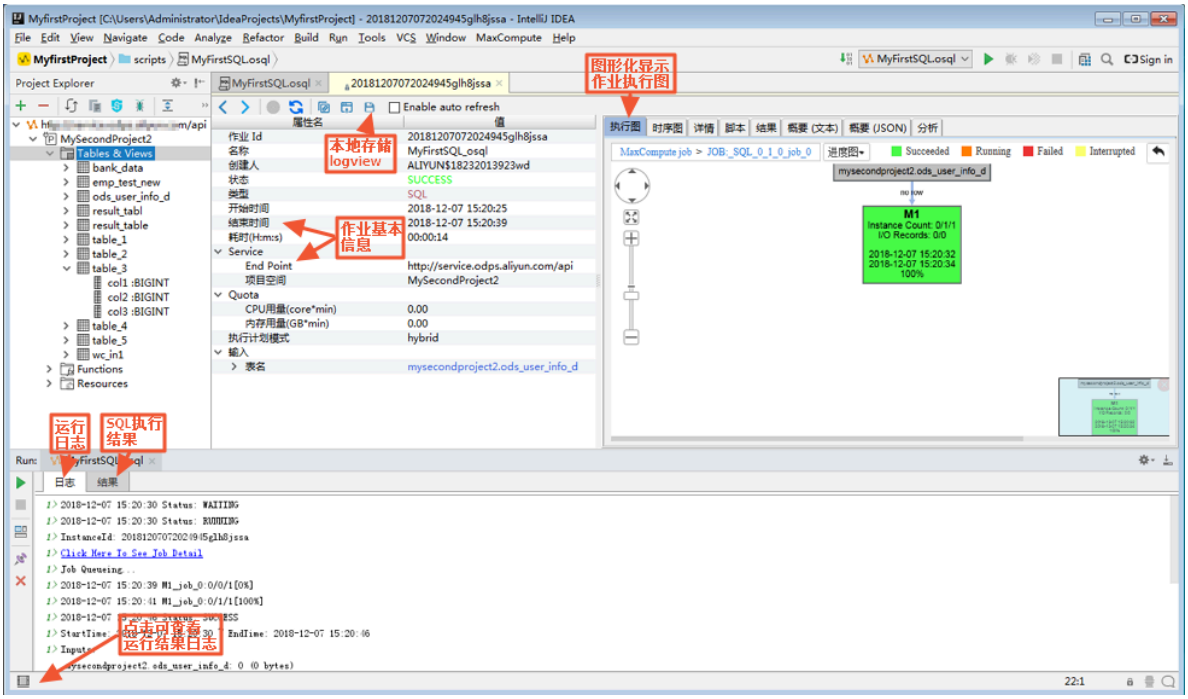
1. 编写好SQL语句后，单击工具栏或侧边栏上的绿色运行图标，或者右键Script Editor，选择Run MaxCompute SQL Script，即可提交到服务端。当SQL中存在变量时（如下图的${bizdate}），会弹出对话框，提示您输入变量值。



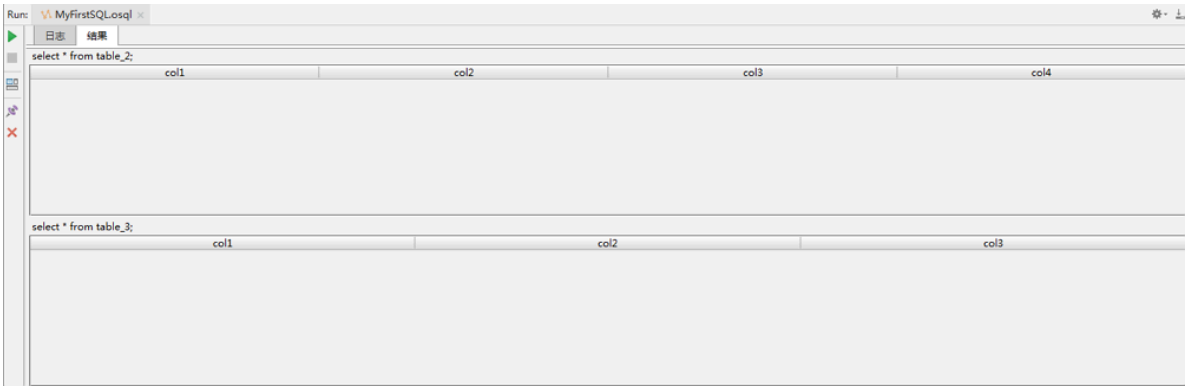
在您的SQL任务运行前，IDEA会向您提示预估的SQL费用。



2. SQL会先被本地编译(依赖于你在Project Explore窗口中添加的项目元数据)，无编译错误后就会提交到服务端执行。SQL执行过程中会显示运行日志，当已经开始在服务端运行时，会打开job detail页，显示作业运行的基本信息以及执行图：



3. 可以在结果页查看SQL结果，单句模式下当存在多条语句时，会显示每条语句的结果。可以选择表格中的一些行或列，Ctrl + C到剪切板中。



2.6 开发Java程序

2.6.1 创建MaxCompute Java Module

MaxCompute Studio能支持用户开发Java UDF和MR，首先需要新建一个MaxCompute Java Module。

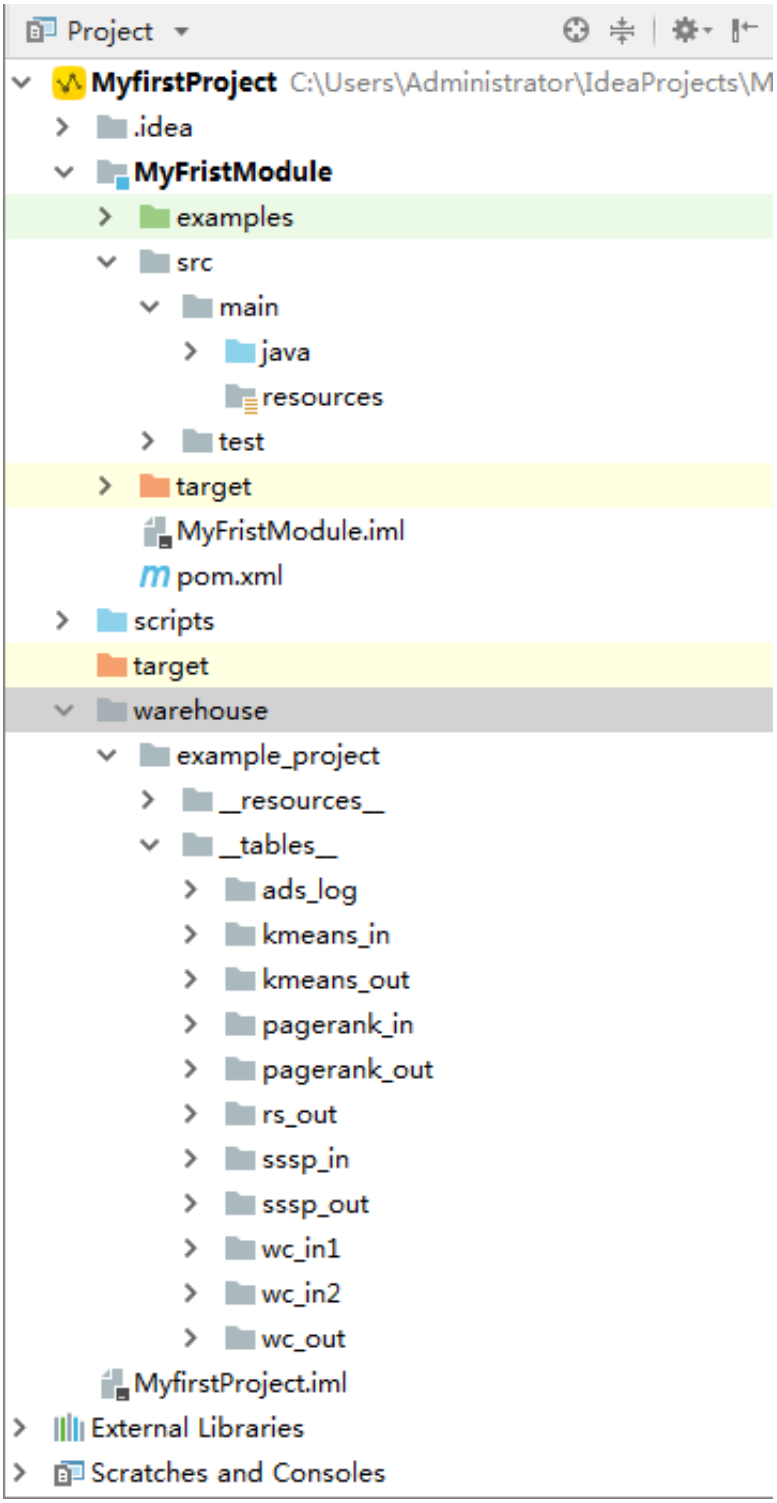
创建module

依次单击File > new > module，module类型为`MaxCompute Java`，配置`Java JDK`。单击next，输入module名，单击finish。studio会帮用户自动创建一个maven module，并引入MaxCompute相关依赖，具体请查看pom文件。

module结构说明

至此，一个能开发MaxCompute Java程序的module已建立，如下图的mDev。主要目录包括:

- src/main/java：用户开发java程序源码。
- examples：示例代码，包括单测示例，用户可参考这里的例子开发或编写UT。
- warehouse：本地运行时需要的schema和data。

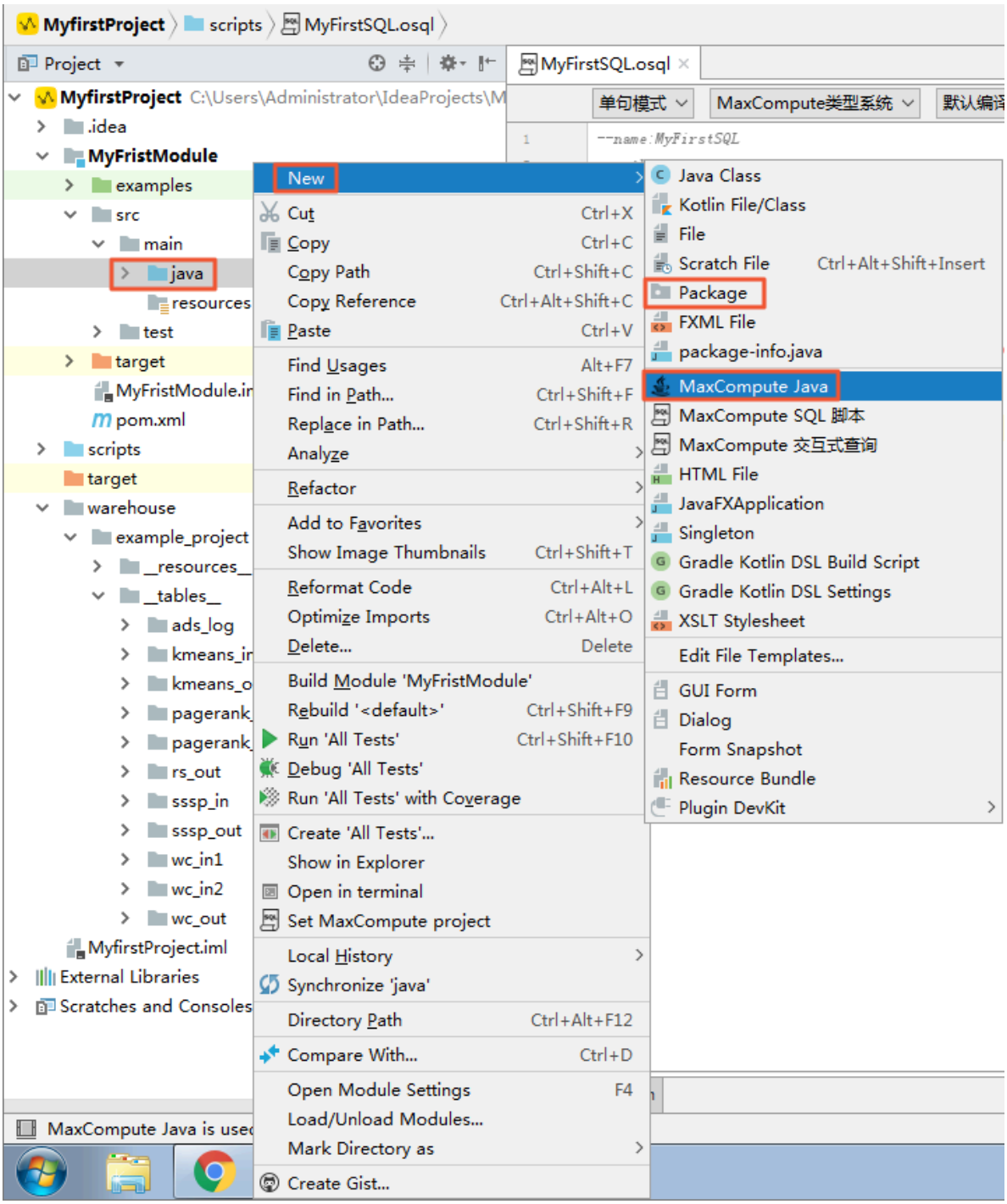

Project
MyfirstProject C:\Users\Administrator\IdeaProjects\M
.idea
MyFristModule
examples
src
main
java
resources
test
target
MyFristModule.iml
pom.xml
scripts
target
warehouse
example_project
__resources__
__tables__
ads_log
kmeans_in
kmeans_out
pagerank_in
pagerank_out
rs_out
sssp_in
sssp_out
wc_in1
wc_in2
wc_out
MyfirstProject.iml
External Libraries
Scratches and Consoles

2.6.2 开发和调试UDF

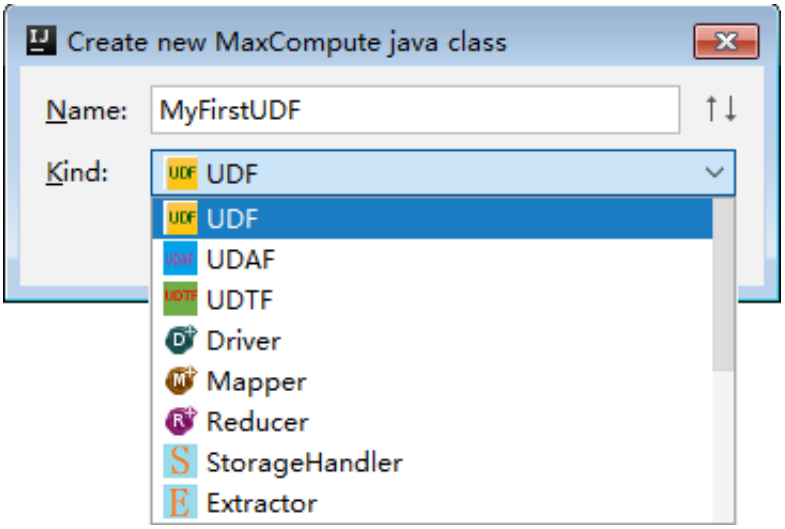
创建完成*MaxCompute Java Module* 后，即可开发UDF。

操作步骤

1. **展开已创建的MaxCompute Java Module目录，导航至src > main > java > new，单击MaxCompute Java 。如下图所示：**

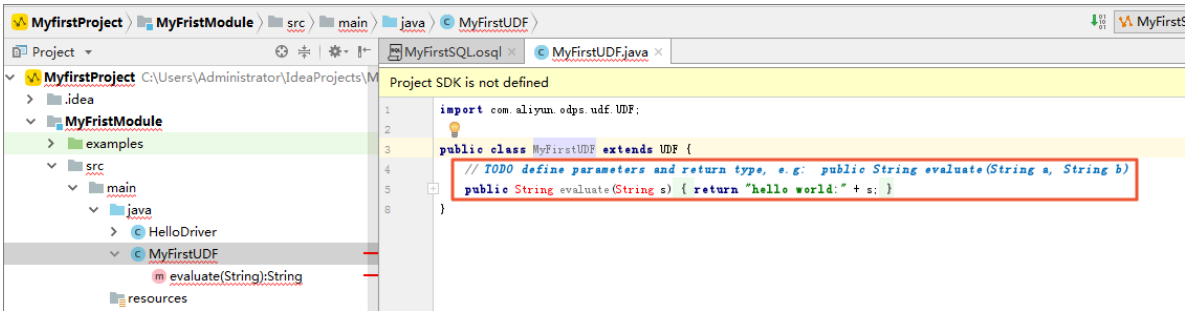


2. 填写name和kind，单击OK。如下图所示：



- Name：填写创建的MaxCompute Java Class名称，如果还没创建package，可以在此处填写 packagename.classname，会自动生成package。
- Kind：选择类型。目前支持的类型有：自定义函数（UDF/UDAF/UDTF）、MapReduce（Driver/Mapper/Reducer）、非结构化开发（StorageHandler/Extractor）等。

3. 创建成功后，即可进行开发、编辑、测试Java程序。



说明：

这里的代码模板可在Intellij中自定义，具体的：Preference > Editor > File > Code Templates，然后在Code标签页中寻找MaxCompute对应的模板修改。

通常情况下，JAVA UDF的开发可以通过以下几种方式：

- 使用MaxCompute Studio完成JAVA UDF开发整个流程。
- 使用*Eclipse*插件开发和调试*JAVA UDF*，导出Jar包，然后通过命令或者DataWorks添加资源后再注册函数。

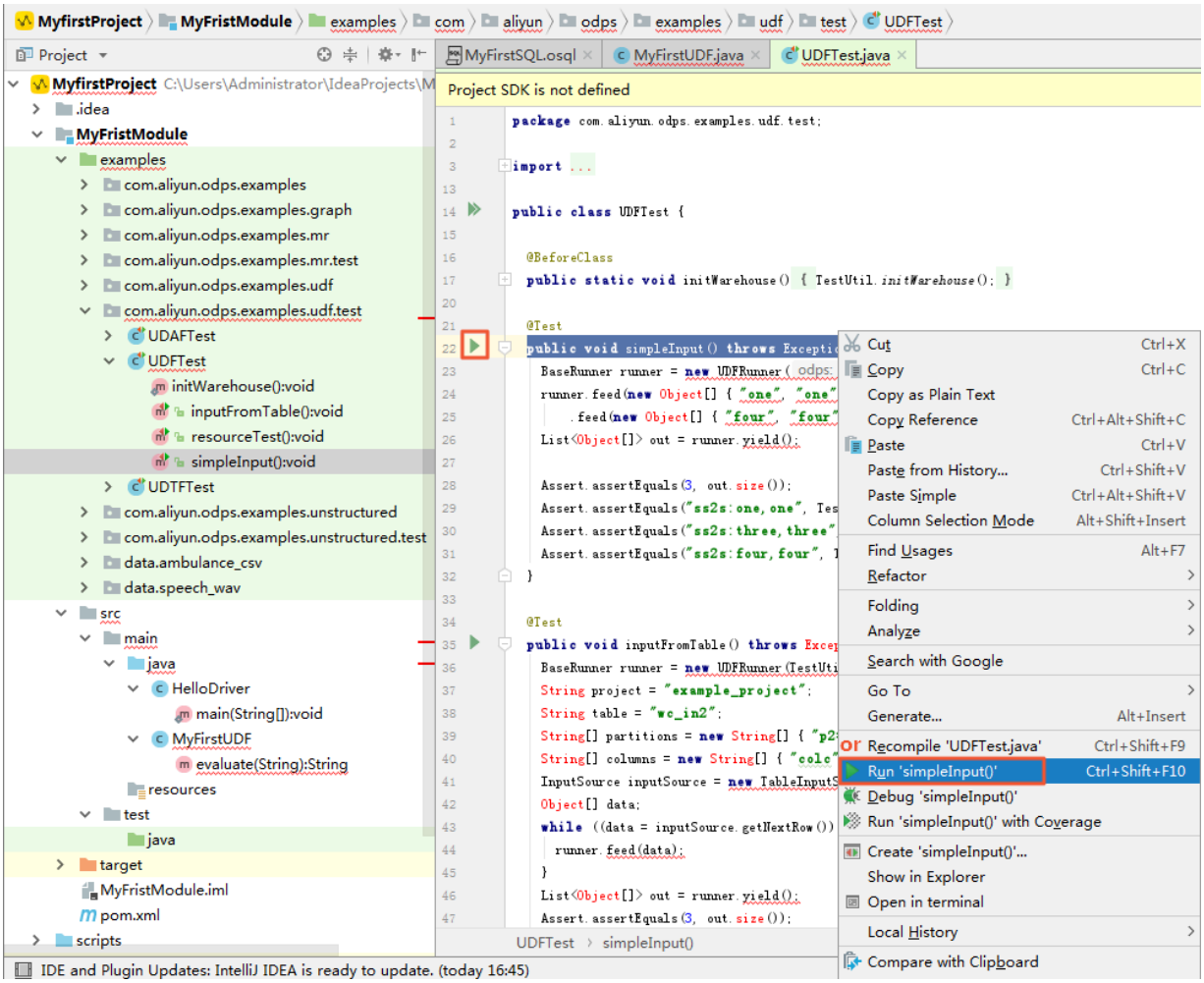
了解详细开发步骤，请参见*JAVA UDF*开发#可选#。

调试UDF

开发UDF完成后，即可通过单元测试和本地运行两种方式进行测试，看是否符合预期。

单元测试

在examples目录下有各种类型的单元测试示例，可参考示例编写自己的Unit Test。



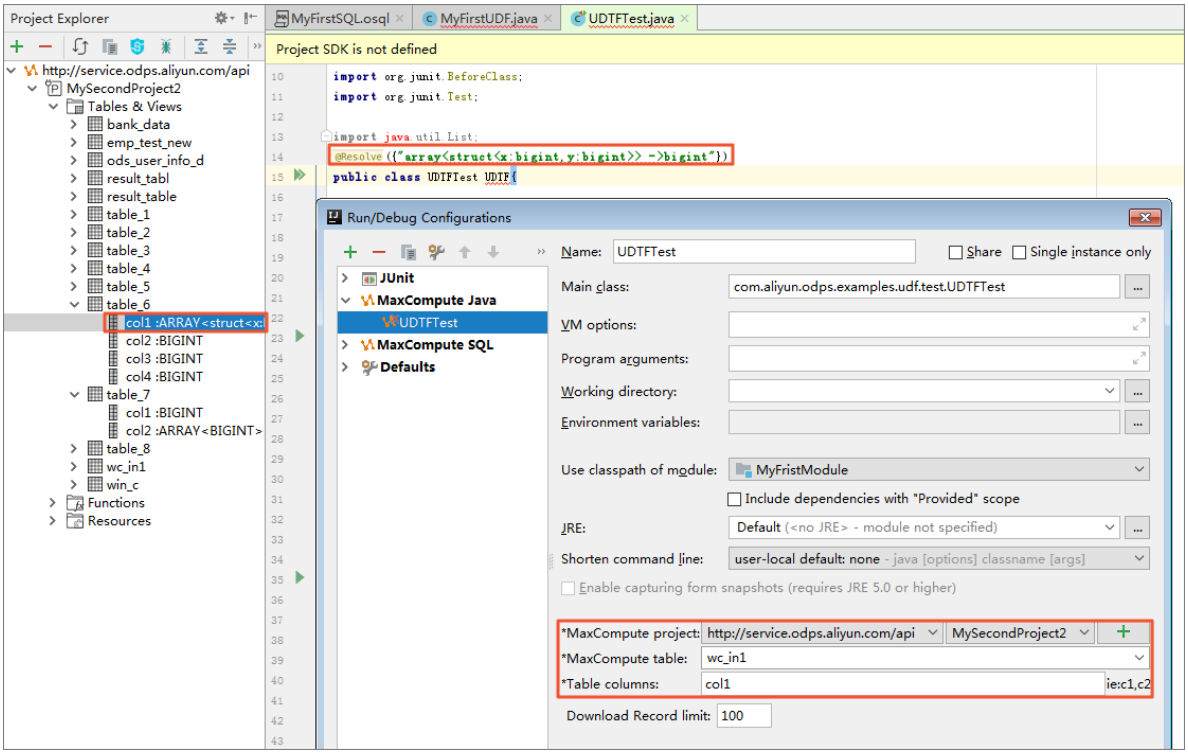
本地运行

本地运行UDF时，需要指定运行数据源，有以下两种方式设定测试数据源：

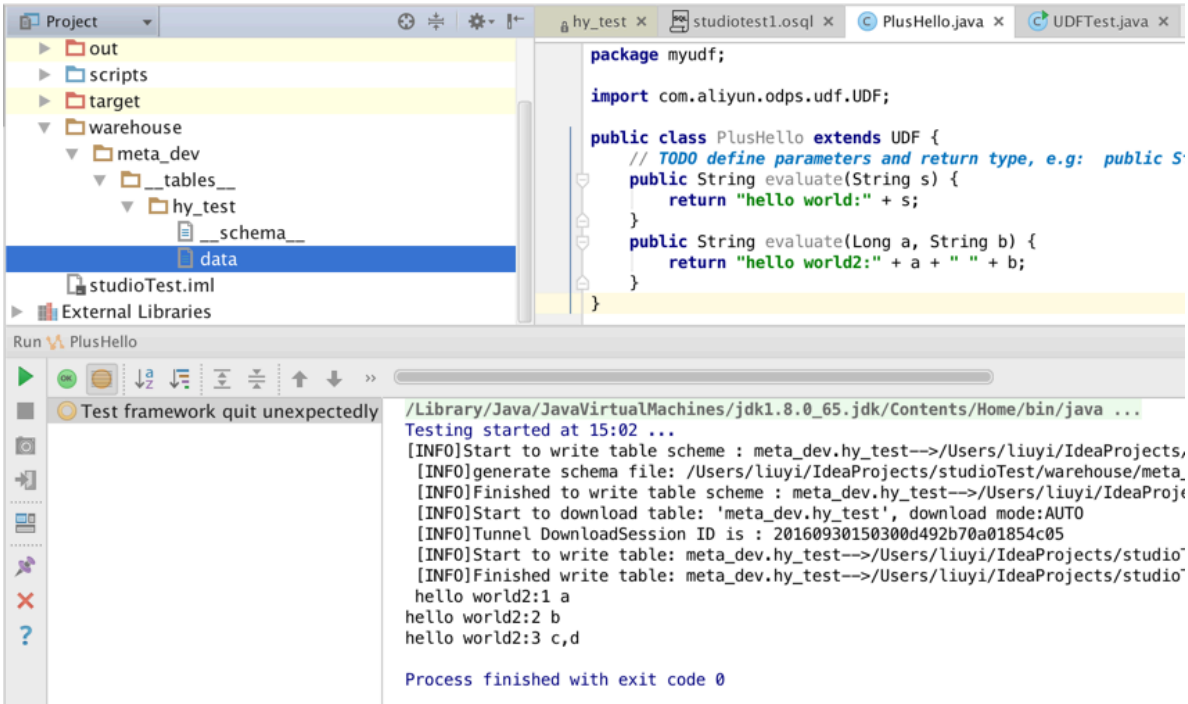
- MaxCompute Studio通过Tunnel服务自动下载指定项目下的表数据到warehouse目录下。
- 提供Mock项目及表数据，您可参考warehouse下的example_project自行设置。

操作步骤

1. 右击UDF类，单击运行，弹出run configuration对话框。UDF/UDAF/UDTF一般作用于select子句中表的某些列，需要配置MaxCompute project，table和column（元数据来源于project explorer和warehouse下的Mock项目）。复杂类型的调试也是支持的，如下图：



2. 单击OK。

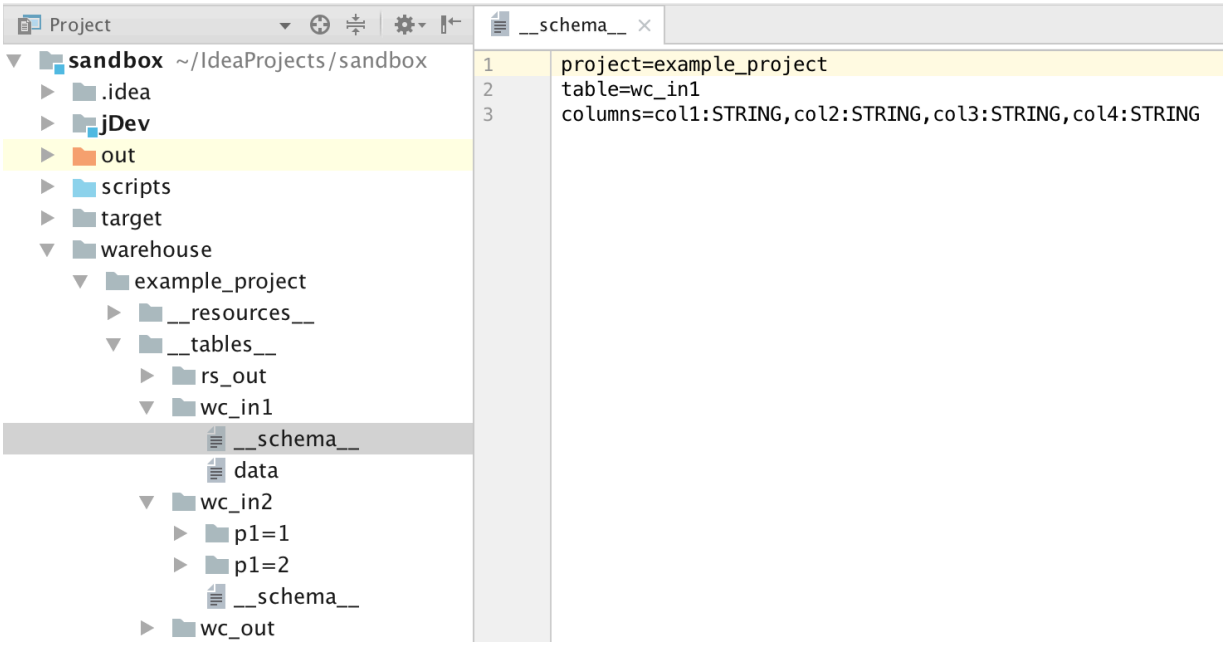


说明：

- 如果指定项目下的表数据未被下载到warehourse中，需要先下载数据，默认下载100条。默认下载100条，如需更多数据，可配置Download record limit项。
- 如果采用Mock项目或已下载数据，则直接运行。
- UDF的local run框架会将warehouse中指定列的数据作为UDF的输入，开始本地运行UDF，您可以在控制台看到日志输出和结果打印。

本地warehouse目录

本地warehouse用于存储表（包括meta和数据）或资源，用于本地执行UDF或MR。warehouse目录如下图所示：



说明：

- warehouse目录下依次是项目名，tables，表名，表schema和sample data。
- schema文件依次配置项目名，表名，以及列名和类型（冒号分隔），分区表还需配置分区列（非分区表参考wc_in1，分区表参考wc_in2）。

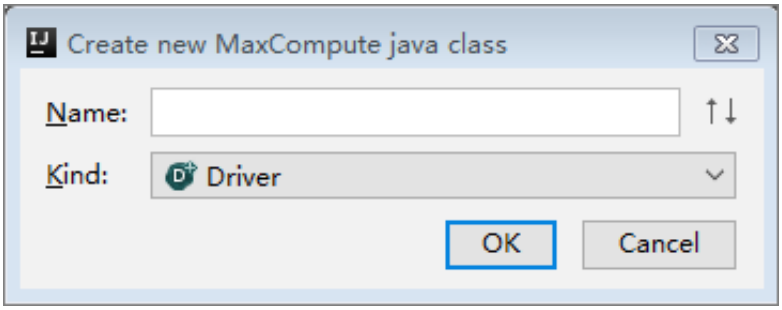
- data文件采用标准csv格式存储表的sample数据：
  - 特殊字符为逗号，双引号和换行（\n或\r\n）
  - 列分隔符为逗号，行分隔符为\n或\r\n
  - 如果列内容里包含特殊字符，需要在该列内容前后加上双引号，例如：3,No -> “3, No”
  - 如果列内容包含双引号，则每个双引号转义成两个双引号，例如：a”b”c -> “a”” b”” c”
  - \N表示该列为null，如果该列内容（string 类型）就是\N，需要转换为 “”” \N”””
  - 文件字符编码为UTF-8

2.6.3 开发MapReduce

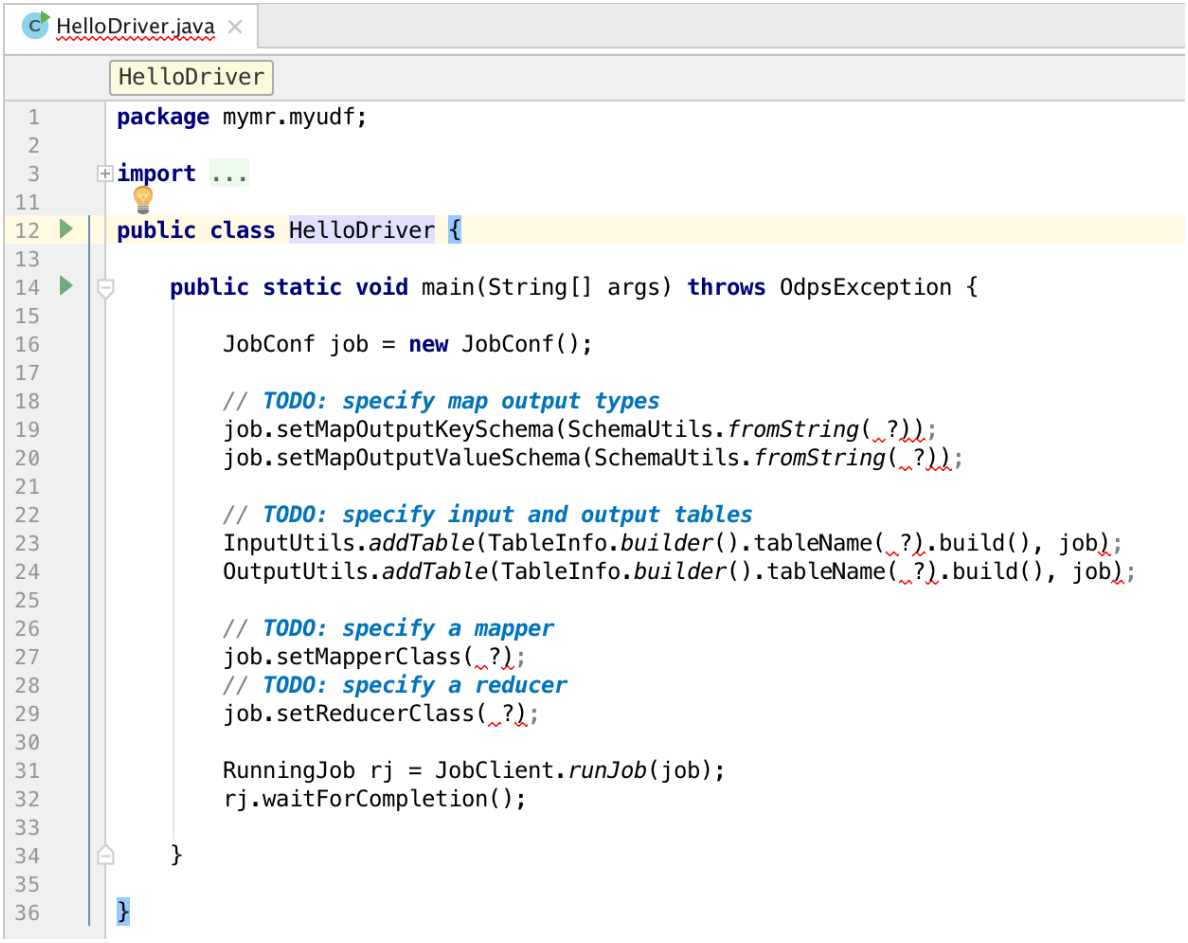
创建完成 *MaxCompute Java Module*后，即可以开始开发*MR*了。

开发MR

1. 在module的源码目录即src > main上右键new > java，选择MaxCompute Java。
2. 分别创建Driver，Mapper，Reducer。



3. 模板已自动填充框架代码，只需要设置输入/输出表，Mapper/Reducer类等即可。

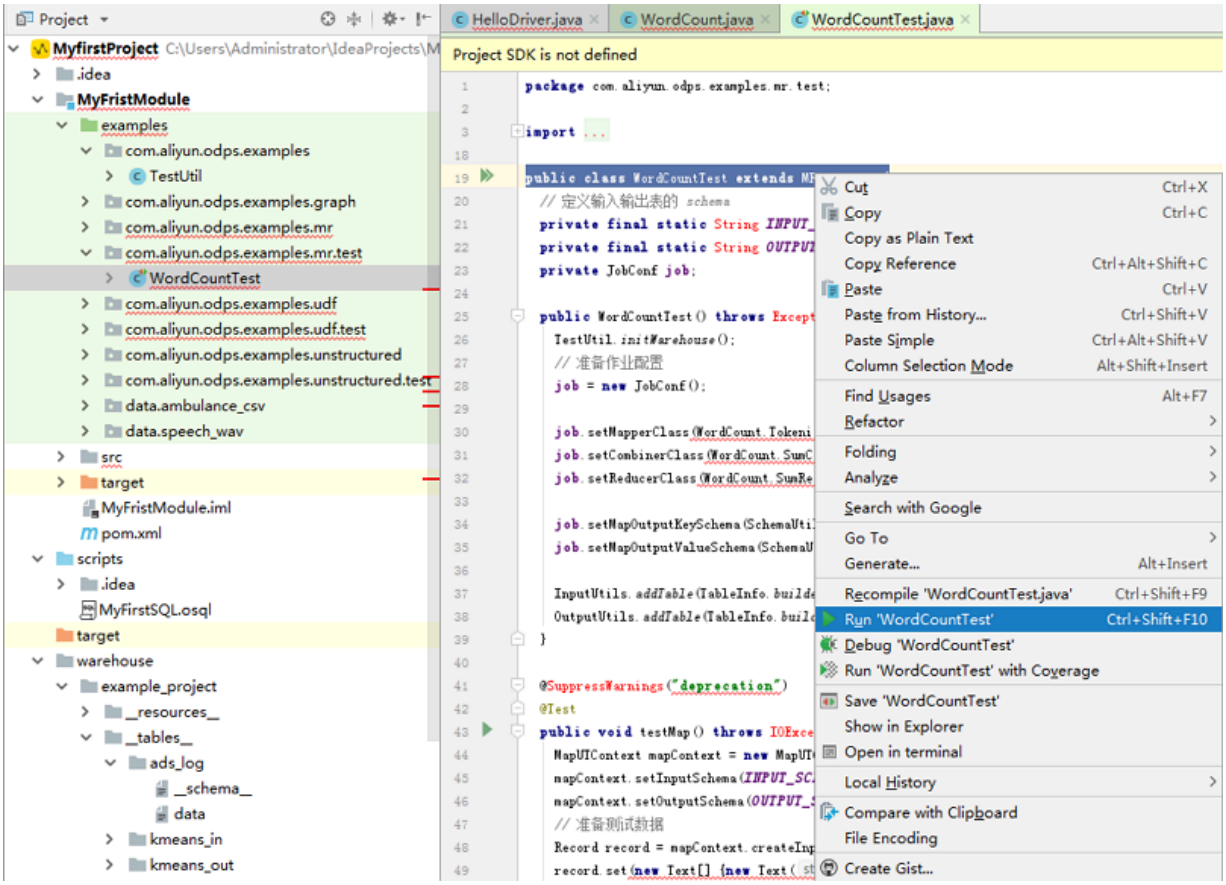


开发MR详情请参见编写MapReduce#可选#。

调试MR

MR开发好后，下一步就是要测试自己的代码，看是否符合预期，我们支持两种方式:

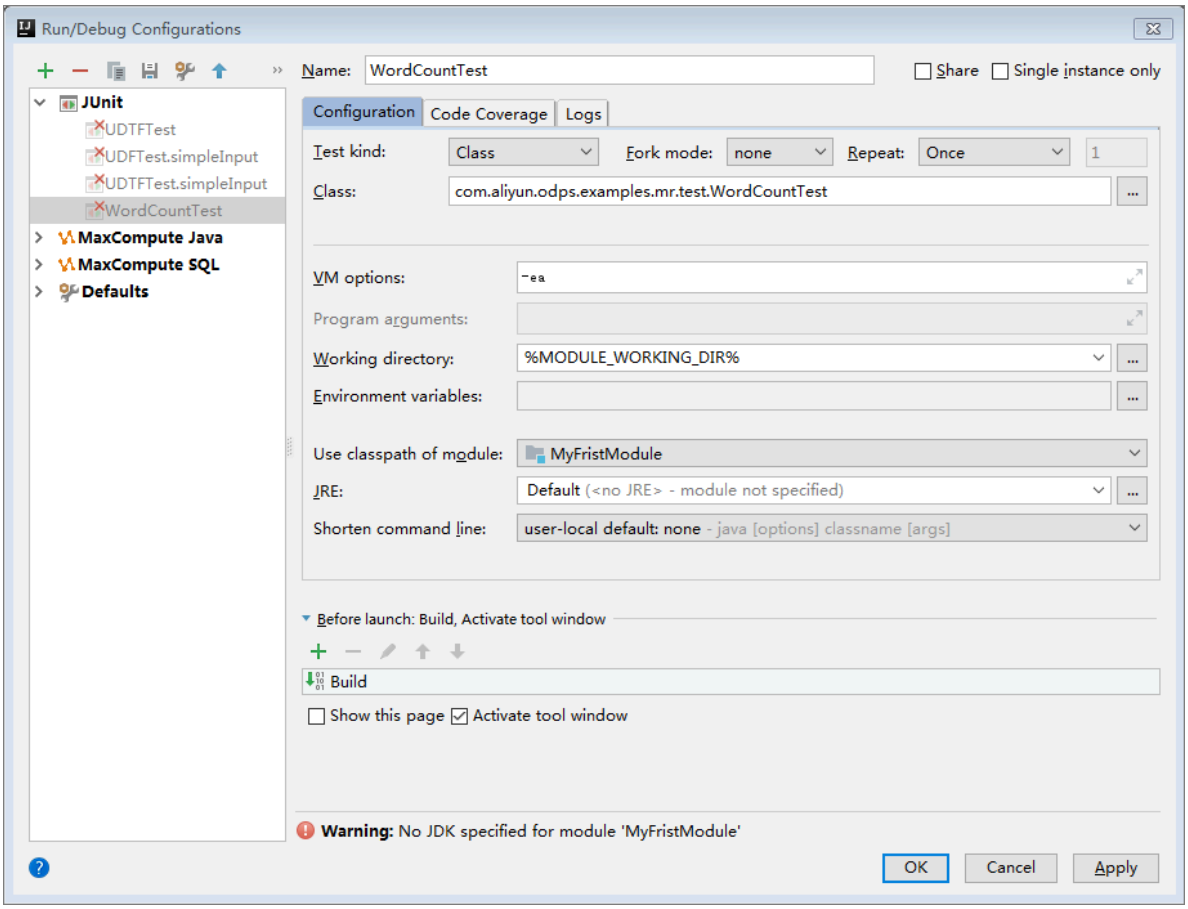
单元测试：在examples目录下有WordCount的单测实例，可参考例子编写自己的UT。



本地运行MR：本地运行时，需要指定运行数据源，有两种方式设定测试数据源：

- **studio通过tunnel服务自动下载指定MaxCompute project的表数据到warehouse目录下。默认下载100条，如需更多数据测试，请自行使用console的tunnel命令或者studio的表下载功能。**
- **提供mock项目(example_project)及表数据，用户可参考warehouse下example_project自行设置。**

1. 运行MR: 在Driver类上右键，点击运行菜单，弹出run configuration对话框，配置MR需要在哪个MaxCompute Project上运行即可。



2. 单击OK，如果指定MaxCompute project的表数据未被下载到warehouse中，则首先下载数据；如果采用mock项目或已被下载则跳过。接下来，MR local run框架会读取warehouse中指定表的数据作为MR的输入，开始本地运行MR，用户可以在控制台看到日志输出和结果打印。

生产运行MR

本地调试通过后，接下来就可以把MR发布到服务端，在MaxCompute分布式环境下运行了：

1. 首先，你需要将自己的MR程序打成jar包，并发布到服务端。详细操作步骤请参考打包、上传及注册。
2. 通过MaxCompute console（无缝集成于studio），打开Project Explorer Window，右键单击project，选择Open in Console，可在console命令行中输入类似如下的jar命令：

```
jar -libjars wordcount.jar -classpath D:\odps\clt\wordcount.jar com.
aliyun.odps.examples.mr.WordCount wc_in wc_out;
```

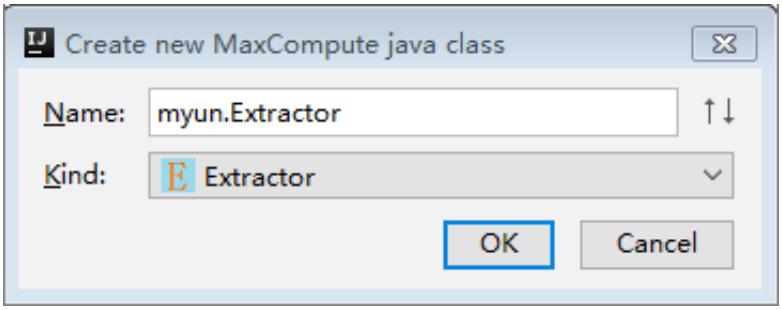
详细命令输入参请见jar命令。

2.6.4 非结构化开发

MaxCompute2.0新增了一套非结构化数据处理框架，支持通过外部表的方式直接访问OSS、OTS等。Studio对此提供了一些代码模板支持，方便用户快速开发。

编写StorageHandler/Extractor/Outputter

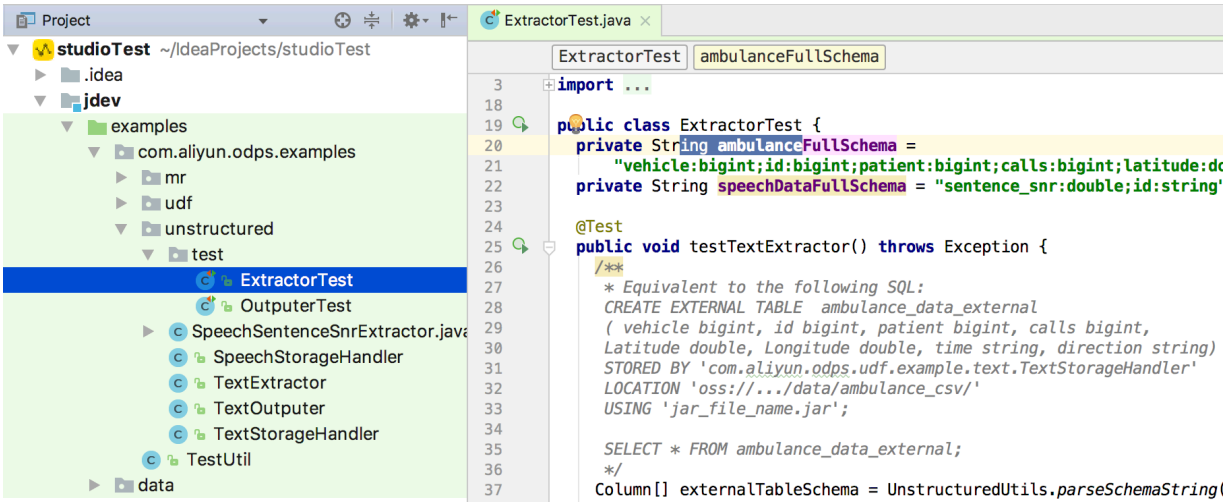
1. 创建MaxCompute Java Module(在examples目录下的unstructured文件夹有示例代码供参考)。
2. 在module的源码目录即src > main上右键new > java，选择MaxCompute Java。
3. 输入包名.类名，如myun.MyExtractor，选择类型Extractor，点击OK。



4. 模板已自动填充框架代码，只需要编写自己的逻辑代码即可。
5. 类似上述步骤可分别完成Outputter和StorageHandler的编写。

单元测试

可参考examples目录下的例子编写unit test，测试自己的Extractor/Outputter。

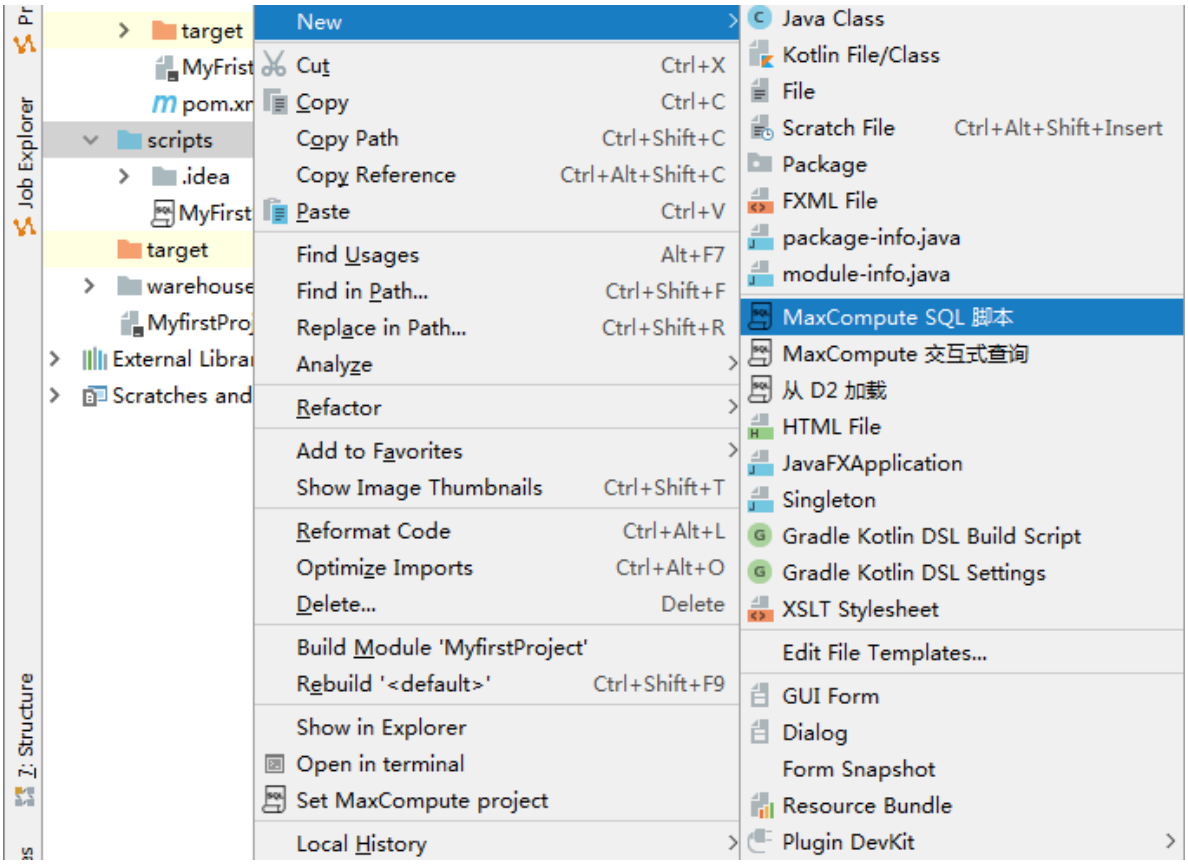


打包上传

StorageHandler/Extractor/Outputter写好后，可以参考打包发布将已写好的java程序打成jar包，并作为resource上传到服务端。

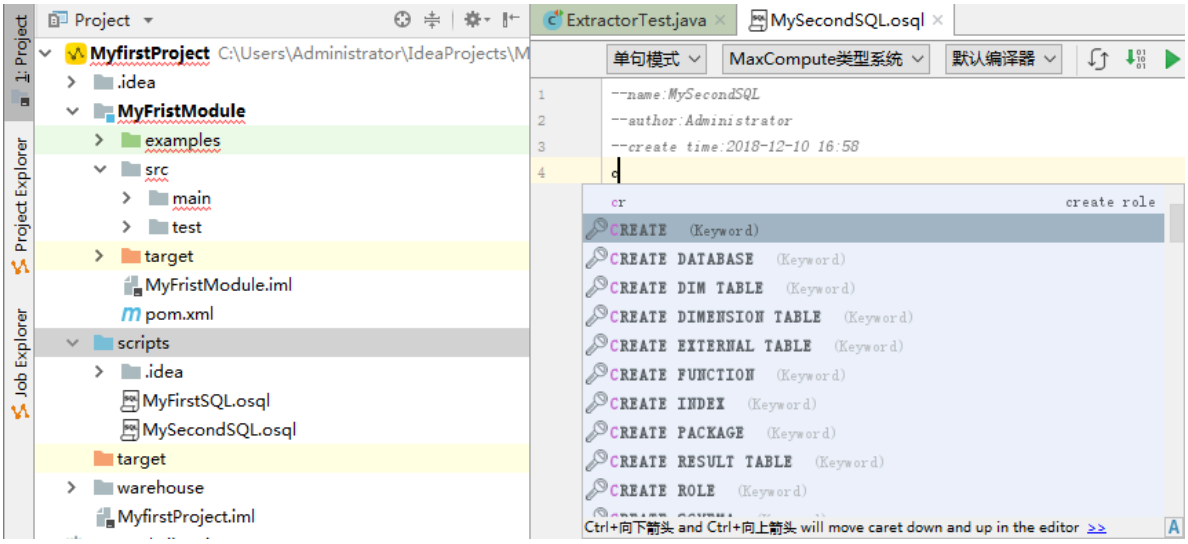
创建外部表

1. 在scripts目录右键new > MaxCompute Script。

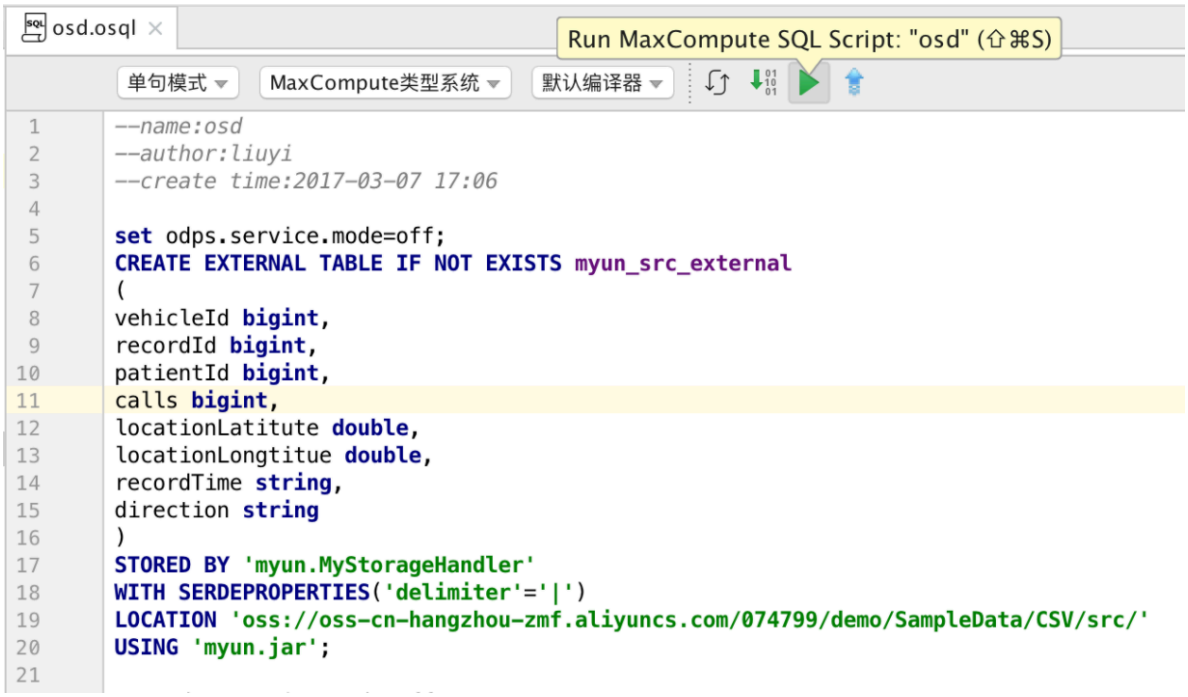


2. 输入SQL脚本名，Target Project选择脚本将要在哪个MaxCompute project下执行，点击OK。

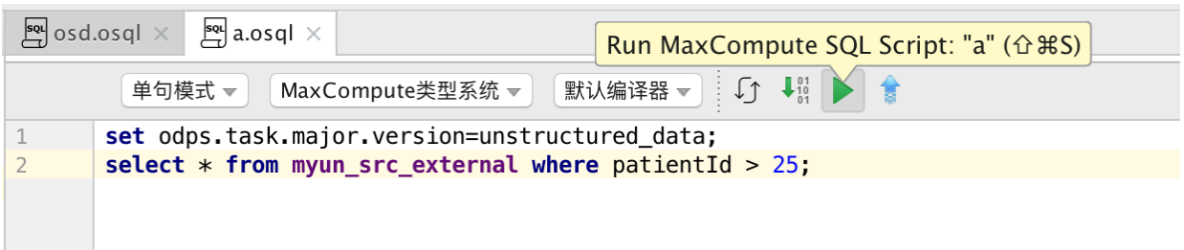
3. editor中唤出create external table live template，能快速插入创建外部表脚本模板：



然后修改外部表名称、列及类型、StorageHanlder类路径、配置参数、外部路径、jar名等，修改完成后点击运行脚本，创建该外部表。



4. 接下来，就可以查询该外部表了，类似：

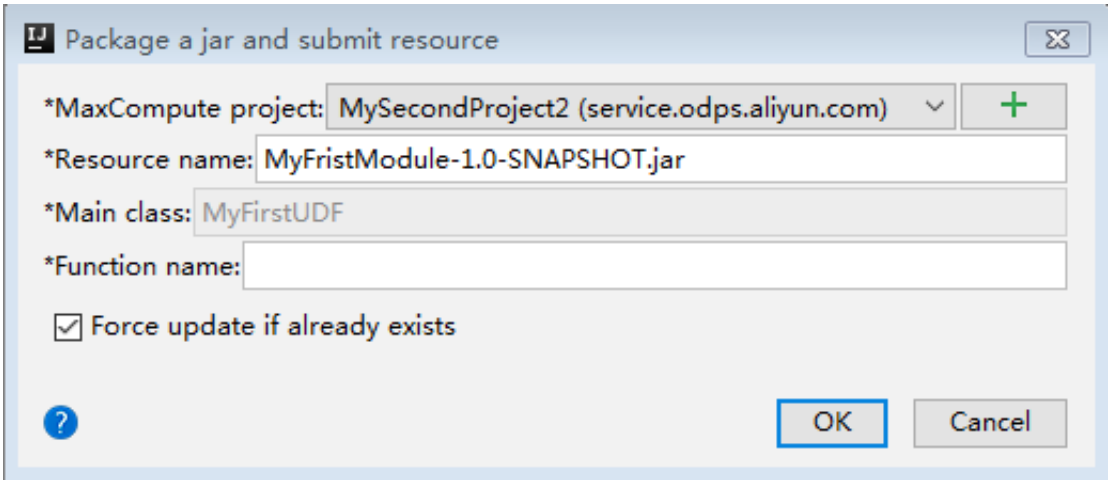


2.6.5 打包、上传和注册

完成*UDF*或 *MR*开发后，需要打包发布到MaxCompute系统。

打包UDF或MR

一个UDF或MR要想发布到服务端供生产使用，要经历打包 > 上传 > 注册三个步骤。针对此，我们提供了一键发布功能（studio会依次执行mvn clean package，上传jar和注册UDF三个步骤，一次完成）。具体的，在UDF或MR类上右键(该类必须在src > main > java子目录下，maven module里编译成功)，选择Deploy to server… 菜单，会弹出如下对话框，选择要部署的MaxCompute project，输入资源名和函数名，点击ok等待后台完成即可。



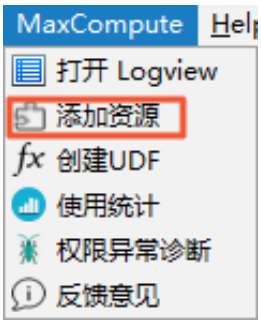
说明:

当然，如果你有特殊的打包需求，那么你可以自行修改pom.xml打包相关配置。打好包后，再依次手工操作如下的上传jar和注册UDF即可。

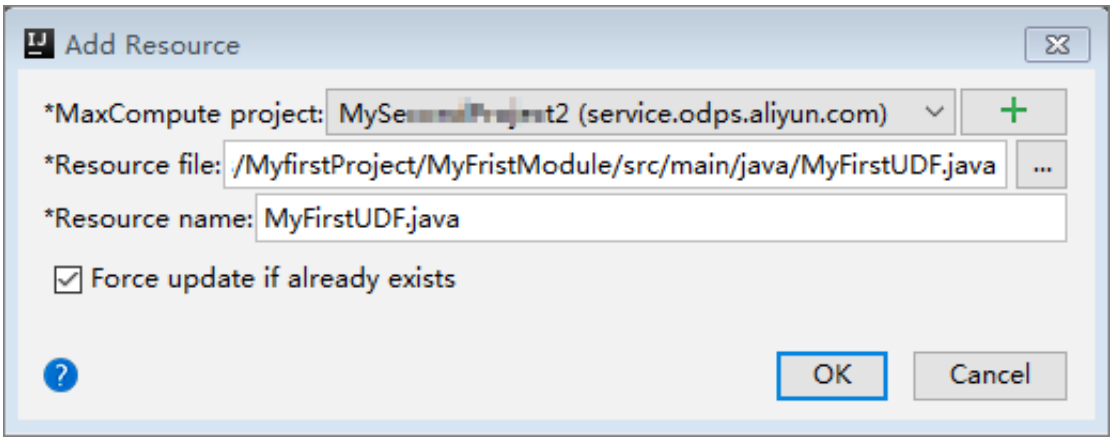
上传jar

打包成功后，就可以将该jar包上传到MaxComptute服务端:

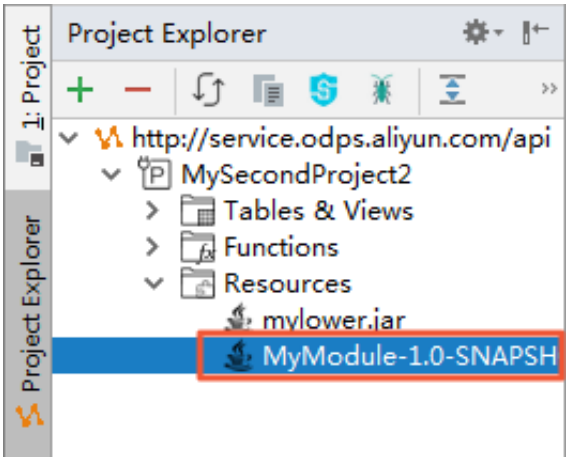
1. 在MaxCompute菜单选择Add Resource菜单项。



2. 选择要上传到哪个MaxCompute project上，jar包路径，要注册的资源名，以及当资源或函数已存在时是否强制更新，然后点击OK。



3. 上传成功后，可以在project explorer窗口的resources节点下看到该资源。

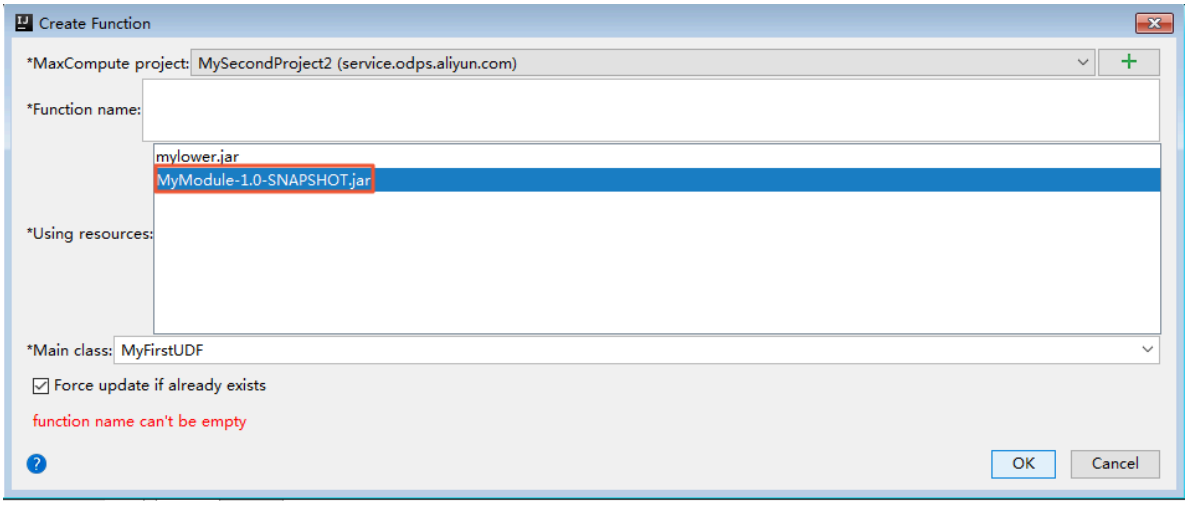


注册UDF

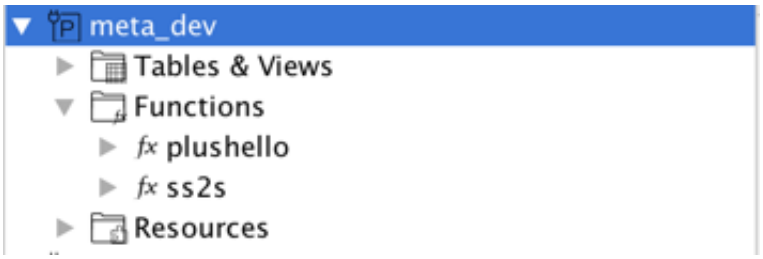
jar包上传完成后，就可以注册UDF函数了。

1. 在MaxCompute菜单选择Create Function菜单项。

2. 选择需要使用的资源jar、jar的主类，输入函数名，然后点击OK。

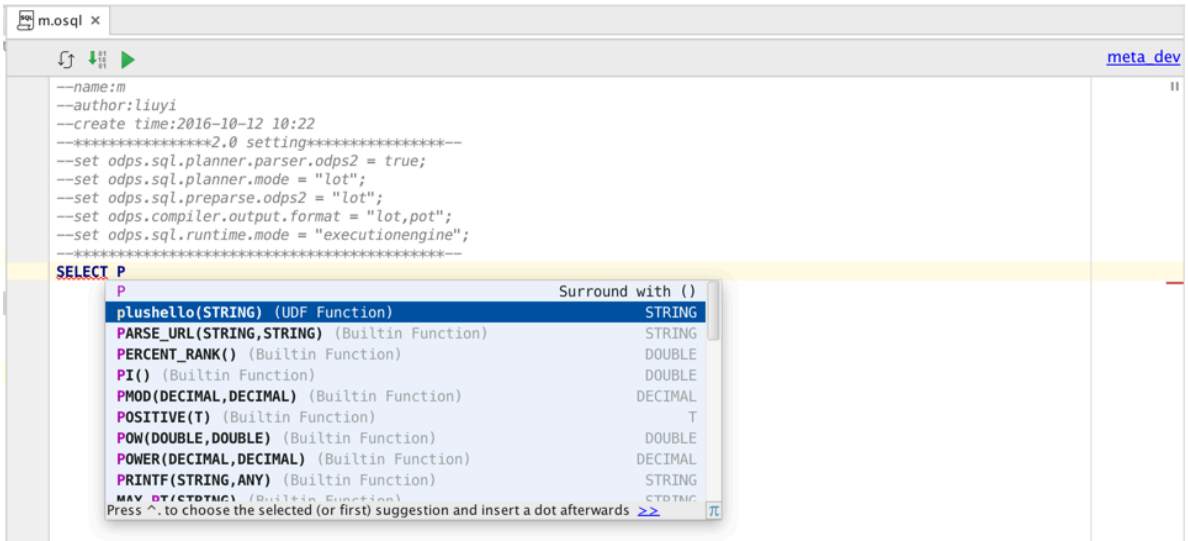


3. 注册成功后，可以在project explorer窗口的functions节点下看到该函数。



使用UDF

· 接下来就可以在SQL中使用新编写的UDF完成后续开发。

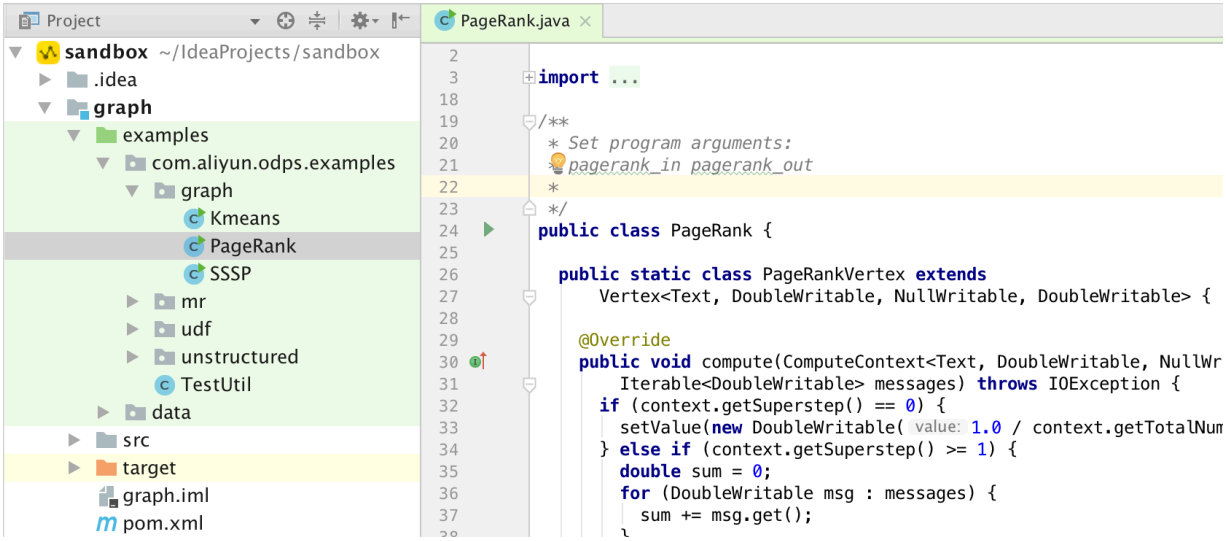


2.6.6 开发Graph

创建完成 *MaxCompute Java Module*后，即可以开始开发*Graph*了。

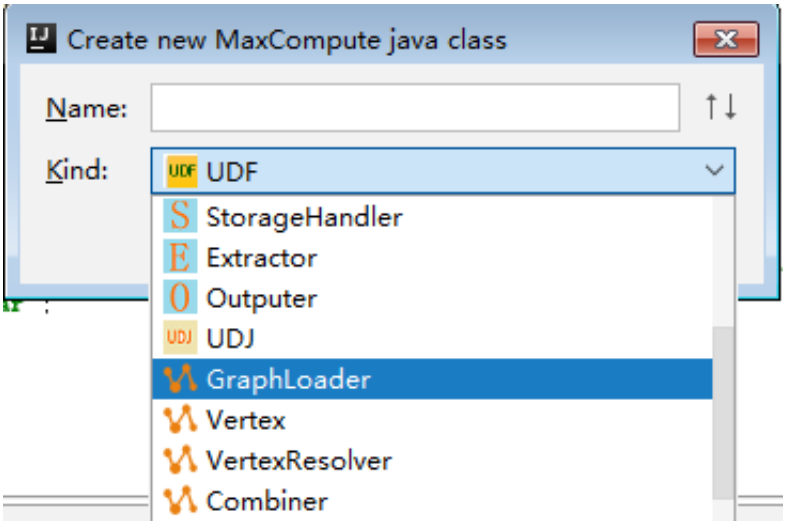
代码示例

在examples目录下有graph的一些代码示例，可参考示例熟悉Graph程序的结构。



编写Graph

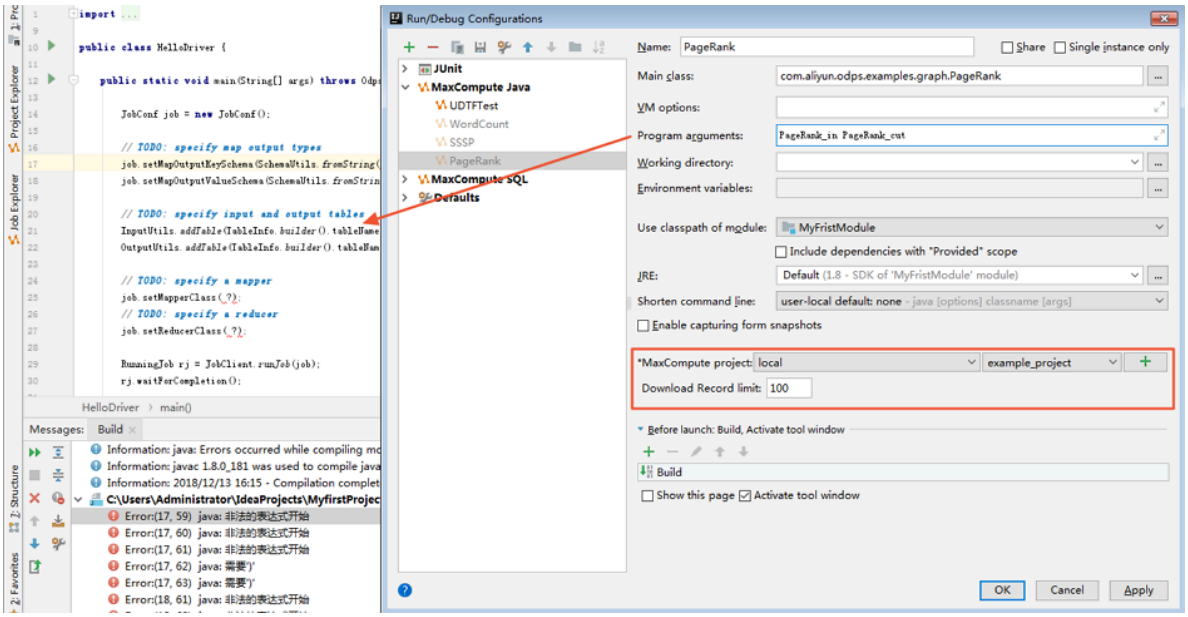
1. 在module的源码目录即src > main > java上右键new，选择MaxCompute Java。
2. 选择GraphLoader/Vertex等类型，Name框中输入类名(支持包名.类名)。点击OK，模板会自动填充框架代码，可在此基础上继续修改。



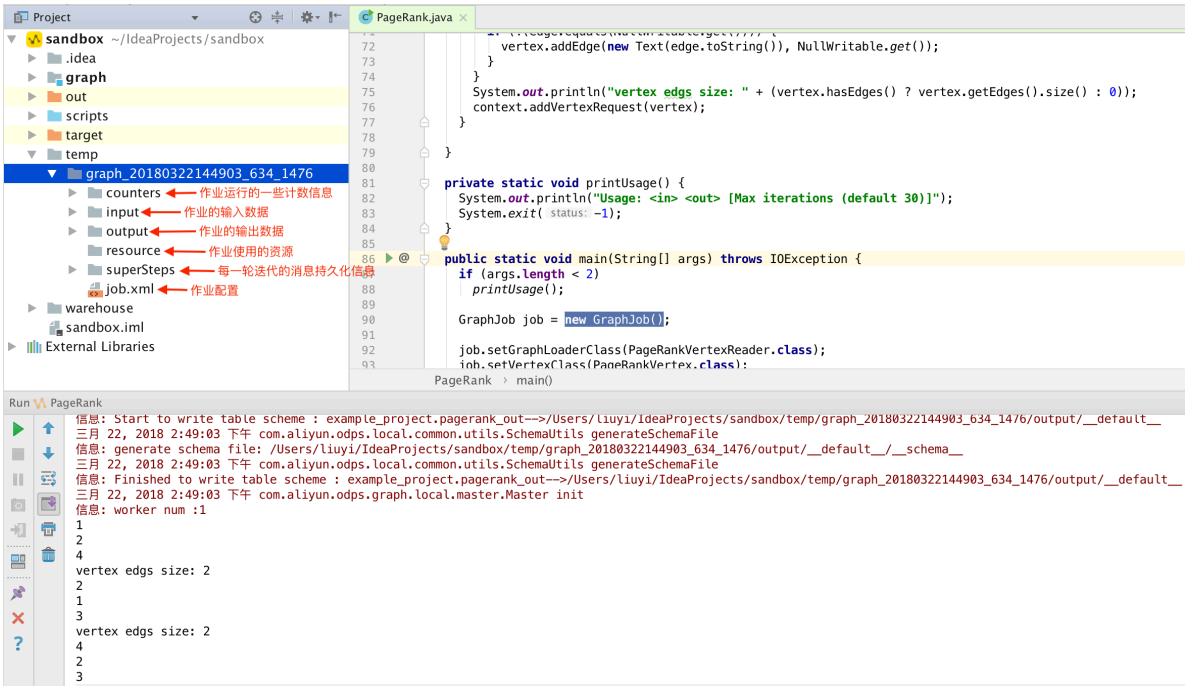
本地调试Graph

Graph开发好后，下一步就是要测试自己的代码，看是否符合预期。我们支持本地运行Graph，具体的：

1. 运行Graph: 在驱动类(有main函数且调用GraphJob.run方法)上右键，点击运行菜单，弹出run configuration对话框，配置Graph需要在哪个MaxCompute Project上运行即可。



2. 点击OK，如果指定MaxCompute project的表数据未被下载到warehouse中，则首先下载数据；如果采用mock项目或已被下载则跳过。接下来，graph local run框架会读取warehouse中指定表的数据作为输入，开始本地运行Graph，用户可以在控制台看到日志输出。每运行一次本地调试，都会在Intellij工程目录下新建一个临时目录，见下图：



说明:

关于warehouse的详细介绍请参考开发UDF中本地warehouse目录部分。

生产运行Graph

本地调试通过后，接下来就可以把Graph发布到服务端，在MaxCompute分布式环境下运行了：

1. 首先，将自己的Graph程序打成jar包，并发布到服务端。如何打包发布?
2. 通过Studio无缝集成的MaxCompute Console（在Project Explorer Window的Project上右键，选择Open in Console），在Console命令行中输入类似如下的jar命令：

```
jar -libjars xxx.jar -classpath /Users/home/xxx.jar com.aliyun.odps.
graph.examples.PageRank pagerank_in pagerank_out;
```

更详细的Graph开发介绍请参见编写Graph#可选#。

2.7 开发Python程序

2.7.1 Python开发使用须知

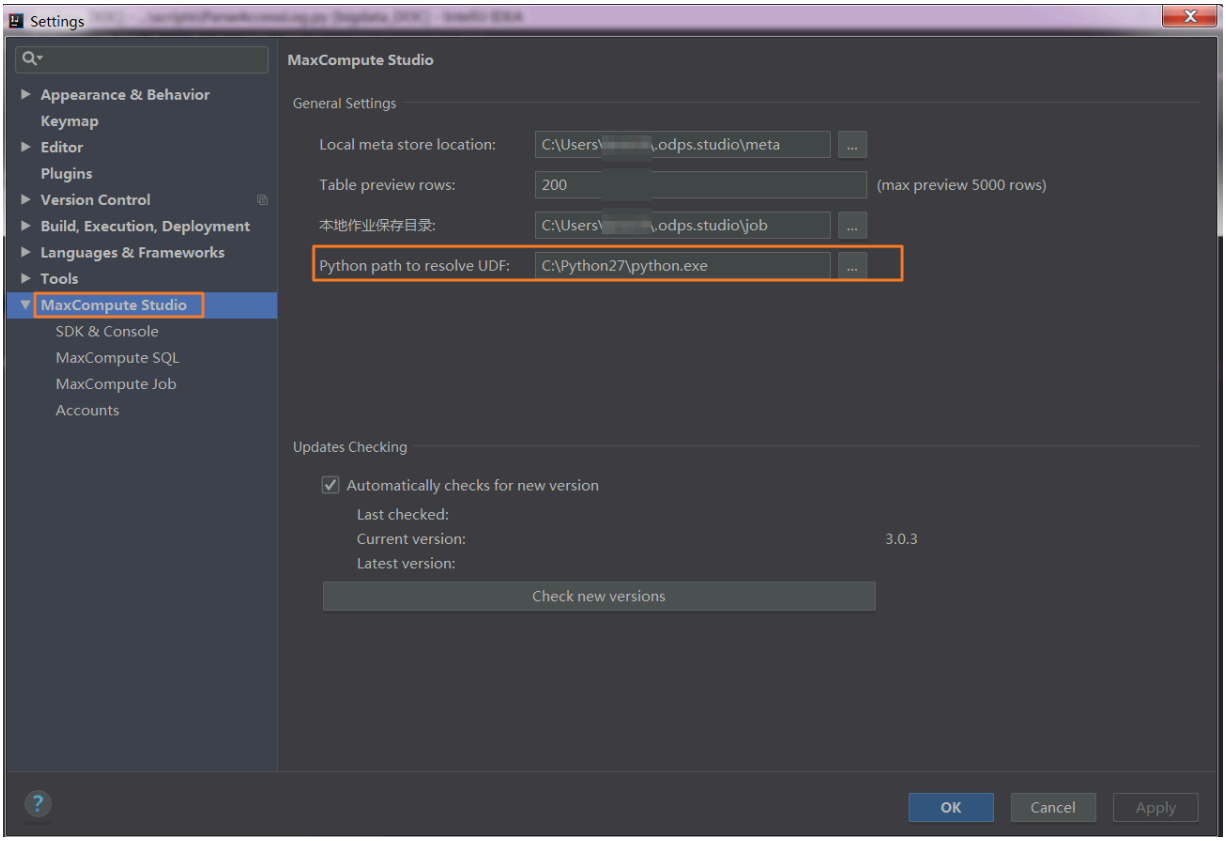
MaxCompute Studio支持您在Intellij IDEA中完成Python相关的开发，包括UDF和PyODPS脚本，但使用前必须安装Python、PyODPS和IDEA的Python插件。

在开始Python项目开发前，我们需要下载相关版本的Python配置Python环境变量。本文使用的Python版本为Python2.7。

安装PyODPS

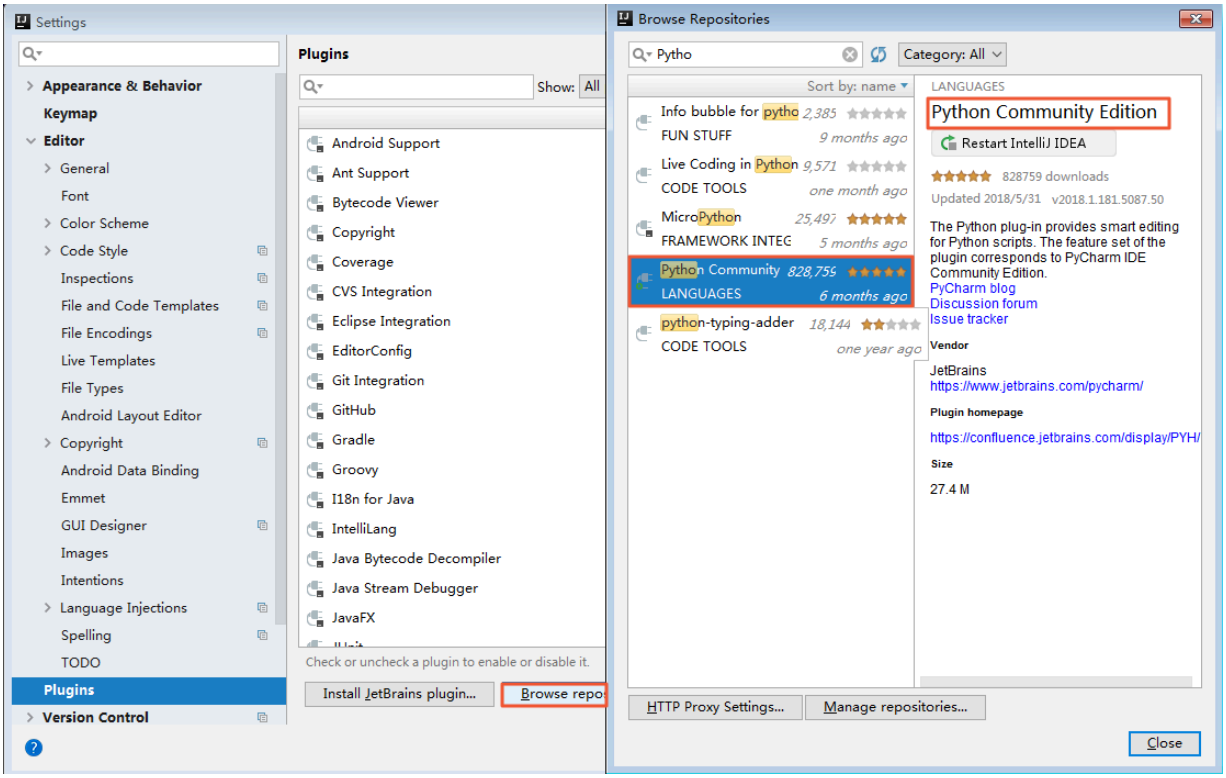
PyODPS是MaxCompute的PyODPS SDK，详情请参见Python SDK。

在Windows环境下，请务必在File > Settins > MaxCompute Studio保证您的Python安装路径设置正确，用于解析Python UDF，举例如下。



安装Python插件

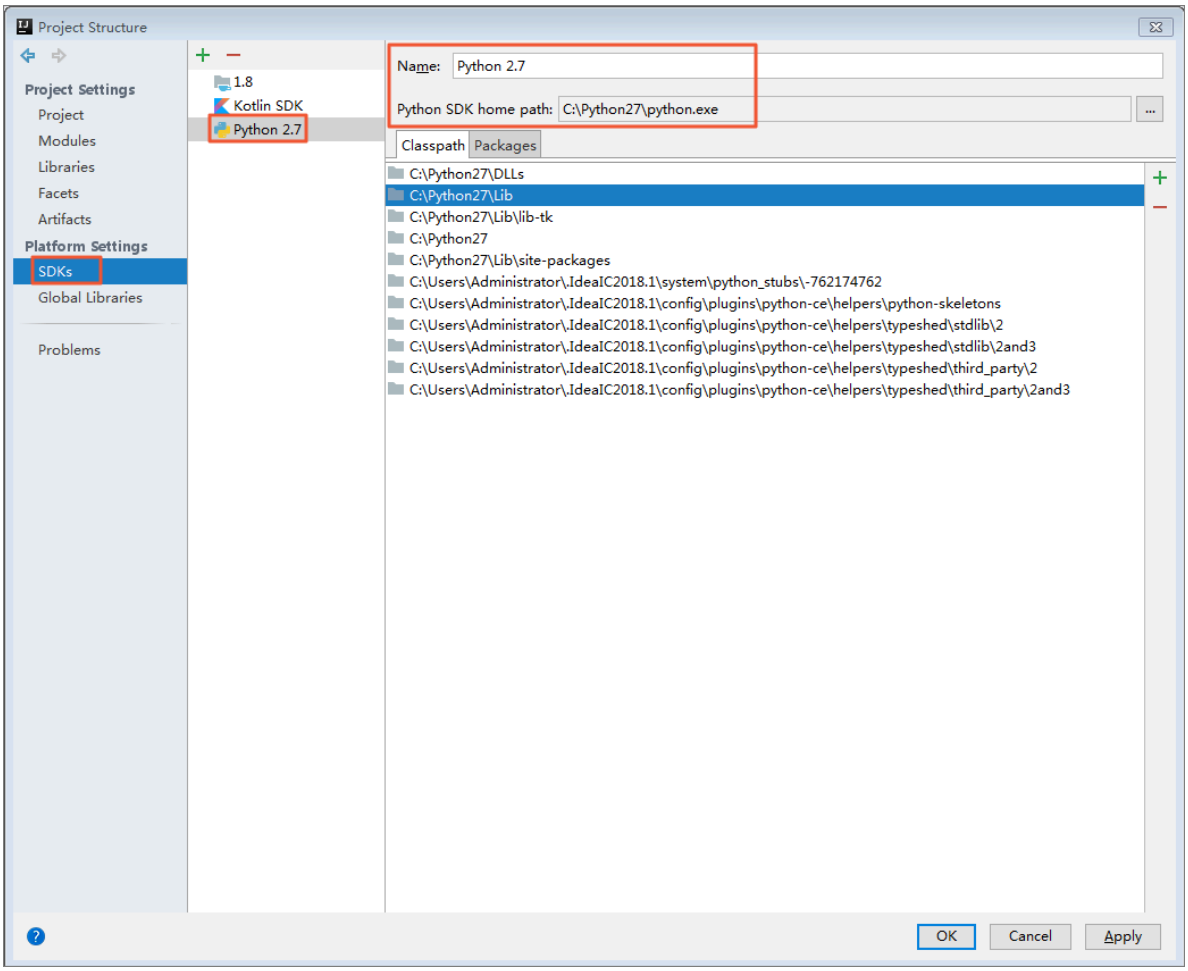
在Intellij IDEA的插件仓库中搜索Python或者Python Community Edition插件并安装。



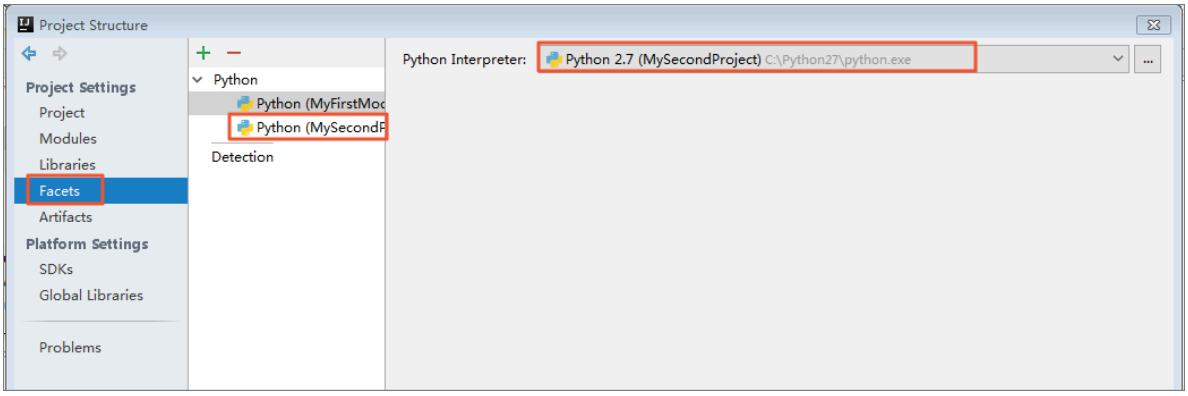
配置Python依赖

配置Studio Module对Python的依赖，即可进行MaxCompute Python的开发。

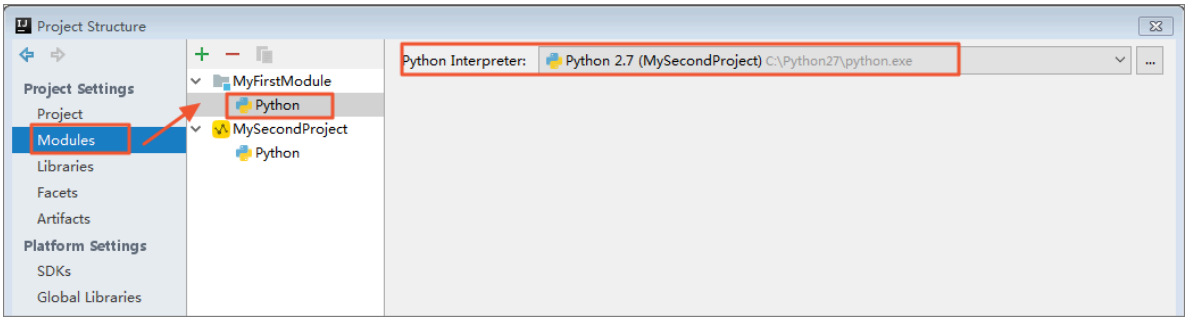
1. 导航至File > Project Structure，添加Python SDK。



2. 导航至File > Project Structure，添加Python Facets。



3. 导航至File > Project Structure，配置Module依赖Python Facets。



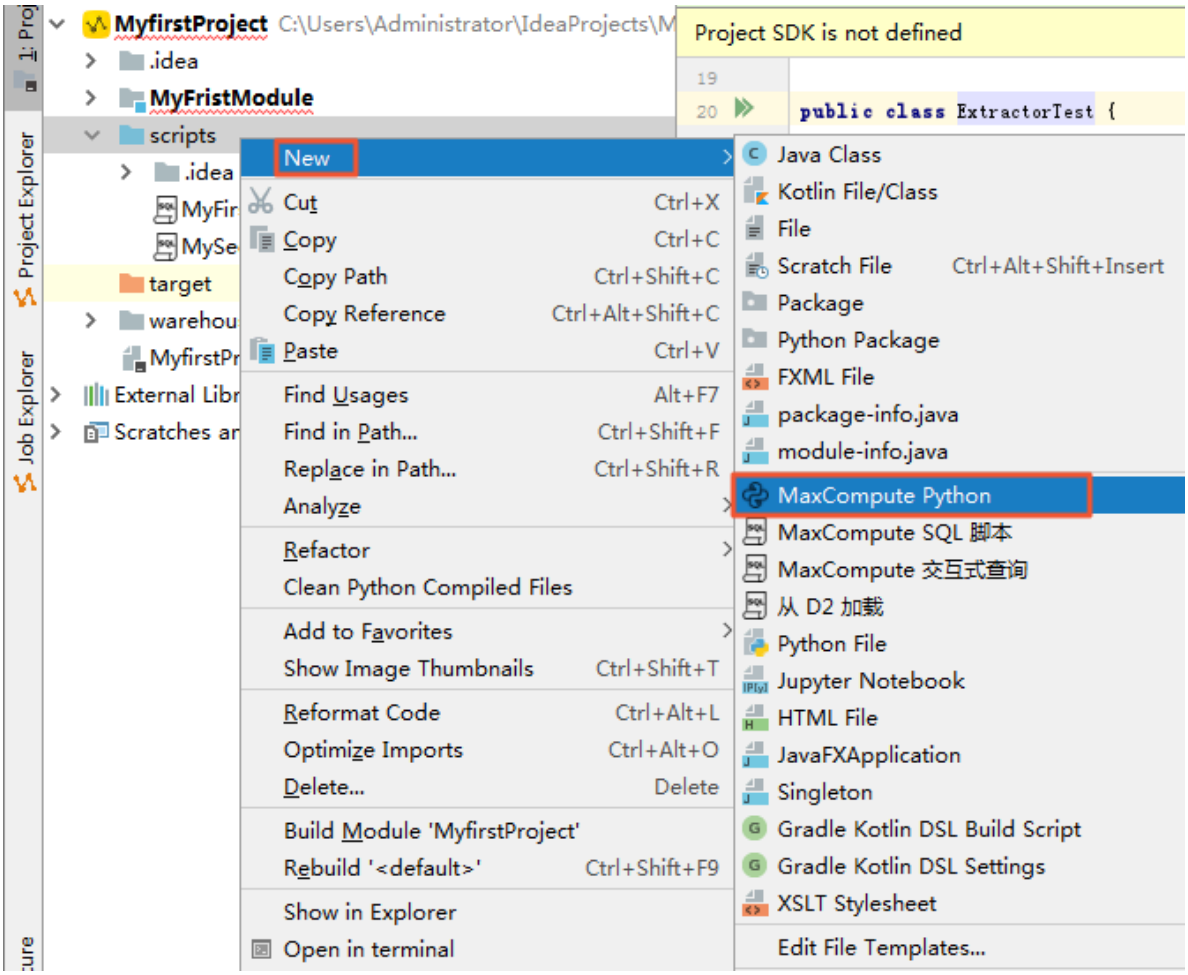
2.7.2 开发Python UDF

MaxCompute Studio支持Python UDF开发，但需要您做好准备工作。

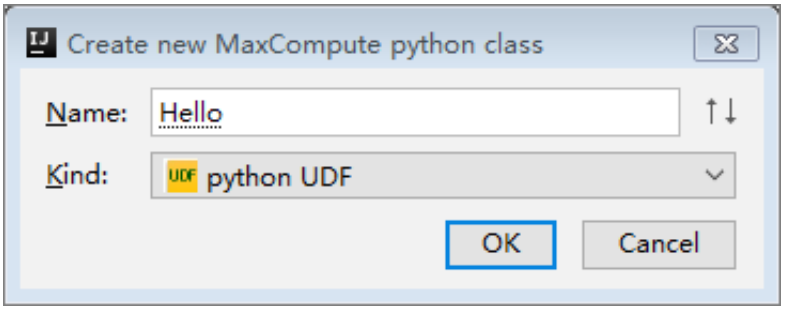
开发

1. 右键单击New > MaxCompute Python。

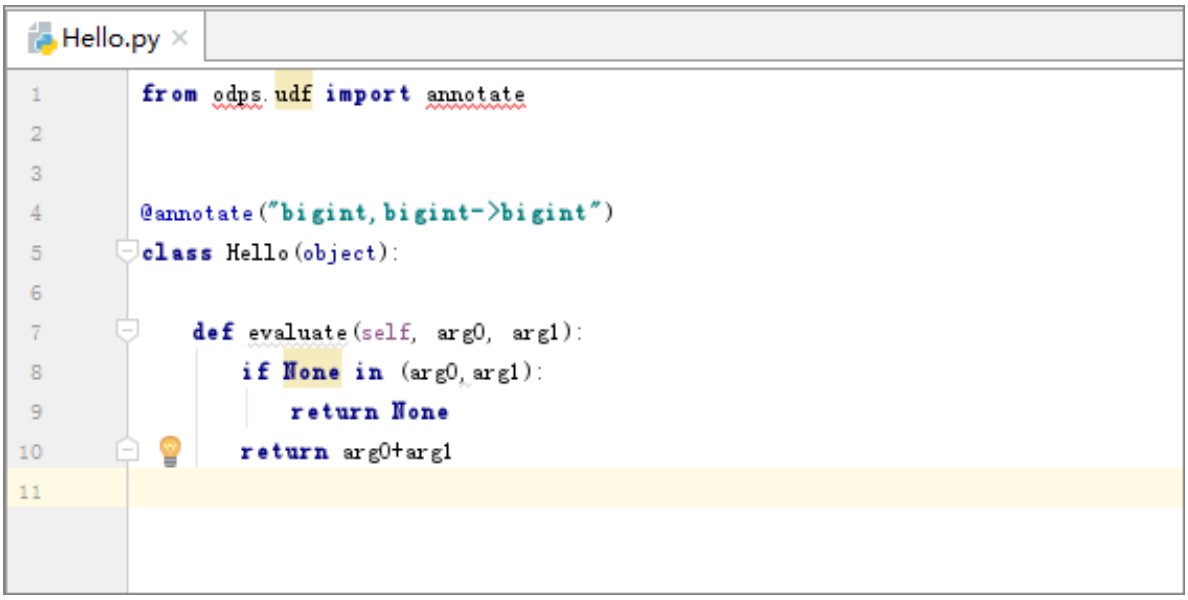
> **说明:**
> 如果没有MaxCompute Python选项说明没有Python插件，请确认是否安装成功。配置和安装过程请参见*Python*开发使用须知。



2. 输入类名，如Hello，选择类型，此处选择Python UDF。填写完成后单击OK。



3. 模板已自动填充框架代码，您只需要编写UDF的入参出参，以及函数逻辑。



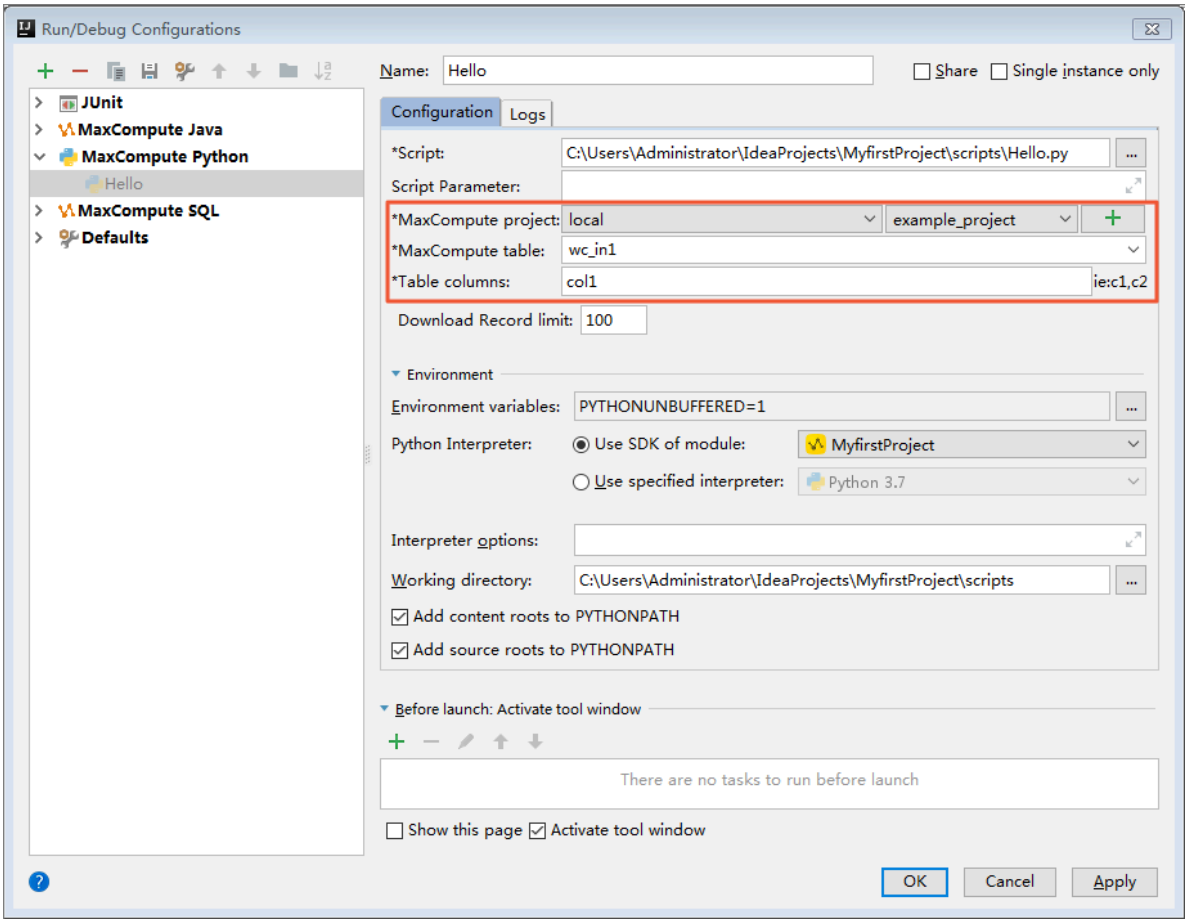
测试

UDF开发完成后，需要测试自己的代码，看是否符合预期。我们支持下载表的部分sample数据到本地运行，进行DeBug，操作如下：

1. 右键单击Editor中的UDF类，单击RUN，弹出Run/Debug Configurations对话框。

说明:

UDF|UDAF|UDTF一般作用于Select子句中表的某些列，此处需配置MaxCompute project、table和columns，元数据来源于project explorer窗口和warehouse下的Mock项目。

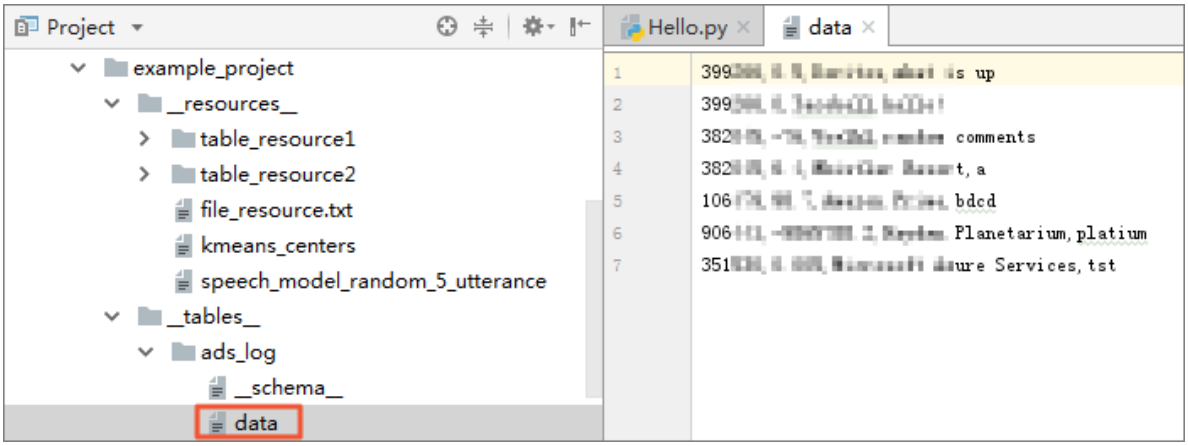


2. 单击OK后，通过Tunnel自动下载您指定表的sample数据到本地warehouse目录。

说明:

- 如果已经下载过，则不会再次重复下载，否则利用Tunnel服务下载数据。
- 默认下载100条，如果需要更多数据测试，请自行使用console的Tunnel命令或者Studio的表下载功能。

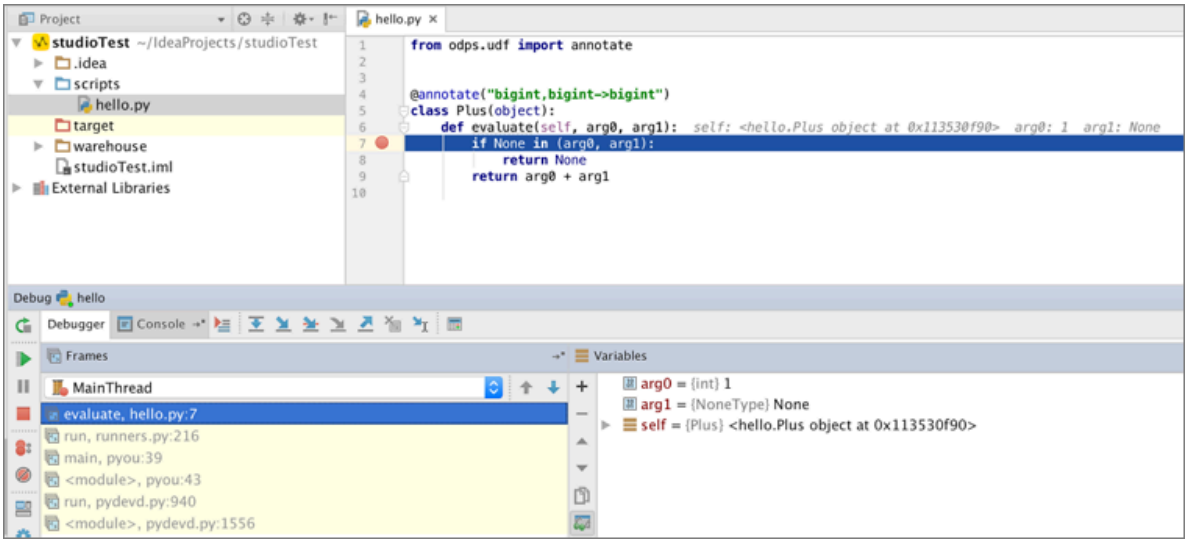
3. 下载完成后，您可以在warehouse目录看到下载的sample数据。您也可以使用Mock data（即warehouse中的数据自己mock，详情请参见开发和调试UDF中的#####warehouse##模块。



4. 本地运行框架会根据您指定的列，获取data文件中指定列的数据，调用UDF本地运行。

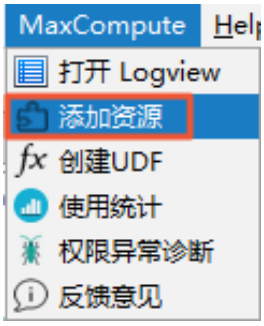
> 说明：
> 本地运行是通过Pyodps的pyou脚本实现的，命令如`pyou hello.Plus<data`。安装完pyodps后可以使用相应的命令检查该脚本是否存在。
>
> - 如果您是windows系统，请运行`${python}/../Scripts/pyou`。
> - 如果您是mac系统，请运行`${python}/../pyou`。

您可以在控制台看到结果打印，也可以在UDF上打断点调试。



注册发布

Python UDF测试通过后，即可注册发布到生产上进行使用。Add Resource后，Create Function即可。详情请参见打包、上传和注册。

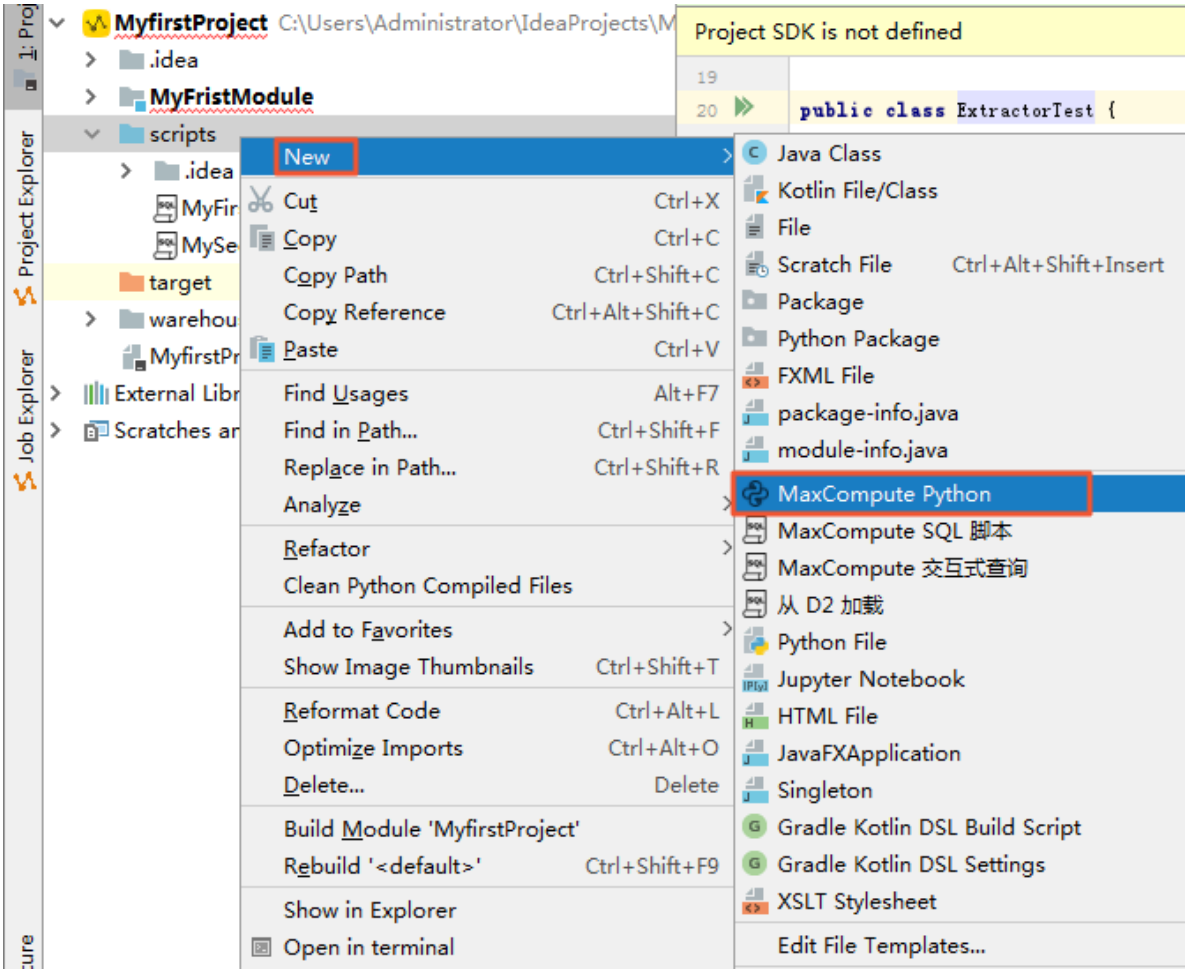

MaxCompute
Help
打开 Logview
添加资源
创建UDF
使用统计
权限异常诊断
反馈意见

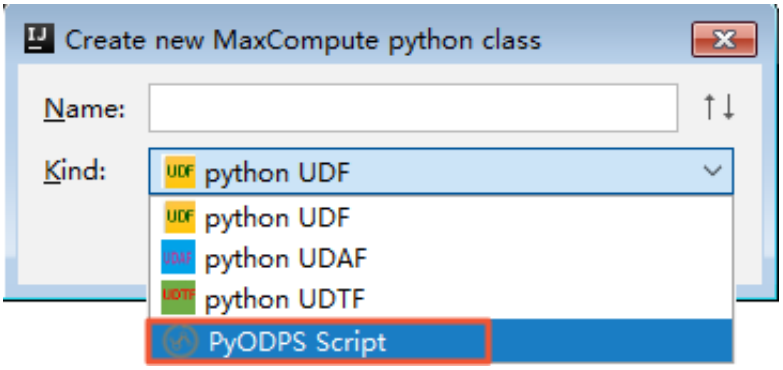
2.7.3 开发PyODPS脚本

PyODPS是MaxCompute Python版本的SDK， 它提供了对MaxCompute对象的基本操作，并提供了DataFrame框架，您可以在MaxCompute上进行数据分析。

开发PyODPS脚本

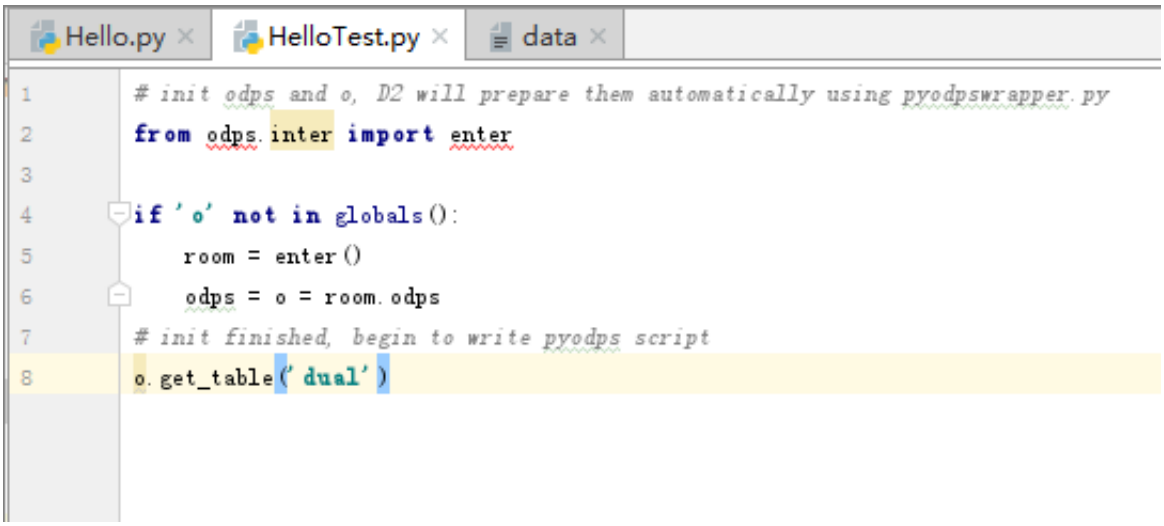
1. 使用PyODPS开发脚本，首先需要安装PyODPS及Python插件，安装完成后即可直接新建一个MaxCompute PyODPS脚本。





2. 新建成功后，模板会通过PyODPS *room对象*，自动初始化odps和o这两个对象。

> **说明:**
> 在公共云Dataworks上运行时会自动在后台创建，所以为了IDEA编译通过，需要显式创建。



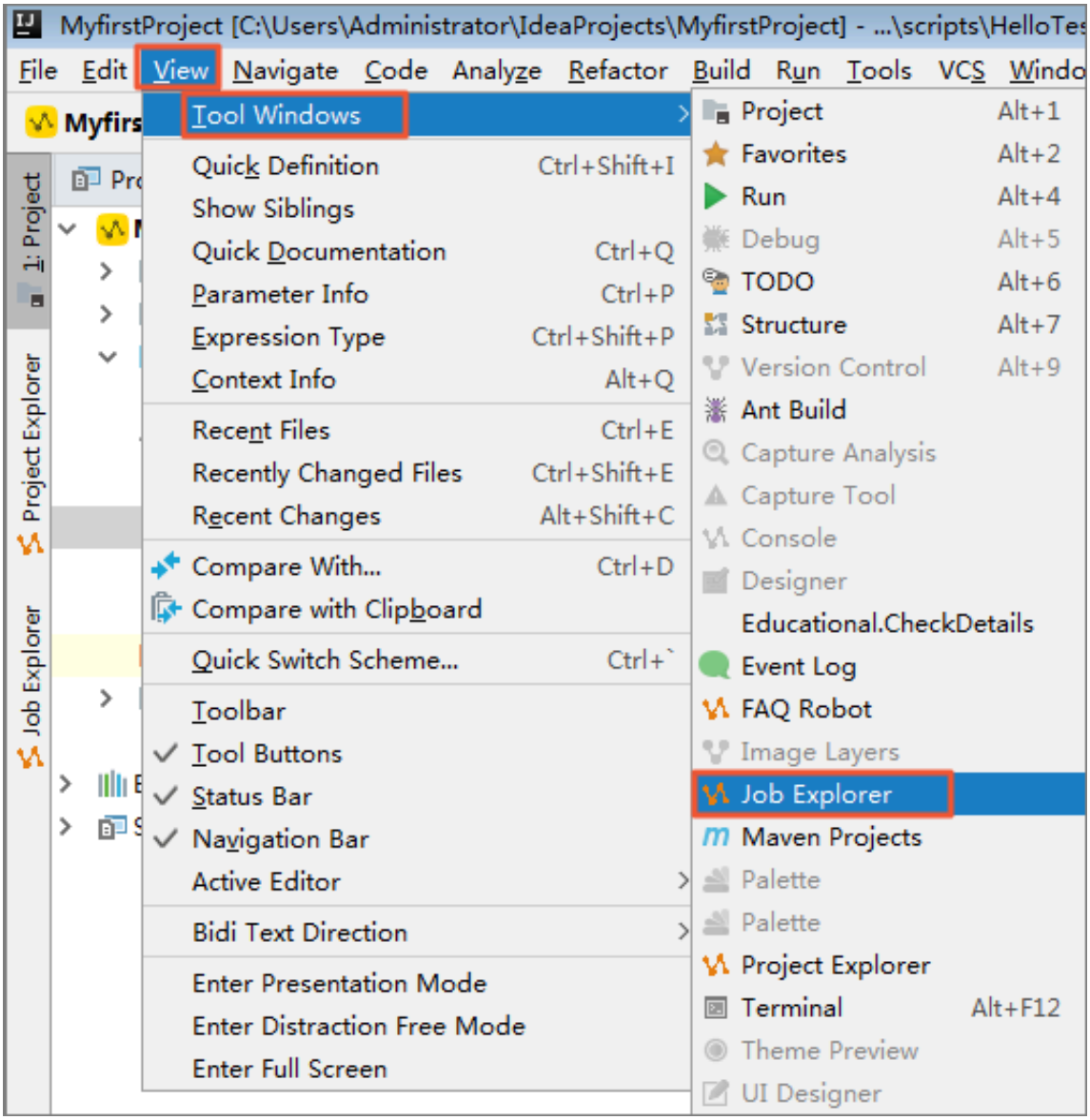
2.8 管理MaxCompute作业

2.8.1 作业浏览

通过Studio可以方便查看当前用户提交的MaxCompute的运行实例，包括运行状态、作业类型、起止时间等。

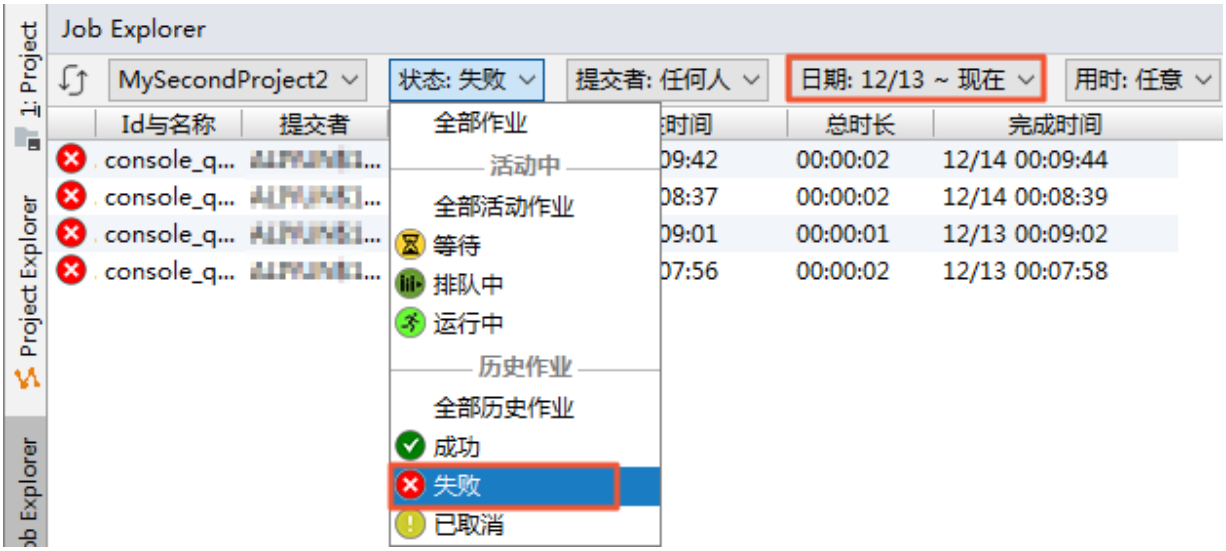
打开Job Explorer

如果Job Explorer View没有在左侧Dock上显示，可以通过View > Tool Windows > Job Explorer 打开。

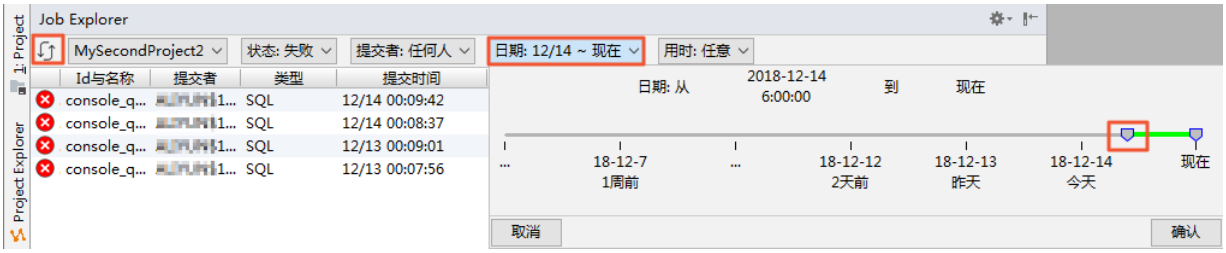


查看项目下所有作业实例

Job Explorer支持按照状态查询提交的作业列表，如下图所示，查看过去24小时内执行失败的作业：



单击日期筛选框，选择其他日期，拖动选择时间。如下图。

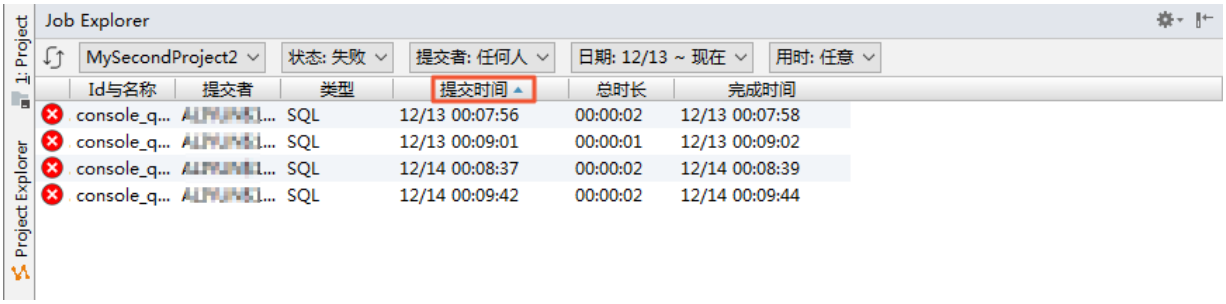


点击刷新按钮获取作业列表。

> **说明:**
> 默认只显示符合条件的前1000条作业，若超过1000条的作业将无法显示，请考虑更新过滤条件。

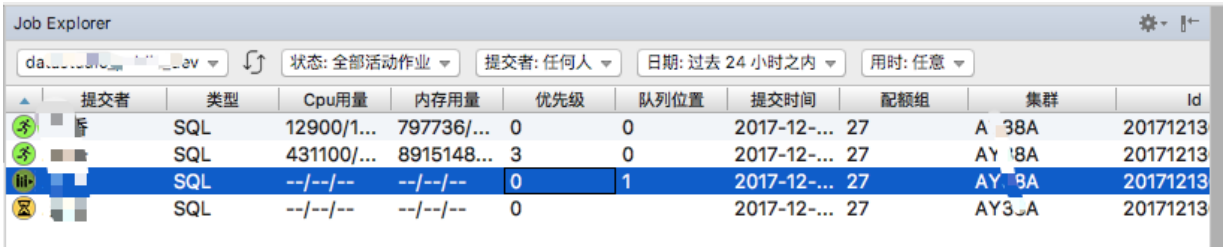
作业列表排序

可以通过点击job列表的列名对job进行简单排序，如下图按提交时间倒序：



作业排队队列

活动状态作业如果正在排队队列中等待调度，作业列表中会展示当前排队的位置和全局优先级：



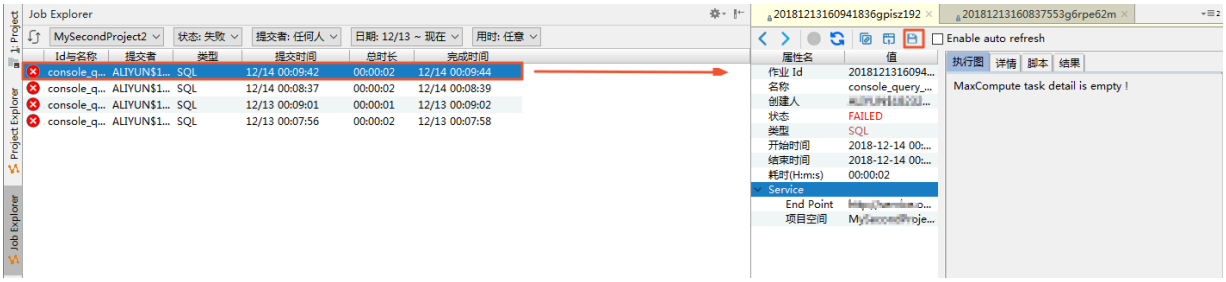
说明:

在Running Instances Tab下的作业状态、队列位置等会自动更新，作业结束后会从列表中移除。

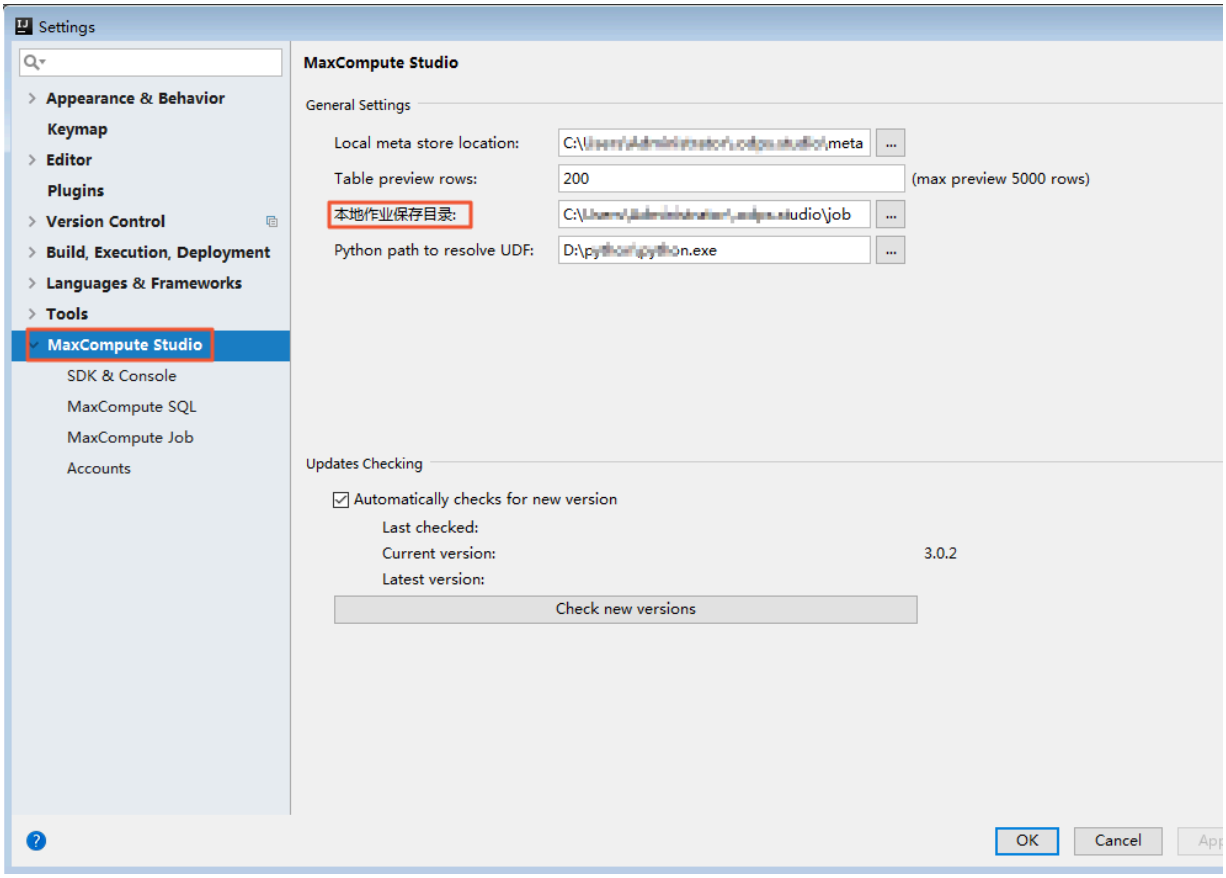
作业日志保存功能

目前job的logview日志默认保存7天，有些重要的logview希望能保留更长时间以便后期进行查看，可以将logview保存到本地：

双击列表中的job，在右侧打开job的详细信息，点击工具栏上的save按钮，将日志保存到本地：



若需要自定义保存文件的目录，可以在Studio的Setting配置项中进行配置：



2.8.2 作业实例详情

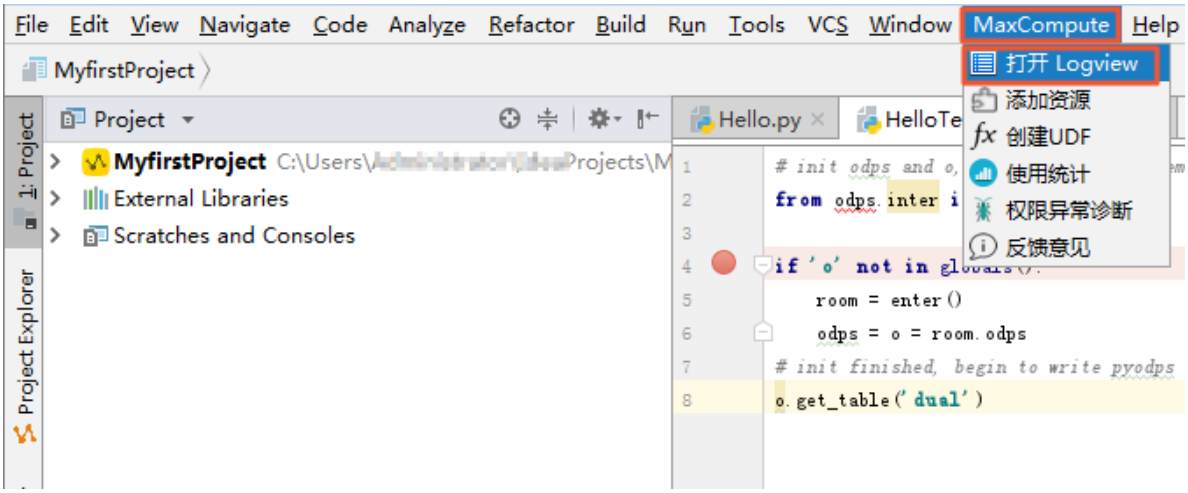
查看作业实例

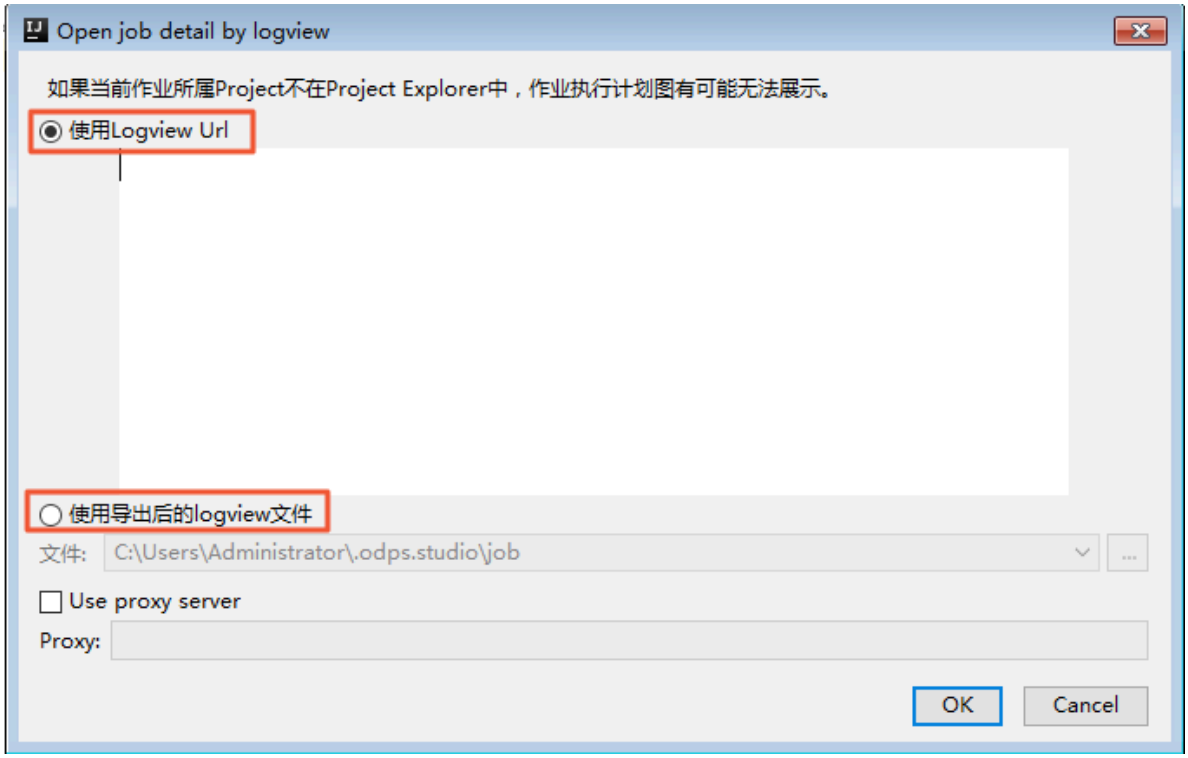
Studio支持两种方式查看MaxCompute作业实例。

1. 通过Logview URL或本地的logview离线文件以只读方式打开作业详情。

   使用Logview来查看一个作业的详细信息是MaxCompute用户熟悉的方式。使用Logview还有一个便利之处是可以查看其他用户在其他项目空间中提交的任务状态。Studio提供通过输入一个有效的Logview URL打开任意一个作业详情的功能。

   在菜单栏中，打开MaxCompute菜单下的Open Logview，即可自动将粘贴板中有效Logview URL地址复制到弹出窗口中或者选择导出在本地的logview离线文件。



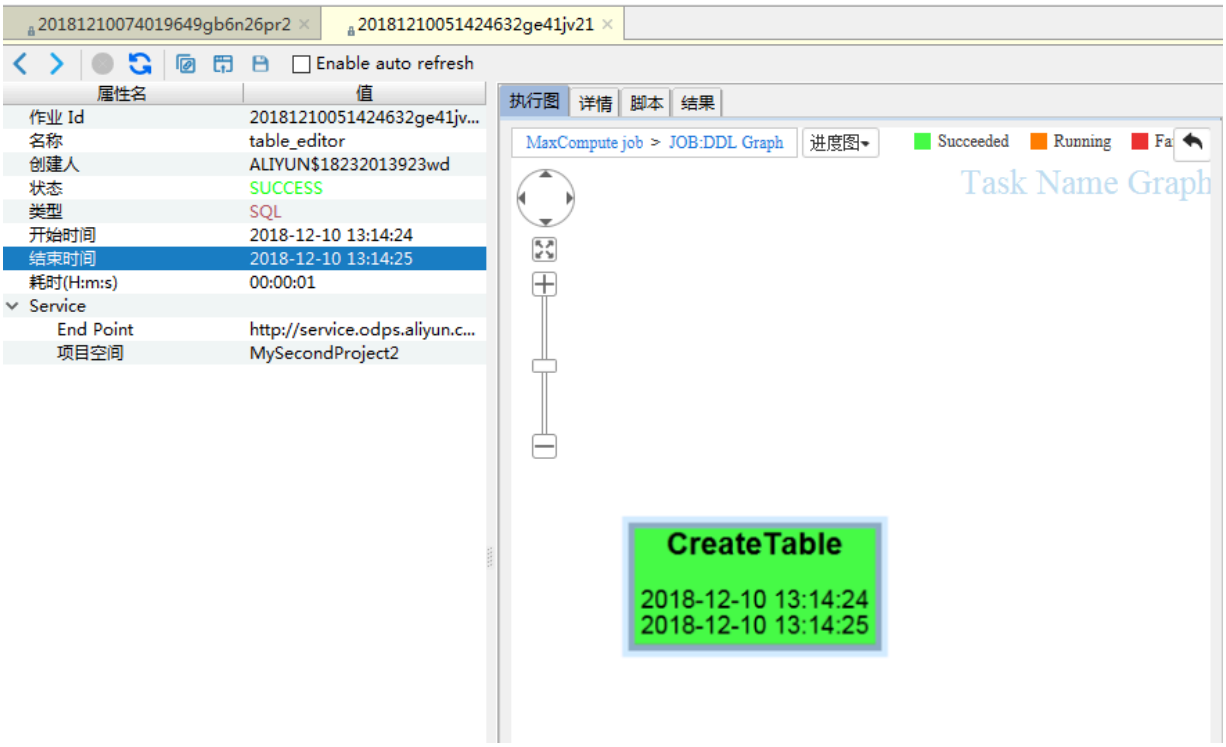


2. 作业浏览信息中，选择某个MaxCompute实例，双击或右键Open，可查看该实例详细信息。

作业详情视图

作业详情页面包括顶部的工具栏，左半部分的基本属性栏以及右半部分详细视图页，其中详细视图页主要包含四个视图：

- 执行图：以DAG图的方式显示作业整体信息，可查看子任务间依赖关系及各个子任务的详细执行计划。
- 时序图：显示作业执行timeline，可以从不同粒度查看执行的时序,并提供了多种过滤器。
- 详情：以table view方式展示作业详细信息，包括子任务列表，各子任务的worker列表，worker处理数据量，执行时间及状态信息等。
- 脚本：显示该作业提交时所对应的SQL语句以及提交作业的参数配置信息。
- 概要（JSON）：以JSON格式显示作业运行详细信息。
- 结果视图：显示该作业运行结果。
- 分析：提供作业执行散点图、长尾柱状图及数据倾斜图。

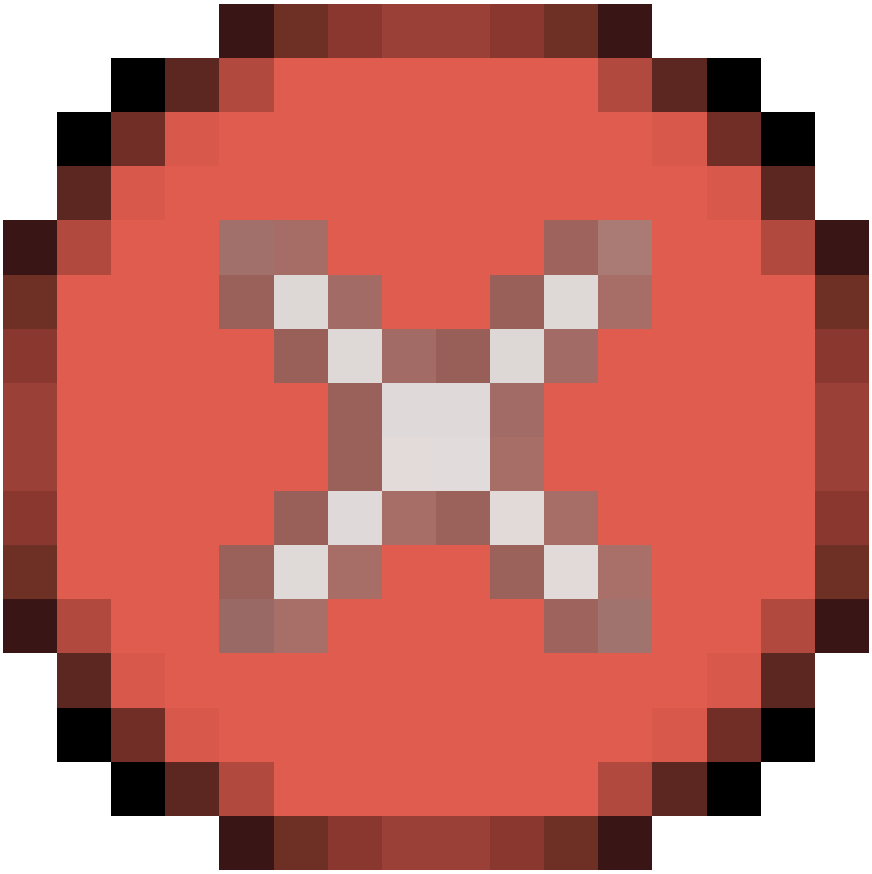


工具栏

页面左右折叠

用来收起或完全展开左右侧视图，允许用户最大化某一个视图进行查看。

停止作业



用来中断正在执行的作业，需要具有响应权限才能停掉作业（owner或管理员）。

刷新详情

对于运行中的作业，它的基本信息，如状态、quota等会自动刷新，但右侧各个详情视图不会自动刷新，如果想要获取最新详情信息，需要手动刷新。

拷贝logview

复制作业对应的logview到剪切板。

在浏览器打开作业详情

生成logview URL，并通过浏览器打开。

作业详情保存到本地

将作业详情信息保存为本地文件。

自动刷新

对于运行中作业，允许自动刷新后，studio会对作业执行全量定时刷新。

基本信息页

展示作业基本信息，包括ID、Owner、状态、起止时间、计算资源用量, 输入输出等。运行中作业的基本信息会自动定时刷新。输入，输出项列出了作业的输入表和输出表，双击表名可查看对应表详情。

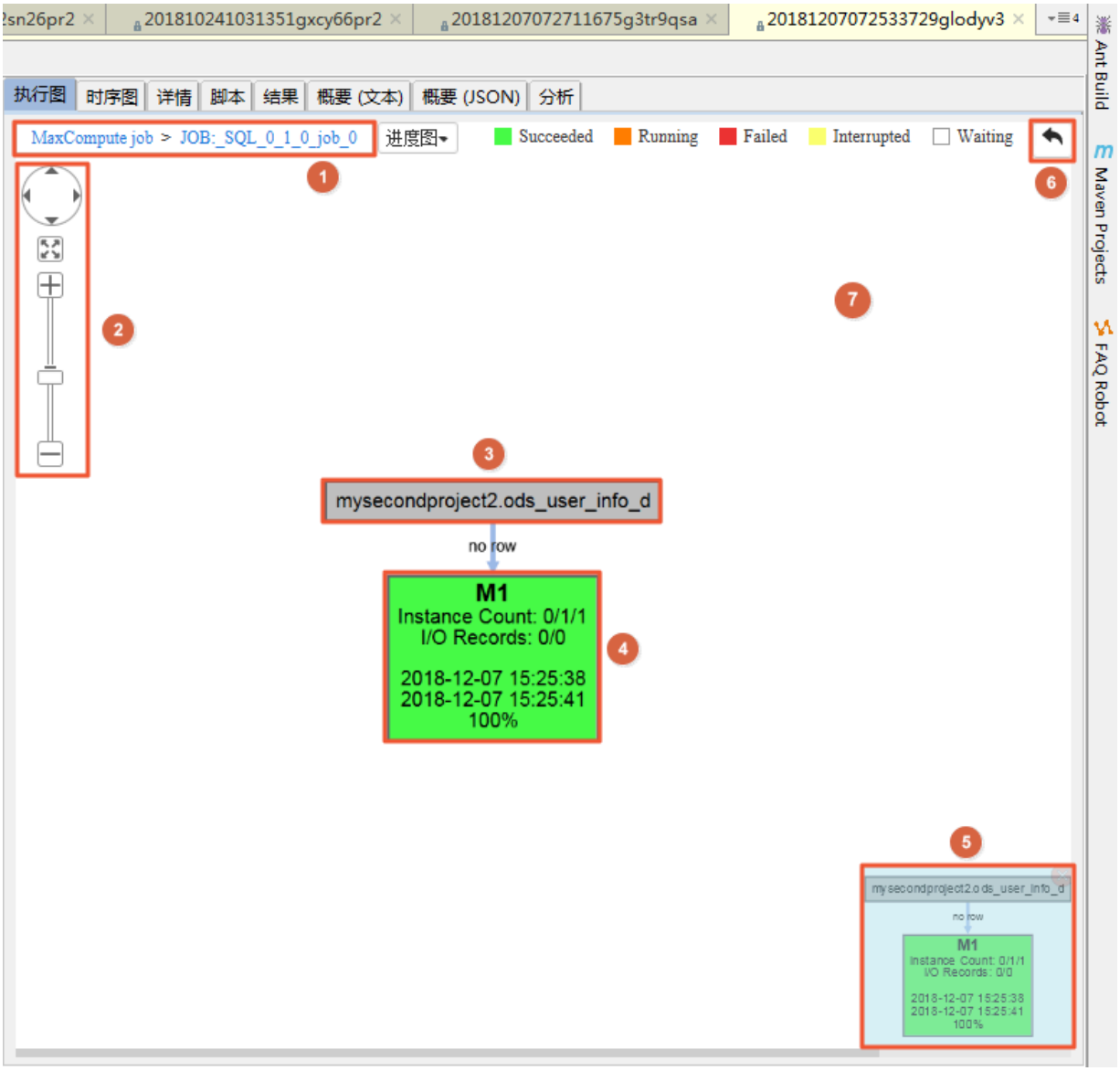
执行图

执行图作为日常主要使用工具，以可视化的方式展示Fuxi Job， FuxiTask以及Operation的依赖关系，同时提供一系列辅助工具，如作业回放，进度图，热度图等，绝对是排查问题的好帮手。

执行图可展示三个维度的作业依赖关系：Fuxi Job层， Fuxi Task层， Operation层，可点击面包屑或向上箭头进行切换，默认会展示Fuxi Task层依赖关系。

说明:

非SQL类型作业，仅能展示Fuxi Job和Fuxi Task层作业，不支持Operation层。



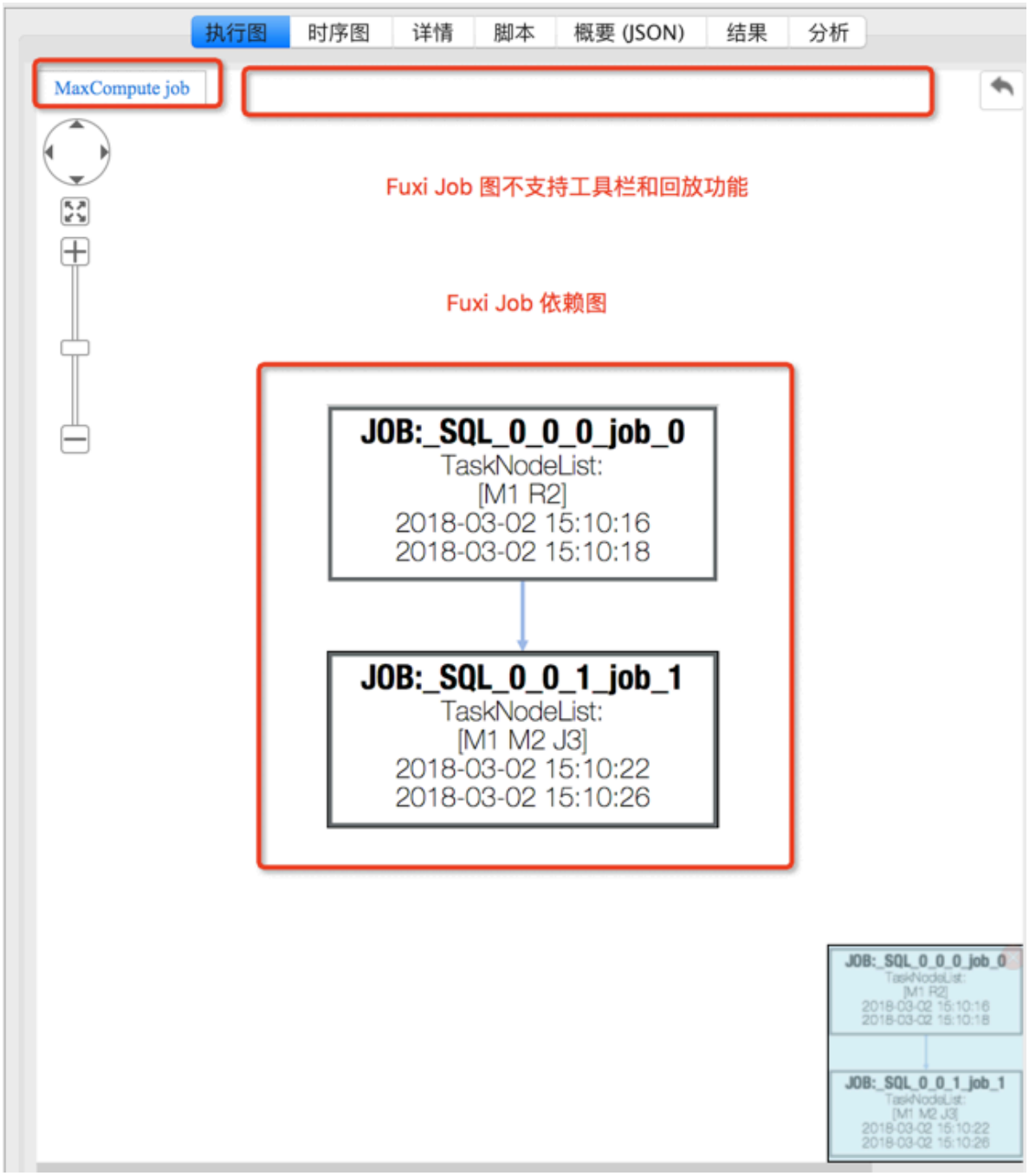
说明:

对应序号说明如下:

1. **可点击跳转其他层次**
2. **缩放辅助工具**
3. **依赖表**
4. **Fuxi Task节点**
5. **鹰眼**
6. **展示**
7. **默认打开FuxiTask层依赖**

- Fuxi Job层:

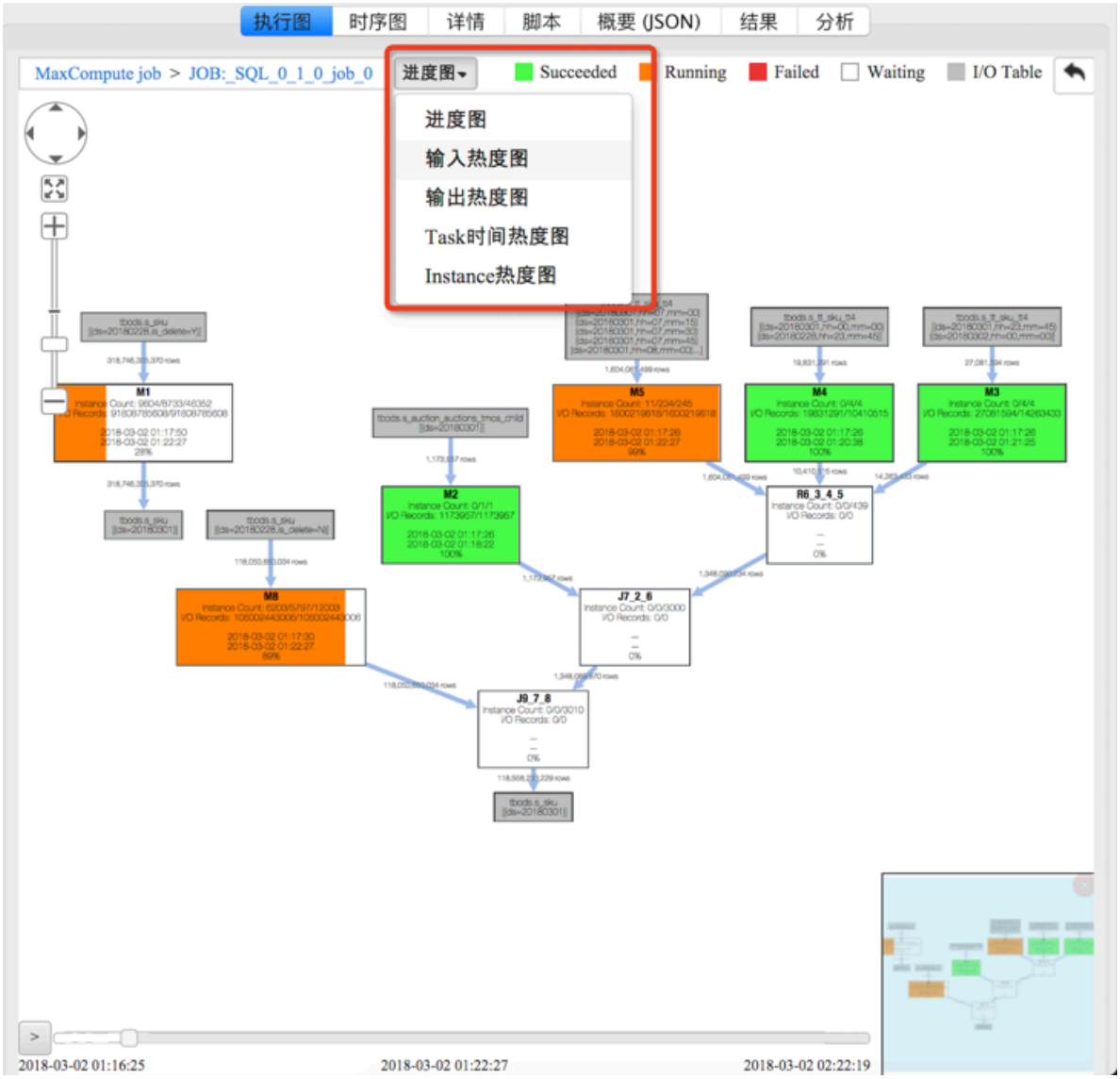
  单击MaxCompute Job或在Fuxi Task层中点击向上箭头即可打开Fuxi Job层。Fuxi Job节点内包含Fuxi Task名称，起止时间等。双击任一Fuxi Job节点即可进入Fuxi Task层。



- Fuxi Task层:

  当有多个Fuxi Job时，默认打开最后一个Fuxi Job的Fuxi Task层。该层可展示Fuxi Task的依赖关系，输入输出表及分区等信息。当作业结束后点击工具栏中下拉框可切换进度图，输入热度图，输出热度图，Task时间热度图，Instance热度图等。进度图表示节点的完成进度，热度图

通过颜色区分节点热度。双击任一Fuxi Task即可打开Operation层，右键可以展开所有Fuxi Task的Operation层。



Fuxi Task节点内容：

1.Instance Count：a/b/c，指某一时刻正在运行子任务实例个数为a，已结束任务实例个数b，总任务实例个数c。

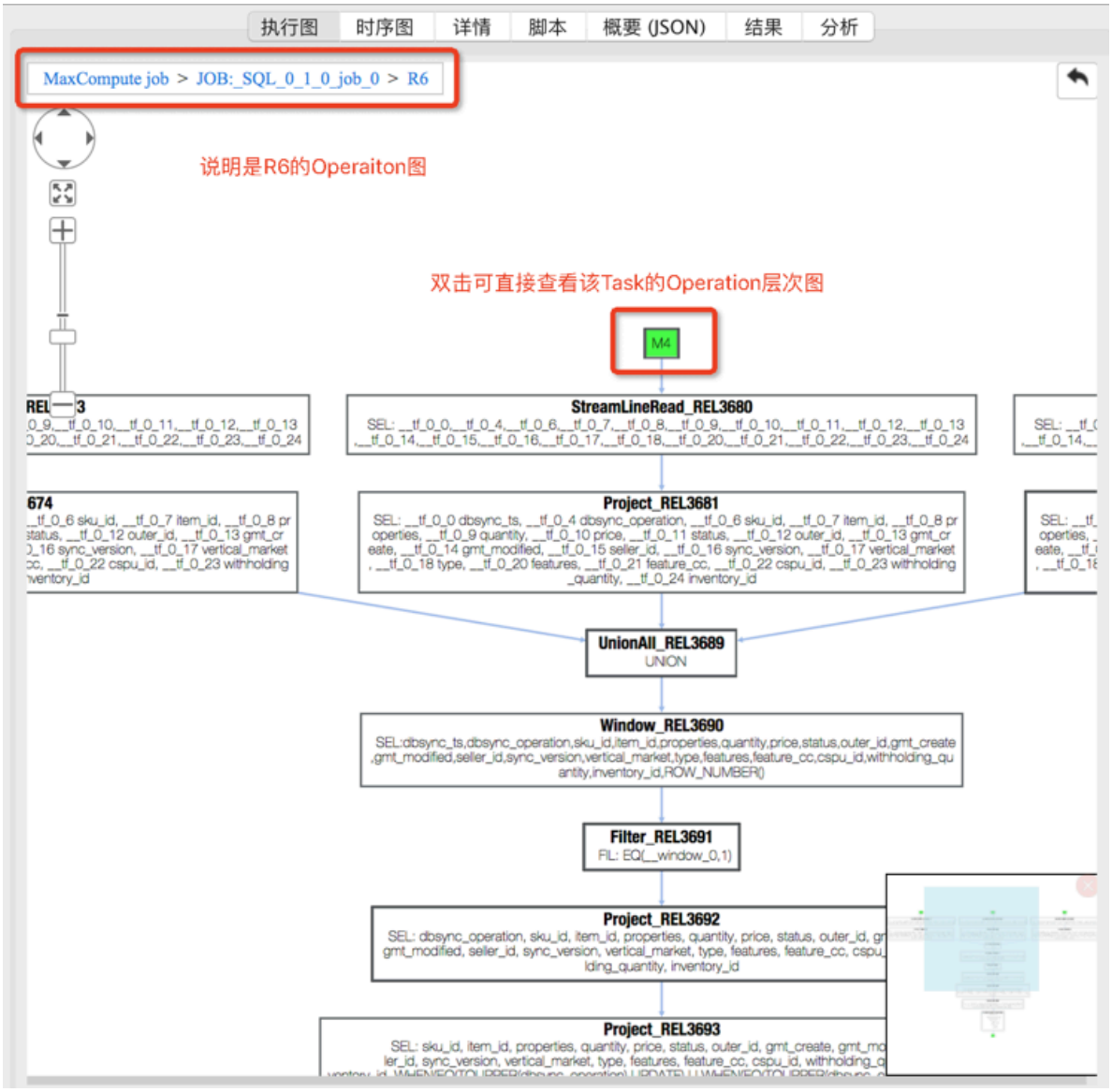
2.I/O records：同理为某一时刻的input records和output records。

3.百分比与橙色进度条：表示该任务运行情况，该比例根据子任务运行实例分析得出。

4.子任务间连线上显示的是输出records数量。箭头表示数据流动方向。

- Operation层

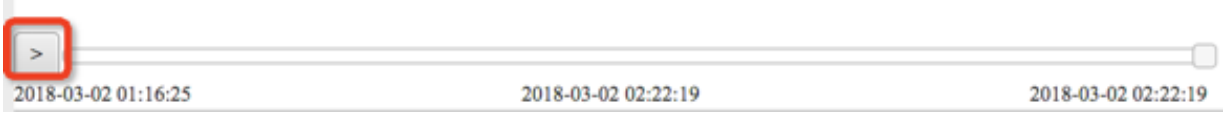
Operation层揭示了Fuxi Task内在的运行方式，单击任一节点即可显示Operaiton完整信息。



作业回放

Studio支持作业回放功能，作业回放就像播放媒体文件一样，可在12s内回顾该Job执行的历史轨迹。该功能主要用于帮助用户了解MaxCompute实例在不同时刻运行状态，快速判断子任务级运行顺序及消耗时间，掌握Job执行关键路径，从而针对运行较慢的子任务进行优化。

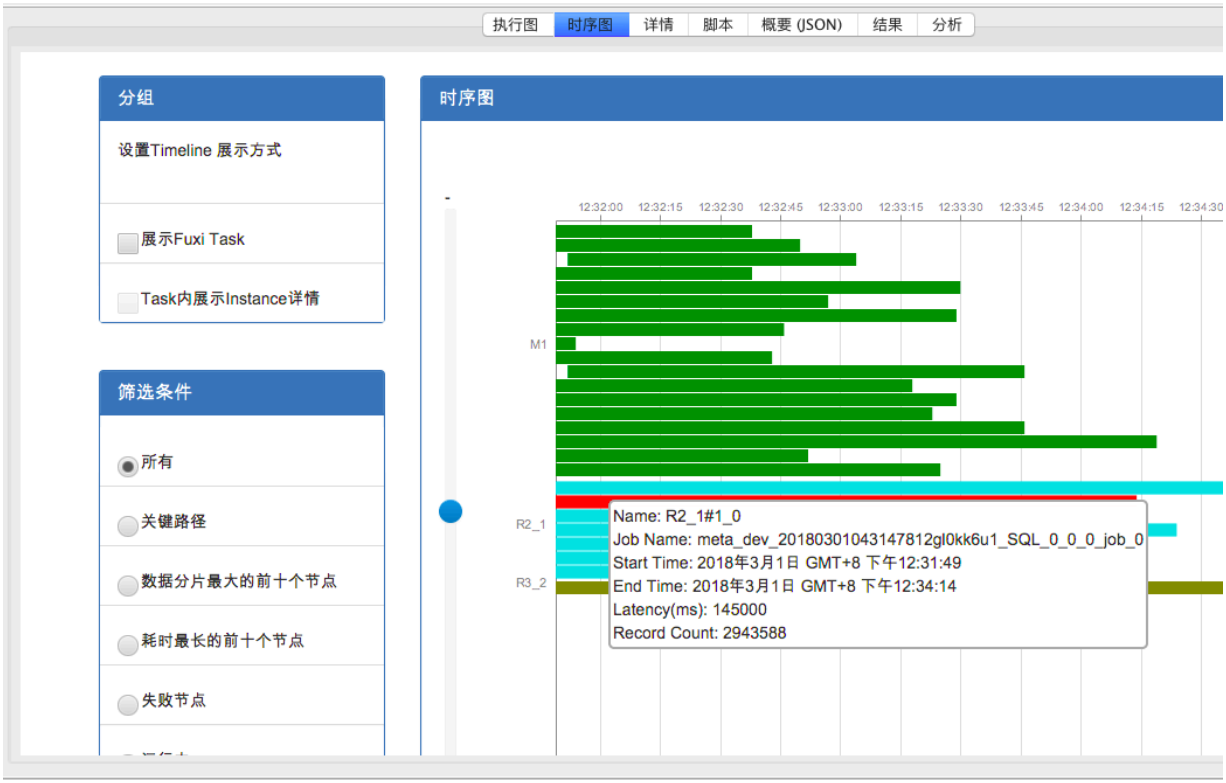
- 点击 > 按钮即可开始播放，再次点击则暂停。用户也可手动拖动进度条。
- 进度条左边为作业开始时间，中间为播放时间，右边为结束时间。



> **说明:**
> 回放功能仅通过时间估算某一个时刻IO数据量，从而确定完成进度，并不能代表该时刻真实IO数据量。Running 状态作业不支持回放功能。

时序图

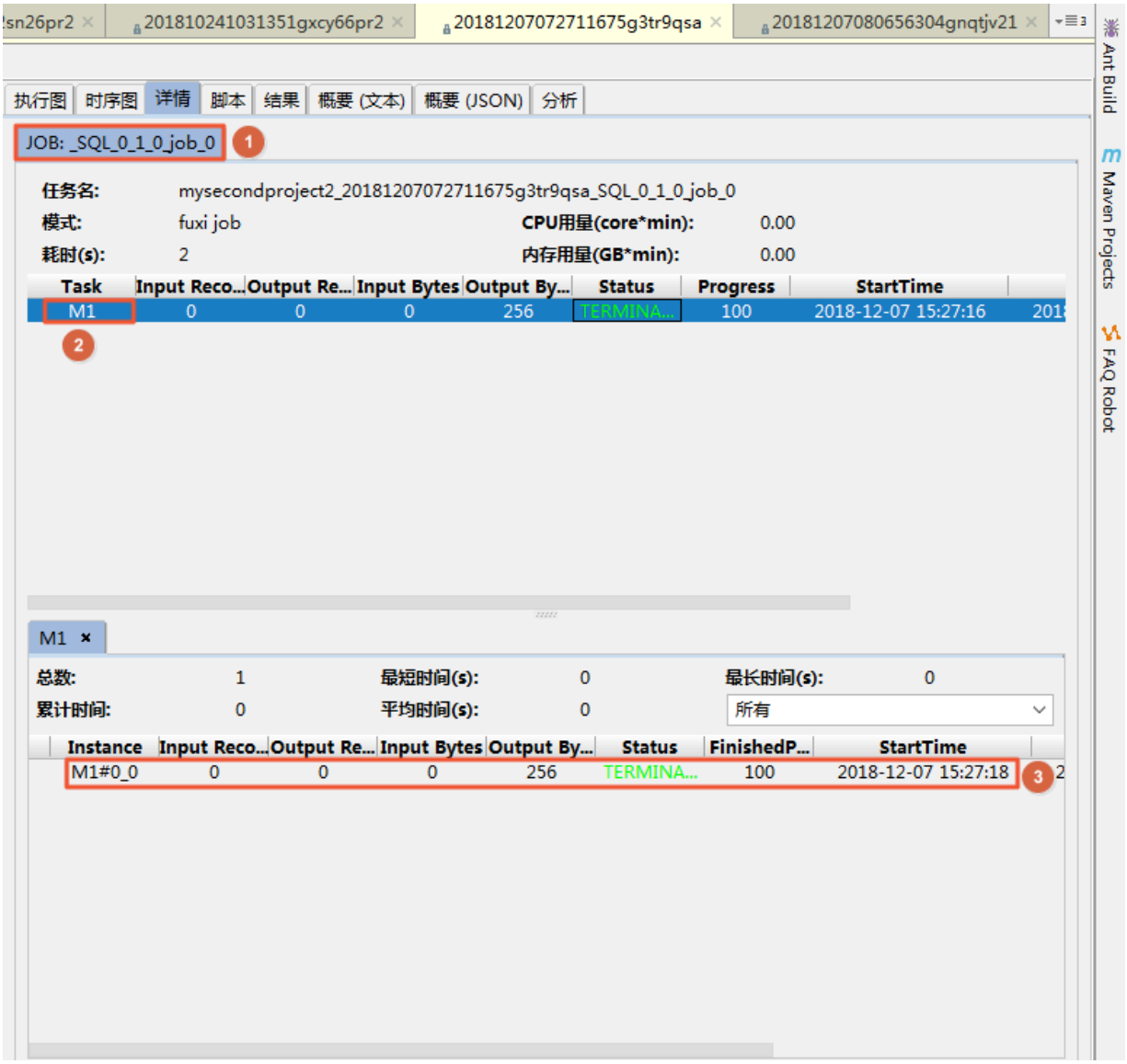
以甘特图的方式展示作业分布式执行的详细数据，可以调整展示粒度，将每一个计算节点都在甘特图中展示。



可以通过甘特图直观的看出作业运行的时间瓶颈、长尾节点等。同时提供多种过滤器，能够直接筛选出作业执行的关键路径、最大数据节点、最长时间节点等。

详情页

主要针对SQL DML类作业，展示作业在计算集群上的Fuxi Task列表、计算节点列表等



说明:

对应序号说明如下:

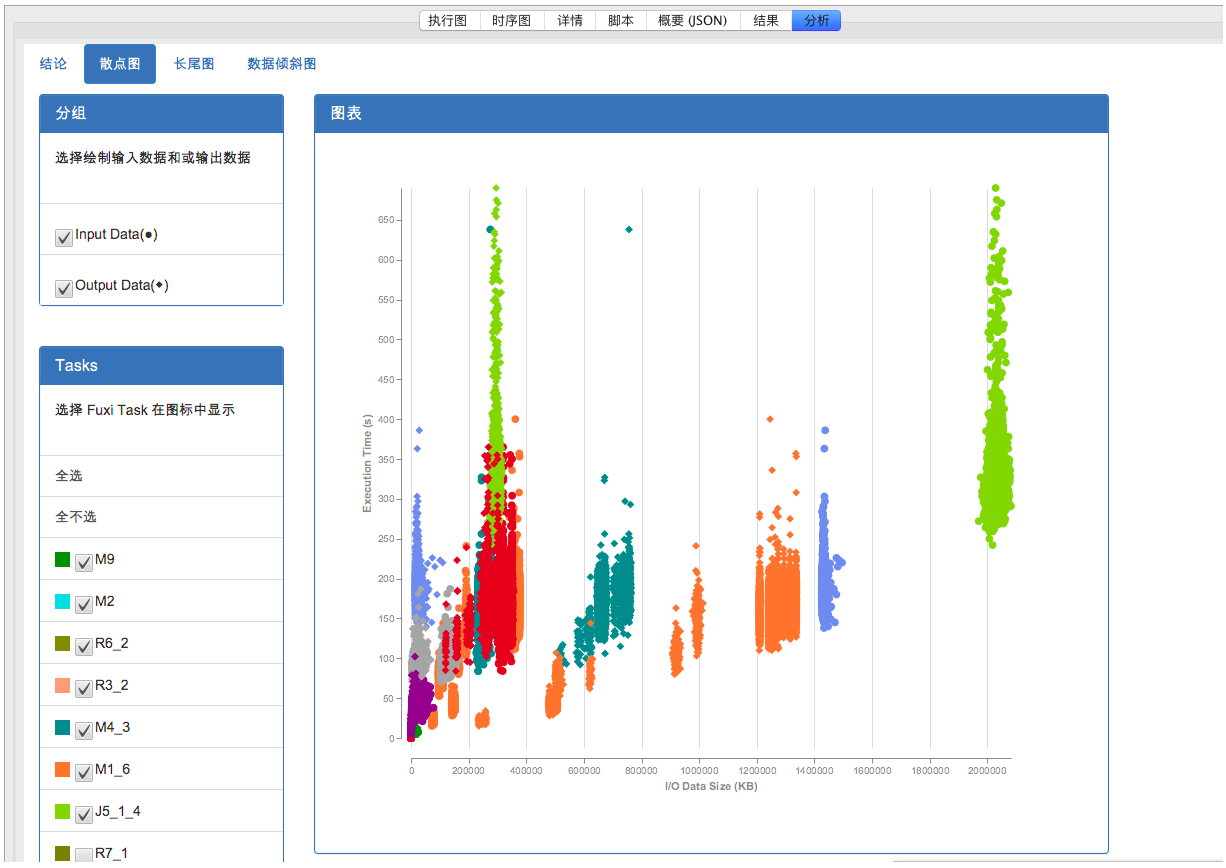
1. **Fuxi Job Tab**
2. **Fuxi Task列表**
3. **每个Fuxi Task详细信息及计算节点列表**

通常一个作业对应一个或多个Fuxi job，每个Fuxi job拆分成多个Fuxi task（阶段）,每个Fuxi task包含多个Fuxi instance（worker）。

在每个Fuxi instance通过右键菜单可以查看作业运行的标准输出、标准错误，及Debug Info。

分析页

展示作业的长尾节点(worker)、数据倾斜节点(worker)。展示节点散点图、及柱状图，辅助作业执行瓶颈诊断



散点图和柱状图支持从图中节点直接调准详情页查看Fuxi instance详情。

结果页

结果页会根据作业类型及提交作业时的参数设置展示不同页面。

- select语句并且设置odps.sql.select.output.format = HumanReadable，select以文本方式展示
- select语句并且未设置output format参数，结果以table方式展示
- 对于数据输出到表的脚本，展示输出表名及到表详情的跳转链接
- 对于异常作业，结果页限制异常详情

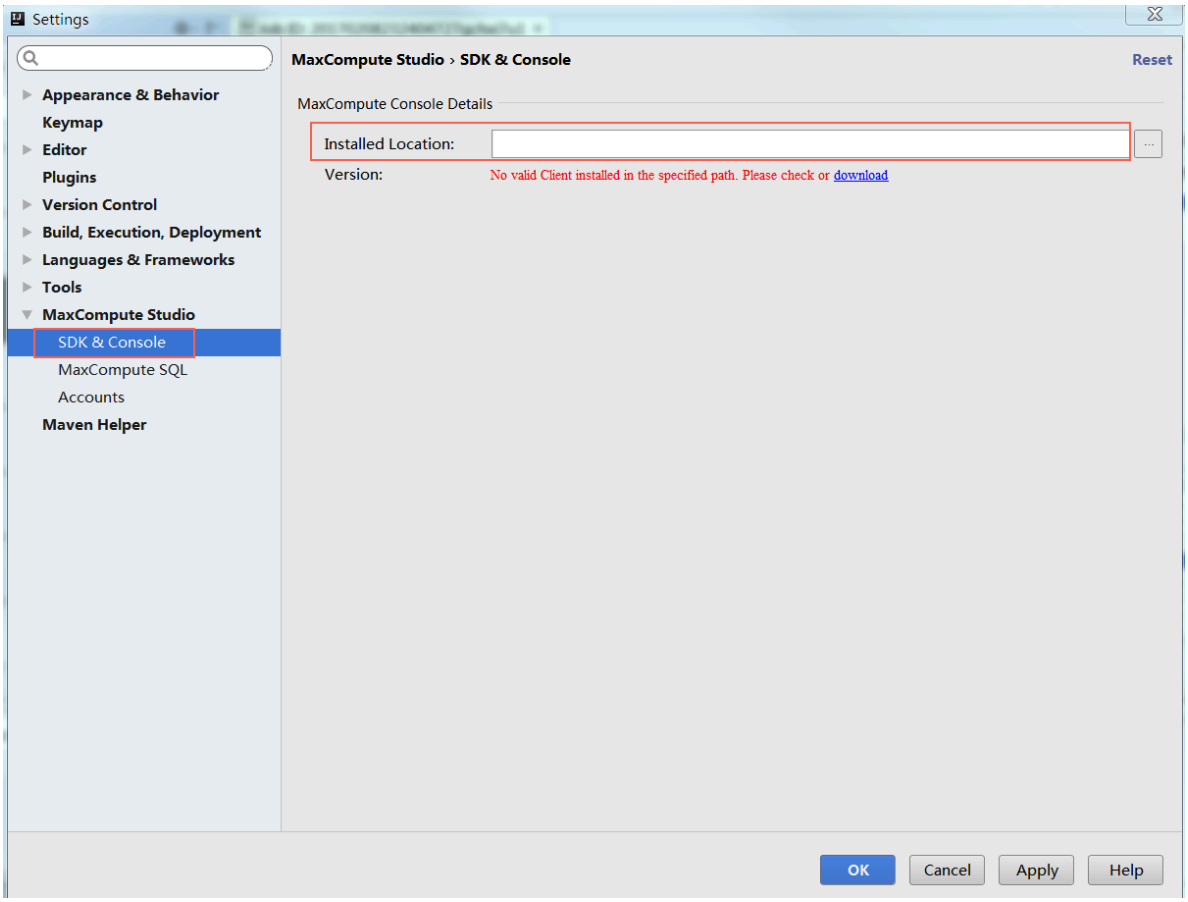
2.9 工具集成

2.9.1 与MaxCompute客户端集成

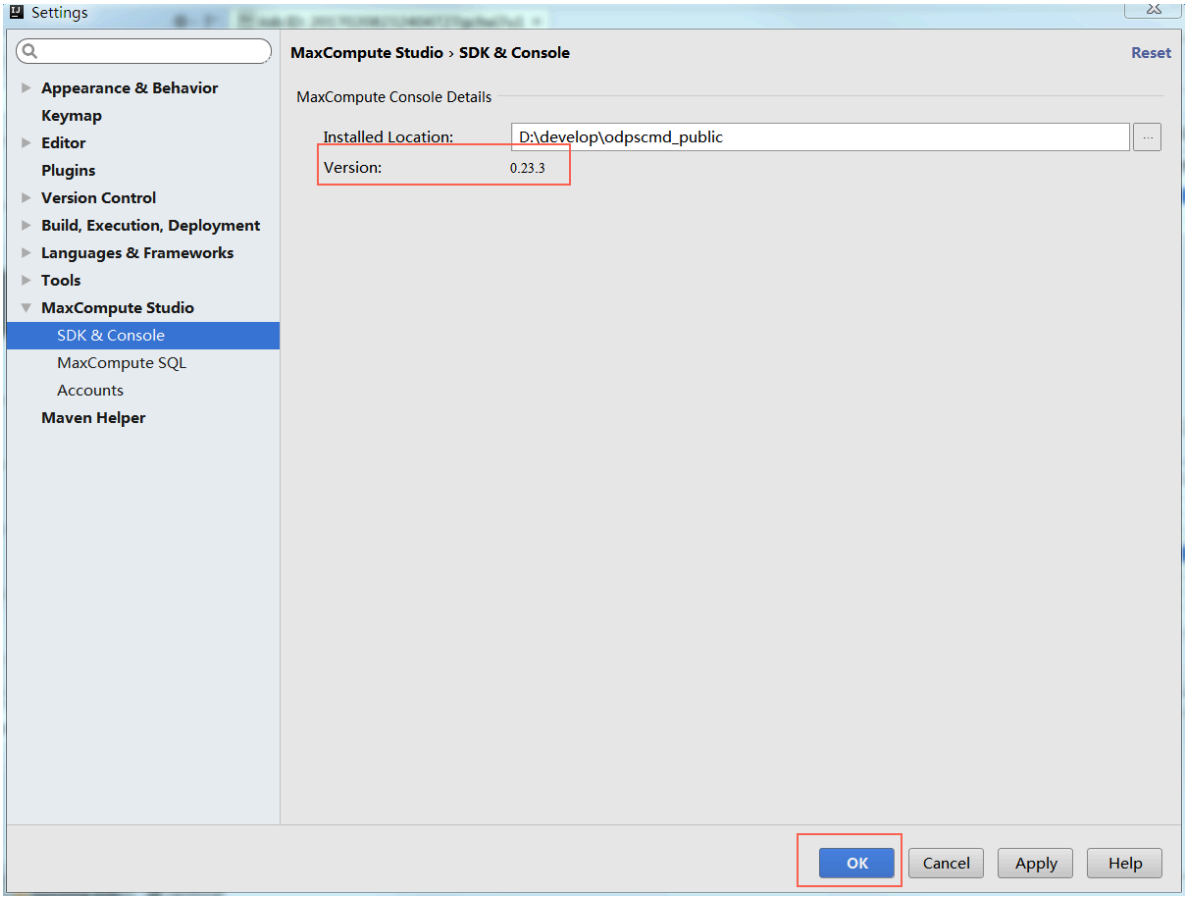
MaxCompute Studio集成了MaxCompute客户端程序，可在Studio中直接打开客户端。

配置客户端安装路径

1. Studio中已包含最新版MaxCompute客户端程序，并指定为默认客户端。用户也可自行指定其他版本客户端程序，单击file > Settings > MaxCompute Studio > SDK & Console，添加客户端程序和路径。



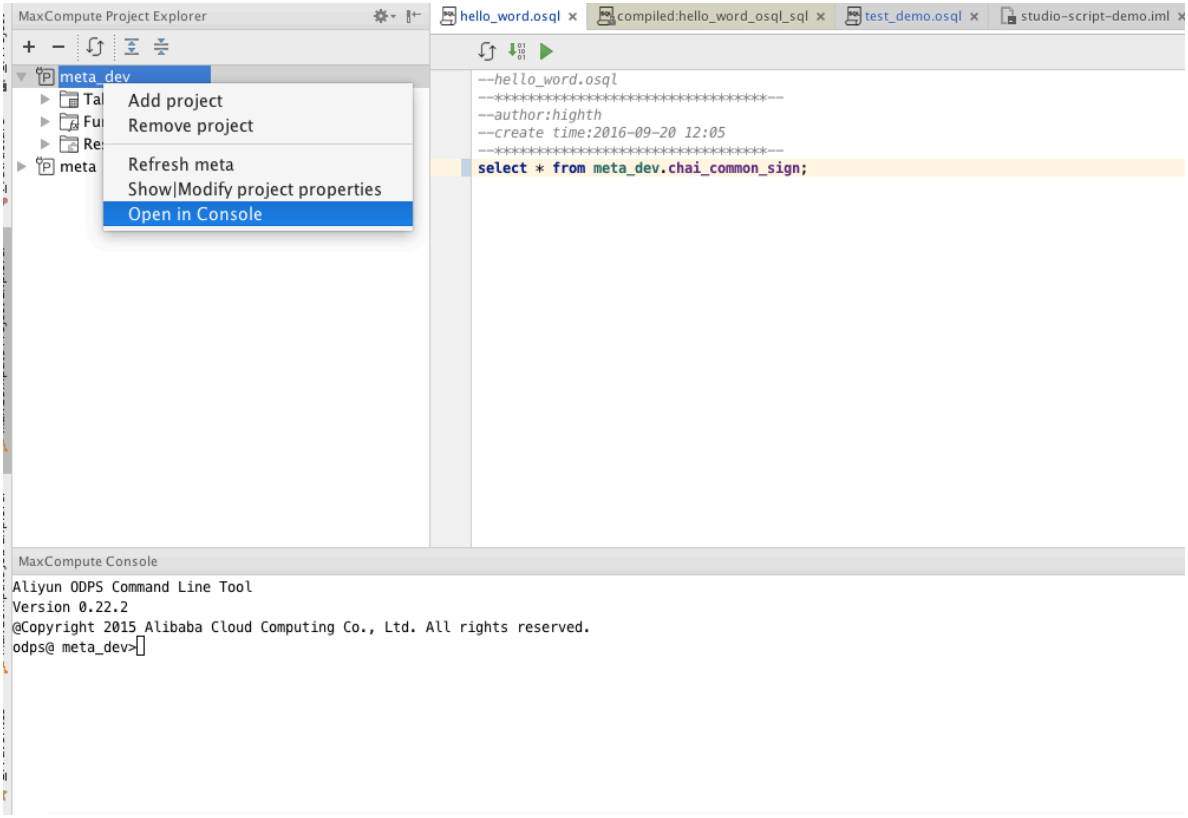
2. 设置成功后会显示MaxCompute客户端版本信息。



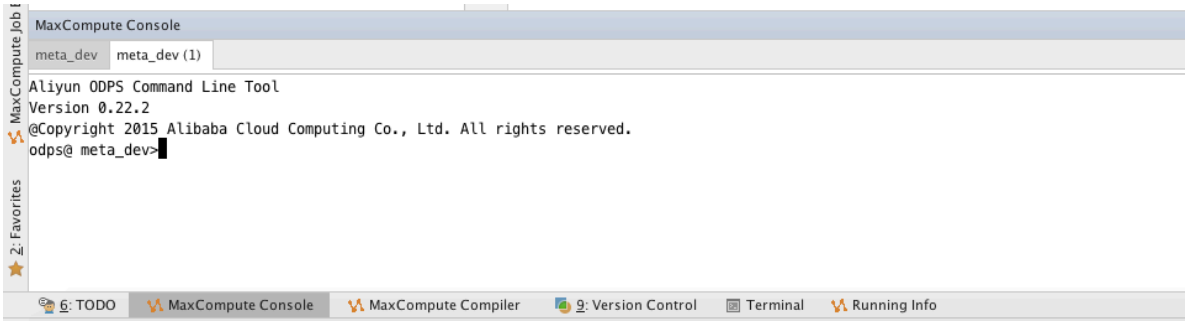
打开MaxCompute客户端

设置MaxCompute客户端安装路径成功后，就可以在Studio中打开客户端程序。

1. 在项目浏览列表中，选中要打开的项目，右键菜单中点击Open in Console。



2. 用户可以打开多个客户端程序，操作步骤同上。



2.10 配置选项

2.10.1 配置 MaxCompute Studio

安装 MaxCompute Studio 插件后，可以在 IntelliJ IDEA 的 Setting 页面的左边栏找到 MaxCompute Studio 的配置项。打开 IntelliJ 配置页面的方法请参考 IntelliJ 的文档。

MaxCompute Studio 配置选项页

MaxCompute Studio 配置选项页提供以下配置项：

1. 本地元数据仓库存储路径

   指定在本地存储 MaxCompute 项目空间元数据的路径。Studio 的缺省设置是本地用户目录下的`.odps.studio/meta`这个隐含目录。

2. 版本更新选项

   - 复选框 Automatically checks for new version 可以控制 Studio 是否自动检查新版本更新
   - 按钮 Check new versions 用于手动检查新版本。 点击检查新版本按钮后，如果有新版本可以更新，将显示 Install new version 按钮，点击可以安装，安装完成后需要重启 IntelliJ IDEA。

SDK & Console 配置选项页

SDK & Console 配置选项页面提供以下配置项：

1. MaxCompute 客户端安装路径

   指定本地安装的 MaxCompute 客户端的安装路径。 Studio 会检测路径中安装的 MaxCompute 客户端的版本，如果检测失败，会提示错误信息。

   > 说明：
   > Studio 2.6.1 之后的版本自带了最新的 MaxCompute 客户端，用户不用特别指定。 如果用户希望使用自己特定版本的 MaxCompute客户端，可以指定路径。

MaxCompute SQL 配置选项页

MaxCompute SQL 配置选项页面提供以下配置项：

1. 启动语法高亮

   勾选Enable syntax coloring 选项，启动语法高亮功能。

2. 启动代码自动补全

   勾选Enable code completion 选项，启动代码自动补全功能。

3. 启动代码格式化

   勾选Enable code formatting 选项，启动代码格式化功能。

4. 编译器选项

在这里这是全局缺省的编译器选项。 以下选项还可以在 SQL 编辑器的工具栏上为每个文件单独设置。

- 编译器模式（Compiler Mode）
  - 单句模式（Statement Mode）：在该模式下，编译器对 SQL 文件单条语句作为单元进行编译、提交
  - 脚本模式（Script Mode）：在该模式下，编译器对整个 SQL 文件作为单元进行编译、提交。 #注#脚本模式有利于编译器和优化器更大程度地优化执行计划#提高整体执行效率#目前处于测试阶段#
- 类型系统
  - 旧有类型系统（Legacy TypeSystem）：原有 MaxCompute 的类型系统
  - MaxCompute 类型系统（MaxCompute TypeSystem）：MaxCompute 2.0 引入的新的类型系统
  - Hive 类型系统（Hive Compatible TypeSystem）：MaxCompute 2.0 引入的 Hive 兼容模式下的类型系统
- 编译器版本
  - 默认编译器（Default Version）：默认版本的编译器，
  - 实验性编译器（Flighting Version）：实验性的版本的编译器，包含正在测试中的编译器的新特性

Account 配置选项页

Account 配置选项页面提供添加和管理用户用于访问 MaxCompute 所用账户，参考 用户认证：

Studio 需要通过用户指定的账号访问 MaxCompute 的项目空间和执行提交作业等操作，目前 Studio 支持的账号类型有：

- 阿里云账号（AccessKey）

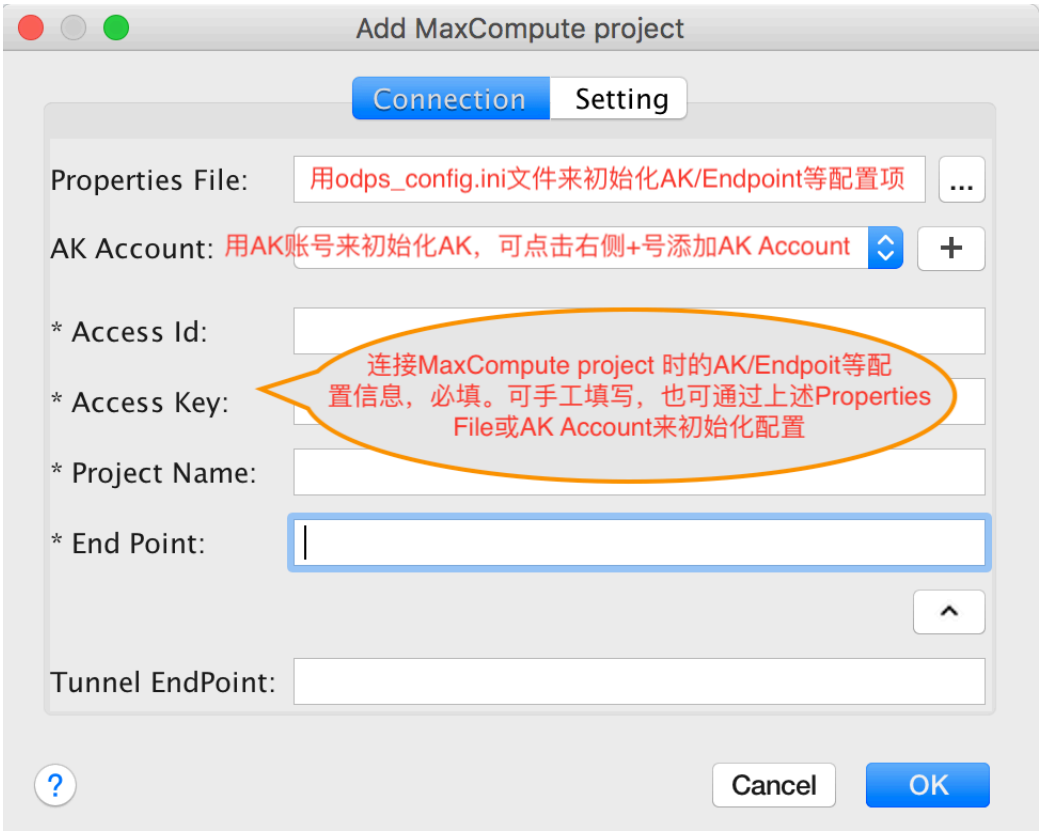
添加账户

在 Account 配置选项页面，执行以下步骤：

1. 点击按钮 + 或者快捷键 `Ctrl-N`
2. 选择账户类型 Aliyun Account by AccessKey

3. 在弹出的 Add Account 窗口中填入:

   - Account Name: 该账户在 Studio 中的标识名称。
   - Using properties file: 从配置文件中读取 AccessKey ID 和 AccessKey Secret。
     - 选择一个在 用户认证 中 conf/odps_config.ini 示例的配置文件。
   - Using properties: 手动填入 AccessKey ID 和 AccessKey Secret。
     - Access Id: 填入用户阿里云账号的 AccessKey ID。
     - Access Key: 填入用户阿里云账号的 AccessKey Secret。



4. 点击按钮 OK 完成添加。 添加完成后账号会出现在 Account 配置选项页面的 Account 列表里。

删除账户

在 Account 配置选项页面，执行以下步骤（该操作仅在 Studio 配置中删除账户配置，对用户账户本身不产生影响）：

1. 在 Account 列表中选择要删除的账户名称。
2. 点击按钮 -。
3. 在弹出的的确认对话框中，选择OK 。

修改账户 AccessKey

在 Account 配置选项页面，执行以下步骤：

1. 在 Account 列表中选择要删除的账户名称。
2. 点击”铅笔”图标进行编辑。
3. 在弹出的 Edit Account 窗口中编辑 Account 信息，内容与上面的Add Account 窗口类似。

查看MaxCompute Region的开通情况和连接方式及Endpoint的设置，请参见配置Endpoint。

2.11 Studio视频介绍

- Studio 安装介绍
- 通过Studio管理数据
- 通过Studio编辑SQL
- Studio SQL Scripting
- 通过Studio开发UDF
- 通过Studio remote debug
- 通过Studio查看所有job

3 Eclipse开发插件

3.1 安装Eclipse插件

本文向您介绍如何在Eclipse安装MaxCompute插件。

为了方便您使用 *MapReduce* 及*UDF*的Java SDK进行开发工作，MaxCompute（原 ODPS）提供了Eclipse开发插件。该插件能够模拟MapReduce及UDF的运行过程，为您提供本地调试手段，并提供了简单的模板生成功能。

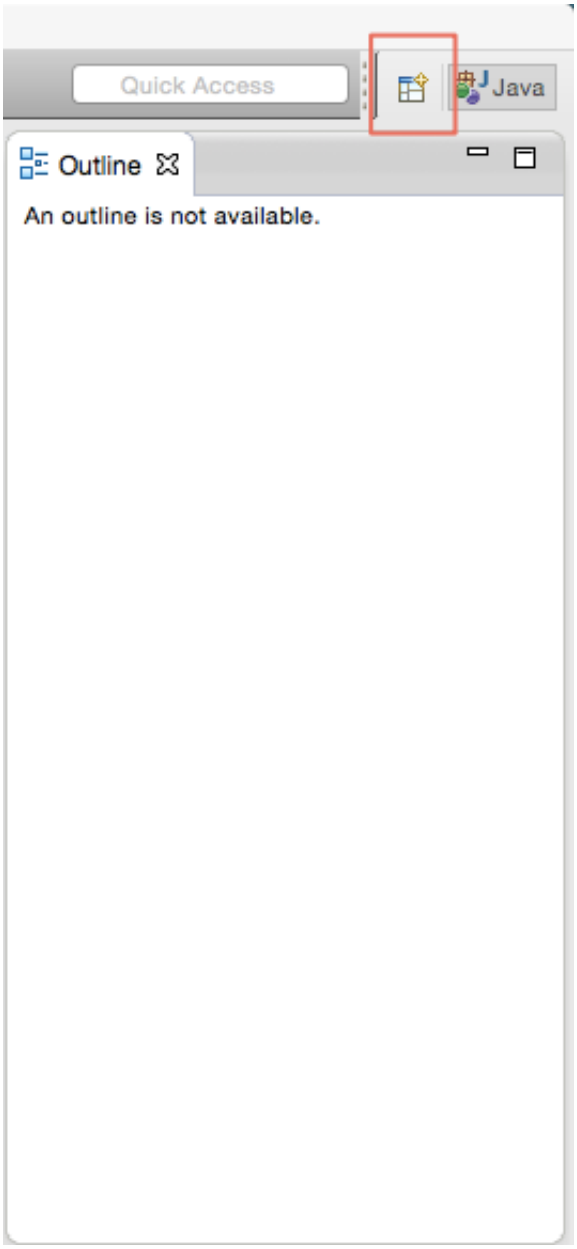
说明:

- 目前高版本的Eclipse Neon有可能会导致插件加载失败，请使用Eclipse Luna版本。
- 请单击 此处 下载插件。
- 与MapReduce提供的本地运行模式不同，Eclipse插件不能够与MaxCompute同步数据。您使用的数据需要手动拷贝到Eclipse插件的warehouse目录下。

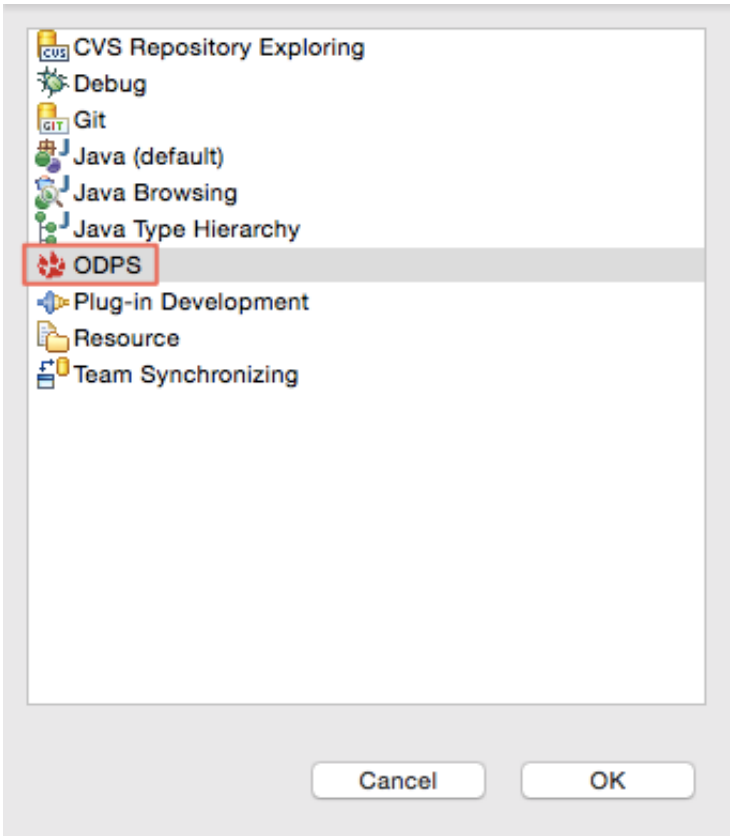
下载Eclipse插件后，将软件包解压，会看到如下Jar内容：

```
odps-eclipse-plugin-bundle-0.16.0.jar
```

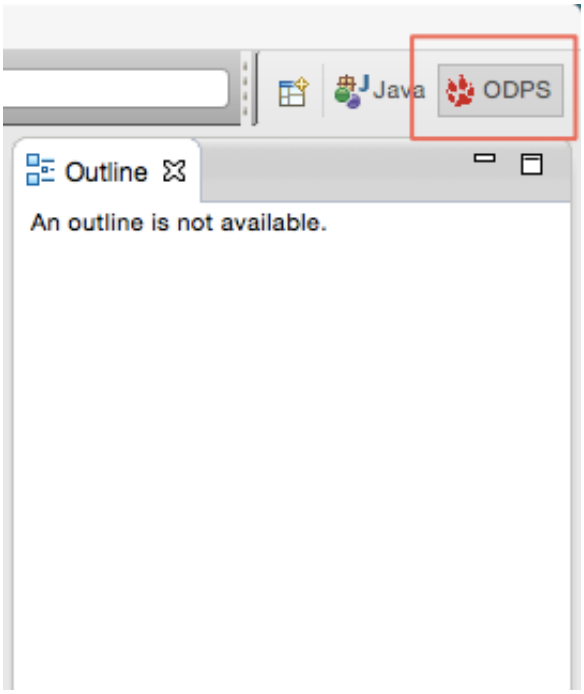
将插件放置在Eclipse安装目录的plugins子目录下。打开Eclipse，单击右上角的打开透视图（Open Perspective），如下图。



单击后出现对话框，如下图。



选择ODPS，单击OK 。同样在右上角会出现ODPS图标，表示插件生效。如下图。

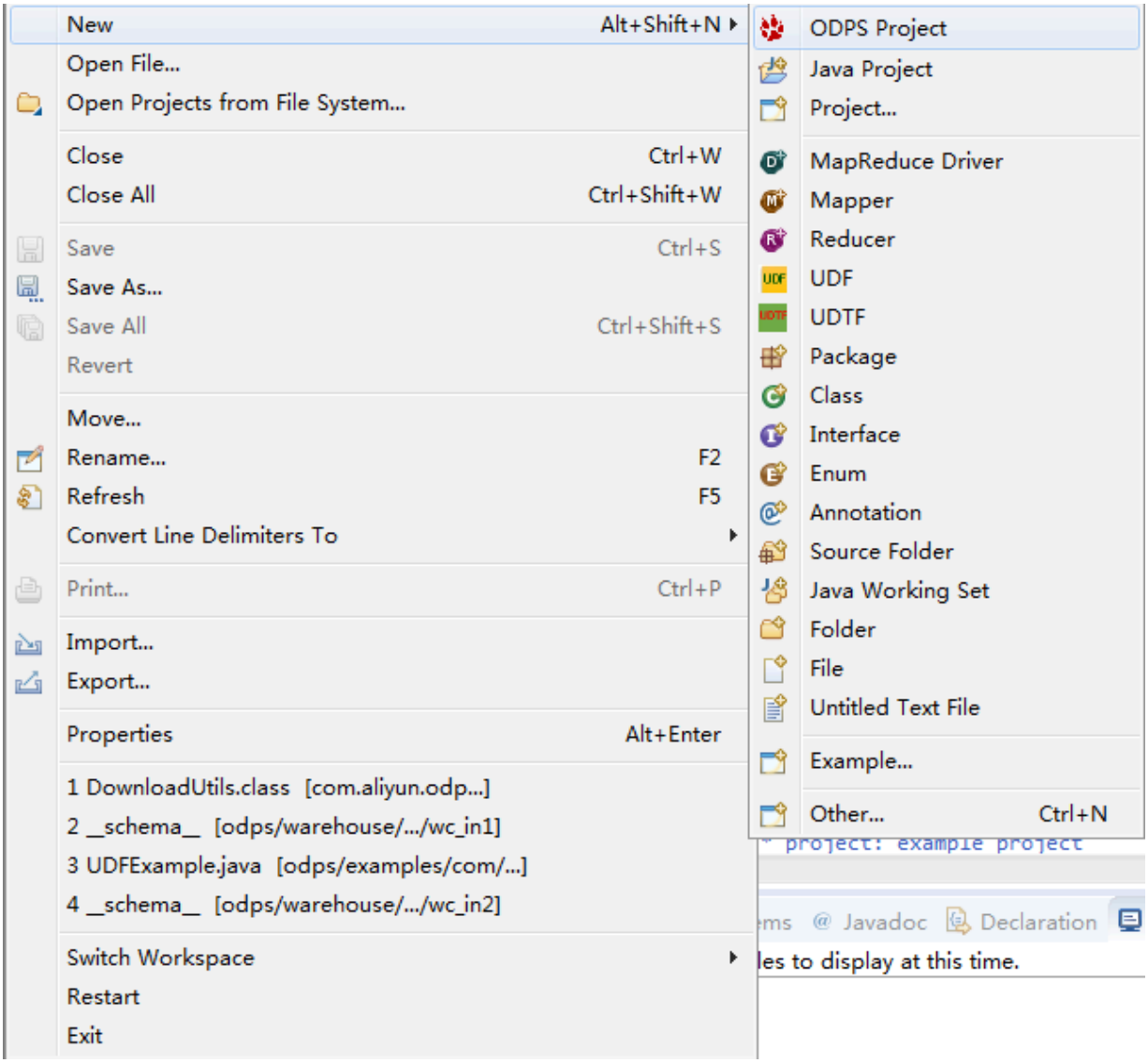


3.2 创建MaxCompute工程

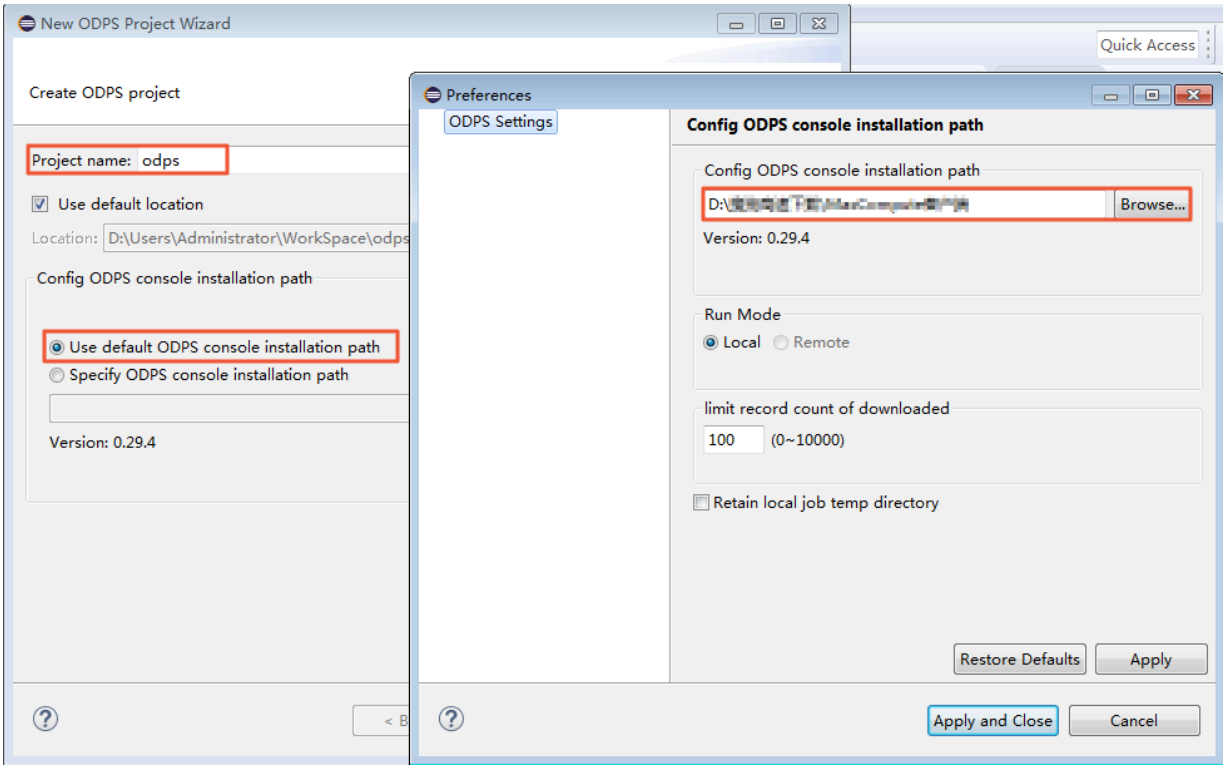
本文向您介绍Eclipse创建MaxCompute工程的两种方式。

方式一

在左上角选择文件(File) > 新建(New) > Project > ODPS > ODPS Project，创建工程(示例中使用ODPS作为工程名)，如下图。



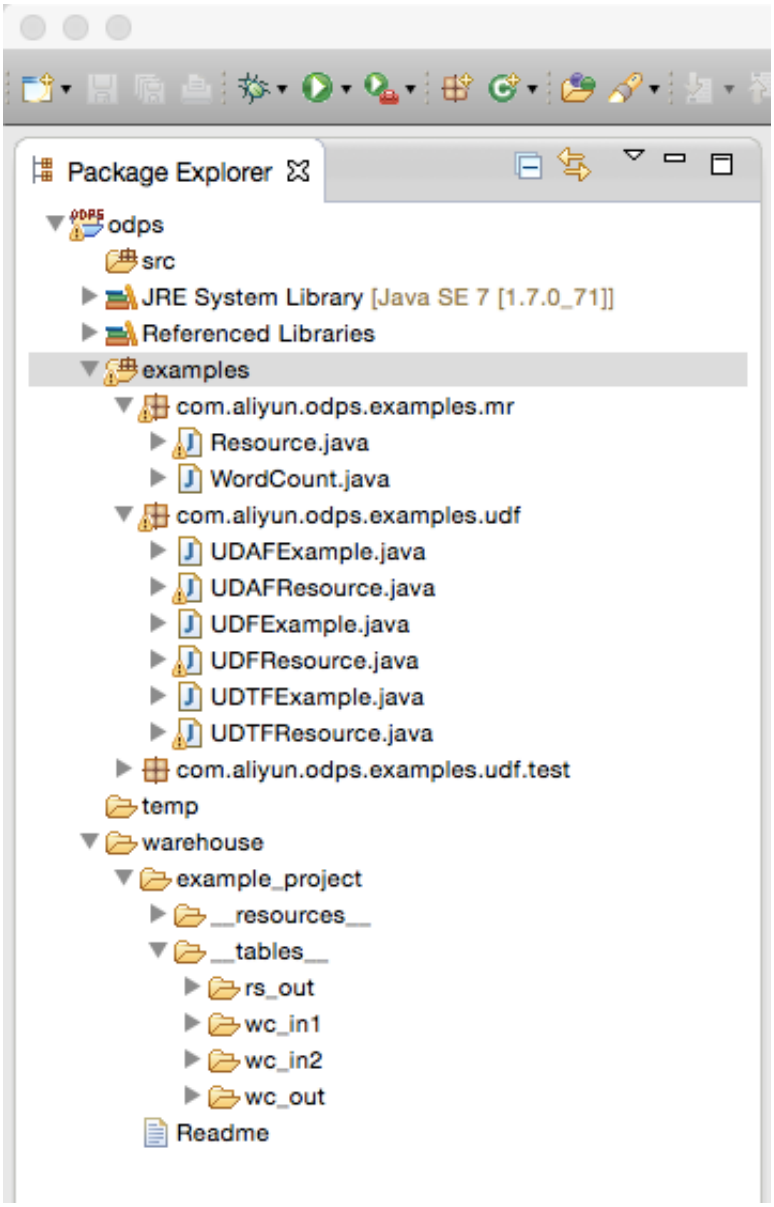
创建MaxCompute工程后会出现如下对话框。输入Project name，选择MaxCompute客户端路径(客户端需要提前下载)，并确认(点击Finish)，如下图。



说明:

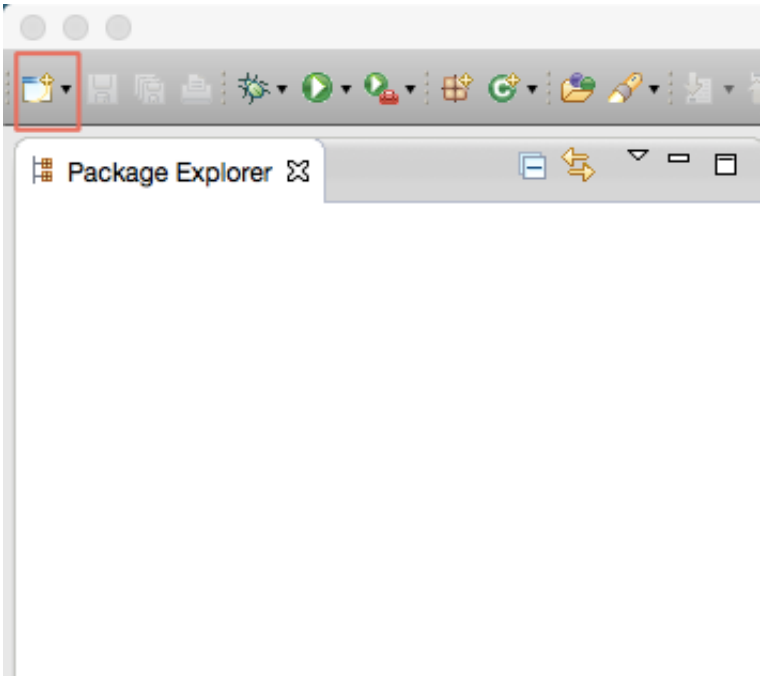
关于MaxCompute客户端的介绍请参考 客户端 。

创建好工程后，在左侧包资源管理器(Package Explorer)中可以看到目录结构，如下图。

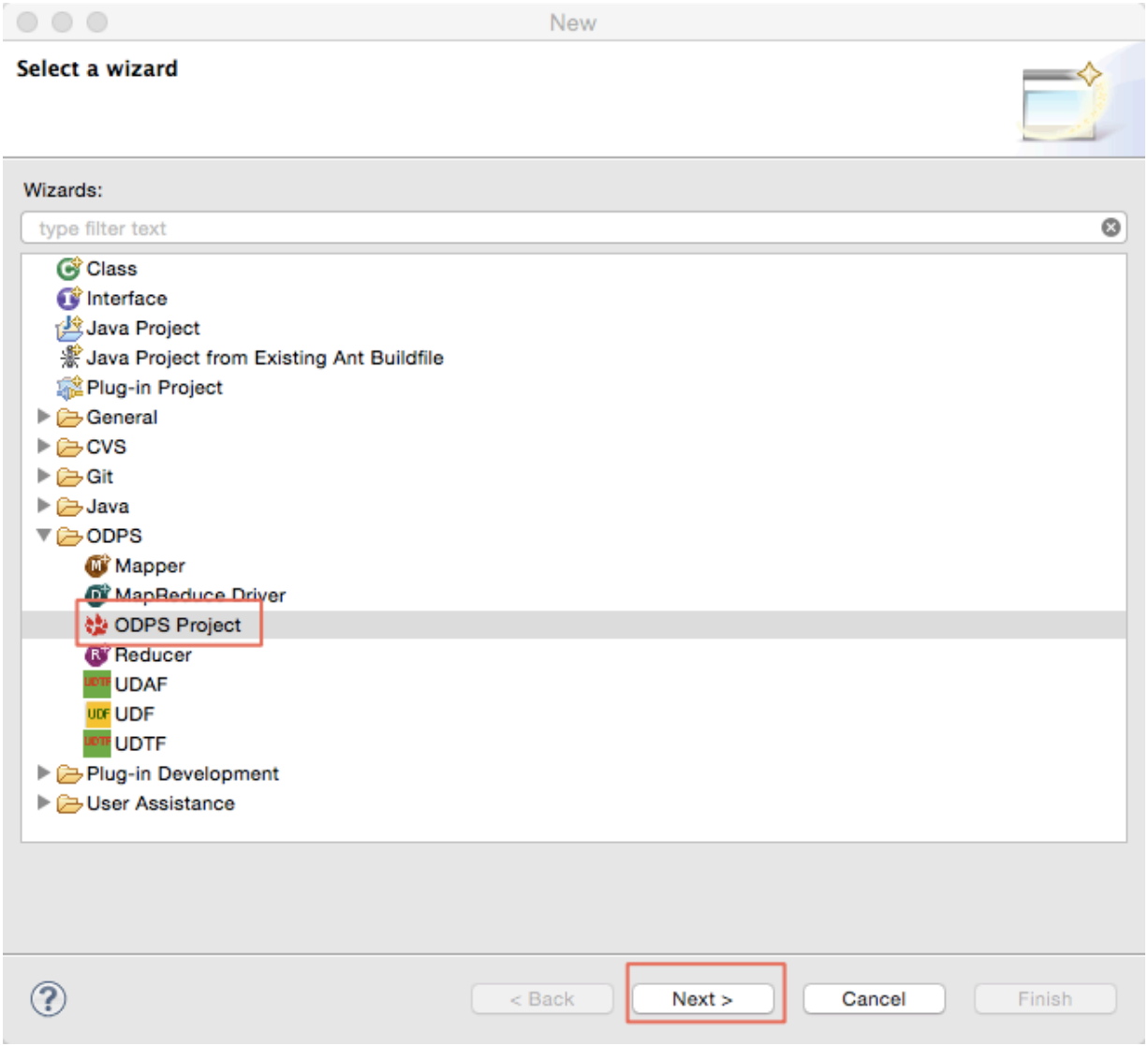


方式二

直接点击左上角的新建，如下图。



弹出对话框后，选择ODPS Project，单击下一步，如下图。



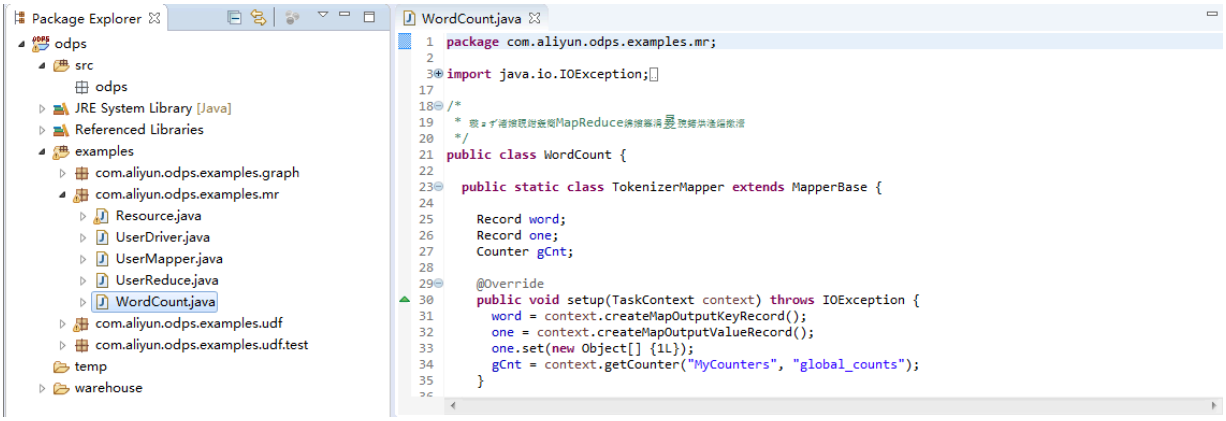
后续操作同方式一。

MaxCompute Eclipse插件安装结束。用户可以使用此插件编写MapReduce或者UDF程序。关于插件中对MapReduce的功能介绍请参考*MapReduce*开发插件介绍，UDF的程序编写请参考*UDF*开发插件介绍。

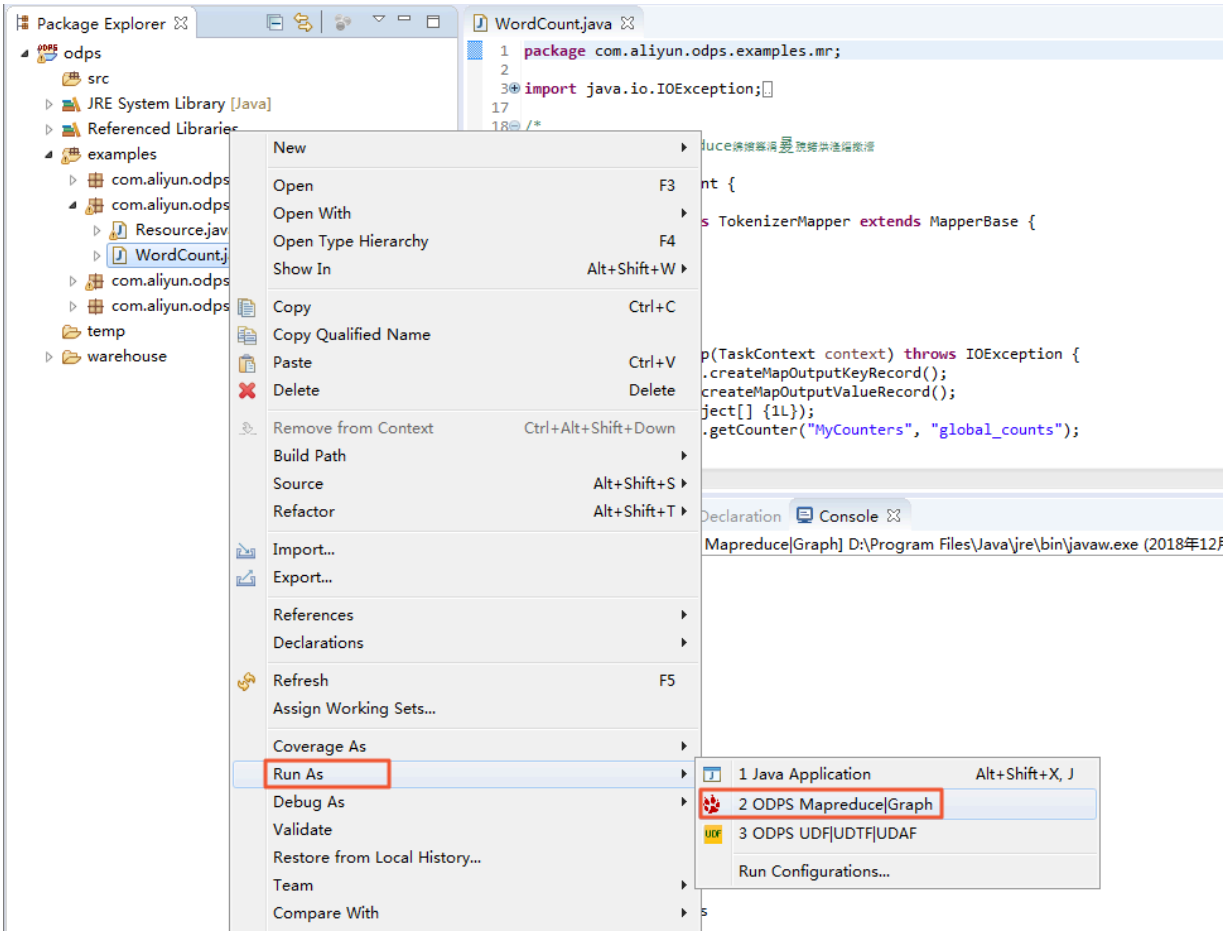
3.3 MapReduce开发插件介绍

本文向您介绍如何使用Eclipse开发和运行MapReduce程序。

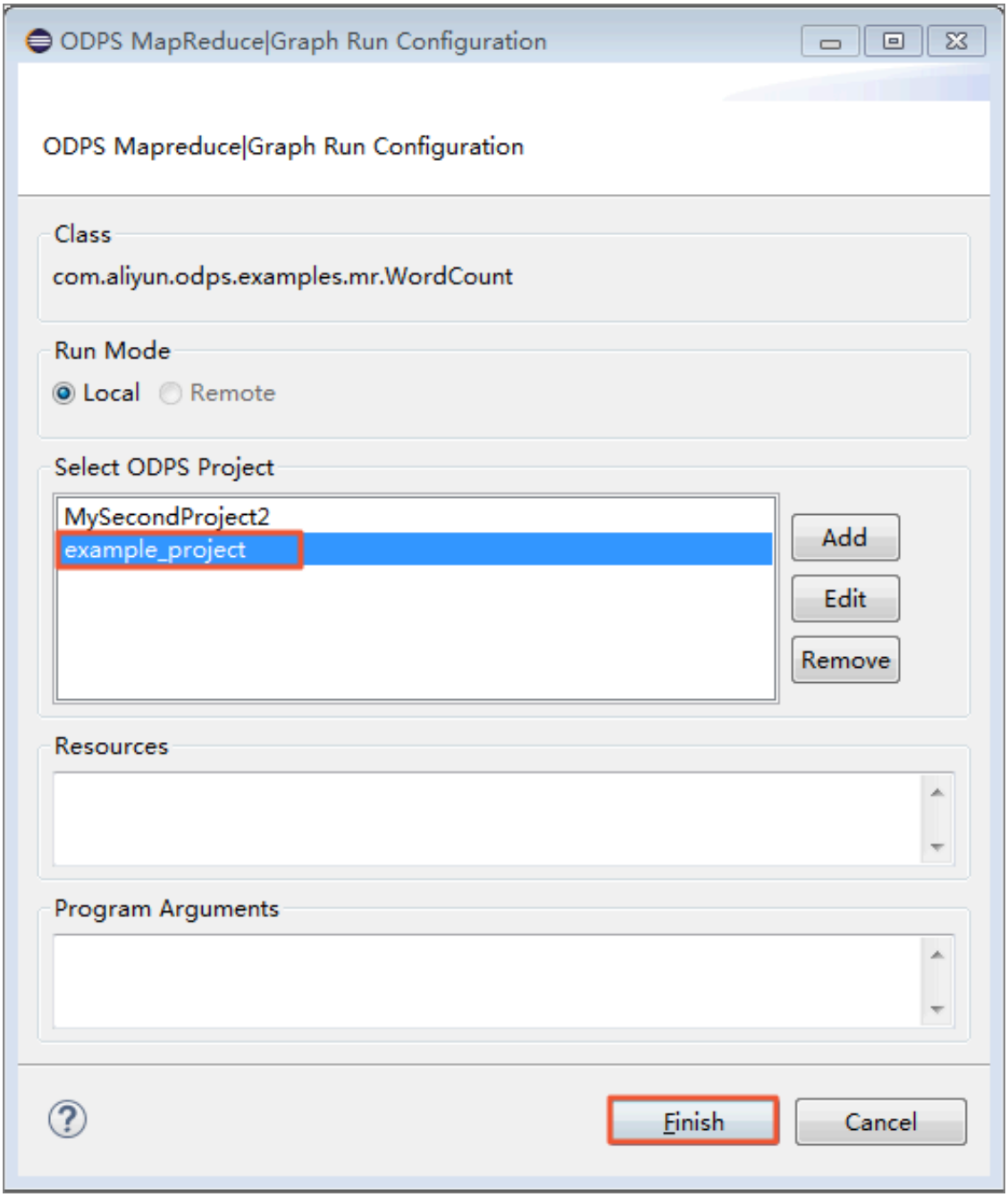
选择ODPS项目中的WordCount示例，如下图。



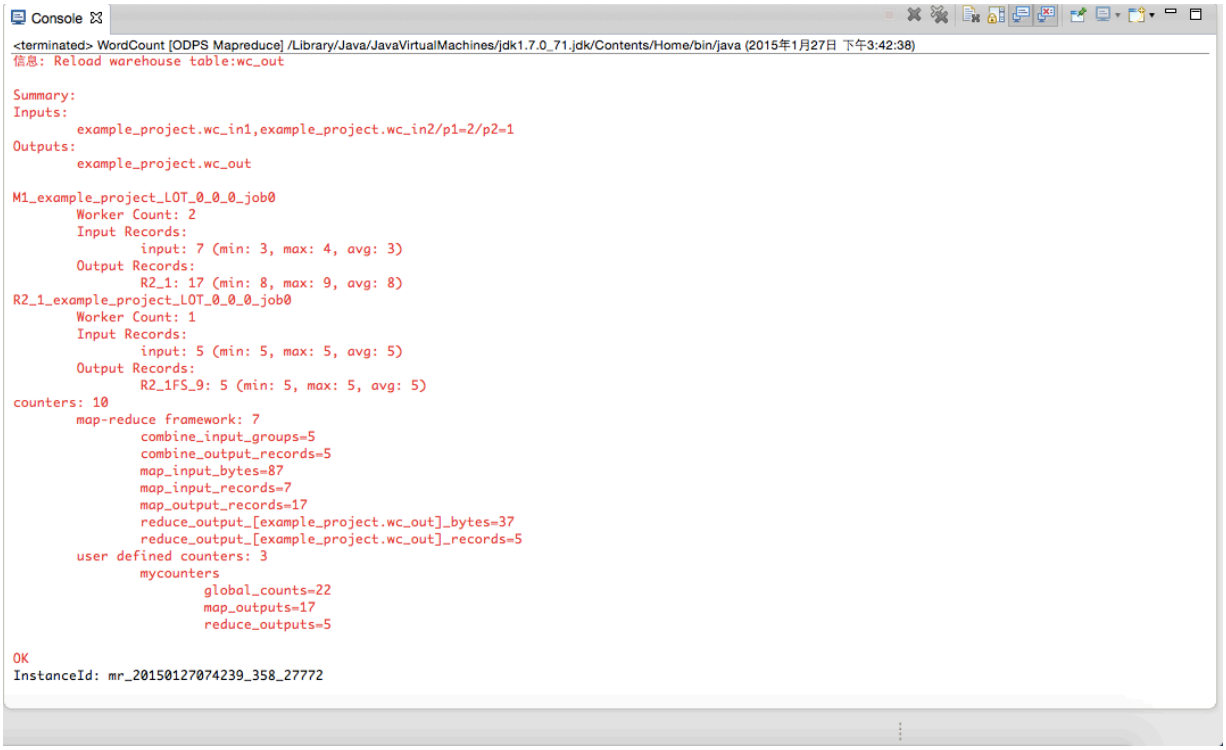
右键单击WordCount.java，依次单击Run As > ODPS MapReduce，如下图。



弹出对话框后，选择example_project，单击确认，如下图。

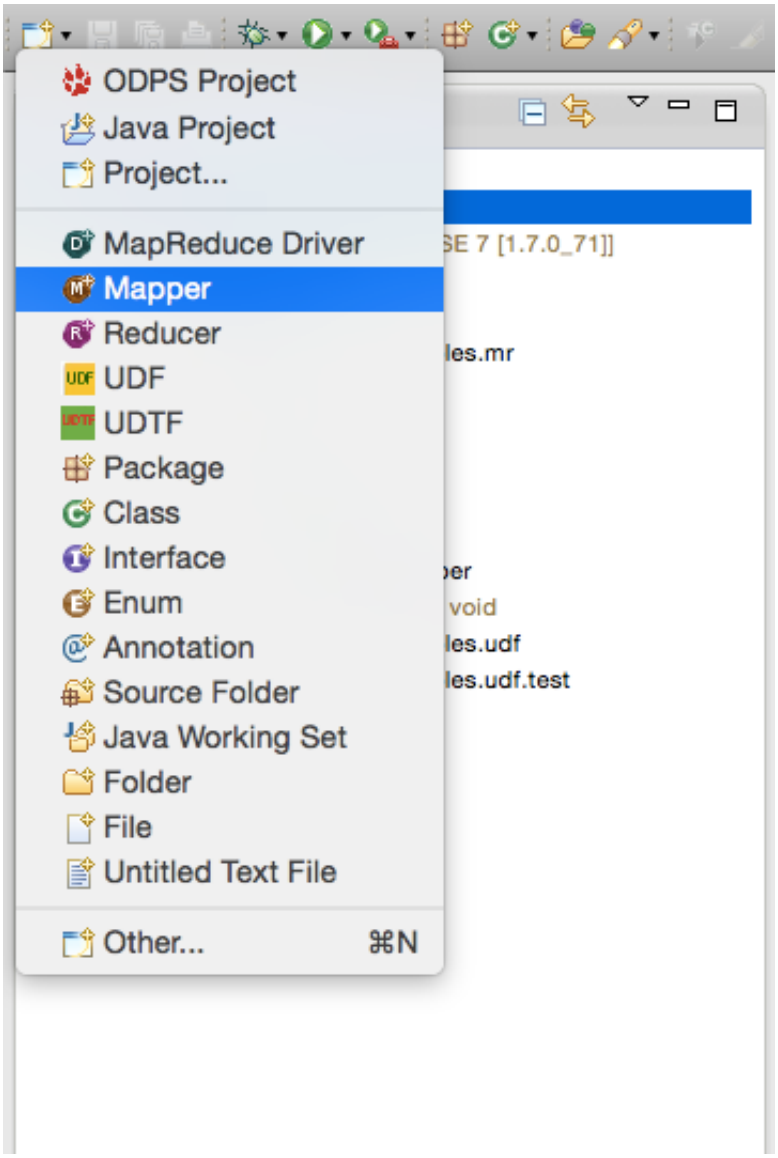


运行成功后，会出现结果提示，如下图。

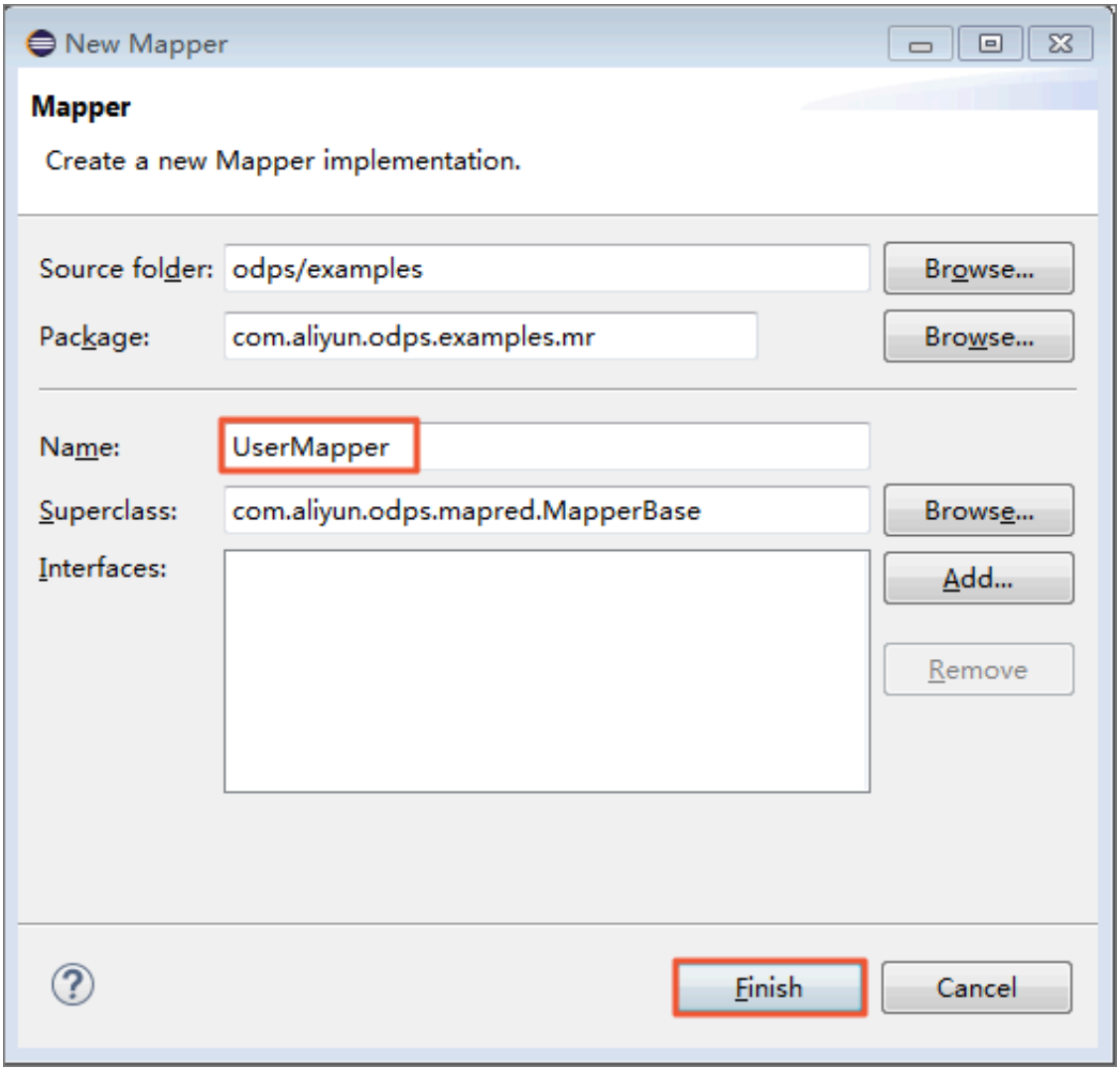


运行自定义MapReduce程序

右键选择src目录，选择新建(New) > Mapper，如下图。



选择Mapper后出现下面的对话框。输入Mapper类的名字，并确认。如下图。



会看到在左侧包资源管理器(Package Explorer)中，src目录下生成文件UserMapper.java。该文件的内容即是一个Mapper类的模板，如下图。

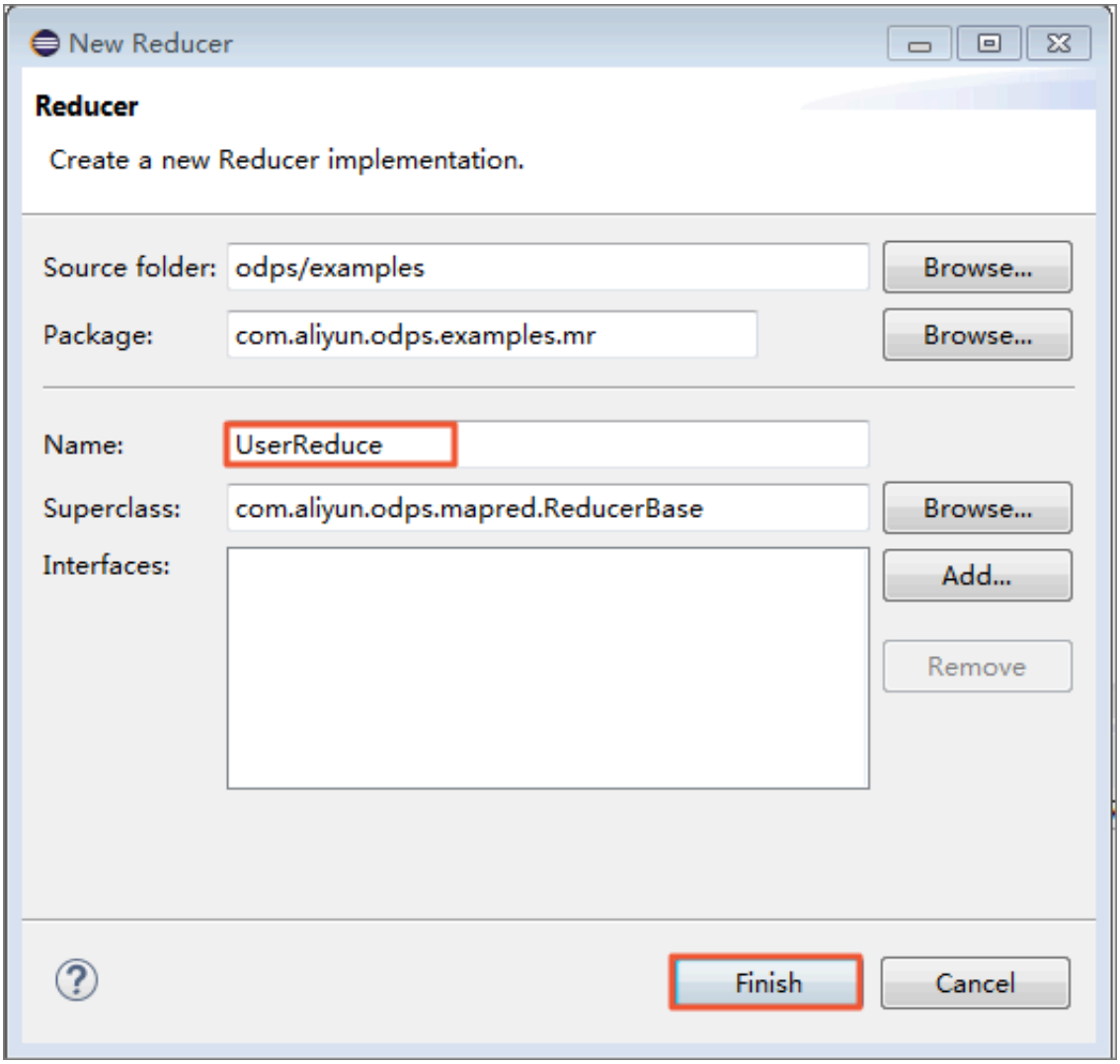
```
package odps;
import java.io.IOException;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.mapred.MapperBase;
public class UserMapper extends MapperBase {
    @Override
    public void setup(TaskContext context) throws IOException {
    }
    @Override
    public void map(long recordNum, Record record, TaskContext context
)
            throws IOException {
    }
    @Override
    public void cleanup(TaskContext context) throws IOException {
    }
```

```
}
```

模板中，将package名称默认配置为“odps”，用户可以根据自己的需求进行修改。编写模板内容，如下图。

```
package odps;
import java.io.IOException;
import com.aliyun.odps.counter.Counter;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.mapred.MapperBase;
public class UserMapper extends MapperBase {
    Record word;
    Record one;
    Counter gCnt;
    @Override
    public void setup(TaskContext context) throws IOException {
          word = context.createMapOutputKeyRecord();
          one = context.createMapOutputValueRecord();
          one.set(new Object[] { 1L });
          gCnt = context.getCounter("MyCounters", "global_counts");
    }
    @Override
    public void map(long recordNum, Record record, TaskContext context
)
            throws IOException {
          for (int i = 0; i < record.getColumnCount(); i++) {
              String[] words = record.get(i).toString().split("\\s+");
              for (String w : words) {
                word.set(new Object[] { w });
                Counter cnt = context.getCounter("MyCounters", "
map_outputs");
                cnt.increment(1);
                gCnt.increment(1);
                context.write(word, one);
              }
            }
          }
    @Override
    public void cleanup(TaskContext context) throws IOException {
    }
}
```

同理，右键选择src目录，选择新建(New) > Reduce，如下图。



输入Reduce类的名字(本示例使用UserReduce):

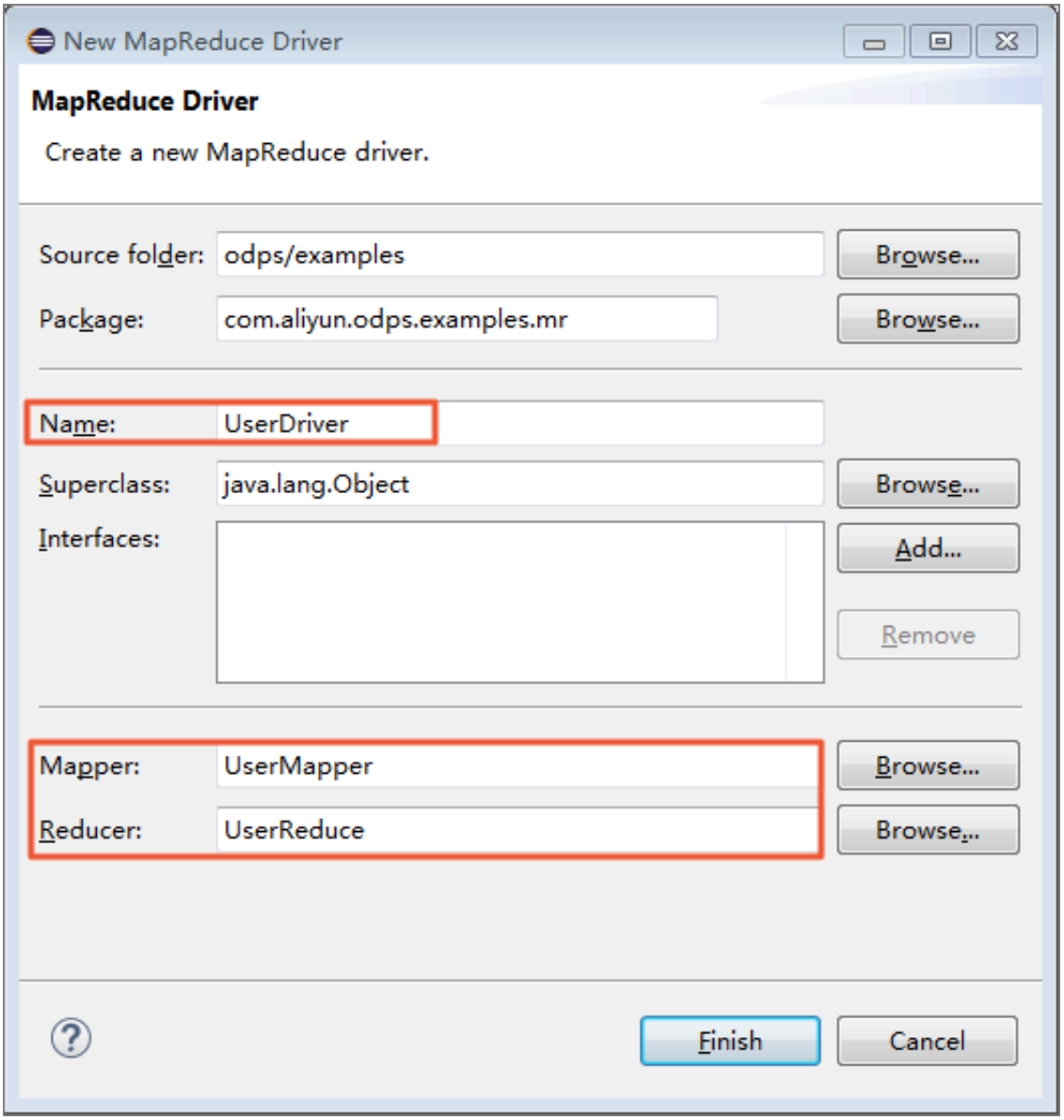
同样在包资源管理器(Package Explorer)中，src目录下生成文件*UserReduce.java*。该文件的内容即是一个Reduce类的模板。编辑模板，如下图。

```
package odps;
import java.io.IOException;
import java.util.Iterator;
import com.aliyun.odps.counter.Counter;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.mapred.ReducerBase;
public class UserReduce extends ReducerBase {
    private Record result;
    Counter gCnt;
    @Override
    public void setup(TaskContext context) throws IOException {
          result = context.createOutputRecord();
          gCnt = context.getCounter("MyCounters", "global_counts");
    }
    @Override
    public void reduce(Record key, Iterator<Record> values, TaskContex
t context)
            throws IOException {
          long count = 0;
```

```
            while (values.hasNext()) {
              Record val = values.next();
              count += (Long) val.get(0);
            }
            result.set(0, key.get(0));
            result.set(1, count);
            Counter cnt = context.getCounter("MyCounters", "reduce_out
puts");
            cnt.increment(1);
            gCnt.increment(1);
            context.write(result);
          }
    @Override
    public void cleanup(TaskContext context) throws IOException {
    }
}
```

创建main函数: 右键选择src目录，选择新建(New) > MapReduce Driver。

填写Driver Name(示例中是UserDriver)，Mapper及Recduce类(示例中是UserMapper及UserReduce)，并确认。同样会在src目录下看到MyDriver.java文件，如下图。

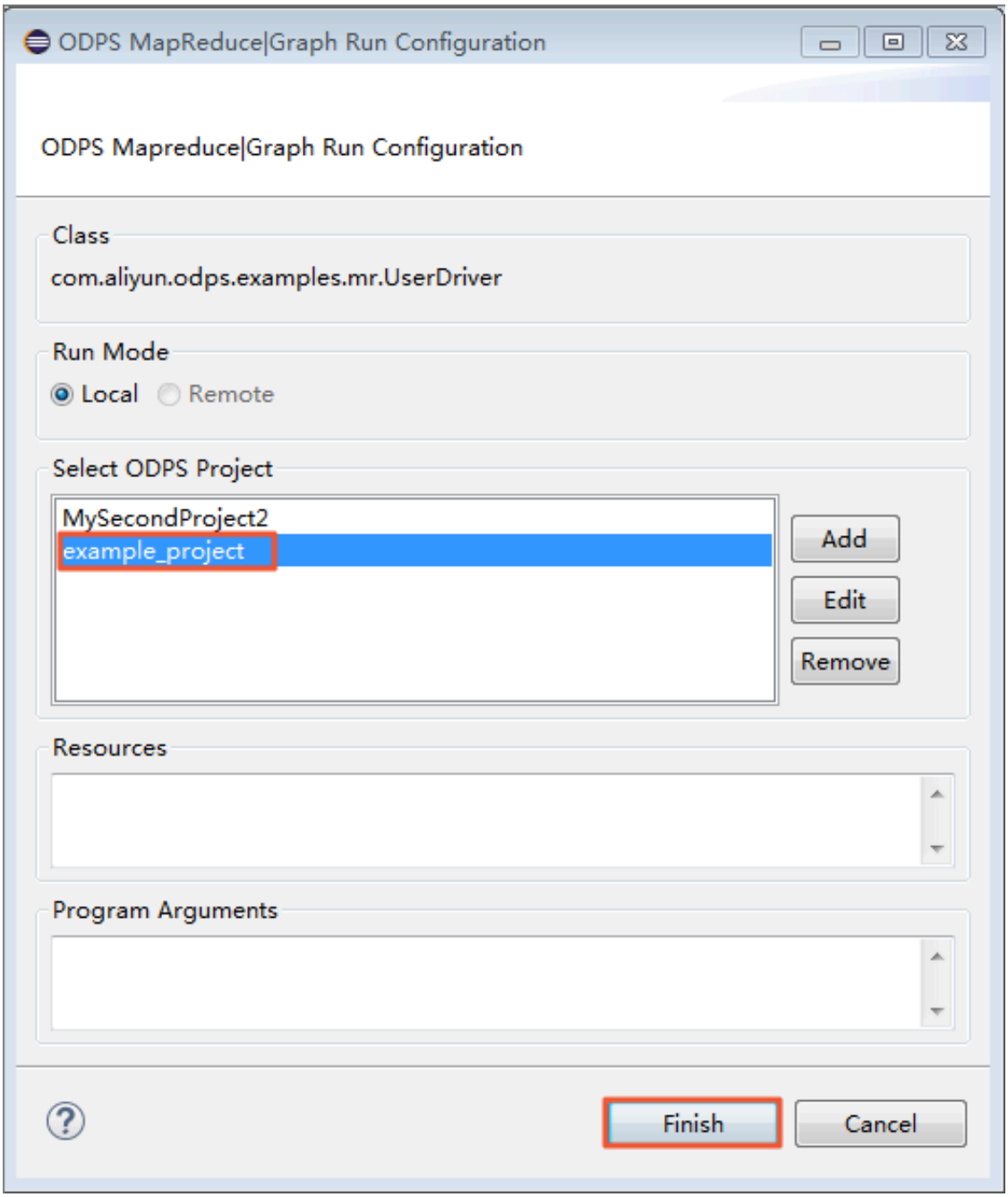


编辑driver内容，如下图。

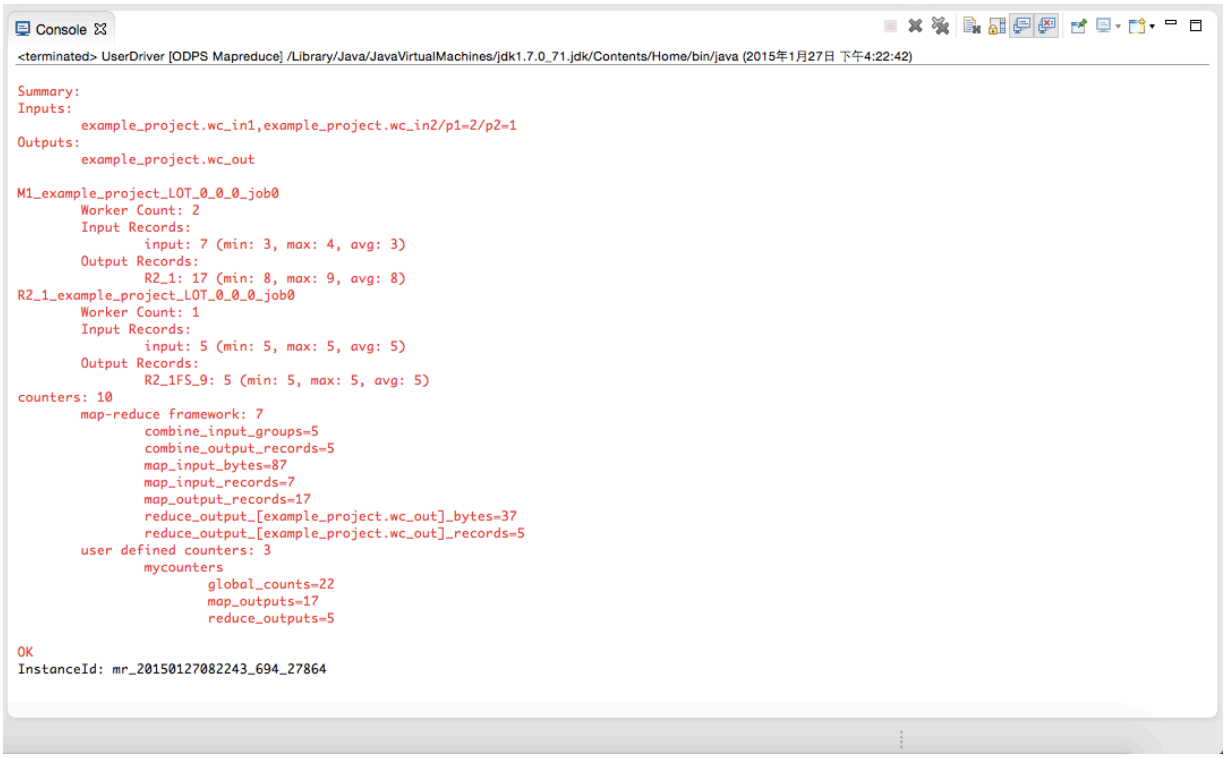
```
package odps;
import com.aliyun.odps.OdpsException;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.examples.mr.WordCount.SumCombiner;
import com.aliyun.odps.examples.mr.WordCount.SumReducer;
import com.aliyun.odps.examples.mr.WordCount.TokenizerMapper;
import com.aliyun.odps.mapred.JobClient;
import com.aliyun.odps.mapred.RunningJob;
import com.aliyun.odps.mapred.conf.JobConf;
import com.aliyun.odps.mapred.utils.InputUtils;
import com.aliyun.odps.mapred.utils.OutputUtils;
import com.aliyun.odps.mapred.utils.SchemaUtils;
public class UserDriver {
    public static void main(String[] args) throws OdpsException {
        JobConf job = new JobConf();
        job.setMapperClass(TokenizerMapper.class);
        job.setCombinerClass(SumCombiner.class);
        job.setReducerClass(SumReducer.class);
        job.setMapOutputKeySchema(SchemaUtils.fromString("word:string
"));
```

```
        job.setMapOutputValueSchema(SchemaUtils.fromString("count:
bigint"));
        InputUtils.addTable(
            TableInfo.builder().tableName("wc_in1").cols(new String[]
 { "col2", "col3" }).build(), job);
        InputUtils.addTable(TableInfo.builder().tableName("wc_in2").
partSpec("p1=2/p2=1").build(), job);
        OutputUtils.addTable(TableInfo.builder().tableName("wc_out").
build(), job);
        RunningJob rj = JobClient.runJob(job);
        rj.waitForCompletion();
    }
}
```

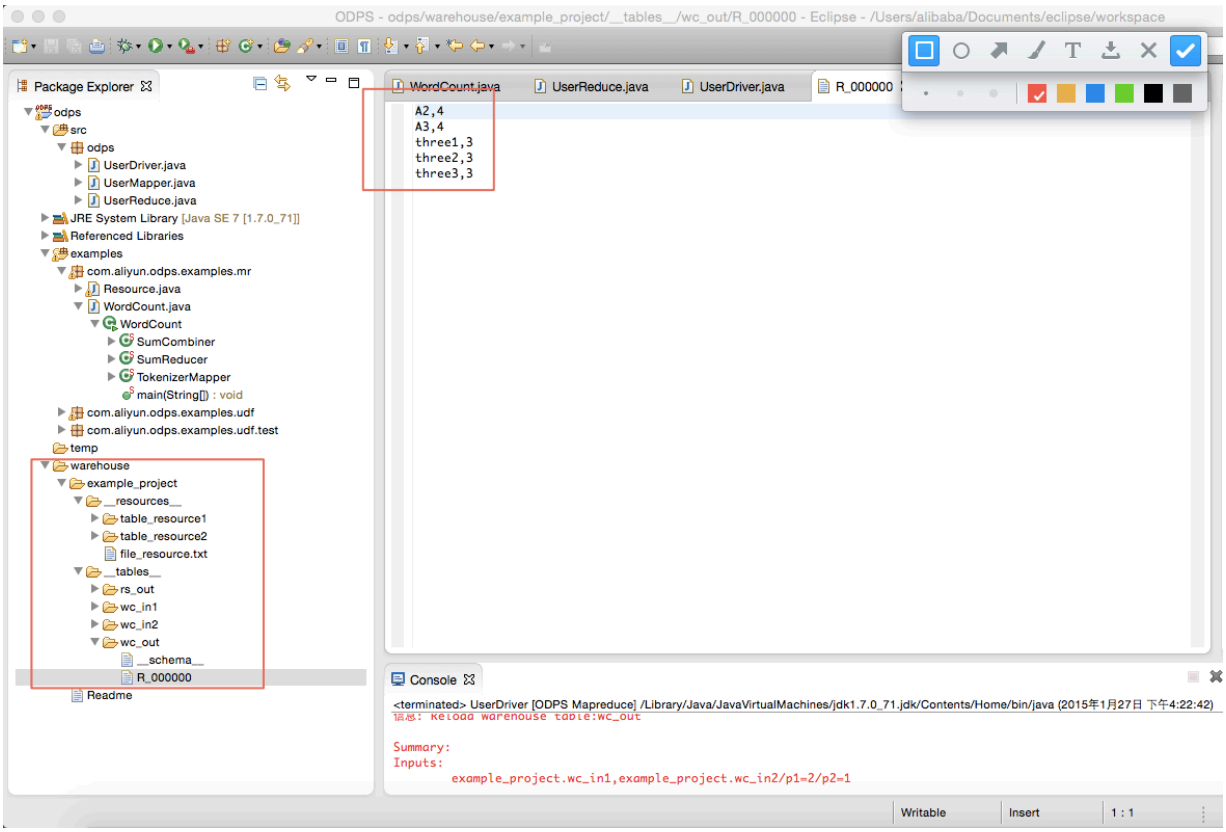
运行MapReduce程序，选中UserDriver.java，右键选择Run As > ODPS MapReduce，点击确认。出现对话框，如下图。



选择ODPS Project为：example_project，单击Finish按钮开始本地运行MapReduce程序，如下图。

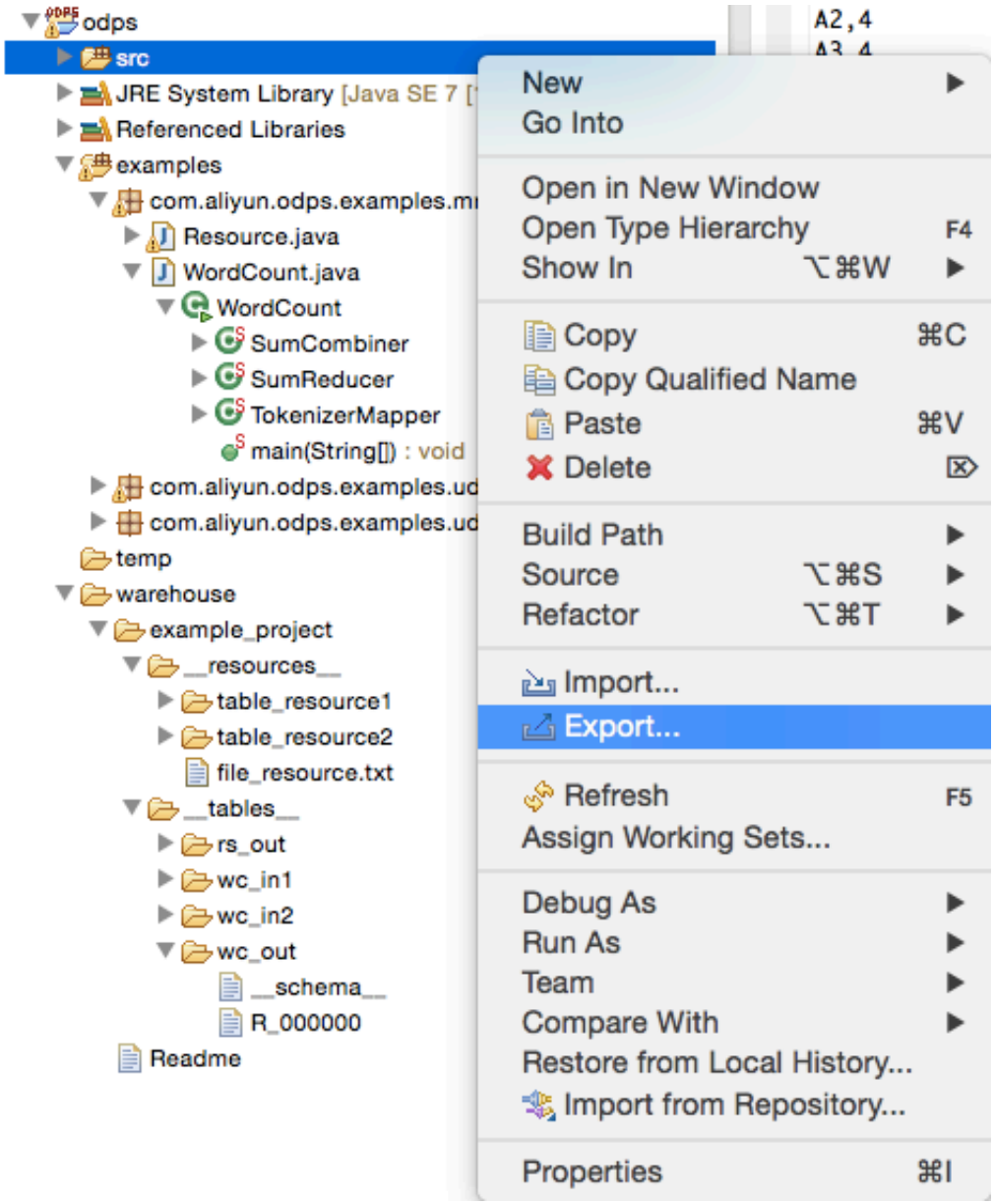


有如上输出信息，说明本地运行成功。运行的输出结果在warehouse目录下。关于warehouse的说明请参考本地运行 。刷新ODPS工程，如下图。

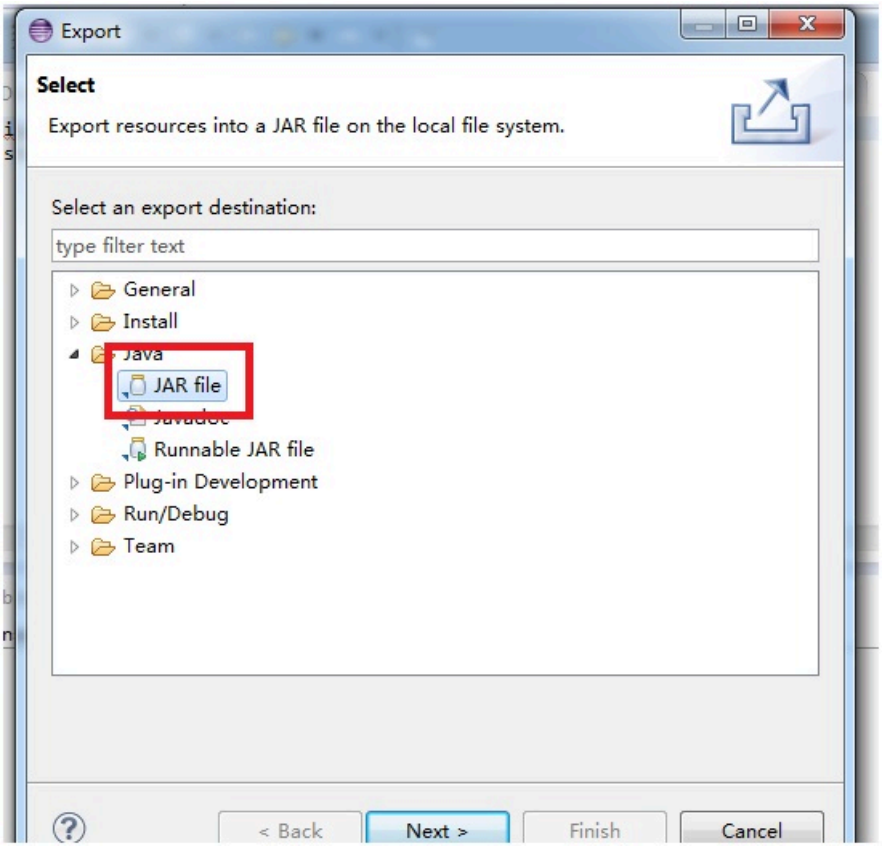


wc_out即是输出目录，R_000000即是结果文件。通过本地调试，确定输出结果正确后，可以通过Eclipse导出(Export)功能将MapReduce打包。打包后将jar包上传到ODPS中。在分布式环境下执行MapReduce，详情请参考 快速入门。

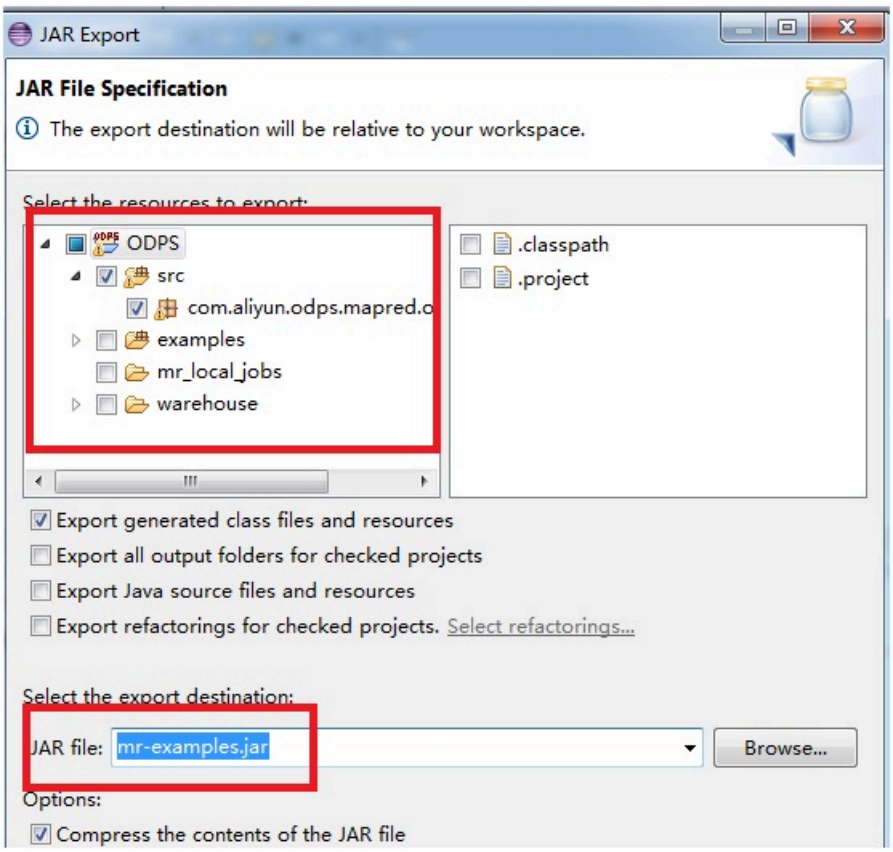
本地调试通过后，用户可以通过Eclipse的Export功能将代码打成jar包，供后续分布式环境使用。在本示例中，我们将程序包命名为*mr-examples.jar*。选择src目录，单击Export，如下图。



选择导出模式为Jar File，如下图。



仅需要导出src目录下package(com.aliyun.odps.mapred.open.example)，Jar File名称指定为*mr-examples.jar*，如下图。



确认后，导出成功。

如果用户想在本地模拟新建Project，可以在warehouse下面，创建一个新的子目录(与example_project平级的目录)，目录层次结构，如下图。

```
<warehouse>
   |____example_project(项目空间目录)
          |____ <__tables__>
          |        |__table_name1(非分区表)
          |        |       |____ data(文件)
          |        |       |
          |        |       |____ <__schema__> (文件)
          |        |
          |        |__table_name2(分区表)
          |             |____ partition_name=partition_value(分区目录)
          |             |           |____ data(文件)
          |             |
          |             |____ <__schema__> (文件)
          |
          |____ <__resources__>
                   |
                   |___table_resource_name (表资源)
                   |          |____<__ref__>
                   |
                   |___ file_resource_name (文件资源)
```

schema文件示例，如下图。

```
非分区表:
project=project_name
table=table_name
columns=col1:BIGINT,col2:DOUBLE,col3:BOOLEAN,col4:DATETIME,col5:STRING
分区表:
project=project_name
table=table_name
columns=col1:BIGINT,col2:DOUBLE,col3:BOOLEAN,col4:DATETIME,col5:STRING
partitions=col1:BIGINT,col2:DOUBLE,col3:BOOLEAN,col4:DATETIME,col5:
STRING
注：当前支持5种数据格式:bigint,double,boolean,datetime,string， 对应到java
中的数据类型-long,double,boolean,java.util.Date,java.lang.String。
```

data文件示例，如下图。

```
1,1.1,true,2015-06-04 11:22:42 896,hello world
\N,\N,\N,\N,\N
注：时间格式精确到毫秒级别，所有类型用\N表示null。
```

说明:

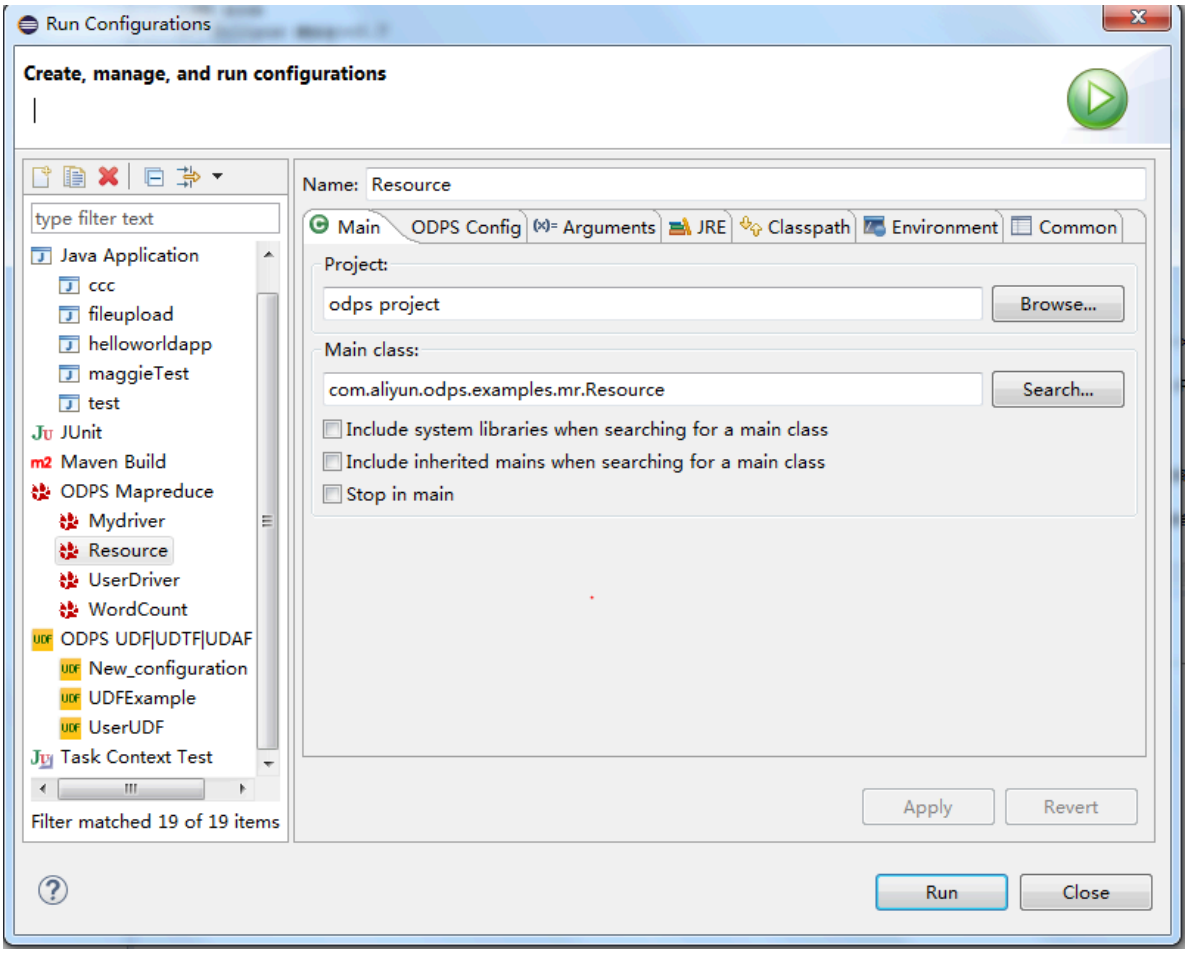
- 本地模式运行MapReduce程序，默认情况下先到warehouse下查找相应的数据表或资源，如果表或资源不存在会到服务器上下载相应的数据存入warehouse目录下，再以本地模式运行。
- 运行完MapReduce后，请刷新warehouse目录，才能看到生成的结果。

3.4 UDF开发插件介绍

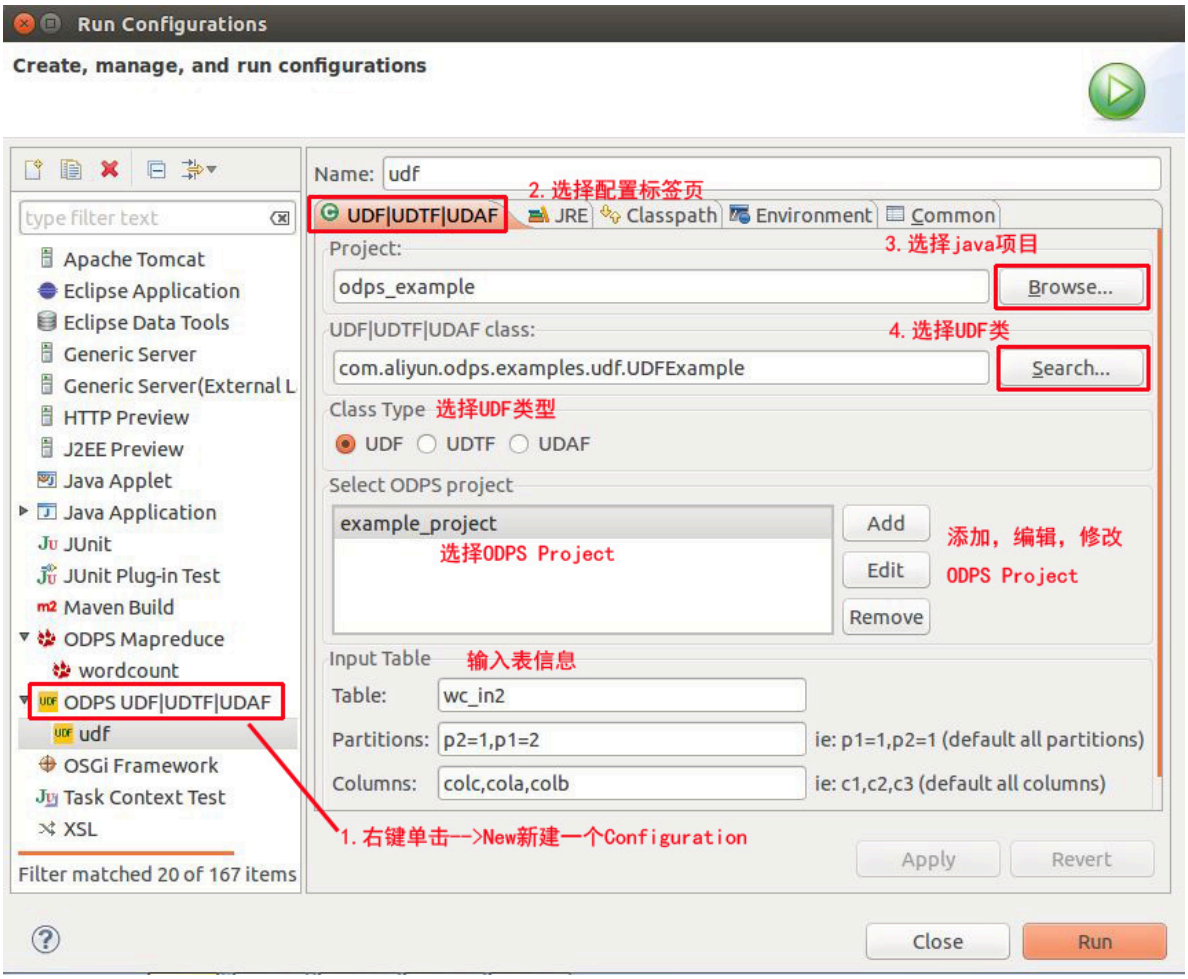
在本章节我们将介绍如何使用Eclipse插件开发并在本地运行UDF。UDAF、UDTF的编写执行过程与UDF类似，均可参考UDF的示例介绍完成。ODPS Eclipse插件提供两种运行UDF的方式：菜单栏和右键单击快速运行方式。

菜单栏运行

1. 从菜单栏选择Run > Run Configurations…弹出如下对话框：

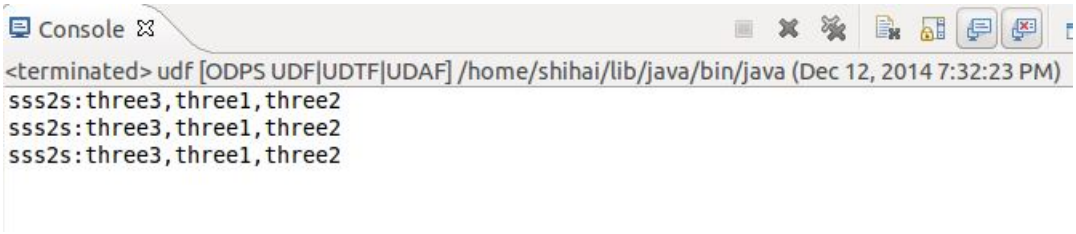


2. 用户可以新建一个Run Configuration，选择运行的UDF类及类型、选择ODPS Project、填写输入表信息，如：



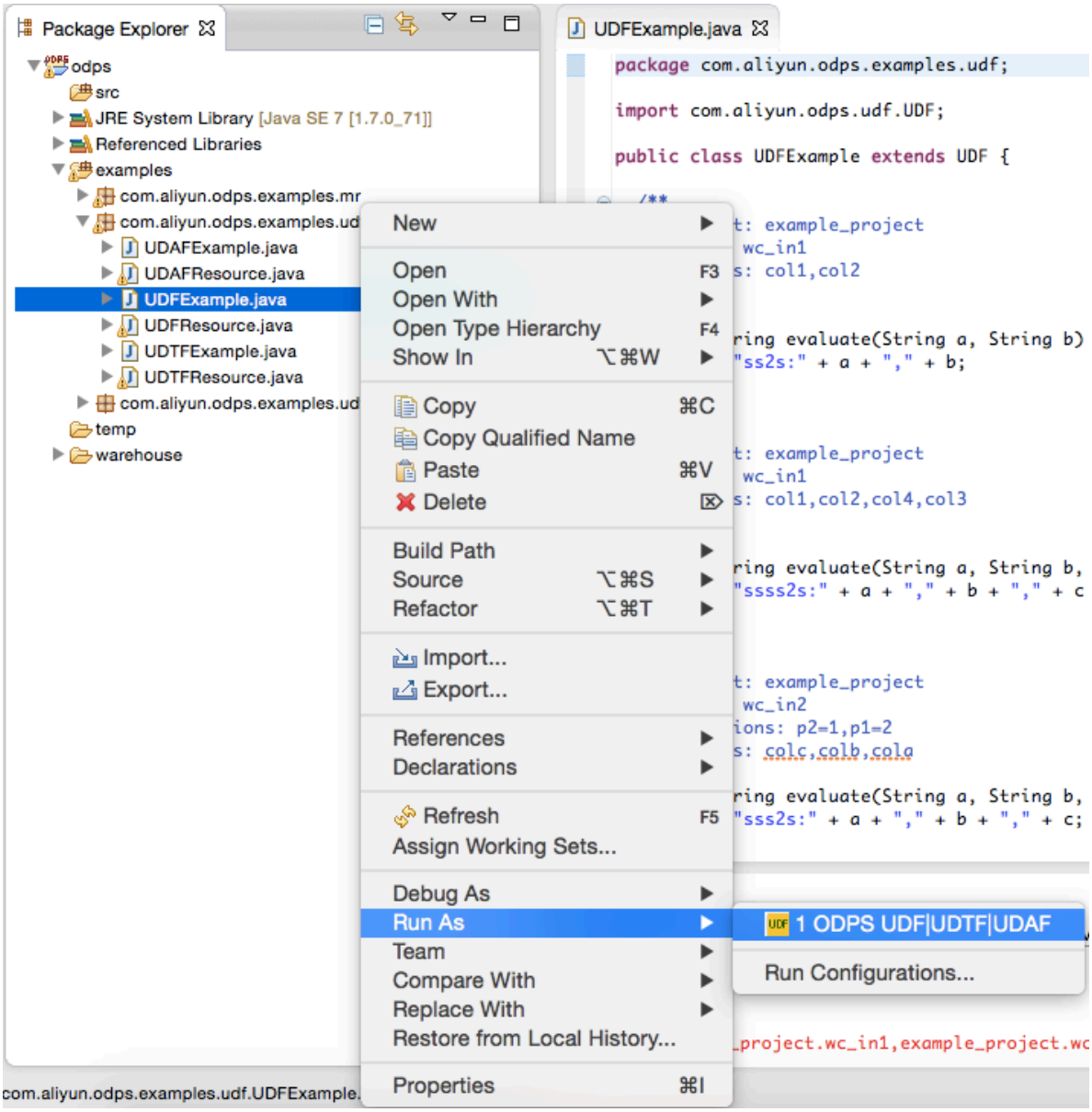
上述配置中，Table表示UDF的输入表，Partitions表示读取某个分区下的数据，分区由逗号分隔，Columns表示列，将依次作为UDF函数的参数被传入，列名由逗号分隔。

3. 点击Run运行，运行结果将显示在控制台中：

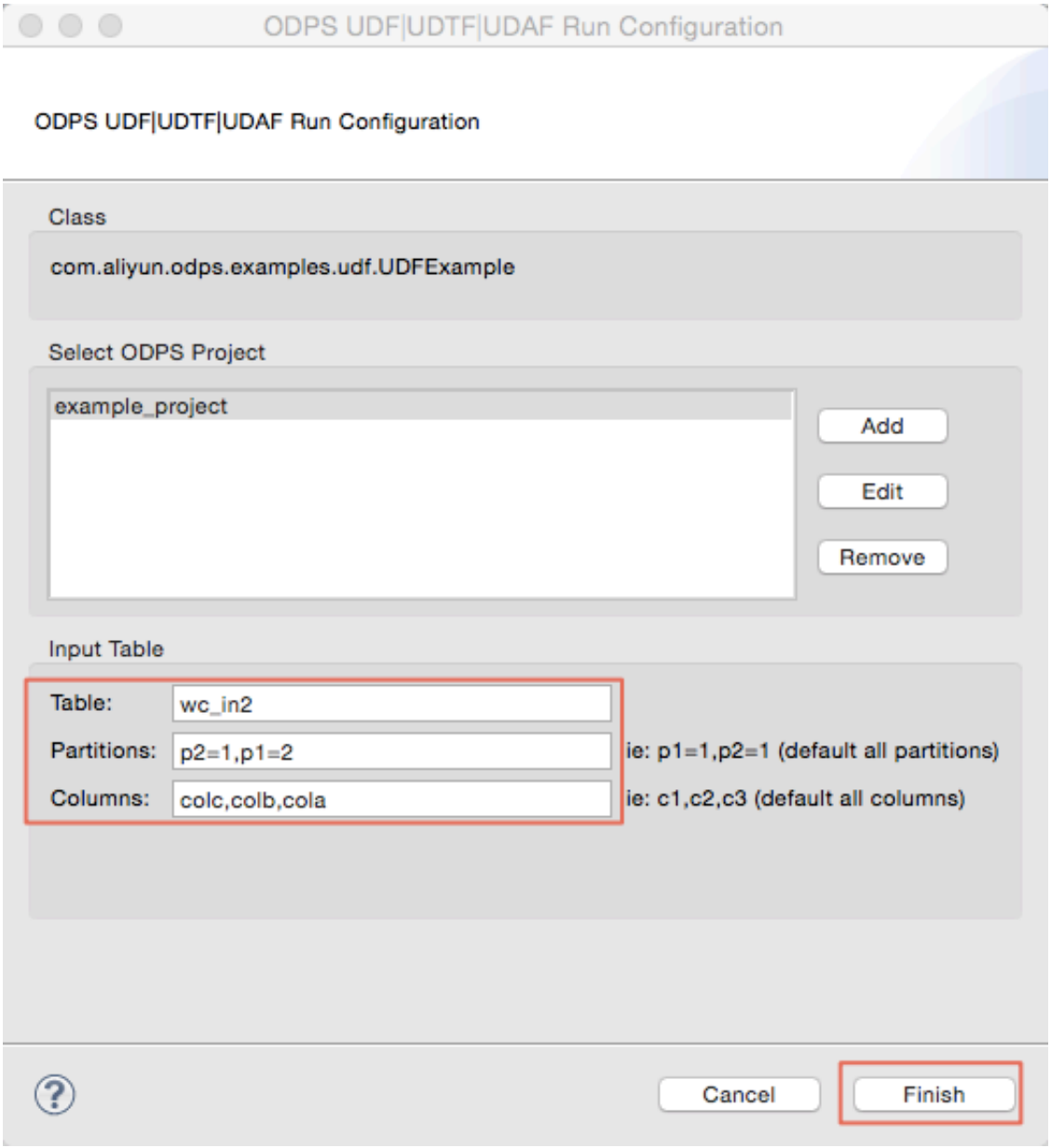


右键单击快速运行

1. 选中一个udf.java文件（比如：UDFExample.java）并单击鼠标右键，选择Run As > Run UDF|UDAF|UDTF。



2. 配置信息如下：



上述配置中，Table表示UDF的输入表，Partitions表示读取某个分区下的数据，分区由逗号分隔，Columns表示列，将依次作为UDF函数的参数被传入，列名由逗号分隔。

3. 点击Finish后，运行UDF，获得输出结果。

运行用户自定义UDF程序

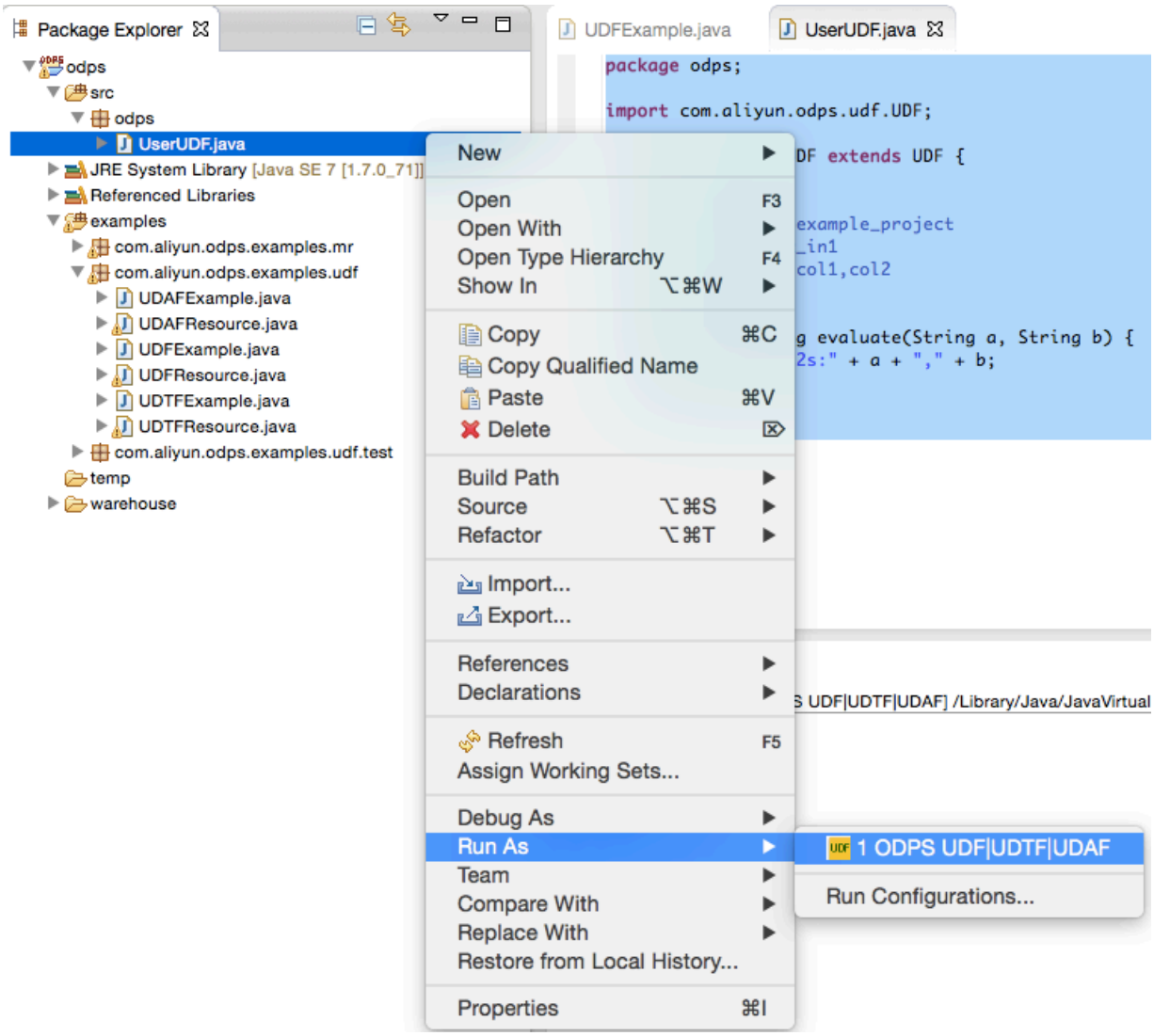
右击一个工程并选择New > UDF（或者选择菜单栏File > New > UDF)。

填写UDF类名然后点击Finish。在对应的src目录下生成与UDF类名同名的Java文件，编辑该java文件内容：

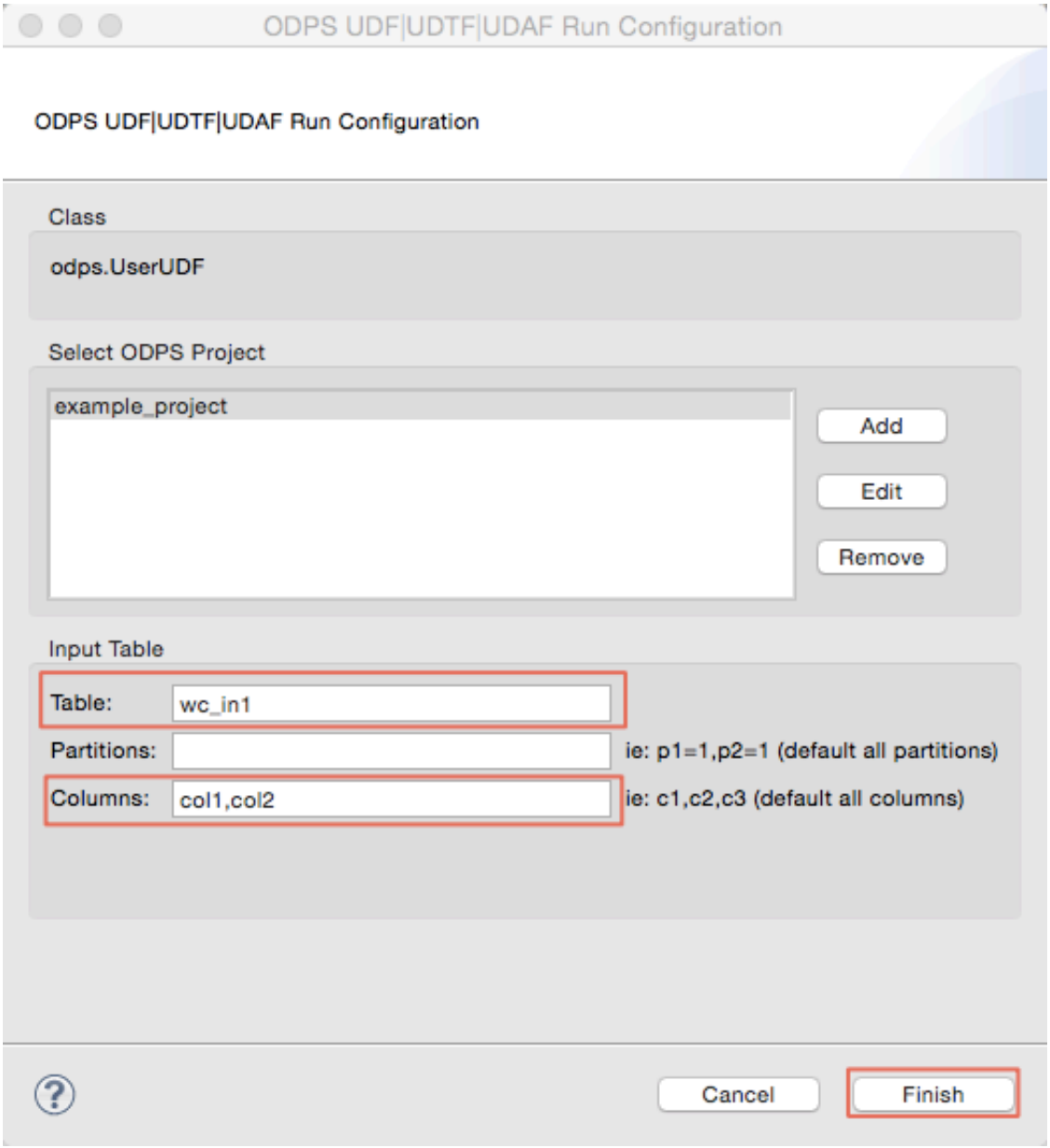
```
package odps;
import com.aliyun.odps.udf.UDF;
public class UserUDF extends UDF {
```

```
        /**
         * project: example_project
         * table: wc_in1
         * columns: col1,col2
         *
         */
        public String evaluate(String a, String b) {
          return "ss2s:" + a + "," + b;
        }
}
```

右击该java文件（如UserUDF.java），选择Run As，再选择ODPS UDF|UDTF|UDAF:



配置如下对话框：



点击Finish，得出结果：

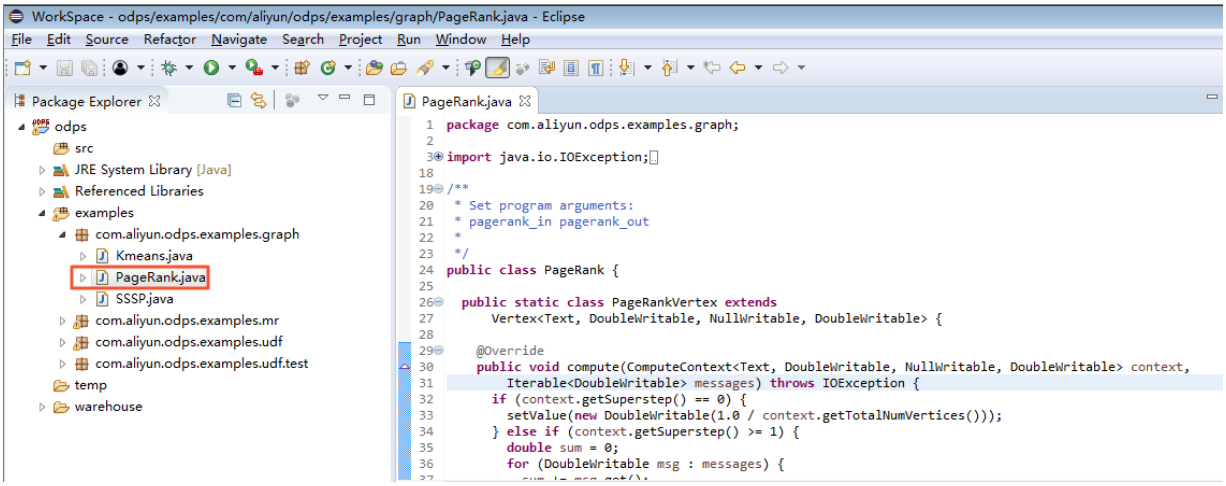
```
ss2s:A1,A2
ss2s:A1,A2
ss2s:A1,A2
ss2s:A1,A2
```

本示例中仅给出UDF的运行示例，UDTF的运行方式与UDF基本相同，不做特殊说明。

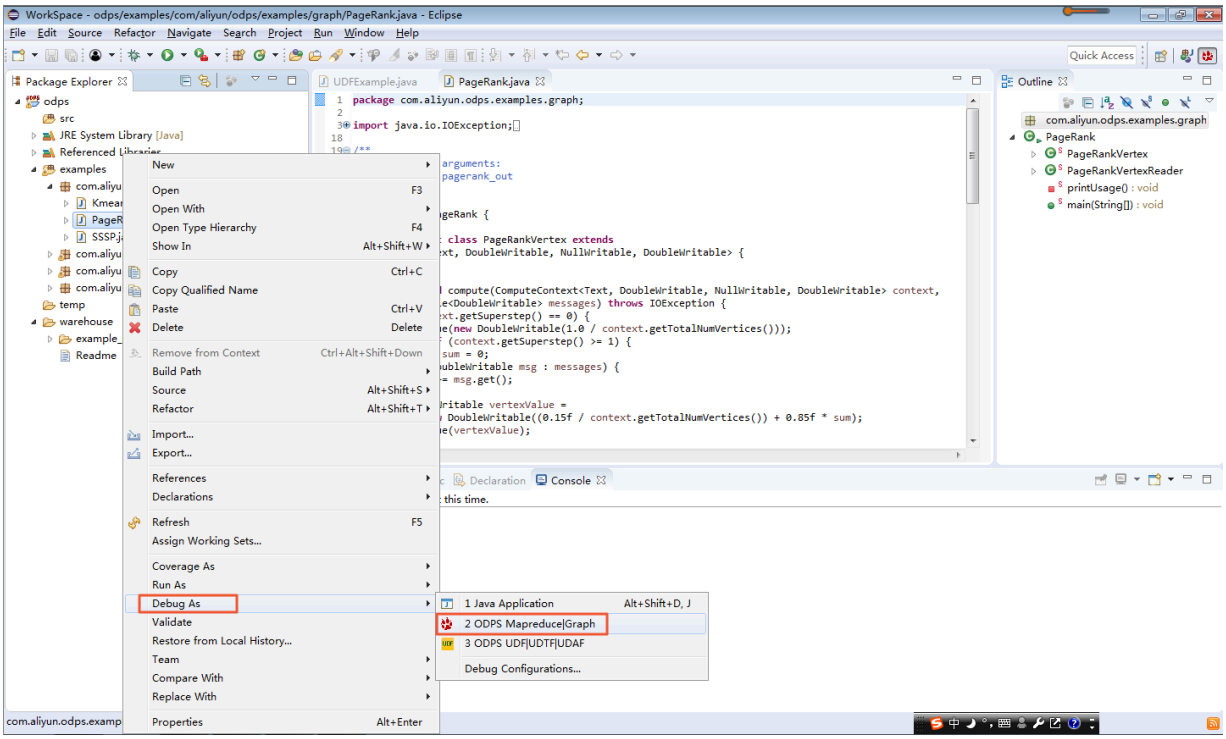
3.5 Graph开发插件介绍

创建ODPS项目后，用户可以编写自己的Graph程序，参照下文步骤操作完成本地调试。

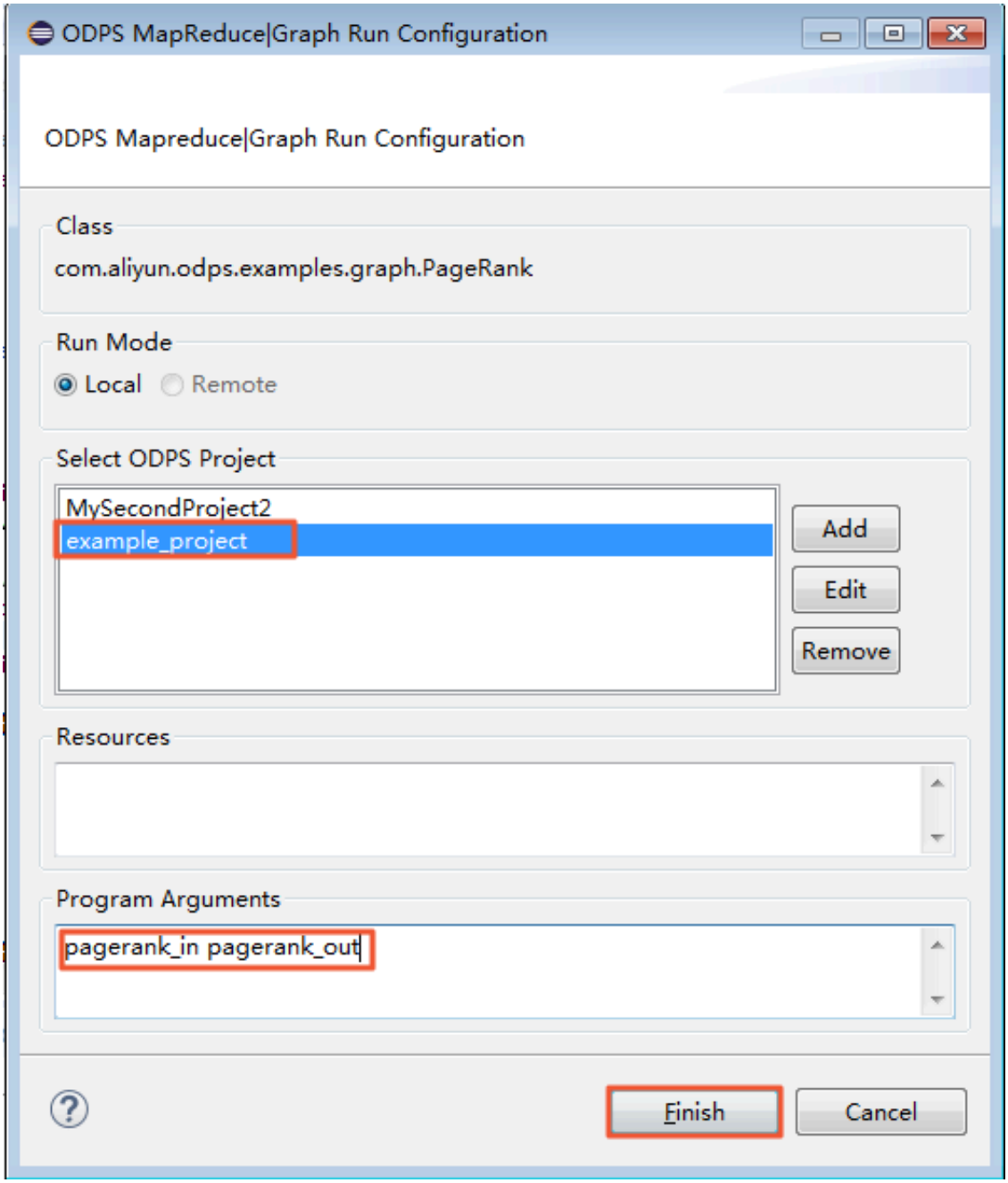
在此示例中，我们选用插件提供的*PageRank.java*来完成本地调试工作。选中examples下的*PageRank.java*文件，如下图。



右键单击，选择Debug As > ODPS MapReduce|Graph，如下图。



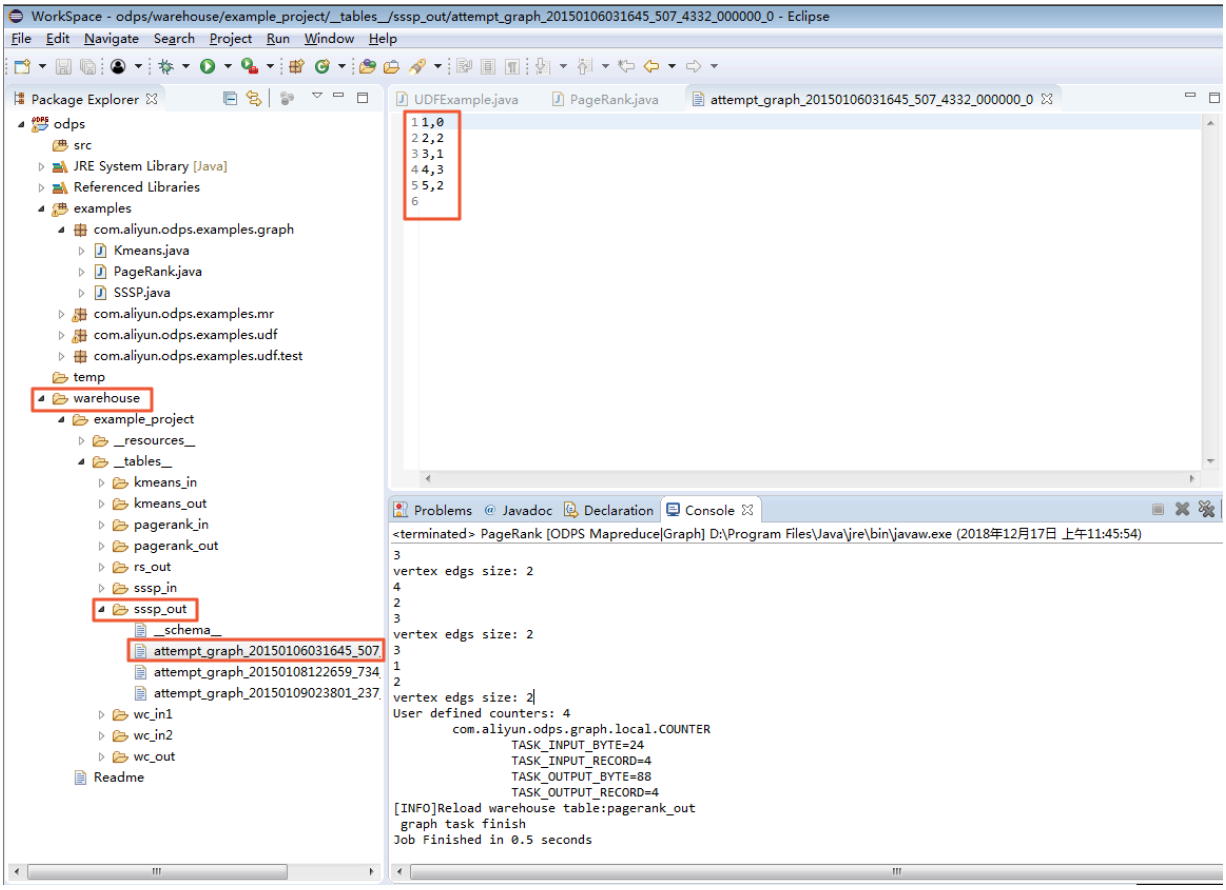
单击后出现对话框，作如下配置。



查看作业运行结果，如下图。

```
Problems  Javadoc  Declaration  Console
<terminated> PageRank [ODPS Mapreduce|Graph] D:\Program Files\Java\jre\bin\javaw.exe (2018年12月17日 上午11:45:54)
3
vertex edgs size: 2
4
2
3
vertex edgs size: 2
3
1
2
vertex edgs size: 2
User defined counters: 4
        com.aliyun.odps.graph.local.COUNTER
                TASK_INPUT_BYTE=24
                TASK_INPUT_RECORD=4
                TASK_OUTPUT_BYTE=88
                TASK_OUTPUT_RECORD=4
[INFO]Reload warehouse table:pagerank_out
 graph task finish
Job Finished in 0.5 seconds
```

可以查看在本地的计算结果，如下图。



调试通过后，用户可以将程序打包，并以Jar资源的形式上传到ODPS，并提交Graph作业。

说明:

- **程序打包过程以及有关本地结果的目录结构介绍，请参见*MapReduce*开发插件介绍。**
- **关于上传Jar资源的详细介绍，请参见资源操作中创建Jar资源部分。**

- 有关提交Graph作业的操作，请参见Graph功能介绍中Jar命令的介绍。

4 相关下载

本文将为您提供在使用MaxCompute过程中，可能用到的相关工具及插件的下载地址。

- SDK下载信息：如果您使用Maven，可以从Maven库中搜索odps-sdk，获取不同版本的Java SDK。
- 新版客户端：点击此处即可下载新版客户端。
- Eclipse开发插件：点击此处即可下载 Eclipse开发插件。
- IntelliJ开发插件：IDEA工具点击此处，Studio插件点击此处即可下载 IntelliJ开发插件。
- JDBC：MaxCompute提供开源JDBC，您可以在GitHub下载JDBC，也可以在云栖社区查看发布信息或提问。
- PHP SDK：您可以在GitHub下载。