Alibaba Cloud

DataWorks APP Studio

文档版本: 20211223



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。	▶ 注意 权重设置为0,该服务器不会再接受新 请求。
⑦ 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 结果确认 页面,单击 确定 。
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 {alb}	表示必选项,至多选择一个。	switch {act ive st and}

目录

1.概述	06
2.App Studio版本历史	10
3.入门教程	11
4.功能介绍	44
4.1. 导航页	44
4.1.1. 工作空间	44
4.1.2. 应用空间	45
4.1.3. 模板空间	54
4.2. 工程管理	55
4.3. 版本管理	59
4.4. 代码编辑	67
4.4.1. 代码编辑概述	67
4.4.2. UT测试	75
4.4.3. 生成代码片段	78
4.4.4. 全文内容搜索	83
4.5. 调试	86
4.5.1. Config配置及启动	86
4.5.2. 在线调试	88
4.5.3. 断点类型	91
4.5.4. 断点及操作	94
4.5.5. 远程调试	98
4.5.6. 终端	99
4.5.7. 热部署	100
4.6. 协同编程	103
4.7. 应用部署	106
4.8. 第三方服务接入	118

4.8.1. 数据服务	118
4.8.2. DataOS API	- 123
4.9. 可视化搭建	147
4.9.1. 可视化搭建概述	147
4.9.2. 基本使用	149
4.9.3. 常用组件	- 155
4.9.4. 代码模式	- 163
4.9.5. DSL语法	163
4.9.6. 全局数据流	- 164
4.9.7. 导航配置	166
4.9.8. 保存、预览、运行和热部署	168
4.9.9. 发布为模板	169

1.概述

App Studio为您提供丰富的前端组件,通过自由拖拽即可快速搭建前端应用。本文为您介绍App Studio的 基本概念和产品优势。

App Studio是一款数据产品的开发工具,您无需下载安装本地IDE或配置维护环境变量,只需要一个浏览器即可编写、运行和调试应用程序,体验和本地IDE一样的编程效果,并且可以在线发布应用。

产品优势

App Studio的核心优势如下:

• 随时随地开发

您无需下载安装本地IDE和配置维护环境变量,只需要一个浏览器,即可在办公室、家或任何可以连接网络的地方,进行您的数据开发工作。

• 功能完备的编辑器

App Studio提供一个基于浏览器的编辑器,您可以使用它轻松地编写、运行和调试项目。当您输入代码时,App Studio会提供智能提示、补全代码和修复建议等功能。您还可以查找方法的引用和定义,自动生成代码。



• 在线调试功能

在线调试包括本地IDE所有的断点类型和断点操作,支持线程切换、过滤,支持变量值查看、监视,支持远程调试和热部署。

←	⇒ C _0 7	安全 pre-stu	dio.data.al	iyun.com	(#/							🖈 🖪 🖼 😒 👒 🗷 🗄
6)	App Studio	工程	文件	编辑	版本	查看	调试	设置	帮助)		main 🗸 🕨 🌺 📕
a	工程				👍 Index	Controller.java		pom.xml				项目已经启动,访问:
יים	demo (j)					xml vers</th <th>ion="1.</th> <th>0" encodin</th> <th>a="UTF</th> <th>-8"?></th> <th>- Richard States</th> <th>前端: http://gateway.studio.data.</th>	ion="1.	0" encodin	a="UTF	-8"?>	- Richard States	前端: http://gateway.studio.data.
_						<project th="" x<=""><th>mlns="h</th><th>ttp://mave</th><th>n.apac</th><th>he.org/POM/4.0.0"</th><th>THE REPORT OF</th><th>dsgzuint6k0/8080/</th></project>	mlns="h	ttp://mave	n.apac	he.org/POM/4.0.0"	THE REPORT OF	dsgzuint6k0/8080/
Ŷ	• src					x	mlns:xs	i="http://	www.w3	.org/2001/XMLSchema-instance"	- Contanorum	后端: http://gateway.studio.data.
	▼ main					X	si:sche	maLocation	=" <u>http</u>	://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-	The sector sectors	aliyun.com/pcsoxspre680ac8f9tb
	▼ java					<model< th=""><th></th><th>>4.0.0<th>delVer</th><th></th><th>To and the second second</th><th>dsgzuint6k0/7001</th></th></model<>		>4.0.0 <th>delVer</th> <th></th> <th>To and the second second</th> <th>dsgzuint6k0/7001</th>	delVer		To and the second second	dsgzuint6k0/7001
	- con	n.alibaba.demo				<packa< th=""><th>ging>ja</th><th>r<th>ng></th><th></th><th>Distante -</th><th>har</th></th></packa<>	ging>ja	r <th>ng></th> <th></th> <th>Distante -</th> <th>har</th>	ng>		Distante -	har
	▶ c	ommon				<groupid>c</groupid>	om.allb	aba.demo </th <th>groupi d</th> <th></th> <th>The second real and the second second</th> <th>Ċ</th>	groupi d		The second real and the second	Ċ
	▼ c	ontroller				<artifacti< th=""><th>0>0emo<!--</th--><th></th><th>u> «/vers</th><th></th><th></th><th></th></th></artifacti<>	0>0emo </th <th></th> <th>u> «/vers</th> <th></th> <th></th> <th></th>		u> «/vers			
		api.demo						0-JIIAF JIIOT				
	-	nage				<depen< th=""><th>dencvMa</th><th>nagement></th><th></th><th></th><th></th><th></th></depen<>	dencvMa	nagement>				
		4 IndexContro	ollor iava				ependen	cies>				
			unchjava				<deper< th=""><th></th><th></th><th></th><th></th><th></th></deper<>					
	▼ 5	ervice						groupId> or	g.spri	ngframework.boota/groupIda		
	•	impl						artifactId	>sprin	g-boot-dependencies		
		OssService.jav	va					version>1.	5.12.R	ELEASE		
		PaiApiService.	.java					type>pom </th <th>type></th> <th></th> <th></th> <th></th>	type>			
		Main.java						scope>impo	rt <th></th> <th></th> <th></th>			
	▶ resour	rces					<th>endency></th> <th></th> <th></th> <th></th> <th></th>	endency>				
	ternet						depender	ncies				
*	F turget					<th>doncios</th> <th>anagement></th> <th></th> <th></th> <th></th> <th></th>	doncios	anagement>				
	» pom.xmi						uencies					
输	出 调用堆栈				F	•						
}" on	nto public or	.springframew	ork.http.	ResponseE	ntity <j< td=""><td>ava.lang.Ob</td><td>ject> or</td><td>rg.springf:</td><td>amewor</td><td>k.data.rest.webmvc.RepositorySearchController.headForSearch(org.sp</td><td>ringframework.</td><td>data.rest.webmvc.RootResourceI</td></j<>	ava.lang.Ob	ject> or	rg.springf:	amewor	k.data.rest.webmvc.RepositorySearchController.headForSearch(org.sp	ringframework.	data.rest.webmvc.RootResourceI
2018-	-08-15 14:04:1	19.574 [o.s.d	.r.w.BasePa	thAwareB	landlerMap	oing	- Mapped "{[/profile],methods=[GET]}" onto org.springframework.ht	tp.HttpEntity<	org.springframework.hateoas.Re
sourc	ceSupport> org	.springframew	ork.data.	rest.webm	vc.Prof	ileControll	er.list	llFormsOf	letadat	a()		
2018-	-08-15 14:04:1 iata.rest.webm	19.574 [wc.ProfileCon	troller.n	ain] INFO	o.s.d	.r.w.BasePa	thAware	landlerMap	oing	- Mapped "{[/profile],methods=[OPTIONS]}" onto public org.springf	ramework.http.	HttpEntity org.springframew
2018-	-08-15 14:04:1	19.575 [m	ain] INFO	o.s.d	.r.w.BasePa	thAwareB	IandlerMap	oing	- Mapped "{[/profile/{repository}],methods=[GET],produces=[applic	ation/alps+jso	n */*]}" onto org.springfra
mewor eInfo	rk.http.HttpEr ormation)	ntity <org.spri< td=""><td>ingframewo</td><td>rk.data.r</td><td>est.web</td><td>mvc.RootRes</td><td>ourceInf</td><td>formation></td><td>org.sp</td><td>pringframework.data.rest.webmvc.alps.AlpsController.descriptor(org.</td><td>springframewor</td><td>k.data.rest.webmvc.RootResourc</td></org.spri<>	ingframewo	rk.data.r	est.web	mvc.RootRes	ourceInf	formation>	org.sp	pringframework.data.rest.webmvc.alps.AlpsController.descriptor(org.	springframewor	k.data.rest.webmvc.RootResourc
2018-	-08-15 14:04:1	19.575 [m	ain] INFO	o.s.d	.r.w.BasePa	thAware	IandlerMap	oing	- Mapped "{[/profile/{repository}],methods=[OPTIONS],produces=[ap	plication/alps	+json]}" onto org.springframew
2018-	-08-15 14:04:1	19.576	Ingrramewo.	ain] INFO	o.s.d	.r.w.BasePa	thAware	andlerMap	perone	- Mapped "{[/profile/{repository}],methods=[GET],produces=[applic	ation/schema+i	son]}" onto public org.springf
ramew forma	work.http.Http ation)	Entity <org.sp< th=""><th>oringframe</th><th>work.data</th><th>.rest.w</th><th>ebmvc.json.</th><th>JsonSche</th><th>ama> org.s</th><th>pringfi</th><th>amework.data.rest.webmvc.RepositorySchemaController.schema(org.spr</th><th>ingframework.d</th><th>ata.rest.webmvc.RootResourceIn</th></org.sp<>	oringframe	work.data	.rest.w	ebmvc.json.	JsonSche	ama> org.s	pringfi	amework.data.rest.webmvc.RepositorySchemaController.schema(org.spr	ingframework.d	ata.rest.webmvc.RootResourceIn
2018-					0.8.j	.e.a.Annota	tionMBea	nExporter		- Registering beans for JMX exposure on startup		
2018-			m	ain] INFO ain] INFO	s.b.c com.a	.e.t.Tomcat libaba.demo	.Main	iServletCo	itainer	- Tomcat started on port(s): 7001 (http) - Started Main in 4.51 seconds (JVM running for 5.873)		
Ш												
👙 D	DEBUG 📃 PR	OBLEM										

• 多功能终端

开发者可以直接进入运行环境,目前的运行环境基于Cent OS作为基础镜像来构建。终端可以支持任意的bash命令,包括VIM等具有交互功能的命令。

2002002020		Tibolr oldrabituditaneworkrottuoder3
✓ main		import org.springframework.web.bind.annotation.GetMapping;
resources		
→ java		
🔻 com alibaba /	tataworka B	
Common		* @date 2018-00-15
► api		@Controller
- page		public class IndexController (
s Index	Controller.java	A REAL PROPERTY AND A REAL PROPERTY AND A REAL PROPERTY AND A REAL PROPERTY.
Service		<pre>guettapping(value = 1 / ", "/index.ntm", "/index.ntm") while States index[Middl and Nil </pre>
		public string indextwoder model/t
Terminal		
drwar-ar-a 1 admin admin 4	1096 9月 5 15:19 plugins	
drwar-ar-a 1 admin admin 4	1096 9月 18 17:02 source	
(admin@posdaily588cebe0jf)	rpsumn9ps-756cbc75bb-rbwhf -	
\$11		
total 28 drawn-wr-w 1 admin admin d	1096 9 B 18 17:02 month	
drwar-ar-a 1 admin admin	1096 9月 10 21:00 bin	
drwar-ar-a 1 admin admin 4	1096 9月 10 21:00 conf	
drwxr-xr-x 3 admin admin 4	1096 9月 18 17:03 demo	
dreat-ar-a 1 admin admin of	1096 979 18 17101 10gs	
drwar-ar-a 1 admin admin 4	1096 9月 18 17:02 source	
<pre>[admin@posdaily588cebe0jf] \$vi conf/nginx.conf</pre>	trpauwan9ps-756cbc75bb-rbwhf _]	
[admin@pcsdaily588cebe0jf] Stop	krpsuvan9ps-756cbc75bb-rhehf _]	
DEBUG ROBLEM	Terminal	

● 协同编辑

您和您的团队成员可以借助App Studio共享开发环境,进行团队协作编程。目前支持8人同时在线编辑同一个工程的同一个文件,提高工作效率。后续协同编辑组建还会支持聊天、弹幕、代码批注、视频等功能,让团队合作更加轻松。



● 插件体系

App Studio支持业务、工具和语言等插件:

- 您可以根据业务在App Studio上定制任意的菜单栏,在界面增加业务入口。
- 。 您可以定制专属于您业务的项目管理过程、工程类型和模板。
- 您可以开发通用工具,例如Git功能增强、代码规则扫描、快捷键、编辑功能增强、代码片段等集成到 App Studio中。
- 您可以通过语言插件扩展App Studio支持的语言,满足自身需求的同时帮助App Studio建设服务更多的语言用户。
- 可视化搭建

App Studio提供丰富的组件,并且深度打通数据服务和数据开发。您能且仅能在App Studio中调用 DataWorks的OpenAPI,并且通过可视化拖拽、配置的方式快速搭建前端应用,真正实现零代码开发Web 应用。

● 丰富灵活的项目管理

App Studio提供了丰富多样的模板工程,您可以基于模板工程进行再次开发,节省人力提高效率。您也可以将您自己的工程保存为模板,供您自己后续开发使用,或分享给其他人使用。

2.App Studio版本历史

本文将为您及时同步App Studio的版本更新。

App Studio 1.0

发布日期: 2019年4月3日

发布内容:在Function Studio的基础上实现一个能做应用发布的IDE,核心功能如下所示:

● LSP语言服务

支持语法高亮、智能提示、智能补全、智能诊断、查找定义、查找引用等本地编辑体验。

● 支持Debug功能

具有本地IDE所有的断点类型和断点操作,支持线程切换、过滤,支持变量值查看、监视,支持远程调试、热部署和多功能终端。

- 支持基于接口定义的前后端开发
 前端可视化的组件可以通过配置后端接口进行前端联动。
- 前端可视化搭建

您可以通过拖拽组件搭建前端应用,具有很大的灵活性,支持没有前端经验的用户开发前端应用。支持前 端模板管理,支持可视化模式和代码模式互转,可兼顾开发者更高的开发需求。

- 具备代码版本管理能力。
- 可以在线部署和实时预览应用。
- 具备协同编辑功能

支持8人同时在线编辑同一个工程同一个文件。

- 支持用户自定义工程模板,提供强大的工程管理能力。
- 具有插件开发和集成能力

支持用户开发插件定制专属于业务的IDE(计划在App Studio1.1版本和插件市场一起发布)。

- 支持Java、JS、CSS、HTML和Python多种语言。
- 支持UT自动生成和运行。
- 可设置项目为可分享状态,并通过链接分享给他人(计划在App Studio1.1版本和插件市场一起发布)。
- 支持开发完成的应用在线发布(计划在App Studio1.2版本发布)。

3.入门教程

通常,工程师搭建一个数据门户需要开发数据、搭建后端服务和开发前端页面三个环节。本文将为您介绍 App Studio的基本功能及如何使用App Studio。

通常,数据工程师在DataWorks进行离线或流式数据开发。随着DataWorks的操作越来越简单,算法工程师、BI分析师、运营、熟悉SQL的产品经理等诸多角色,也逐渐可以在DataWorks进行数据开发。

针对不同种类的用户, App Studio可以助您快速搭建看数据的网页、查数据的App。

AppStudio	首页 报表					
	and the second s	i (Y				
运营教程		活动管理		商品管理		订单管理
互联网产品运营管理包含产品	管理、运营管理、团队管理、	活动运营,是一门说难又很	简单,每个人都能成为票友"玩	商品管理是指一个零售商从分	计标顾客的需求人手,对商品组	订单管理是客户关系管理的有效延伸,能更好的把个性化、
) 古官理、云贝官理、女王官 杏看教程	理	一票 的工种; 但也是一门 点击进入	光间早	古、定切方法、促制活动,以 点击进入	4x	左并化版夯有机的融入到各广 自击进入
- 87/12		AN MARKED V		AN UL ALL Y		
运营数据						
* 时间: 请选择时间	台 搜索					
产品图片	产品名称	产品编号	订单数量	订单金额	订单时间	订单状态
1	产品#1	1	12	22.5	2018-03-20	已完成
1	产品#2	2	22	21.5	2018-03-20	派送中
1	产品#9	9	40	50.5	2018-03-20	已完成
1	产品#6	6	209	90.5	2018-03-20	已完成
1	产品#10	10	69	90.2	2018-03-20	派送中
1	产品#4	4	87	205.5	2018-03-20	已完成
1	产品#7	7	20	20.2	2018-03-20	派送中
1	产品#5	5	112	120.5	2018-03-20	派送中
1	产品#8	8	30	13.5	2018-03-20	已完成
1	产品#3	3	51	23.5	2018-03-20	已完成
			AppStudio @2019 Cr	eated by DataWorks		



了解App Studio

● 菜单栏

工程 文件 编辑 版本 查看 调试 设置 模板 帮助 反馈	工程	文件	编辑	版本	查看	调试	设置	模板	帮助	反馈
-------------------------------	----	----	----	----	----	----	----	----	----	----

○ 工程

您可以通过工程菜单中的子菜单进行工程配置和查看当前工程属性。当前工程属性中包括工程ID、工程名、工程类型、创建时间和UUID等工程相关信息。

6	🛆 Ар	p Studio)							
ŵ	工程	文件	编辑	版本	查看	调试	设置	帮助	反馈	
ð	工程配 当前工 {}sett	置 程属性 tings.json				بر	S index.js 1 impo 2 impo 3 4 impo 5	× int Rea int {re int Exa	<pre>{} settingsjson act, {Component ender} from 're ample from '/ p artends Component</pre>	} from act-do

○ 文件

您可以通过文件菜单中的子菜单新建文件、打开最近的文件。

○ 编辑

您可以通过编辑菜单栏进行常用的编辑操作,全文搜索是对工程内所有代码内容进行搜索,并可打开相关的文件。全文搜索的详情请参见全文内容搜索。

撤销	ЖZ
恢复	¥介Z
剪切	жх
复制	жс
查找	₩F
替换	℃₩F
全文搜索	総 仓F
向上复制-	-行 ℃☆↑
向下复制-	-行 ℃☆↓
向上移动一	-行 ℃↑
向下移动一	-行 ℃↓

○ 版本

您可以进行**切换分支、拉取、推送、查看变更、提交、日志、初始化&关联远程仓库**和Merge Abort等操作。

6	App Studio								
ធ៌	工程文件编辑	版本	查看	调试	设置	发布	模板	帮助	反馈
ŋ	工程								
P	asset (j) > .alicode								
	> .settings > APP-META								
	> santa > src								
	> target 🛓 .classpath								
	E .factorypath	初始化&	关联远程位	金库					
	E .project								
	 appstudio.deploy.properties LICENSE 								
	pom.xml			<					

■ 切换分支

您可以通过+创建新分支创建本地新分支,然后推送到远程仓库。您可以选择一个本地分支,单击右 边弹出框中的checkout。您也可以通过merge,将选中的分支合并到当前分支。

切换分支	×	
选择需要切换的分支或创建一个新分支:		
+ 创建新分支		
brancj1	>	checkout
jj origin/br	ancj1 🔉	merge
master	>	
origin/brancj1	>	
origin/master	>	

您可以选择一个远程分支,单击右边弹出框中的check out as a new local branch,将该远程分支checkout到本地并重新命名。您也可以通过merge,将选中的分支合并到当前分支。

切换分支	×	Ø 8≡ << 1 ↓8
洗择需要切换的分支或创建一个新分支:		
+ 创建新分支		
Local Branches		
brancj1	>	
ij	origin/brancj1 🗲	
master	>	
Remote Branches		
origin/brancj1	>	checkout as a new local branch
origin/master	>	merge

■ 拉取

可以将远程分支的代码拉取到本地分支。

■ 推送

可以将本地分支的变更暂存后推送到远程分支。

∎ 查看变更

单击查看变更后,右侧导航栏会弹出本地变更文件列表。

工程	! 文件	编辑	版本	查看	构建
R	源代码管理:	GIT		~	c
"	暂存的更改				- 0
v	🛓 IndexCo	ntroller.c	lass targ	et/classes	;/ — M
4	更改				2 + 13
Ÿ	≣ demo.s	anta sant	a/pages/	demo.s	ッ + u
	F home.sa	anta sant		home.s	9 + M
	🎄 IndexCo	ontroller.j	ava src/m		9 + M
		a santa/g	bages/list		9 + M
	🛓 Newder	noApiCor	troller.cla	ass tar	ッ + U
	🎂 Newder	noApiCor	ntroller.jav	/a src/	🤊 🕂 🛡
	🎄 Newder	noBO.cla	ss target/	classe	୭ 🕂 🍕
	🛓 Newder	noBO.jav	a src/mai		ッ + U
	🛓 Newder	noService	e.class ta	rget/cl	୭ + ଏ
	🛓 Newder	noService	elmpl.clas	s targ	୭ + ଏ
	🎄 Newder	noService	elmpl.java		ッ + U
	🛓 PaiApiS	erviceImp	ol.class ta	arget/cl	7 + M
	🛓 PaiApiS	erviceImp	ol.java sro		7 + M

标识	说明
14	代表变更文件的个数。
U	代表新增的文件。
м	代表变更的文件。
2	单击后可以撤销更改。
+	单击后可以暂存更改。
-	单击后可以撤销暂存。
\checkmark	单击后可以提交或提交并推送暂存的代码。
•••	更多中包括推送和拉取操作。

■ 提交

可以将本地分支的变更提交以暂存,需要输入commit信息。

■ 日志

可以查看分支的所有提交记录,并可以筛选。

Log History				× ^			
message: keyword	ssage: keyword branch: All v user: All v date: All v date: All v 经检日期 - 结束日期 茴 > newshowproject						
path: path keyword C	ath: path keyword C newadd						
commit	message			ad63e05 2019-03-			
ad63e05			2019-03-24 19:07	In 3 branches: local_master_origin/masterShow All			
3c40ca1	add		2019-03-24 19:04				
6102d86	初始化工程		2019-01-25 15:45				
🗊 OUT 🏾 🌺 DEBUG	PROBLEM 🕅 Terminal 🏼 🕨 Version	Control					

■ 初始化&关联远程仓库

新建的工程可以关联远程仓库,从而进行版本管理。

○ 查看

您可以通过**切换全屏**将IDE设置成全屏,然后通过Esc键退出全屏。您可以通过**切换侧边栏**和**切换状态** 栏收起侧边和状态栏。

查看	构建
输出	☆≋U
	<mark>ዡ</mark> ^ዤF
切换侧	边栏眯B
切换状	态栏

- 调试
 - 如果您建的是前端工程,调试选项如下所示。

调试	设置			
	٦CR			
打开配置				
添加配置				
自定义镜像				

您可以配置运行参数、添加自定义镜像。

■ 如果您建的是后端工程,调试选项如下所示。



App Studio支持Java Debug,在后端工程的调试中,除配置和自定义镜像操作外,还有很多调试相关的操作。同时会有全量构建、增量构建、编译的操作入口。

○ 设置

您在开始使用App Studio前,需要配置SSH KEY和GIT CONFIG。您也可以通过偏好设置,设置自己偏好的属性,目前仅支持字体大小,后续会支持颜色、样式、主题、快捷键等。

设置	发布
SSH KI	EY
GIT TO	KEN
GIT CC	ONFIG
偏好设	Ħ

○ 帮助

您可以在帮助中查看产品使用文档、查看快捷键、查看版本历史和清空本地缓存。



○ 反馈

您可以通过反馈提交问题和需求。



- 左边栏
 - 。 入口

单击下图中的图标,即可展开工程区。



单击下图中的图标,即可展开接口定义区。



。 接口定义区

您可以添加接口并自动生成接口类代码,还可通过箭头,将左边的新代码同步到右边的本地代码中。

接口列表								
Q API名称 / 路径		接口分类: 请)	选择					+添加接口
接口名	API路径		请求方法	接口分类	版本	生成时间	操作	
getdetail	/getdetail		GET	newdemo		2019 - 3 - 19 12:58:44	接口详情 编辑 生成代码 版本选择 增加	
getList	/getlist			demo		2019 - 2 - 18 18:7:25		

添加接口								×
● 接口之教・	701 - Dat Oktore							
	Ø: Petstore							
● 接口说明:	Mi /demo/gett	151						
120000								
* 接口分类:	请输入							
■ 请求方法:	🧿 GET 💮 P	OST 🔘 P	UT 🕕 DEL	.ETE				
生成方式:	 自定义 	基于数据服务	ş					
入参定义:	参数名称	参数描述	参数类型		是否必填	默认值		
	参数名	参数描述		参数类型		默认值	操作	
				没有数据	居			
出参定义:	参数名称	参数描述	参数类型					
	参数名	参数	放描述		参数类型		操作	
				没有数据	居			
	📃 输出是否为数约	ß						
	·····································							
确认生成代码								×
GetdetailSen GetdetailApid GetdetailApid GetdetailBO.	rice.java src/mai riceImpl.java src/ Controller.java sr ava src/main/jav	1 package co 2 import com 3 import jav 4 5 public int 6 7 /** 8 * 月석 9 - 本 10 */ 11 List <g 12 }</g 	m.alibaba.dataw h.alibaba.datawo ra.util.List; rerface Getdetai ruggdt理波論 aram uid GetdetailBO> biz	orks.service	e.newden 1 .bo.Getc 2 3 4 5 6 7 7 3 8 9 9 9 11 12	package com.alibaba. import com.alibaba. import java.util.Lis public interface Get /## [* 具体业务处理逻辑 */ * List <getdetailb(}</getdetailb(<pre>.dataworks.servico Jataworks.servico jt; it idetailService { ii i> bizProcess();</pre>	.newdem bo.Getd

- ∘ 工程区
 - 文件夹操作

如果您创建的是后端工程,文件模板新建后,会帮您自动生成一些框架代码。

✓ common		
👍 Coc	新建 >	
👍 Res	新建文件夹	
🛓 Syr	上传文件	Раскаде
> configu	重命名	Java Annotation
> control	复制	Java Class
> demo.c	粘贴	Java Enum
✓ service	mite	Java Liiuili
× bo	1829 P-85	Java Interface

■ 文件操作

如果您创建的是前端工程,则新建操作只有文件一个选项。

Code '				29
Resu	重命名			
- -	复制			
Sync	32,123			
figur	删除			
trolle		>	Show F	listory
ani.new	demo			
aprille i	acino			

您可以重命名、复制和删除文件,也可以查看文件的GIT提交历史并进行版本对比。

● 编辑区

○ 右键操作

Go to Definition	ℋ F12
Peek Definition	℃F12
Find All References	
	ЖР
Go to Symbol	企業の
	ЖМ
	F2
Change All Occurrences	₩F2
Format Document	û℃F
Cut	
Сору	
Command Palette	F1

操作	说明
Go to Definition	单击后跳转至定义。
Peek Definition	单击后可以预览定义。
Find All References	单击后可以查找所有引用。
Workspace Symbol	单击后可以在项目中查找符号。
Go to Symbol	单击后可以跳转至符号。
Generate	单击后可以生成代码。
Rename Symbol	单击后可以重命名符号。
Change All Occurrences	单击后可以修改当前文件中的所有该符号名字。
Format Document	单击后可以格式化文件。
Cut	剪切。
Сору	复制。
Command Palette	单击后可以进入命令面板。

。 智能提示

。 智能补全



○ 智能诊断

	🛓 WordCount.java 🗴 🎍 PrintStream.class 🗴 👙 TestExample.java 🗙	
2 (j)	1 package com.package02;	a second
nples		I DELC
ain	4 sublic class WordCount {	
resources		
iava	7 public static void main(String[] args) throws Exception {	
✓ test nackade01	8 if (args.length != 2) {	
	<pre>9 System.err.println("Usage: WordCount <in_table> <out_table>");</out_table></in_table></pre>	
	10 System.out.print("hello world");	
 test.package02 	11 System.exit(2);	
🛓 WordCount.java		
est	13 Sustam put print(true):	
et	15 TestEXamolet = new TestEXamole():	
m.xml	16 t.init01("x", 1, 2);	
	19 System.out.println("hello");	
	25 public void test001() {	
	30 }	
	34 public void test002() {	
	39 } 100% Ready	

。 查找定义



○ 查找引用



。 自动导入



○ 查找符号



○ 多光标编辑



o 查找、替换



代码格式化



。 括号匹配

9	App Studio 工程 文件 编辑	1 版本 責書 调试 设置 帮助	Edit Config 🗸 🕨 🌺 📕
۵ ت	IE :: yungi ① > santa * arc * main * java * com.alibaba.dataworks > common	<pre> & Mainjava x</pre>	i i i i i i i i i i i i i i i i i i i
	 controller.page demo.controller service Main.java 	9 * 主京,入口录 10 */ 11 #SpringBotApplication 12 @EnableAutoConfiguration 13 @ComponentScan(basePackages = "con.alibaba.dataworks") 14 public class #ai {	l i
	Fresources ▶ target ≱ pom.xml	<pre>public static void main(string[] args)§ 16 5pringApplication.rum(Main.class, args); 17 18 19 19 20 20 20 20 20 20 20 20 20 20 20 20 20</pre>	

- 右上角图标区
 - ∘ 编码规约



⑦ 说明 构建需要在工程运行或者debug时才能进行。

• Run/Debug Configurations

🛡 i 👫 Unnamed 🗸 🕨	*		
Run/Debug Configurations			×
添加 删除	Name: Unnamed		
🗸 📮 Application	• Main class:	com alibaba dataworks Main	
Unnamed	VAL antiona		
	vivi options:		
	Program arguments:		
	Environment Variables:		
	JRE:	1.8 - SDK	
	PORT:	7001	
	机器:	2vCPU, 4G内存 ~	
	开启HOTCODE:	● 是 ○ 否	
		Cancel Apply OK	

○ Debug入口



从左到右的图标依次代表运行、Debug和停止工程。

● 底边栏

○ DEBUG/RUN面板

单击运行或Debug工程,该面板会弹出,展示进度和信息。

输出		
â	019-03-25 16:40:01.854_INFO_509 [main] s.v.s.m.m.a.RequestKappingHandlerMapping : Mapped "{[/error]}" onto public org.springframework.http.ResponseEntity <je< th=""><th></th></je<>	
	a.util.Map-java.lang.String, java.lang.object>> org.springiramework.boot.autoconfigure.web.BasicErrorController.error(javax.serviet.http.httpservietkequest)	
	019-03-25 16:40:01.886 INFO 509 [main] o.s.W.s.handler.SimpleOriBandlerRapping : Mapped URL path [/web]ars/**] onto handler of type [class org.springframew	
	rk.web.servlet.resource.ResourceHttpRequestHandler]	
	019-03-25 16:40:01.886 INFO 509 [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**] onto handler of type [class org.springframework.web.	
	ervlet.resource.ResourceHttpRequestHandler]	
	019-03-25 16:40:01.914 INFO 509 [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**/favicon.ico] onto handler of type [class org.springfr	
	mework.web.servlet.resource.ResourceHttpRequestHandler]	
	019-03-25 16:40:02.580 INFO 509 [main] o.s.j.e.a.Annotation#BeanExporter : Registering beans for JMX exposure on startup	
	019-03-25 16:40:02.607 INFO 509 [main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 7001 (http)	
	019-03-25 16:40:02.611 INFO 509 [main] com.alibaba.dataworks.Main : Started Main in 5.297 seconds (JVM running for 6.031)	
ā	NUT RUN = PROBLEM INTerminal & Version Control	

○ PROBLEM面板

当工程有问题时,运行或Debug工程该面板会弹出。

输出	
	2013-03-25 16140161.854 ITPD 509 [main] a.v.s.m.s.requestAppingHandlerKapping : Mapped "{(/error)}" onto public org.springframework.http.ResponseEntity <ja ava.lang.string_java.lang.object="" va.utl.maprig=""> org.springframework.http.ResponseEntity<ja ava.lang.string_java.string_java.stri<="" th="" va.utl.maprig=""></ja></ja></ja></ja></ja></ja></ja></ja>
	2019-03-25 16:40:01.886 INFO 509 [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/webjars/**] onto handler of type [class org.springframew
	ork.web.servlet.resource.ResourceHttpRequestHandler]
	2019-03-25 16:40:01.886 INFO 509 [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**] onto handler of type [class org.springframework.web.
	servlet.resource.ResourceHttpRequestHandler]
	2019-03-25 16:40:01.914 INFO 509 [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**/favicon.ico] onto handler of type [class org.springfr
	amework.web.servlet.resource.ResourceHttpReguestHandler]
	2019-03-25 16:40:02.580 INFO 509 [main] o.s.j.e.a.Annotation#BeanExporter : Registering beans for JMX exposure on startup
	2019-03-25 16:40:02.607 INFO 509 [main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 7001 (http)
	2019-03-25 16:40:02.611 INTO 509 [main] com.alibaba.dataworks.Main : Started Main in 5.297 seconds (JVM running for 6.031)
向	OLIT DIN PROBLEM DIFFERENCE Version Control

○ TERMINAL面板

当工程运行或Debug时,可以通过Terminal触达机器进行bash、vim命令操作。

		× ^
	Local	
	[admin@webide /etc]	
	315 additime remeave hashre centes-release-unstream dhus-1 DTP COLORS 256-olor environment gnung gobadow hostname init d kde	14
	and the second machine-id mash ont passing of profiled relative for the second second profiled relative for the second se	8 811
	buid sudo-ldap.conf system-release tmpfiles.d weetrc yum	
	aliases binfmt.d chkconfig.d default DIR_COLORS.lightbgcolor exports GREP_COLORS gshadow- hosts inputrc krb5.conf	
	.so.conf.d localtime modprobe.d nsswitch.conf os-release pkcsll prelink.conf.d protocols rc2.d rc6.d resolv.conf sasl2 services skel	su
	do.conf sysconfig system-release-cpe udev X11 yum.conf	
	alternatives BUILDTIME csh.cshrc depmod.d dracut.conf filesystems group gss hosts.allow issue krb5.conf.d	1 11
	baudit.conf login.defs modules-load.d mswitch.conf.bak pan.d pki printcap python rcJ.d rc.d rpc securety shadow ssi	su
	doers systet.a terminic voorsole.com xag yum.repos.a bash completion d centes-release osh login DTE COLORS dramut conf d governt group- bost conf bosts denv igsue pet 1d so cache	. 11
	$p_{\rm max}$ p_{\rm	d su
	doers.d systemd timezone virc xinetd.d	
	[admin@webide /etc]	
	\$	
Ð	OUT 🕨 RUN 🗮 PROBLEM 🔛 Terminal 🔰 Version Control	

○ VERSION CONT ROL面板

该面板展示Git history和Git log两部分内容。

• 右边栏

• Runtime

工程运行完成时会展开这个面板,并展示机器信息和访问链接。

×	Runtime	
	Machine Ready Release	
	CPU: 2 Core Memory: 4 G	Share
	项目已经启动,访问: 前端: 打开链接 复制链接	Data
	归喻。 <u>打开链按</u> 显向链接	

- 如果是后端工程, 仅展示后端访问链接。
- 如果是前端工程, 仅展示前端链接。
- 如果是可视化搭建工程,可展示前端访问链接和后端访问链接。

• Share

×	Share		88邀请	R	
	协作者	权限		untime	
	alicode_cloud(我) is editing In	所有者			
				Data	
遗	请协作者				×
	•用户名:				
	请输入阿里云账号搜索				
	* 权限:				
			确认		取消

您可以邀请他人协同编程,目前支持8人同时编辑同一工程同一文件。

∘ Data

数据服务是承接DataStudio和App Studio的重要一环。

X Data				+ 前往	DataService 新增 API	R
Q请辅	ì入 API 名称	alicode_pre		请选择服务分组		ntime
ID	API 名称	API Path	Project	API 分组	操作	
1	test	/test	91772			Sha
2	脚本模式	/scirpttest1	83949			re
3	zishutest	/zst/test1	83949			

数据服务在App Studio中有两种使用方式,更多详情请参见数据服务。

- 可以在代码中直接使用,或者对接口结果进行再加工。
- 可以在可视化搭建中直接配置为组件的数据源。

• Preview

如果您创建的是前端工程,在右边栏会有Preview入口,在运行工程时可以实时预览前端页面。



• 可视化搭建区

您的工程必须是可视化搭建工程,在工程中找到santa-pages下的.santa文件,双击打开。

88	布局	基础	表单	图表	高级	更多	搜索组件	Q			0))	+	*	•	模板	保存
									国内 遺先以下部指动満加一个组件							
									()							

左上角是组件区,您可以选择需要的组件,也可以通过搜索组件名称选择需要的组件。右上角的图标分别 是切换为代码模式、导航配置、全局数据流配置、撤销、重做、预览、保存为模板和保存。 拖动一个表格组件到画布,单击表格组件,右边会弹出组件配置区,可以对组件的属性、样式进行配置, 也可以进行组件联动配置。

움 布局 基础	表单 图表 高级 更多 搜索组件 Q		♠ < 	● 模板 保存
ID	Name	组件配置		
	Turro	属性 样式		
100	ajkoajkoajkoajkoajkoajkoajkoajkoajko	数据源 🕐		
101	ajkoajkoajkoajkoajkoajkoajkoajkoajko	请输入		
102	ajkoajkoajkoajkoajkoajkoajkoajkoajko	循环请求问题	时间 (单位:秒) +	
103	ajkoajkoajkoajkoajkoajkoajkoajkoajko	请求方法		
104	ajkoajkoajkoajkoajkoajkoajkoajkoajko	Get		
105	ajkoajkoajkoajkoajkoajkoajkoajkoajko	授东参数 变量名	变量值	操作
106	ajkoajkoajkoajkoajkoajkoajkoajkoajko			
107	ajkoajkoajkoajkoajkoajkoajkoajkoajko			
108	ajkoajkoajkoajkoajkoajkoajkoajkoajko	添加		编辑代码
109	ajkoajkoajkoajkoajkoajkoajkoajkoajko	返回数据处 理	函数	
		编辑代码		
		表格列配置功		
Body → DataTable	BDODIEU II Terminal II Vereian Castrol	✓ 字段		

创建后端工程

- 1. 基于样例工程新建工程。
 - i. 进入App Studio页面,单击工作空间页面的通过代码创建工程。

🌀 🛛 App Studi	o		
■ ① 工作空间 Q 应用空间	欢迎来到 App Studio		
▲ 橋飾云山	C 通过模板的建工程	回 通过代码创建工程	∰ide#≑}\I£
	我的工程		
	DataOS_App 11 小时前要新		
	⑦ 管理员 的建模板 管理		

ii. 填写新建项目对话框中的工程名和工程描述,选择运行环境为springboot样例模板。

6	App Studio							
© Q	■ 工作空间 应用空间	工作空间 > 新建项目 新建项目						
Ŷ	模板空间	概版工程 代码工程 导入GH工程 ●工程名: 请输入工程名称,英文字符开头,只能包含数字、英文字符、 ・						
	* 透神运行环境: React - 组件卸手架 appstudio JOK8 + SpringBoot + 可提化活建							

- iii. 配置完成后,单击**提交**。
- 2. 配置运行参数。

填写好配置的名称,选择运行的main函数,选择机器规格,单击OK即可完成配置。 您可以通过左边的**添加**按钮添加多个配置,运行时选择不同的配置运行。

🛛 i 👫 Unnamed 🗸							
Run/Debug Configurations X							
添加 删除	Name: Unnamed						
✓ ■ Application Unnamed	* Main class: 👔 请选择						
	VM options:						
	Program arguments:						
	Environment Variables:						
	JRE: 1.8 - SDK						
	PORT: 7001						
	购买资源包 机器: 4vCPU,8G内存						
	Pre-Launch Option: 👔 请选择	÷ 🗸					
	开启HOTCODE: 💿 是 🔵 否						
		m <pm< p=""> <pm< p=""> <pm< p=""> <pm< p=""> <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> m <pm< p=""> m <pm< p=""> m <pm< p=""> <pm< p=""> <pm< p=""> <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> <pm< p=""> m <pm< p=""> m <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> <pm< p=""> <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""> m <pm< p=""> <pm< p=""></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<></pm<>					

3. 运行工程

单击红框中的运行图标开始运行工程。

	Unnamed	*		
--	---------	---	--	--

第一次运行需要分配机器、初始化语言服务,需要较长时间运行完成,完成后会弹出runtime窗口,展 示访问链接。



4. 访问工程

单击**打开链接**,即可访问工程。

\leftarrow	\rightarrow	С	① 不安全 pcsoxspre45743ad0g4mgkz4nt9gnj-80.r-app-data.aliyun.co	m
Hell	o! /	Арр	o Studio!	

在链接中加上/testapi并刷新页面。



创建前端工程

App Studio提供完善的前端开发能力,支持与本地一致的前端开发体验。您可以在App Studio中创建前端 工程,以自己熟悉的方式进行HT ML、CSS、JS和React的开发,并且您在开发过程中不需要掌握和理解新的 概念。

1. 基于样例工程新建工程。

i. 进入App Studio页面, 单击工作空间页面的通过代码创建工程。

. .

ii. 填写新建项目对话框中的工程名和工程描述,选择运行环境为react-demo样例模板。

6	App Studio)							
Ð	三 工作空间	工作空间 > 新建项目							
۹	应用空间	新建坝日							
Ŷ	模板空间	慢频工程 代码工程 导入GH工程							
		* 工程名:	请输入工程名称,英文字符开头,只能包含数字。	英文字符					
		* 工程描述:	请输入工程描述						
		* 选择运行环境:	react-component React組件	~	react-demo 样例模板			springboot 样例模板	
			appstudio 样例模板	~					
		提交							

iii. 填写工程名和工程描述, 单击**确认**。

2. 配置运行参数。

您可以选择机器规格、配置端口,如果没有特殊需求可以直接使用默认的配置,单击OK即可。

🔍 i (main 🔍 🕨 🔆			
Run/Debug Configurations			×
添加	Name:	Unnamed	
✓ Server Ser		Install Cmd:	nom install
Unnamed		Start Cmd:	nom start
		· · · · · · · · · ·	
	Env	ironment variables:	
		Initialize Script: 🚹	
		PORT:	3000
		羽买资源包 机器:	4vCPU, 8G内存 ~ ~ ~

3. 运行工程

单击右上角的运行图标开始运行工程,目前支持以 tnpm start 的方式启动前端工程,配置了webpac k-dev-server的工程可以无缝对接运行。
启动运行后,开发者可以在日志中看到依赖安装及应用启动的日志,运行完成后右边会弹出页面的预览 页面。您可以实时修改代码并进行保存,便可实时生效。



4. 访问工程

单击链接边的箭头即可打开访问页面,App Studio对于前端工程的编辑开发提供了与本地IDE一致的开发体验,包括HTML、CSS、LESS、SCSS、JavaScript、TypeScript、JSX和TSX等文件的智能补全、函数 签名、重构和跳转等功能。同时您不需进行搭建环境、下载依赖等操作,可以在模板基础上进行前端开发。

搭建前端可视化工程

1. 基于样例工程新建工程。

i. 进入App Studio页面, 单击工作空间页面的通过代码创建工程。

ii. 填写新建项目对话框中的工程名和工程描述,选择运行环境为App Studio样例模板。

\$	App Studio	i										
© Q	三 工作空间 应用空间	工作空间 > 新建项目 新建项目										
Ŷ	模板空间	模板工程 代码工	模板工程 代码工程 与入C比工程									
		* 工程名:	请输入工程名称,英文字符开头,只能包含数字。	英文字符、								
		• 工程描述:	请输入工程描述									
		●选择运行环境:	react-component React + 组件脚手架	~	react-demo React		~	Springboot JDK8 + SpringBoot	•			
			appstudio JDK8 + SpringBoot + 可現化語鍵									
		提交										

- iii. 配置完成后,单击**提交**。
- 2. 打开home.santa文件。

在santa-pages目录下找到.santa文件,有home和list两个样例页面。



i. 打开home.santa, 是一个简单的报表页面。



ii. 选中一个组件,右边会弹出组件配置。



iii. 单击数据源输入框, 会弹出接口列表。



App Studio为您提供一些数据服务接口,以便您入门使用。您可以单击+新增数据服务接口前往数据服务中新增接口,通过API路径查看现在的组件对应的接口。

⑦ 说明 您可以尝试去掉接口自行配置,体验组件配置数据源的效果,也可以对样式进行修改。

3. 添加组件&配置接口。

i. 从图表中拖动一个柱状图到画布上。

单	图表	高级	更多	搜索组件 (٦	
	数据表格					
	折线图		DoO	000	Col	000
lipav	柱形图	,	基础林形图	分组柱形图	堆叠柱形图	百分比堆叠柱形图
p.c.y	条形图		E 94112712134	23 9011277764	AL 211// BI	
	饼图					
	面积图					
/un:	词云					



ii. 选中组件,单击弹出的组件配置框中的数据源输入框。

iii. 选择第7个接口,单击**选用**,便成功配置接口。

程	文件	编辑	版本]	宣看	जार स	2置 (模板	帮助	反馈								Applic	ation 🗸	•
5	home.san	ta ×																	
							搜索	组件	Q										
Г										BasicCo	lumn					组件配置			
	30											API	5称 / 路径	dataworks_public ~	选择分组	数据源			
	25															请输入			
	20													API路径	API分组	循环请求问照时间			
														/dsproxy/dSaPild=230	AppStudi	- 0 +			
	15															请求方法			
	10													/dsproxy/dSaPild=230 1	AppStudi 0	Get			
	-													/dsproxy/dSaPild=230	AppStudi	提索参数			
	5															变量名	变量值		操作
	0 —				Max		4.4			1.0	i.i.			/dsproxy/dSaPild=229 9	AppStudi o	year			
-		Jan		eu	mai		Api	Ma	y	Jun	Jui			/dsproxy/dSaPild=229	AppStudi	添加		編制	代码
			用白	士间本》	5									8		返回数据处理函数			
			HI	ጠባም	3 5									ord /dsproxy/dSaPild=229 7	o	编辑程行专马			
		tiann	nao: 0.17					80						/dsproxy/dSaPild=229 6	AppStudi o	图表配置 ②			
Bod	lv → Basi	icColumn	-	_		aliyun:	0.33	70				8		/dsproxy/dSaPild=229	AppStudi	编辑代码			

iv. 此时组件中没有返回结果,是因为此接口需要填写入参和返回的列。

工程	文件	銅	粗	版本	查看	调试 :	设置 模	板 帮助	反性	1									¢ 1	1 App	lication 🗸)	
a	E hom	.santa																					
"								援索组件	c	۱										* * *			North
۳.											Rasic	Column	接口列						组件配置				ā
											buon		数据目						展性 将武	高級			
													API	3称/ 路径	dat	aworks_public V	洗槿分組		数据源				6
																			/api/1.0/dsproxy	?dsApiPath	=/project/9	177: {}	aie
													ID			A D198 63	AD16340		播环请求间隔时间	1) (单位: 利			
													10			/deproyu/dSaDild=220			- 0 +				
													1			2	AppStudio		请求方法				Lat.
													2			/dsproxy/dSaPild=230	AppStudio		Get				
																/dsproxy/dSaPild=230			投索参数				
													3				AppStudio		变量名	交量值		銀作	
													4			/dsproxy/dSaPild=229 9	AppStudio		year	交量(
١.													5			/dsproxy/dSaPild=229 8	AppStudio		滅加			編輯代码	
P				用户	访问来演	原							6			/dsproxy/dSaPild=229 7	AppStudio		返回数据处理函数 编辑代码	24 日数接处理函数 编 代码			
			alipa	y: 0.21	-	tiar	nmao: 0.17		80				7			/dsproxy/dSaPild=229 6	AppStudio		图表配置 ①				
									60				8			/dsproxy/dSaPild=229 5	AppStudio		编辑代码				
									50			-	9			/dsproxy/dSaFild=229	AppStudio						
							taobao:	0.2	40							/dsproxy/dSaPild=229			图表标题				
			aliyun:	0.33	-				30				10				Appstudio		BasicColumn				
									20		-						etta 🔟 🕻	- D >	因表数据				
		• ti	ianmao	• tar	obao 🔹 a	aliyun 😐	alipay		10			-							编辑代码				
									0 -										X轴字段				
										二月	-4	тя ;	1						×				
																			Y轴字段				
											7〕 単/	运 早趋势							y				
	Body →	BasicC	Column																				

您可以单击第7个接口的详情,查看请求和返回的内容。

API 详情						2
API activeUser			*		[制调用地址	复制带参数调用地址
测试						
≔ API 基本信息	~	请求参数				
API ID 2296		▼ 应用请求參数				
API 分组 AppStudio 负责人		参数名称	參数类型	操作符	是否必填	示例值
创建时间 2019-03-25 23:09:16 描述 activeUser		year	string	EQUAL	是	
M HTTP 接口信息	~	返回参数				
API调用地址		▼ 应用返回参数				
		参数名称	参数类型	示例值		
请求方式 GET 返回类型 JSON		month	string			
🖾 数据源信息	~	value	float			
名称 AppStudio		year	string			
类型 lightning 连接信息		正常返回示例				
JDBC Url	8	{ "data"				
用户名表名		Ì	"month": "九月" "year": "2019"			
数据描述		}, {	Value : 12			

⑦ 说明 由于这是示例项目,您会无权访问,建议您搭建可视化工程时,使用自己的账号到数据服务创建接口。

v. 按照下图填写组件配置。



配置完成后,即可看到组件中已有配置好的数据。

4. 打开list.santa文件。

App Studio可视化搭建不仅可以搭建报表,还可以搭建应用。

打开*list.santa*文件,是一个简单的数据应用。其中包括图标、链接、视频、列表和搜索等组件,详情请参见<mark>可视化搭建概述</mark>。

5. 导航配置。

您搭建一个应用,通常不会只有一个页面,多个页面之间需要一个导航配置。

单击右上角的导航配置标识,即可打开导航配置页面。

<> 🖬 ≘ ♦	A O 1	版	
全局导航配置		*** ***	
是否显示		自贝 报衣	
主题			
浅色 ~			
Logo 图片			
https://img.alicdn.com/tfs/TB1WIQOmpz 上传			
「标题			
AppStudio			
■是否固定于页面顶部			
菜单项			
1链接名称			
首页			
1 链接地址			
/			
路由文件			
pages/home.santa			
是否隐藏			
【链接名称			
报表			
┃ 链接地址			
1 Plant to the			
pages/list.santa			AppStudio ©2019 Created by DataWorks

- 6. 配置运行参数,可参见搭建后端工程的操作。
- 7. 运行工程。

单击右上角的运行标识即可开始运行工程,运行后会弹出Runtime面板,单击里面的前端链接即可访问 工程。

4.功能介绍 4.1. 导航页 4.1.1. 工作空间

您可在工作空间页面创建和管理工程。

App Studio的工作空间页面,将为您展示当前创建的工程列表,并提供三种创建工程的方式,详情请参见工 程管理。

欢迎来到 App Studio			
¢.	۵	Ð	
通过模板创建工程	通过代码创建工程	通过Git导入工程	
我的工程			
Q 请输入 搜案			
	-	-	-
			8 天前更新
S管理员 创建模板 管理	管理员 创建模板 管理	管理员 创建模板 管理	② 管理员 创建模板 管理

单击工程卡片,即可进入工程开发页面。您也可单击创建模板或管理,进行相关操作。

创建模板

- 1. 单击相应工程下的创建模板。
- 2. 填写生成模板对话框中的配置。

生成模板					×
* 模板名称:	请输入模板名	吕称			
* 模板描述:	请简要描述相	莫板功能			
* 分类:	数据报表	数据应用	数据大屏	其他	
					生成
配置				说明	
模板名称				输入模板的名称。	
模板描述				对模板进行简单描述。	
分类				选择模板的分类。	

3. 填写完成后,单击**生成**。

管理

您的工程可以发布为一个应用,为方便您的版本管理,您可将工程发布成不同的版本,然后再进行应用发 布。

1. 单击相应工程下的管理,即可进入工程管理页面。

工作空间 > 工程详情	代码空间	代码仓库	发版
workshop			
项目描述			
工程成员:			
論理员:			
开发人员:			
\$9A2:			
历史版本 已发布应用			
and a second			6 天前
Market			

2. 单击右上角的发版,选择要更新版本的应用。

选择要更新版本的应用]		×
test01 6 天前	+ 新建)	- 坂本	
部署参数配置:	Кеу	Value	操作
		没有数据	
描述:	请填写本次发版的相关描述,便于跟踪们	自息	
			更新取消

⑦ 说明 您需要将工程绑定Git仓库后,方可进行发版。

3. 配置完成后,单击更新,即可产生一个新的版本。

4.1.2. 应用空间

应用空间包括我开发的应用、我分享的应用和第三方应用三大模块。

? 说明

- 仅购买旗舰版的用户,可以查看我分享的应用。
- 仅购买企业版和旗舰版的用户,可以查看第三方应用。

我开发的应用

我开发的应用页面为您展示已开发的应用,您可以对应用进行发布,也可以通过**部署控制台**进入**应用运** 维页面。

我开发的应用(10)	我分享的应用(6)	第三方应用(1)				111
						11
Q,请输入		搜索				
-						
● 运行中	部署控制台	发布 分享	● 未部署	部署控制台	发布	分享
in the second			-			
_						
● 部署失败	部署控制台	发布 分享	🔒 部署失败	部署控制台	发布	分享

⑦ 说明 分享入口仅对购买旗舰版的用户可见。

部署控制台

单击相应应用下的部署控制台,即可进入运维页面。

	操作 > 概览	监控 镜像 变]	更 资源				
ム 应用详情	应用信息 描述: asdfas	● 正常 QPS(req/s): 0.1	RT(ms): 66.67	分组信息 ● 总共 1 ● 正常 1 ● 异常 0 		 机器値 ・ 息共 ・ 正常 ・ 昇常 	1. 1. 0.
	分组列表						+ 创建分约
	分組名 ţ	☆ 实例规格 🛟	マ 阿 県	₽ \$	⊽ 描述 💲		状态
					demo		• 正常
	机器列表						
	分组名		▽ 主机名 💲		IP地址 ≑	实例规格 🛟	☆ 状态
			10012-00000-000			-	• 正常

运维页面为您展示所有应用的运维情况,您可以在左侧下拉框选择需要查看的应用。

● 概览

概览页面为您展示应用信息、应用状态、分组信息、机器信息、分组列表和机器列表等信息。

● 监控

监控页面为您展示应用的详细运维指标,包括3个应用指标、8个系统指标和7个JVM指标。



● 镜像

镜像页面为您展示每个分组使用的镜像和构建时间。

列表			
分組名 💠 🛛 镜像ID 🛊	构建时间 🔷	描述	
NAME AND A DESCRIPTION OF A DESCRIPTION OF A DESCRIPTION OF A DESCRIPTIONO	2019-05-31 00:29:53.		

● 变更

变更页面为您展示进行的部署、应用扩容或机器下线等操作。单击**变更单ID**,即可查看变更详情。

变更单								
变更单ID	安更类 型 \$ ♡	変更対象 💲	创建者 🛟	创建时间 🝦	结束时间 ţ	运行时长	状态	描 述 [◆] ♡
app:57889806390- 155	app_deploy			2019-05-31 00:23:45.0	2019-05-31 00:32:55.0	9分10秒	• 成 功	
app: 559233071699	app_dilatation			2019-05-31 00:17:51.0		280小时31分 38秒	• 执 行中	asdfa

当应用正在部署时,可以在此查看详细的部署信息和日志。

←返回变更列表	B000084-180800808188	重试 终止
变更单ld:	变更类型: app_deploy	状态
10000000	变更对象:deploy_0530_1	成功
创建者:1633057889806390	创建时间: 2019-05-31 00:23:45.0	
结束时间:2019-05-31 00:32:55.0	描述:	
进度		
✓ 创建发布单 ✓)提交发布	
SUCCEEDED	SUCCEEDED SUCCEEDED	SUCCEEDED
	查看详情	

● 资源

资源页面为您展示您所购买的VPC。购买VPC后,需要在此进行新增操作。

VPC列表	Ę				新增VPC	С
ID	角色标识 ţ	☆ 安全组ID 🖕	交换机ID 🝦	操作	¢	8
11						

单击相应的ID,即可进入VPC详情页。

←返回 VpclD:11				
角色标识:		安全组ID:	attaches de la constantia	
交换机ID:		描述:		
ENI列表				新增ENI
EnilD 🛟	∵ EcsID 🗢	▽ 描述 💲	▽ 操作 💲	
0.0001000000000	L-Pathante	Created	by OPEN API 绑定ECS	
	1.001(000)000	Created	by OPEN API #REECS	

● 操作

您可以进行应用重启、机器重启、机器下线和应用扩容四项操作。

	Annu, 1998, 1	操作~ 概览	监控 —	镜像 变更	资源	
ふ 应用详情	(1.1:) 应用信息			▲ 应用状态		രംജ
	dem		/ ۲	• F #		6
	Gon	❷ 机器下线		с тж		•
	描述: asdfas	目 应用扩容	QPS	6(req/s): 0.1	RT(ms): 50	
	分组列表					

○ 应用重启

在应用重启对话框中,对当前操作进行简要描述。单击执行,即可重启整个应用。

应用重启			х
	描述:	请输入描述	
		执行	

○ 机器重启

在机器重启对话框中,选择**分组**和机器,并进行简要描述。单击**执行**,即可对当前应用的某个分组下 的某台机器进行重启。

机器重启			Х
	■分组:		
	10 00		
	≠机器:	×	
	描述:	请输入描述	4
		执行	

○ 机器下线

在机器下线对话框中,选择分组和机器,并进行简要描述。单击执行,即可将当前应用的某个分组下的某台机器移除,放回资源池中。

机器下线	×	<
*分组:		
* 机器:		
描述:	请输入描述	
	执行	

○ 应用扩容

在**应用扩容**对话框中,选择**扩容分组、可用机器**,并进行简要描述。单击**执行**,即可将您的资源池中的机器,加入到当前应用的某个分组下。

应用扩容	X	
★扩容分组:		
* 可用机器:	请选择	
	购买机器	
描述:	请输入描述	
	执行	

发布

单击相应应用下的发布,即可进行发布操作,详情请参见应用部署。

分享

单击相应应用下的**分享**,购买企业版及以上版本的用户,可以将应用分享给其他用户。分享成功后,您可以 在**我分享的应用**列表中进行查看,对方可以在第三方应用的列表中进行查看。

应用分享			×
* 名称:	不超过50位数字、字母、下划线线	1成的字符	
* 地域:	cn-shanghai		
部署参数配置:	Кеу	Value	操作
			删除
		-	删除
	添加		
* 阿里云账号:	请输入阿里云账号ID,请到账号管理	巨面查看	
备注:	可以填写分享应用的备注		
法律声明	<<阅读相关法律条文>>		
			分享 取消

我分享的应用

我开发的应用(10) 我分享	[的应用(6) 第三方应用(1)		-77
Q 请输入	搜索		

_			
	部署通知		部署通知
		NOR17	
	部署通知		部署通知

进入应用空间 > 我分享的应用页面,即可查看分享过的应用。

单击相应应用下的部署通知,可以将应用的代码更新推送给被分享的用户,进行应用部署。

应用更新通知			×	<
* 名称:	****			
* 地域:	shanghai			
部署参数配置:	Key	Value	操作	
		没有数据		
	添加			
备注:	可以填写分享应用的备注			
			いしていた。	

第三方应用

进入**应用空间 > 第三方应用**页面,可以查看别人分享给您的应用,并进行部署和发布等操作,操作方式和 应用空间一致。

我开发的应用(10)	我分享的应用(6)	第三方应	证用(1)	
Q 请输入			叟索	
-				
● 未部署	部署招	制台	发布	

4.1.3. 模板空间

模板空间为您展示所有通过工程创建的模板。

数开发的模板	HH.		
Q 请输入 搜索			
订单报表	销售数据报表	电商销售数据大盘	发货管理
该模板用于电商领域订单数据的报表展示 ,	该模板通过报表的形式展现企业的销售数据	该模板用于展示电商领域的销售数据,让决策者 能一目了然知道企业的销售情况	该模板可用于电商领域的发货管理
61#II#			6181.W
教授委托	妥体听动教师十周	Daebhoard.若可	
84,316 18 134	示机血红欧路入阱	Dasilualu-目贝	
该模板是用来做一些数据的纯报表展示	该模板主要是对一些系统监控数据做大屏展示, 从而更好的知道系统的运维情况,及时采取应对 措施	该模板可以做一些运营管理,包括一些运营的教 程、订单和活动管理。也可以展 示一些订单和销 售数据。	
ês Mê I. Mî		CIAR I AR	创建工程
< 上一页 1 2 下一页 >			

您可以单击相应的模板卡片,进入模板详情页面。然后单击代码空间,即可查看模板相应的工程代码。

您也可以直接单击相应模板下**创建工程**,即可跳转至通过模板创建工程页面,基于当前模板创建工程。

模板工程 代码工	程 导入GIt工程		
*工程名:	请输入工程名称,英文字符开头,只能包含数字、英文字符	Fx _x +	
* 工程描述:	请输入工程描述		
*选择模板:	全部模板 数据报表 数据大屏 数据应用		
	CHEA	genomen anno	某某电商销售数据大盘
	1226 I 942 I 2256 I 972 I 972 WRR 5674 I 7541 I 6512 I ¥12854 mm mm		
	Init Init <th< th=""><th>000</th><th></th></th<>	000	
	订单报表	销售数据报表	电商销售数据大盘
	ESER		

4.2. 工程管理

本文将为您介绍如何新建和管理工程。

您可以通过模板、代码和Git导入三种方式新建工程。

通过模板创建工程

- 1. 进入App Studio页面,单击工作空间页面的通过模板创建工程。
- 2. 填写新建项目对话框中的工程名和工程描述,选择相应的模板。



? 说明

- 。 您可以选择自己定义的模板,也可以选择系统提供的模板创建工程。
- 。 通过模板创建的都是可视化工程。

3. 配置完成后,单击**提交**。

通过代码创建工程

如果想进行纯代码开发的工程,可以通过代码创建工程。App Studio提供了4种运行环境的代码模板,您可以根据自身需求进行选择。

- 1. 进入App Studio页面,单击工作空间页面的通过代码创建工程。
- 2. 填写新建项目对话框中的工程名和工程描述,选择相应的模板。

6	App Studio							
0	三 工作空间 应用空间	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)	爱 导入Glt工程					
		● ፲程名:	请输入工程名称,英文字符开头,只能包含数字。	英文字符				
		* 工程描述:	请输入工程描述					
		* 选择运行环境:	react-component React组件		react-demo 样例模板	,	springboot 样例模板	
			appstudio 样砂模板	~				
		提交						

3. 配置完成后,单击提交。

通过Git导入工程

如果您已经有Git代码,可以直接导入Git代码创建工程。此处仅支持Code中的Git代码导入。

- 1. 进入App Studio页面, 单击工作空间页面的通过Git导入工程。
- 2. 填写新建项目对话框中的Git 地址、工程名和工程描述,选择相应的运行环境。

三 ① 工作空间 Q 应用空间	工作空间 > 新建项目 新建项目						
★ 模板空间	模板工程 代码工程	星_ <u>导入GIt工程</u>					
	* Git 地址:	请输入 Git 地址					
	* 工程名:	请输入工程名称,英文字符开头,只能包	含数字、英文字符、				
	* 工程描述 :	请输入工程描述					
	* 选择运行环境:	react-component React组件	~	react-demo 样例模板	-	springboot 样 例模 板	~
		appstudio 样例模板					
	提交						

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。

6	▲ App Studio	HAR DE BELEVEL VERDO				
Ш	G	ţ.		ិ	ব	
Ð	通过模板的	创建工程	通过代	码创建工程	通过Git	入工程
۹						
ŵ						<i>211</i> 111
	我的工程					
	Q 请输入	搜索				
	-		-		100000	
	39 分钟前更新		1 小时前更新			
	♥ 管理员	创建模板 管理	◎ 管理员	创建模板	◎ 管理员	创建模板 管理
	-				100000.000	
	3 小时前更新		7小时前更新		17 小时前更新	6-128-45 45 MK18
	○管理员				●管理风	

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

⑦ 说明 如果是其他人分享给您的工程,将不能进行创建模板的操作。

App 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。

∆ App Studio	♂ 开发	∂ 运维	
工作空间 → 工程详情		代码空间	代码仓库 发版
S			
工程成员:			
□ 年40			
参与人员:			
历史版本 已发布应用			
4 This will be a first the first of the firs			2 小时前
1 million constraints and constraints and			
			2.1.0+00
			2 /1/4/ #0
f.			
2			7 小时前

您可以对工程进行**发版**,然后进入**应用空间**页面,部署相应的工程版本。

⑦ 说明 工程需要关联Git才能进行发版。

4.3. 版本管理

App Studio集成了通用的Git服务,本文将为您介绍在App Studio中如何使用VCS-git。

新建工程关联Git系统

- 1. 新建工程。
- 2. 录入用户基本信息。

关联Git操作前,需要首先录入用户基本信息。

打开已导入的Git工程,单击菜单栏中的设置,生成一个SSH Key,并根据提示添加到代码仓库所属的账 户公钥列表中。



⑦ 说明 新创建的工程默认未关联Git服务。如果需要Git服务,请关联当前项目至自己的Git仓库。

3. 新建Git仓库。

https	://code.aliyun.com	* 0	Θ 🔕 🗖	0
制台	产品与服务▼			alicoda,ciou
	ŧ1系统提圈:近日少数用户反馈对项目Internal权限存在错误理解风险,特此提圈请谨慎设置。			
	你的项目 星标项目 浏览项目		_	
	通过项目名称过滤		+ 9	建项目
	c Manual Angle Space			★0
	S			★0

新项目	
项目路径	http://code.aliyun.com/ alicode_cloud * / my-awesome-project
	希望将几个相关联的项目放置于同一个命名空间下?创建项目组
导入项目	○ GITHUB
描述 (可选)	
可见等级 (?)	● Private 项目必须明确授权给每个用户访问。
	● Internal 出于风控考虑, "internal"的克隆功能暂时关闭(代码库成员并不受影响), 敬请谅解 项目可以被所有已登录用户克隆。注意:设置该权限的项目内代码对所有登录本站(https://code.aliyun.com)的用户可见,请谨慎设置。
	○ ♀ Public 项目可以被任何用户克隆。 一些可见等级已被管理员限制。
创建项目	

4. 获取当前仓库的SSH地址。

				С			
				ching-template ching-template			
* 0	h 0	SSH	HTTPS	git@code.aliyun.com:alic	ß	Ŧ	+

- i. 单击SSH,即可获取当前仓库的SSH地址。
- ii. 单击右侧的复制按钮, 即可复制SSH地址至剪贴板。
- 5. 关联Git仓库。
 - i. 选择菜单栏中的版本 > 初始化&关联远程仓库。

	版本	查看	调试	
				≣
1				
lik mi				4
nt	初始化8	8关联远程	仓库	

ii. 填写关联远程仓库对话框中的Git地址,单击提交。

iii. 关联完成后,在App Studio页面的左侧导航栏增加版本控制标签。



iv. 单击版本下拉列表中的推送,即可将本地代码推送至远程仓库。

Git操作入口

Git相关的操作均集成在左侧的Git面板,以及顶端的版本控制菜单栏中。



Git控制面板

Git控制面板会动态更新文件的编辑状态。

	App Studio 工程 文件 编辑	版本 查看 调试 设置 帮助	Edit Config 🗸 🕨 🌺 🔳
ര	工程 這	👙 Main,java 🗙	T
ш [,]	git_demo1 (i)	1 package com.alibaba.dataworks;	
	IE ::::::::::::::::::::::::::::::::::::	<pre>Mahjava x package com.alibaba.dataworks; import org.springframework.boot.autoconfigure.EnableAutoConfiguration; import org.springframework.contSpringAppLication; import org.springframework.context.annotation.ComponentScan; /** * 主要, \LDE, d目要定SUVybatis@mapper * deuthor SQI * delet 2012/07/1 * delet 2012/07/1 * delet 2012/07/1 * delet 2012/07/1 * gedate 2012/07/1 *</pre>	Interest constraints
*			
746			

您可以在Git控制面板中,完成基本的 git add/rm/commit/revert 等操作。

Git基本操作

Git面板中以列表形式展示变动的文件,包括文件名、路径、以及右侧支持的基本操作。



如上图所示,红框中包含了可操作按钮,以及文件标识(icon)。

● 源代码管理

您可在此处进行commit、refresh、pull和push等操作。

- commit操作:选择√下的commit&push操作。
- refresh操作:单击乙,刷新当前控制面板内容,相当于执行git status,并刷新界面。
- pull/push操作:单击 ,根据自身需求选择拉取或推送。

● 暂存的更改

- **—**: 放弃所有修改,相当于执行git reset。
- ②: 文件数量标签。
- 🛃 SyncPaiApiClient.java src/main/java/co... M : 显示文件更改。
- 更改
 - 2: 放弃所有更改。
 - **-**:将所有文件添加至缓存区,相当于执行git add。
 - ②: 文件数量标签。
 - 显示条目包含的操作如下:
 - 🦻: 放弃修改(revert)。
 - **+**: 暂存更改(add)。
 - M: 文件改动标识(Modified)。

commit/push操作示例

示例一

APP Studio·功能介绍

Dat aWorks

6	App Studio 工程 文件 编辑	辑 版本 查看 调试 设置 帮助	Edit Config 🗸 🕨 🌺 🔳
n	工程 :	:Ξ 🐇 Main.java 🗙 👙 Result.java 🗙 👙 SyncPeiApiClient.java 🗙	
Ū V	If R :: git_demot ()	<pre>Mainjava X & Resultjava X & SyncPalApiClientjava X package con.alibaba.datoworks; import org.springframework.boot.autoconfigure.EnableAutoConfiguration; import org.springframework.boot.autoconfigure.SpringBootApplication; import org.springframework.context.annotation.ComponentScan; /** * #BM, \LDM, tbm#BEt2XMybatisPomepper * @uthor SQI * @date 2018/07/1 * @date 2018/07/</pre>	
\$	pom.xml		

示例二



? 说明

- Git客户端逻辑一致, 您需要主动调用push, 本地的代码才会推送至远程仓库。
- 与push同理, 您需要主动调用pull, 远程仓库的代码才会拉取至本地。

Branch管理

打开分支管理弹窗。



单击窗口下侧状态栏中显示的当前Branch名称,即可弹出Branch管理窗口。

新建本地分支



分支创建后,会自动切换至新创建的分支。

创建/切换/合并分支



⑦ 说明 新创建的本地分支,可直接推送至远程,远程分支名与本地一致。

通过Diff页面解决merge conflict

	6	App Studio				alicode_cloud	ф;
з	[程	文件 编辑 版本 查看	调试	设置 模	· 帮助 反馈	😇 i 👫 main 🤍 🕨 🎘	
a	3 2	原代码管理: GIT	· • •	. 🔺 Main	iava 💿 Main.java (Working tree) ×		
Ц	۳ ×	突	+ (8)		package com.alibaba.dataworks;	package com.alibaba.dataworks;	
		factorypath Jactorypath IndexController.class target/cla IndexController.java src/main/ja logback-spring.xml target/class Main.class target/classes/com/ Main.class target/classes/com/ Main.java src/main/java/com/ali pom.xml pom.xml #F#pJEX addOrder.santa santa/pages/dashboa. admin.santa santa/pages/dashboa.	フ + C フ + C - 15 - A		Import org.springframework.boot.autoconfigure.EnableAutoConfigu Import org.springframework.boot.springApplication; Import org.springframework.boot.autoconfigure.SpringBootApplica Import org.springframework.context.annotation.ComponentScan; /** * 主先、入口类 */ @formigBootApplication @formigBootApplication @formigBootApplication @formigBootApplication @formigBootApplication gformigBootApplica	import org.springframework.boot.autoconfigure.EnableAutoConfigu import org.springframework.boot.springApplication; import org.springframework.context.annotation.ComponentScan; /** * 主先、入口失 */ @foringBootApplication @forbubAutoConfiguration @forbubAutoConfiguration @forbubLextloan(basePackages = "com.alibaba.dataworks") public class Main { public static void main(String[] args){ <ccccce bar="" dddd");<br="" head.upint("foo="">System.out.print("foo bar dddd"); System.out.print("foo bar dddd"); System.foot.print("foo bar dddd"); System.foot.print("foo bar ddd"); System.foot.print("foo bar ddd"); System.foo</ccccce>	
		 business.santa santa/pages/report detaili.santa santa/pages/dataset/d DsClientConfig.class target/classe. DsClientConfig.java src/main/java/. home.santa santa/pages/dataset/h home.santa santa/pages/report/ho home.santa santa/pages/dashboar monitor.santa santa/pages/dashboar 			<pre>SpringApplication.run(Main.class , args) ; }</pre>	<pre>>>>>>> refs/heads/master >>>>>> refs/heads/master >>>>>> springApplication.run(Main.class , args) ; } </pre>	
		orderList.santa santa/pages/dashb				保存更改	
¢		🗊 OUT 🛛 💥 DEBUG 🛛 🗮 PROBI	LEM I	Terminal	Version Control		

Show History

右键单击文件,选择**Git > Show History**,即可查看当前文件的历史记录,对特定的commit与当前version 进行Diff。

	DataWorks	s				English
工程	文件 编辑	版本 查看 🕴	构建 调试	公置 模板 帮助 反馈		
n)	工程			Mainjava ×		
Ξ.				1 package con.alibaba.dataworks;	D STREET WEAKING	Sun
Ŷ				2 import ora, springframwork, boot. SpringApplication:		me
	▼ main			import or graphing inclusion to over pringing textuals in a second se		
Ÿ	▼ java ▼ com alib	aha datawarke		5 . eSerineBeatAndication		
	∮ Main	.java		 get ingoot spitiation public class Main { 		Silario
	 resources 			s aublic static wold anis(String[] area) /		l ľ
				<pre>public state watering in a gar (pringApplication.run(Main.class, args);</pre>		
	application.pro	perties				
	pom.xml					Jata
						a
						lavs
**						
¥						
	UT 🔆 DEBUG	PROBLEM 📓	Terminal P	/ Version Control		

Git Log

单击菜单栏中的**版本 > 日志**,打开Git log面板,即可查看提交的信息、时间、作者,您可以通过信息、分支、作者、时间筛选提交日志。

Log History				
message: keyword brand	:h: All 🗸 user: All	〜 date: All 〜 起始日	日期 - 结束日期 苗	> Demo
path: path keyword $ C$				初始化工程
commit	message	committer	time	b26b5b6 guonic @gmail.com> on 2019-03-26 18:58
b26b5b6	初始化工程		2019-03-26 18:58	In 1 branches: master Show All

4.4. 代码编辑

4.4.1. 代码编辑概述

代码编辑包括自动补全、智能提示、语法诊断和全局内容搜索等常见的IDE具备的功能。

Dat aWorks



目前语言和对应的功能支持情况,如下表所示。

基本功能	Java	Python	JavaScript/TypeScript
Completion自动补全	支持	支持	支持
Hover智能提示	支持	支持	支持
Diagnostics语法诊断提示	支持	支持	支持
Signat ureHelp函数参数 提示	支持	支持	支持
Definition跳转定义	支持	支持	支持
References查找引用	支持	支持	支持
Implementation查找实现 类	支持(comming soon)	不支持	不支持
Document Highlight变量 高亮	支持	支持	支持
DocumentSymbol查找类 成员	支持	支持	支持
WorkspaceSymbol全局 查找类/函数	支持	支持	支持
CodeAction修复建议	支持(Alibaba Java Guidelines is coming soon)	支持	支持
CodeLens行操作提示	References Implementation	不支持	不支持

APP Studio·功能介绍

基本功能	Java	Python	JavaScript/TypeScript
Formatting 格式化代码	支持	支持	不支持
RangeFormatting局部格 式化	支持	不支持	不支持
FindInPath全局内容搜索	支持	支持	支持
高级功能	Java	Python	JavaScript/TypeScript
Rename重命名	支持	支持	支持
WorkspaceEdit多文件修 改	支持	不支持	不支持
UnitTest单元测试 (quickstart)	支持	不支持	不支持
MainClass查找main函数 入口	支持	不支持	不支持
MainClassQuickStart快捷 运行	不支持	不支持	不支持
ListModules查找所有模 块	支持	不支持	不支持
Generate生成代码片段	Constructor Override Getter/Setter Implement	不支持	不支持

基本功能

• 自动补全

Dat aWorks

🔮 Wor	dCo	unt.java 🗙
		· · ·
		©Override
		<pre>public void reduce(Record key, Iterator<record> values, TaskContext context)</record></pre>
		throws IOException {
		long count = 0;
		while (values.hasNext()) {
		Record val = values.next();
		<pre>count += (Long) val.get(0);</pre>
		result.set(0, key.get(0));
		result.set(1, count);
		context.write(result);
		sublic static void main(String[] args) throws Evention (
		if (arcs length = 2) {
		System.er.println("Isage: WordCount <in table=""> <out table="">"):</out></in>
		System.exit(2):
		JobConf job = new JobConf();
		<pre>job.setMapperClass(TokenizerMapper.class);</pre>
		job.setCombinerClass(SumCombiner.class);
		job.setReducerClass(SumReducer.class);
		<pre>job.setMapOutputKeySchema(SchemaUtils.fromString("word:string"));</pre>
		job.setMapOutputValueSchema(SchemaUtils.fromString("count:bigint"));
104		<pre>InputUtils.addTable(TableInfo.builder().tableName(args[0]).build(), job);</pre>
		OutputUtils.addTable(TableInfo.builder().tableName(args[1]).build(), job);
107		JobClient.runJob(job);
	ı	
	r	
And the second		

● 智能提示



• 语法诊断

	<pre>@Resolve({"double->double"})</pre>
14	<pre>public class AggrAvg extends Aggregator {</pre>
15	private static class AvgBuffer implements Writable {
16	private double sum = 0;
17	<pre>private long count = 0;</pre>
18	@Override
19	<pre>public void write(DataOutput out) throws IOException {</pre>
20	<pre>out.writeDouble(sum);</pre>
	out.writeLong(count)
22	}
23	@Override
24	<pre>public void readFields(DataInput in) throws IOException {</pre>
25	<pre>sum = in.readDouble();</pre>
26	<pre>count = in.readLong();</pre>
	}
28	
29	<pre>private DoubleWritable ret = new DoubleWritable();</pre>
30	@Override
	<pre>public Writable newBuffer() {</pre>
32	return new AvgBuffer();
33	}
34	@Override

• 函数参数提示



• 跳转定义
🐇 Wo	rdCount.java 🗙	
	package com.aliyun.odps.open.example.mapred;	TIN.
	<pre>import java.io.IOException;</pre>	
	<pre>import java.util.Iterator;</pre>	
	<pre>import com.aliyun.odps.data.Record;</pre>	
	<pre>import com.aliyun.odps.data.TableInfo;</pre>	
	<pre>import com.aliyun.odps.mapred.JobClient;</pre>	
	<pre>import com.aliyun.odps.mapred.MapperBase;</pre>	and the second
	<pre>import com.aliyun.odps.mapred.ReducerBase;</pre>	
	<pre>import com.aliyun.odps.mapred.conf.JobConf;</pre>	
	<pre>import com.aliyun.odps.mapred.utils.InputUtils;</pre>	
	<pre>import com.aliyun.odps.mapred.utils.OutputUtils;</pre>	
	<pre>import com.aliyun.odps.mapred.utils.SchemaUtils;</pre>	
	public class WordCount {	
	public static class TokenizerMapper extends MapperBase {	
	private Record word;	
	private Record one;	
	*	
	@Override	
	<pre>public void setup(TaskContext context) throws IOException {</pre>	
	<pre>word = context.createMapOutputKeyRecord();</pre>	
	<pre>one = context.createMapOutputValueRecord();</pre>	
	<pre>one.set(new Object[] { 1L });</pre>	
	<pre>System.out.println("TaskID:" + context.getTaskID().toString());</pre>	
)	
	@Override	

● 查找引用



• 查找当前类成员



• 全局查找类/函数



• 代码格式化



4.4.2. UT测试

App Studio目前支持自动生成UT代码、检测UT测试入口、运行UT代码和展示运行结果等功能。

自动生成UT代码

打开相应文件后,右键单击代码编辑区,选择**Generate > Create Test**,即可在Test目录下自动生成测试 类和测试代码。



Dat aWorks

C La									
iCo (va Lt ing put	<pre>ontroller { alue = { "/tes testApi(){ g,String> resu t("name","apps f("description</pre>	Cons stapi Over It = Crea studio"); ""."welcome	Generate structor ride Method te Tests	s ,Stri lio"):	.ng> <u>(</u>				
工利	呈 文件 编	辑 版本	查看 计	调试 i	受置	模板	帮助	反馈	
3	工程			≣	🛓 A	piContro	ller.java	🛔 ApiControllerTest.java 🛛 🛛	
	 > .settings > src > main > java > co > > v co > > resou > test > java > co > > iava <li< td=""><td>m.alibaba.data common controller ~ api @ ApiCor ~ page @ IndexC Main.java rces m.alibaba.data controller.api ApiContro Sampierest.ja</td><td>nworks htroller.java Controller.java nworks pillerTest.java ava</td><td>a ,</td><td></td><td>impo Run pub</td><td>ort org.; ort stat: Test Lic <u>c</u>lass @Test Run Test public \ }</td><td>junit.Test; ic org.junit.Assert.*; s ApiControllerTest ₪ void testApi() {</td><td></td></li<>	m.alibaba.data common controller ~ api @ ApiCor ~ page @ IndexC Main.java rces m.alibaba.data controller.api ApiContro Sampierest.ja	nworks htroller.java Controller.java nworks pillerTest.java ava	a ,		impo Run pub	ort org.; ort stat: Test Lic <u>c</u> lass @Test Run Test public \ }	junit.Test; ic org.junit.Assert.*; s ApiControllerTest ₪ void testApi() {	

检测UT测试入口

? 说明

- UT类需要写在*src/test/java*目录下。如果Java UT类文件不在该目录下,将无法被正常识别成 Java UT类。
- @Test注解下的方法会出现Run Test的UT运行入口。

完成Java类的创建后,在对应的测试用例方法上添加 org.junit.Test 的@Test注解即可。



运行UT代码

单击右上角的运行按钮,即可运行UT的测试用例。

工利	程 文件 编辑 版本 查看 构建 调	试 设置 模板 帮助 反馈	🖁 Edit Config 🗸 🕨 🌉 🔳
n	工程 🗄	🔬 Main.java 🗴 🔬 SampleTest.java 🗙	
יש	TestUT (j)	7 * Sample Java Test Class	
P	▼ src		Substation on O
	▼ main	Run lest	
	▼ iava		
	✓ com.alibaba.dataworks	11 @Test	
	🔄 Main java	Run Test	Sha
		12 public void testPass() {	с ^о
		<pre>13 System.out.println("hello world!");</pre>	
		14 }	
	✓ test	15 aTest // 川下远行	
	✓ java	Run Test	
*	✓ com.alibaba.dataworks	17 public void testFail()	
		18 Assert.assertEquals(true, false);	
输出			中止运行按钮
	.pom 2013-01-12 22:50:21.903295 :: [INF0] DownLo pom (3.4 kB at 379 kB/s) 2013-01-12 22:50:21.903558 :: [INF0] DownLo red-utils=-0.1.pom 2013-01-12 22:50:21.915237 :: [INF0] DownLo ed-utils=-0.1.pom (4.0 kB at 674 kB/s) 2013-01-12 22:50:21.9124701 :: [INF0] DownLo =shared-components=-18.pom 2013-01-12 22:50:21.924701 :: [INF0] DownLo shared-components=-18.pom (4.9 kB at 705 kB/ 2013-01-12 22:50:21.924701 :: [INF0] DownLo en-shared-incremental-1.1.pom (919-01-12 22:50:21.93466 :: [INF0] DownLo -shared-incremental-1.1.pom (4.7 kB at 677 2015-01-12 22:50:21.934966 :: [INF0] DownLo -shared-components-19.pom	<pre>ded from nexus-tbmirror: http://100.67.148.246/repository/maven-public/org/apache/maven/maven-toolcha ding from nexus-tbmirror: http://100.67.148.246/repository/maven-public/org/apache/maven/shared/maven- ded from nexus-tbmirror: http://100.67.148.246/repository/maven-public/org/apache/maven/shared/maven- ding from nexus-tbmirror: http://100.67.148.246/repository/maven-public/org/apache/maven/shared/maven- ded from nexus-tbmirror: http://100.67.148.246/repository/maven-public/org/apache/maven/shared/maven- ding from nexus-tbmirror: http://100.67.148.246/repository/maven-public/org/apache/maven/shared/maven- ding from nexus-tbmirror: http://100.67.148.246/repository/maven-public/org/apache/maven/shared/maven- ding from nexus-tbmirror: http://100.67.148.246/repository/maven-public/org/apache/maven/shared/maven- ding from nexus-tbmirror: http://100.67.148.246/repository/maven-public/org/apache/maven/shared/maven- k8/s) ding from nexus-tbmirror: http://100.67.148.246/repository/maven-public/org/apache/maven/shared/maven- k8/s)</pre>	in/1.0/maven-toolchain-1.0. -shard-utils/0.1/maven-shar shard-utils/0.1/maven-shar -arced-components/1//maven- shared-incremental/1.1/mave shared-incremental/1.1/mave shared-incremental/1.1/mave
. سے ح	Running tests		许爱琳(wb-xal250971)WB250971

展示UT运行结果

Dat aWorks

工利	· 文件 编辑 版本 查看 构建 调试	式 设置 模板 帮助 反馈	🛃 👫 Edit Config 🗸 🕨
ð	工程 ∷Ξ TestUT ①	▲ Main.java × ▲ SampleTest.java × 7 * Sample Java Test Class	
*	 src main java com.alibaba.dataworks Main.java resources application.properties test java 	8 */ Run Test ※ 9 public class SampleTest (每个UT方法的运行结果, 10 @Test Run Test ♥ 12 public void testPass// { 13 System.out.priftln("hello world!"); 14 } 15 @Test	1 Passed 1 Failed 0 Skipped ¥細报告 Method Status Duration + com.alibaba.datawork Passed Oms - com.alibaba.datawork Failed 3ms
\$	✓ com.alibaba.dataworks SampleTest.java target	Run Test X 17 public void testFail() [] 18 Assert.assertEquals(true, false); 19] 20 }	Message: expected: <true> but was:<false> Stack trace:</false></true>
輸出	2019-01-12 22:50:25.045713 :: €€ <testrunner- 2019-01-12 22:50:25.045767 :: hello world1 2019-01-12 22:50:25.045767 :: hello world1 2019-01-12 22:50:25.046176 :: €€<testrunner- 2019-01-12 22:50:25.046176 :: €€<testrunner- 2019-01-12 22:50:25.061485 :: EEXCUTE UT SUC 2019-01-12 22:50:25.061485 :: EEXCUTE UT SUC 2019-01-12 22:50:25.1470141 :: (UTIOND OSE)=- 2019-01-12 22:50:25.205078 :: (RESULT_URI) = 2019-01-12 22:50:25.755838 :: ●FASS 2019-01-12 22:50:25.751893 :: ●FASS 2019-01-12 22:50:25.751895 :: ●FASS 2019-01-12 22:50:25.751895 :: ●FASS</testrunner- </testrunner- </testrunner- 	<pre>("name":"testStarted", "attributes":{"name":"com.alibaba.da ec ret = 0 ("name":"testFinished", "attributes":{"name":"com.alibaba.d ("name":"testFinished", "attributes":{"name":"com.alib ("name":"testStummary", "attributes":{"massage":"Total tests CRSSFULXy ad Done processing folder /home/webide/release loaded: ut.log > SBT kbuild/result/1624-1547304616530/ut.log : {"message":null,"code:"1200, "success":true,"data":1,"time 运行结果简报,点击此处可显示出右边的详细报告 one processing fonter/successing/success</pre>	<pre>java.lang.AssertionError: expected:<true> but was:<fal at="" java.lang.reflect.method.invoke(method5].r="" org.junit.assert.assertequals(assert.java:="" org.junit.assert.fail(assert.java:88)="" org.junit.assert.fail(notequals(assert.java:="" org.junit.runners.blockjunit4classfundedendedendedendedendedendedendedended<="" org.junit.runners.model.frameworkmethod5i.r="" org.junit.runners.parentrunner.runleaf(pare="" org.lett.nativemethodaccessorimpl.invoke="" sun.reflect.nativemethodaccessorimpl.in="" sun.reflect.nativemethodaccessorimpl.invoke="" th=""></fal></true></pre>

4.4.3. 生成代码片段

目前App Studio在Java语言中,支持生成Java类的构造函数(Constructor)、Getter函数、Setter函数,也 支持生成该类所继承父类的Override方法、所要实现的接口方法等。

功能入口

目前Java的代码生成入口有以下两种:

• 鼠标右键单击代码区域,选择Generate。

6	🛆 Арр	Studio											
ធ	工程	文件	编辑	版本	查看	调试	设置	发布	模板	帮助	反馈		
ŋ	工程					4	Main.ja	ava ×					
	DM2222 > .alicode > .settings > APP-MET > santa V src V main V ja	Va / com.alib > comr > contri > contri > contri	aba.datan non guration oller.page	works				Liconcos con thi The (the Li the Lice <u>htt</u> Unless r distribu WITHOUT See the limitati	tense (); ense (); ense () cp:// Go requi rted WARF Fin Lice Wa Go	you may u may ob to Definiti ek Definitie d All Refer orkspace Sy to Symbo	not use tain a c ion ences ymbol	Equadation No ard the copy of the Ctrl+F12 Alt+F12 Shift+F12 Ctrl+P Ctrl+Shift+O	(ASF) DTICE ing of Apao excep Licer to : on , e: vern
		🛓 Mai	n.java			< 19			Gei	nerate		Ctrl+M	
	 > re > target 	sources hth path re lio.deploy.	properties	5) im 1 im 2 im 3 im 5 /** 5 /* 5 /* 6 (\$) 8 (\$) 9 (\$	port org. port org. port org. * 主类,入[/ pringBoot	spri spri spri Rer spri Chi 口类 For Appl Cut	name Sym ange All O mat Docu t	bol ccurrence ment	F2 s Ctrl+F2 Shift+Alt+F	nab] on; prir ompo
	E LICENSI	E) pul 1 2 3 4 }	blic clas public Spr }	s Ma Co stat ring4 Co	py mmand Pa	lette	F1);

• 通过快捷键Ctrl+M进入。

Constructor

进入Generate Code面板后,选择Constructor。



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。

	Generate		
	Constructor		
	Getter		
	Setter		
	Getter And Setter		
	Override Methods	etaTableRequest.	
10 10	Create Tests	etaTableResponse; aDBRequest;	
	2 33 88 加里诺	如如米沙方在何今的	

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,则没有图中的 三个选项,因为此时该类不需要生成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 🗙						
	1 package co	om.alibaba.dataworks.	udf;						
	3 import con	n.aliyun.odps.udf.UDF	;						
	4 import con	at.ILower;							
	6 public fir	6 public final class Lower extends UDF implements							
		~~~~~							
	8 privat	te int id;							
	9 privat	te String name;							
	10								
	11 public	C Lower(int id, String	g name) {						
		115.10 = 10;							
	15								
	16 <b>@0ver</b>	ride							
	17 public	c void close() throws	UDFException {						
	18   SI	<pre>uper.close();</pre>							
	19 <b>}</b>								
	20 21 <b>00</b> ver	ride							
	22 public	<pre>String toString() {</pre>							
	23 <b>r</b> e	eturn super.toString(	);						
	24 }								
	25								
	26 @Overn	ride							
	27 public	c void interfaceHeigh	t(int id) {						
	28 20 <b>1</b>								
	30								

### **Create Test**

进入Generate Code面板后,选择**Create Test**,即可在Test目录下自动生成测试类和测试代码。详情请参 见<mark>UT测试</mark>。

# 4.4.4. 全文内容搜索

App Studio支持全文内容搜索功能。

App Studi	0							
工程 文件	编辑	版本	查看					
工程	撤销		Ctrl Z					
ddd ⑦ > .alicode	恢复		Ctrl Y					
> .settings	剪切		Ctrl X					
> APP-META > santa	复制		Ctrl C					
✓ src	查找		Ctrl F					
∨ main >iava	替换		Ctrl H					
> resources	全文搜	秦 Ctrl	Shift F					
∨ test ∨java	向上复	主文機業() 制一行Shi	ft Alt †					
∨ com.alil	向下复	制—行Shi	iftAlt↓ le					
🛓 Der	向上移	动一行	Alt ↑					
✓ target ▲ .classpath	向下移	动一行	Alt ↓					
≣ .factorypath								
<ul> <li>.gitignore</li> </ul>	.gitignore     .							
<ul> <li>project</li> <li>pom.xml</li> </ul>								
	<ul> <li>              App Studi</li></ul>	文件       編輯         工程       文件       編輯         工程       撤销         ddd ⑦       恢复         > .alicode       恢复         > .alicode       類切         > .settings       剪切         > APP-META       复制         > santa       星制         > src       查找         > java       音文理         > resources       全文理         < test       向上复         < test       向上复         < locasspath       向下移:         .classpath       向下移:         .classpath       の下移:         .classpath       「下移:         .classpath       「下移:         .classpath       「下移:         .classpath       」.gitignore         .classpath       .gitignore        gitignore       .gitignore	工程       文件       編輯       版本         工程       歳相       版本         工程       撤销       板本         ddd ①       快复       第切         > .alicode       好       第切         > .alicode       現       日本         > .alicode       日本       日本         > .giva       日本       日本         > target       日本       日本         ①       .classpath       日本         正       .factorypath       日本         .gitignore       王       .project         .pom.xml       日本       .project					

#### 鼠标悬停至顶部菜单栏中的编辑,单击全文搜索。

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。



#### 支持根据**项目、模块**和目录进行查找。

全文搜索 🗌 大小写四	記 单词匹配 正则匹配 文件	后缀 请选择 💙
Q Contro 项目 模块 目录	ම ර src/main	
package com.alibaba.dat import org.springframew @Controller public class IndexContro package com.alibaba.dat	<ul> <li>.settings</li> <li>src</li> <li>main</li> <li>java</li> <li>resources</li> <li>test</li> <li>target</li> </ul>	IndexController java 1 IndexController java 3 IndexController java 10 IndexController java 10 ApiController java 1 ApiController java 1
@Rest <mark>Contro</mark> ller public class Api <mark>Contro</mark> lle		
	Cancel	

选中文件后,可以定位到文件中的搜索内容,并在编辑器内打开该文件。



# 4.5. 调试

# 4.5.1. Config配置及启动

您可通过配置入口函数,单击调试、断点等步骤,进行程序的调试。

配置入口函数

#### Dat aWorks

#### APP Studio·功能介绍

¢	→ C ① 不安全	alicode.	aliyun.test	/#/										☆ 📴	🔁 📀	<u> </u>	Ø :
6	App Studio	工程	文件	编辑	版本	查看	调试	设置	帮助					main	$\vee$	×	E
	App Studio T程 demots() + target * src main + resources + java * santa + pages + index.html pom.xml	1	x(# Run/Debi + x main	setti ∷≘	版本		调武 Nami VM c Prog Ervir	设置 e: main in class: ptions: ram argume Var	帮助 ants: riables:	com.alibaba.demo.Main 🖕	Cancel	Apply	×	main		~	
*							4 88										
	配直					is	记明										
	MainClass					悠	欧可以	从多	个配	置中选择需要启动的main函数	文。						

VM options您可以配置在JVM启动时,例如-D-Xms-Xmx等配置。Program arguments您可以添加启动参数,此参数会被main函数的args参数接收。Environment环境变量参数。PORT端口,表示本程序需要暴露的端口信息,例如springboot经典的7001、8080等端 口。机器您可以选择需要的机器配置进行调试。2vCPU, 4g构存 2vCPU, 2c여存 4vCPU, 3c여存 4vCPU, 8c两存HotCode此配置仅在run模式下生效,默认使用公司的HotCode2插件进行启动。	MainClass	您可以从多个配置中选择需要启动的main函数。						
Program arguments您可以添加启动参数,此参数会被main函数的args参数接收。Environment环境变量参数。PORT端口,表示本程序需要暴露的端口信息,例如springboot经典的7001、8080等端 口。水晶晶子 化器您可以选择需要的机器配置进行调试。2vCPU,4c内存 2vCPU,4c内存 4vCPU,8c内存HotCode此配置仅在run模式下生效,默认使用公司的HotCode2插件进行启动。	VM options	您可以配置在JVM启动时,例如-D -Xms -Xmx等配置。						
Environment环境变量参数。PORT端口,表示本程序需要暴露的端口信息,例如springboot经典的7001、8080等端 口。水森森您可以选择需要的机器配置进行调试。2vCPU,4G内存 2vCPU,2G内存 4vCPU,8G内存 4vCPU,8G内存HotCode此配置仅在run模式下生效,默认使用公司的HotCode2插件进行启动。	Program arguments	您可以添加启动参数,此参数会被main函数的args参数接收。						
PORT         端口,表示本程序需要暴露的端口信息,例如springboot经典的7001、8080等端 口。           您可以选择需要的机器配置进行调试。         2vCPU,40内存           2vCPU,26内存         2vCPU,40内存           4vCPU,86内存         4vCPU,86内存           4vCPU,86内存         4vCPU,86内存	Environment	环境变量参数。						
机器         您可以选择需要的机器配置进行调试。           2vCPU, 4G内存         ^           2vCPU, 2G内存         2vCPU, 2G内存           2vCPU, 3G内存         4vCPU, 8G内存           4vCPU, 8G内存         1	PORT	端口,表示本程序需要暴露的端口信息,例如springboot经典的7001、8080等端 口。						
1       2vCPU, 4G内存       へ         2vCPU, 2G内存       2vCPU, 2G内存       く         ✓ 2vCPU, 4G内存       √       4vCPU, 8G内存         HotCode       此配置仅在run模式下生效,默认使用公司的HotCode2插件进行启动。		您可以选择需要的机器配置进行调试。						
<ul> <li>机器</li> <li>2vCPU, 2G内存</li> <li>2vCPU, 4G内存</li> <li>4vCPU, 8G内存</li> <li>HotCode</li> <li>此配置仅在run模式下生效,默认使用公司的HotCode2插件进行启动。</li> </ul>		2vCPU, 4G内存 ^						
<ul> <li>✓ 2vCPU, 4G内存</li> <li>4vCPU, 8G内存</li> <li>HotCode</li> <li>此配置仅在run模式下生效,默认使用公司的HotCode2插件进行启动。</li> </ul>	机器	2vCPU, 2G内存						
4vCPU, 8G内存         HotCode         此配置仅在run模式下生效,默认使用公司的HotCode2插件进行启动。		✓ 2vCPU, 4G内存						
HotCode 此配置仅在run模式下生效,默认使用公司的HotCode2插件进行启动。		4vCPU, 8G内存						
	HotCode	此配置仅在run模式下生效,默认使用公司的HotCode2插件进行启动。						

### 启动调试

#### 选择菜单栏中的调试 > 启动调试。

6	App Studio	工程	文件	编辑	版本	童看	调试	19 <b>2</b>	帮助		main	V I	
<b>F</b> N	工程				🔬 Mair	n.java ×	启动调试						20
<u> </u>	demotest $(\bar{i})$						停止				INSER.	181Gaz	untir
¥	<ul> <li>target</li> <li>src</li> <li>main</li> <li>resources</li> <li>java</li> <li>com.ali</li> <li>com</li> <li>cont</li> </ul>	baba.demo non roller				import or import or import or import or /** * 主美, ) * @autho * @date	(重启调试 运行 打开配置 添加配置 继续			configure.EnableAutoConfiguration; ygApplication; :onfigure.SpringBootApplication; unotation.ComponentScan;			me Share
	▶ servi Mai	ce n.java				*/ @SpringBo	单步跳过						41
	<ul> <li>santa</li> <li>pages</li> <li>index.html</li> <li>pom.xml</li> </ul>				14 15 16 17 18 19 20 21	@Component public cl publi S	单步执行 强制进入 单步跳出		F7 √ F7 ☆ F8	.ibaba.demo") args){ ass, args);			

因为需要为您准备运行环境和下载mvn依赖,第一次启动时速度较慢。重启调试会跳过此步骤,启动速度逐 渐接近本地编辑器的体验。

	App Studio 工程 文件 编辑	本 查看 调试 设置 帮助	main 🗸 🕨 🌉
ß	工程 🗄	ndexController.java 🗶 🎂 Result.java 🗶	
U'	demo5 (i)	package com.alibaba.demo.common:	
	k canta		
Ÿ		import java.io.Serializable;	
	• src		The second
	✓ main	5 public class Result <t> implements Serializable {</t>	man and a second s
	✓ java	<pre>private static final long serialVersionUID = 7154887528070131284L;</pre>	
	<ul> <li>com.alibaba.demo</li> </ul>	7 private String message;	
	▶ common	private Integer code;	Normal Sector Se
	▼ controller	private Boolean success;	Martine
	▼ api.demo	<pre>private Long timestamp = System.currentTimeMillis():</pre>	
	OssDemoController java	private String sessionId:	- Mail statumer
		private Integer errCode;	alle strange
	Demoxpicontroller.java	4 private String errMsg;	a Definition and a second s
	✓ page	5 private String requestId;	
	🛓 IndexController.java		a District and a second s
	service	7 public Result() {	Parameter - Parame
	🛓 Main.java	8 }	Ran year.
	✓ resources	y authic static Decult of East (Chains and Tatana and ) [	New York
	▼ templates	public static Result oferfor(string msg, integer code) {	-Acceptions
	error.html	2 }	1 Martin 1 M
	o index.html		
	Joshack-enring yml	<pre>public static Result ofError(String msg) {</pre>	Notage Version
		<pre>5 return of(msg, Code.ERROR.code, (Object)null, false);</pre>	P, appp, 2, et al.
	2 application.properties		Hanny server.
	▶ target		ing generation , we way the second se
	s pom.xml	public static Result ofError(String msg, String sessionId) {	
		return of(msg, Code.ERROR.code, (Object)hull, Talse, Sessionid);	Plagger Access
			and the state of the
		<pre>public static <t> Result<t> ofSuccess(T data) {</t></t></pre>	
		return of((String)null, Code.SUCCESS.code, data, true);	
			Bigger som er sammen av sammen som sammen
-		<pre>5 public static <t> Result<t> ofBaseSuccess(T data) {</t></t></pre>	
*		<pre>7 return ofBase((String)null, Code.BASE_SUCCESS.code, data, true);</pre>	

# 4.5.2. 在线调试

在线调试支持Java Application和基于SpringBoot的Web工程。

进行在线调试前,首先要配置入口函数和启动调试,完成上述步骤后,进行后续操作。

### 透出服务

程序成功启动后,会提供两个基本服务,您可以单击后端链接,对后端Java代码进行调试。



### 面板介绍

• 输出面板

输出面板会显示所有程序的标准输出(暂不支持System.in),支持ansi颜色,体验与本地终端基本一致。

輸出								Cii		×		
s]]* onto public org.springframwork.http.ResponseEntity <jsva.lang.object> org.springframwork.data.rest.webmvc.RepositorySearchController.headTorflearch(org.springframwork.data.rest.webmvc.RepositorySearchController.headTorflearch(org.springframwork.data.rest.webmvc.RepositorySearchController.headTorflearch(org.springframwork.data.rest.webmvc.RepositorySearchController.headTorflearch(org.springframwork.data.rest.webmvc.RepositorySearchController.headTorflearch</jsva.lang.object>												
reerintos	cina caron, j	ava. Lang. of	cr rug,	· · · ·					A			
2018-08-	-14 16:07					101	INPO .	0.8	d.r.w.hasePathAwaremandlerwapping - Mapped "((/profile),methods=[GHT])" onto org.springframework.http.mttpmtityco	g.springiramework.nateoas.		
Resource	sSupport>	org.spring	greane	MORK	.data	.T081	c.web	eve.1	ProfileController.listAllFormsOfMetadata()			
2018-08								0.5	.d.r.w.BasePathAwareHandlerMapping - Mapped "{[/profile],methods=[OPTIONS]}" onto public org.springframework.http.Ht	tpEntity org.springfram		
ework.de	sta.rest.	webmvc.Prod	EileCo	ontro	ller.	profi	ile0p	tions	a()			
2018-08-								0.5	.d.r.w.BasePathAwareHandlerMapping - Mapped "{[/profile/{repository}],methods=[GE7],produces=[application/alps+ison	*/*1)" onto org.springf		
ramework ourceInt	k.http.Ht	tpEntity <on< td=""><td>rg.sp</td><td>ringf</td><td>ranew</td><td>ork.</td><td>data.</td><td>rest.</td><td>.webmvc.RootResourceInformation&gt; org.springframework.data.rest.webmvc.alps.AlpsController.descriptor(org.springframework)</td><td>k.data.rest.webmvc.RootRes</td></on<>	rg.sp	ringf	ranew	ork.	data.	rest.	.webmvc.RootResourceInformation> org.springframework.data.rest.webmvc.alps.AlpsController.descriptor(org.springframework)	k.data.rest.webmvc.RootRes		
2018-08-							INFO	0.5	.d.r.w.BasePathAwareHandlerHapping - Mapped *{{/profile/{repository}},methods=[0P710NS],produces=[application/alps+	son]}" onto org.springfram		
ework.ht	ttp.EttpE	ntity<7> or	rg.sp	ringf	Tamew	ork.c	data.	rest.	.webmvc.alps.AlpsController.alpsOptions()			
2018-08-								0.5	.d.r.w.BasePathAwareHandlerHapping - Mapped "{[/profile/{repository}],methods=[GE7],produces=[application/schema+jsc	m]}" onto public org.sprin		
gframewo	ork.http.	HttpEntity-	corg.	prin	gfran	ewori	k.dat	a.res	st.websvc.json.JsonSchema> org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.springframework.data.rest.websvc.RepositorySchemaController.schema(org.spri	lata.rest.webmvc.RootResour		
ceInform	nation)											
2018-08-								0.5	.j.e.a.AnnotationHBeanExporter - Registering beans for JMX exposure on startup			
2018-08-								s.b.	.c.e.t.TomcatEmbeddedServletContainer - Tomcat started on port(s): 7001 (http)			
2018-08-								008.	alibaba.demo.Main - Started Main in 4.562 seconds (JVM running for 5.674)			
0												

● 调用堆栈



断点

断点面板为您展示当前设置的所有断点,后续将为您介绍断点类型及使用。

输出调用	目堆栈 断点	ā į)	► ¥	7 3	<u> </u>	T		
+ - 0							IndexContr	roller.java18
🗸 🗹 Java	Line Break	points					Enabled	
🔽 Inc	dexControlle	er.java:18					Condition	请选择
📃 Java E	xception Bre	eakpoints						
📃 Java M	lethod Breal	kpoints						
DEBUG	PROB	LEM						

#### • PROBLEM

如果程序遇到编译问题,会展示在PROBLEM面板上,您可通过单击跳转至对应的文件行。

	问题	
	src/mai	n/java/com/alibaba/demo/common/Result.java
	0	Warning:(41 ,18) Result is a raw type. References to generic type Result <t> should be parameterized</t>
	0	Warning:(45 ,18) Result is a raw type. References to generic type Result <t> should be parameterized</t>
	0	Warning:(49 ,18) Result is a raw type. References to generic type Result <t> should be parameterized</t>
	0	Warning:(70 ,8) Result is a raw type. References to generic type Result <t> should be parameterized</t>
	0	Warning:(70 ,28) Result is a raw type. References to generic type Result <t> should be parameterized</t>
	0	Warning:(71,8) Type safety: The method setData(Object) belongs to the raw type Result. References to generic type Result <t> should be parameterized</t>
	0	Warning:(72 ,15) Type safety: The expression of type Result needs unchecked conversion to conform to Result <t></t>
	0	Warning:(76 ,8) Result is a raw type. References to generic type Result <t> should be parameterized</t>
3	DEBUG	PROBLEM

### 断点介绍

App Studio支持普通行断点、函数断点和异常断点,详情请参见断点类型。

### 调试按钮

调试界面如下所示:

←	→ ℃ ① 不安:	≌ pre-stu	idio.data.a	iliyun.com	/#/							🛱 🔛 🖼 🧐 📲 🖉 🗄	
6	App Studio	工程	文件	编辑	版本	查看	调试	设置	帮助	b		main 🗸 🕨 🌺 🔳	
n	工程				🔬 Index	Controller	.java ×	pom.xml				项目已经启动,访问:	
יים	demo (i)					xml v</th <th>ersion="1.</th> <th>0" encodin</th> <th>g="UTF</th> <th>-8"?&gt;</th> <th>A CONTRACTOR OF A CONTRACT OF</th> <th>前端: http://gateway.studio.data.</th> <th>ñ</th>	ersion="1.	0" encodin	g="UTF	-8"?>	A CONTRACTOR OF A CONTRACT OF	前端: http://gateway.studio.data.	ñ
						<project< th=""><th>t xmlns="</th><th>ttp://mave</th><th>n.apac</th><th>he.org/POM/4.0.0"</th><th>Contraction</th><th>dsgzuint6k0/8080/</th><th>ime</th></project<>	t xmlns="	ttp://mave	n.apac	he.org/POM/4.0.0"	Contraction	dsgzuint6k0/8080/	ime
Ŧ	+ sic						xmlns:xs	i="http://	www.w3	.org/2001/XMLSchema-instance"	Sectores.	后端: http://gateway.studio.data.	
	▼ main						xsi:sche	maLocation	=" <u>http</u>	://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-	The control of control of the contro	aliyun.com/pcsoxspre680ac8f9tb	l.
	✓ java					<mc< th=""><th>delVersior</th><th>&gt;4.0.0<th>delVer</th><th></th><th>Contraction and and</th><th>dsgzuint6k0/7001</th><th>s</th></th></mc<>	delVersior	>4.0.0 <th>delVer</th> <th></th> <th>Contraction and and</th> <th>dsgzuint6k0/7001</th> <th>s</th>	delVer		Contraction and and	dsgzuint6k0/7001	s
	▼ com.ali	baba.demo				<pa <pa a a a</pa </pa 	ickaging>ja	ir <th>ng&gt;</th> <th>d-</th> <th>10 5925 mile</th> <th></th> <th>han</th>	ng>	d-	10 5925 mile		han
	comr	mon				groupi		/artifactT	d d		Tillen		0
	✓ contr	roller				<ve< th=""><th>rsion&gt;1.0.</th><th>Ø-SNAPSHOT</th><th><th></th><th></th><th></th><th></th></th></ve<>	rsion>1.0.	Ø-SNAPSHOT	<th></th> <th></th> <th></th> <th></th>				
	▶ ap	i.demo											
	▼ pa	ge					pendencyMa	nagement>					
		IndexContro	oller.iava										
	▼ servi	ce .					<depe< th=""><th>endency&gt;</th><th></th><th></th><th></th><th></th><th></th></depe<>	endency>					
	▶ im	nl						groupId>or	g.spri	ngframework.boot			
	·	pi InnConvine in						artifactio	>sprin	g-boot-dependencies			
	<u>_</u> 0	ssoervice.ja	va •					type>pom </th <th>5.12.H</th> <th>ELEASE</th> <th></th> <th></th> <th></th>	5.12.H	ELEASE			
	<u>≜</u> P	aiApiService.	.java					scone>impo	rt <th>000</th> <th></th> <th></th> <th></th>	000			
	🔬 Mai	n.java					<th>endency&gt;</th> <th></th> <th></th> <th></th> <th></th> <th></th>	endency>					
	resources						<th></th> <th></th> <th></th> <th></th> <th></th> <th></th>						
*	target						lependency!	lanagement>					
*	» pom.xml					<de< th=""><th>pendencies</th><th></th><th></th><th></th><th></th><th></th><th></th></de<>	pendencies						
输	出 调用堆栈 断点				<b>G</b>							×	<
}" 0	nto public org.sp	ringframew	ork.http.	Responsel	Intity <j< td=""><th>ava.lanç</th><th>.Object&gt; o</th><th>rg.springf</th><th>amewo</th><td>rk.data.rest.webmvc.RepositorySearchController.headForSearch(org.sp</td><td>pringframework.</td><td>data.rest.webmvc.RootResourceI</td><td>j</td></j<>	ava.lanç	.Object> o	rg.springf	amewo	rk.data.rest.webmvc.RepositorySearchController.headForSearch(org.sp	pringframework.	data.rest.webmvc.RootResourceI	j
2018	-08-15 14:04:19.5	String)			o.s.d	.r.w.Bas	ePathAware	HandlerMap	ping	- Mapped "{[/profile].methods=[GET]}" onto org.springframework.ht	tp.HttpEntity4	org.springframework.hateoas.Re	
sour	ceSupport> org.sp	ringframew	ork.data.	rest.webm	wc.Prof	ileContr	oller.list	AllFormsOf	letadat	sa()			
2018	-08-15 14:04:19.5 data.rest.webmvc.	74 [ ProfileCon	troller.r	main] INFC	o.s.d	.r.w.Bas	ePathAware	HandlerMap	ping	- Mapped "{[/profile], methods=[OPTIONS]}" onto public org.spring!	ramework.http.	<pre>HttpEntity<?> org.springframew</pre>	
				ain] INFC	0.s.d	.r.w.Bas	ePathAware	HandlerMap	ping	- Mapped "{[/profile/{repository}],methods=[GET],produces=[applic	ation/alps+jsc	n    */*]}" onto org.springfra	
eInf	rk.http.HttpEntit ormation)	y <org.spri< td=""><td>ingframewo</td><td>rk.data.r</td><td>rest.web</td><th>mvc.Root</th><th>ResourceIn</th><th>formation&gt;</th><th>org.si</th><td>pringframework.data.rest.webmvc.alps.AlpsController.descriptor(org.</td><td>springframewor</td><td>k.data.rest.webmvc.RootResourc</td><td>l</td></org.spri<>	ingframewo	rk.data.r	rest.web	mvc.Root	ResourceIn	formation>	org.si	pringframework.data.rest.webmvc.alps.AlpsController.descriptor(org.	springframewor	k.data.rest.webmvc.RootResourc	l
2018-	-08-15 14:04:19.5	75 [	noframewo	main] INFC	o.s.d	.r.w.Bas	ePathAware	HandlerMap	ping Intion	- Mapped "{[/profile/{repository}],methods=[OPTIONS],produces=[aga()]	plication/alps	+json]}" onto org.springframew	
2018	-08-15 14:04:19.5	76 [		main] INFC	o.s.d	.r.w.Bas	ePathAware	HandlerMap	ping	- Mapped "{[/profile/{repository}],methods=[GET],produces=[applic	cation/schema+j	[son]}" onto public org.springf	
form	work.http.HttpEnt ation)	ity <org.sp< td=""><td>pringframe</td><td>work.data</td><td>.rest.w</td><th>ebmvc.js</th><th>on.JsonSch</th><th>ema&gt; org.sp</th><th>pringfi</th><td>ramework.data.rest.webmvc.RepositorySchemaController.schema(org.spr</td><td>ingframework.d</td><td>ata.rest.webmvc.RootResourceIn</td><td></td></org.sp<>	pringframe	work.data	.rest.w	ebmvc.js	on.JsonSch	ema> org.sp	pringfi	ramework.data.rest.webmvc.RepositorySchemaController.schema(org.spr	ingframework.d	ata.rest.webmvc.RootResourceIn	
					o.s.j	.e.a.Anr	otationMBe	anExporter		- Registering beans for JMX exposure on startup			Г
2018 2018 П			п	ain] INFC ain] INFC	com.a	libaba.d	icatEmbedde lemo.Main	dServletCo	itainei	<ul> <li>Tomost started on port(s): 7001 (http)</li> <li>Started Main in 4.51 seconds (JVM running for 5.873)</li> </ul>			ſ
- -	DEBUG E PROBL	.EM											ſ

上图中从左到右的每个按钮所代表的含义如下:

功能	说明
continue	恢复当前断点时,当前线程继续运行。
step over	执行到下一行。
step in	进入函数。

功能	说明
force step in	强制进入函数,与step in的区别在于,它可以引导断点 执行到Java自带的类库中。
step out	从当前函数跳出。
restart	目前的restart的实现方式较为简单(可能无法完成程序 清理等工作),正在优化中。
stop	停止。

#### 操作的快捷键如下:

快捷键参考	
快捷键	功能
τD	启动调试
<del>ሄ</del> F2	停止
Ctrl R	重启调试
⊂ R	运行
F9	继续
F8	单步跳过
F7	单步执行
℃ F7	强制进入
<b>① F8</b>	单步跳出

# 4.5.3. 断点类型

App Studio支持普通行断点、函数断点和异常断点三种断点类型。

普通行断点

通过单击文件行号前的空白区域,可以生成针对该行的断点,同时断点面板中会显示该断点。

\$	App Studio	工程 文件	牛编辑	版本	查看	调试 设置	帮助	D .		main 🗸 🕨 🔆	
0	工程 demo5 () > sarta * src * main * java * com.alibab * com.alibab	a.demo ar inno issDemoControl issDemoControllor,ja dexController,ja	ilerjava erjava NVa	✓ Inde 5 6 7 8 9 10 11 12 13 14 15 16 16 17 17 18 19 20 21 22	xControllerjava import org.s * ***********************************	<pre>x Result. pringframework 0I 8-08-14 IndexControll ing(value = { tring index(Mc rm "index";</pre>	java × web.bi .er { "/","/in ddel mod	<pre>Main.java x nd.annotation.GetMapping; ndex" , "/index.htm" , "/index.html"}) el){</pre>	1111111 1012 1013 1013 1013 1013 1013 10	項目已经启动, 访问: 前時: http://gateway.studio.data, aliyun.com/pcossgre08da76eaix c062biwigo/8080/ 方話: http://gateway.studio.data. aliyun.com/pcsossgre08da76eaix c062biwigo/7001	Runtime Share
输	出 调用堆栈 <b>断点</b>	▶ 포	<u>× × 4</u>	2 🖬							
+	<ul> <li> <ul> <li>Java Line Breakpoin</li> <li>IndexController.jav</li> <li>Java Exception Breakpoin</li> <li>Java Aethod Breakpoin</li> </ul> </li> </ul>	ts va:18 boints ints			IndexCont	roller.java18 请选择	3				
, ∰r D	EBOG PROBLEM							构建 100%			

### 函数断点

函数断点相比异常断点与行断点的不同点为:函数断点会触发两次事件,即entry/exit。您可以手动添加一个函数断点,也可在函数被定义的地方打断点,同样会产生一个函数断点。

	✓ controller					
	🔻 api.demo		priva	te String foo(){		
	≦ OssDemoController.java			Add Method BreakPoint		
	<ul> <li>⊇ DemoApiController.java</li> <li>▼ page</li> </ul>			* Class Pattern		
	🛓 IndexController.java			com.alibaba.demo.controller.page.IndexController		
*	▶ service ﴿ Main.java		}	* Method Name		
*	▼ resources			foo		
输	出调用堆栈 断点 🕪 포 浊 💆					
+			com.al		Cancel OK	
	Java Line Breakpoints		Enabled	3		
	Java Exception Breakpoints		Conditi	on 请选择		
$\sim$	Java Method Breakpoints					
	com.alibaba.demo.controller.page.IndexControlle	ər				

触发它后可以看到,进入该函数时会暂停,即将跳出程序时也会暂停。

G	App Studio	工程	文件	编辑	版本	查看	调试	设置	帮助				n	nain $\lor$	· 🕨 🕷	
ŋ	工程				🤞 Index	Controller.java		Result.java		<u></u> (	Main.java 🗙		项目已经 前端:	圣启动,访问: otto://cateway	studio data	₽
	demo5 ()	ibaba.demo mon troller pi.demo ossDemoC DemoApiCo age IndexContr ice	Controller, jav ontroller, java roller, java	ra a		<pre>* @date 2/ */ @Controlle public cla public cla public re } privat st if } re }</pre>	vile-08-14 r ss IndexC pping(val String i turn foo( e String ring[] ar (array.le return turn "err	ontroller ue = { "/' ndex(Mode ); foo(){ ray = new ngth == 3 "index" ; or" ;	{ l mode String	dex", l){ g[]{";	, "/index.htm" , "/index.html"}) a","v","d");		新聞: aliyun.gyot]8j 后端: ! aliyun.g gyot]8j	http://gateway. om/pcsoxspre- ad3iu/8080/ http://gateway. om/pcssoxsere adv-//2001	studio.data. 08da76ea5 studio.data, 08da76ea5	untime Share
*	Ma ▼ resource ▼ temple	in.java S Itae			29											
4010	口 明用相伐 四环	π														^
+ -	<ul> <li> <ul> <li></li></ul></li></ul>	iints eakpoints eakpoints lemo.controlli	er.page.Inde	exControll	er											
∰ D	EBUG 📃 PROB	LEM			1						bze 10.02/					
gatewa	iy.studio.data.aliyun.	com/pcsoxspr	reveda/6ea5	gyonglada	10/7001					1	内理 100%					

### 异常断点

如果配置了异常断点,当程序在遇到异常时,会在出现异常的地方进行断点。

	▼ main		* 前端入口			
	▼ java					
	▼ com.alibaba.demo					
	▶ common					
	▼ controller		@Controller			
	Controller		public class	IndexController {		
	▼ api.demo					
	🛓 OssDemoController.java		@GetMappi	ng(value = { "/","/index" , "/index.htm"	", "/index.html"})	
	🛓 DemoApiController.java		public St	ring index(Model model)		
	▼ page		En En	ter Exception Class		
	IndexController.java					
	service		<u>۹</u>	lease Input the full name of the exception		
	( Main java		ja	va.lang.NullPointException		
25						
*	✓ resources					
1001		_→ _			Cancel	
78J C	山 调用堆栈 断尿 🕪 💌 💆 💆	. L <b>II</b> -				
<b>(+)</b>			IndexContr	oller.java18		
	I Java Line Breaknointe		🔽 Enabled			
	IndexController.java:18		Condition	请选择		
	Java Exception Breakpoints					
	Java Method Breakpoints					
_						

触发index,由于出现了NullPointerException,所以断点在23行。

6	App Studio 工程 文件 编辑	版本 查看 调试 设置 帮助	main 🗸 🕨 🌺 🔳
0	T程 :≣ demo6 ① > santa > sarca > sarca > main - java > com.alibaba.demo > common - controller - api.demo - api.demo - api.demo - api.demo - api.demo - bomponController.java - page - page - service - service - service - service - service - service - service - resources	<pre>IndexController.java X Result_java X Main.java X IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII</pre>	現日已経過む。 た時に 開催: http://gatewaystudio.data. aliyun.com/csospre08da78ea5 gyo1j8jad33uj080/ 后端: http://gatewaystudio.data. aliyun.com/cscospre08da76ea5 gyo1j8jad3iu/7001
输出	出调用堆栈 断点 🗼 玊 놀 差	· · · · · · · · · · · · · · · · · · ·	×
+	<ul> <li> <ul> <li>ava Line Breakpoints</li> <li>Java Exception Breakpoints</li> <li>java.lang.NullPointerException</li> <li>Java Method Breakpoints</li> </ul> </li> </ul>		
∰ D	EBUG PROBLEM	构建 100%	

# 4.5.4. 断点及操作

断点面板为您展示当前设置的所有断点,本文将为您介绍断点的操作。

断点包括普通行断点、函数断点和异常断点三种断点类型,详情请参见断点类型。

输			断点	₽	Ŧ	¥	¥	2	(	•	
+		۲								IndexContro	oller.java18
$\sim$	<b>-</b> 1	Java Line B	reakpoints							Enabled	
	<b>V</b>	IndexCon	troller.java:18							Condition	请选择
	Jav	/a Exceptio	n Breakpoints	•							
	Jav	va Method	Breakpoints								
	DEBU	IG 🔳 F	ROBLEM								

调试操作

#### 调试界面如下所示:

←	→ C ① 不安全 pre	e-studio.data.a	liyun.com	/#/								🖈 📴 🎇 📀 💺 🗷 🗄
ග	App Studio 工程	主 文件	编辑	版本	查看	调试	设置	帮	助			main 🗸 🕨 🎘 📕
n	工程			🔹 Inde	xController.java		pom.xml				T	项目已经启动,访问:
יים	demo (j)				xml vers:</th <th>ion="1.0</th> <th>encoding</th> <th>g="UT</th> <th>'F-8"?&gt;</th> <th>The second s</th> <th>- 1</th> <th>前端: http://gateway.studio.data.</th>	ion="1.0	encoding	g="UT	'F-8"?>	The second s	- 1	前端: http://gateway.studio.data.
P	▼ src				<project th="" x<=""><th>nlns="<u>ht</u></th><th>tp://mave</th><th>h.apa</th><th>iche.org/POM/4.0.0"</th><th>A STATE OF A STATE OF A</th><th></th><th>dsgzuint6k0/8080/</th></project>	nlns=" <u>ht</u>	tp://mave	h.apa	iche.org/POM/4.0.0"	A STATE OF A		dsgzuint6k0/8080/
	✓ main				XI	mlns:xsi	=" <u>http://</u>	WW.W	/3.org/2001/XMLSchema-instance"	Sectore and	- 4	后端: http://gateway.studio.data.
	▼ iava				<model< th=""><th>Version&gt;</th><th>4.0.0<th>delVe</th><th>ersion&gt;</th><th>The support of the support</th><th></th><th>dsgzuint6k0/7001</th></th></model<>	Version>	4.0.0 <th>delVe</th> <th>ersion&gt;</th> <th>The support of the support</th> <th></th> <th>dsgzuint6k0/7001</th>	delVe	ersion>	The support of the support		dsgzuint6k0/7001
	✓ com.alibaba.de	emo			<packag< th=""><th>ging&gt;jar</th><th><th>ng&gt;</th><th></th><th>Terrer-</th><th></th><th>Sha</th></th></packag<>	ging>jar	<th>ng&gt;</th> <th></th> <th>Terrer-</th> <th></th> <th>Sha</th>	ng>		Terrer-		Sha
	▶ common				<groupid>co</groupid>	om.aliba	ba.demo </th <th></th> <th></th> <th></th> <th></th> <th>are</th>					are
					<pre><artifactio< pre=""></artifactio<></pre>	d>demo </th <th>artifactI</th> <th></th> <th></th> <th>-18-100-</th> <th></th> <th></th>	artifactI			-18-100-		
	▶ ani.demo					on>1.0.0	SNAPSHOT		'sion>			
					<denend< th=""><th>dencyMan</th><th>agement&gt;</th><th></th><th></th><th></th><th></th><th></th></denend<>	dencyMan	agement>					
	· page	ontroller inve			<de< th=""><th>ependenc</th><th>ies&gt;</th><th></th><th></th><th></th><th></th><th></th></de<>	ependenc	ies>					
		Sond Onen.java				<depen< th=""><th></th><th></th><th></th><th></th><th></th><th></th></depen<>						
	- Service						roupId>or	g.spr	ingframework.boot			
	• IIIpi						rtifactId	⊳spri	ing-boot-dependencies			
	OssServ	ce.java				<	ersion>1.	5.12.	RELEASE			
	A PaiApiSe	rvice.java				<	cone>impo	rt <th>scope&gt;</th> <th></th> <th></th> <th></th>	scope>			
	🛓 Main.java					<th>ndency&gt;</th> <th></th> <th></th> <th></th> <th></th> <th></th>	ndency>					
	resources					dependen						
4	target				<th>ndencyMa</th> <th>inagement&gt;</th> <th></th> <th></th> <th></th> <th></th> <th></th>	ndencyMa	inagement>					
-	pom.xml				<depend< th=""><th>dencies&gt;</th><th></th><th></th><th></th><th></th><th></th><th></th></depend<>	dencies>						
输出	出 调用堆栈 断点			<b>1</b>								
}" on	to public org.springfo ation.java.lang.String	amework.http.	Response	ntity <j< th=""><th>ava.lang.Ob</th><th>ject&gt; or</th><th>g.springfr</th><th>amewo</th><th>ork.data.rest.webmvc.RepositorySearchController.headForSearch(org.s</th><th>pringframework</th><th>.data</th><th>a.rest.webmvc.RootResourceI</th></j<>	ava.lang.Ob	ject> or	g.springfr	amewo	ork.data.rest.webmvc.RepositorySearchController.headForSearch(org.s	pringframework	.data	a.rest.webmvc.RootResourceI
2018-	08-15 14:04:19.574			0.8.0	l.r.w.BasePat	thAwareH	andlerMapp	ing	- Mapped "{[/profile], methods=[GET]}" onto org.springframework.h	ttp.HttpEntity	<org< th=""><th>.springframework.hateoas.Re</th></org<>	.springframework.hateoas.Re
sourc	eSupport> org.springf) 08-15 14:04:19.574	amework.data.	rest.webm	o.s.d	11eControlle	er.listA thAwareH	andlerMapp	etada	<pre>ata()     - Mapped "{[/profile],methods=[OPTIONS]}" onto public org.spring</pre>	framework.http	.Htt	pEntity org.springframew
ork.d	ata.rest.webmvc.Profil	eController.p	profileOpt	ions()								
mewor eInfo	<pre>08-15 14104119.575 k.http.HttpEntity<org. pre="" rmation)<=""></org.></pre>	springframewo	ork.data.r	o.s.d est.web	mvc.RootRes	ourceInf	ormation>	org.s	<ul> <li>Mapped ~{[/profile/{repository}],methods=[GET],produces=[appli springframework.data.rest.webmvc.alps.AlpsController.descriptor(org</li> </ul>	springframewo	ork.da	*/*]}" onto org.springfra ata.rest.webmvc.RootResourc
2018-	08-15 14:04:19.575	enringfrom	main] INFO	0.5.0	.r.w.BasePat	thAwareH	andlerMapp	ing	<pre>- Mapped "{[/profile/{repository}],methods=[OPTIONS],produces=[ang()]</pre>	pplication/alp	s+jso	on]}" onto org.springframew
2018-	08-15 14:04:19.576	springrramewo	main] INFO	0.5.0	l.r.w.BasePat	thAwareH	andlerMapp	ing	- Mapped "{[/profile/{repository}],methods=[GET],produces=[appli	cation/schema+	+json	<pre>}" onto public org.springf</pre>
ramew forma	ork.http.HttpEntity <on tion)</on 	g.springframe	work.data	.rest.w	webmvc.json.	JsonSche	ma> org.sp	ring	framework.data.rest.webmvc.RepositorySchemaController.schema(org.sp	ringframework.	data	.rest.webmvc.RootResourceIn
2018-		n	main] INFO	0.8.j	.e.a.Annotat	tionMBea	nExporter		- Registering beans for JMX exposure on startup			
2018- 0			main] INFO	com.a	libaba.demo.	.Main	ServietCon	caine	- runeat started on port(s)? 7001 (http) - Started Main in 4.51 seconds (JVM running for 5.873)			

上图中从左到右的每个按钮所代表的含义如下:

功能	说明
continue	恢复当前断点时,当前线程继续运行。
step over	执行到下一行。
step in	进入函数。
force step in	强制进入函数,与step in的区别在于,它可以引导断点执行到Java自带的类库 中。
step out	从当前函数跳出。
restart	目前的restart实现方式可能无法完成程序清理等工作,正在优化中。
stop	停止调试。
Drop Frame	删除当前栈,回退到上一个函数。
Run to Cusor	执行到当前行,可以在某一行打一个临时断点。
计算表达式	可以任意执行一个表达式进行计算。

操作的快捷键如下所示:

快捷键参考	
快捷键	功能
νD	启动调试
<del>ሄ</del> F2	停止
Ctrl R	重启调试
[™] R	运行
F9	继续
F8	单步跳过
F7	单步执行
ΥF7	强制进入
<b>① F8</b>	单步跳出

• 对变量进行赋值

您可以直接在断点上对一个变量进行赋值。



双击某个字段,然后构造一个表达式对当前值进行赋值,按enter键生效。



● 表达式计算

打开计算表达式面板, 输入可执行表达式。

表达式计算		×
表达式:		
array		
结果:		
v array = java.lang.String[1] (id=23)		
∨ <b>♂</b> 0 = a		
₿hash = 0		
>		
	取消	计算

● 变量监视

您可以右键单击变量,选择添加监视。

7	~	Ľ	•	×	M	
	Varial	oles				
	ø ar	gs = {				) java.lang.String[0] (id=6)
	)≣ a	rray =				()) java. () () () () () () () () () () () () ()
						编码
						有制
						<b>经</b> ,即9

添加后,即可在右边面板看到相应的变量监视。



您也可以在watch上手动新增变量。

+ - 6
✓ & array = java.lang.String[3] (id=7)
> 🕜 0 = a
> 🕜 1 = b
> 🕜 2 = c
63 this = Cannot evaluate expression 'this'
> 🕜 0 = a
> 🕜 1 = b
> 🕜 2 = c
6a args = java.lang.String[0] (id=6)
6ð arra
array - String
array. Surrigu
Array - java.lang

● 线程操作

您可以在调试面板查看线程操作。



根据当前线程的运行进度,下拉框中会显示RUNNING或WAIT等不同的信息。当您选中另外的线程时,变量的面板信息也会随之改变。

# 4.5.5. 远程调试

由于调试机器在日常环境,因此只能调试日常环境上部署的应用。

1. 配置调试信息。

Run/Debug Configurations					×					
+ ×	Name: Unnamed									
✓ ₽ Remote										
Unnamed	• Host:	30.5.36.144								
> 📮 Application	* Port:	8000								
	Command line arguments for running remote JVM:									
	-Xdebug -Xrunjdwp:transport=dt_sockst_server=y_suspend=n,address=8000									
	机器:	2vCPU, 4G内存								
		(	Cancel	Apply	ОК					

⑦ 说明 您需要填写Host和Port信息,告知JVMTI需要连接哪个远端服务。

2. 单击调试,遇到Debugger信息时,代表连接成功,即可开始调试。

环境准备完成 [Warn] The debugger and the debuggee are running in different versions of JVMs. You could see wrong source mapping results. Debugger JVM version: 1.8.0_181 Debuggee JVM version: 1.8.0_101]

远程调试使用JVMTI进行Socket连接,本质上Debugger和Debuggee之间仅传输JVM运行信息,不会传输标准输出和错误输出。

4.5.6. 终端

Terminal按钮显示在页面底部。

App Studio支持常规的ls、cat 等Shell命令和vi、top等带有交互的命令。

您可以开启多个终端。



# 4.5.7. 热部署

热部署是指代码运行过程中,您手动修改的代码可以在不重启服务的情况下生效。

例如SpringBoot在运行/调试过程中,修改完代码后无需重启,保存即可生效,App Studio已经默认包含此功能。

除调试模式外,运行模式下也支持这项功能。触发热部署无需安装插件和手动编译文件,您只需保存文件即可。

DataW	App Studio						🍰 +文			
工程	文件 编辑 版	本 查看	构建 调试	设置	模板 帮助 反馈	4	👫 Unnamed 🗸 🕨 🎇 🗖			
	工程 idw-aone-demo ① ▼ src ▼ main ▼ java		=	applicatio 30 37 38 39 40 41	n.properties X & MyMain.java X & Main.java X & I @Autowired MyService myService ; @GetMapping(value = {"/testHotCode"}) @ResponseBody		项目已经启动,访问: 后端: https://pcsproductffde70b7hlavw m8rpvkqr-80.r-alicode.alibaba-in c.com/			
	✓ com.alibaba.s ✓ idw.ide ④ Bo.java ④ IndexCo ▲ Main iav	earch ontroller.java			<pre>globalcistory public 350N0bject testHotCode(Model model){     JS0N0bject json = new JS0N0bject();     json.put("webide","appstudio");     return json; }</pre>		Share			
*	ي المسابق المسابق المسابق المسابق المسابق المسابق المسابق	va 1.java 1111.iava			<pre>@GetMapping(value = {"/","/index", "/index.htm", "/in public String index(Model model) {    System.out.println("abc:" + abc);    System.out.println("abc:" + abc); </pre>		Cere			
Termir	al						×			
+	<pre>- Local <script src="//g.alicdn.com/??alilog/mlog/aplus v2.js,retcode/log/log.js"></script> <script src="//g.alicdn.com/alife/lib/0.0.11/??react/16.0.0/react.production.min.js,react-dom/16.0.0/react-dom.production.min.js"></script></pre>									
< < < < < < < < < < < < < < < < < < <	<pre>cscript src="//gialcdn.com/alte/lib/U.0.11/?#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/#moment/2.20.1/##moment/2.20.1/##moment/2.20.1/##moment/2.20.1/##moment/2.20.1/##moment/2.20.1/##moment/2.20.1/##moment/2.20.1/##moment/2.20.1/####################################</pre>									
( \$ @ OI	admin∉webide _] JT ॐ DEBUG ■	PROBLEM	I Terminal	P Version C	ontrol					

如果您正在Debug中进行代码变动,会自动删除当前运行栈,回退到函数入口。

Dati	App Studio		🚔 ^{中文}
工和	星 文件 编辑 版本 查看 构建 调 ¹	设置 模板 帮助 反馈	🕻 👫 Unnamed 🗸 🕨 🎉 🔳
D	I程 :=	🛓 application.properties 🗙 👙 MyMain.java 🗶 🛓 Main.java 🗶 🛓 IndexControl	项目已经启动,访问: 后端:
8	v src ▼ main	48 return json; 49 }	https://pcsproductffde70b7hlavw m8rpvkqr-80.r-alicode.alibaba-in c.com/
Ð	<ul> <li>✓ java</li> <li>✓ com.alibaba.search</li> </ul>	51     public String fool(){     Press       52     String foo = "bar";     Press       53     return fool     Press	<u>م</u>
	▼ idw.ide	54 }	hare
	<ul> <li>IndexController.java</li> <li>Main.java</li> <li>Mmt.java</li> <li>MyMain.java</li> <li>MyMain.java</li> <li>MyMainTI.java</li> <li>MyGnietor invo</li> </ul>	56     @GetMapping(value = {"/", "/index", "/index.htm", "/in "minor public String index(Model model){       57     public String index(Model model){       58     System.out.println("abc:" + abc);       59     System.out.println("abc:" + abcc);       60     myService.foo();       61     Thread thread = new Thread(() -> {       62     try {	L Data
*	4 Dala lava	63         System.out.println("start");	Dat
Term	inal		×
+	Local		
-	("webide": "webide-666") [admin@webide -] Scurl localhost:7008/testHotCode ("webide": "webide-666") [admin@webide -] Scurl localhost:7008/testHotCode ("webide ': webide-666"}	I	
<b>a</b>	[adminewebide _] \$	۲ Version Control	

### 运行模式下的热部署配置

1. 在配置面板上主动开启热部署。

#### APP Studio·功能介绍

#### Dat aWorks

Run/Debug Configurations				×
+ ×	Name: Unnamed			
> ≈ Remote ✓ 📮 Application	* Main class: 🚺	com.alibaba.dataworks.Main		
Unnamed	VM options:			
	Program arguments:			
	Environment Variables: JRE:	1.8 - SDK		
	PORT:	7001		
	机器:	2vCPU, 4G内存		
	开启HOTCODE:	● 是 ○ 否		
			Cancel Apply	ОК

启动后,即可在输出中看到HotCode2的输出信息。

HotCode2
Hello, HotCode2 (Ver: 2.0.1.20171017034407) !!!
Start JVM with HotCode2 on Java_1.8.0_181-b13 @ JBoss-Linux-3.10.0-327.ali2012.alios7.x86_64
HotCode2 Path: /home/admin/plugins/hotcode2.jar
Web Container: JBoss
Monitered Resource Paths :
42e4da33-0cfd-4f14-947e-d6b6441cd04b-1.0.0-SNAPSHOT.jar> /42e4da33-0cfd-4f14-947e-d6b6441cd04b/target/classes
Enabled Plugins: [sofamvc3_plugin, ibatis_plugin, mybatis_plugin, classmate_plugin, spring_plugin, sofarest_plugin, sofa3_plugin, webx3_plugin, s
HetCode2

2. 触发热部署。



当您对文件进行修改时,需要手动触发文件保存。

2013-01-10 20101143.137 [ Math] INFO Com.allbaba.dataworks.main	- Started Main in 9.201 seconds (JVM running for 10.407)
HotCode2	
Hello, HotCode2 (Ver: 2.0.1.20171017034407) !!!	
Start JVM with HotCode2 on Java 1.8.0 181-b13 @ JBoss-Linux-3.10.0-327.ali2012.alios7.x86	64
HotCode2 Path: /home/admin/plugins/hotcode2.jar	
Web Container: JBoss	
Monitered Resource Paths :	
42e4da33-0cfd-4f14-947e-d6b6441cd04b-1.0.0-SNAPSHOT.jar> /42e4da33-0cfd-4f14-947	e-d6b6441cd04b/target/classes
Enabled Plugins: [sofamvc3 plugin, ibatis plugin, mybatis plugin, classmate plugin, spring	plugin, sofarest plugin, sofa3 plugin, webx3 plugin, sofa
HotCode2	
2019-01-16 20:01:33.135 [-nio-7001-exec-1] INFO o.a.c.c.C.[Tomcat].[localhost].[/]	- Initializing Spring FrameworkServlet 'dispatcherServle
2019-01-16 20:01:33.135 [-nio-7001-exec-1] INFO o.s.web.servlet.DispatcherServlet	- FrameworkServlet 'dispatcherServlet': initialization ;
2019-01-16 20:01:33.151 [-nio-7001-exec-1] INFO o.s.web.servlet.DispatcherServlet	- FrameworkServlet 'dispatcherServlet': initialization (
Success to reload class class com.alibaba.dataworks.controller.page.IndexController	

3. 当代码增量同步完成后,控制台显示Reload某个类的输出,则代表热部署生效。代码示例如下:

```
public class IndexController {
    @RequestMapping("/")
    @ResponseBody
    public String index(){
        return "cccc";
    }
}
```

您可以将Return字符串内容改为其它字符串,让其立即生效。

### Debug模式下的热部署

您可以通过JDI原生方法实现Debug模式下的热部署,但由于JVM的限制,在给某个类增加或删除方法时,无 法进行热部署。您同样只需保存文件即可触发热部署。

⑦ 说明 JVM原生不支持对类结构进行变动后的热部署,新增或删除类等其他操作都可以支持热部署。

• 新增方法或删除方法。

Datal	App Studi	io									中文
工程	文件 编辑	版本 查看	构建 调词	式 设置 模	板 帮助	反馈		4	Unnamed		× 🔳
Ď 22 ₽	工程 idw-aone-demo ① ▼ src ▼ main ▼ java ▼ com.aliti ▼ idw.ic ♀ Br	D baba.search ie o,java	=	<ul> <li>▲ application.pt</li> <li>40</li> <li>41</li> <li>42</li> <li>43</li> <li>44</li> <li>45</li> <li>46</li> <li>47</li> <li>48</li> <li>46</li> </ul>	operties × GetMapping( ResponseBod Jblic JSONO JSONObje json.put return j I GetMapping(	<pre>MyMain.java x value = {"/testHotCo y bject testHotCode(M ct json = new JSONO ("webide", "webide-6" son; value = {"/", "/inde:</pre>	<pre>Main.java × bde"}) bdel model){ bject(); 56"); </pre>	IndexControll	项目已经启动,访问 后端: https://pcsproduct m8rpvkqr-80.r-alice c.com/	fde70b7hla de.alibaba	Runtime Share
*	을 In 클 M 클 M 글 M	IdexController.java Iain.java Imt.java IyMain.java IvMain111.iava			blic Strin System.o System.o myServic Thread t try	<pre>g index(Model model ut.println("abc:" + ut.println("abcc:" - e.foo(); hread = new Thread( {</pre>	<pre>{     abc);     abcc);     () -&gt; { </pre>	a transmission and a second se			Data
Termi	nal										X
+ 0	Local script src="//al script yrc="test script yrc="test script yrc="test script yrc="test script localhost:" ("webide":"webide" script localhost:" ("webide":"webide" script localhost:" ("webide":"webide" script localhost:" ("webide":"webide" script localhost:" script lo	linw.alicdn.com/ st/javascript" s 7008/testHotCode udio"} 7008/testHotCode =-666"} PROBLEM	/onebox/statio src="http://g e e III Terminal	c/bear/l.l.0/l .alicdn.daily. P Version Cont	oader-min.j taobao.net/ rol	js"> cdn-versions/alishu-	app-studio/cn-shang	hai-daily/pages	/aone/index.js">		

• 新增字段。



4.6. 协同编程

本文将从实时协同编辑、邀请协作者、加入写作项目、协作者面板和权限等方面为您介绍协同编辑。

App Studio支持实时协同编辑功能,团队中多个成员可以同时在同一个项目中开发、编写代码,并实时查看 其它成员的改动。能够避免同步代码、合并分支的繁琐,显著提升开发效率。

### 邀请协作者

项目的所有者可以邀请其他开发者加入项目进行协作。

- 1. 打开要分享的项目。
- 2. 单击右侧的Share展开协作者面板。
- 3. 单击右上角的邀请,进入邀请流程。

设	置	发布	模板	帮助	反馈					۵	<   1	Edit Config 🔨	< 🕨 💥	
🛃 Ma	in.java	×								× Share		< 分享链	接 88邀请	
1		ge com.	alibaba.	datawori										R
2					et sutecepfiques Cesh	letuteConfiguration.				协作者		权限		1 tim
2	import	org.s	pringfra	mework I	ot SpringApplication:	TEAULOCONTIGUIACION;								•
5	import	org s	nringfra	mework i	ot autoconfigure Sori	ngRootApplication.						is viewina 所有	皆	
6	import	org.s	pringfra	mework.	ntext.annotation.Com	onentScan:								
7														
8														SP2
9														6
10														
11	@Sprin	ngBootA	pplicati											
12	@Enabl	leAutoC	onfigura	tion										
13	@Compo	onentSc	an(baseF	ackages		<b>```</b> )								
14			Main											at
15		JDIIC S	ratic vo	id main	(Main class and )									
17		opra	пвчррттс	ac100.00	(nain.class , args) ;									
18														
19														
														S I

4. 填写邀请协作者对话框中的各配置项。

邀请协作者	×
项目协作仅限在相关的主账号和子账号之间进行,新建子账 号请前往RAM控制台 *用户名:	
请输入阿里云账号搜索 ∨	
* 权限 : ① 只读 () 读写	
确认	取消

配置	说明
用户名	填写邀请的写作者的用户名。
权限	根据自身需求选择 <b>只读</b> 或 <b>读写</b> 权限。

5. 单击确认,即可成功邀请。

#### 加入协作项目

当您被邀请加入其他开发者的项目后,可以在打开的工程面板下,选择**我参与的**,查看您加入的协作项目。 单击即可加入项目,开始实时协同编辑。

选择已有工程						
我创建的	我参与的					
est						
est						

### 协作者面板

实时协同编辑时,协作者们可以互相查看当前的状态。



- 1. 单击页面右侧的Share展开协作者面板。
- 2. 查看相应协作者的在线状态、正在编辑的文件和拥有的权限。

⑦ 说明 项目所有者可以移除协作者。

### 权限说明

在协同编辑的过程中,参与的协作者权限分为以下三种:

- 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项目中的所有文件, 也可以对这些文件进行编辑。
- 只读权限: 拥有只读权限的协作者只能查看项目中的文件, 但是无法进行编辑。

# 4.7. 应用部署

本文将为您介绍如何在App Studio上新建一个应用并部署到生产环境,获得一个可以通过公网访问的应用。

#### 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DataWorks图标,选择**全部产品 > App Studio**。

🜀 💥 DataStudio	••			
Ξ 全部产品 • >	数据汇聚		应用开发	
Os 数据集成	Co 数据集成		🛕 App Studio	
X DataStudio(数据开发)				
🌺 运维中心(工作流)	数据开发		机器学习	
<table-cell-columns> 任务发布</table-cell-columns>	💸 DataStudio(数据开发)	Ø	巴 机器学习PAI	Ø
② 数据质量	A 数据服务	Ø		
❷ 数据地图(数据管理)	<b>F</b> Function Studio New	Ø		
🏂 数据保护伞	<b>畫</b> 数据分析			
巴 机器学习PAI				
🐥 数据服务	任务运维			
🗲 Function Studio	🌸 运维中心(工作流)			
	🦪 任务发布			
	跨项目克隆			
	数据治理			
	② 数据质量	Ø		
	≫ 数据保护伞			
	参数据地图(数据管理)	Ø		
日 返回控制台	资源优化			

3. 进入App Studio页面后,您可以通过模板、代码和Git导入三种方式创建工程。

\$	▲ App Studio		
Ш			
•			
Q	欢迎来到 App Studio		
ŵ			
	¢;	D	
	通过模板创建工程	通过代码创建工程	通过Git导入工程

根据自身需求选择相应的新建方式,并填写配置后,单击提交,即可新建工程。详细操作请参见工程管理。

### 关联Git

发布应用前,需要初始化Git。

1. 在Code页面新建一个repo,并保存仓库的SSH地址。

← → C = ###	https:	://code.aliyun.com/wumou.wm/dataos_fancy] C 🛧 — 📑 💆 🖉 🛸 e 🗇 🕲 👘 : 2 🖞 content 1992 — Induction the Content of State () induction the						
	管理控制台	产品与服务 ▼ alicode_test						
		为了更好的为您提供服务, 云效 邀请您使用持续交付相关功能。云效结合ECS、EDAS等服务为您提供完备的发布、部署、测试全研发流程,大大提升您的研发效率. <u>了解更多&gt;</u>						
合目以								
📮 项目								
월 动态		rioject uniato_janty wassullessiumy leaneu.						
合并请:	求	1 mm						
<ul> <li></li></ul>		M PRIVALE						
① 缺陷		D						
🗐 Wikis								
— ●● 成员		dataos_fancy						
>> 标金		★ 0 SSH HTTPS git@code.aliyun.com:wumou.xm/da 🚯 +						
<b>¢</b> ° 设置								
		此在口外近十八年月前的						
		[[[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]]] [[]] [[]]] [[]] [[]]] [[]]] [[]] [[]]] [[]]] [[]] [[]]] [[]] [[]]] [[]] [[]] [[]]] [[]] [[]]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[][] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[]] [[						
		如有已有又件请按照下面的命令行指令推送它们。 武学可以出 <b>维如</b> DE ADME 文化开始本语 P						
		35日可607A <u>36加ITCFUUTE</u> 入口力因争项目。						
		命令行指令						
		Git 全局设置						
		git config —global user.name "alicode_test" git config —global user.email "wumou.wm@alibaba-inc.com"						

2. 进入App Studio页面, 打开新建的工程, 单击版本, 选择初始化&关联远程仓库。

6	▲ App Studio					
ធ	工程文件编辑	版本	查看	调试	设置	发布
n)	工程					
	DataOS_App 🕧					
P	> .alicode					
	> APP-META					
	> santa					
	> src					
	🛓 .classpath					
	≣ .factorypath					
	<ul> <li>.gitignore</li> </ul>			_		
	pom.xml	初始化	3关联远程	仓库		
		Merge	Abort			

3. 填写关联远程仓库对话框中的配置,单击提交。

关联远程仓库	×
● 此操作包含初始化(init, add, commit, remote add),操作完成后页面将会刷新 Git 地址:	
请输入要关联的git仓库地址,格式:****.git	
远程仓库名 origin	
提交信息	
初始化工程	
· · · · · · · · · · · · · · · · · · ·	

⑦ 说明 如果您未绑定SSH Key或Git 用户名邮箱,可以根据页面引导进行操作。

### 发版

关联Git完成后,即可通过**发版**创建应用。

1. 返回工作空间页面,单击相应工程下的管理。
|                                          | 0                  |          |            |
|------------------------------------------|--------------------|----------|------------|
| 三 「「「「「」」「「「」」「「」」「「」」「「」」「「」」「「」」「「」」「「 | 亚训车到 App Studio    |          |            |
| Q 应用空间                                   | 永远未主 App Studio    |          |            |
| 模板空间                                     |                    |          |            |
|                                          | ¢.                 | ō        | <u>(7)</u> |
|                                          | 通过模板创建工程           | 通过代码创建工程 | 通过Git导入工程  |
|                                          |                    |          |            |
|                                          | 我的工程               |          |            |
|                                          | Q. 清编入 <b>投</b> 決決 |          |            |
|                                          | DataOS_App         |          |            |
|                                          |                    |          |            |
|                                          | 1 分钟前更新            |          |            |
|                                          | ◎ 管理员 创建模板 管理      |          |            |

2. 单击右上角的发版,填写工程发版对话框中的配置。

							发版
					/	/	
				/			
工程发版			×				
* 应用名称:	不超过50位数字、字母、下划线组成的字符						
Region:	华东2(上海)						
部署参数配置:	Key	Value	操作				
描述	请填写本次发版的相关描述,便于跟踪信息						
			发版取消				

3. 配置完成后,单击**发版**。

# 部署应用

- 1. 单击左侧菜单栏中的应用空间,进入应用空间页面。
- 2. 单击已发版应用下的发布。

9	App Studi	0	
5	三		
	工1F空间 应用空间		
<u>ت</u> ټ	模板空间	Q 请输入 搜索	
		● 未部署 部署控制台 发布	

3. 单击应用部署提示框中的购买链接,根据指引在相应的区域购买AppStudio运行空间独享资源。

应用	部署								×
	检测到您当前应用没有分组或分组内均没有机器资源。请购买独立资源组并前往部署控制台新建分组并进行应用扩容 接入幂助 购买链接 部署控制台								
Da	taWorks独国	享资源(包年包	1月)						
(i) ii	青点击DataWorks独	享资源(包年包月)价格	計算器进行成本核算。	计费方式太复杂、仍然	《不知道如何购买?请您	四日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日	方案。		
	地域	华东1(杭州)	华北2(北京)	华东2(上海)	华南1(深圳)			<b>当前配置</b> 地域:	华东2(上海)
	独享资源类型	独享调度资源	独享数据集成资源	AppStudio运行空间 (生产环境)				独享资源类型: AppStudio运行空间	AppStudio运行空间 (生产环境) 〕 2 vCPU 4 GiB
alari		DataWorks独享资源使	用场暴及计费标准请您	参考:DataWorks独享	资源			(王)小鬼): App Studio网段选择	¥: 192.168.0.0/16
至本配置	AppStudio运行 空间 ( 生产环 境 )	2 vCPU 4 GiB	4 vCPU 8 GiB	8 vCPU 16 GiB				资源数 <del>星:</del> 计费周期: 配置费用:	1 1个月
	App Studio网段 选择	192.168.0.0/16	172.16.0.0/12	10.0.0/8				* 0.00	
		该网段用于在非用户侧 务数据源所在网段不同 署运行空间,如您仍选	部署App Studio生产玩 的网段。 如:您的RDS 择172.16.0.0/12网段,	境运行空间计算资源, 5 for Mysql部署于172.1 则App Studio生产环境	为保证运行空间与您的 .6.0.0/12网段,则您应 运行空间无法与该网的	业务数据源网络互通,然 选择192.168.0.0/16、1 段的云产品实例网络互通	3务必需选择与您的业 10.0.0.0/8网段用于部 •	立即购买	
	资源数量	1							
国家部	计费周期	1个月	2个月	3个月	4个月	5个月	6个月		
		7个月	8个月	9个月	1年	2年	3年		

4. 购买成功后,单击部署控制台,进入运维页面。

⑦ 说明 此时需要解绑之前绑定的Host。

- 5. 单击分组列表下的创建分组,完成分组的创建。
- 6. 选择操作 > 应用扩容,将刚刚购买的机器加入创建的分组中。

操作	E∨ 概览 监	空 镜像 变更 资源			
应用信息 Date		-₩ ^{应用状态} • 异常	分组信息 ● 总共 0 ■ 正常 0		机器信息 • 总共 0 • 正常 0
描述: DataOS_Xiuc 🗧		QPS(req/s): 0 RT(ms): 0	● 异常 0		
分组列表					+ 创建分组
分组名	\$ 7 \$	例規格	\$ ☑ 网段	\$ ♂ 描述	
			No Data		
机器列表					
分组名	\$ ♡ 实例ID	\$ ♀ 主机名	\$ ∀ IP地址	◆ ▽ 实例规格	\$ ♀ 状态
			No Data		

7. 完成后刷新应用空间,单击部署,将应用发布到默认的分组即可。

应用部署				×
ProdGroup	▶ 勝挪机器 应用管理			接入帮助
	2	3	4	5
选择分组	2	3 构建镜像	4 部署应用	5 发布完成

出现下图中的状态,代表发布完成。此时应用已经部署到您的ECS,并启动服务。

应用部署				×
ProdGroup 上次部署由	✓ 部署 应用管理 ● 发起于 2019/4/30 上午11:34:30 发	\$布到 ProdGroup 分组		接入帮助
· · · · · · · · · · · · · · · · · · ·				
选择分组	<b>构建应用</b> 构建成功(38秒) 查看日志	<b>构建镜像</b> 构建成功(186秒) 查看日志	<b>部署应用</b> 发布成功(123秒) <u>查看发布</u> 单	发布完成

# VPC下沉

VPC下沉是指将VPC加入到用户购买机器的网段。该操作需要在阿里云和App Studio应用运维平台实现,且 每个项目仅需执行一次,之后的版本迭代只需执行上面的部署应用即可。

### VPC接入授权

App Studio用于发布的ECS通过弹性网卡和用户VPC连通,需要用户给App Studio的服务账号添加网卡权限,提交给运维平台。

1. 进入角色管理页面,单击新建RAM角色。

RAM访问控制	RAM访问控制 / RAM角色管理				
概览	RAM角色管理				
人员管理 /	、 什么是RAM角色?				
用户组	RAM角色机制是向您信任的实体(例如:RAM用户、某个应用或阿里云服务)进行授权的一种安全				
用户	- 恋云城市下的一下RAM用户(可能是代表一下移动App的运病服务) - 其他云账户中的RAM用户(需要进行跨账户的资源访问)				
设置	- ECS买例上运行的应用程序代码(需要对云资源执行操作) - 某些阿里云服务(需要对您账户中的资源进行操作才能提供服务)				
SSO 管理	- 企业的身份提供商IdP,可以用于角色SSO RAM角色颁发短时有效的访问令牌(STS令牌),使其成为一种更安全的授予访问权限的方法。				
权限管理 /	特别说明: 、 RAM角色不同于传统的教科书式角色(其含义是指一组权限集)。如果您需要使用教科书式角色的				
授权					
权限策略管理	新建RAM角色  输入角色名称或备注    Q				
RAM角色管理	RAM角色名称 备注				

2. 在新建RAM角色对话框中,选择类型为阿里云账号,单击下一步。

新建RAM角色
1 选择类型     2 配置角色     3 创建完成
<ul> <li>当前可信实体类型</li> <li>阿里云账号</li> <li>受信云账号下的子用户可以通过扮演该RAM角色来访问您的云资源,受信云账号可以是当前云账号,也可以是其他云账号</li> <li>阿里云服务</li> <li>受信云服务可以通过扮演RAM角色来访问您的云资源</li> <li>身份提供商</li> </ul>
身份提供商功能,通过设置SSO可以实现从企业本地账号系统登录阿里云控制台,帮您解决企业的统一用户登录认证要求
下—步 关闭

3. 填写角色名称,并选择云账号为其他云账号(此处固定选择为1591568227964362)。

新建RAM角色
送择类型     2     配置角色     3     创建完成
选择可信实体类型 阿里云账号
* 角色名称
不超过64个字符,允许英文字母、数字,或"-"
备注
最大长度1024字字符
* 选择云账号
<ul> <li>○ 当前云账号</li> <li>● 其他云账号</li> <li>1591568227964362</li> <li>可以访问 账户管理-&gt;安全设置 获取帐号ID</li> </ul>
上一步 完成 关闭

4. 单击完成,跳转至RAM角色管理页面。

5. 单击新建RAM角色后的添加权限,为其添加管理ECS弹性网卡的权限。

新建R	AM角色 输入角色名称或备注 Q			
	RAM角色名称	备注	创建时间	操作
	A		2019年1月29日 19:21:01	添加权限 精确授权 删除
	A	and the second sec	2018年11月19日 22:21:20	添加权限 精确授权 删除

6. 选择完成后,单击确定。

RAM访问控制 / RAM角色管理	添加权限	×
RAM角色管理	被授权主任	
什么是RAM角色?	The second	
RAM角色机制是向您信任的实体(eg, RAM用户、某个 - 您云账户下的一个RAM用户(可能是代表一个移动Ar	选择权限	
- 其他云账户中的RAM用户(需要进行跨账户的资源访 - ECS实例上运行的应用程序代码(需要对云资源执行	系统权限策略 💛 netw	◎ Q 已选择 (1) 清勝
- 某些阿里云服务(需要对您账户中的资源进行操作才) - 企业的身份提供商IdP,可以用于角色联合登录	权限策略名称 备注	AliyunECSNetworkInterface ×
RAM角色颁发短时有效的访问令牌(STS令牌),使其成	AliyunECSNetworkInterface 管理ECS弹性网卡的权限	
和1779 JURN · RAM角色不同于传统的教科书式角色(其含义是指一组	8	
5400AMAAA (6) AAA 7300231		
制 在 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1		
RAM用巴石体		
ALICODE-ROLE		
ROLE-WEBIDE		
二 流加权限		
	确定 取消	

### 7. 进入相应的RAM角色, 查看ARN。

RAM访问控制 / RAM角色管理 / ALICODE-ROLE		
← ALICODE-ROLE		
基本信息		
RAM角色名称	创建时间	2019年4月30日 13:46:03
备注	ARN	
切阳险田 信任学家管田		
修改住在第28		
PP IV. In 12 WHI		
1		
2 "Statement": [		
3 {		
4 "Action":		
5 "Effect":		
8 "acs:ram:		
9		
10 }		
11 }		
12 ],		
13 "Version": "1"		
14 }		

# 创建专有网络和交换机

创建专有网络和交换机需要在App Studio相同的区域进行,此处以上海区域为例。

登录VPC控制台创建专有网络,具体操作请参见搭建IPv4专有网络。

⑦ 说明 专有网络的IPv4网段需要选择与部署应用前选择的网段不同的网段。

创建完成后,在交换机页面记录下交换机的ID进行备用。

专有网络	交换机									② 如何创建	交换机
专有网络路由表	创建交换机  刷新	自定义						实例名称 🗸	请输入ID进行	精确查询	Q
交换机	实例ID/名称	所属专有网络 状态	IPv4网段	可用IP数	默认交换机	可用区 77	路由表	路由表类型	资源组	操作	
共享带宽 共享流量包	VSW-VPC-XIUDE	●可用	10.0.0/24	251	否	上海 可用区B	last	系统	默认资源组	管理 删除 购买、	AP
= 2前村-公司10											

# 创建安全组

登录ECS控制台创建安全组,详细操作请参见创建安全组。

安全组创建完成后,请记录安全组的ID进行备用。

(-)	管理控制台 🛛 🤷 华东2	(上海) 🔹	按索	Q	消息 ⁹⁹¹ 费用	工单 备案 1	全业 支持与服务 🚬	🐂 简体中文 🌘
	云服务器 ECS	安全组列表					⑦ 安全組限制与规	则 🗘 创建安全组
=	821X11 JF-34:67							
	▼ 存储	专有网络ID ♥ vpc-uf6j1depwtyiumwzu59w5 股素	●标签					-
æ	云盘	□ 安全组ID/名称 标签 所属专有网络	相关实例	可加入IP数 网	络类型(全部) 👻	创建时间	描述	操作
	文件存储 NAS							
0	<ul> <li>快照相缆螺</li> <li>体原则率</li> </ul>		0 1	1999 专	有网络	2019年4月30日 13:50	-	修改 克隆 还原规则
×	快照链	SG-XIUDE					官祖夫的	配量规则 管理评证的下
~	自动快照策略							
⊕	快照容量						共有1条,每页显示: 10条	<u></u>
	镜像							
-	▼ 网络和安全							
•	弹性网卡							
	安全组							
	密钥对							
	部署集							
	③ 专有网络 VPC							
	③ 弹性公网 IP							
	云助手							
	问题诊断							
	标签管理							
	任务管理							

# 在运维平台添加用户VPC

- 1. 登录DataWorks控制台,进入App Studio页面。
- 2. 单击页面右上角的运维。

6	App Studio	)			₽ 开发	₽ 运维	-
Ð	三						
۹	应用空间	欢迎来到 App Studio					
Ŷ	模板空间						
			۵			1111	
		新建模板工程	新建代码工程	导入Git工程			

3. 进入资源 > VPC页面,单击新增VPC。

App Studio							开发	运维	***	中文
springboot 🗸 🖂	springboot	概览	监控	镜像	变更	1				
よ 应用详情	操作手册		2							
	VPC列表								3 新增VF	PC

4. 在**新增vpc**对话框中,填写之前保存的**角色标识**(即ARN)、**安全组ID**和**交换机ID**,并进行相应的描述。

新增vpc		Х
*角色标识:	请输入角色标识	
* 安全组ID:	请输入安全组ID	
* 交换机ID:	请输入交换机ID	
描述:	请输入描述	
	执行	

5. 配置完成后,单击执行。

# 创建弹性网卡并绑定ECS

1. 单击相应VPC的ID,进入ENI管理页面。

keshihua1 🗸 🔄	-		概览	监控	镜像	变更	资源								
ム 应用详情	操作引	≞ <del>nn</del>													
	VPC列	表											新	₿VPC	
		角色标	识					安全组ID	<b>\$</b> 7	交换机ID	÷	操作			8

2. 单击新增ENI。

操作手册 VPC				
←返回 VpclD:3		中全省内		
交换机ID:		安主组D: 描述:		
ENI列表				新增ENI
EnilD	✿ ੲ EcsID	\$ ♀ 描述	\$ ☑ 操作	
		No Data		

3. 新增完成后,单击**绑定ECS**。

ENI列表							新增E	ENI
EnilD		EcsID		描述		操作	¢	
11-081-081-081-081-081-081-081-081-081-0				Created by OPEN API				

4. 在绑定ecs对话框中,选择相应的VpcID、EniID、分组和机器。

绑定ecs	>	×
* VpcID :	3	
* EnilD :		
* 分组:	ProdGroup V	
* 机器 :	192.168.1.242	

完成上述操作后, App Studio会为您创建弹性网卡,并绑定到机器实例。

### 公网访问

接下来,您可以通过将弹性网卡绑定至弹性公网IP的方式,将应用透出至公网。您也可以在其中加入负载均 衡的服务。

通过弹性公网IP将应用透出至公网的操作,如下所示。

- 1. 登录VPC控制台购买弹性公网IP,具体操作请参见申请EIP。
- 2. 绑定弹性网卡,具体操作请参见绑定弹性网卡。
- 3. 完成上述操作后,即可通过公网IP访问您的服务。

← → C ① 不安全 #/report       AppStudio     首页		×	os 😭 💻 🧱 🧷 米 🔹 🕕 🛈   📵 E
运营教程	活动管理	商品管理	订单管理
互联网产品运营管理包含产品管理、运营管理、团队管理、 广告管理、会员管理、安全管理…	活动运营,是一门说难又很简单,每个人都能成为票友"玩一 票"的工种;但也是一门说简单	商品管理是指一个零售商从分析顾客的需求人手,对商品组 合、定价方法、促销活动,以及	订单管理是客户关系管理的有效延伸,能更好的把个性化、 差异化服务有机的融入到客户
查看教程	点击进入	点击进入	点击进入
运营数据			
*时间: 请选择时间           搜索			
产品图片 产品名称	产品编号 订单数量	订单金额 订单时间	订单状态
	AppStudio ©2019 C	reated by DataWorks	

# 4.8. 第三方服务接入

# 4.8.1. 数据服务

本文为您介绍如何在App Studio中查看用户有权限调用的数据服务,并通过App Studio生成快速访问数据服务API的代码片。

如果您想要获取更多数据服务API的申请、SDK以及调用方法,请参见数据服务。

### 准备工作

开始操作前,您需要准备以下内容:

• 确认在数据服务中有相关工作空间的权限和API。

由于本文内容适用于有权限的数据服务API,所以请首先进入数据服务页面,查看是否有DataWorks工作空间,并查看相应工作空间下是否存在有权限的API。



● 在App Studio页面准备一个Java项目。

以springboot类型的项目为例,为您介绍代码片生成的功能。

- i. 进入App Studio页面, 单击工作空间页面的新建代码工程。
- ii. 填写新建项目对话框中的工程名和工程描述,选择运行环境为springboot。

三 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	工作2019 > 新建项目 新建项目	
横板空间	模板工程 代码工程 导入GN工程	
	* 工程名: 请编入工程名称,英文字符开头,只能包含数字、英文字符、	
	<ul> <li>工程描述: 資輸入工程描述</li> </ul>	7
	* 选择运行环境: react-component React + 銀件靜手架 文 React + 銀件靜手架	
	appstudio JDK8 + SpringBoot + 可貌化措建	
	<u>較</u>	

iii. 配置完成后,单击**提交**。

项目创建完成后,请确保pom.xml中有数据服务的依赖。

```
<dependency>
  <groupId>com.aliyun.dataworks</groupId>
  <artifactId>aliyun-dataworks-dataservice-java-sdk</artifactId>
  <version>0.0.1-aliyun</version>
</dependency>
```

# 在App Studio中使用数据服务

您可以直接在代码中使用数据服务,也可以在可视化搭建中使用数据服务。

• 直接在代码中使用数据服务。

此步骤将为您介绍如何在App Studio中方便地根据关键字、项目和业务分组查看可用的数据服务,同时利用生成代码片的功能快速生成,并调用某个数据服务API的代码。

i. 查看数据服务API列表。

单击App Studio页面右侧的**Data**,为您展示数据服务API列表。支持根据API名称、工作空间和服务 分组进行筛选。

★ Data +前往 DataService 新增 API						
Q 请输	入API名称	v		请选择服务分组		Runtime
ID	API名称	API Path	Project	API分组	操作	
1	J.	/demo/user	77379	TEST	选用   详情	Share
				く上一页	1 下一页 >	
						Data
						DataOS

ii. 在数据服务页面新增API。

单击右上角的**前往DataService新增API**,跳转至**数据服务**页面新增API,以满足调用API的需求。

iii. 查看数据服务API详情。

单击相应数据服务API右侧的详情,即可跳转至数据服务页面查看API详情。

⑤ 条 数据服务							<b>දැ</b> 2
API 详情 <b>一</b> API MI			r (#	週用地址	前开参数调用地址		
Ⅲ API 基本信息	请求参数						~
API ID 2095	▼ 应用请求参数						
API 分组 TEST 负责人	参数名称	参数类型			示例值		
创建时间 2019-03-07 19:40:13		string	EQUAL		001d9e7863049	001d9e7863049	
描述 user_query	返回参数						~
Interperation and a set of the set of t	▼ 应用返回参数						
APi调用地址 ht d(	参数名称	参数类型					
请求方式 GET		string				UID	
返回类型 JSON	region	string				地域	
	device	string					
名称 rds_workshop_log 类型 mysql							
连接信息 JDBC Url	gender	string					
用户名	age_range	string					
☆日 数振描述 rds日志数振同步	zodiac	string					

iv. 快速生成访问代码。

App Studio支持一键生成访问代码的方式, 自动填充appkey、appsecret, 生成样例代码, 方便您 直接插入项目。

单击相应数据服务API右侧的选用,即可打开包含样例访问代码的详情页。

接口详情		×		
Save				
1	package com.alibaba.dataworks.dataservice;	inninases-		
	import com.alivun.datawor Protocol:			
	import com.alivun.datawor nt:		TEST	选用 详情
	import com.aliyun.datawor t;			
	import org.slf4j.Logger:		く上一页	1 下一页 >
	import org.slf4j.LoggerFa			
	import org.springframewor	Sama 🔺		
	import org.springframewor	The second secon		
	import org.springframewor	"Million and a second and a sec		
	import org.springframewor			
	<pre>import org.springframework.web.bind.annotation.RestController;</pre>			
	<pre>import java.lang.reflect.Field;</pre>			
	import java.util.HashMap;			
	* 使用必读:			
	* - 生成代码示例仅供参考			
	* - 数据服务SDK调用文档: <u>https://yuque.antfin-inc.com/docs/share/b7d5e528-10f8-46b4-9a5e-d94fa94</u>			
	* - 使用前,请确保pom.xml包含[最新的,最新的,最新的]data-service-client依赖.			

完整的controller示例如下所示,仅供参考。在生成的InvokeApi2252方法中,您访问这个数据服务需要的path、host、key和secret都会被自动填充,ApiRequest2252DTO则包含了访问该服务的所有参数。

```
package com.alibaba.dataworks.dataservice;
import com.aliyun.dataworks.dataservice.model.api.protocol.ApiProtocol;
import com.aliyun.dataworks.dataservice.sdk.facade.DataApiClient;
import com.aliyun.dataworks.dataservice.sdk.loader.http.Request;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;
import java.lang.reflect.Field;
import java.util.HashMap;
/**
 * @author ****
 * @date 2019-03-21T17:23:17.040
 * - 使用前,请确保pom.xml包含最新的data-service-client依赖。
       <dependency>
           <groupId>com.alibaba.dataworks</groupId>
           <artifactId>data-service-client</artifactId>
 *
           <version>${latest-data-service-version}</version>
 *
       </dependency>
   - 使用前,确保配置spring config类,需要单独配置,不可与其他config合并。
 *
 *
      @Configuration
 *
      @ComponentScan(basePackageClasses = { DsClientConfig.class })
      public class DsClientConfig {
 *
          @Bean
           public BeanRegistryProcessor beanRegistryProcessor() {
              return new BeanRegistryProcessor();
 *
           }
 *
       }
 */
@RestController
public class Test2252Controller {
    private Logger logger = LoggerFactory.getLogger(Test2252Controller.class);
    @Autowired
    private DataApiClient dataApiClient;
    /**
     * Sample Result:
     * {
           "data": {
     *
     *
              "totalNum": 1000,
               "pageSize": 100,
               "rows": [
                  {
                       "pageNum": "...", // 分页默认参数: 页编号, Integer类型。
     ÷
                       "pageSize": "...", // 分页默认参数: 页大小, Integer类型。
                       "totalNum": "...", // 分页默认参数: 总记录数, Integer类型。
                       "id": "...", // Integer类型。
                       "name": "...", // String类型。
                       "sex": "....", // String类型。
                       "age": "...", // Integer类型。
                   }
     *
                   . . . . . .
              ],
     *
               "pageNum": 1
     *
           },
     *
           "errCode": 0,
           "requestId": "478cae2f-0***-42fb-a439-c0***e6f",
     *
           "errMsg": "success"
     * }
     */
    private HashMap InvokeApi2252(ApiRequest2252DTO dto) throws Exception {
       Portionst request - new Portionst () .
```

```
vednest tednest - new vednest(),
       request.setMethod("GET");
       request.setAppKey("15810204");
       request.setHost("http://example.aliyundoc.com");
       request.setPath("/test");
       for (Field f : dto.getClass().getDeclaredFields()) {
           try{
               if(f.get(dto)!= null) {
                  request.getBodys().put(f.getName(), f.get(dto).toString());
               }
           }catch(Exception e){}
       }
       request.setApiProtocol(ApiProtocol.HTTP);
       return dataApiClient.dataLoad(request);
   }
   /**
    * Response:
    */
   @RequestMapping(value = "/sample/test2252", method = RequestMethod.POST)
   public HashMap testApi(@RequestBody ApiRequest2252DTO dto) throws Exception {
       return InvokeApi2252(dto);
   }
}
/**
* Request
*/
class ApiRequest2252DTO {
   public Integer pageNum;
   public Integer pageSize;
   public Integer id;
   public String name;
   public String sex;
   public Integer age;
}
```

⑦ 说明 您可以参考生成的代码样例,也可以直接单击保存,将代码添加到当前代码目录的 dat aservice包中。

• 在可视化搭建中使用数据服务。

可视化搭建的组件和数据服务接口进行了深度的融合,数据服务返回数据的默认格式,即为可视化组件接收数据的格式。可以实现即配即用,详情请参见可视化搭建。

# 4.8.2. DataOS API

本文将为您介绍DataOS API的功能、输入、输出等详情,以及如何进行配置使用。

# CheckMetaTable

- 功能:判断table是否存在。
- 输入: tableGuid (必选)。
- 格式: odps.<project>. 。

- 输出: true或false。
- 示例如下:
  - 输入:

request.setTableGuid("odps.autotest.daily_test");

#### ○ 输出:

{"requestId":"0b85c9d915548770462378104e","errMsg":"success","errCode":0,"data":true}

# GetMetaDB

- 功能: 获取MaxCompute项目的信息。
- 输入:项目GUID(必选)。
- 格式: odps.<project> 。
- 输出: 项目详情。

参数	描述
appGuid	项目唯一标识
project	项目英文名称
projectNameCn	项目名称
comment	备注
ownerld	Owner的ID
createTime	创建时间
modifyTime	修改时间

• 示例如下:

#### 。 输入:

request.setDbGuid("odps.autotest");

```
○ 输出:
```

```
{
    "requestId": "0bfaefec61500****",
    "errMsg": "success",
    "errCode": 0,
    "data": {
        "appGuid": "odps.meta",
        "projectNameCn": "ODPS元仓",
        "comment": "",
        "ownerId": "13101879118",
        "createTime": "2014-02-18",
        "modifyTime": "2018-04-16"
    }
}
```

# GetMetaTable

- 功能: 获取MaxCompute表的信息。
- 输入: tableGuid (必选)。
- 格式: odps.<project>. 。
- 输出:表的详情。

参数	描述
appGuid	项目唯一标识
tableGuid	表唯一标识
tableName	表名称
id	数据库ID
ownerld	owner的ID
hasPart	是否为分区表
dataSize	表数据的大小
createTime	表的创建时间
last DdlT ime	表DDL最后的更新时间
last Modif yT ime	表最后的修改时间

#### • 示例如下:

。 输入:

request.setTableGuid(tableGuid);

```
○ 输出:
```

```
{
   "requestId": "0b8906da817****",
   "errMsg": "success",
    "errCode": 0,
    "data": {
       "appGuid": "odps.meta",
       "tableGuid": "odps.meta.m_table",
        "tableName": "m table",
       "id": 64809,
       "OwnerId": "dp-base-odps@example.com",
       "hasPart": 1,
        "dataSize": 49397610904693,
       "createTime": "2014-12-10 21:20:23",
       "lastDdlTime": "2017-04-18 10:10:06",
       "lastModifyTime": "2019-04-09 20:24:08"
 }
}
```

# ListMetaTableColumn

- 功能: 获取MaxCompute的列信息。
- 输入: tableGuid (必选)。
- 格式: odps.<project>. 。
- 输出: 列详情。

参数	描述
appGuid	项目唯一标识
tableGuid	表唯一标识
tableName	表名称
columnGuid	列唯一标识,格式为 odps. <project><c ol&gt;</c </project>
columnName	列名
columnType	列类型
seqNumber	列编号(从1开始)
isPartitionCol	是否为分区列
comment	备注
safeLevel	安全等级

● 示例如下:

#### 。 输入:

request.setTableGuid(tableGuid);

#### 。 输出:

```
{
   "requestId": "0b8906d9796****",
   "errCode": 0,
   "errMsg": "success",
   "columnList": [{
       "appGuid": "odps.meta",
       "tableGuid": "odps.meta.m table",
        "tableName": "m_table",
       "columnGuid": "odps.meta.m_table.project_name",
       "columnName": "project_name",
       "columnType": "string",
       "seqNumber": 1,
       "isPartitionCol": 0,
       "comment": "Project名称",
       "safeLevel": "C2"
 },
{
   "appGuid": "odps.meta",
   "tableGuid": "odps.meta.m_table",
   "tableName": "m table",
   "columnGuid": "odps.meta.m_table.name",
   "columnName": "name",
   "columnType": "string",
   "seqNumber": 2,
   "isPartitionCol": 0,
   "isPrimaryKey": 0,
   "isNullable": 0,
   "comment": "表名",
   "safeLevel": "C2"
 } ... ]
}
```

# ListMetaTablePartition

- 功能: 获取MaxCompute的分区信息。
- 输入:

参数	说明
tableGuid	格式为 odps. <project>.</project>
pageNum	页码
pageSize	每页最多显示记录数

• 输出: 表分区的详情。

参数	描述
appGuid	项目唯一标识
tableGuid	表唯一标识
tableName	表名称
partitionGuid	分区唯一标识,格式为 odps. <project> <partition></partition></project>
partitionName	分区名称
createTime	分区的创建时间
modifyTime	分区的修改时间
dataSize	分区的数据大小
records	分区的记录数
pageNum	当前分页页码
pageSize	当前分页大小
totalNum	总记录数

#### • 返回示例如下:

```
{
   "requestId": "0baf3e0502****",
   "errCode": 0,
   "errMsg": "success",
   "pageNum": 1,
   "pageSize": 10,
    "totalNum": 1101,
   "partitionList": [{
       "appGuid": "odps.meta",
       "tableGuid": "odps.meta.m_table",
        "tableName": "m table",
       "id": 168504514,
       "partitionGuid": "odps.meta.m_table.ds\u003d20190408",
        "partitionName": "ds\u003d20190408",
        "createTime": "2019-04-08 13:59:52",
       "modifyTime": "2019-04-08 19:54:51",
       "dataSize": 273248012568,
        "records": 720503170
 } ... ]
}
```

# SearchMetaTables

• 功能: 模糊查找表。

#### ● 输入:

参数	说明
keyword	表名称的关键字
pageNum	页码
pageSize	每页最多显示的记录数

### ● 输出:

参数	描述
appGuid	项目唯一标识
tableGuid	表唯一标识
tableName	表名称
ownerld	Owner的ID
createTime	表的创建时间
lastDdlTime	表DDL最后的更新时间
last Modif yT ime	表最后的修改时间

#### • 示例如下:

○ 输入:

request.setKeyword("test");

○ 输出:

```
{
   "message": null,
   "code": 200,
   "success": true,
    "data": {
       "requestId": "0be41b2227759****",
       "errCode": 0,
       "errMsg": "success",
       "pageNum": 1,
       "pageSize": 2,
       "totalNum": 5000,
   "data": [{
       "appGuid": null,
       "tableGuid": "odps.ant_p13n.finance_newsrec_tab_dataset_ds",
       "tableName": "finance newsrec tab dataset ds",
       "createTime": "2018-07-06 16:24:41",
       "lastModifyTime": "2019-04-26 10:49:23",
       "lastDdlTime": null,
       "lastAccessTime": null,
       "ownerId": "163585"
   },
    {
       "appGuid": null,
       "tableGuid": "odps.tbcdm.dws_tm_itm_cate_food_ftr_test_cm",
       "tableName": "dws tm itm cate food ftr test cm",
       "createTime": "2017-11-23 17:06:18",
       "lastModifyTime": "2019-04-26 20:34:12",
       "lastDdlTime": null,
       "lastAccessTime": null,
       "ownerId": "108292"
   }]
 },
   "timestamp": 1556452227875,
   "sessionId": null
}
```

# GetDQCEntity

- 功能: 获取分区表达式信息。
- 请求参数:

参数	类型	是否必选	示例值	描述
EnvType	STRING	否	MaxCompute	项目类型 <i>,</i> 包括 MaxCompute、 Hive和 Streaming。
MatchExpression	STRING	否	dt=\${yyyymmdd- 1}	分区表达式,可以 不填写。如果不填 写,则返回表下的 所有分区表达式。

#### APP Studio·功能介绍

参数	类型	是否必选	示例值	描述
ProjectName	STRING	否	autotest	MaxCompute项目 名称。
TableName	STRING	否	test_dqc_decimal _1119_2	表名称。

#### ● 返回参数:

参数	类型	示例值	描述
ReturnCode	STRING	0	返回码
ReturnValue	-	-	返回的分区表达式列表
ld	LONG	4003918	分区表达式ID
ProjectName	STRING	autotest	MaxCompute项目名称
TableName	STRING	test_dqc_decimal_1119 _2	表名称
EnvType	STRING	MaxCompute	项目类型,包括 MaxCompute、Hive和 Streaming
MatchExpression	STRING	dt=\${yyyymmdd-1}	分区表达式
EntityLevel	INT EGER	1	分区表达式级别 。 0: SQL级别 。 1: 任务级别
OnDuty	STRING	50624	分区表达式责任人
ModifyUser	STRING	50624	最近修改人
GmtCreate	STRING	2018-11-26 23:18:34	创建时间
GmtModify	STRING	2018-11-26 23:18:34	修改时间
Sql	INT EGER	0	是否支持修改为SQL级别 。 0: 支持 。 -1: 不支持
Task	INT EGER	0	是否支持修改为任务级别 。 0: 支持 。 -1: 不支持
Followers	STRING	050624	订阅人,逗号分隔

#### Dat aWorks

参数	类型	示例值	描述
HasRelativeNode	BOOLEAN	false	是否关联调度

#### ● 请求示例:

/?Action=GetDQCEntity &EnvType=odps &MatchExpression=dt=\${yyyymmdd-1} &ProjectName=autotest &TableName=test_dqc_decimal_1119_2 &<**公共请求参数**>

#### • 正常返回示例:

```
{
  "ReturnCode":"0",
  "ReturnValue":{
    "Entity":[
     {
        "EntityLevel":0,
        "EnvType":"odps",
        "Followers":"050624",
        "GmtCreate":"2018-11-26 15:06:32",
        "GmtModify":"2018-11-26 15:06:32",
        "HasRelativeNode":false,
        "Id":4003918,
        "MatchExpression":"dt=${yyyymmdd-3}",
        "OnDuty":"050624",
        "ProjectName":"autotest",
        "Sql":0,
        "TableName":"test dqc decimal 1119 2",
        "Task":0
      },
      {
        "EntityLevel":0,
        "EnvType":"odps",
        "Followers":"050624",
        "GmtCreate":"2018-11-26 22:31:13",
        "GmtModify":"2018-11-26 22:31:13",
        "HasRelativeNode":false,
        "Id":4003922,
        "MatchExpression":"dt=${yyyymmdd-1}",
        "OnDuty":"050624",
        "ProjectName":"autotest",
        "Sql":0,
        "TableName":"test_dqc_decimal_1119_2",
        "Task":0
      },
      {
        "EntityLevel":0,
        "EnvType":"odps",
        "Followers":"050624",
        "GmtCreate":"2018-11-26 23:18:34",
        "GmtModify":"2018-11-26 23:18:34",
        "HasRelativeNode":false,
        "Id":4003923,
        "MatchExpression":"dt=${yyyymmdd-2}",
        "OnDuty":"050624",
        "ProjectName":"autotest",
        "Sql":0,
        "TableName":"test dqc decimal 1119 2",
        "Task":0
      }
   ]
 }
}
```

• 异常返回示例:

```
{
    "ReturnCode":"500"
}
```

# GetDQCFollower

- 功能: 获取告警订阅信息。
- 请求参数:

参数	类型	是否必选	示例值	描述
Entityld	LONG	否	4003922	分区表达式ID
ProjectName	STRING	否	MaxCompute	MaxCompute项目 名称

#### ● 返回参数:

参数	类型	示例值	描述
ReturnCode	STRING	0	返回码
ReturnValue	-	-	返回的订阅列表
ld	LONG	4003918	分区表达式ID
ProjectName	STRING	autotest	MaxCompute项目名称
TableName	STRING	test_dqc_decimal_1119 _2	表名称
Entityld	STRING	4003922	分区表达式ID
Followers	STRING	050624	订阅人,逗号分隔
AlarmMode	INT EGER	1	包括以下4种告警模式: <ul> <li>1:邮件</li> <li>2:邮件和短信</li> <li>3:钉钉群机器 人/hook</li> <li>4:钉钉群机器人 @ALL</li> </ul>

● 请求示例:

/?Action=GetDQCFollower &EntityId=4003922 &ProjectName=odps &<**公共请求参数**>

• 正常返回示例:

```
{
    "ReturnCode":"0",
    "ReturnValue":{
        "Follower":[{
            "AlarmMode":1,
            "EntityId":"4003922",
            "Follower":"050624",
            "Id":1726,
            "ProjectName":"autotest",
            "TableName":"test_dqc_decimal_1119_2"
        }]
    }
}
```

• 异常返回示例:

```
{
    "ReturnCode":"500"
}
```

# GetDQCRule

- 功能: 获取规则详情。
- 请求参数:

参数	类型	是否必选	示例值	描述
ProjectName	STRING	否	autotest	MaxCompute项目 名称
Entityld	LONG	否	4003922	分区表达式ID

• 返回参数:

一级参数	二级参数	三级参数	类型	示例值	描述
ReturnCode	-	-	STRING	0	返回码
		ld	INT EGER	4003918	分区表达式ID
		ProjectName	STRING	autotest	MaxCompute 项目名称
		TableName	STRING	test_dqc_deci mal_1119_2	表名称
		EntityId	INT EGER	4003922	分区表达式ID
		Property	STRING	table_count	参数
		MethodId	INT EGER	8	采集方法ID
		MethodName	STRING	table_count	采集方法名称

一级参数	二级参数	三级参数	类型	示例值	描述
		OnDuty	STRING	050624	规则配置人
		RuleType	INT EGER	0	规则类型
		BlockType	INT EGER	1	强弱性 。 0: 弱规则 。 1: 强规则
		TemplateId	INT EGER	7	模板ID
	TemplateRule s(模板规则列 表)	T emplat eNam e	STRING	SQL任务表行 数,1、7、30 天波动检测	模板名称
		RuleCheckerRe lationId	INT EGER	1008007	规则内部关联ID
		Checkerld	INT EGER	7	校检器ID
		FixCheck	BOOLEAN	false	是否是固定值 校验
		Trend	STRING	ир	趋势
		WarningThres hold	STRING	20	橙色阈值
		CriticalT hresh old	STRING	90	红色阈值
		HistoryWarnin gThreshold	STRING	history max: 40%,hist ory min: 10%	历史橙色告警 阈值
		HistoryCritical Threshold	STRING	history max:40%,hist ory min:10%	历史红色告警 阈值
	PropertyKey	STRING	table_count	用于DQC前端 规则联动,可 以忽略	
		MatchExpressi on	STRING	dt=\${yyyymm dd-1}	分区表达式
		ProjectName	STRING	autotest	MaxCompute 项目名称
		TableName	STRING	test_dqc_deci mal_1119_2	表名
	Id	INTEGER	279580664	规则ID	

一级参数	二级参数	三级参数	类型	示例值	描述
		Entityld	INT EGER	4003922	分区表达式ID
ReturnValue( 返回的规则列		Property	STRING	table_count	参数
表)		MethodId	INT EGER	21	采集方法ID
		MethodName	STRING	count或 table_count	采集方法名称
		WhereCondit i on	STRING	id>10	过滤条件或自 定义SQL
		OnDuty	STRING	050624	规则配置人
		RuleType	INT EGER	1	规则类型
		BlockType	INT EGER	1	强弱性: 。 0: 弱规则 。 1: 强规则
		TemplateId	INTEGER	7	模板ID
	TemplateNam e	STRING	SQL任务表行 数 <i>,</i> 1、7、30 天波动检测	模板名称	
	RuleCheckerRe lationId	INT EGER	1008006	数据质量监控 的内部规则映 射表	
		Checkerld	INT EGER	6	校检器ID
SelfserviceRul es(自定义规 则列表)	Checker	INT EGER	9	校验器,此参 数对应于前端 的ID标识,需要 由pkld转换	
	CheckerName	STRING	compared with a fixed value	校检器名称	
		FixCheck	BOOLEAN	false	是否是固定值 校验
		Trend	STRING	abs	趋势
	CheckResult	INT EGER	0	规则的校检结 果,该接口不 返回,可以忽 略	

一级参数	二级参数	三级参数	类型	示例值	描述
		WarningThres hold	STRING	20	橙色阈值
		CriticalT hresh old	STRING	60	红色阈值
		HistoryWarnin gThreshold	STRING	history max:40%,hist ory min:10%	历史橙色告警 阈值
		HistoryCritical Threshold	STRING	history max:40%,hist ory min:10%	历史红色告警 阈值
		HistoryActual Threshold	STRING	history max:10%,hist ory min:10%	历史真实波动 率
	PropertyKey	STRING	table_count	用于DQC前端 规则联动进行 映射,可以忽 略	
	MatchExpressi on	STRING	dt=\${yyyymm dd-1}	分区表达式	

# GetQualityByEntity

- 功能:根据分区表达式获取质量汇总信息。
- 请求参数:

参数	类型	是否必选
Action	STRING	是
EndDate	STRING	是
Entityld	INTEGER	是
PageSize	INTEGER	是
PageStart	INTEGER	是
ProjectName	STRING	是
StartDate	STRING	是

● 返回参数:

参数	类型	描述
Count	INTEGER	搜索结果的总行数
ReturnCode	STRING	返回码
ReturnValue	-	分区下规则的校验详情

#### ● 请求示例:

```
http(s)://[Endpoint]/?Action=GetQualityByEntity
&EndDate=2019-08-25 00:00
&EntityId=1526081
&PageSize=15
&PageStart=1
&ProjectName=autotest
&StartDate=2019-08-20 00:00:00
&<公共请求参数>
```

● 返回示例:

```
{
  "returnCode": "0",
  "returnMessage": null,
  "returnErrorSolution": null,
 "returnErrorOper": null,
  "requestId": "40911dfc-a936-4092-bfe9-6fcd****",
  "returnValue": [{
    "id": 787700746,
   "taskId": "15668171746013d72950384444****80",
    "entityId": 1508198,
    "ruleId": 28410562,
    "property": "table count",
   "bizdate": 1566662400000,
   "dateType": "YMD",
    "actualExpression": "ds=20190826",
    "matchExpression": "ds=${yyyymmdd}",
    "blockType": 1,
    "checkResult": 0,
    "checkResultStatus": 0,
    "methodName": "count",
    "comment": null,
    "whereCondition": "",
    "beginTime": 1566817174000,
    "endTime": 1566817355000,
    "timeConsuming": "181s",
    "externalType": "CWF2",
    "externalId": "100033671",
    "discrete": false,
    "fixedCheck": true,
    "referenceValue": [{
     "bizDate": 32535100800000,
     "discreteProperty": "table_count",
     "value": 0.0,
```

```
"threshold": null,
  "singleCheckResult": 0
}],
"sampleValue": [{
 "bizDate": 1566662400000,
  "discreteProperty": null,
 "value": 1306012.0
}],
"checkResultDetail": {
  "checkerId": null,
 "checkerType": null,
 "isDiscrete": false,
 "warningThreshold": null,
  "criticalThreshold": null,
  "op": ">",
  "expectValue": 0,
  "trend": null,
  "externalId": "100033671",
  "status": 0,
  "qualityStatus": 0,
  "actualResult": null,
  "detail": [{
    "currentSample": {
      "bizdate": 1566662400000,
      "property": "table_count",
      "distinctProperty": null,
      "value": 1306012,
      "detail": "1306012"
    },
    "historicalSamples": [{
      "bizdate": null,
      "property": null,
      "distinctProperty": null,
      "value": null,
      "detail": "--",
      "isExisted": null,
      "fluctuatedValue": null,
      "fluctuatedValueType": null,
      "checkResult": null
   }]
 }]
},
"trend": "abs",
"warningThreshold": null,
"criticalThreshold": null,
"expectValue": 0.0,
"op": ">",
"projectName": null,
"tableName": null,
"templateId": null,
"templateName": null,
"resultStr": null,
"checkerId": null,
"checkerType": 0,
            II.....] = NT= ... = II .
```

```
"ruleName": null,
"isPrediction": false,
"upperValue": null,
"lowerValue": null,
"checkerName": "与固定值比较"
}]
}
```

# GetQualityByRule

- 功能: 根据规则查询校验详情。
- 请求参数:

参数	类型	是否必选
Action	STRING	是
EndDate	STRING	是
PageSize	INTEGER	是
PageStart	INTEGER	是
ProjectName	STRING	是
RuleId	INTEGER	是
StartDate	STRING	是

#### ● 返回参数:

参数	类型	描述
Count	INT EGER	数据总条数
ReturnCode	STRING	返回码
ReturnValue	-	返回的校验详情

#### ● 请求示例:

```
http(s)://[Endpoint]/?Action=GetQualityByRule
&EndDate=2019-08-26 00:00:00
&PageSize=15
&PageStart=1
&ProjectName=cdo_meta
&RuleId=28791246
&StartDate=2019-08-22 00:00:00
&<公共请求参数>
```

● 返回示例:

```
{
   "returnCode": "0",
```

"returnMessage": null,

```
"returnErrorSolution": null,
"returnErrorOper": null,
"requestId": "f34d529b-d559-4699-b850-9***b58",
"returnValue": {
 "ruleCheckDto": [{
   "id": 787723200,
   "taskId": "156681910181215a8b8f9cd154***34e7d",
   "entityId": 1346320,
   "ruleId": 28791246,
   "property": "-",
   "bizdate": 1566662400000,
   "dateType": "YMD",
   "actualExpression": "NOTAPARTITIONTABLE",
   "matchExpression": "NOTAPARTITIONTABLE",
   "blockType": 1,
   "checkResult": 0,
   "checkResultStatus": 0,
   "methodName": "table count",
   "comment": "监控表行数波动",
   "whereCondition": null,
   "beginTime": 1566819101000,
   "endTime": 1566819103000,
   "timeConsuming": "2s",
   "externalType": "CWF2",
   "externalId": "1607534",
   "discrete": false,
   "fixedCheck": false,
   "referenceValue": [{
     "bizDate": 32535100800000,
     "discreteProperty": null,
     "value": 3778.0,
     "threshold": 0.1852832186,
     "singleCheckResult": 0
   }, {
     "bizDate": 32535100800000,
     "discreteProperty": null,
     "value": 3772.0,
     "threshold": 0.3446447508,
     "singleCheckResult": 0
   }, {
      "bizDate": 32535100800000,
     "discreteProperty": null,
     "value": 3691.0,
     "threshold": 2.5,
     "singleCheckResult": 0
   }],
   "sampleValue": [{
     "bizDate": 1566662400000,
     "discreteProperty": null,
     "value": 3785.0
   }],
   "checkResultDetail": {
     "checkerId": null,
     "checkerType": null,
```

```
"isDiscrete": false,
  "warningThreshold": 10,
  "criticalThreshold": 50,
  "op": null,
  "expectValue": null,
  "trend": null,
  "externalId": "1607534",
  "status": 0,
  "qualityStatus": 0,
  "actualResult": null,
  "detail": [{
   "currentSample": {
      "bizdate": 1566662400000,
      "property": "-",
      "distinctProperty": null,
      "value": 3785,
      "detail": "3785"
   },
    "historicalSamples": [{
     "bizdate": 32535100800000,
      "property": null,
      "distinctProperty": null,
      "value": 3778,
      "detail": "3778, 0.1852832186%",
      "isExisted": null,
      "fluctuatedValue": 0.185283,
      "fluctuatedValueType": null,
      "checkResult": 0
   }, {
      "bizdate": 32535100800000,
      "property": null,
      "distinctProperty": null,
      "value": 3772,
      "detail": "3772, 0.3446447508%",
      "isExisted": null,
      "fluctuatedValue": 0.344645,
      "fluctuatedValueType": null,
      "checkResult": 0
   }, {
      "bizdate": 32535100800000,
      "property": null,
      "distinctProperty": null,
      "value": 3691,
      "detail": "3691, 2.5%",
      "isExisted": null,
      "fluctuatedValue": 2.5,
      "fluctuatedValueType": null,
      "checkResult": 0
   }]
 }]
},
"trend": "abs",
"warningThreshold": 10.0,
"criticalThreshold": 50.0,
```

"expectValue": null, "op": "abs", "projectName": null, "tableName": null, "templateId": 7, "templateName": null, "resultStr": null, "checkerId": null, "checkerType": 1, "ruleName": "表行数监控", "isPrediction": false, "upperValue": null, "lowerValue": null, "checkerName": null }], "count": 4325, "checkerType": 0

# 使用DataOS API

}

#### 添加Maven依赖如下所示。

```
<dependency>
   <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-dataworks-enterprise-ultimate</artifactId>
    <version>0.0.4</version>
</dependency>
<dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.8.5</version>
</dependency>
<!-- 要求gson 2.8.5以上版本 -->
    <dependency>
        <groupId>com.aliyun</groupId>
        <artifactId>aliyun-java-sdk-core</artifactId>
        <version>4.4.0</version>
    </dependency>
<dependency>
   <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
   <scope>test</scope>
</dependency>
```

Java代码如下所示,其中创建IClient Profile时,需要指定云账号的AccessKeyID和AccessKeySecret,详情请参见下文的常见问题。

⑦ 说明 将测试代码直接放在src/test/java目录下,会有UT快速启动入口。

import com.aliyuncs.DefaultAcsClient;
```
import com.aliyuncs.IAcsClient;
import com.aliyuncs.dataworks enterprise ultimate.model.v20190424.CheckMetaTableRequest;
import com.aliyuncs.dataworks enterprise ultimate.model.v20190424.CheckMetaTableResponse;
import com.aliyuncs.dataworks enterprise ultimate.model.v20190424.GetMetaDBRequest;
import com.aliyuncs.dataworks enterprise ultimate.model.v20190424.GetMetaDBResponse;
import com.aliyuncs.dataworks enterprise ultimate.model.v20190424.GetMetaTableRequest;
import com.aliyuncs.dataworks enterprise ultimate.model.v20190424.GetMetaTableResponse;
import com.aliyuncs.dataworks enterprise ultimate.model.v20190424.ListMetaTableColumnReques
t;
import com.aliyuncs.dataworks enterprise ultimate.model.v20190424.ListMetaTableColumnRespon
se;
import com.aliyuncs.dataworks enterprise ultimate.model.v20190424.ListMetaTablePartitionReq
uest;
import com.aliyuncs.dataworks enterprise ultimate.model.v20190424.ListMetaTablePartitionRes
ponse;
import com.aliyuncs.dataworks enterprise ultimate.model.v20190424.SearchMetaTablesRequest;
import com.aliyuncs.dataworks enterprise ultimate.model.v20190424.SearchMetaTablesResponse;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;
import com.google.gson.Gson;
import org.junit.BeforeClass;
import org.junit.Test;
public class Sample {
   private static final String REGION ID = "cn-shanghai";
   private static IAcsClient client = null;
    @BeforeClass
   public static void setup() throws ClientException {
        String endpoint = "dataworks-ee-ue-share." + REGION ID + ".aliyuncs.com";
        IClientProfile profile = DefaultProfile.getProfile(
            REGION ID,
            "<!!!!your.accessId>",
            "<!!!!your.accessKey>");
        DefaultProfile.addEndpoint(REGION ID, "dataworks-enterprise-ultimate", endpoint);
        client = new DefaultAcsClient(profile);
    }
    @Test
    public void testCheckMetaTable() throws ServerException, ClientException {
        String tableGuid = "odps.meta.m table";
        CheckMetaTableRequest request = new CheckMetaTableRequest();
        request.setTableGuid(tableGuid);
        CheckMetaTableResponse response = client.getAcsResponse(request);
        System.out.println(new Gson().toJson(response));
    }
   @Test
    public void testGetProject() throws ServerException, ClientException {
        String appGuid = "odps.meta";
        GetMetaDBRequest request = new GetMetaDBRequest();
       request.setDbGuid(appGuid);
       GetMetaDBResponse getMetaDBResponse = client.getAcsResponse(request);
        System.out.println(new Gson().toJson(getMetaDBResponse));
    }
    0Test
```

```
public void testGetPartitions() throws ServerException, ClientException {
    String tableGuid = "odps.meta.m table";
    ListMetaTablePartitionRequest request = new ListMetaTablePartitionRequest();
    request.setTableGuid(tableGuid);
    request.setPageNum(1);
    request.setPageSize(10);
    ListMetaTablePartitionResponse response = client.getAcsResponse(request);
    System.out.println(new Gson().toJson(response));
1
@Test
public void testSearchTables() throws ServerException, ClientException {
    SearchMetaTablesRequest request = new SearchMetaTablesRequest();
    request.setKeyword("test");
    request.setPageNum(1);
    request.setPageSize(10);
    SearchMetaTablesResponse response = client.getAcsResponse(request);
    System.out.println(new Gson().toJson(response));
QTest
public void testGetColumns() throws ServerException, ClientException {
   String tableGuid = "odps.meta.m_table";
    ListMetaTableColumnRequest request = new ListMetaTableColumnRequest();
   request.setTableGuid(tableGuid);
   ListMetaTableColumnResponse response = client.getAcsResponse(request);
   System.out.println(new Gson().toJson(response));
}
@Test
public void testGetTable() throws ServerException, ClientException {
   String tableGuid = "odps.meta.m_table";
    GetMetaTableRequest request = new GetMetaTableRequest();
    request.setTableGuid(tableGuid);
    GetMetaTableResponse response = client.getAcsResponse(request);
    System.out.println(new Gson().toJson(response));
```

## 常见问题

}

• 无法访问API, 错误提示如下所示:

Exception in thread "main" com.aliyuncs.exceptions.ClientException: InvalidApi.NotFound : Specified api is not found, please check your url and method. RequestId : B081CCF1-9F19-473E-9B99-68F20****

#### 错误原因:没有获取API权限。

• 如何查询AccessKeyID和AccessKeySecret?

单击页面右上角账号下的accesskeys,即可进行查询。

消息 ⁵⁵ 费用	工单	备案	企业	支持与服	鎊	>_	Ħ	简体	中文	0
			首次实	aui 🌔			-	••••		
					基本资	料	实名认	ん证	安全	设置
				ø	安全	管控				
				е	访问	腔制				
				8	acce	esskeys				
				~	会员	权益				
				•	会员	积分				
					推荐	返利后台	â			
						退	出管理	控制台		

#### • 调试时启动服务,配置项报错Circular Reference。

java.lang.IllegalArgumentException: Circular placeholder reference 'dataworks.dataos. auth.accessId' in property definitions



Run/Debug Configurations			×
添加 删除	Name: Unnamed		
✓ ■ Application Unnamed	* Main class: (j)	请选择	
	VM options:	-Ddataworks.dataos.auth.accessId=xxxx -Ddataworks.dataos.auth.accessKev=xxxx	
	Program arguments:		
	Environment Variables:		
	JRE: PORT:	1.8-SUK 7001	
	购买资源包 机器:	4vCPU, 8G内存	
	Pre-Launch Option: ()	请选择	
	开启HOTCODE:	● 是 ○ 否	
		Cancel Apply	ок

# 4.9. 可视化搭建 4.9.1. 可视化搭建概述

App Studio可视化搭建是辅助生成前端页面的工具,提供了一系列常见的网页组件,让开发者可以通过简单的拖拽,便可生成前端页面。本文将为您介绍App Studio可视化搭建的特点。

#### 框架无感知

无论您使用React、Angular或Vue, App Studio可视化搭建系统都可以适配,因为底层使用了一种通用的描述语言来描述页面的结构、表现、行为等属性。



# 集成简单的数据处理来满足复杂交互需求

App Studio集成了一个全局的状态管理方案,来完成页面数据管理以及组件之间的交互。



## 提供代码模式来满足复杂交互页面的搭建

App Studio可视化搭建的底层使用了通用的结构化可描述性语言(DSL)作为中间层。您可以直接基于DSL 进行代码模式的修改,实现了代码模式与可视化拖拽模式的互转,对于高阶开发者来说这是一种进阶的使用 体验。

## 可视化方式配置组件联动

App Studio可视化搭建提供了一种非常简单的可视化连线方式,来进行组件之间的交互联动。



# 无需构建即可直接发布运行

App Studio可以将中间层的DSL在线编译为一份可直接在浏览器中执行的代码,进行页面渲染。

## 对接DataWorks数据服务,快速集成数据接口

App Studio无缝对接DataWorks数据服务接口,可实时调试接口。

## 丰富的组件、模板市场

App Studio提供了丰富的组件,同时支持您自定义组件并上传到组件库。

另外, App Studio也提供了丰富的模板, 您可以直接基于某一个模板快速生成页面, 也可以将页面保存为模板并发布到模板市场供他人使用。

# 4.9.2. 基本使用

本文将为您介绍可视化搭建系统的新建工程、可视化搭建等基本操作。

#### 新建工程

1. 进入App Studio页面,单击工作空间页面的新建代码工程。

⑤ ▲ App Stud	lio		
三 ① 工作空间			
Q 应用空间	欢迎来到 App Studio		
◆ 模板空间     ◆			
	Ŷ	លិ	Ø
	新建模板工程	新建代码工程	导入Git工程

2. 填写新建项目对话框中的工程名和工程描述,选择运行环境为appstudio。

6	App Studio	)						
© Q	三 工作空间 应用空间	工作空间 > 新建项目 新建项目						
Ŷ	模板空间	模板工程(代码工	程 导入GIt工程					
		* 工程名:	请输入工程名称,英文字符开头,只能包含数	如子、英文 <del>字</del> 符、				
		* 工程描述:	请输入工程描述					
		*选择运行环境:	react-component React + 组件脚手架	~	react-demo React	~	springboot JDK8 + SpringBoot	
		[	appstudio JDK8 + SpringBoot + 可铌化搭建					
		提交						

- 3. 配置完成后,单击**提交**。
- 4. 打开santa/pages目录。



5. 单击任意一个.santa文件进入可视化搭建。

您也可以右键单击pages,选择新建 > 模板文件,基于模板进行开发。

6		🛆 Арр	) Studi	0			
ធ៌		工程	文件	编辑	ŧ J	版本	查看
ര	ב	[程					
<b>U</b> '	te	est 🕧					
	>	alicode.					
Y	>	.settings					
	>	APP-MET	A				
	~	/ santa					
		> pages	3 —	_			-
		index	c.htı 📑	瓍	>	X	#
	>	src	á	服建文件	挟	模	反文件
	>	target	-	L传文件	ŧ		
		🛓 .classpa	th ī	<b>[</b> 命名			
		≣ .factory	oath g	夏制 Ct	rl C		
		gitignor .	e	沾贴 Ct	rl V		
		≣ .project	я	開設会 Ctu	rl 🗆		
		pom.xm	I .				

## 可视化搭建

可视化搭建页面主要由组件列表和操作面板组成。

6	🛆 Арр	Studio											ø	开发	<i>8</i> iz	維	•	-	
ធ	工程	文件	编辑	版本	查看	调试	设置	发布	模板	帮助	反馈			۲	414	Ed	it Config	<b>v )</b>	
n.	≣ test.sa	nta ×																	Ξ
۲.	日布局	88 基础	≣表	单(	⁽¹⁾ 翻表	台高级	搜索	组件	Q	组件	列表			•	Ð	Э	89	50	•
Ÿ	用户访问	可来源									每周用	户活跃量							
			а	lipay: 0.	21		taob	ao: 0.29			8 7 6	10 10 10							
			tianmao:	0.17		alia		aliyun: 33'	6		5 4 3	i0 i0 i0							

• 组件列表

组件列表为您展示可视化搭建系统中,所有的系统预设组件,包括**布局、基础、表单、图表和高级**等组件。



Ĝ App Studio	♂开发 ♂运堆 ■
① 工程 文件编辑 版本 查看 画版 设置发布 機板 帮助 反馈	🛡 🛃 👯 Edit Config 🗸 🕨 🌺 🔳
E test.santa ×	=
日本時の設備を設備を改善する。	_ <u> </u>
	API 组件配置 3
	雇性 样式 高级
	י Layout צַי
jtanmao: 0.17 80	Display Setting
aliyur: 0.3	ABC ABC Ø
50	P
taobao.0.29 🚦 40	U \$
30	Ţ
alipsy. 0.21/ 20	04 0 0 04 0 g
• tanmao • taobao • alipay • aliyon 10 · · · · · · · · · · · · · · · · · ·	PACONG 🗻 80S
	0
	Width auto - Height auto -
	Min W auto - Min H auto -
	Max W none - Max H none -
订单 运单趋势	
	> Text
订单/运单趋势	> Background
160	> Border

展开组件列表,拖拽某一个组件至可视化操作面板,单击该组件,即可在右侧进行组件配置。

● 操作面板

操作面板包括**撤销、重做、预览、代码模式、全局样式、导航配置、全局数据流配置、发布为模 板**和保存等操作。



单击操作面板中的导航配置图标,即可打开导航配置页面进行配置,详情请参见导航配置。

#### 配置全局数据流

配置全局数据流的详情请参见全局数据流。

• 配置组件属性

组件属性配置面板主要负责可视化的方式配置组件属性。

根据组件的属性配置规则,组件属性配置面板将会生成一个可视化表单,让您输入组件的属性配置。在组件属性配置表单中更改组件属性后,可视化操作区域将会根据接收到的组件属性,进行重新渲染。您可以 实时查看组件不同属性的渲染结果。

• 配置组件样式

组件样式面板主要负责组件样式的相关设置。

组件样式配置面板将会生成一个通用的样式配置可视化页面,您可以基于该面板定制组件基本的外观样 式,包括布局、文字、背景、边框、效果等常用样式配置。 在组件样式配置面板中添加、修改组件样式,可视化搭建系统将会收集所有的样式设置到组件上,可视化 操作区域将会根据新的样式设置重新渲染对应组件,您可以实时查看配置后的组件效果。

• 配置组件联动高级

组件联动高级设置面板主要负责组件之间的联动设置。

单击可视化操作区域中的某一个组件,选中高级面板。高级设置面板中,将会在左侧列出当前选中组件对 应的组件属性,单击右侧的放大镜按钮选择需要关联的另一个组件。

组件配置				
属性	样式	高级		
	BigFo	orm	请选择联动组件	٩
ID	: Bigfurt			
	lay	out		
	formlte	ems		
	searchī	- ext		
	onSea	rch		
sea	rchParam	าร		

组件配置		
属性	样式	高级
Bi	igForm	DataTable Q
ID:		ID:
	layout	requestUrl
forr	nltems	requestMethod
sear	chText	requestParams
ons	Search	dataFilter
searchPa	rams	columns
		size
		bordered
		showHeader
		pageSize

选中需要关联的另一个组件后,高级设置面板右侧将会出现对应的组件属性。

• 单击左侧属性列表中的某一个属性, 连线至右侧属性列表中的另一个属性。

组件配置	
属性  样式	高级
BigForm	DataTable Q
ID: I	ID:
layout	requestUrl
formItems	requestMethod
searchText 🦼	requestParams
onSearch	dataFilt
searchParams 🥤	搜索参数(code)
	size
	bordered
	showHeader
	pageSize

该操作将会实现两个组件之间的属性联动,左侧组件的searchParams参数变更将会及时传递到右侧组件的requestParams参数,从而实现两个组件基于属性之间的联动配置。

## 代码模式

代码模式提供了一种更高级的方式来满足更复杂的交互场景的需求,详情请参见代码模式。

#### 保存、预览、运行和热部署

详情请参见保存、预览、运行和热部署。

# 4.9.3. 常用组件

APP Studio可视化搭建系统自带80多个组件,可以满足您搭建基本页面的需求。本文将为您介绍可视化搭建系统默认自带的组件。

## 布局组件

#### 布局组件为您提供一个24栅格系统组件。

API	组件配置	
	属性 样式 高级	
I		
	自定义比例 12:12	
	水平排列方式	
	两端对齐(两边留白)	~
	■重直方向对齐方式	
	顶部对齐	~
	【栅格间隔宽度 ──	

● 栅格比例

系统默认将24栅格切割成一个12:12的栅格系统,您可以切换至其他常见的栅格比例,也可以自定义栅格比例。只需要保证所有栅格比例加起来是24的总数,布局组件将会根据各个栅格的比例进行布局切割。

• 水平排列方式

水平排列方式定义了栅格在父节点中的排版方式。



• 垂直排列方式

垂直排列方式定义了子元素垂直方向的对齐方式。



#### ● 栅格间隔宽度

栅格常常需要和间隔进行配合,您可以使用该配置来定义栅格间隔。

col-6	col-6	col-6	col-6

• 区块容器

区块容器是一个块状的容器组件,区块容器组件可以作为一系列组件的父组件,类似于HT ML中的div容器。

#### 基础组件

基础组件均支持组件相关的常用属性设置。

- 文字
  - 文字
     単一
     単一
     単一
     単二
     <li
  - 。 段落

API	组件配置	
Content		
	属性 样式 高级	
	组件大小	
	中号	
	117公方式展示校落	
	展示所有文本	

■ 组件大小

定义了段落文字大小。

■ 什么方式展示段落

用于区分短文本和长文本,短文本的行间距会更小(通常三行以内)。

- 媒体
  - 视频



- **视频链接**:需要播放的视频地址。
- 封面地址:视频封面图片地址。

■ 是否自动播放:是否在组件加载完之后自动播放视频。

○ 图片

API 组件配置
属性样式高级
┃链接地址
https://ima.alicdn.com/tfs/TB1p8phNi

链接地址:显示的图片地址,可以上传图片。

图标



- 。 **指定亚小咖**杆 图 你 指定图标的类型。
- 按钮



按钮属性的详情请参见按钮文档。

# 链接 链接 #P 组件配置 属性 样式 高级 IGH表示 Gittabut IGH表示 IGH表示 IGH表示 IGH表示 IGH表示 IGH表示 IGH表示 IGH表示 IGH表示 IGHALL IGHALL</

- 链接地址:单击链接的跳转地址。
- 链接属性: 在本窗口打开和在新窗口打开。

## 表单组件

				 	 	API	组件配置		
*	名称:	请输入名称	搜索	 	 		属性		
							┃布局方式		
							行内		
							表单项		
							输入类	型	
							文本输	入框	
							字段名	称	
							name		$\{\cdot\}$
							标签文	案	
							名称:		Ð
							Placet	older	
							请输入	名称	{}
							必填 ()		
							表单验	证报错信息	
							请输入	名称	$\{\cdot\}$
							默认值		_
							请输入		Ð
							数据源		
							+		
							X		
Boo	ly → Big	gForm					+		
± /	ъ́ – +:		ᅣᆂᅳᆅᅔ						

表单包括行内、水平和垂直三种布局方式。 上传图片和附件详情请参见<mark>上传附件</mark>。 筛选详情请参见<mark>搜索</mark>。 输入框详情请参见<mark>输入框</mark>。

## 图表

● 数据表格

#### Dat aWorks

		API	组件配	<b>E</b>		
ID	Name		属性	样式		
			数据测	0		
100	ajkoajkoajkoajkoajkoajkoajkoajkoajko					{}
			请求之	;;#		
101	ajkoajkoajkoajkoajkoajkoajkoajkoajko		Get	112		
			□畑赤参	2 <i>**</i> #7		
102	ajkoajkoajkoajkoajkoajkoajkoajkoajko		安量	nau Z	娈量值	操作
102	aikasikasikasikasikasikasikasika					
105	ajkoajkoajkoajkoajkoajkoajkoajkoajko					
104	aikoaikoaikoaikoaikoaikoaikoaikoaiko					
		_	新増			编辑代码
105	ajkoajkoajkoajkoajkoajkoajkoajkoajkoajko		■返回数	。 (据处理函数	友	
		-	编辑	代码		
106	ajkoajkoajkoajkoajkoajkoajkoajkoajko		表格列	配置项		
		-	-	字段	显示列名	
107	ajkoajkoajkoajkoajkoajkoajkoajkoajko		i i	d	ID	
108	ajkoajkoajkoajkoajkoajkoajkoajkoajko		r 🗹	hame	Name	
100	aikaaikaaikaaikaaikaaikaaikaaika	编辑代码				
109	ajkvajkvajkvajkvajkVajKVajKVajKVajKV		尺寸			
			正常			
			是否显	示表格边相	E	

Body > DataTable

配置	说明
数据源	请求接口地址。
请求方法	请求方法: Get/Post/Put/Delete。
搜索参数	接口请求参数。
返回数据处理函数	接口数据返回后的数据处理函数。
表格列配置项	定义表格需要显示的表格列。
尺寸	设置表格尺寸。
是否显示表格边框	设置是否显示表格边框。
是否显示表格头	设置是否显示表格头。

是否显示表格头

带分页的数据表格多了一项每页显示数量的配置项,定义分页中每一页的显示数量。

• Excel

A B C 1 1 2 3 2 4 5 6	API 组件配置 <u>属性</u> 致据源 请输入 请输入	样式 高級 【♪ 【♪ 【♪ ② 【】 ② 【】 】 ②				
	新増   返回数据公 編輯代码   数据   编辑代码	编辑代码 计理函数 3				
配置	说明					
数据源	请求接口地址。					
请求方法	请求方法: Get/Post/Put/Delete。					
搜索参数	接口请求参数。					
返回数据处理函数	接口数据返回后的数据处理函数。					
数据	直接配置Excel需要显示的数据。					

● 折线图



lv	>			n	е
- /					

配置	说明
数据源	请求接口地址。
请求方法	请求方法:Get/Post/Put/Delete。
搜索参数	接口请求参数。
返回数据处理函数	接口数据返回后的数据处理函数。
图表配置	通过代码对图表进行配置。
是否显示图表标题	设置是否显示图表标题。
图表标题	显示图表标题。
图表数据	直接配置图表需要显示的数据。
X轴字段	定义返回数据中显示到X轴的数据字段名。
Y轴字段	定义返回数据中显示到Y轴的数据字段名。

? 说明 柱状图、条形图、面积图、饼图、地图、词云和散点图等图表组件的配置,请参见折线 图。

# 高级组件

高级组件均支持组件相关的常用属性设置。

- 选择组件包括选择器、复选按钮、级联选择、单选框、区段选择器、开关组件和评分。
- 交互:您可以通过Tab选项卡,在不同子任务、视图、模式之间切换,它具有全局导航的作用,是全局功能的主要展示和切换区域。详情请参见Tab选项卡。
- 轮播图:轮播组件以幻灯片的方式,在页面中横向展示诸多内容的组件。详情请参见图片轮播。
- 步骤条:默认情况下,Step定义为展示型组件。上层组件可以通过修改传入的current属性值来修改当前的步骤,同时可以设置每个节点的click事件,来自定义回调。详情请参见步骤。
- 进度条:进度指示器可以为您展示操作的当前进度。详情请参见进度指示器。
- 菜单:您可根据自身需求选择相应的菜单,详情请参见菜单。
- 导航:导航包括顶部导航和侧边导航。顶部导航提供全局性的类目和功能,侧边导航提供多级结构来收纳 和排列网站架构。详情请参见导航。

# 4.9.4. 代码模式

代码模式提供了一种更高级的方式来满足更复杂的交互场景的需求。

打开新建工程中santa/pages目录下的.santa文件,进入可视化搭建。

单击操作面板中的**代码模式**图标,即可在页面右侧出现代码区域。 —————————————————————



可视化搭建使用DSL描述语言作为中间层的代码,基于该DSL进行可视化与代码模式的互转。可以简单地将 DSL看作简化版的React,语法与React基本一致。

DSL将一个组件使用标签进行描述,标签的属性即组件的Props属性。属性值支持简单的数据类型,例如 STRING或NUMBER。属性值也支持表达式,您可以直接输入 state.xxx 来获取全局数据流中的数据。

代码模式具有以下特点:

- 在可视化视图中进行的拖拽、组件属性配置等操作, 会实时更新至代码。
- 代码中的修改会实时更新至可视化区域。
- 在可视化视图中进行的拖拽、组件属性配置等操作,与代码模式的修改可以互相转换。

# 4.9.5. DSL语法

DSL是一种以React JSX与Vue template的语言特性为基础,更符合UI编排的组件化语言。

#### JSX

Dat aWorks

DSL语法类似于React.render方法中的JSX部分,JSX的简单理解如下:

通过 {} ,将HTML作用域切换为JS作用域。JS作用域可以写任何合法的JS表达式,返回值会输出到页面
 上,例如 <div>{'Hello' + 'Relim'}</div> 。

⑦ 说明 { } 内可以写任何计算语句或字面量等JS表达式。

- 通过HT ML标签,将JS作用域切换为HT ML作用域,例如 {<div>Hello Relim</div>}。
- HTML和JS作用域切换可以嵌套进行,例如 {<div>{'Hello' + ' Relim'}</div>} 。

JSX的更多详情请参见React JSX。

#### 合法的JS表达式

```
//计算语句的情形
```

```
{aaa} // √ 变量aaa需要有定义
{aaa * 111} // √
{1 == 1 ? 1 : 0} // √
{/^123/.test(aa)} // √
{[1,2,3].join('')} // √
{(()=>{return 1})()} //自执行函数 √
//字面量
{1}
{true}
{[11,22,33]} // √
{[aa:"11",bb:"22"}} // √
{()=>1} //描述一个函数,合法,但无意义 √
```

⑦ 说明 如果遇到较为复杂的逻辑,一条计算语句不能实现,需拆分为多条语句的需求。可以将其包装为自执行函数,自执行函数是合法的表达式。示例如下:

```
{(function() {
    //将一个数字数组的偶数为求和。
    var input = [1,2,3,4,5,6,7,8,9,10];
    var temp = input.filter(i => i % 2 == 0)
    return temp.reduce((buf, cur) => buf + cur, 0)
})()}
```

#### 非法的JS表达式

```
{ var a = 1 } // 赋值语句。
{ aaa * 111; 2} // 出现分号的多条语句。
```

# 4.9.6. 全局数据流

全局数据流是前端数据管理的概念,多个组件为共享状态时,共享状态和组件间通信较为困难。此时将共享 状态抽取出来,用全局数据流的方式使之变得简单。

#### 全局数据流的原理

全局数据流使用了单一的数据流转方式,来实现全局数据的传递。在全局数据中声明的数据,只要变更后便 会执行如下图所示的数据流转。



- 1. 组件触发一个Action(例如通过鼠标单击触发)。
- 2. Action触发全局数据变更。
- 3. 全局数据变更会自动触发引用了该全局状态的相关组件的重新渲染。

#### 全局数据流的适用场景

全局数据流适用于页面中两个组件或者多个组件之间的组件联动,可以通过将公共数据提炼到全局数据中进 行统一管理,再利用全局数据流机制串联两个或多个组件。

#### 全局数据流的定义

1. 单击操作面板中的全局数据流配置图标。



2. 在全局数据流配置对话框中,填写变量名和变量值。

全局数据流配置		×
变量名	变量值	操作
变量名	变量值	删除
+ 添加		保存

- 。 变量值可以为数字、字符串或JSON串。
- 。 变量值声明为一个接口地址, 接口获取到的数据将会成为变量名对应的值。
- 3. 单击保存。

#### 使用全局数据流

• 获取全局数据

组件中通过 state.name 来获取全局数据。

```
<Input value={state.name} />
```

• 修改全局数据

组件中通过\$setState方法修改全局数据。

<Input onChange={value => \$setState({ name: value })} />

⑦ 说明 请务必使用\$setState方法修改全局数据,如果使用 state.name = 'new value',将会 无法触发重新渲染。

# 4.9.7. 导航配置

本文将为您介绍如何设置可视化搭建站点的导航。

App Studio可视化搭建为应用提供公共头部、底部和侧边栏,提供了丰富的菜单配置、主题配置。如果您不需要显示系统提供的公共头部和侧边栏,可以进行配置。

单击操作面板中的导航配置按钮,即可打开导航配置页面。



#### 配置公共头部

您可以根据自身需求对公共头部进行配置。 × 全局导航配置 头部 是否显示 主题 深色 Logo 图片 https://img.alicc zqK 上传 标题 AppStudio 是否固定于页面顶部 + 🗉 🌣 🗊 菜单项 首页 🖉 pages/home.santa 报表 🖉 pages/list.santa

配置	说明
是否显示	设置是否显示公共头部。
主题	您可以选择深色或浅色的主题样式。
Logo图片	显示的站点Logo图片,您可以输入一个图片地址,或者选择本地上传一张 图片。

配置	说明
标题	设置显示的站点标题。
是否固定于页面顶部	是否让公共头部一定固定于页面顶部(页面滚动时,公共头也将一直位于页 面顶部)。
菜单项	您可以定义公共头部可以显示的链接名称、链接地址等菜单项。

# 配置侧边栏

您可以根据自身需求对侧边栏进行配置。

(例边栏 是否显示 ● 1 主题 浅色 ■	
配置	说明
是否显示	设置是否显示侧边栏。
主题	您可以选择深色或浅色的主题样式。
是否可折叠	设置侧边栏菜单是否具有折叠功能。

# 4.9.8. 保存、预览、运行和热部署

可视化搭建系统支持保存、预览、操作和热部署等操作。

## 保存

可视化搭建系统会定时保存您的修改,您也可以单击操作面板中的保存图标,进行保存。



预览

在可视化搭建系统中,可视化操作区域处于编辑的状态。有部分组件针对编辑状态进行特殊处理,只有在正式的运行状态下才能执行正常的渲染逻辑。如果您想查看正常的渲染结果,可以单击操作区域的**预览**图标。



## 运行

可视化搭建系统单次只能打开一个可视化文件进行编辑。如果您想要以整个应用的视角进行查看,可以运行 整个应用来查看结果。

您可以单击App Studio Debug面板中的启动图标,来运行整个应用。



#### 热部署

应用启动后,如果您发现页面不符合预期,可以继续返回到可视化搭建系统进行调整。 调整完成后进行保存,您的修改将会支持热部署的方式生效至运行的页面。

# 4.9.9. 发布为模板

您可以将搭建好的前端页面发布为模板,后续基于该模板进行开发。

1. 打开可视化搭建文件,单击操作面板右上角的发布为模板图标。



2. 在模板发布对话框中,单击截图图标。

模板发布		×
用户访问来源	每周用户活跃量	<ul> <li>- 模板名称:</li> <li>▲</li> <li>● 模板类型:</li> </ul>
tianmao: 0.17 alipay: 0.21 taoba	80 70 60 50 40	<b>数据应用 × ∨</b> 评分: ★★★★★★ 模板描述:
● aliyun ● taobao ● alipay ● tianmao	30 20 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	等待录制
		→ 御込 取消

3. 待右侧的等待录制框中显示截图内容后,填写模板名称和模板类型,单击确认。

模板发布			×
用户访问来源	每周用户活跃量	● 模板名称: ▲ test	^
tianmao: 0.17 alipay: 0.21	80 70 60 50 40	数据应用 × V 评分: ★★★★★ 模板描述: test	
● aliyun ● taobao ● alipay ● tianmao	30 20 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		
		御入取	▼ 肖

4. 单击左侧菜单栏中的工程文件,打开santa目录。



5. 右键单击pages,选择新建>模板文件。



6. 选择发布为模板的文件,单击选择模板。

全部模板	数据应用
数据报表	Andela beanses
数据应用	and an
数据大屏 其他	
	Deshboerd添加订单 test

7. 在添加文件对话框中填写文件名称,单击创建,即可新建一个页面,并基于模板进行开发。

