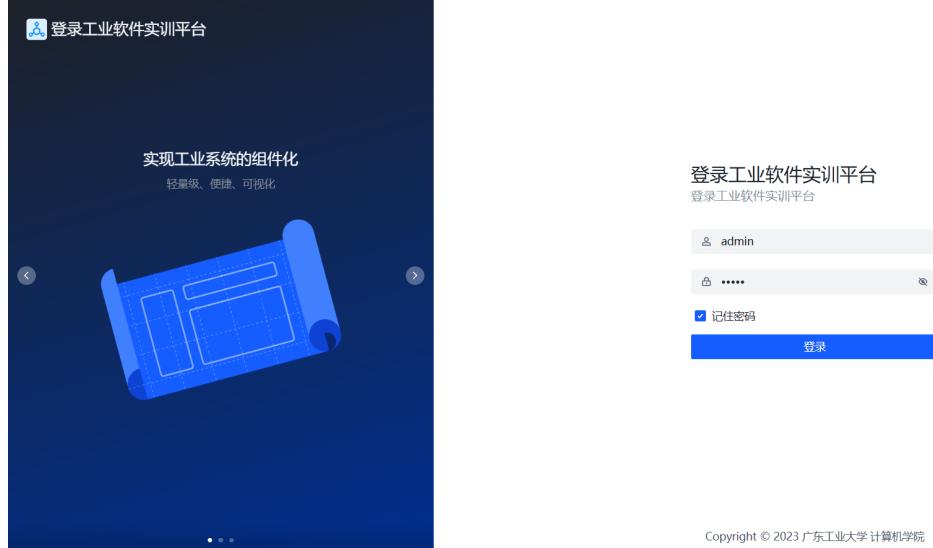


操作说明

登录

在登录页输入账号密码，登录到系统中。

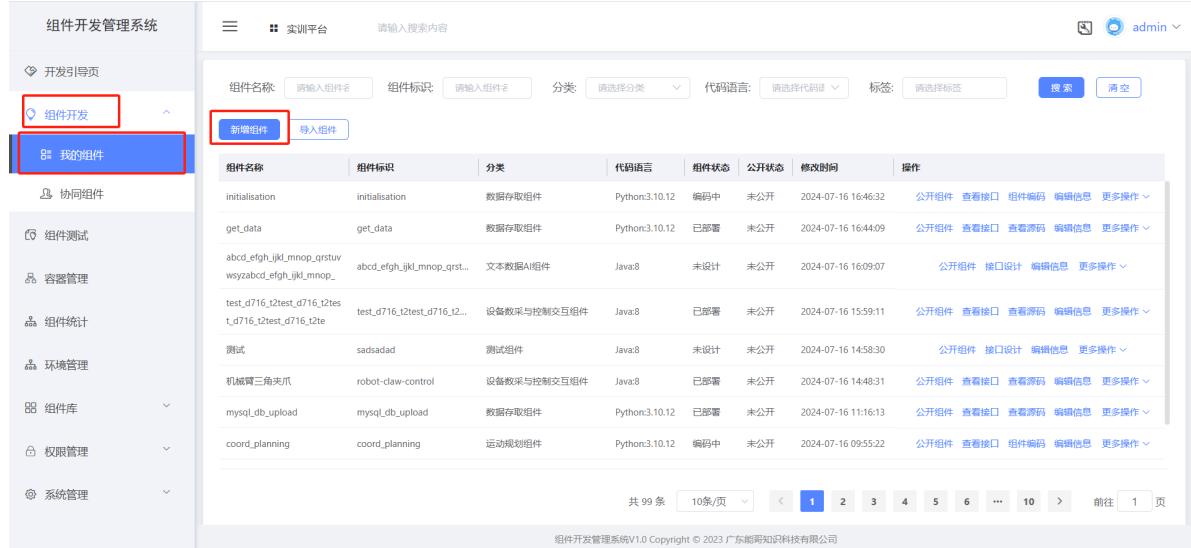


进入组件开发管理平台

组件开发

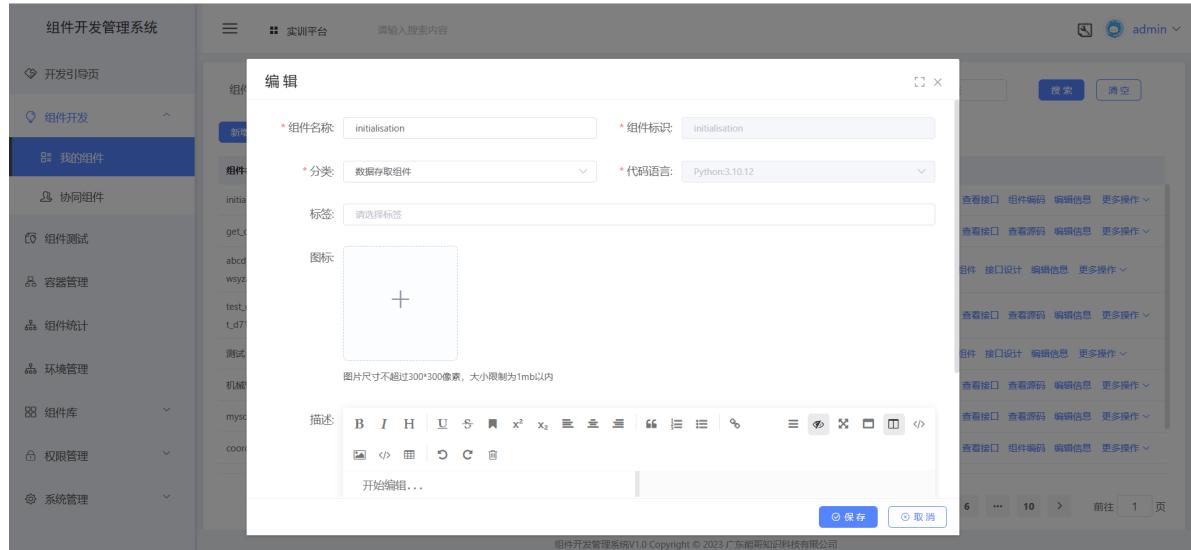
组件的创建

点击新增



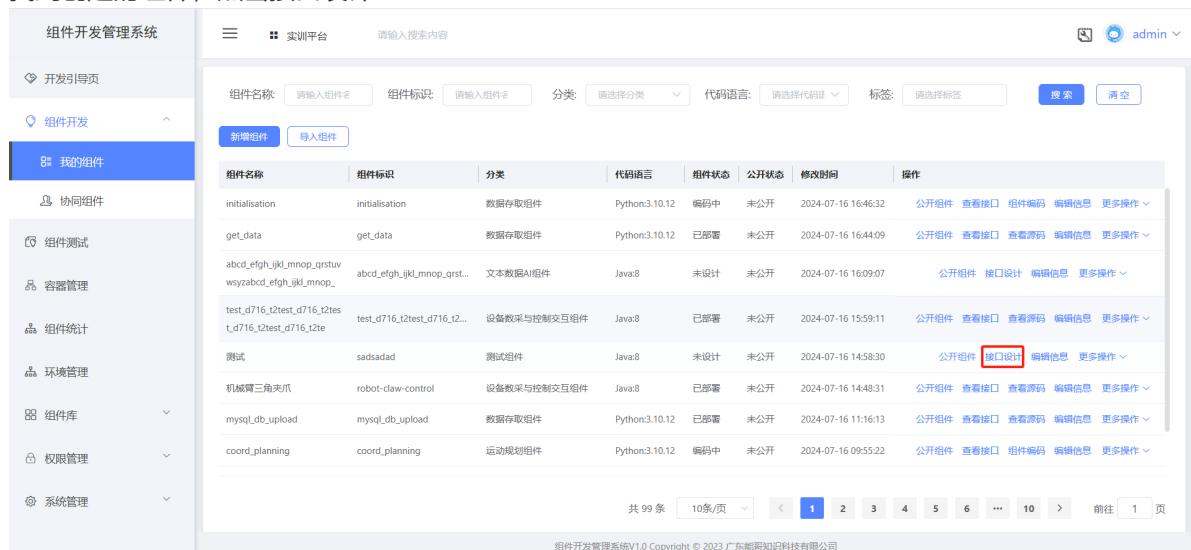
The screenshot shows the 'Component Development Management System' interface. On the left, a sidebar lists various management categories: 'Development Guide Page', 'Component Development' (highlighted with a red box), 'Collaborative Components', 'Component Testing', 'Container Management', 'Component Statistics', 'Environment Management', 'Component Library' (highlighted with a red box), 'Permission Management', and 'System Management'. The main content area is titled 'Training Platform' and contains a search bar. Below the search bar is a component list table with columns: 'Component Name', 'Component Identifier', 'Category', 'Code Language', 'Status', 'Open Status', 'Modification Time', and 'Operations'. A red box highlights the 'New Component' button in the top-left corner of the component list area. The footer of the page includes the text 'Component Development Management System V1.0 Copyright © 2023 Guangdong Nanyang Knowledge Technology Co., Ltd'.

填写组件基本信息，点击保存



The screenshot shows the 'Edit' form for a component. The sidebar on the left is identical to the previous screenshot. The main form has a title 'Edit' and contains the following fields: 'Component Name' (initialisation), 'Component Identifier' (initialisation), 'Category' (Data Storage Component), 'Code Language' (Python:3.10.12), 'Tags' (empty), and an 'Icon' section with a placeholder image and a note about file size. Below these are 'Description' and 'Content' sections with rich text editors. At the bottom right are 'Save' and 'Cancel' buttons. A red box highlights the 'Save' button. The footer of the page includes the text 'Component Development Management System V1.0 Copyright © 2023 Guangdong Nanyang Knowledge Technology Co., Ltd'.

找到创建的组件，点击接口设计

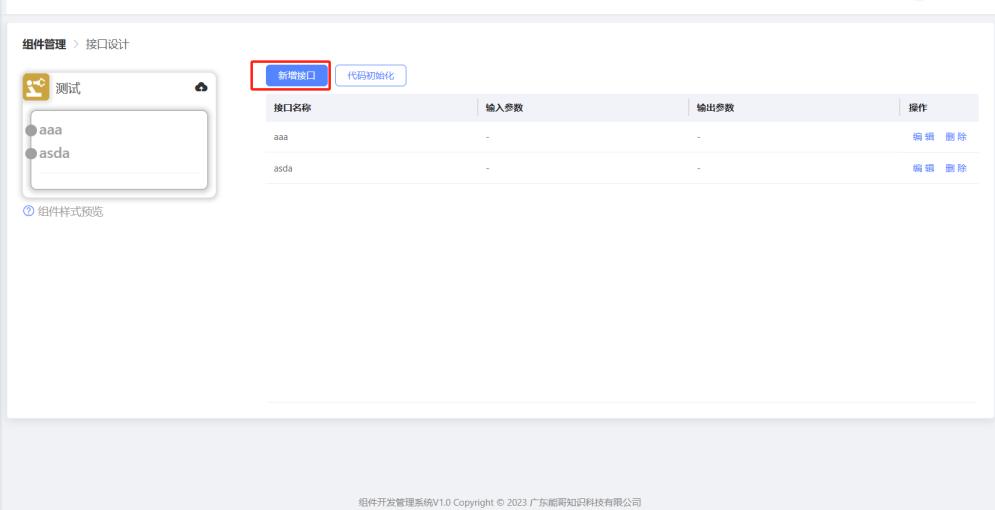


The screenshot shows the component list again. The sidebar and main interface are identical to the previous screenshots. A red box highlights the 'Interface Design' button in the 'Operations' column for the first component in the list. The footer of the page includes the text 'Component Development Management System V1.0 Copyright © 2023 Guangdong Nanyang Knowledge Technology Co., Ltd'.

组件开发

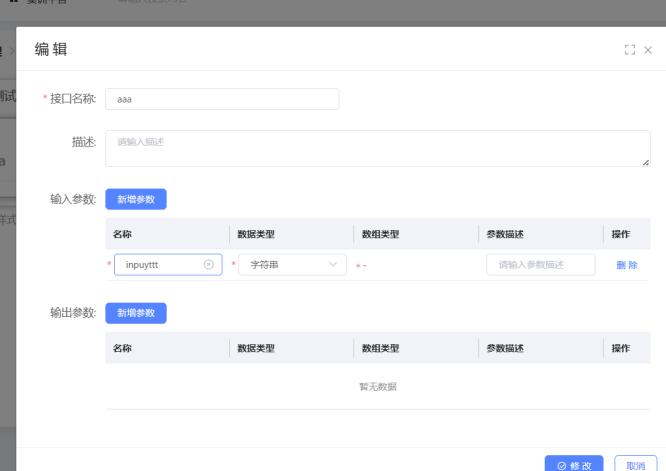
样式设计

点击进行新增操作



接口名称	输入参数	输出参数	操作
aaa	-	-	编辑 删除
asda	-	-	编辑 删除

依次新增操作，并每个操作配置参数



名称	数据类型	数组类型	参数描述	操作
inputtt	字符串	-	请输入参数描述	删除

完成后点击初始化，生成代码，注意组件外观在代码初始化之后不可再次修改！

The screenshot shows the 'Component Management' system interface. On the left, a sidebar lists various management categories: Development Guide, Component Development, My Components, Collaborative Components, Component Testing, Container Management, Component Statistics, Environment Management, Component Library, Permission Management, and System Management. The 'Component Development' section is expanded, showing 'My Components' with items 'aaa' and 'asda', and a 'Component Style Preview' section.

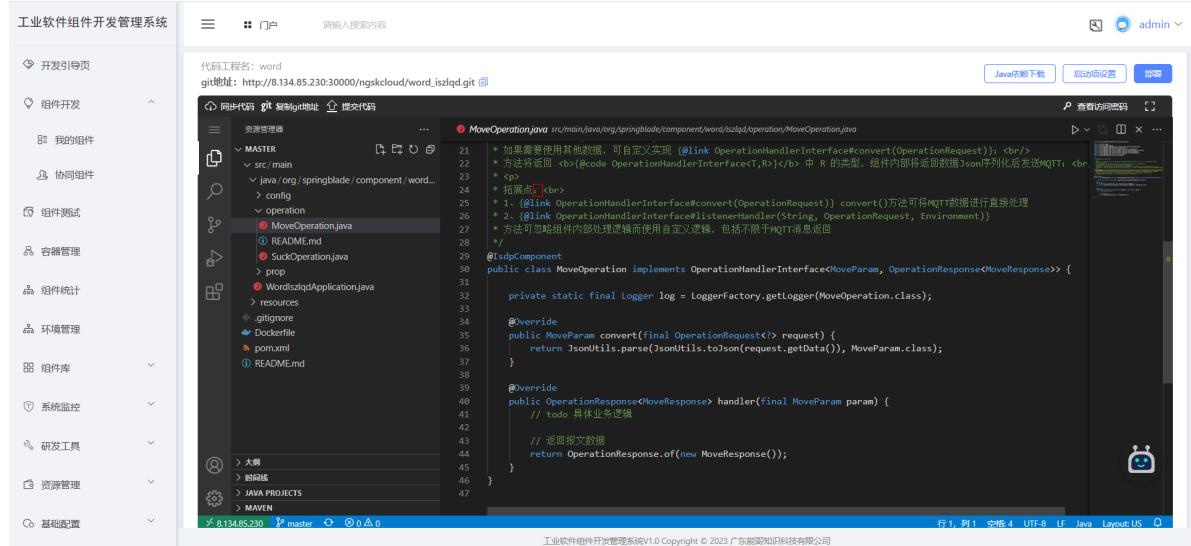
The main content area is titled 'Component Management > Interface Design'. It displays a table of interfaces with columns for 'Interface Name', 'Input Parameters', 'Output Parameters', and 'Operations'. Two rows are shown: 'aaa' and 'asda'. The 'Operations' column for 'aaa' includes 'Edit' and 'Delete' buttons. The 'Operations' column for 'asda' includes 'Edit' and 'Delete' buttons.

A red box highlights the 'Code Initialization' button in the top right corner of the interface design area. Below this, a modal dialog titled 'Code Initialization Confirmation' is displayed. It contains the following text: 'Initialization completed, you can now modify the logic code, but you cannot modify the component interface. If you need to modify the component interface, you must reinitialize it, and the new initialization will not retain the original logic code.' It also contains the text 'Confirm to initialize the interface design of component 'aaa''. At the bottom of the modal are two buttons: 'Start Initialization' (highlighted with a red box) and 'Cancel'.

At the bottom of the interface, a copyright notice reads: 'Component Management System V1.0 Copyright © 2023 Guangdong Nanyang Technology Co., Ltd.'

逻辑编码

进入逻辑编码，开始进行代码也可以复制git地址将代码拉取到本地开发再进行提交



工业软件组件开发管理系统

代码工程名: word
git地址: http://8.134.85.230:30000/ngskcloud/word_iszldq.git

同步代码 git 复制git地址 提交代码

```
MoveOperation.java src/main/java/org/springblade/component/word/iszldq/operation/MoveOperation.java
21 * 如果需要使用其他数据，可自定义实现 {@link OperationHandlerInterface#convert(OperationRequest)}; <br/>
22 * 方法将返回 <br>{@code OperationHandlerInterface<R>} 中 R 的类型，组件内部将返回数据.json序列化后发送MQTT; <br/>
23 * 拓展点<br>
24 * 1. {@link OperationHandlerInterface#convert(OperationRequest)} convert()方法可将MQTT数据进行直接处理
25 * 2. {@link OperationHandlerInterface#addListenerHandler(String, OperationRequest, Environment)}
26 * 方法可忽略组件内部处理逻辑而使用自定义逻辑，包括不限于MQTT消息返回
27 */
28
29 @IszldqComponent
30 public class MoveOperation implements OperationHandlerInterface<MoveParam, OperationResponse<MoveResponse>> {
31
32     private static final Logger log = LoggerFactory.getLogger(MoveOperation.class);
33
34     @Override
35     public MoveParam convert(Final OperationRequest<> request) {
36         return JsonUtils.fromJson(request.getData(), MoveParam.class);
37     }
38
39     @Override
40     public OperationResponse<MoveResponse> handler(Final MoveParam param) {
41         // todo 具体业务逻辑
42
43         // 返回报文数据
44         return OperationResponse.of(new MoveResponse());
45     }
46
47 }
```

MASTER

src / main

java / org / springblade / component / word ...

config

operation

MoveOperation.java

README.md

SuckOperation.java

prop

WordsqlApplication.java

resources

.gitignore

Dockerfile

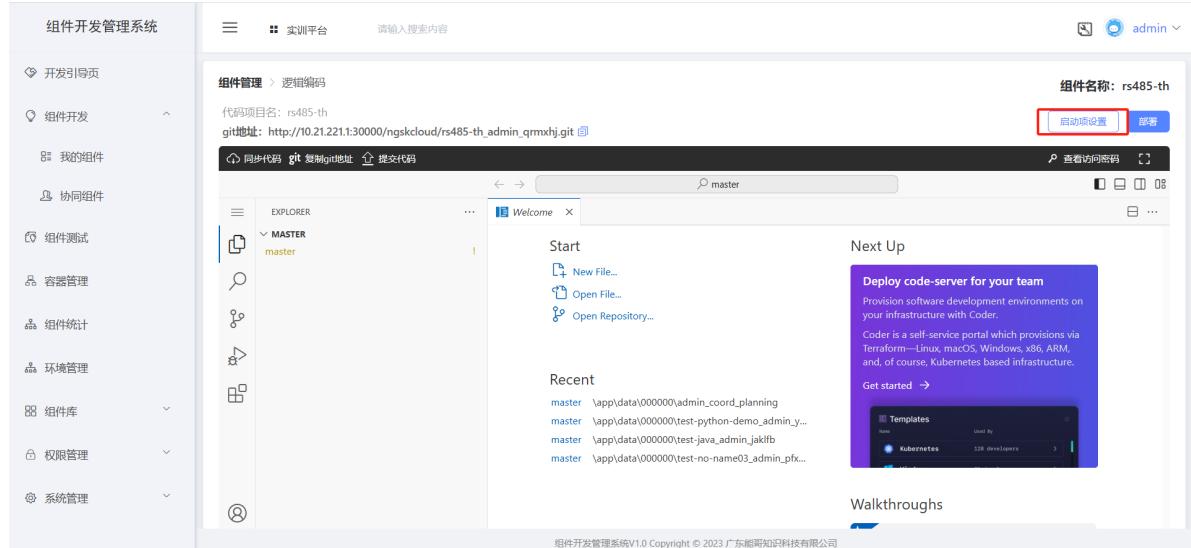
pom.xml

READMEmd

8.134.85.230 master 0.0.0

工业软件组件开发管理系统V1.0 Copyright © 2023 广东能瑞知识科技有限公司

也可以在右上角为项目配置默认启动项



组件开发管理系统

实训平台

请输入搜索内容

组件管理 > 逻辑编码

代码项目名: rs485-th
git地址: http://10.21.221.1:30000/ngskcloud/rs485-th_admin_qrmxh.git

同步代码 git 复制git地址 提交代码

master

组件名称: rs485-th

启动项设置

查看访问密钥

Welcome

Start

New File...

Open File...

Open Repository...

Recent

master \app\data\000000\admin_coord_planning

master \app\data\000000\test-python-demo_admin_y...

master \app\data\000000\test-java_admin_jakib

master \app\data\000000\test-no-name03_admin_pfx...

Next Up

Deploy code-server for your team

Provision software development environments on your infrastructure with Coder.

Coder is a self-service portal which provisions via Terraform—Linux, macOS, Windows, x86, ARM, and, of course, Kubernetes based infrastructure.

Get started →

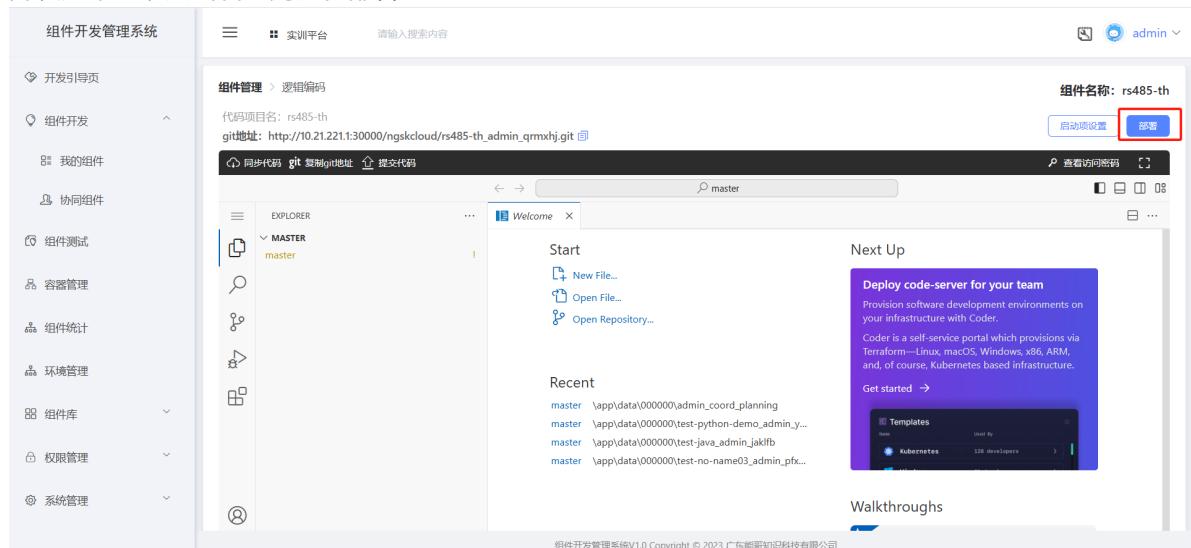
Templates

Kubernetes

Walkthroughs

组件开发管理系统V1.0 Copyright © 2023 广东能瑞知识科技有限公司

开发完毕可以点击右上角进行部署



组件开发管理系统

实训平台

请输入搜索内容

组件管理 > 逻辑编码

代码项目名: rs485-th
git地址: http://10.21.221.1:30000/ngskcloud/rs485-th_admin_qrmxh.git

同步代码 git 复制git地址 提交代码

master

组件名称: rs485-th

启动项设置

部署

查看访问密钥

Welcome

Start

New File...

Open File...

Open Repository...

Recent

master \app\data\000000\admin_coord_planning

master \app\data\000000\test-python-demo_admin_y...

master \app\data\000000\test-java_admin_jakib

master \app\data\000000\test-no-name03_admin_pfx...

Next Up

Deploy code-server for your team

Provision software development environments on your infrastructure with Coder.

Coder is a self-service portal which provisions via Terraform—Linux, macOS, Windows, x86, ARM, and, of course, Kubernetes based infrastructure.

Get started →

Templates

Kubernetes

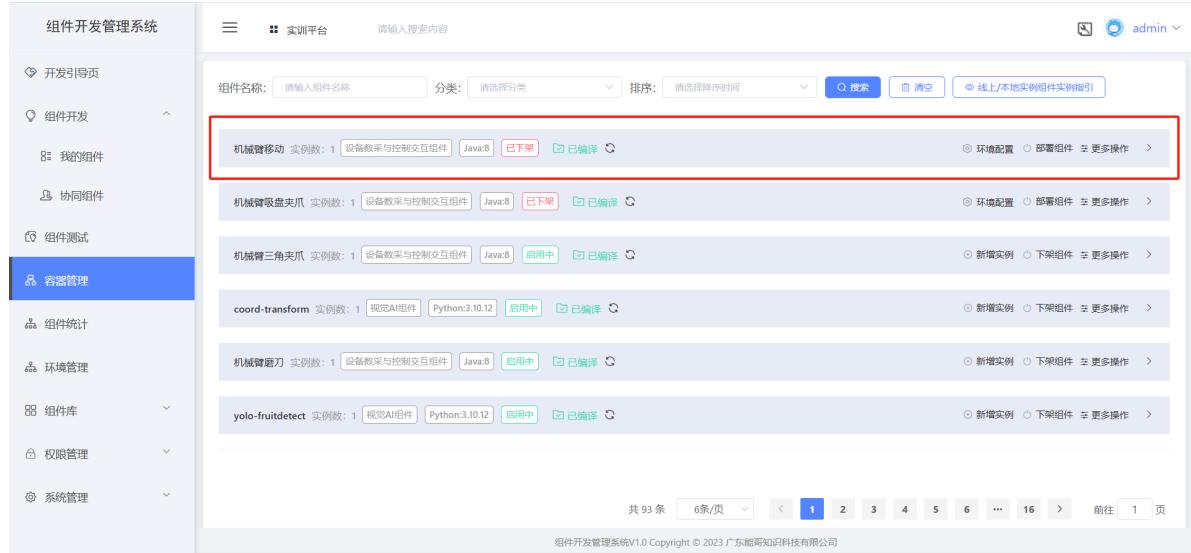
Walkthroughs

组件开发管理系统V1.0 Copyright © 2023 广东能瑞知识科技有限公司

部署管理

部署组件列表

在部署列表找到该组件，点击新增实例



组件名称: 分类: 排序: 搜索 清空 线上/本地实例组件实例指引

机械臂移动 实例数: 1 [设备数据采集与控制交互组件] Java:8 已下架 已编译 >

机械臂吸盘夹爪 实例数: 1 [设备数据采集与控制交互组件] Java:8 已下架 已编译 >

机械臂三角夹爪 实例数: 1 [设备数据采集与控制交互组件] Java:8 启用中 已编译 >

coord-transform 实例数: 1 [视觉AI组件] Python:3.10.12 启用中 已编译 >

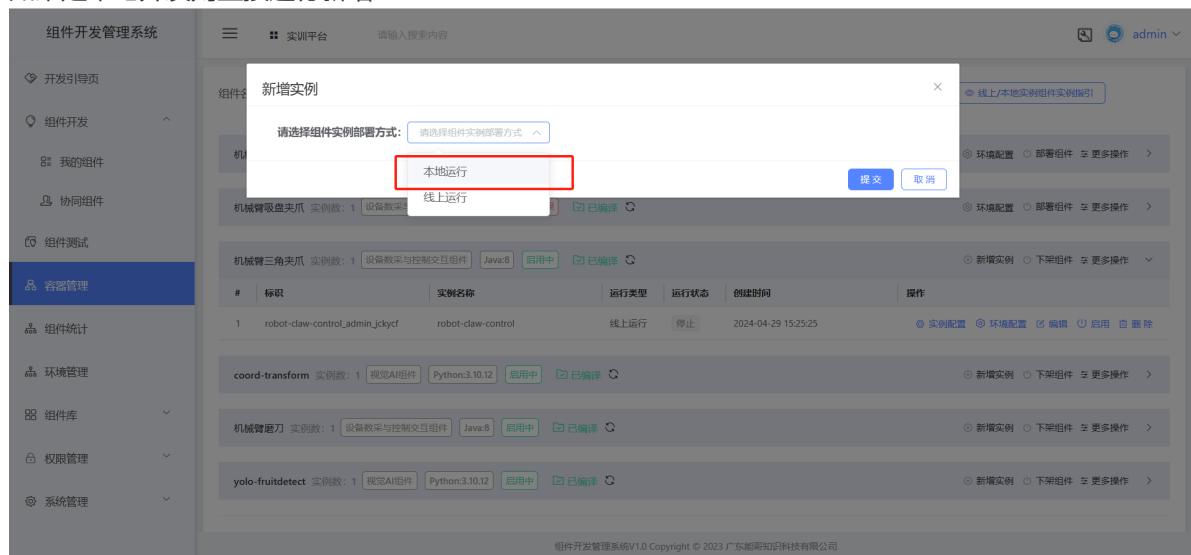
机械臂磨刀 实例数: 1 [设备数据采集与控制交互组件] Java:8 启用中 已编译 >

yolo-fruitdetect 实例数: 1 [视觉AI组件] Python:3.10.12 启用中 已编译 >

共 93 条 6条/页 1 2 3 4 5 6 ... 16 > 前往 1 页

本地部署

如果是本地开发则直接进行新增



新增实例

请选择组件实例部署方式: 本地运行 线上运行

提交 取消

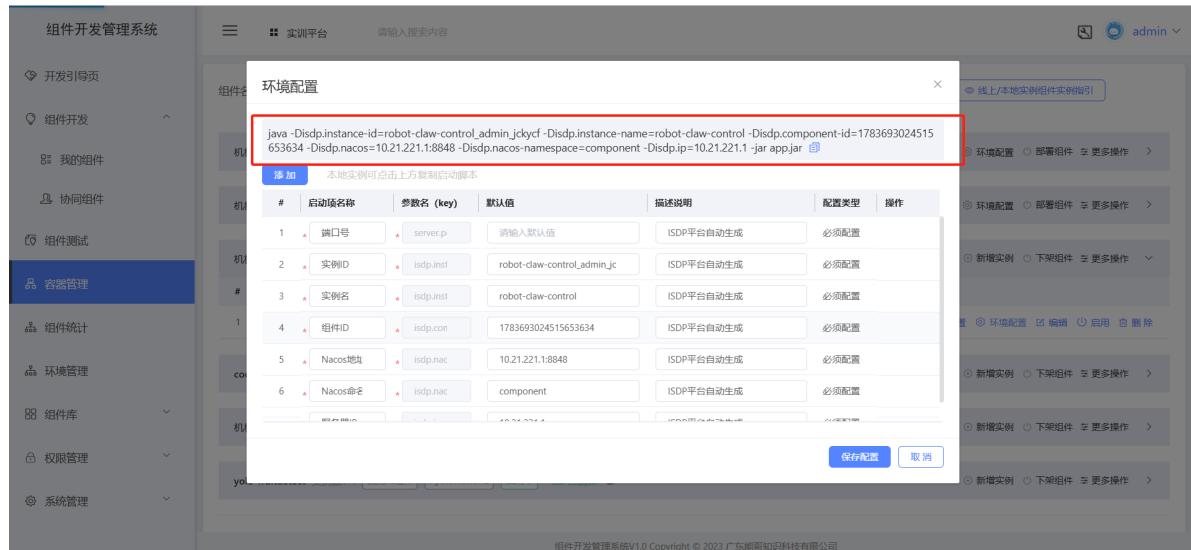
机械臂吸盘夹爪 实例数: 1 [设备数据采集与控制交互组件] Java:8 启用中 已编译 >

coord-transform 实例数: 1 [视觉AI组件] Python:3.10.12 启用中 已编译 >

机械臂磨刀 实例数: 1 [设备数据采集与控制交互组件] Java:8 启用中 已编译 >

yolo-fruitdetect 实例数: 1 [视觉AI组件] Python:3.10.12 启用中 已编译 >

点击环境配置，可以复制本地启动脚本进行本地启动



环境配置

java -Disdp.instance-id=robot-claw-control_admin_jckycf -Disdp.instance-name=robot-claw-control -Disdp.component-id=178369302451563634 -Disdp.nacos=10.21.221.1:8848 -Disdp.nacos-namespace=component -Disdp.ip=10.21.221.1-jar app.jar

#	启动项名称	参数名 (key)	默认值	描述说明	配置类型	操作
1	端口号	server.port	请输入默认值	ISDP平台自动生成	必须配置	
2	实例ID	isdp.inst	robot-claw-control_admin_jc	ISDP平台自动生成	必须配置	
3	实例名	isdp.inst	robot-claw-control	ISDP平台自动生成	必须配置	
4	组件ID	isdp.con	178369302451563634	ISDP平台自动生成	必须配置	
5	Nacos地址	isdp.nac	10.21.221.1:8848	ISDP平台自动生成	必须配置	
6	Nacos命名	isdp.nac	component	ISDP平台自动生成	必须配置	

保存配置 取消

线上部署

选择线上部署填写资源，进行新增



组件开发管理系统

实训平台

请输入搜索内容

admin

新增实例

请选择组件实例部署方式: 线上运行

* 部署环境: 生产 * 主机: 请选择主机

CPU核数: 1核 4核 是否使用GPU: 否 是

最大内存: 64MB 1GB 2GB 4GB 2048 MB

* 网络模式: 桥接模式 * 网卡: bridge IP: 属性子网: -

端口映射: 主机端口 容器端口 备注

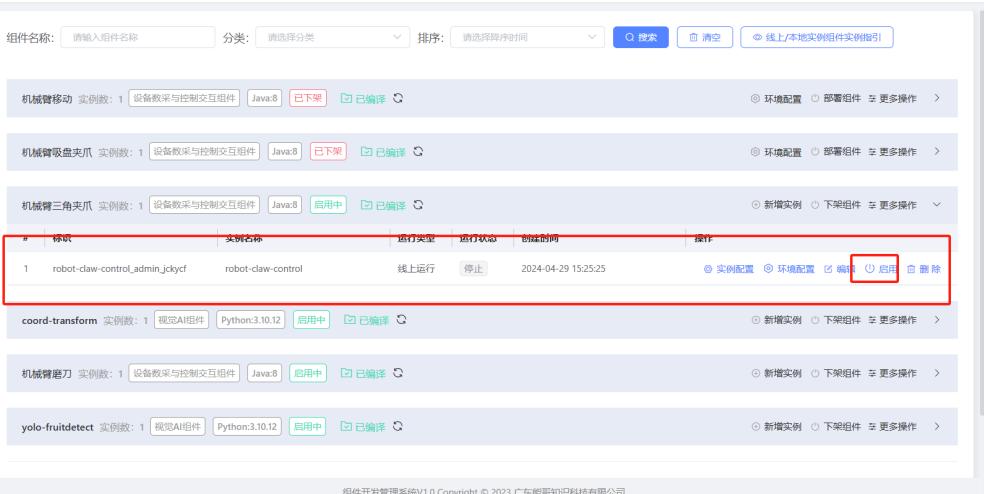
目录挂载: 主机路径 容器路径 备注

环境配置 部署组件 更多操作 >

新增实例 下架组件 更多操作 >

实例列表

点击进行启用



组件开发管理系统

实训平台

请输入搜索内容

admin

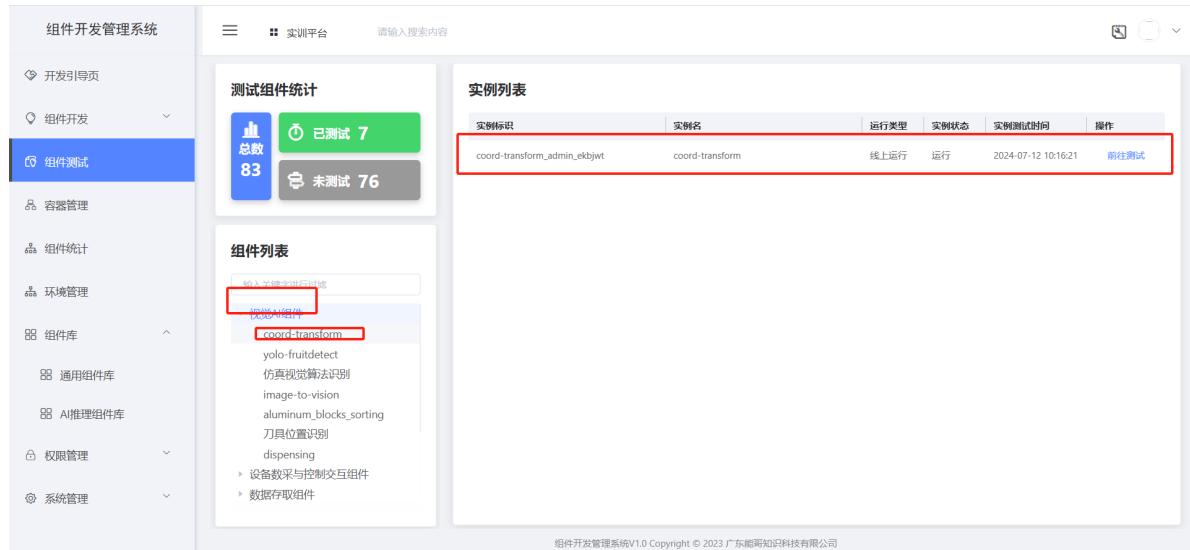
组件名称: 搜索框 分类: 请选择分类 排序: 请选择排序时间 搜索 清空 线上/本地实例组件实例指引

编号	实例名称	运行类型	运行状态	创建时间	操作
1	robot-claw-control_admin_jckyf	robot-claw-control	线上运行	停止	实例配置 环境配置 编辑 启用 白删除
2	coord-transform	视觉AI组件	Python:3.10.12	启用中	已编译
3	机械臂夹爪	设备数据采集与控制交互组件	Java:8	已下架	已编译
4	机械臂三角夹爪	设备数据采集与控制交互组件	Java:8	已下架	已编译
5	机械臂磨刀	设备数据采集与控制交互组件	Java:8	启用中	已编译
6	yolo-fruitedetect	视觉AI组件	Python:3.10.12	启用中	已编译

组件开发管理系统V1.0 Copyright © 2023 广东能斯知识科技有限公司

组件测试

从左边搜索对应的组件，选择对应的实例进行测试，实例需要是启用状态



组件开发管理系统

实训平台

请输入搜索内容

测试组件统计

已测试 7

总数 83

未测试 76

实例列表

实例标识	实例名	运行类型	实例状态	实例测试时间	操作
coord-transform_admin_ekbjwt	coord-transform	线上运行	运行	2024-07-12 10:16:21	前往测试

组件列表

组件搜索框

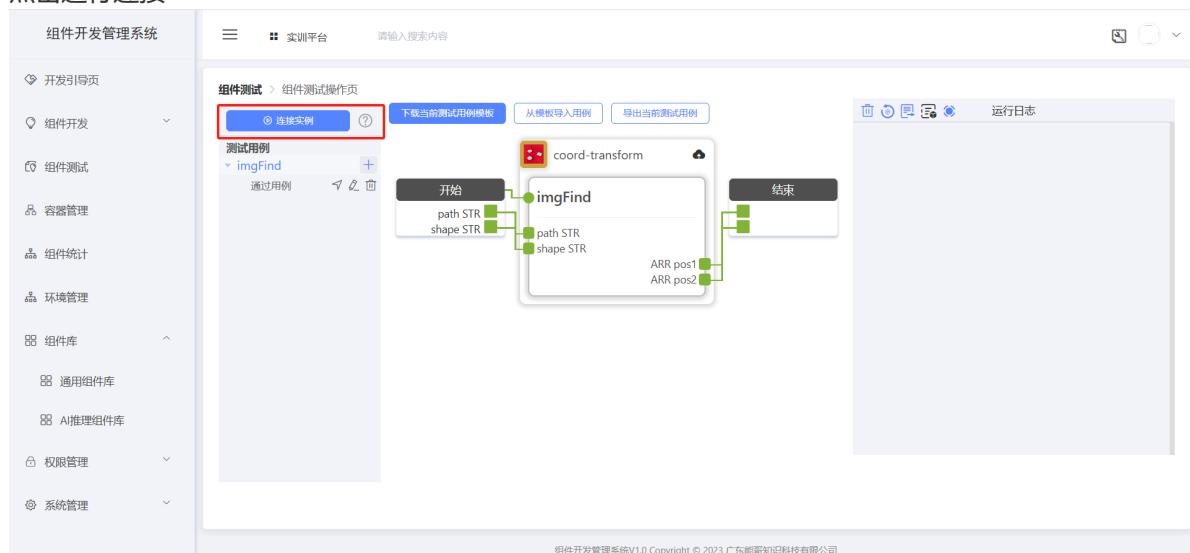
组件名称

coord-transform

- yolo-fruitdetect
- 仿真视觉算法识别
- image-to-vision
- aluminum_blocks_sorting
- 刀具位置识别
- dispensing
- 设备数据采集与控制交互组件
- 数据存取组件

组件开发管理系统V1.0 Copyright © 2023 广东能斯知识科技有限公司

点击进行连接



组件开发管理系统

实训平台

请输入搜索内容

组件测试 > 组件测试操作页

连接实例

下载当前测试用例模板

从模板导入用例

导出当前测试用例

运行日志

测试用例

通过用例

imgFind

开始

coord-transform

结束

path STR

shape STR

path STR

shape STR

ARR pos1

ARR pos2

组件开发管理系统V1.0 Copyright © 2023 广东能斯知识科技有限公司

连接成功后可对组件的单个接口进行测试



组件开发管理系统

实训平台

请输入搜索内容

组件测试 > 组件测试操作页

断开连接

下载当前的测试用例模板

从模板导入用例

导出当前测试用例

运行日志

连接信息: 连接成功!
未启动任务或任务已完成

连接信息: 连接成功!
连接信息: 连接成功!

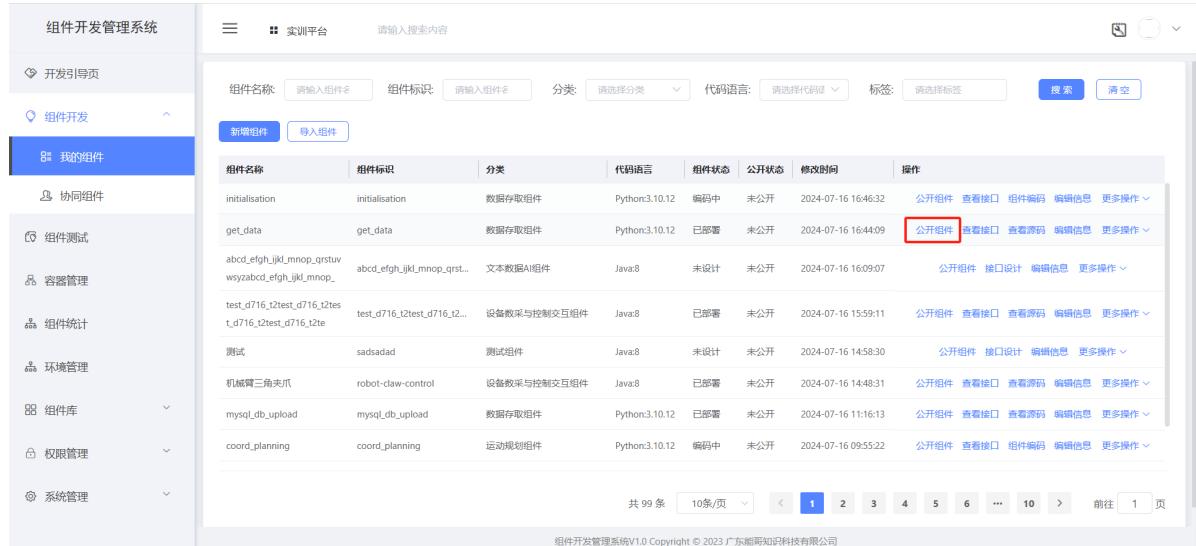
发送数据:

```
{ "testCaseName": "通过用例", "operation": "imgFind", "expectDataIns": { "path": { }, "shape": { } }, "expectDataouts": { "pos1": { }, "pos2": { } } }
```

组件开发管理系统V1.0 Copyright © 2023 广东能斯知识科技有限公司

组件发布

直接进行公开



组件开发管理系统

实训平台

请输入搜索内容

组件名称: 组件标识: 分类: 代码语言: 标签: 搜索

我的组件

协同组件

组件测试

容器管理

组件统计

环境管理

组件库

权限管理

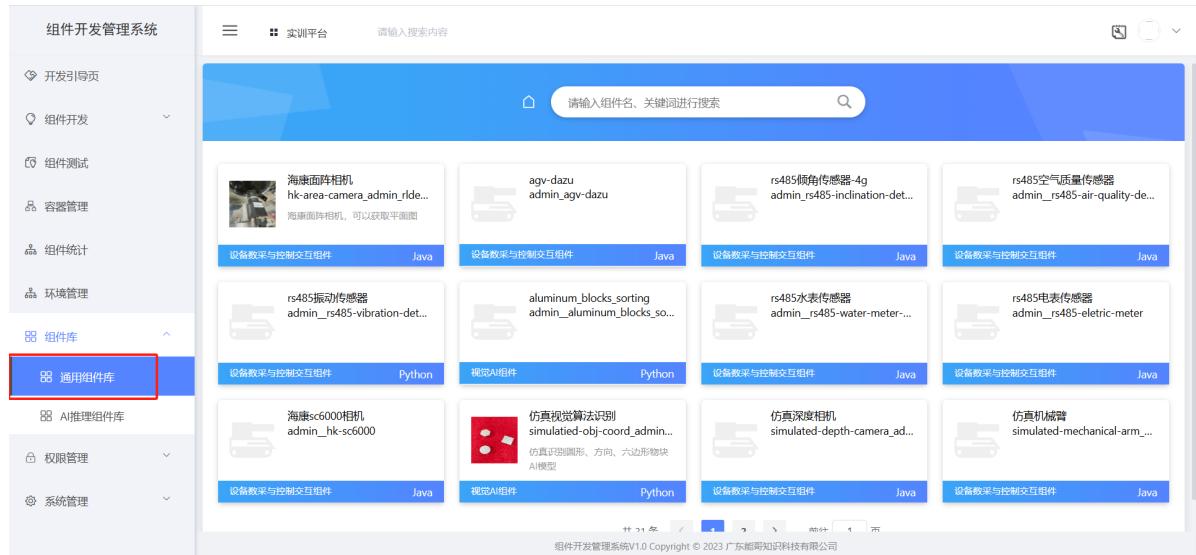
系统管理

组件名称	组件标识	分类	代码语言	组件状态	公开状态	修改时间	操作
initialisation	initialisation	数据存取组件	Python:3.10.12	编码中	未公开	2024-07-16 16:46:32	公开组件 查看接口 组件编码 编辑信息 更多操作
get_data	get_data	数据存取组件	Python:3.10.12	已部署	未公开	2024-07-16 16:44:09	公开组件 查看接口 查看源码 编辑信息 更多操作
abcd_efgh_ijkl_mnop_qrstw wsyzabcd_efgh_ijkl_mnop_	abcd_efgh_ijkl_mnop_qrstw wsyzabcd_efgh_ijkl_mnop_	文本数据AI组件	Java:8	未设计	未公开	2024-07-16 16:09:07	公开组件 接口设计 编辑信息 更多操作
test_d716_12test_d716_12tes t_d716_12test_d716_12te	test_d716_12test_d716_12tes t_d716_12test_d716_12te	设备数据与控制交互组件	Java:8	已部署	未公开	2024-07-16 15:59:11	公开组件 查看接口 查看源码 编辑信息 更多操作
测试	sadsadad	测试组件	Java:8	未设计	未公开	2024-07-16 14:58:30	公开组件 接口设计 编辑信息 更多操作
机械臂三角夹爪	robot-claw-control	设备数据与控制交互组件	Java:8	已部署	未公开	2024-07-16 14:48:31	公开组件 查看接口 查看源码 编辑信息 更多操作
mysql_db_upload	mysql_db_upload	数据存取组件	Python:3.10.12	已部署	未公开	2024-07-16 11:16:13	公开组件 查看接口 查看源码 编辑信息 更多操作
coord_planning	coord_planning	运动规划组件	Python:3.10.12	编码中	未公开	2024-07-16 09:55:22	公开组件 查看接口 组件编码 编辑信息 更多操作

共 99 条 10条/页 < 1 2 3 4 5 6 ... 10 > 前往 1 页

组件开发管理系统V1.0 Copyright © 2023 广东能斯知识科技有限公司

在组件库即可查看



组件开发管理系统

实训平台

请输入搜索内容

开发引导页

组件开发

组件测试

容器管理

组件统计

环境管理

组件库

通用组件库

AI推理组件库

权限管理

系统管理

请输入组件名、关键词进行搜索

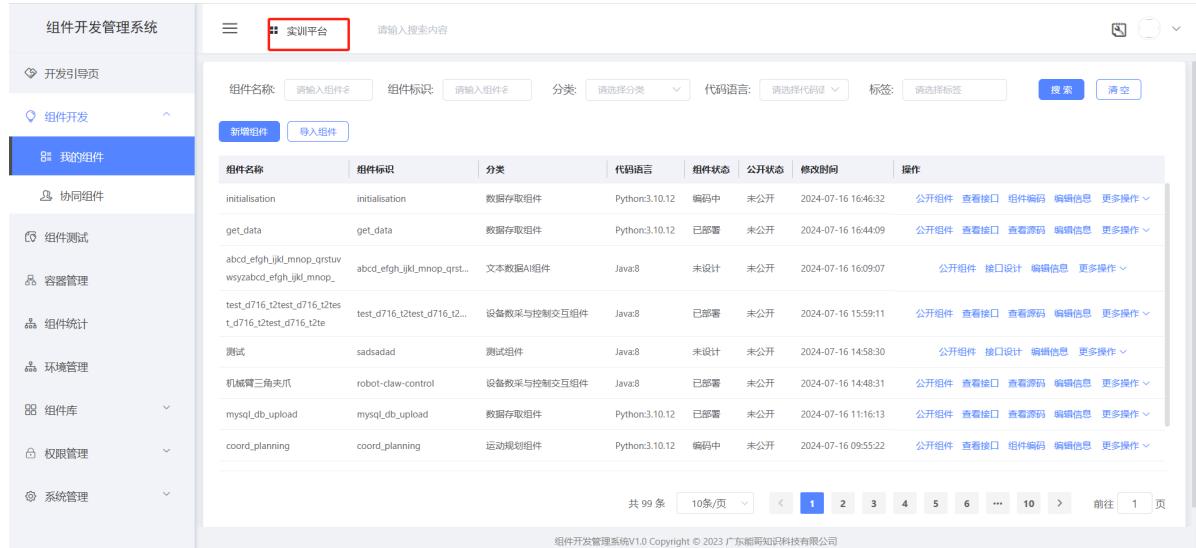
海康面阵相机 hk-area-camera_admin_rlde...	agv-dazu admin_agv-dazu	rs485倾角传感器-4g admin_rs485-inclination-det...	rs485空气质量传感器 admin_rs485-air-quality-de...
设备数据与控制交互组件 Java	设备数据与控制交互组件 Java	设备数据与控制交互组件 Java	设备数据与控制交互组件 Java
rs485振动传感器 admin_rs485-vibration-det...	aluminum_blocks_sorting admin_aluminum_blocks_so...	rs485水表传感器 admin_rs485-water-meter-...	rs485电表传感器 admin_rs485-electric-meter
设备数据与控制交互组件 Python	视觉AI组件 Python	设备数据与控制交互组件 Java	设备数据与控制交互组件 Java
海康sc6000相机 admin_hk-sc6000	仿真视觉算法识别 simulated-obj-coord_admin...	仿真深度相机 simulated-depth-camera_ad...	仿真机械臂 simulated-mechanical-arm_...
设备数据与控制交互组件 Java	视觉AI组件 Python	设备数据与控制交互组件 Java	设备数据与控制交互组件 Java

共 21 条 < 1 2 3 4 5 6 ... 11 > 前往 1 页

组件开发管理系统V1.0 Copyright © 2023 广东能斯知识科技有限公司

组件编排

切换到实训平台



组件开发管理系统

实训平台

请输入搜索内容

组件名称: 组件标识: 分类: 代码语言: 标签:

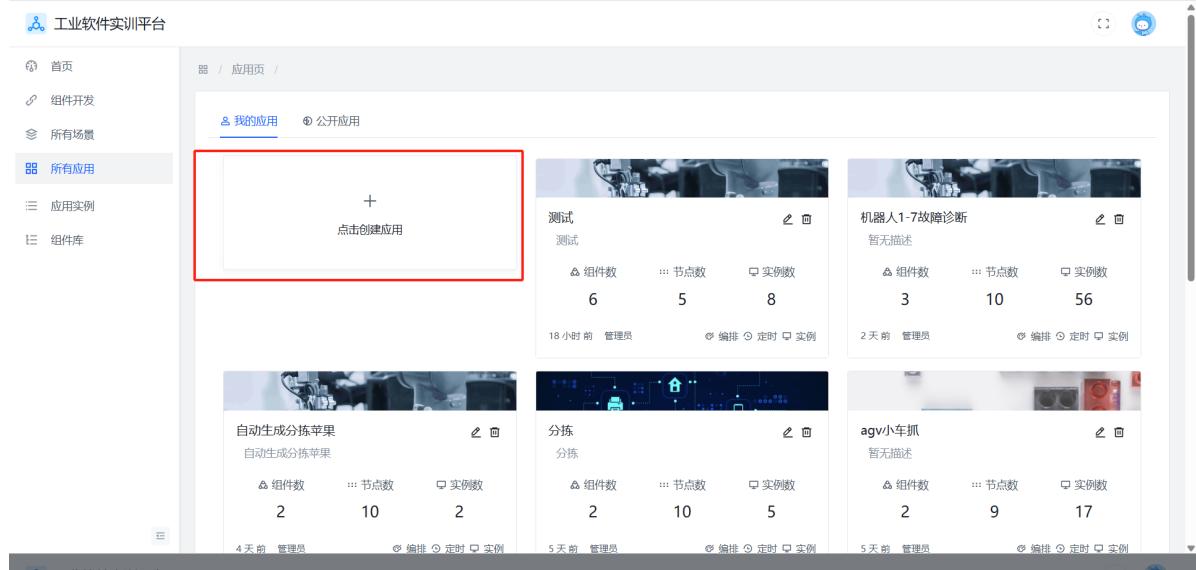
新增组件 导入组件

组件名称	组件标识	分类	代码语言	组件状态	公开状态	修改时间	操作
initialisation	initialisation	数据存取组件	Python:3.10.12	编码中	未公开	2024-07-16 16:46:32	公开组件 查看接口 组件编码 编辑信息 更多操作
get_data	get_data	数据存取组件	Python:3.10.12	已部署	未公开	2024-07-16 16:44:09	公开组件 查看接口 查看源码 编辑信息 更多操作
abcd_efgh_ijkl_mnop_qrstw	abcd_efgh_ijkl_mnop_qrstw	文本数据AI组件	Java:8	未设计	未公开	2024-07-16 16:09:07	公开组件 接口设计 编辑信息 更多操作
test_d716_12test_d716_12test	test_d716_12test_d716_12test	设备数据采集与控制交互组件	Java:8	已部署	未公开	2024-07-16 15:59:11	公开组件 查看接口 查看源码 编辑信息 更多操作
测试	sadsadad	测试组件	Java:8	未设计	未公开	2024-07-16 14:58:30	公开组件 接口设计 编辑信息 更多操作
机械臂三角夹爪	robot-claw-control	设备数据采集与控制交互组件	Java:8	已部署	未公开	2024-07-16 14:48:31	公开组件 查看接口 查看源码 编辑信息 更多操作
mysql_db_upload	mysql_db_upload	数据存取组件	Python:3.10.12	已部署	未公开	2024-07-16 11:16:13	公开组件 查看接口 查看源码 编辑信息 更多操作
coord_planning	coord_planning	运动规划组件	Python:3.10.12	编码中	未公开	2024-07-16 09:55:22	公开组件 查看接口 组件编码 编辑信息 更多操作

共 99 条 10条/页 < 1 2 3 4 5 6 ... 10 > 前往 1 页

组件开发管理系统V1.0 Copyright © 2023 广东能斯知识科技有限公司

新建应用

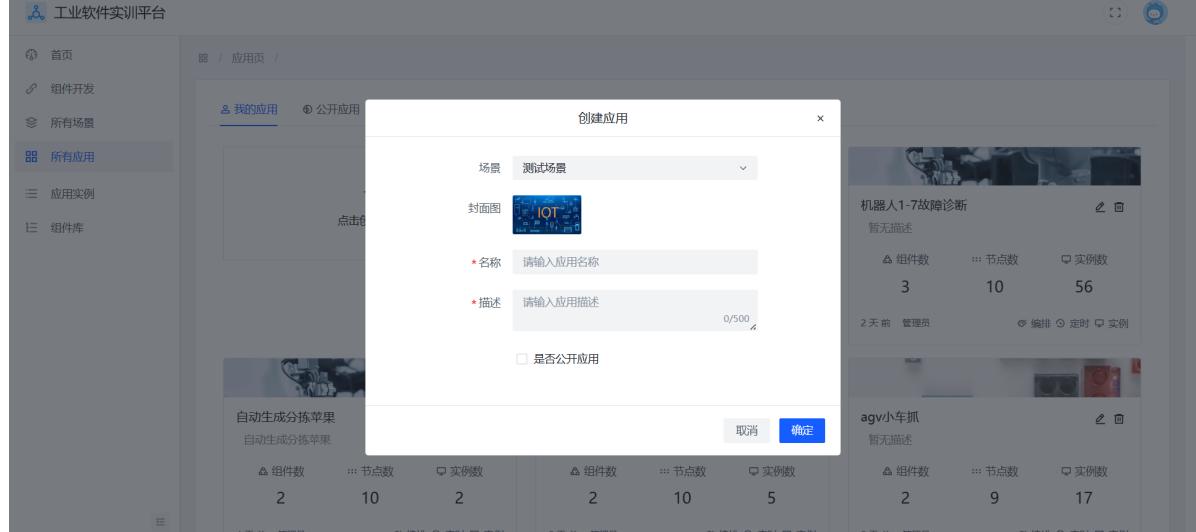


工业软件实训平台

我的应用 公开应用

点击创建应用

应用名称	组件数	节点数	实例数
测试	6	5	8
机器人1-7故障诊断	3	10	56
自动生分拣苹果	2	10	2
分拣	2	10	5
agv小车抓	2	9	17



工业软件实训平台

我的应用 公开应用

创建应用

场景: 测试场景

封面图: IOT

*名称: 请输入应用名称

*描述: 请输入应用描述 0/500

是否公开应用

取消 确定

应用名称	组件数	节点数	实例数
测试	6	5	8
机器人1-7故障诊断	3	10	56
自动生分拣苹果	2	10	2
分拣	2	10	5
agv小车抓	2	9	17

进入流程编排

首页

组件开发

所有场景

应用所有应用

应用实例

组件库

应用 / 应用页 /

我的应用 公开应用

+ 点击创建应用

测试 测试

△ 组件数 ... 节点数 □ 实例数
6 5 8
18小时前 管理员 编排 定时 实例

机器人1-7故障诊断 测试

暂无描述
△ 组件数 ... 节点数 □ 实例数
3 10 56
2小时前 管理员 编排 定时 实例

自动生成功拣苹果

自动生成功拣苹果

△ 组件数 ... 节点数 □ 实例数
2 10 2
4天前 管理员 编排 定时 实例

分拣 分拣

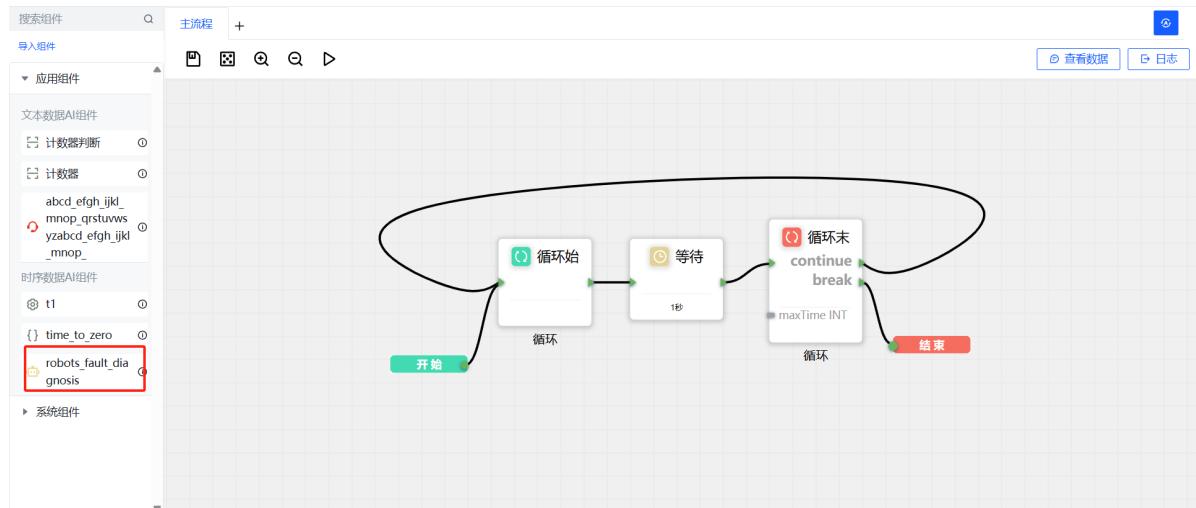
△ 组件数 ... 节点数 □ 实例数
2 10 5
5天前 管理员 编排 定时 实例

agv小车抓 测试

暂无描述
△ 组件数 ... 节点数 □ 实例数
2 9 17
5天前 管理员 编排 定时 实例

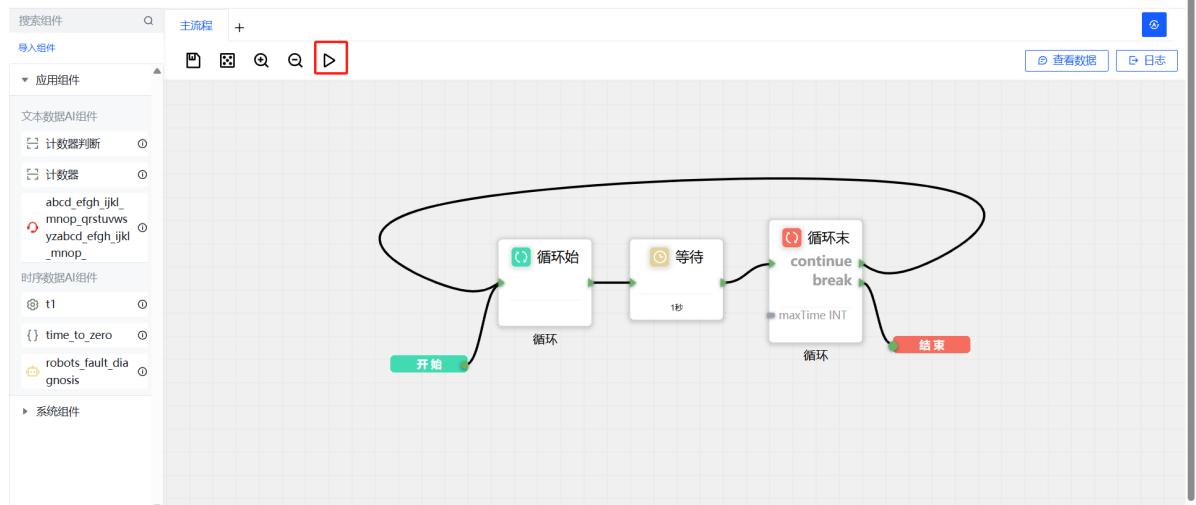
找到创建的实例，拉取出来并进行连线

测试

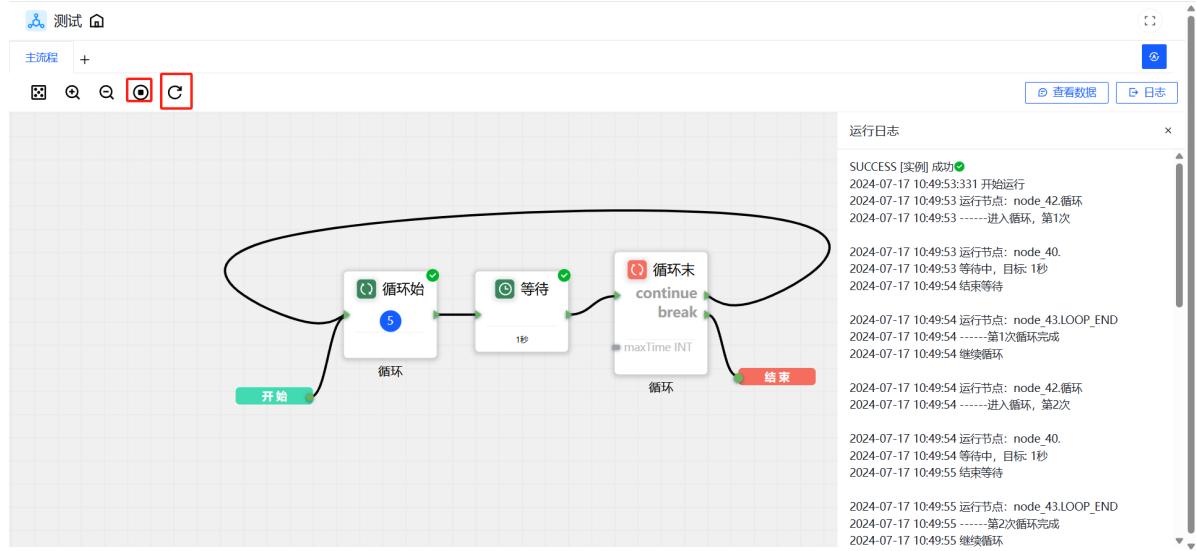


连线完成后就可以进行调试使用了

测试



可以停止或者重新运行



开发组件实例

Java组件开发

创建一个Java组件

The screenshot shows the 'Create Component' dialog in the component development interface. The dialog has the following fields:

- 组件名 (Component Name): test-java-demo
- 分类 (Category): 测试组件 (Test Component)
- 代码语言 (Code Language): Java8 (highlighted with a red box)
- 标签 (Tags): 请选择标签 (Select Tag)
- 图标 (Icon): A placeholder for an icon with a plus sign.
- 描述 (Description): A rich text editor with various formatting options.

At the bottom of the dialog are '保存' (Save) and '取消' (Cancel) buttons. The background shows a list of existing components on the right.

点击接口设计 新增一个接口

The screenshot shows the 'Add Interface' dialog in the component development interface. The dialog has the following fields:

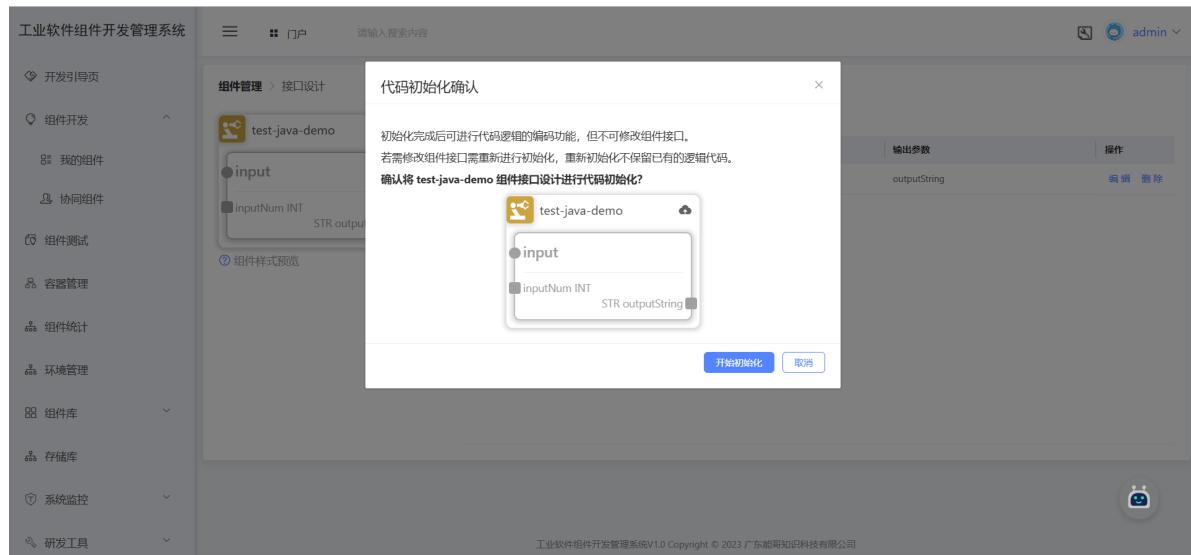
- 接口名称 (Interface Name): input
- 描述 (Description): 请输入描述 (Please enter description)
- 输入参数 (Input Parameters):
 - 新增参数 (Add Parameter):

名称 (Name)	数据类型 (Data Type)	数组类型 (Array Type)	参数描述 (Parameter Description)	操作 (Operation)
inputNum	整数 (Integer)		请输入参数描述 (Please enter parameter description)	删除 (Delete)
- 输出参数 (Output Parameters):
 - 新增参数 (Add Parameter):

名称 (Name)	数据类型 (Data Type)	数组类型 (Array Type)	参数描述 (Parameter Description)	操作 (Operation)
outputString	字符串 (String)		请输入参数描述 (Please enter parameter description)	删除 (Delete)

At the bottom of the dialog are '保存' (Save) and '取消' (Cancel) buttons. The background shows a list of existing interfaces on the right.

进行代码初始化

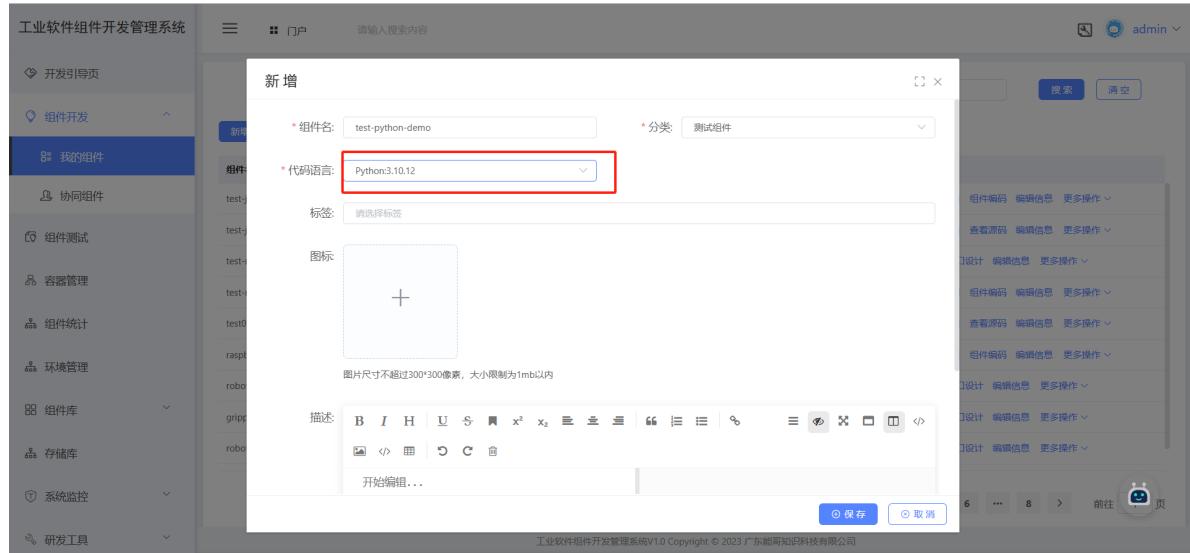


然后在 operation 找到对应接口的java文件编写我们的逻辑业务即可

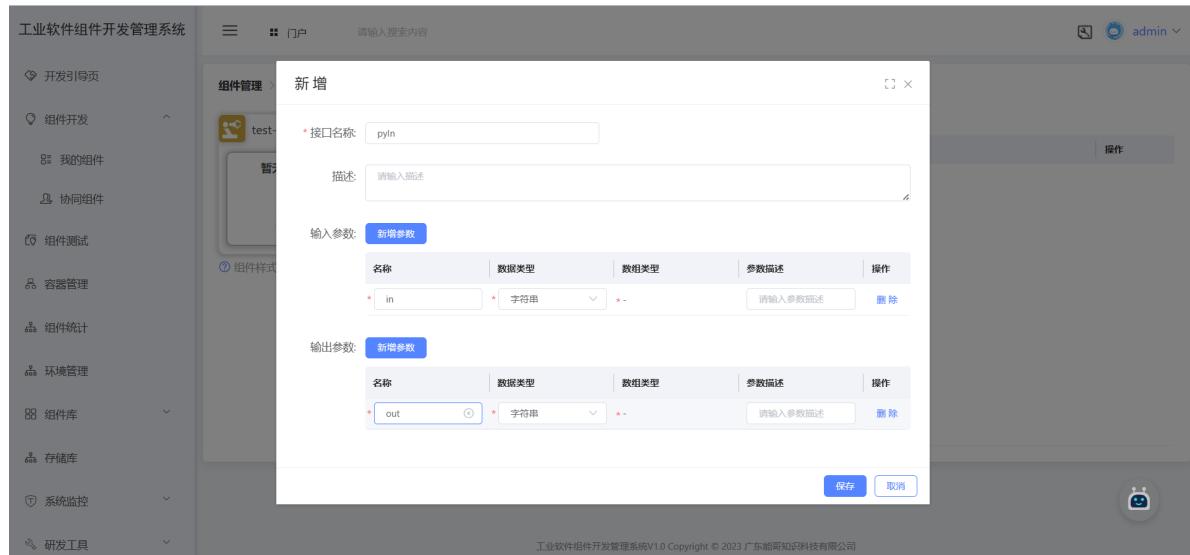
```
① InputOperation.java src/main/java/org/springblade/component/test/java/demo/admin/oqiuui/operation/InputOperation.java
25 * 1、@link OperationHandlerInterface<OperationRequest> convert()方法可将MQTT数据进行直接处理
26 * 2、@link OperationHandlerInterface<Listener> listenerHandler(String, OperationRequest, Environment)
27 * 方法可忽略组件内部处理逻辑而使用自定义逻辑，包括不限于MQTT消息返回
28 */
29 @IsDpcComponent
30 public class InputOperation implements OperationHandlerInterface<InputParam, OperationResponse<InputResponse>> {
31
32     private static final Logger log = LoggerFactory.getLogger(InputOperation.class);
33
34     @Override
35     public InputParam convert(final OperationRequest<> request) {
36         return JsonUtils.parse(JsonUtils.toJson(request.getData()), InputParam.class);
37     }
38
39     @Override
40     public OperationResponse<InputResponse> handler(final InputParam param) {
41         // todo 具体业务逻辑
42
43         // 返回报文数据
44         // return OperationResponse.of(new InputResponse());
45     }
46 }
47
```

Python组件开发

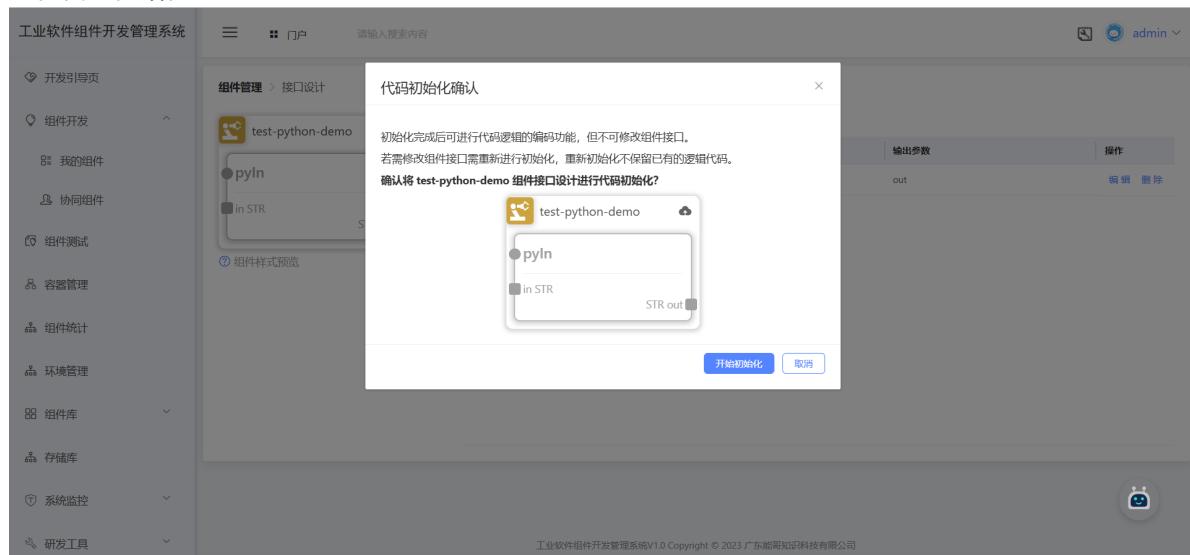
创建一个Python组件



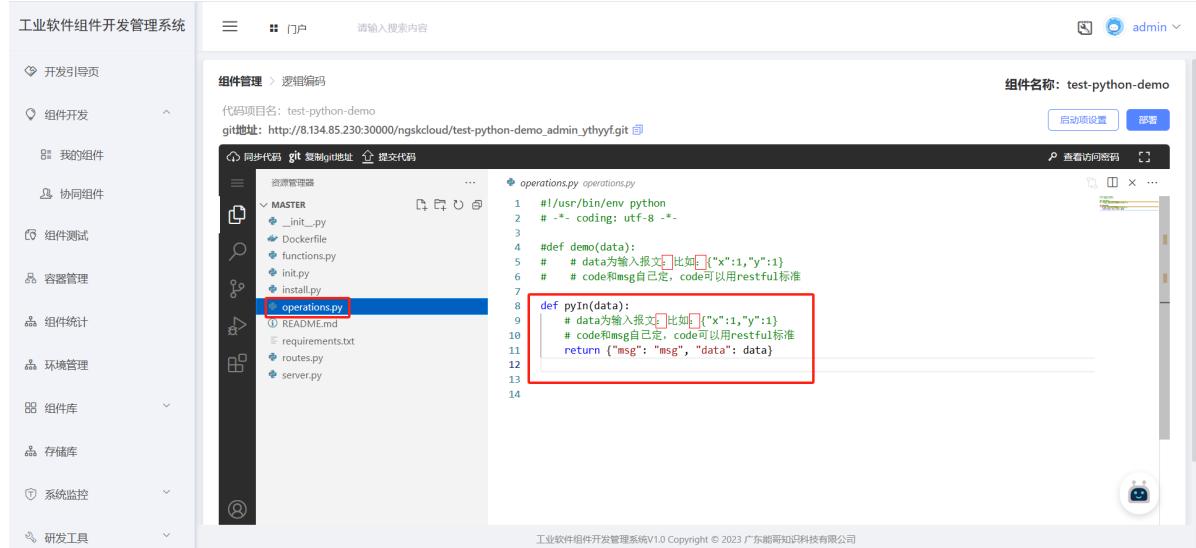
点击接口设计 新增一个接口



进行代码初始化



然后在 operations.py 编写我们的逻辑业务即可



The screenshot shows the 'Industrial Software Component Development Management System' interface. The left sidebar contains navigation links: '开发引导页', '组件开发', '我的组件', '协同组件', '组件测试', '容器管理', '组件统计', '环境管理', '组件库', '存储库', '系统监控', and '研发工具'. The main area is titled '组件管理 > 逻辑编码' and shows a component named 'test-python-demo' with a git URL: 'http://8.134.85.230:30000/ngskcloud/test-python-demo_admin_ythyf.git'. The code editor displays the 'operations.py' file, which contains the following code:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#def demo(data):
#    # data为输入报文,比如{"x":1,"y":1}
#    # code和msg自己定, code可以用restful标准
#    return {"msg": "msg", "data": data}

def pyIn(data):
    # data为输入报文,比如{"x":1,"y":1}
    # code和msg自己定, code可以用restful标准
    return {"msg": "msg", "data": data}
```

The code editor has a red box highlighting the 'operations.py' file in the file tree and the code block for 'pyIn(data)'.