



# DataWorks Function Studio

文档版本: 20220112



### 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。

- 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。
- 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。
- 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
⚠ 危险	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。	⚠ 危险 重置操作将丢失用户配置数据。
⚠ 警告	该类警示信息可能会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 时间约十分钟。
〔) 注意	用于警示信息、补充说明等,是用户必须 了解的内容。	大) 注意 权重设置为0,该服务器不会再接受新 请求。
? 说明	用于补充说明、最佳实践、窍门等,不是 用户必须了解的内容。	⑦ 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在 <b>结果确认</b> 页面,单击 <b>确定</b> 。
Courier字体	命令或代码。	执行    cd /d C:/window    命令,进入 Windows系统文件夹。
斜体	表示参数、变量。	bae log listinstanceid
[] 或者 [alb]	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。	switch {act ive st and}

# 目录

1.概述	05
2.新建工程	06
3.UDF开发	80
4.UDF调试	09
5.发布UDF	14
6.MapReduce功能开发	17
7.Git版本管理	20
8.协同编辑	22
9.单元测试	24
10.全文搜索	27
11.自动代码生成	30

### 1.概述

Function Studio是由阿里云自主开发的、面向函数开发场景的Web项目代码编辑开发工具,是DataWorks开发平台的重要组成部分。

基于底层创新性的支撑架构, Function Studio占有资源少、支持高并发,方便、灵活且高效。

Function Studio提供语法高亮、代码自动补全、智能纠错和语法错误提示等功能,并支持在线开发、在线调试、多人协同编辑、一键发布UDF资源和函数至DataWorks。

#### 功能概述

- 支持MaxCompute Java UDF函数的编辑,可以编译和一键发布至DataWorks。
- 支持工程下的文件和文件夹对象的各种管理操作。
- 提供上下文相关的智能编辑器。支持智能化的多Java文件同时编辑,支持查找定义、查找引用、智能提示 代码补全、语法关键字高亮和实时语法错误提示等功能。
- 支持UDF、UDAF和UDTF等函数模板,支持自动发布资源和函数至DataWorks的业务流程中,提升UDF函数 的开发效率。
- 开发环境集成了提交、推送等常见的Git操作,实现了代码文件的版本管理。
- 支持从DataStudio一键跳转至Function Studio, 查看UDF函数的源代码, 方便在线进行UDF的维护和管理。

#### 版本历史

发布版本: Function Studio 1.0

- 发布日期: 2018年12月11日
- 发布范围: 华北2(北京)、华东2(上海)、华南1(深圳)和华东1(杭州)
- 发布内容:
  - 全新推出支持在线开发UDF (Java) 的IDE产品。
  - 支持一站式开发UDF项目,编译发布UDF资源或函数。
  - 函数或资源发布后,您可以进入Function Studio页面,对其进行维护或二次开发。
  - 支持Java的高级编辑功能:代码提示、跳转和重构。
  - 。 支持Git的各项功能。
  - 支持在线调试,支持运行或调试模式下的热部署。

#### 未来发展

后续Function Studio将支持Python等更多语言,支持实时计算等更多平台上的函数开发场景。

# 2.新建工程

本文将为您介绍如何新建和管理工程。

#### 您可以新建代码工程和导入Git 工程。

😑 👘 DataWor	n DataWorks   Function Studio							
三								
➡ 横板空间	欢迎来到 Function S	tudio						
		同新建代码工程	(小 导入Ciu工程					

#### 新建代码工程

如果您需要进行UDF代码开发,则可以创建代码工程。Function Studio为您提供2种运行环境的代码模板,您可以根据自身需求进行选择。

≡	ntaWor	ks   Function Studi					
•	三 工作空间 模板空间	工作空间 > 新建项目	3				
		* 工程名:	请输入工程名称,英文字符开头,只能包含	[程名称,英文字符开头,只能包含数字、英文字符、			
		* 工程描述:	程描述: 请输入工程描述				
		* 选择运行坏 境:	udfpython UDFPython Project		udfjava UDFJava Project		~
		提交					

- 1. 在工作空间页面, 单击新建代码工程。
- 2. 在新建项目对话框中, 输入工程名和工程描述, 并选择运行环境。
- 3. 配置完成后,单击提交。

### 导入Git工程

如果您已经有Git代码,可以直接导入Git代码创建工程。

● Function Studio支持直接导入Git工程, 仅支持HTTP格式(支持将SSH方式转化为HTTP方式)。

● 您在导入Git工程前,需要录入当前用户的用户名、Email、对应Git服务商的SSH等基本信息。如果没有设置,会报错并弹出设置对话框,引导您进行设置。

$\bigcirc$	Fx Function S	tudio	~				
¢	三 工作空间	工作空间 > 新建项目 新建项目					
ŵ	模板空间						
		模板工程 代码	工程 导入GIt工程				
				设置			×
				SSH Key			
			请输入工程名称,英文-	Git Config	* User Name:	user name	
				偏好设置	* Email:	email	]
						保存	

您也可以进入工程的编辑页面,单击顶部菜单栏中的**设置**,对SSH Key、Git Config和偏好设置进行修改。

1. 在工作空间页面, 单击导入Git工程。

2. 在新建项目对话框中, 输入Git 地址、工程名和工程描述, 并选择运行环境。

3. 配置完成后,单击提交。

#### 工程列表

您可以在工作空间页面查看创建的工程。

直接单击相应的工程名称,即可进入工程编辑页面。您也可以单击相应工程下的**创建模板**,通过工程创建模板。

Function Studio对工程可以进行部署的版本管理,单击相应工程下的管理,即可进入部署版本管理页面。

### 3.UDF开发

新建工程完成后会自动生成一个框架代码,支持新建UDF、UDAF和UDTF,本文将以新建UDF为例为您介绍如 何进行开发。

1. 选择**新建 > UDF**。



2. 在弹出框中输入类名, 单击确认, 即可自动生成框架代码。



3. 根据自身需求修改evaluate方法中的内容,来实现UDF的开发。

### 4.UDF调试

UDF调试包括UDF(仅支持Java)、UDAF和UDTF的调试。

### UDF调试(目前仅支持Java)

1. 新建main函数。

Function Studio目前支持通过main函数调用UDF,进行在线调试。



2. 设置Debug配置。

单击右上角的config,进入配置页面。

Run/Debug Configurations			×
+ ×	Name: Application		
Application	* Main class: 👔	com.alibaba.dataworks.udf.Lower	
	VM options:	com.alibaba.dataworks.mapred.WordCount	
	Program arguments:	com.alibaba.dataworks.mapred.StringUDTF	
	Environment Variables:	✓ com.alibaba.dataworks.udf.Lower	
	JRE:	1.8 - SDK	
	PORT:		
	机器:	2vCPU, 4G内存 ~	
	开启HOTCODE:	○ 是 • 否	
		Cancel Apply OK	

您只需要选择新建的main函数,其他的信息都是自动生成的。

配置	说明
Main class	选择需要调试的入口函数,必填项。

配置	说明
VM options	需要启动的JVM参数,非必填项。
Program Variables	启动参数,非必填项。
JRE	目前仅支持JDK1.8。
PORT	需要开放的HTTP端口,UDF/UDAF/UDTF类的项目此配置为非必 填项。
机器	机器规格。
开启HOT CODE	是否需要支持热部署。

#### 3. 启动Debug。

在Lower的evaluate函数上断点后,启动Debug。



启动成功后即可进行正常的调试。可通过step into进入到UDF方法内,并查看变量值。



#### UDAF调试

UDAF的调试需要自己构造相关数据,并且使用warehouse来模拟MaxCompute的数据。warehouse下会保存相关表的Schema以及Data,然后编写相关测试的main函数。



初始化warehouse后,调用相关的UDAF进行测试。

n	MaxCompute工程	<b>⋪</b> ≔ _	schema_ ×	≣ data :	× 4 StudioUDAF	Test.java ×				
	test1 (j)			ss StudioUDAF	Test {					
છ	▼ src									
8	▼ main			static void m	main( <mark>String</mark> [] args)	throws Except	tion {			
	▼ iava									
	▼ com alibaba dataworke		Те	stUtil.initWar	rehouse();					
	- manual data works		si	<pre>mpleInput() ;</pre>						
	✓ mapred		24 inputFromTable();							
	WordCount.java		3							
	✓ udaf		public	static void	impleInput() throw	s Exception {				
	🛔 StudioUDAFTest.java		Ba	seRunner runne	er = new Aggregator	Runner(null.				
	🛓 StudioUDAF.java			"com.alibaba	.dataworks.udaf.St	udioUDAF");				
	▶ udf		ru	nner.feed(new	<pre>Object[] {"one", "</pre>	one"}).feed(ne	<pre>w Object[] {"three", "three"})</pre>			
	▶ udtf			.feed(new Ot	<pre>oject[] {"four", "f</pre>	our"});				
	4. TestUtil.iava		Li	<pre>st<object[]> 0</object[]></pre>	out = runner.yield(	0;				
	target		As	sert.assertEqu	<pre>uals(1, out.size())</pre>					
			Sy	stem.out.print	:ln("size:" + out.s	<pre>size());</pre>				
	• warenouse		As	sert.assertEqu	<pre>lals(24L, out.get(0 lals(24L, out.get(0)))</pre>	))[0]); +(0)[0]);				
	studio_project		, sy	stem.out.print	:in("get:" + out.ge	et(0)[0]);				
	<ul> <li>demo_project</li> </ul>		,							
	▼ _tables_		public	static void i	inputFromTable() th	rows Exception	1.4			
	✓ demo_01		Ba	seRunner runne	r = new Aggregator	Runner(null,				
	≣ data			"com.alibaba	a.dataworks.udaf.St	udioUDAF");				
	≣ _schema_		St	<pre>ring project =</pre>	<pre>studio_project";</pre>					
	s pom.xml		St	<pre>ring table = '</pre>	'wc_in2";					
			St	<pre>ring[] partiti</pre>	<pre>ions = new String[]</pre>	{"p2_1", "p1_	_2"};			
			St	ring[] columns	<pre>s = new String[] {"</pre>	'colc", "cola"]				
			In	putSource inpu	itSource = new Tabl	leInputSource(p	project, table, partitions, columns);			
			Ob	ject[] data;						
				ruppon food	(dota)	:Row()) != null				
\$			}	runner. reed	(uata);					

### UDTF的调试

UDTF与UDAF的调试一样,初始化工程中已存在UDTF的测试类,直接执行该类,即可模拟真实数据对UDTF 进行调试。

udftest111 (j)	
▼ src	
🔻 java	
▼ com.a	libaba.dataworks
▶ map	bred
▼ uda	f
	StudioUDAFTest.java
	StudioUDAF.java
▼ udf	
	LowerTest.java
	Lower.java
▼ udtf	f
	StudioUDTF.java
	StudioUDTFTest.java
🛓 Tes	stUtil.java

单击**执行**,如果程序没有抛异常,则表示运行通过。

# 5.发布UDF

本文为您介绍如何发布资源或函数至DataWorks开发环境。

### 背景信息

Function Studio仅支持发布资源或函数至DataWorks开发环境。如果您需要发布资源或函数至生产环境,请完成本文的操作后,单击**数据开发**页面右上方的**任务发布**。详情请参见发布任务。

### 进入Function Studio工程

- 1. 进入数据开发页面。
  - i. 登录DataWorks控制台。
  - ii. 在左侧导航栏, 单击工作空间列表。
  - iii. 选择工作空间所在地域后,单击相应工作空间后的进入数据开发。
- 2. 单击左上方的 Sala A 法择全部产品 > 数据开发 > Function Studio, 默认进入工作空间页面。
- 3. 在我的工程区域,单击相应的工程卡片,进入该工程的编辑页面。

#### 提交资源至DataWorks开发环境

1. 在工程的编辑页面,鼠标悬停至<и图标,单击提交资源至DataWorks开发环境。



2. 在提交资源至DataStudio开发环境对话框中,选择目标业务空间、目标业务流程,并输入资源。

? 说明

- 资源名称的长度为2~100个字符,必须添加资源类型的后缀,且只能包含数字、字母、小数点(.)和下划线(\_)。
- 默认选中如果已经存在强制更新,您可以根据自身需求设置。
- 3. 单击确认。
- 4. 在输出页签,确认显示发布至DataWorks成功后,单击资源链接,查看提交的资源。

	輸出	
		2020/06/09 15:27:45 [INFO ]:: ##################################
		2020/06/09 15:37:45 [INFO ]:: # STEP PASS
		2020/06/09 15:37:45 [INFO ]:: [Updaterrojectbulldtatus] -> Invoked
		S 10041 S Received S Alera Average Spect 1 ame 1 ame Curters
		100 272 0 255 100 17 8967 597
		2020/06/09 15:37:45 [INFO ]:: [UpdateFrojectBuild] => Invoked
		* Total * Received * Xferd Average Speed Time Time Time Current
		Dload Upload Total Spent Left Speed
		\r\n\r\n正在获取 DataWorks 发布任务状态: \r\n
		发布至DataWorks成功! \r\n
		资源: https://ide-cn-shanghai
-	- I	
*	đ	OUT 🔆 DEBUG 🗮 PROBLEM 📧 Terminal 🕴 Version Control

提交函数至DataWorks开发环境

- 1. 在工程的编辑页面,鼠标悬停至<и图标,单击提交函数至DataWorks开发环境。
- 2. 在提交函数至DataStudio开发环境对话框中,配置各项参数。

提交函数至 Data Studio 开发环境	×
目标业务空间:	
目标业务流程:	
×	
资源:	
) O.jer	27/100
额外资源 (可选) :	
请输入额外资源名称	
类名:	
com. f.Lower	
函数名:	
Lower	5/100
✔ 如果已经存在强制更新	
	<b>确认</b> 取消
参数	描述
目标业务空间	发布函数的目标工作空间。
目标业务流程	发布函数的目标工作空间下的目标业务流程。

参数	描述
资源	资源名称的长度为2~100个字符,必须添加资源类型的 后缀,且只能包含数字、字母、小数点(.)和下划线 (_)。
额外资源(可选)	可选参数。
类名	实现UDF的主类名。
函数名	UDF函数名,即SQL中引用该函数所使用的名称。
如果已经存在强制更新	默认选中。

3. 单击**确认**。

4. 在输出页签,确认显示发布至DataWorks成功后,单击函数链接,查看提交的函数。

# 6.MapReduce功能开发

本文将以WordCount为例,为您介绍新建工程后,如何在DataWorks进行MapReduce开发。

#### 新建工程

- 1. 登录DataWorks控制台,单击相应工作空间后的进入数据开发。
- 2. 单击左上角的图标,选择全部产品 > Function Studio。
- 3. 单击工作空间页面的新建代码工程。
- 4. 在新建项目对话框中,输入工程名和工程描述,并选择运行环境(本文以UDFJava Project为例)。
- 5. 配置完成后,单击提交。

### 开发项目

进入*src/main/java/com.alibaba.dataworks/mapred*下,双击**WordCount.java**,打开WordCount的 MapReduce示例代码。



示例代码可以对输入表中的单词进行次数统计,将统计结果写入输出表中。输入输出分别是两个表,详情请参见WordCount示例。

#### 调试项目

目前Function Studio不支持MapReduce项目进行Debug,需要发布代码至DataWork开发环境后,跳转至 DataWorks中进行逻辑验证。

⑦ 说明 目前Function Studio仅支持编码和编译打包两个功能。

### 发布项目

- 1. Function Studio编译打包代码发布至Data Studio开发环境。
  - i. 鼠标悬停至 🖌 ,单击提交资源至Data Studio开发环境。
  - ii. 在提交资源至Data Studio开发环境对话框中,选择目标业务空间和目标业务流程,并输入资源名称(本文以WordCountDemo\_1.0.0.jar为例)。
  - iii. 默认勾选如果已经存在强制更新,单击确认。
  - iv. 发布成功后, 在**输出**中获取资源在DataStudio中的定位链接。单击该链接即可跳转至**数据开发 >** 上传资源页面。

輸出	
Ē	
2	019/08/29 23:06:42 [INFO ]:: <b>##################################</b>
2	019/08/29 23:06:42 [INFO ]:: # STEP PASS
2	:019/08/29 23:06:42 [INFO ]:: ##################################
2	019/08/29 23:06:42 [INFO ]:: [UpdateProjectBuildStatus] => Invoked
	8 Total & Received & Xferd Average Speed Time Time Time Current
	Dload Upload Total Spent Left Speed
1	
2	019/08/29 22-06-42 [INFO ] [IndateProjectBuild] => Involed
-	
	* lobal * Received * Alera Average spece lime lime lime time Current
	Dioad Upicad Total Spent Left Speed
I	E在获取 Data Studio 提文任务状态:
	···
1	
3	を源: https://ide-cn-shanghai.data.aliyun.com/#command=%7B%22 D

- 2. 在DataWorks中创建MapReduce节点进行测试。
  - i. 在数据开发页面, 打开相应的业务流程。
  - ii. 右键单击MaxCompute,选择新建 > ODPS MR,详情请参见创建ODPS MR节点。
  - iii. 在ODPS MR节点的编辑界面, 创建测试表并添加测试数据。

```
--创建输入表
create table if not exists wc_input (key string,value string);
--创建输出表
create table if not exists wc_out (key string,cnt bigint);
--插入数据至输入表
insert overwrite table wc_input values ('hello','odps');
```

iv. 右键单击上传的资源,单击**引用资源**,在右侧的ODPS MR节点编辑界面自动显示 --@resource\_re

ference{"WordCountDemo\_1.0.0.jar"} .



v. 输入上传的JAR包的 jar -resources 和 -classpath 信息, 示例如下。

jar -resources WordCountDemo\_1.0.0.jar -classpath ./WordCountDemo\_1.0.0.jar com.alibaba.dataworks.mapred.WordCount wc\_inpu t wc out

vi. 单击工具栏中的 💽。

- 3. 单击节点编辑页面右侧的调度配置,即可进入调度配置页面,详情请参见调度配置模块。
- 4. 完成调度配置后,单击工具栏中的图,提交(提交并解锁)到开发环境。
- 5. 发布节点任务, 详情请参见发布任务。
- 6. 在生产环境进行测试,详情请参见查看周期任务。

# 7.Git版本管理

Function Studio支持新建的工程和Git进行关联,关联后即可进行常规的Git操作。本文将为您介绍如何对工程进行Git版本管理。

Git是一个开源的分布式版本控制系统,可以有效、高速地处理项目的版本管理。

- 1. 登录DataWorks控制台,单击相应工作空间后的进入数据开发。
- 2. 单击左上角的图标,选择全部产品Function Studio
- 3. 在工作空间页面,单击需要进入的Git工程。

⑦ 说明 仅Git工程支持Git版本管理,新建工程的具体操作请参见导入Git工程。

4. 单击左侧导航栏中的Git版本管理图标。



- 单击文件后的+图标,即可暂存更改。
- 单击√图标,即可进行Commit和Commit&Push操作。
- 单击...图标,即可进行**拉取**和推送操作。
- 5. 单击页面底部的master,可以切换和创建分支。



### 8.协同编辑

Function Studio支持邀请多人协同编辑同一工程的同一文件,提高开发效率。本文将为您介绍如何邀请协作者,以及协作者的权限说明。

### 邀请协作者

- 1. 登录DataWorks控制台,单击相应工作空间后的进入数据开发。
- 2. 单击左上角的图标,选择全部产品 > Function Studio。
- 3. 在工作空间页面,单击需要进入的工程。
- 4. 双击打开需要分享的项目,单击右侧的Share。

6	Free Function Studio			a intervisione	
1	☆ 工程 文件 編編 版本	5. 查看 调试 设置 发布 模板 帮助 反馈		Edit Config 🗸 🕨 🌺	
đ	MexCompute工程 demo ④	<b>4</b> () settings json × 1 := ]{	× Share	< <p>&lt; 分享链接 8.邀请</p>	R
¥	> .alicode > .settings > src	2 "_comment": This is a conf 3 "version": 1.0, 4 "charset": UTI-0, 5 ⊠ "files": {	<b>协作者</b> ●	权限 iewing所有者	ntime
	> target	6			Share
	» pom.xml				

5. 单击面板右上角的邀请, 在邀请协作者对话框中, 选择用户名和权限。



#### 6. 单击**确认**。

邀请完成后,协作者可以同时编辑同一工程的同一个文件。

#### 权限说明

在协同编辑的过程中,参与的协作者包括以下3种权限:

- 所有者:所有者是项目的创建者,无法变更。所有者可以邀请其他开发者加入项目,也可以移除其他的协作者。
- 读写权限: 拥有读写权限的协作者, 可以查看和编辑项目中的所有文件。
- 只读权限: 拥有只读权限的协作者, 仅支持查看项目中的文件, 无权限编辑文件。

### 9.单元测试

Function Studio支持单元测试的运行,本文为您介绍如何进入单元测试、运行单元代码和展示运行结果。

#### 背景信息

单元测试(Unit testing)是指对软件中的最小可测试单元进行检查和验证,其中单元是人为规定的最小的被测功能模块。例如,C语言中的单元指一个函数,Java中的单元指一个类。

#### 进入单元测试

- 1. 进入数据开发页面。
  - i. 登录DataWorks控制台。
  - ii. 在左侧导航栏, 单击工作空间列表。
  - iii. 选择工作空间所在地域后,单击相应工作空间后的进入数据开发。
- 2. 单击左上方的 图标,选择 2部产品 > 数据开发 > Function Studio。
- 3. 在工作空间页面,单击需要进入的工程。如果您需要新建工程,请参见新建工程。

🜀 🗲 Function	Studio		
三 ● 工作空间 ● 模板空间	欢迎来到 Function St	tudio	
	C 新建模板工程	<b>⊡</b> 新建代码工程	⑦ 导入Git工程
	Lest	demo	demo
	几秒前更新 © 普理员 创建模板 管理 删除	20天前更新 10 答理员 的建模板 答理 MMAA	2个月前更新 ② 管理员 创建模板 管理 副联

4. 双击进入src/test/java下的单元测试类。



#### ? 说明

- 单元测试类需要写在*src/test/java*目录下。如果不在该目录下,将无法被正常识别为Java测 试类。
- o Java类创建完成后,在对应的测试用例方法上添加 org.junit.Test 的 @Test 注解即 可。

### 运行单元测试代码

单击右上方的 图标,运行单元测试代码。

ធ	工程文件编辑版本查看调试	设置 发布 模板 幕助 反馈	📲 Edit Config 🗸 ▶ 斄 🔳
n	工程 這	🛓 Main.java 🛪 👍 SampleTest.java 🛪	_
"	TestUT (j)		Runti
	▼ src		Surgenza and
	▼ main	9 public class SampleTest {	
	▼ java		
	✓ com.alibaba.dataworks		
	🛓 Main.java	Run Test	hare
	▼ resources	12 public void testPass() { 13 System out println("hello world!");	L 1
	🛓 application.properties	14 }	
	▼ test		
	▼ java		D2
	com.alibaba.dataworks	Run Test	<u>a</u>
*		17 public void testrail() ju 18 Assert.assertEquals(true. false):	
输出			~
-183 []			
	.pom 2019-01-12 22:50:21.903295 :: [INFO] Download	ed from nexus-tbmirror: http://100.67.148.246/repository/maven-public/org/apache/maven/maven-to	n-1.0.
	pom (3.4 kB at 379 kB/s) 2019-01-12 22:50:21.908358 :: [INFO] Download	ing from nexus-thmirror: http://100.67.148.246/repository/maven-public/org/apache/maven/shared/	en-sha
	red-utils-0.1.pom		
	2019-01-12 22:50:21.915237 :: [INFO] Download ed-utils-0.1.pom 44.0 kB at 674 kB/s)	ed from nexus-tbmirror: http://100.67.148.246/repository/maven-public/org/apache/maven/shared/m	n-shar
	2019-01-12 22:50 1 916662 :: [INFO] Download	ing from nexus-tbmirror: http://100.67.148.246/repository/maven-public/org/apache/maven/shared/	/maven
	2019-01-12 22:50:21.924201 :: [INFO] Download	ed from nexus-tbmirror: http://100.67.148.246/repository/maven-public/org/apache/maven/shared/m	naven-
	shared-components-18.pom (4.9 kB at 705 kB/s)	ing from navig theirvor, http://100.67.148.246/wapogitory/mayon_public/org/anache/mayon/shaved/	1/2011
	en-shared-incremental-1.1.pom	ing from nexus-comfront incpr//ivv.o/ii40.z40/repository/maven-public/org/apache/maven/shared/	. 1/ mav
	2019-01-12 22:50:21.933468 :: [INFO] Download	ed from nexus-tbmirror: http://100.67.148.246/repository/maven-public/org/apache/maven/shared/m	1/mave
	2019-01-12 22:50:21.934966 :: [INFO] Download	ing from nexus-tbmirror: http://100.67.148.246/repository/maven-public/org/apache/ma	3
	-shared-components-19.pom	🔁 Running Tes	ts
451		II Version Control	
	PROBLEM M Ierminal	P Version Control	
- (	Running tests		

序号	说明
0	展示单元测试类的运行日志。
2	运行中的指示窗口。
3	单击该按钮,即可终止任务的运行。

#### 展示运行结果

任务运行成功后,您可以在页面下方的输出中,查看运行结果。

单击每个Run Test后的成功或失败图标,即可在右侧查看详细报告。



# 10.全文搜索

Function Studio支持全文搜索功能,即根据关键词在整个工程内查找相关内容。本文将为您介绍如何在 Function Studio中进行全文搜索。

- 1. 登录DataWorks控制台,单击相应工作空间后的进入数据开发。
- 2. 单击左上角的图标,选择**全部产品 > Function Studio**。
- 3. 在**工作空间**页面,单击需要进入的工程。如果您需要新建工程,请参见新建工程。

S Fx Function Studio ✓			
三 ● 工作空间 ● 模板空间	欢迎来到 Function St	udio	
	C: 新建模板工程	☑ 新建代码工程	④ 导入Git工程
	】 <b>我的工程</b> Q、 谢翰入		
	test	demo	demo
	几秒前更新 ◎ 管理员 创建模板 管理 ■■除	20 天前更新 ② 管理员 创建模板 管理 副版	2 个月前更新 ② 管理员 创建模板 管理 副除机

4. 单击编辑,选择全文搜索。



5. 在全文搜索对话框中,筛选搜索条件,选择需要的文件。

全文搜索 🔽 大小写匹配 📄 单词匹配 📄 正则匹配 📄 文件后缀	§ 请选择 ✔
Q tes	13 matches in 6 files 🛛 🛞
项目 模块 目录	
logalirefupda <mark>tes</mark> = true	
fetch = +refs/heads/*:refs/remo <mark>tes</mark> /origin/*	config 7
public void testDirectMode() throws Exception {	
/*lp2Region <mark>tes</mark> t = new lp2Region();	
String result = test.evaluate("14.136.107.248");	
public void testRedirectMode1() {	
public void <mark>tes</mark> tRedirectMode2() {	
public void testRedirectMode1Mode2() {	
.git/config	
2 repositorytormatversion = 0	The second s The second secon
3 filemode = true	Line and the second sec
4 logaliretupdates = true	-
f [remote origin ]	ud
<pre>7 fetch = +refs/heads/*:refs/remotes/origin/*</pre>	
8 [branch "master"]	
9 remote = origin	
10 merge = refs/heads/master	
11 [user]	
1) name - nenomin	使爆哭肉打开

○ 全文搜索支持**大小写匹配、单词匹配、正则匹配**,并支持根据**文件后缀**查找指定的文件类型。

模板	
.al 全文搜索 🔲 大小写匹配 📄 单词匹配 📄 正则匹配 🔽 文件后缀	请选择
jun: Q search	*.as
ic · 项目 模块 目录	*.properti
	*.mxml
s D	*.css
uni .	*.js
	*.java
	*.xml
	*.jsp
	*.html

○ 选择匹配条件后,您可以根据项目、模块和目录进行查找。



6. 单击编辑器内打开。

# 11.自动代码生成

本文将为您介绍Function Studio中的自动代码生成功能。

自动代码生成功能支持生成基础函数。例如,新建一个类,可以自动生成该类的构造函数、Getter函数、 Setter函数、该类所继承父类的Override方法、所要实现的接口方法等代码。

#### 进入工程

- 1. 登录DataWorks控制台,单击相应工作空间后的进入数据开发。
- 2. 单击左上角的图标,选择**全部产品 > Function Studio**。
- 3. 在**工作空间**页面,单击需要进入的工程。如果您需要新建工程,请参见新建工程。

🜀 🗲 Function	Studio		
=			
□ 工作空间	水油本型 Eunction St	udio	
☆ 模板空间	从 迎 未 主 j Function St	udio	
		n l	
	đ;	Ð	
	新建模板工程	新建代码工程	导入Git工程
	我的工程		
	Q. 请输入 搜索		
	test	aemo	aemo
	♥ 管理员 创建模板 管理 ■除	管理员 创建模板 管理 删除	◎ 管理员 创建模板 管理 删除

4. 双击打开相应的Java类。

### 功能入口

目前, Java有以下2种代码生成入口:

• 鼠标右键单击代码区域,选择Generate。



● 通过快捷键Ctrl+M进入。

### Constructor

1. 进入Generate Code面板,选择Constructor。



2. 选择构造函数中要包含的字段,即可生成包含相应字段初始化语句的构造函数。



**Getter Add Setter** 

#### 您可以参见Constructor的生成方式,来生成Getter和Setter函数。



⑦ 说明 如果该Java类没有任何字段,或者该Java类已经被lombok的@data注解覆盖,则没有上图中的三个选项,因为此时该类无需生成Getter或Setter函数。

### **Override Methods**

选择生成Override Methods的一级菜单后,在二级菜单中会罗列所有可以Override的方法。



#### 选择后即可生成相应的方法。

.≉ ⊞	👙 TestUtil.java 🗙 👙 ILower.java 🗙 🁙 Lower.java 🗙
	<pre>1 package com.alibaba.dataworks.udf;</pre>
	<pre>3 import com.aliyun.odps.udf.UDF;</pre>
	4 import com.alibaba.dataworks.udf.iLower;
	o public final class lower extends UDE implements Tlower /
	8 private int id;
	9 private String name;
	<pre>11 public Lower(int id, String name) {</pre>
	12 this.id = id;
	13 this.name = name;
	14 }
	17 public void close() throws UDEException {
	18 super.close():
	19 }
	21 @Override
	<pre>22 public String toString() {</pre>
	<pre>23 return super.toString();</pre>
	24 }
	25 26
	20 world interfaceHeight(int id) {
	29 }
	30