

Astro 轻应用

快速入门

文档版本 01
发布日期 2024-04-28



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目录

1 入门导读	1
2 初学入门	4
2.1 注册租户	4
2.1.1 申请免费试用	4
2.1.2 购买商用租户	10
2.1.3 (可选) 购买沙箱	22
2.2 添加开发者账号	23
2.3 创建业务用户	27
2.4 认识 AstroZero	33
2.5 零代码应用开发示例	43
2.6 交通管理大屏开发示例	52
2.7 探索示例应用	63
2.7.1 场景介绍	63
2.7.2 步骤 1：使用应用模板快速创建调查问卷	65
2.7.3 (可选) 步骤 2：自定义调查问卷项	73
2.7.4 (可选) 步骤 3：自定义调查问卷页面	76
2.7.5 步骤 4：预览并体验应用	81
2.7.6 步骤 5：编译发布应用到运行环境	83
2.7.7 步骤 6：创建业务用户并登录应用	89
2.8 简易出差审批应用开发	94
2.9 创建 HelloWorld 应用	110
3 进阶实战	153
3.1 设备维修管理应用开发	153
3.1.1 注册登录指导及环境说明	153
3.1.1.1 注册账号及开通 AstroZero 服务	153
3.1.1.2 进入经典版开发环境	154
3.1.2 业务场景和流程设计介绍	155
3.1.3 功能模块分类说明	159
3.1.4 创建工作队列	160
3.1.5 创建应用	163
3.1.5.1 创建“设备维修管理系统”应用	163
3.1.5.2 了解应用目录及菜单	169

3.1.5.3 创建“业务用户登录”页面.....	170
3.1.5.3.1 背景与原理.....	170
3.1.5.3.2 上传自定义登录组件.....	172
3.1.5.3.3 组装“业务用户登录”页面.....	173
3.1.6 设备管理开发.....	182
3.1.6.1 开发前必读.....	182
3.1.6.2 定义数据对象.....	182
3.1.6.2.1 背景和原理（对象）.....	183
3.1.6.2.2 方法和实践.....	184
3.1.6.3 开发“编辑设备”功能.....	192
3.1.6.3.1 背景和原理.....	192
3.1.6.3.2 创建业务逻辑.....	203
3.1.6.3.3 组装“编辑设备”页面.....	211
3.1.6.4 开发“管理设备”功能.....	230
3.1.6.4.1 背景及原理（服务编排）.....	230
3.1.6.4.2 创建业务逻辑.....	231
3.1.6.4.3 组装“设备管理”页面.....	246
3.1.6.5 开发“设备监控”功能.....	263
3.1.6.5.1 生成设备的经纬度信息.....	263
3.1.6.5.2 组装“设备监控”页面.....	270
3.1.7 工单管理开发.....	276
3.1.7.1 开发前必读.....	277
3.1.7.2 定义数据对象“工单对象”.....	277
3.1.7.3 开发“客服人员创建工单”功能.....	281
3.1.7.3.1 本节导读.....	281
3.1.7.3.2 开发“生成工单”功能.....	282
3.1.7.3.3 开发“客服人员查询工单”功能.....	302
3.1.7.3.4 验证.....	314
3.1.7.4 开发“派单员派发工单”功能.....	315
3.1.7.4.1 创建“查询维修人员”脚本.....	316
3.1.7.4.2 创建“派单功能”脚本.....	318
3.1.7.4.3 创建公共接口.....	320
3.1.7.4.4 组装“派单”对话框.....	321
3.1.7.4.5 组装“工单列表（派单员）”页面.....	325
3.1.7.4.6 验证.....	330
3.1.7.5 开发“维修工程师处理工单”功能.....	331
3.1.7.5.1 创建“处理工单”脚本.....	331
3.1.7.5.2 创建“判断下一步状态”脚本.....	333
3.1.7.5.3 创建公共接口.....	335
3.1.7.5.4 组装“处理工单”对话框.....	336
3.1.7.5.5 组装“待处理工单”页面.....	340
3.1.7.6 开发“管理员管理工单”功能.....	358

3.1.7.6.1 创建“删除工单”脚本及公共接口.....	358
3.1.7.6.2 组装“工单管理”页面.....	360
3.1.7.7 定义工单流转 BPM.....	365
3.1.7.7.1 背景与原理（BPM）.....	365
3.1.7.7.2 创建并开发 BPM.....	368
3.1.7.8 验证工单管理功能.....	385
3.1.7.8.1 挂载前端页面.....	385
3.1.7.8.2 验证.....	388
3.1.8 用户管理功能开发.....	392
3.1.8.1 背景与知识.....	392
3.1.8.2 开发业务逻辑.....	393
3.1.8.2.1 创建用户注册脚本.....	393
3.1.8.2.2 创建用户登录脚本.....	395
3.1.8.2.3 创建用户登录服务编排.....	397
3.1.8.2.4 创建公共接口.....	409
3.1.8.3 组装页面.....	410
3.1.8.3.1 组装“业务用户注册”页面.....	410
3.1.8.3.2 组装“业务用户管理”页面.....	414
3.1.8.3.3 验证.....	425
3.1.8.4 创建业务凭证.....	429
3.1.8.5 创建权限配置.....	430
3.1.8.6 添加接口级业务权限凭证.....	439
3.1.9 应用业务测试.....	442
3.1.9.1 本节导读.....	442
3.1.9.2 管理业务用户.....	442
3.1.9.3 管理设备信息.....	448
3.1.9.4 处理工单.....	448
3.1.10 打包发布.....	455
3.1.11 FAQ.....	459
3.1.11.1 线下开发自定义登录组件.....	459
3.1.11.2 如何更新自定义组件.....	462
3.1.11.3 派单不成功，报错“无法获取 rootID”.....	464
3.2 告警监控大屏应用开发.....	464
3.2.1 场景说明.....	464
3.2.2 前提条件.....	465
3.2.3 开发大屏项目.....	467
3.2.4 迁移大屏项目.....	486
4 高阶实战.....	490
4.1 发布 WeLink We 码应用.....	490
4.2 发布 WeLink 轻应用.....	513
4.3 发布微信小程序.....	539

5 入门实践..... 545

1 入门导读

Astro轻应用（简称AstroZero）是华为云为行业客户、合作伙伴、开发者量身打造的一款零代码和低代码应用开发平台。通过AstroZero可轻松构建专业级应用，创新随心所欲、敏捷超乎想象。

初次使用AstroZero的您，如果对AstroZero还不太了解，不知道从何处入手，请仔细阅读该入门导读。通过入门导读可帮助您由浅入深的学习AstroZero，轻松完成专业级应用的构建。

图 1-1 入门指引

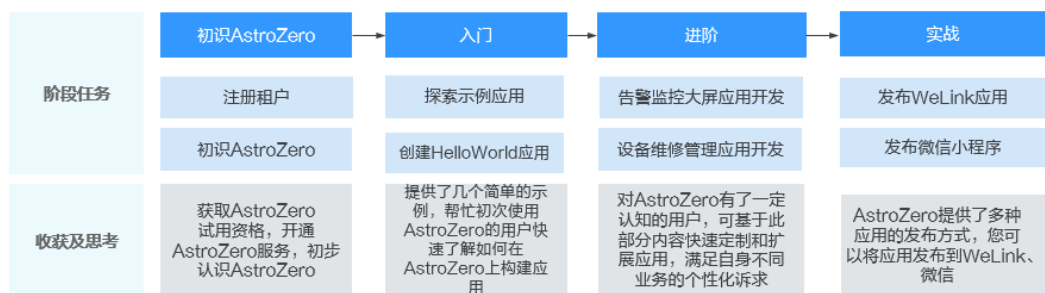


表 1-1 阶段分解

阶段	任务	内容说明	难易程度
初学入门 AstroZero	注册租户	<p>AstroZero提供免费版、标准版、专业版和专享版四种规格套餐，各规格说明可参见规格说明。</p> <p>说明 开通AstroZero免费版、标准版和专业版实例时，会同步创建一个Astro workflow基础版实例。Astro workflow（Astro Flow，简称AstroFlow）可帮助开发者快速构建业务流程及自动化工作流，轻松实现人财物事的调、转、入、离、审、评和批等任务的数字化需求。更多关于Astro workflow的介绍，请参见Astro workflow AstroFlow。</p>	-

阶段	任务	内容说明	难易程度
	添加开发者账号	在同一华为账号下，可通过添加开发者账号实现多人协同开发。	-
	创建业务用户	业务用户是应用的使用者，在业务配置中心，可根据需要创建业务用户。	-
	认识AstroZero	初次使用AstroZero时，通过阅读本章节，初步了解AstroZero。	-
	零代码应用开发示例	使用零代码应用模板（培训报名）创建一个零代码应用，快速体验在AstroZero中开发一个零代码应用的大致流程。	★
	交通管理大屏开发示例	使用Astro Canvas中预置的模板构建一个交通管理大屏，快速体验在Astro Canvas中构建可视化页面的大致流程。	★
	探索示例应用	使用低代码应用模板（调查问卷）创建一个示例应用，快速体验在AstroZero中开发一个轻应用的大致流程。	★
	简易出差审批应用开发（BTA）	使用AstroZero的BPM功能，快速创建一个简易版的出差审批应用。	★
	创建HelloWorld应用	从空白应用开始，带您一步步创建一个HelloWorld轻应用。HelloWorld轻应用着重带您了解创建应用每一步的原理及概念。	★★
进阶实战	设备维修管理应用开发	如果您已经有一定AstroZero开发基础，可以挑战该应用，从场景规划、开发步骤、以及开发能力发现AstroZero更多能力。	★★★
	告警监控大屏应用开发	通过自由拖拽组件到画布区域，创建一个告警监控大屏应用。	★★
高阶实战	发布WeLink We码应用	WeLink是华为云的一款智能办公云服务平台，AstroZero支持将开发完成的应用发布成WeLink小程序。	★★★★
	发布WeLink轻应用	<ul style="list-style-type: none"> • We码应用：运行在WeLink APP中的小程序。其提供了丰富的API，开发者可以利用前端技术写出接近原生体验的应用。 • 轻应用：采用H5方式开发的应用，企业可快速将已有系统或自己全新开发的Web页面接入到WeLink。 	★★★★

阶段	任务	内容说明	难易程度
	发布微信小程序	AstroZero允许将标准页面或高级页面发布到微信小程序。本示例带您了解如何将AstroZero上的应用发布到微信小程序。	★★★★★

2 初学入门

2.1 注册租户

2.1.1 申请免费试用

AstroZero支持申请免费试用，试用版可以使用的资源有限。AstroZero免费试用版本不限制使用时间，不涉及AstroZero退订及资源释放。开通AstroZero免费版实例时，会同步创建一个Astro workflow基础版实例。Astro workflow（Astro Flow，简称AstroFlow）可以帮助开发者快速构建业务流程及自动化 workflow，轻松实现人财物事的调、转、入、离、审、评和批等任务的数字化需求。更多关于Astro workflow的介绍，请参见[Astro workflow AstroFlow](#)。

前提条件

只有华为账号或具有“Astro Zero Instance ManageAccess”权限的IAM用户，才能开通AstroZero服务。如何给用户配置权限，请参见[创建用户组并授权](#)。如果以IAM用户开通AstroZero服务，其对应的华为账号也会同时开通AstroZero服务。

图 2-1 给 IAM 用户所在用户组赋权（选择策略）



图 2-2 给 IAM 用户所在用户组赋权（设置最小授权范围）

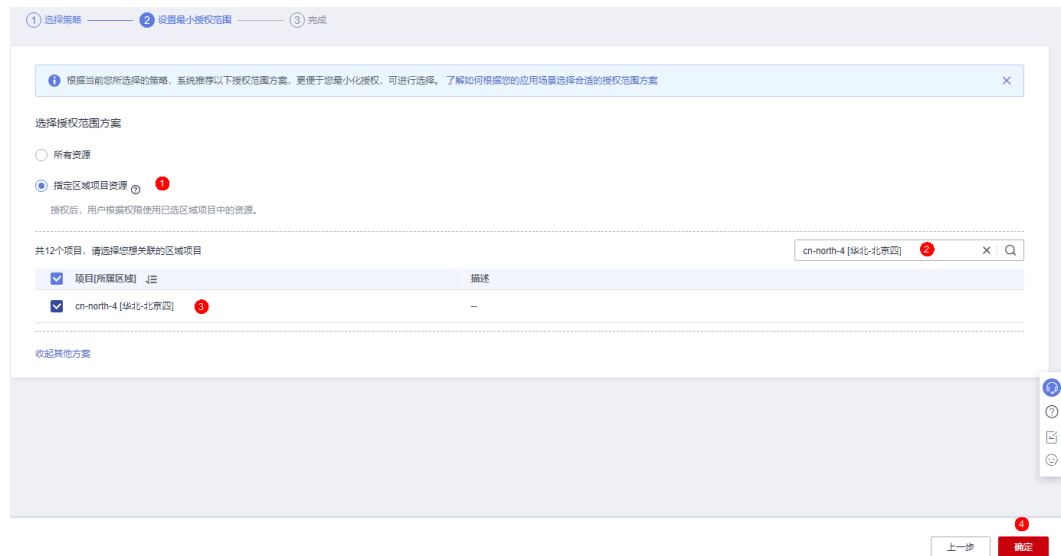


表 2-1 权限说明

权限	作用范围	权限说明
Astro Zero Instance ManageAccess	区域级项目	Astro轻应用实例管理权限，拥有订购、退订、查看和升级AstroZero实例的权限。
Astro Zero Instance ViewAccess	区域级项目	Astro轻应用实例查看权限，只拥有查看AstroZero实例的权限，不可进行退订、升级等操作。 如果IAM用户登录AstroZero服务控制台，查看不到AstroZero实例，请按照如下两种方式中的一种进行处理： <ul style="list-style-type: none"> 给IAM用户所在的用户组，添加“Astro Zero Instance ViewAccess”权限。 不要将IAM用户添加到任何用户组中。
Astro Zero IAM User QueryAccess	全局服务	Astro轻应用IAM用户查询权限，只有华为账号或具有“Astro Zero IAM User QueryAccess”权限的IAM用户，才能执行添加开发者账号的操作。

本章节以**华为账号**，申请免费试用AstroZero服务为例。

限制和说明

- AstroZero免费版不收费，但最多支持添加10个用户（包括业务用户），且可使用的资源有限。
- 对于低代码应用，AstroZero免费版仅提供低代码应用开发环境，不提供将低代码应用发布至运行环境的能力。如果需要将应用部署到运行环境，请参考[购买商用租户](#)中操作，购买专业版。

- 免费版提供零代码应用开发能力，不需要使用任何编程语言，通过拖、拉、拽即可快速构建应用，操作零门槛，人人都可成为开发者。

操作步骤

步骤1 登录[华为云官网](#)。

如果您没有华为账号，请立即注册并开通华为云，相关操作请参见[注册华为账号并开通华为云](#)。注册后参考[个人账号如何完成实名认证](#)或[企业账号如何完成实名认证](#)完成个人或企业账号实名认证。

步骤2 在“产品”中，选择“开发与运维 > 低代码平台 Astro > Astro轻应用 Astro Zero”。

图 2-3 选择 Astro 轻应用



步骤3 在AstroZero产品介绍页，单击“立即使用”，进入AstroZero服务控制台。

图 2-4 进入 AstroZero 服务控制台



步骤4 在AstroZero服务控制台，单击页面右上角的“免费试用”。

图 2-5 免费试用



步骤5 在弹出的页面，单击“免费开通”，即可免费开通AstroZero实例。

图 2-6 免费开通 AstroZero 实例



步骤6 开通免费版AstroZero实例后，可根据需要进入AstroZero新版/经典版开发环境首页或零代码应用开发工作台。

- 单击“进入首页”，进入AstroZero新版开发环境首页，可创建各类型应用。

图 2-7 新版本开发环境



- 单击“旧版入口”，进入AstroZero经典版低代码应用开发环境。

图 2-8 经典版开发环境



- 单击“管理运行环境”，进入AstroZero零代码应用开发工作台（免费版不提供运行环境）。

图 2-9 零代码工作台




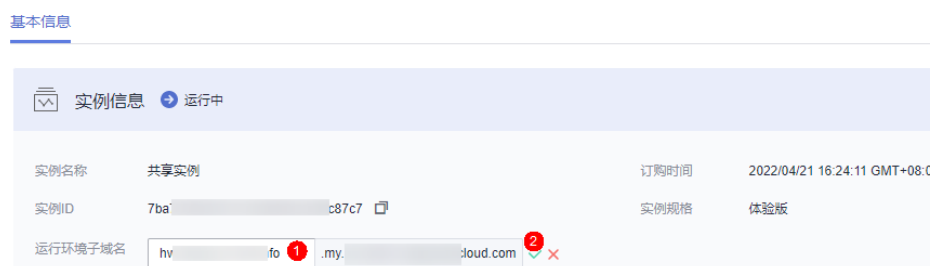
- 在实例中，单击“操作 > 详情”，进入实例信息页面。单击“基本信息 > 运行环境子域名”后的 ，可修改运行环境子域名。修改后，业务用户可使用子域名访问业务应用。

图 2-10 修改子域名



- 在实例中，单击“操作 > 退订”，可退订已订购的AstroZero实例。

图 2-11 退订实例



----结束

2.1.2 购买商用租户

AstroZero免费版可使用的资源有限，标准版、专业版和专享版更适用于个人或企业的商用开发。本章节将为您介绍，如何购买商用租户。开通AstroZero标准版和专业版实例时，会同步创建一个Astro工作流基础版实例。Astro工作流（Astro Flow，简称AstroFlow）可帮助开发者快速构建业务流程及自动化工作流，轻松实现人财物事的调、转、入、离、审、评和批等任务的数字化需求。更多关于Astro工作流的介绍，请参见[Astro工作流 AstroFlow](#)。

前提条件

只有华为账号或具有“Astro Zero Instance ManageAccess”权限的IAM用户，才能开通AstroZero。如何给用户配置权限，请参见[创建用户组并授权](#)。如果以IAM用户开通AstroZero，其对应的华为账号也会同时开通AstroZero服务。

图 2-12 给 IAM 用户所在用户组赋权（选择策略）



图 2-13 给 IAM 用户所在用户组赋权（设置最小授权范围）

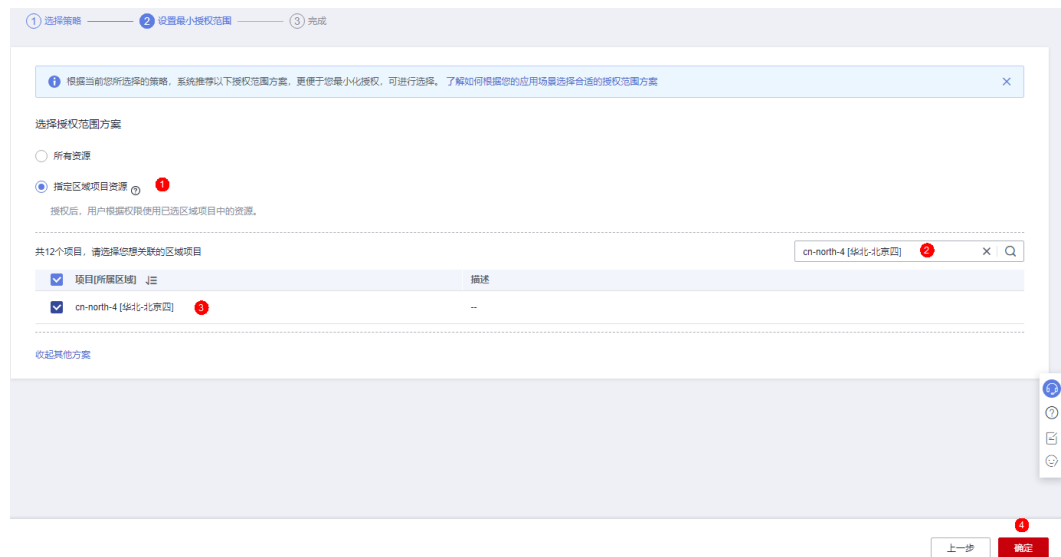


表 2-2 权限说明

权限	作用范围	权限说明
Astro Zero Instance ManageAccess	区域级项目	Astro轻应用实例管理权限，拥有订购、退订、查看和升级AstroZero实例的权限。
Astro Zero Instance ViewAccess	区域级项目	Astro轻应用实例查看权限，只拥有查看AstroZero实例的权限，不可进行退订、升级等操作。 如果IAM用户登录AstroZero服务控制台，查看不到AstroZero实例，请按照如下两种方式中的一种进行处理： <ul style="list-style-type: none"> 给IAM用户所在的用户组，添加“Astro Zero Instance ViewAccess”权限。 不要将IAM用户添加到任何用户组中。
Astro Zero IAM User QueryAccess	全局服务	Astro轻应用IAM用户查询权限，只有华为账号或具有“Astro Zero IAM User QueryAccess”权限的IAM用户，才能执行添加开发者账号的操作。

本章节以华为账号，购买AstroZero标准版、专业版和专享版实例为例向您介绍。

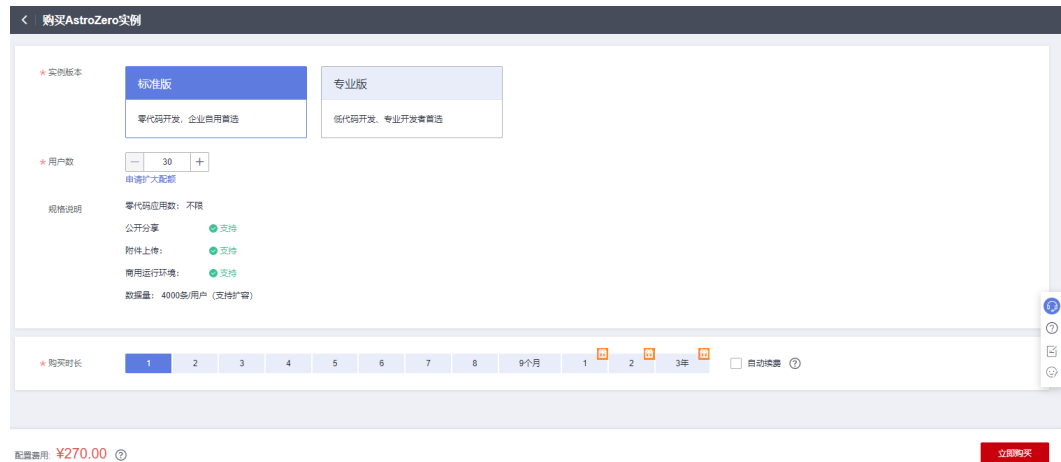
购买标准版

标准版提供开发环境和开发零代码应用的运行环境，不提供低代码应用的运行环境，适用于零代码应用开发场景，开发者不需要有代码开发经验，企业自用首选。

步骤1 进入[购买Astro轻应用标准版实例页面](#)。

步骤2 在购买AstroZero实例页面，设置相关参数，单击“立即购买”。

图 2-14 购买标准版



- 实例版本：即规格，请选择“标准版”。
- 用户数：购买的用户数，此处的用户数包括业务用户数。
- 规格说明：AstroZero标准版实例对应的规格，详细介绍请参见[规格说明](#)。
- 购买时长：设置实例的购买时长。

说明

实例购买后，可在“费用中心 > 订单管理”中，查看已购买的AstroZero订单。

步骤3 选择支付方式，单击“去在线支付”，完成订单支付。

图 2-15 支付订单



步骤4 订单支付成功后，单击“返回Astro轻应用控制台”，即可进入AstroZero服务控制台。

步骤5 标准版实例创建成功后，可根据需要进入AstroZero开发环境和运行环境。

创建AstroZero实例，预计需要几分钟，请耐心等待。

图 2-16 已订购服务



步骤6（可选）在已创建的实例中，可对实例执行续费、退订等操作。

- 续费：选择“操作 > 续费”，可进行续费操作，包括延长购买时长。
- 升级规格：选择“操作 > 升级规格”，可增加用户数，升级购买的实例规格。
- 详情：选择“操作 > 详情”，可查看当前实例的详情。在开发环境和运行环境中，租户创建的资源也会展示在详情页。
- 退订：选择“操作 > 退订”，根据提示可退订服务，降低为免费版规格，退订后不会删除数据。

步骤7（可选）配置运行环境子域名。

标准版不仅提供了AstroZero开发环境，还提供了AstroZero运行环境。运行环境是指正式发布上线的真实环境，更多环境介绍请参见[基本概念](#)。

在开发环境开发好应用并完成测试后，可将应用发布至生产环境，即运行环境，供业务用户使用。AstroZero支持设置运行环境个性化的子域名，设置后，业务用户可使用子域名访问业务应用。华北-北京四服务区域，AstroZero运行环境默认域名为“appcuberun.cn-north-4.huaweicloud.com”。

1. 在实例中，单击“操作 > 详情”。
2. 在实例详情页面，单击“运行环境子域名”后的✎，可修改运行环境子域名。修改域名后，业务用户可使用子域名访问业务应用。您也可以在运行环境的环境配置页面，设置访问域名，更多介绍请参见[管理域名配置](#)。

----结束

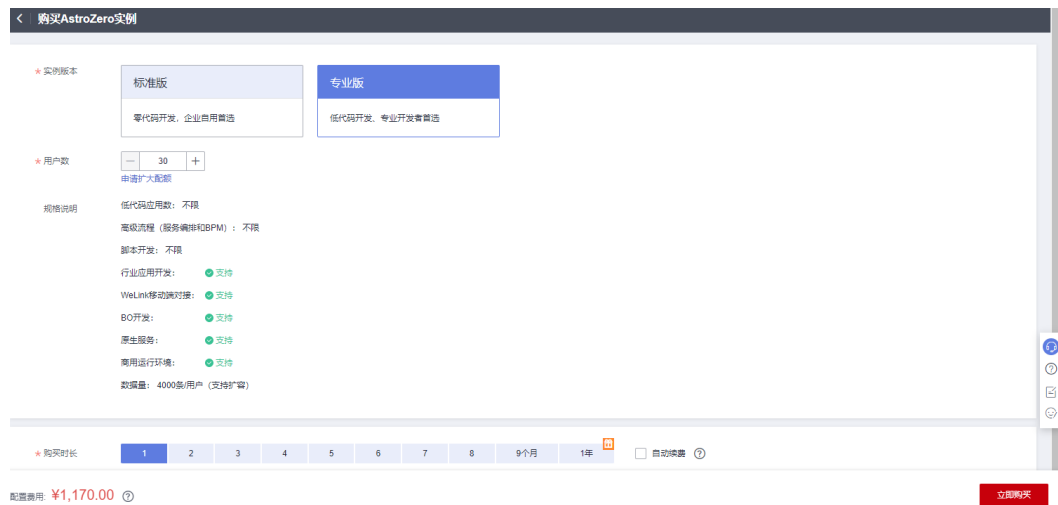
购买专业版

专业版提供了更多功能和资源，不仅提供了开发环境，还提供了运行环境（多租户共享运行环境），适用于专业开发者。

步骤1 进入[购买Astro轻应用专业版实例页面](#)。

步骤2 在购买AstroZero实例页面，设置相关参数，单击“立即购买”。

图 2-17 购买专业版实例



- 实例版本：即规格，请选择“专业版”。
- 规格说明：AstroZero专业版实例对应的规格，详细介绍请参见[规格说明](#)。
- 用户数：购买的用户数，此处的用户数包括业务用户数。
- 购买时长：设置实例的购买时长。

说明

实例购买后，可在“费用中心 > 订单管理”中，查看已购买的AstroZero订单。

步骤3 确认支付方式，单击“去在线支付”，完成订单支付。

图 2-18 支付订单

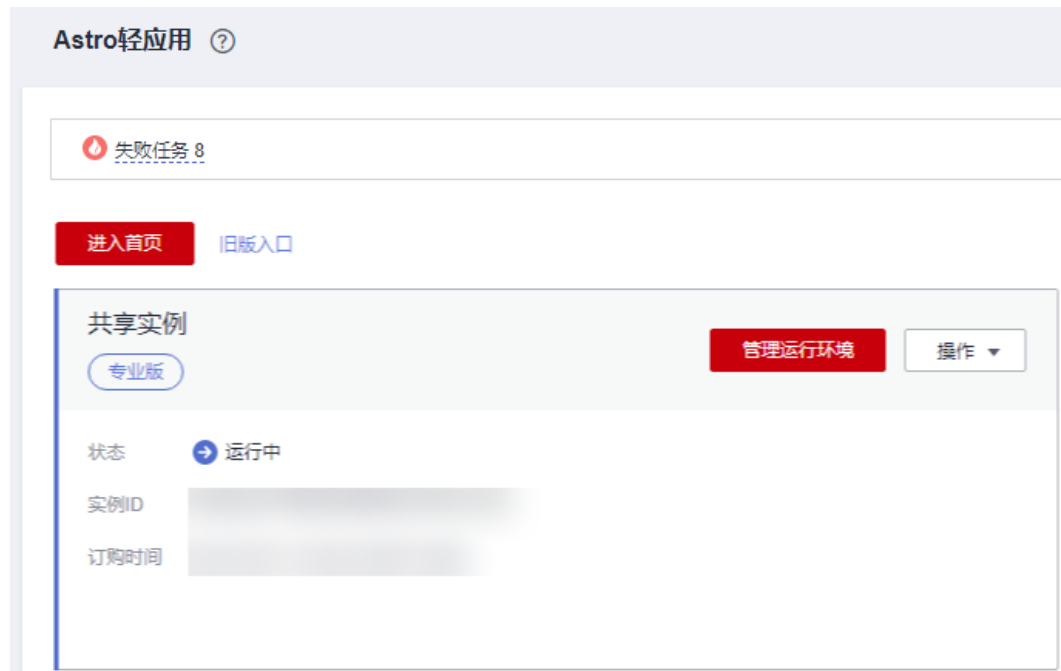


步骤4 订单支付成功后，单击“返回Astro轻应用控制台”，即可进入AstroZero服务控制台。

步骤5 专业版实例创建成功后，可根据需要进入AstroZero开发环境或运行环境。

创建AstroZero实例，预计需要几分钟，请耐心等待。

图 2-19 已订购服务



- 单击“进入首页”，进入AstroZero新版开发环境首页，可创建各类型应用。

图 2-20 新版本开发环境



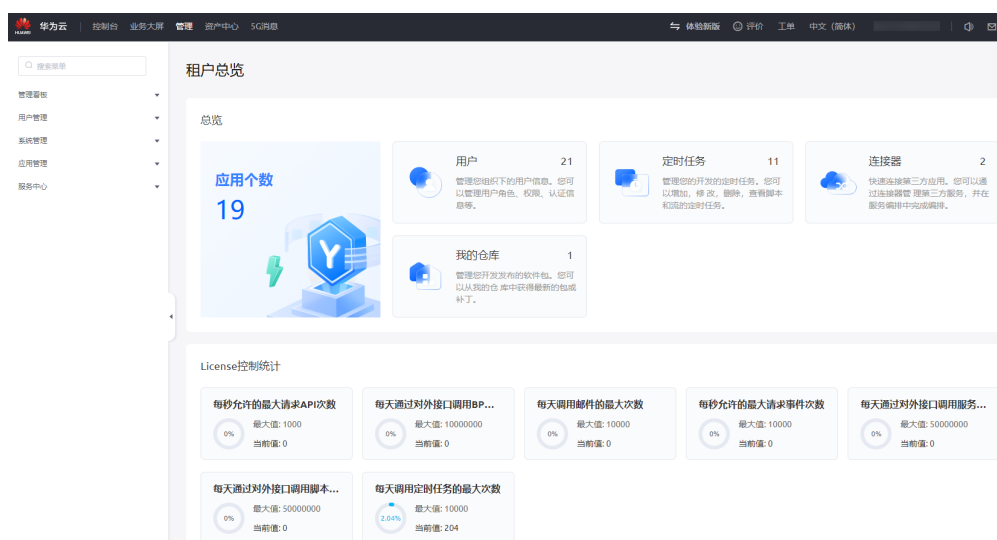
- 单击“旧版入口”，进入AstroZero经典版低代码应用开发环境。

图 2-21 经典版开发环境



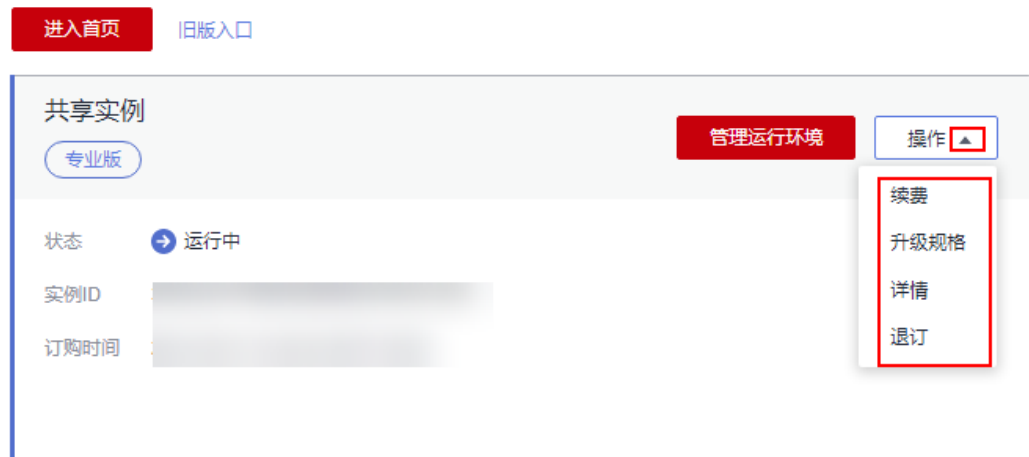
- 单击“管理运行环境”，进入AstroZero运行环境。

图 2-22 AstroZero 运行环境



步骤6（可选）在已创建的实例中，可对实例执行续费、退订等操作。

图 2-23 操作实例



- 续费：选择“操作 > 续费”，可进行续费操作，包括延长购买时长。
- 升级规格：选择“操作 > 升级规格”，可增加用户数，升级购买的实例规格。
- 详情：选择“操作 > 详情”，可查看当前实例的详情。在开发环境和运行环境中，租户创建的资源也会展示在详情页。
- 退订：选择“操作 > 退订”，根据提示可退订服务，降低为免费版规格，退订后不会删除数据。

步骤7 （可选）配置运行环境子域名。

专业版不仅提供了AstroZero开发环境，还提供了AstroZero运行环境。运行环境是指正式发布上线的真实环境，更多环境介绍请参见[基本概念](#)。

在开发环境开发好应用并完成测试后，可将应用发布至生产环境，即运行环境，供业务用户使用。AstroZero支持设置运行环境个性化的子域名，设置后，业务用户可使用子域名访问业务应用。华北-北京四服务区域，AstroZero运行环境默认域名为“appcuberun.cn-north-4.huaweicloud.com”。

1. 在实例中，单击“操作 > 详情”。
2. 在实例详情页面，单击“运行环境子域名”后的✎，可修改运行环境子域名。
修改域名后，业务用户可使用子域名访问业务应用。您也可以运行环境的环境配置页面，设置访问域名，更多介绍请参见[管理域名配置](#)。

----结束

购买专享版

专享版实例需单独付费购买，请确认已[提交工单](#)申请开通专享版，否则，将无法购买专享版。专享版提供了物理隔离的运行环境和沙箱环境，运行环境、沙箱环境实例发放到租户虚拟私有云中。

步骤1 登录[华为云官网](#)。

如果您没有华为账号，请立即注册并开通华为云，相关操作请参见[注册华为账号并开通华为云](#)。注册后参考[个人账户如何完成实名认证](#)或[企业账号如何完成实名认证](#)完成个人或企业账号实名认证。

步骤2 （可选，如果之前没有创建过虚拟私有云，请执行此步骤）参考[创建虚拟私有云和子网](#)中操作，创建虚拟私有云VPC。

虚拟私有云可以为您的弹性云服务器构建隔离的、用户自主配置和管理的虚拟网络环境。当用户购买专享版AstroZero，且需要使用其他云服务（如ROMA Connect）或原生服务时，请保证AstroZero和其他云服务、原生服务处于同一VPC。如果处于不同VPC，则需要参考[VPC对等连接](#)中操作，创建VPC对等连接，打通VPC。

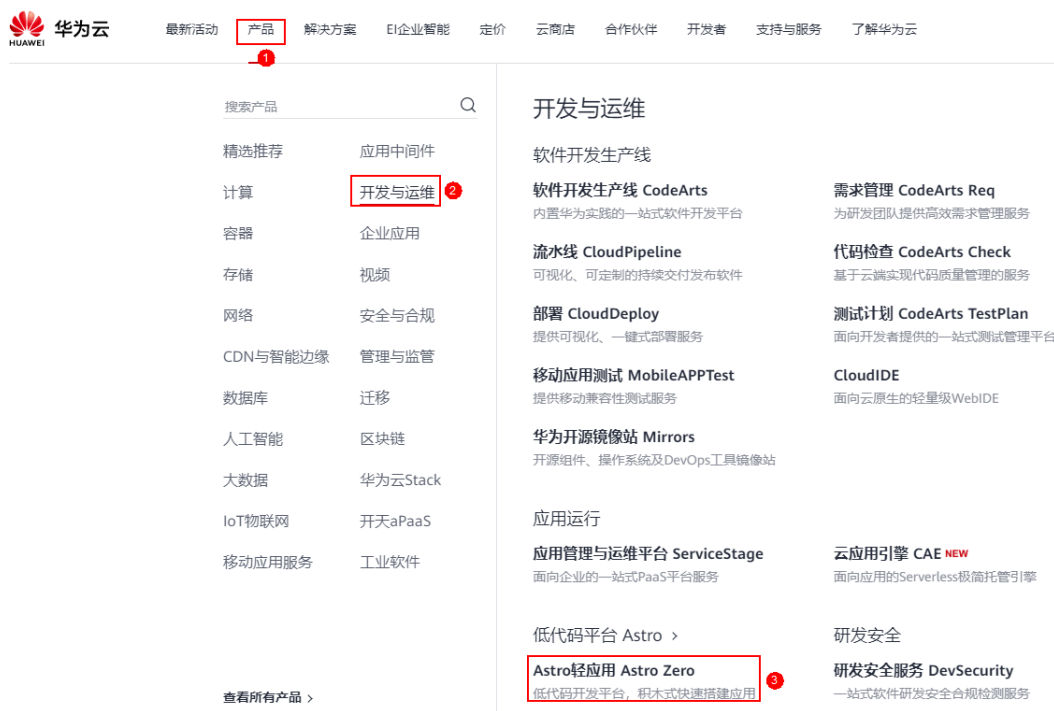
步骤3 （可选，如果不使用默认的安全组，请执行此步骤）参考[创建安全组](#)中操作，创建新的安全组。

安全组是一个逻辑上的分组，为具有相同安全保护需求并相互信任的云服务器提供访问策略。安全组创建后，用户可以在安全组中定义各种访问规则，当云服务器加入该安全组后，即受到这些访问规则的保护。

系统会为每个用户默认创建一个默认安全组，默认安全组的规则是在出方向上的数据报文全部放行，入方向访问受限，安全组内的云服务器无需添加规则即可互相访问。您可以直接使用默认安全组，详情请参见[默认安全组和规则](#)。

步骤4 在“产品”中，选择“开发与运维 > 低代码平台 Astro > Astro轻应用 Astro Zero”。

图 2-24 选择 Astro 轻应用



步骤5 在AstroZero产品介绍页，单击“立即使用”，进入AstroZero服务控制台。

图 2-25 进入 AstroZero 服务控制台



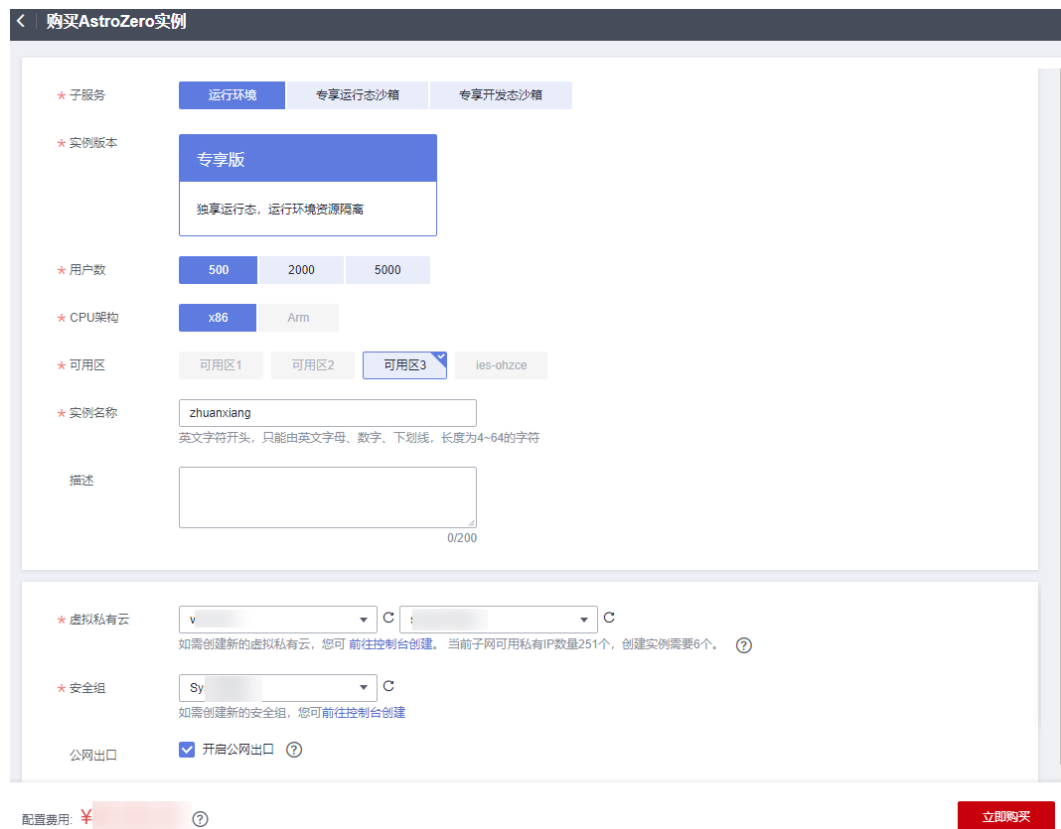
步骤6 在AstroZero服务控制台页面右上方，单击“购买AstroZero”。

须知

如果已购买过专业版AstroZero实例，请单击“购买专享版”。

步骤7 在购买AstroZero实例页面，设置实例参数，单击“立即购买”。

图 2-26 购买专享版实例



- 子服务：选择购买的子服务。
 - 运行环境：物理隔离的运行环境。
 - 专享运行态沙箱：物理隔离的运行态沙箱环境。
 - 专享开发态沙箱：物理隔离的开发态沙箱环境。
- 实例版本：请选择专享版，更多介绍请参见[规格说明](#)。
- 用户数：使用AstroZero的用户数（包括业务用户）。
- CPU架构：新建AstroZero实例的CPU架构，支持X86和ARM。
- 可用区：新建AstroZero实例，所在的可用区。可用区是在同一区域下，电力、网络隔离的物理区域，可用区之间内网互通，不同可用区之间物理隔离。如果需要提高实例的高可靠性，建议将实例创建在不同的可用区。
- 实例名称：新建AstroZero实例的名称，同一账号下实例不可重名，创建后不可修改。
- 描述：新建AstroZero实例的描述信息。
- 虚拟私有云：新建实例所在的虚拟私有云，实例创建后不可更改。
虚拟私有云是通过逻辑方式进行网络隔离，提供安全、隔离的网络环境。如果需要创建新的虚拟私有云，请单击“前往控制台创建”进行创建，完成创建后单击刷新。
- 安全组：新建实例所在的安全组，实例创建后不可更改。
安全组是一个逻辑上的分组，为同一个VPC内具有相同安全保护需求并相互信任的实例提供访问策略。如果需要创建新的安全组，请单击“前往控制台创建”进行创建，完成创建后单击刷新。

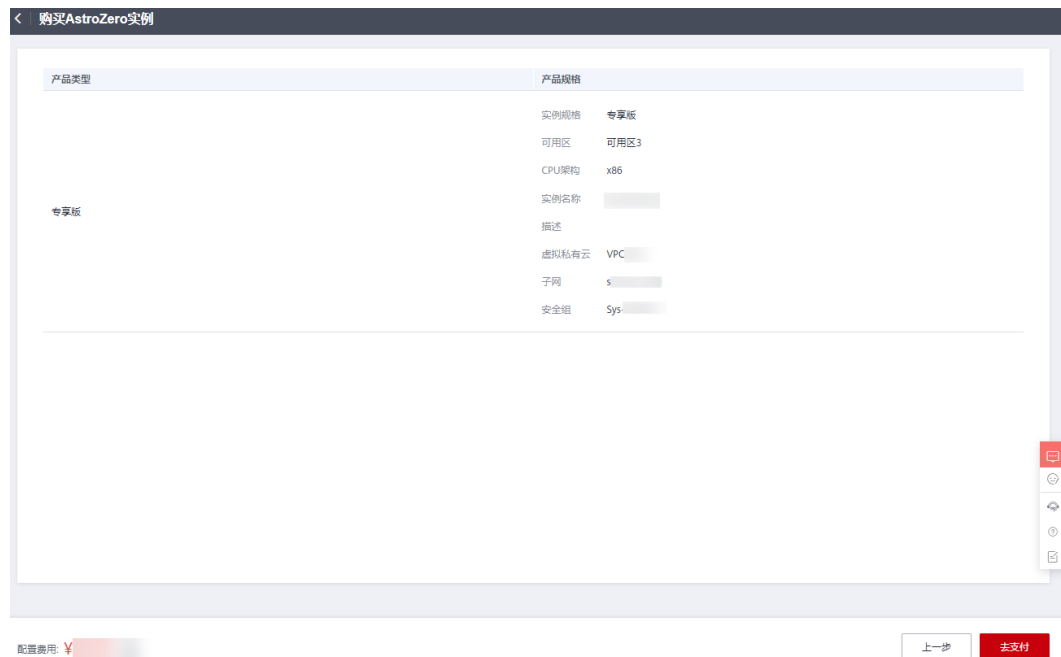
- 公网出口：是否开启服务器访问外部公网/局域网内大网网络，网络环境与云平台EIP（弹性公网IP）网络环境一致，当AstroZero需要访问外部邮箱服务或其它第三方网站等需求时，建议开启。实例创建后，参数**不支持**修改。
- 购买时长：选择实例的购买时长。

📖 说明

在“费用中心 > 订单管理 > 待支付订单”中，单击对应AstroZero订单后的“取消”，可取消订单。

步骤8 确认规格无误后，单击“去支付”。

图 2-27 规格确认



步骤9 支付完成后，单击“返回Astro轻应用控制台”，即可进入AstroZero服务控制台。

步骤10 AstroZero实例创建成功后，单击专享版实例中的“管理运行环境”，即可进入独立的运行环境。

步骤11 （可选）设置专享版运行环境租户子域名。

1. 在实例列表中，单击实例中的“详情”，进入实例信息页面。
2. 单击“访问地址”后的“更改”，并单击“请选择证书”，选择域名证书文件。

当前仅支持HTTPS加密协议的SSL证书（SSL Certificates）。域名证书文件必须为zip格式，且zip包中只能包含证书和密钥两个文件，文件格式可以为“.key”、“.crt”、“.cert”、“.csr”或“.pem”。使用用户自定义证书时，必须从正规渠道申请可信的证书和域名。

修改域名后，可以通过域名访问AstroZero运行环境，请确保域名已正确解析到弹性公网IP。您也可以在运行环境的环境配置页面，设置访问域名，更多介绍请参见[管理域名配置](#)。

----结束

2.1.3 （可选）购买沙箱

在应用正式发布上线前，需要在测试环境进行测试和数据配置。在AstroZero中，以沙箱（Sandbox）环境来作为测试环境，您也可以直接在开发环境进行测试。在沙箱环境测试的优势是不影响开发环境的配置，沙箱环境（Sandbox）是与开发环境完全隔离的。

操作步骤

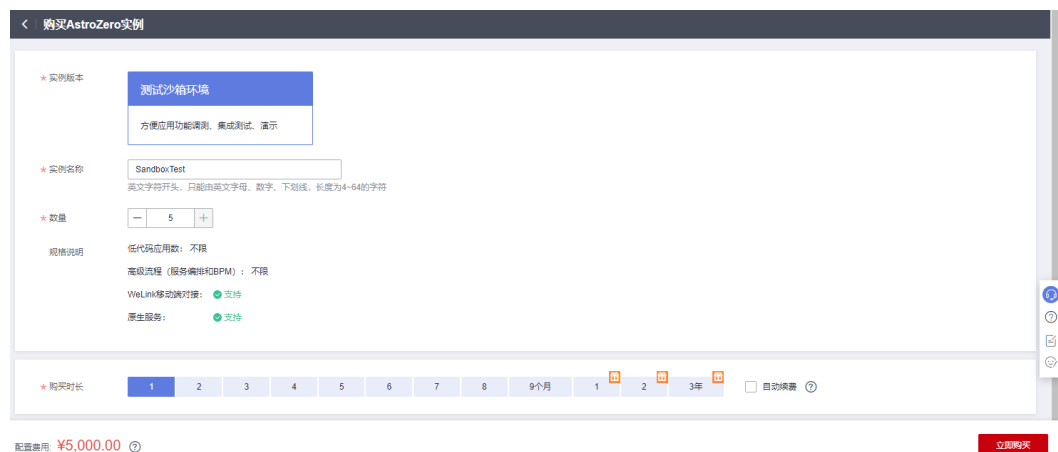
步骤1 进入[购买Astro轻应用沙箱实例页面](#)。

📖 说明

申请免费试用AstroZero或购买商用租户后，AstroZero服务控制台才会显示“购买沙箱”入口。

步骤2 在购买AstroZero实例页面，设置对应参数，单击“立即购买”。

图 2-28 购买沙箱



- 实例版本：当前仅支持功能调试沙箱。
- 实例名称：新建AstroZero实例的名称，同一账号下实例不可重名，创建后不可修改。
- 数量：购买的沙箱数量（一次性购买最大支持5个，可以进行多次购买）。
- 购买时长：设置沙箱的购买时长。

步骤3 在购买Astro轻应用页面，确认支付方式，单击“去在线支付”。

图 2-29 支付订单

购买Astro轻应用

请于 7天 内完成付款，超过时间订单将自动取消

云服务订单 订单已生成，请核对订单金额并尽快支付

订单号	商品名称	服务提供方	订单金额	应付金额
CS2301181412S1W7F	Astro轻应用	华为云	¥5,000.00	¥5,000.00

选择支付方式 [申请线上合同请款](#)

您需支付: ¥5,000.00

余额/在线支付

余额支付 使用账户余额支付 ¥ 9.91 (现金余额: ¥9.91)
若您有自动扣款的资源，如按需资源、自动续费资源等，请规划好资金，避免扣款失败影响您的使用

在线支付 使用第三方在线支付 ¥4,990.09
支持 等多种在线支付方式

余额支付 ¥9.91
在线支付 ¥4,990.09

[去在线支付](#)

步骤4 支付成功后，单击“返回Astro轻应用控制台”，即可进入AstroZero服务控制台。

步骤5 进入AstroZero沙箱环境。

创建功能调试沙箱实例预计需要几分钟，请耐心等待。创建成功后，控制台会显示已成功购买的沙箱环境实例，单击相应沙箱环境实例的“管理沙箱环境”，即可进入相应环境。

----结束

2.2 添加开发者账号

华为账号免费试用或购买AstroZero服务后，在同一个华为账号下，可以通过添加开发者账号（即子账号）的方式实现多人协同开发，即账号下所有开发者可以共享应用资源，例如查看、使用已创建的应用、服务编排、脚本等。

背景信息

开发者账号也可称为用户，其信息存储在AstroZero的标准对象User中。在AstroZero中，可以通过如下两种方式添加开发者账号：

- 添加IAM开发者：添加的IAM开发者账号，必须是从华为云IAM同步过来的。
- 添加WeLink开发者：将WeLink用户添加到AstroZero中，作为开发者账号，详情请参见[添加WeLink用户为开发者账号](#)。

本章节以添加IAM开发者为例，向您介绍如何在AstroZero中添加开发者账号。

前提条件

- 添加的IAM开发者账号是从华为云IAM同步过来的，所以执行添加IAM开发者账号前，请先登录IAM，创建IAM用户，详细操作请参见[创建用户](#)。
- 只有华为账号或具有“Astro Zero IAM User QueryAccess”权限的IAM用户，才能执行创建子账号的操作。如何给用户配置权限，请参见[创建用户组并授权](#)。

添加 IAM 用户为开发者账号

步骤1 在AstroZero开发环境添加IAM用户为开发者账号。



1. 使用华为账号，登录[华为云网站](#)，在顶部导航栏右侧单击“控制台”，进入华为云控制台。
2. 在左侧导航栏上方，单击，选择服务实例所在的区域项目。
3. 单击，在查找框中搜索“Astro轻应用”，单击查找到的结果，即可进入AstroZero服务控制台。

图 2-30 搜索 Astro 轻应用



4. 在AstroZero服务控制台的主页中，单击“进入首页”，进入新版应用开发页面。

图 2-31 新版开发环境




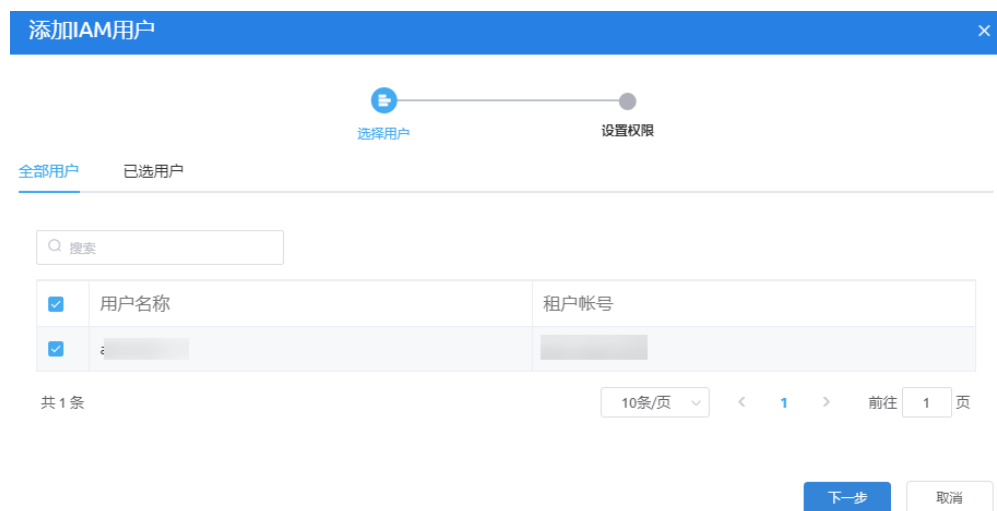
5. 在页面左上角，单击，选择“开发环境管理 > 环境配置”，进入AstroZero开发环境的环境配置。

图 2-32 进入 AstroZero 开发环境的环境配置



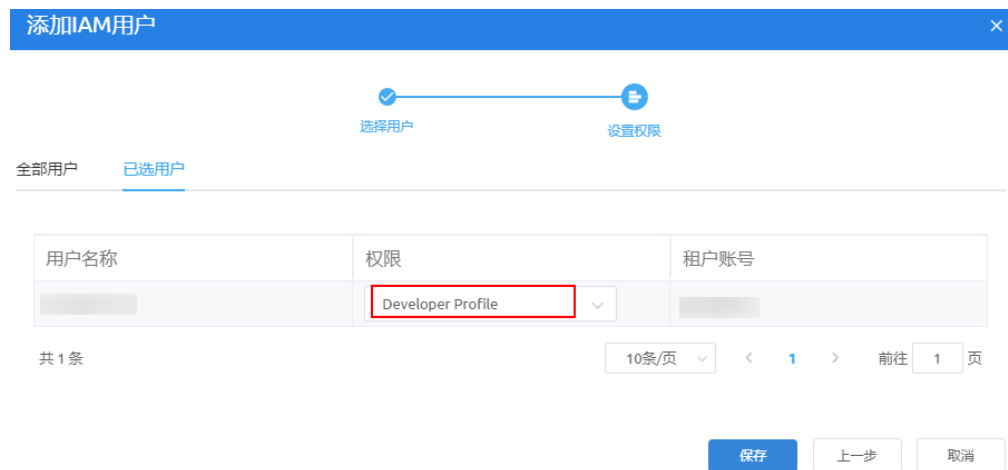
6. 在左侧导航栏中，选择“用户安全 > 用户”，单击“添加IAM用户”。
7. 在“添加IAM用户”页面，选中需要添加的用户，单击“下一步”。

图 2-33 添加 IAM 用户



8. 在设置权限中，选中Developer Profile或System Administrator Profile，为IAM用户添加开发者权限。
 - System Administrator Profile：管理员权限，默认开启全部权限。赋予该权限的用户为管理员用户，可添加其他开发者账号。为了保证应用开发的安全性，不建议赋予应用开发者System Administrator Profile权限。
 - Developer Profile：开发者权限，赋予该权限的用户为应用开发者，可以在AstroZero开发应用，如新增一个对象、为对象增加一个字段、新增一个流程等。

图 2-34 配置权限



9. 单击“保存”，返回用户列表页面。
在用户列表中，可查看到已添加的IAM用户。

步骤2 使用已添加的IAM用户账号，登录AstroZero开发环境。

1. 使用**步骤1**中添加的IAM用户，登录华为云。

图 2-35 IAM 用户登录



2. 单击管理控制台左上角的📍，选择区域，如华北-北京四。
3. 单击☰，选择“开发与运维 > Astro轻应用（原AppCube）”，进入AstroZero服务控制台。
4. 在主页中，根据需要进入对应的环境。

----结束

2.3 创建业务用户

业务用户是指访问在AstroZero中开发的应用的账号，即应用的使用者。业务用户的信息存储在AstroZero的标准对象PortalUser中。

业务用户拥有哪些能力

- 根据权限访问应用、查看导航条菜单。
- 根据权限查看应用的页面布局。
- 根据权限查看、创建、编辑和删除应用中的对象。
- 根据权限查看、编辑应用中的对象字段。
- 根据权限执行应用中，开发的服务编排、脚本和工作流。
- 根据权限访问在应用中，自定义的公共接口。

在运行环境新增业务用户




- 步骤1** 使用华为账号，登录[华为云网站](#)，在顶部导航栏右侧单击“控制台”，进入华为云控制台。
- 步骤2** 在左侧导航栏上方，单击，选择服务实例所在的区域项目。
- 步骤3** 单击，在查找框中搜索“Astro轻应用”，单击查找到的结果，进入AstroZero服务控制台。
- 步骤4** 单击“进入首页”，进入应用开发页面。
- 步骤5** 在“工作台”页签中，单击页面左上角的，选择“业务配置中心”，如[图2-36](#)，进入运行环境业务配置中心。

图 2-36 进入运行环境业务配置中心



步骤6（可选）创建角色权限，如果还未规划给业务用户赋予权限，可跳过该步骤。

📖 说明

购买AstroZero专业版实例时，才会显示“角色权限”菜单。

业务配置中心的角色权限都是基于权限“Portal User Profile”创建而来的。不同的业务用户权限可能有所不同，可通过配置扩展权限集给业务用户分配不同的权限。关于扩展权限集的更多介绍，请参见[设置扩展权限集](#)。

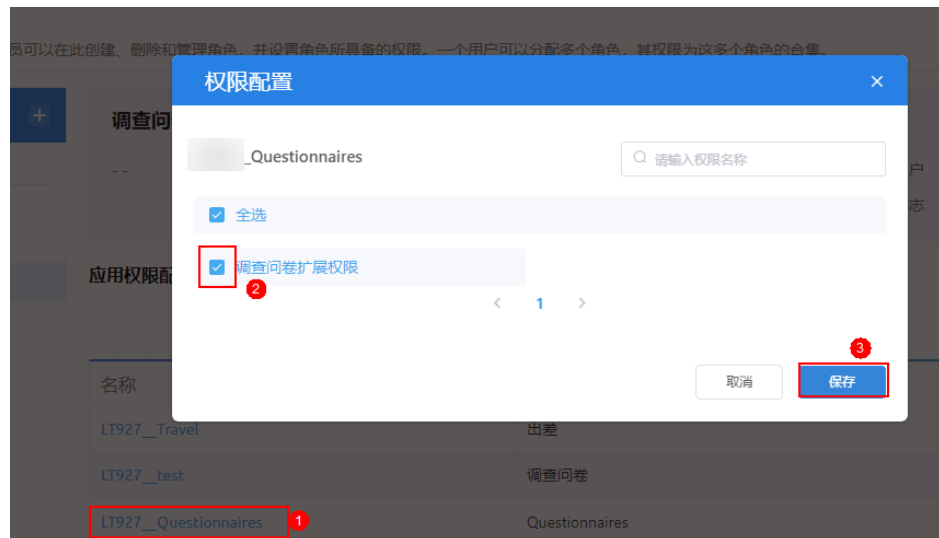
1. 在“业务配置中心”页面，选择“组织用户”。
2. 在左侧导航栏中，选择“角色权限”。
3. 单击“所有角色”后的“+”，设置添加的角色名称，例如“调查问卷权限”，并配置应用为可见和默认应用。

图 2-37 配置应用可见和默认



4. 给权限添加扩展权限集，该权限集是在应用中导入的扩展权限集。

图 2-38 添加扩展权限集



步骤7 创建业务用户，例如“testQuestionnaires”。

1. 在“业务配置中心”页面的主菜单中，选择“组织用户”。
2. 在左侧导航栏中，选择“用户管理”，单击“创建用户”。
3. 在新建用户页面，填写用户信息，用户名设置为“testQuestionnaires”。
4. （可选）为业务用户添加角色。
 - 如果未创建角色权限，“角色”区域可暂不指定权限，后续再对业务用户进行权限配置。
 - 如果已创建角色权限，在“角色”区域中从左侧勾选已创建的角色，如“调查问卷权限”，再单击“>”赋予该用户相应的权限。
5. 设置完成后，单击“保存”。

----结束

业务用户如何登录应用

步骤1 参考[如何登录经典版运行环境的环境配置](#)中操作，登录AstroZero运行环境管理中心。

📖 说明

免费版不提供运行环境，请参考[购买商用租户](#)购买标准版、专业版或专享版。

步骤2 开发者用户在运行环境管理中心的“系统管理 > 域名配置”中，查看访问域名并分享给业务用户。

业务用户通过获取到的运行环境的访问域名，即可登录应用。更多关于租户访问域名的介绍，可参见[管理域名配置](#)。

图 2-39 获取访问域名



步骤3 (可选) 创建业务用户 (如在运行环境新增业务用户) 时，在“角色”区域未配置业务用户权限，请执行此步骤。

📖 说明

购买AstroZero专业版实例时，才会显示“角色权限”菜单。

在运行环境安装完应用后，需要配置使用该应用的业务用户的权限，该权限需要具备应用中使用到的对象、接口、脚本等相关权限。这里以给调查问卷应用（应用的创建打包过程请参考[探索示例应用](#)）的业务用户配置权限为例进行说明。




1. 使用华为账号，登录[华为云网站](#)，在顶部导航栏右侧单击“控制台”，进入华为云控制台。
2. 在左侧导航栏上方，单击，选择服务实例所在的区域项目。
3. 单击，在查找框中搜索“Astro轻应用”，单击查找到的结果，进入AstroZero服务控制台。
4. 单击“进入首页”，进入应用开发页面。
5. 在“工作台”页签中，单击页面右上角的，选择“环境配置”，如[图2-40](#)，进入运行环境的环境配置。

图 2-40 进入运行环境管理中心




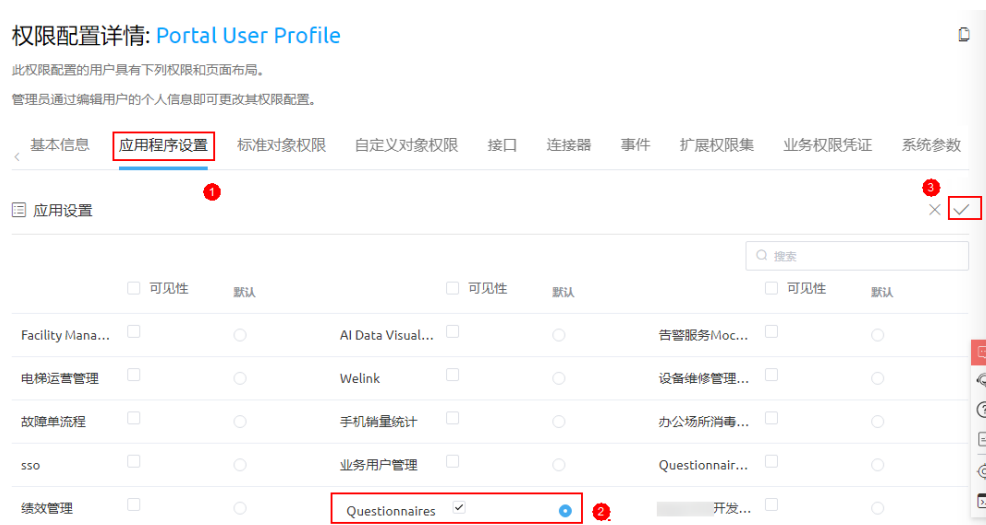
6. 在左侧导航栏中，选择“用户安全 > 权限”。
 7. 单击业务用户使用的权限，如“Portal User Profile”，进入权限详情页。
 8. 在“应用程序设置”页签，单击 ，设置调查问卷应用“Questionnaires”的“可见性”及“默认”。
- 设置后，拥有该权限的业务用户可查看到应用。


图 2-41 设置应用可见性及默认



9. 在“应用程序设置”页签，单击调查问卷应用“Questionnaires”，设置菜单可见性。

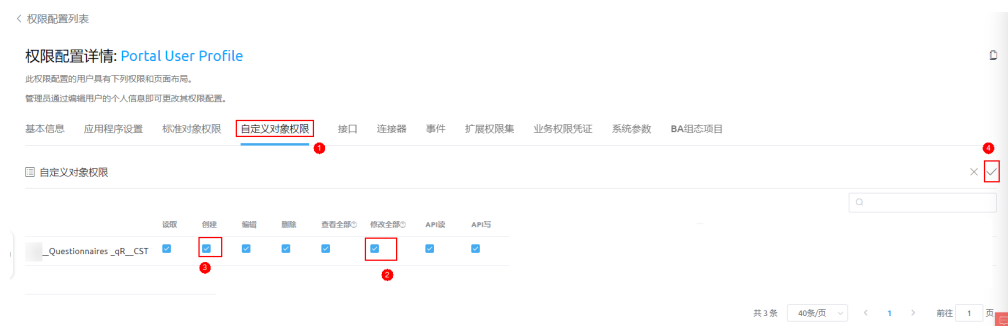
图 2-42 菜单可见性设置



10. 在“自定义对象权限”页签，单击 ，在搜索框中输入“Questionnaires”，然后勾选对象（命名空间 _Questionnaires_qR__CST）的所有权限，并在提示弹窗中单击“确定”。

设置后，拥有该权限的业务用户可填写并提交问卷调查。

图 2-43 自定义对象权限



- 步骤4** 业务用户在浏览器中输入运行环境子域名，即可访问业务应用的登录页，输入账号和密码登录应用。

图 2-44 登录应用



步骤5 业务用户首次登录需要重置密码，重置后使用新密码再次登录应用Portal运行页面。

步骤6 登录体验应用后，如果应用非默认应用，可单击应用前的图标切换应用。

图 2-45 切换应用



----结束

2.4 认识 AstroZero

首次使用AstroZero，可以通过阅读本入门，初步了解AstroZero。建议您跟随文档，创建实例应用，在实操的过程中逐步学习AstroZero。

环境分类

使用AstroZero进行应用开发，一般分为**开发->测试->上线**三个阶段，每个阶段分别对应**开发环境**、**沙箱环境**和**运行环境**。独立的环境可以实现各阶段数据隔离，保证正式上线版本稳定，更多环境的详细介绍，请参见[基本概念](#)。

图 2-46 应用开发



进入 AstroZero 开发环境

开通AstroZero服务后，可以使用华为账号，登录[华为云网站](#)，从控制台进入AstroZero开发环境。

- 步骤1** 使用华为账号，登录[华为云网站](#)，在顶部导航栏右侧单击“控制台”，进入华为云控制台。
- 步骤2** 在左侧导航栏上方，单击📍，选择服务实例所在的区域项目。
- 步骤3** 单击☰，在查找框中搜索“Astro轻应用”，单击查找到的结果，进入AstroZero服务控制台。

图 2-47 搜索 Astro 轻应用



- 步骤4** 在AstroZero服务控制台实例页面，单击“进入首页”或“旧版入口”，即可进入AstroZero开发环境。

📖 说明

- 免费试用版本无运行环境权限，请以实际界面为准。
- 在AstroZero中，开发低代码应用时，可以根据自身喜好，选择进入经典设计器或新版设计器。

----结束

新版首页说明











在AstroZero服务控制台中，单击“进入首页”，进入AstroZero新版首页。

图 2-48 AstroZero 新版首页



表 2-3 功能区域说明

区域	说明
工作台	<p>AstroZero零代码应用构建平台入口。</p> <p>AstroZero零代码应用构建平台，无需任何编程语言，只需通过拖、拉、拽，即可快速搭建应用程序，畅快创作，量身定制，让想法秒变应用。更多关于零代码的介绍，请参见《用户指南（零代码）》。</p>

区域	说明
应用开发	<p>低代码应用开发页面，在该页面可以快速构建多种类型的低代码应用。</p> <ul style="list-style-type: none">  主页：在主页中，可快速构建多种类型的应用，并直接在线发布使用。 <ul style="list-style-type: none"> 低代码应用：呈现当前用户最近创建、访问和收藏的应用。 学习中心：提供了AstroZero的功能操作和介绍视频，方便您更直观地了解AstroZero。 模板中心：预置了AstroZero零代码和低码应用模板，可以直接使用这些预置的应用模板快速创建应用。 产品动态：AstroZero通知公告/升级公告。  应用：在应用页面，可开发并管理所有低代码应用。低代码平台提供了经典设计器和体验新版设计器两种应用开发界面，更多介绍请参见《用户指南（低代码）》。  资产：在资产页面，可管理环境中的所有组件资产，包括可复用的BO资产、传统语言开发的原生服务和BO2.0资产。  业务大屏：Astro Canvas大屏应用开发入口，Astro Canvas是AstroZero提供的可视化大屏、小屏页面构建服务。Astro Canvas提供了丰富的可视化组件、灵活的数据接入和AI识别页面布局构建能力，支持多屏适配，帮助开发者快速构建和发布专业水准的实时可视化大屏、小屏页面。更多关于Astro Canvas的介绍，请参见《用户指南（Astro Canvas）》。  学习中心：提供了AstroZero的功能操作和介绍视频，方便您更直观地了解AstroZero。  模板中心：AstroZero内置了丰富的零代码和低码应用模板，您可以直接使用这些预置的应用模板快速创建应用。
右上角区域	<ul style="list-style-type: none"> ：微信扫一扫，可关注AstroZero公众号。 ：单击可切换语言。 ：华为云消息中心入口。消息中心是华为云为更好的建立与客户的联系而建设的服务平台，可以为用户提供各类通知消息的接收和管理，用户可以及时快捷的接收消息，从而避免信息遗漏造成不必要的损失。 ：AstroZero帮助文档入口。 用户名：显示当前登录账号的名称，单击账号名称，可查看系统版本信息、退出系统等。

经典首页说明




在AstroZero服务控制台，单击“旧版入口”，进入经典版开发环境首页。

图 2-49 经典版开发环境首页



表 2-4 经典版开发环境首页

区域	域说明
首页	<ul style="list-style-type: none"> 为您推荐：为您推荐有价值的内容。 项目：项目包括轻应用、行业应用、业务大屏和移动小程序四类，详细分类说明请参见应用场景。“我的应用”中，展示了正在开发的所有应用项目。 库：显示租户订购并安装的资产，包括应用App、Addon应用、BO、Native Service等，这些应用类型为资产包，内部组件受保护。 学习中心：提供一些AstroZero的功能操作和介绍视频，方便您更直观的了解AstroZero。 实战演练：供您进行实际演练，解锁后，可按照提示进行操作，完成典型应用的开发。
管理	租户层级的管理配置能力，例如用户权限管理、公司设置、数据管理等，更多介绍请参见 应用环境配置 。
资产中心	ROMA资产中心入口。

区域	域说明
右上角区域	<ul style="list-style-type: none">工单：我的工单和新建工单入口，当遇到问题时请提交工单联系华为云客服为您解答。简体中文：单击可切换语言。用户名：显示当前登录账号的名称，单击账号名称，可切换环境、查看系统版本信息、退出系统等。：AstroZero通知公告/升级公告页面入口。在该页面中，可以查看当前华为云官网上线的AstroZero版本号及各个版本上线/优化的特性。 说明 此处展示的版本为AstroZero正式商用版本，商用版本包括免费版、标准版、专业版和专享版。登录环境一致，仅功能规格不同，详细规格及收费标准请参见规格说明。：华为云消息中心入口。消息中心是华为云为更好的建立与客户的联系而建设的服务平台，可以为用户提供各类通知消息的接收和管理，用户可以及时快捷的接收消息，从而避免信息遗漏造成不必要的损失。：AstroZero帮助文档入口。

经典低代码应用设计器

图 2-50 经典低代码应用设计器

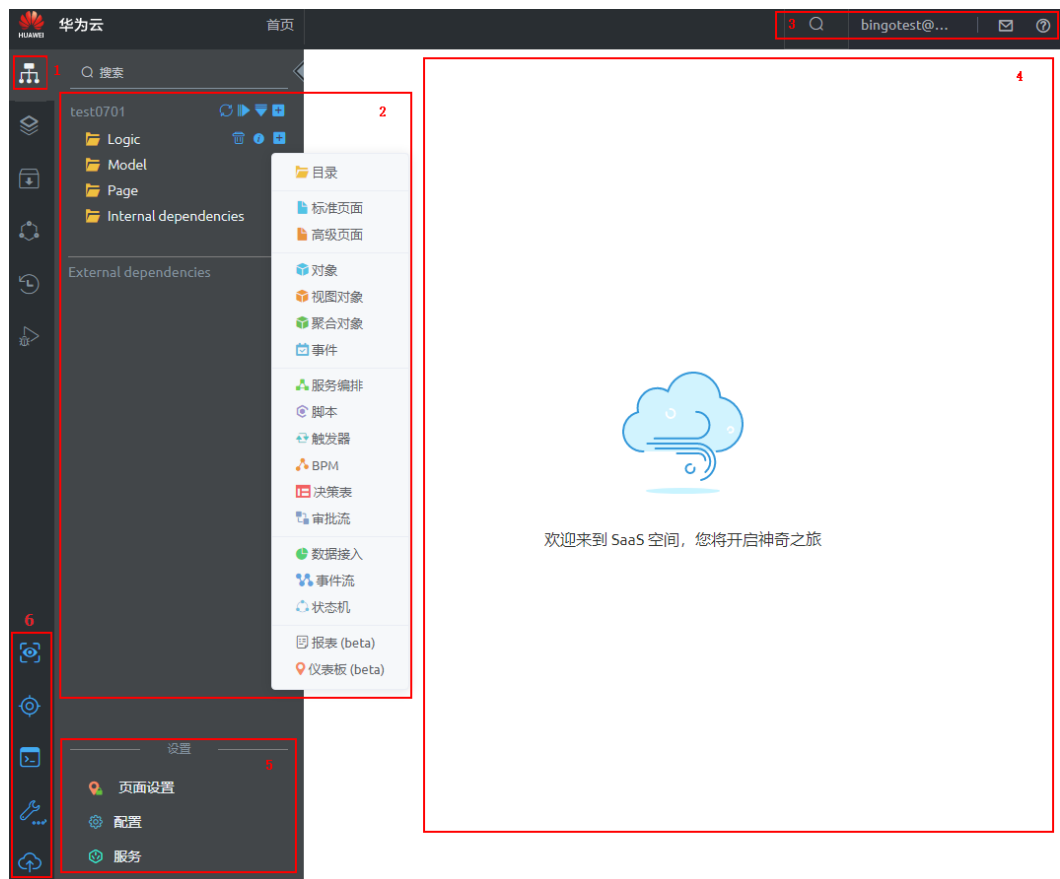


表 2-5 应用开发界面功能说明

区域	说明
1	以单个应用为维度的项目视图，可以看到一个应用下所有的开发内容，即区域2所示。
2	可以删除或新建文件夹对开发的内容进行分类管理，依赖区域放置该应用中所依赖的BO服务和Native Service。 单击文件夹后的加号，可开发这个App的数据模型、服务编排和页面等。也可以同时打开多个标签页，例如数据模型、服务编排、脚本页面。
3	<ul style="list-style-type: none"> 提供资源搜索功能：以关键字搜索该租户下所有资源的名称或元数据内容，同时支持高级搜索，可按照资源类型（例如脚本、高级页面等）进行搜索。搜索结果支持实时刷新（需要先单击“更新搜索索引”，来更新索引库，如果数据量大，更新索引库可能耗时较长，再以关键字搜索来刷新搜索结果。不建议短时间多次刷新，否则会影响系统性能）。例如，需要查看关联A对象所有资源，以对象名称为关键字，选择按照内容进行搜索。或者搜索包含某具体方法的所有脚本有哪些，可以用方法名作为关键字搜索所有脚本，搜索结果中会展示包含该方法的所有脚本。单击搜索结果中的具体某一脚本，可显示脚本的描述信息，以及包含方法的关键脚本代码。搜索结果最多展示300条。 提供帮助内容。
4	呈现出区域2或最左侧目录中所选择菜单的开发页面。 当在区域4打开的页面较多时，为了方便切换页面，可在区域3中单击“...”从下拉选项中选择页面进行切换。当区域4中，无打开菜单页面时，区域3中不显示“...”。
5	提供如下功能的快速入口： <ul style="list-style-type: none"> 页面设置：设置高级页面相关的配置，包括配置站点的运行态域名、站点首页，查看并管理当前应用已引用到的所有插件，组件/桥接器以及事件/动作、设置该站点页面在Web或者Mobile设备下的视图显示、设置站点级的变量等。 配置：配置应用的布局与颜色、导航条、选项列表、系统参数等。 服务：配置应用的自定义接口以及进行接口权限控制。
6	<ul style="list-style-type: none"> 预览所开发的应用。 是否开启日志跟踪。 是否打开数据调试界面。 编译应用、BO、原生服务，为发布做准备。 发布应用。





新版低代码应用设计器

图 2-51 新版低代码应用设计器



表 2-6 应用开发界面功能说明

序号	区域	说明
1	-	<ul style="list-style-type: none">• 主页：返回AstroZero应用开发页面的首页，即“应用开发 > 主页”。• 开发环境管理 > 环境配置：AstroZero开发环境的环境配置入口，主要提供管理应用、配置应用以及部署/运行应用功能。• 开发环境管理 > 业务配置中心：AstroZero开发环境的业务配置中心入口，业务配置中心将应用配置过程进行精简，便于应用管理员快速管理权限、配置应用资源。
2	应用管理区	<p>在应用管理区，可修改应用图标和描述、切换应用版本。</p> <ul style="list-style-type: none">• 修改应用图标和描述：可修改应用名称、应用描述、应用图标和图标颜色。• 切换版本：切换应用版本，该功能即将上线。

序号	区域	说明
3	生命周期区	<p>在生命管理区，可直接运行应用、一键部署应用、发布应用和检查应用的兼容性。</p> <ul style="list-style-type: none"> 运行：运行所开发的应用。 <ul style="list-style-type: none"> 立即运行：在当前开发环境，预览开发中的应用。 在本地沙箱中运行：在本地沙箱中运行当前开发中的应用，与本地系统做集成联调测试。 发布：应用开发好后，可发布应用。 <ul style="list-style-type: none"> 一键部署：打包、安装、运行、部署一步到位。 发布版本：应用变更后，可发布版本方便管理。 安装包：适用于初次开发或应用有较大变动时打包。 补丁包：适用于问题修复类打包。 发布设置：设置应用打包的相关参数，发布应用或打包应用时将会使用这些设置。 检查：应用编译检查，用于检查应用中的各项资源是否配置正确或启用，如果有错误信息，会展示在界面上，修改后才可发布应用。
4	系统区	<p>在系统区，可切换到经典开发模式、切换亮色/暗色模式，以及查看服务的用户指南等。</p> <ul style="list-style-type: none"> 经典模板：切换新版低代码应用开发工作台为老版界面。 ：为了方便同一个开发团队成员之间，能够更好地配合项目开发工作。低代码平台支持不同应用由不同团队开发，区分开发者对应用的不同权限，来控制开发者对应用的查看、开发、管理等操作。 ：服务产品资料文档入口。 ：开发者可以切换应用设计器的模式为“亮色模式”或“暗色模式”，也可以查看AstroZero发给华为账号的系统消息。 ：选择“关于”，可查看当前AstroZero的版本等信息。单击“退出登录”，可退出登录，并返回华为云登录页。

序号	区域	说明
5	导航区	<p>在导航区，可切换各个视图进行开发，包括页面、数据对象、业务逻辑、流程和集成开发视图等。</p> <ul style="list-style-type: none"> ● 开始：开发一个应用的向导页，开发者通过开始视图可快速创建常用工程元素，如页面、流程（工作流或定时任务）、报表、数据对象等常用元素，可设置应用导航。 ● 页面：存放应用的标准页面与报表的目录，展开可新建标准页面、高级页面和报表。 ● 数据：存放应用对象、视图对象和事件的目录，展开可新建对象、视图对象、事件。单击下方设置中的“选项列表”，可展示应用的选项列表项。 ● 逻辑：存放应用编排、脚本和触发器的目录，展开可新建编排、脚本和触发器。单击下方设置中的“系统参数”，可新建或导入系统参数。单击“自定义错误码”，可自定义错误码。 ● 流程：存放应用工作流和定时任务的目录，展开可新建工作流和定时任务。 ● 集成 <ul style="list-style-type: none"> - 开放接口：存放应用的自定义公共接口。 - 消息接入：存放应用的数据接入。 - 连接器：存放应用涉及的连接器，用于对接第三方服务。 ● 设置：应用设置入口，提供应用信息设置、外观设置和应用主题管理等功能。
6	工具区	<p>工具区集中呈现了搜索、调试、命令行、代码管理等辅助功能，更加方便开发者使用。</p> <ul style="list-style-type: none"> ● 搜索：提供资源搜索功能，以关键字搜索该租户下所有资源的名称或者元数据内容。支持高级搜索，可按照资源类型（例如脚本、高级页面等）进行搜索，搜索结果支持实时刷新（需要先单击“更新搜索索引”，来更新索引库。如果数据量大，更新索引库可能耗时较长，可以关键字搜索来刷新搜索结果）。例如，需要查看关联A对象所有资源，可以以对象名称为关键字，选择按照内容进行搜索，或者搜索包含某具体方法的所有脚本有哪些，可以用方法名作为关键字搜索所有脚本，搜索结果中会展示包含该方法的所有脚本，单击搜索结果中的具体某一脚本，可显示该脚本的描述信息，以及包含该方法的关键脚本代码，搜索结果最多展示300条。 ● 调试：是否开启日志跟踪。 ● 命令行：进入开发者控制台，可执行部分SQL命令收集信息，便于定位问题。 ● 代码管理：该功能即将上线。

2.5 零代码应用开发示例

场景描述

通过使用AstroZero的零代码应用模板，快速创建零代码应用，带您了解在AstroZero中开发一个零代码应用的大致流程，并认识相关概念。本入门以AstroZero零代码中预置的“培训报名”模板为例进行介绍，该模板提供如下功能：

- 外部用户即匿名用户微信扫码，填报活动报名信息，支持查看自己填写的报名信息。
- 内部用户可在手机端微信扫码或者WeLink扫码填报信息，支持在PC端新窗口打开或使用链接在浏览器访问填报页，支持查看自己填写的报名信息。
- 培训活动管理员基于用户的填报信息，获取培训报名人员信息，以及用户希望了解的培训内容。

前提条件

如果需要将填报页分享给内部用户进行填报，需要提前创建内部用户并分配权限。

说明

如果内部用户为微信或企业微信用户，可直接扫码绑定AstroZero，将微信或企业微信用户绑定为AstroZero内部用户。


1. AstroZero开发者用户（以华为账号为例）登录AstroZero服务控制台，单击“进入首页”。
2. 在主菜单中，选择“工作台”，进入零代码开发的工作台。
3. 单击页面左上角的，选择“运行环境管理 > 业务配置中心”，进入运行环境业务配置中心。

图 2-52 进入业务配置中心



4. 在“组织用户”页签下，选择“用户管理”，单击“创建用户”。
在用户管理页面，您可以查看到已创建的所有业务用户。
5. 填写用户信息，在“角色”区域中从左侧勾选角色再单击“>”，赋予业务用户相关权限，如“企业成员”权限，该权限不可编辑和删除应用，可填写表单数据提交表单。
如果需要编辑已有业务用户，可单击具体的业务用户名，进入详情页，单击“编辑”，设置角色。相关权限的说明，请参见[应用角色分类](#)。

图 2-53 编辑角色

角色信息



操作步骤

步骤1 进入零代码应用开发平台。

1. 使用开发者账号登录[华为云网站](#)，在顶部导航栏右侧单击“控制台”，进入华为云控制台。
2. 在左侧导航栏上方，单击📍，选择服务实例所在的区域项目。
3. 单击☰，在查找框中搜索“Astro轻应用”，单击查找到的结果，进入AstroZero服务控制台。
4. 在AstroZero服务控制台，单击“进入首页”。
5. 在主菜单中，选择“工作台”，即可进入零代码应用开发工作台。

步骤2 通过模板创建应用。

1. 在“全部应用”区域，单击“开始创建”，选择“使用模板创建”。

图 2-54 选择“使用模板创建”



2. 根据需求选择预置的模板，本入门选择“培训报名”，即单击培训报名模板。
3. 在培训报名模板中，单击“安装模板”，安装完成后即可进入应用编辑页面。

图 2-55 应用编辑页面



培训报名模板由“报名信息记录页”、“效果预览”和“使用说明”三个页面组成。

- 报名信息记录页：管理员和填报用户使用的是“报名信息记录页”，该页面有“全部信息”和“我的报名”两个视图。
 - “全部信息”视图可查看到所有填报用户填报的信息，管理员可通过该视图了解所有报名信息。
 - “我的报名”视图仅显示当前用户的报名填报信息，填报用户可通过该视图查看自己的填报信息。
- 效果预览：用于预览培训报名应用效果。
- 使用说明：培训报名模板的使用说明。

思考：如何控制不同的视图显示不同的数据？

解惑：选中视图，单击 ▾，选择“编辑视图”，在“数据范围”中可设置显示数据的过滤条件，还可设置数据排序方式和隐藏字段。

图 2-56 编辑视图



图 2-57 设置数据范围



步骤3（可选）当需要基于已有的模板页面进行定制时，可执行此步骤。


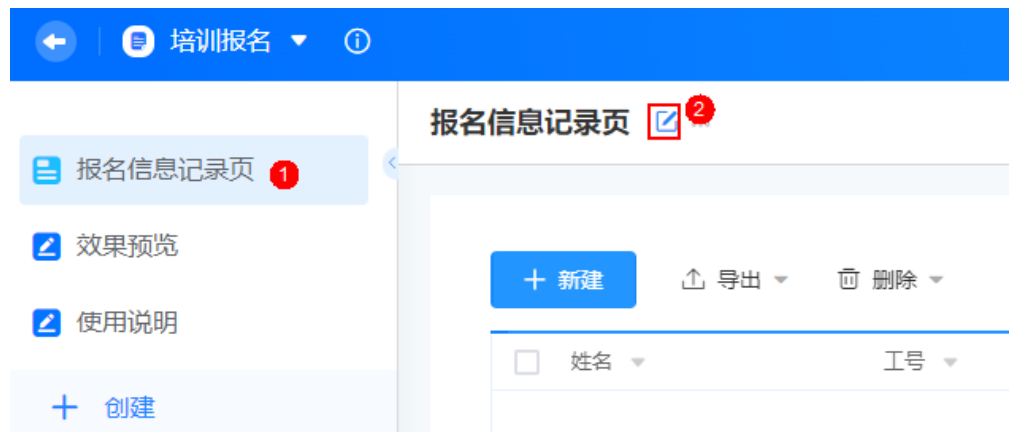
1. 如图2-58所示，选择需要定制的页面，单击编辑表单图标.

图 2-58 编辑表单



2. 从左侧组件库中，拖拽所需组件（如“邮箱”）到页面编辑区域并设置组件属性。

图 2-59 设置组件



3. 单击页面上方的“保存”。
4. 单击页面左上方的“< 报名信息记录页”，返回应用开发页面。

步骤4 在报名信息记录页，单击“分享问卷”，分享填报页给用户填报。


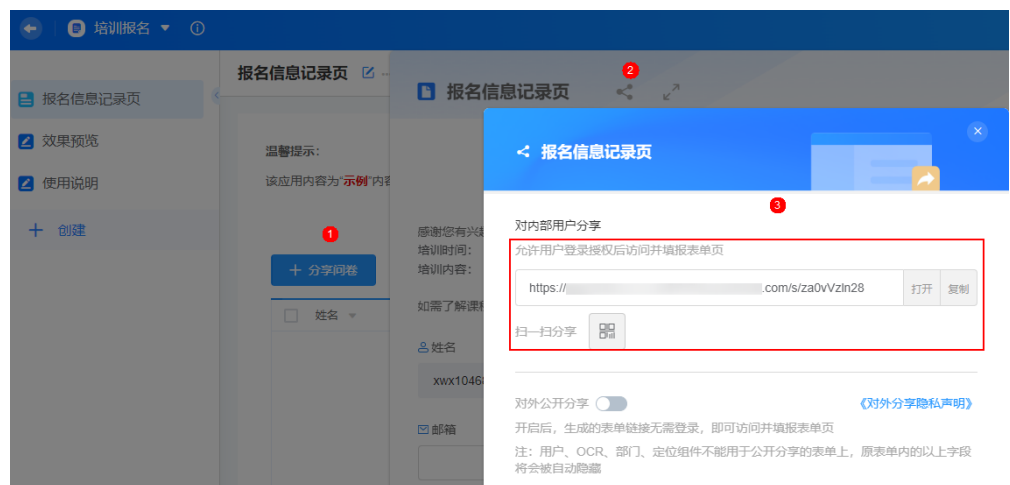
- 对内部用户分享：分享该页面给内部用户进行访问。分享后，内部用户输入用户名和密码登录后，可填报数据。
 - 打开：在新窗口打开填报。
 - 复制：复制链接分享给用户填报。
 - ：普通二维码扫码填报（支持WeLink扫码、微信扫码和企业微信扫码）。


图 2-60 对内部用户分享



- 对外公开分享：分享给外部用户进行访问。支持匿名用户访问，匿名用户无需登录，即可访问并填报表单页。

图 2-61 对外分享表单



- 打开：在新窗口打开填报。
- 复制：复制链接分享给匿名用户填报。
- ：普通二维码扫码填报（支持WeLink扫码、微信扫码和企业微信扫码）。

说明

微信或企业微信可直接扫码绑定AstroZero，将微信或企业微信用户绑定为AstroZero内部用户。

步骤5 管理员在零代码应用“报名信息记录页”中，获取培训报名人员信息，以及用户希望了解的培训内容。

选中全部记录单击“导出”，可获取全部报名信息。

图 2-62 导出填报数据



步骤6（可选）创建统计页面，用于查看、分析填报数据。

1. 在报名信息记录页的编辑页面，单击“查看统计”。

图 2-63 查看统计



2. 之前没有创建过统计页面，会显示如图2-64所示，单击“点击创建”。

图 2-64 创建统计报表入口



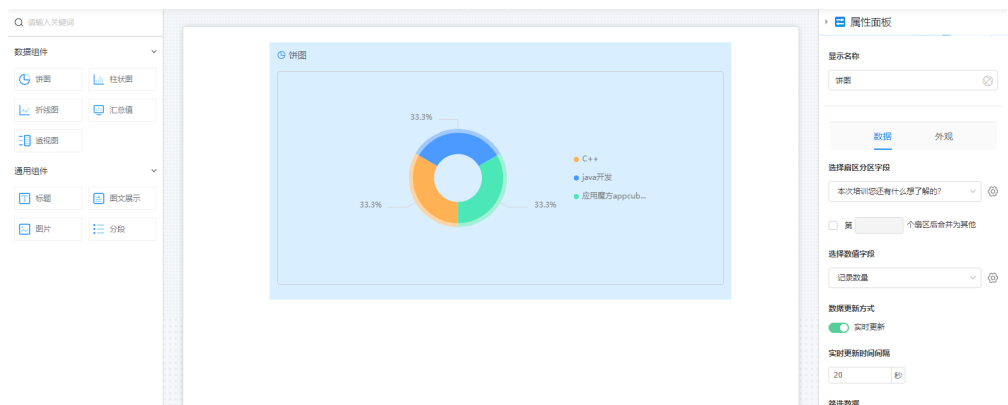
您还未创建数据报表，[点击创建](#)

3. 从左侧选择合适的组件类型拖入到中间区域，并设置组件属性。
默认已拖入饼图组件，参考图2-65在“属性面板”中配置相关属性。例如，根据填报者想了解的培训内容进行统计。

说明

鼠标悬浮在组件上，出现复制组件图标，当页面组件数大于一个时，还会出现删除组件图标。

图 2-65 设置组件属性



4. （可选）单击右侧“数据范围”，可设置统计数据的范围。
5. 单击页面上方的“保存”，保存设置。
6. 单击页面左上角的“< 数据统计1”，返回应用开发页面。
7. 单击“查看统计”，可查看统计数据。

----结束

2.6 交通管理大屏开发示例

场景说明

交通管理大屏用于监控城市的交通状态，如进出城车辆、路况信息和交通指数等，效果如图2-66所示。本入门以使用Astro Canvas中预置模板构建一个交通管理大屏为例，向您介绍如何在Astro Canvas中快速构建可视化页面。

图 2-66 交通管理大屏效果



在Astro Canvas中，构建交通管理大屏的步骤如下所示：

- **步骤一：创建大屏&PC端项目**
- **步骤二：构建可视化大屏页面**
- **步骤三：发布大屏项目**
- **步骤四：安装大屏项目**

前提条件

- 已参考[如何购买Astro Canvas实例](#)中操作，购买Astro Canvas实例。
- 已参考[如何登录Astro Canvas](#)中操作，登录Astro Canvas界面。

步骤一：创建大屏&PC 端项目

在新建大屏页面前，需要先创建一个项目。一个项目对应一个站点，在该站点中可以创建多个页面。

步骤1 在“项目列表”中，单击“+ 新建项目”。

图 2-67 进入新建项目页面



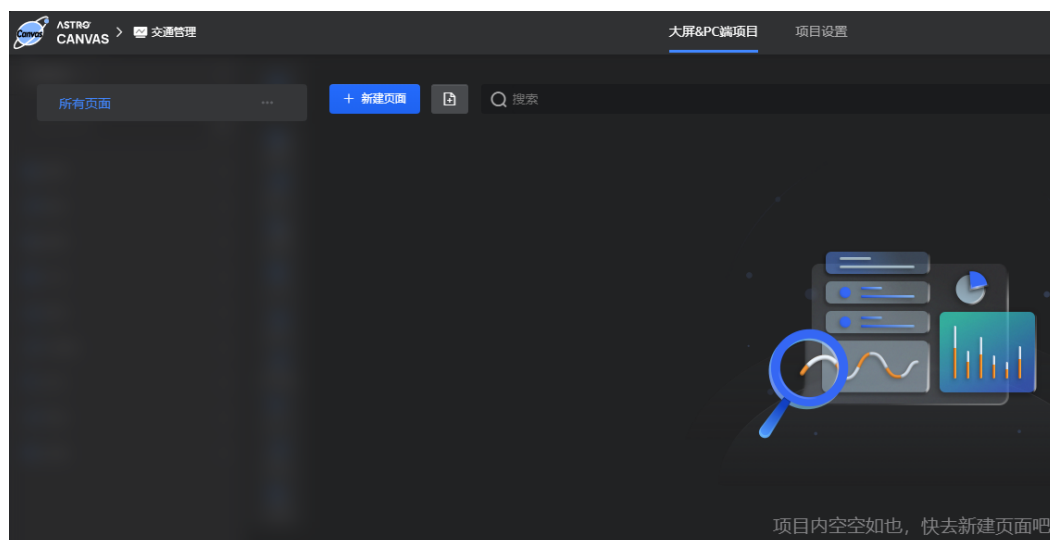
步骤2 在新建项目页面，选择“大屏&PC端”，输入项目名称（如交通管理），单击“新建”。

图 2-68 新建大屏&PC 端项目



系统自动进入该新建项目的“大屏&PC端项目”页面，如图2-69。

图 2-69 大屏&PC 端项目页面



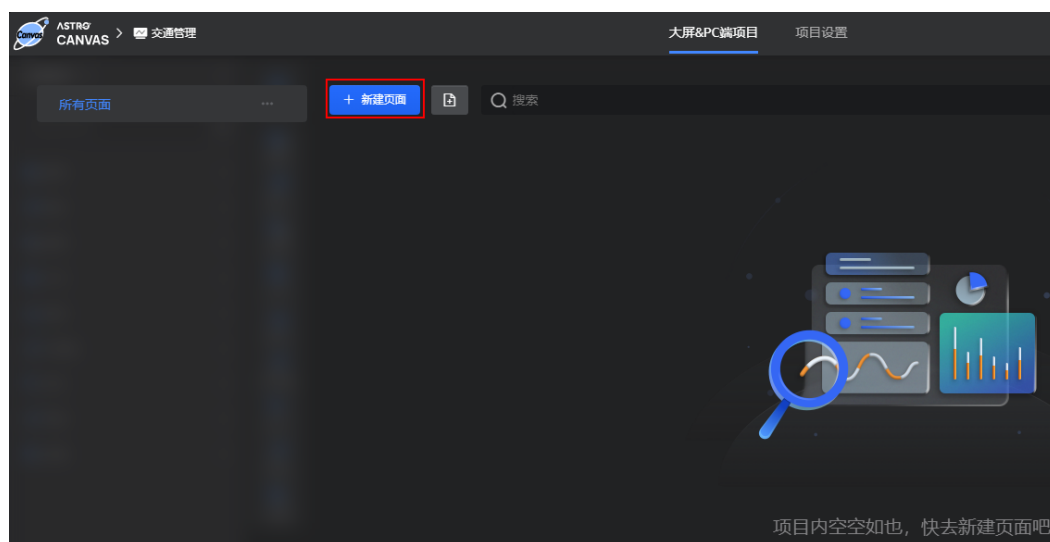
----结束

步骤二：构建可视化大屏页面

Astro Canvas中预置了一些可复用的页面样例作为模板（预置模板和自定义模板），通过模板创建新的页面，可充分利用已有资源，减少重复开发，提高交付效率。本入门以交通管理模板为例，向您介绍如何快速构建大屏页面。

步骤1 在“大屏&PC端项目”页面，单击“新建页面”。

图 2-70 单击新建页面



步骤2 在预置模板中，选择“交通管理”，单击“使用此模板”。

图 2-71 选择交通管理模板



步骤3 输入页面标题（如A市交通管理分析大屏），单击“新建”。


图 2-72 设置页面标题



系统自动进入大屏开发页面，如图2-73。您可以直接使用该大屏页面，也可以根据业务需求进行二次开发。如何从零开始创建一个大屏应用，请参考[开发大屏页面](#)。

图 2-73 大屏开发页面




步骤4 单击页面上方的, 保存大屏页面。

----结束

步骤三：预览并发布大屏页面

在Astro Canvas中开发大屏页面时，支持直接在线预览效果，确保开发编辑效果与发布后运行效果一致。预览效果符合预期后，可直接发布大屏页面。

步骤1 单击页面上方的, 预览页面效果。

如果预览效果不符合预期，可进行微调，直至符合预期。


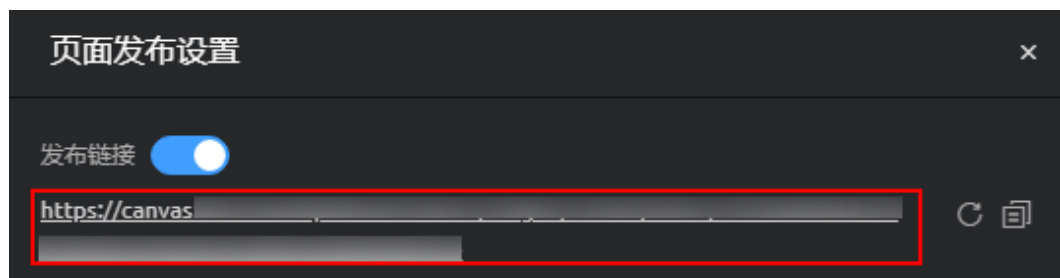


步骤2 单击, 打开“发布链接”开关，获取页面访问地址。

图 2-74 打开发布链接



单击, 可生成新的链接，原链接将不可访问。单击, 可复制该链接。

步骤3 (可选) 版本管理。

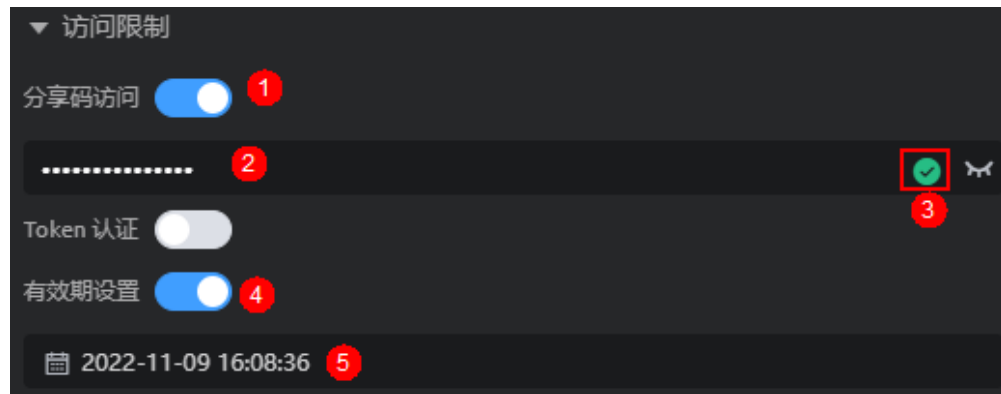
通过管理页面版本，设置发布具体哪个版本的页面或者回退到具体哪个版本的页面进行开发。更多介绍，请参见[管理页面版本](#)。

本示例不新建版本，使用默认的实时更新版本作为发布页。

步骤4 (可选) 分享码设置。

设置页面分享码后，只能通过分享码来访问页面，提高安全性。更多介绍，请参见[如何设置分享码](#)。

图 2-75 分享码设置




步骤5 (可选) 设置Token认证。

通过设置Token认证，可对大屏交互时传递的参数进行签名鉴权，保证大屏的URL访问链接的参数不会被篡改，从而提高大屏数据及用户信息的安全性。如何设置Token认证，请参见[如何设置Token认证](#)。

图 2-76 设置 Token 认证信息

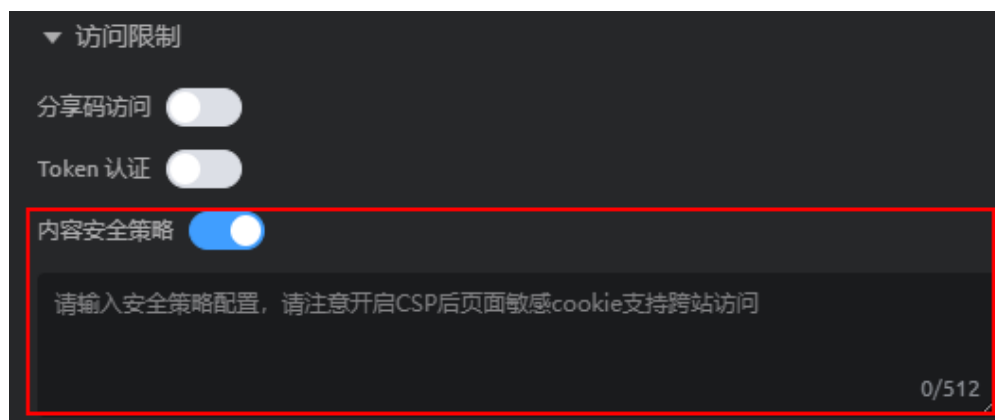


- : 打开Token认证开关后，Astro Canvas会生成一个Token。单击该图标，可复制Token。
- 签名地址只允许打开一次：开启后，签名地址仅允许打开一次，再次访问时需要重新获取签名地址。如果不开启，不限制访问次数，但支持设置有效期。
- 有效期设置：设置Token认证有效期。如果不设置有效期，Token认证将永久有效。

步骤6 (可选) 设置内容安全策略。

大屏页面嵌入第三方系统时（例如以iframe形式嵌入），如果涉及跨域访问，请开启内容安全策略。设置内容安全策略的主要目的是减少和报告XSS攻击，缓解跨站脚本攻击。除限制可以加载内容的域，还可指明哪种协议允许使用，缓解数据包嗅探攻击。如何设置内容安全策略，请参见[如何设置内容安全策略](#)。

图 2-77 设置内容安全策略



开启“内容安全策略”时，如果在浏览器无痕模式下预览页面，请确保“阻止第三方 Cookie”已关闭，否则无法正常预览页面。

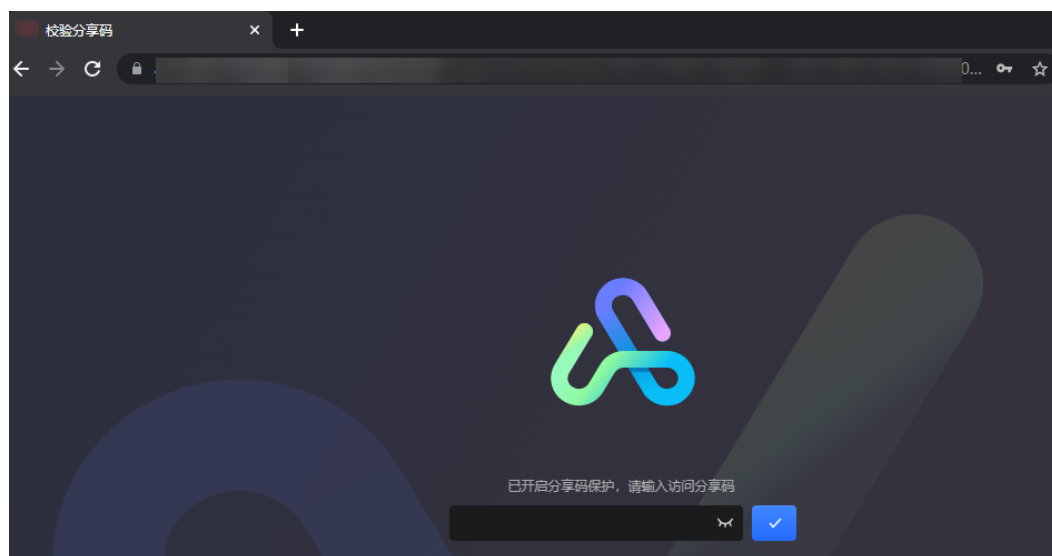
图 2-78 关闭阻止第三方 Cookie



步骤7 设置完成后，在浏览器中打开发布链接地址，即可访问大屏页面。

如果设置了分享码，输入分享码后，单击 ，即可访问大屏页面。

图 2-79 输入分享码



----结束

步骤三：发布大屏项目

项目开发完成后，可以直接在线预览，也可以一键打包从A环境发布、B环境安装，实现开发 > 测试 > 上线的快速部署，从而实现不同环境之间大屏资产的快速迁移。发布项目，支持将项目中的数据集数据一并进行打包。

步骤1 在项目列表页面，单击**步骤一：创建大屏&PC端项目**中创建项目上的，发布项目。

图 2-80 发布项目



📖 说明

发布项目前，请确保项目中的页面已发布。

步骤2 在版本管理中，单击“+ 新建版本”。

步骤3 设置版本号和版本描述（可选），勾选待发布的页面，选择页面版本，单击“确定”。

只有已生成发布链接的页面，才会显示在如图2-81所示页面中。

图 2-81 新建版本




步骤4 单击, 设置项目当前发布版本。

图 2-82 设置当前发布版本



步骤5 在弹出的提示框中，单击“确定”。


发布成功后，页面显示“版本发布成功”。

步骤6 设置项目访问限制。

图 2-83 设置项目访问限制



- 按照项目限制：项目中页面首次完成验证后，其余页面无需二次验证。例如，页面A、B、C设置了Token认证，页面D、E、F设置了分享码。勾选按照项目限制时，校验过页面A的Token认证后，不会再校验页面B和C。校验了页面D的分享码后，同样不会再校验页面E和F。
- 按照页面限制：项目遵循各页面访问限制。

步骤7 设置完成后，返回项目列表页面，单击项目上的，预览设置效果。

如果**步骤二：构建可视化大屏页面**中设置了分享码，输入分享码后才可以预览项目中的页面。

----结束

步骤四：安装大屏项目

在项目发布的版本管理中，可下载对应版本的项目包并在其他环境安装项目。例如，在当前环境下载项目包，在B环境安装。

步骤1 在项目列表页面，单击**步骤一：创建大屏&PC端项目**中创建项目上的.

图 2-84 进入项目发布页面




步骤2 在版本管理中，鼠标放在对应版本后的“操作”下，单击。

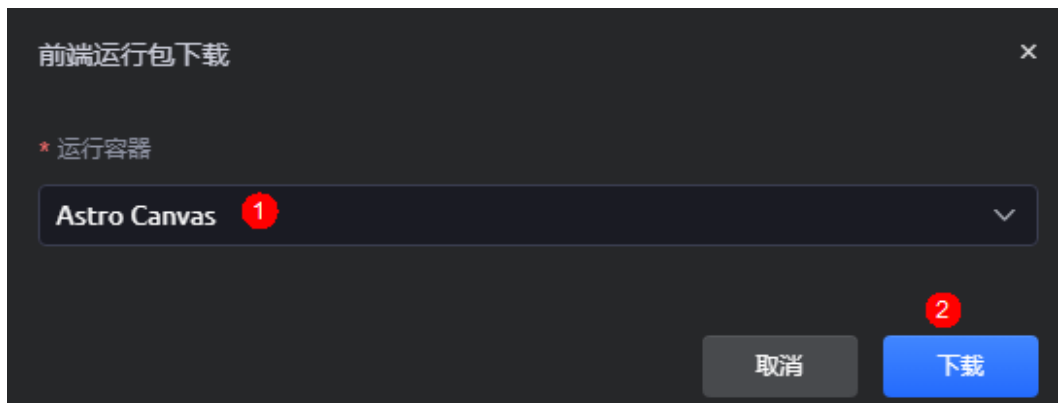
图 2-85 下载对应版本的项目包



步骤3 选择运行容器，本示例选择“Astro Canvas”，单击“下载”。

更多运行容器的介绍，请参见[安装项目](#)。

图 2-86 选择运行容器



步骤4 将下载到本地的项目安装包发送给其他需要安装的用户。

步骤5 其他用户在B环境的Astro Canvas界面，单击“导入项目包”。

图 2-87 导入项目包



安装成功后，在已安装的项目上，单击项目图标，即可进入项目。进入项目后，可根据业务需求进行二次开发。

----结束

2.7 探索示例应用

2.7.1 场景介绍

通过使用AstroZero的应用模板，创建一个示例应用（问卷调查应用），带您一步步探索并分析这个示例应用。在探索示例应用的过程中，您可以了解在AstroZero中开发一个应用的大致流程，并认识应用开发中的相关概念。

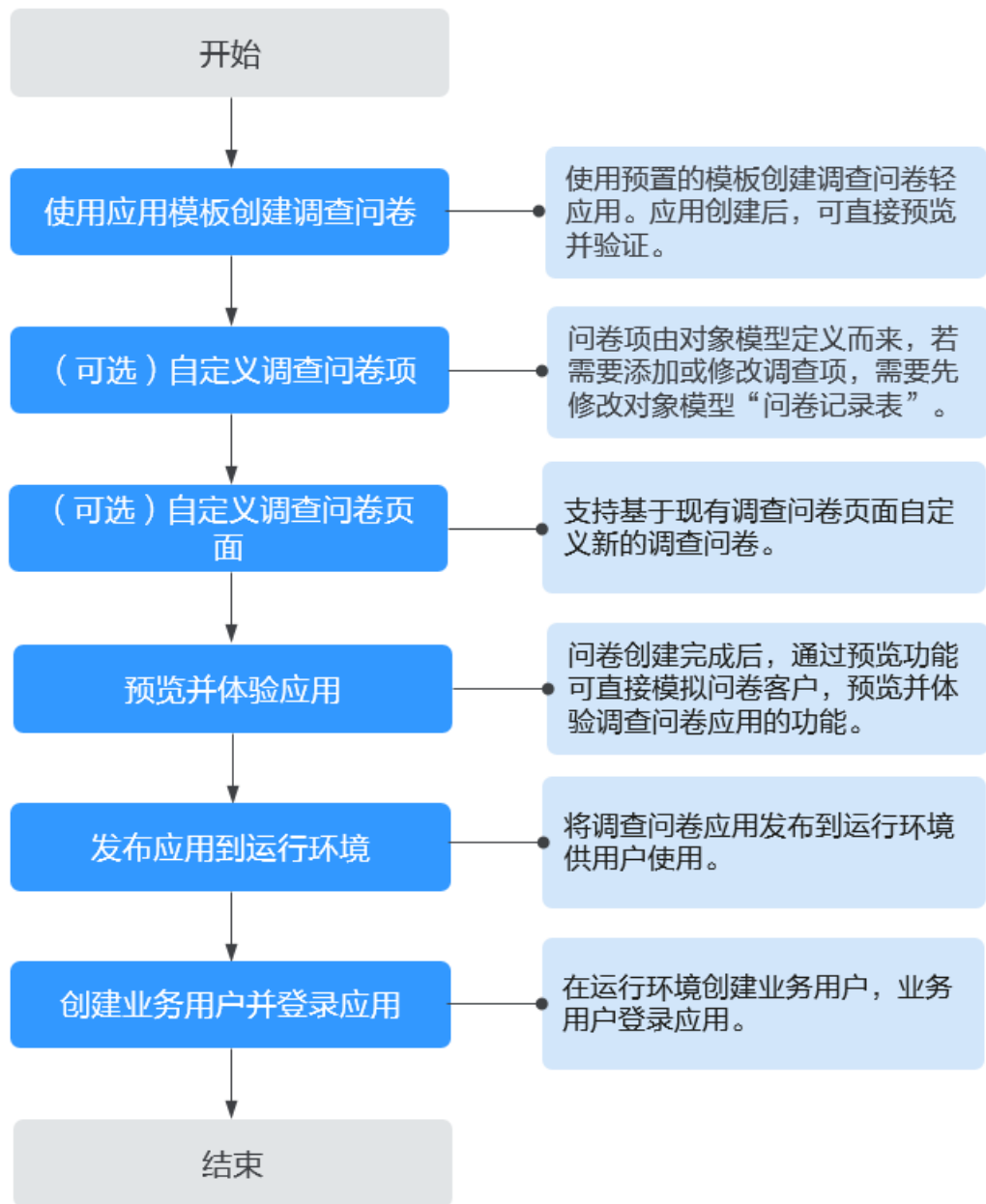
前提条件

已有一个可用的AstroZero账号，如果当前还未申请AstroZero账号，请参照[注册租户](#)中操作，申请公测，并获取AstroZero免费试用资格。

开发流程

问卷是一组与产品、社会活动等有关的问题，或者说是一份为进行调查而编制的问题表格，又称调查表。调研人员借助该调查表可对产品使用过程或社会活动过程进行准确、具体的测定，并应用社会学统计方法进行量的描述和分析，获取所需要的调查资料。在AstroZero中，使用“问卷调查”模板，创建调查问卷应用的流程如下：

图 2-88 示例应用开发流程



示例应用继承应用模板全部功能，创建后即是一个具备完整功能的应用，可直接进行预览、测试及发布，应用最终效果如图2-89所示。

图 2-89 问卷调查应用最终效果



2.7.2 步骤 1：使用应用模板快速创建调查问卷

轻应用一般为轻量级应用，不涉及复杂化的代码，本节以对开发者展开调查问卷为例，介绍如何使用“调查问卷”模板快速创建、配置以及发布调查问卷轻应用。

通过该应用，您可以了解AstroZero中的标准页面、公共接口、导航页面设置、对象模型等能力。

创建轻应用



- 步骤1** 使用华为账号，登录[华为云网站](#)，在顶部导航栏右侧单击“控制台”，进入华为云控制台。
- 步骤2** 在左侧导航栏上方，单击，选择服务实例所在的区域项目。
- 步骤3** 单击，在查找框中搜索“Astro轻应用”，单击查找到的结果，进入AstroZero服务控制台。
- 步骤4** 在AstroZero服务控制台，单击“旧版入口”，进入AstroZero经典版开发环境。
- 步骤5** 在AstroZero开发环境首页的“项目”页签下，单击“轻应用”。
- 步骤6** 鼠标放在“问卷调查”下的“调查问卷”模板上，会出现“查看模板”和“使用模板”图标，单击“使用模板”。

图 2-90 使用模板



说明

除了“问卷调查”模板，AstroZero还预置了“问卷调查[高阶]”模板。“问卷调查[高阶]”模板用于快速定制个性化调查问卷，发布匿名/企业问卷。如果想了解更多关于高阶模板的相关使用指导，可将鼠标放在“问卷调查”下的“问卷调查[高阶]”模板上，单击“查看模板”，再单击“查看说明文档”获取相关指导。

步骤7 （可选）首次创建应用时，您需要先定义专有的命名空间。在提示窗中，单击“创建”，输入“命名空间”，例如“CNAME”，再单击“创建”。

注意

命名空间一旦设定不能修改，且唯一，请慎重定义。建议使用公司或团队的缩写，作为命名空间。

后续创建的应用、应用元素（例如对象、脚本、服务编排、页面等）的名称前，系统会为其增加命名空间。例如，将命名空间定义为“CNAME”，后续创建的所有应用、应用元素（对象、脚本、页面等）都将以“CNAME_”开头。

图 2-91 输入命名空间



步骤8 命名空间创建完成后，在“创建轻应用”页面填写应用的标签“Questionnaires”，单击“创建”。

图 2-92 创建轻应用



表 2-7 创建轻应用页面参数说明


参数	参数说明
添加图标	为轻应用设置图标。
标签	轻应用的展示名称。
描述	轻应用的描述信息。
高级设置 > 运行时版本	开发的资产包依赖所选择的运行时版本，如果线下运行版本不一致，可能产生不兼容。

图 2-93 创建后的应用



说明

在应用开发页面，如果提示“您的租户没有配置默认邮件服务器地址，将会导致在服务编排、工作流、脚本中发送邮件功能不可用”，请直接忽略，本示例不涉及。

调查问卷轻应用继承了模板功能，即调查问卷是一个具备完整功能的应用，无需做任何修改，单击左侧，可直接模拟问卷客户，预览并体验调查问卷应用的功能，具体操作请参见[步骤4：预览并体验应用](#)。

如果想在模板应用的基础上，自定义应用的问卷项及页面，请执行（可选）[步骤2：自定义调查问卷项](#)、（可选）[步骤3：自定义调查问卷页面](#)。

----结束

了解应用

说明

【技巧】本小节旨在为您简要介绍应用中的目录结构、前端页面、数据对象相关概念知识，并不是创建应用中的必要步骤，如果想尽快体验创建应用，请跳过本节，直接阅读（可选）[步骤2：自定义调查问卷项](#)。

- **认识应用目录**

如[图2-94](#)所示，当前应用有4个默认目录。默认目录是AstroZero推荐使用的应用文件目录结构，轻应用在创建后，默认已自动创建应用目录，目录中的文件组成了完整的应用。后续需要开发其他应用时，这些元素则需要您自己设计并创建。您也可以根据自己的开发习惯，单击应用名右侧的“+”，添加新的应用目录，自定义放置应用资源，如[图2-95](#)。

图 2-94 应用目录及导航栏图标

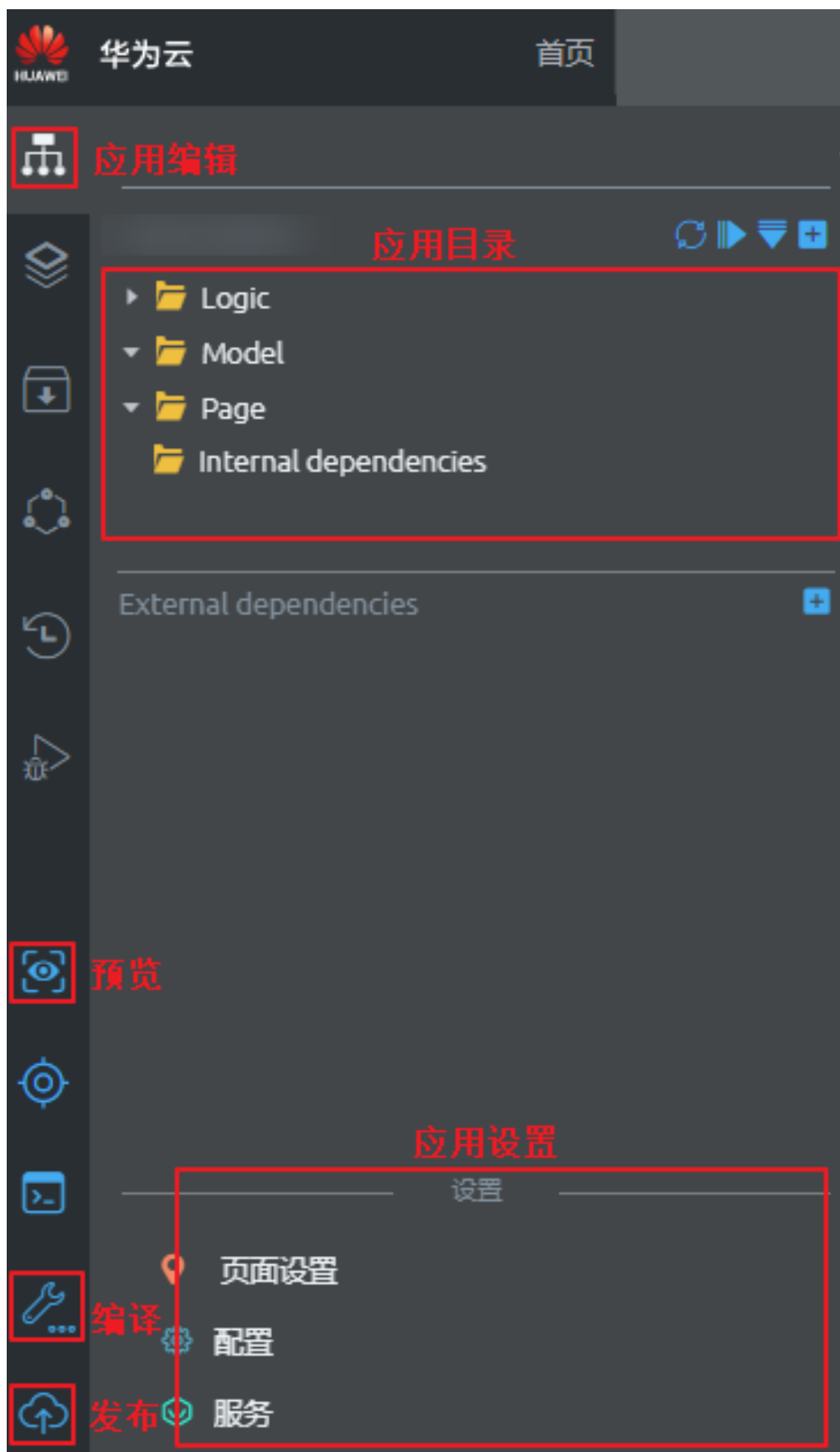


图 2-95 添加自定义应用目录

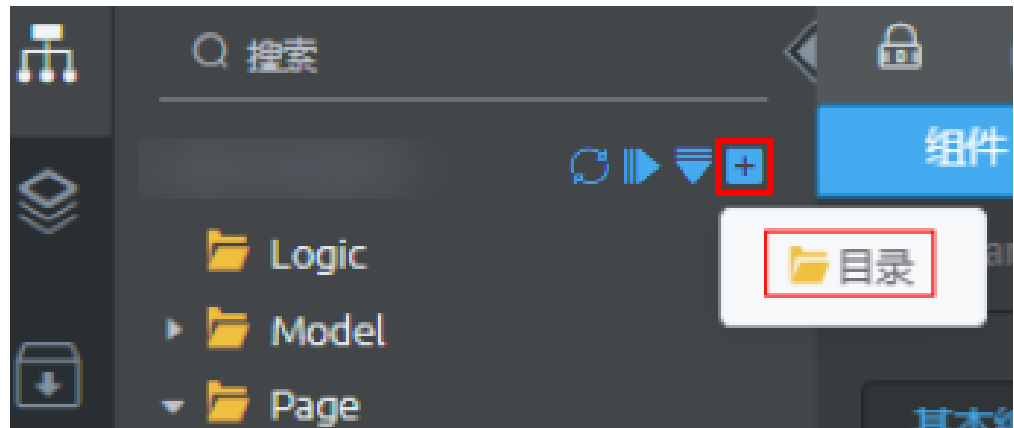


表 2-8 应用目录介绍

文件夹名	说明
Logic	用于存放业务中的后台逻辑，例如服务编排、脚本等。业务功能复杂时，需要创建后台逻辑完成。本应用中未涉及。
Model	用于存放数据对象。 数据对象相当于数据库中的一张表，根据业务功能不同，您可以在应用中创建各种数据对象，用来存储业务数据。本应用中的问卷记录对象“Questionnaires_qR_CST”，是用于保存前端页面获取的“问卷记录”中的数据，实际创建的数据对象名为“ CNAME _Questionnaires_qR_CST”，命名规则对应是“ 命名空间_应用名_qR_CST ”，单击数据对象名，即可查看或编辑数据对象。
Page	用于存放前端页面。示例应用中有3个前端页面，分别是问卷提交页、问卷记录管理页以及问卷提交成功页。其中页面名称命名规则对应是“ 命名空间_应用名_页面名 ”。
Internal dependencies	存放内部依赖文件，例如BO、原生服务，本应用中未涉及。
External dependencies	存放导入的外部依赖文件，例如BO、原生服务，本应用中未涉及，单击“+”可以添加BO或原生服务。

• 了解应用设置

- 页面设置：应用中页面的相关设置，本应用不涉及。
- 配置：用于配置应用中，各种应用公共参数（运营配置、导航条、布局与颜色、选项列表、系统参数等）。

本应用中，仅用到“导航条”的相关功能，即在导航条页签下，设置应用导航菜单。

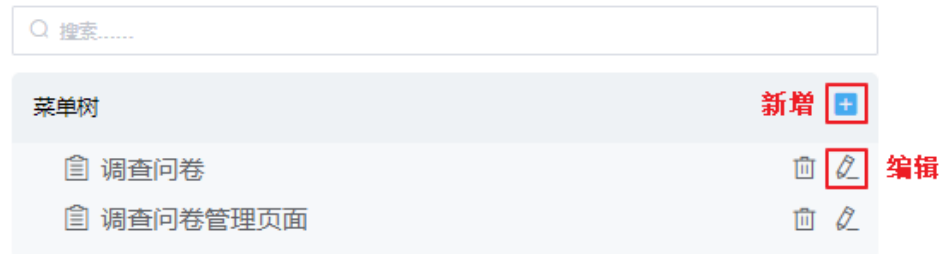
本应用中，预置了2个导航菜单页面，即问卷提交页、问卷记录管理页。后续开发其他应用或页面时，可以在“配置”中新增、编辑导航菜单。

图 2-96 配置应用导航

导航条

您可以点击添加按钮添加菜单、页签。

拖拽调整菜单项顺序。



- 服务：应用中服务的相关设置，本应用不涉及。


- 帮助信息

使用应用模板创建的应用，默认提供了帮助信息。即通过视频操作的方式，向您介绍如何在前端页面中新增问卷问题等。

图 2-97 模板帮助信息



说明

如果帮助信息未显示，请单击界面右下角 ，即可再次显示。

- 标准页面编辑区

首次进入应用，默认显示调测问卷首页，调查问卷首页是一个标准页面，关闭帮助信息后，可以看到完整的标准页面编辑区，如图2-98所示。

图 2-98 标准页面编辑区



标准页面是在AstroZero开发应用常用的页面形式，以调查问卷页面“命名空间 `Questionnaires_questionnairePage`”为例，带您初步了解前端页面、页面模型与后端数据对象之间的关系。

- a. 在调查问卷页面中，选中“表单”组件，右侧属性区域自动显示表单组件的相关属性，在“数据绑定”下，系统自动为“表单”组件绑定了一个“form_1”。
其中，“form_1”是一个页面对象模型，页面对象是在“模型视图”中定义的，本应用中默认已经创建。

图 2-99 选择表单组件查看组件属性



- b. 单击页面底部“模型视图”，可以查看到当前页面中的2个页面模型 form_1、editFlag。

说明

前端页面模型根据绑定的后端资源不同，分为自定义对象、对象模型、服务模型和事件模型四种，详细介绍请参见[标准页面](#)。

图 2-100 页面的模型视图



- form_1为页面对象模型，在当前页面中，form_1映射绑定了后端数据对象“命名空间_Questionnaires_qR_CST”。同时，在前端页面中form_1又被表单组件进行了数据绑定，当用户输入问卷信息后，单击“提交”按钮，触发按钮的“点击”事件，即可将这些数据存入到后端数据对象，完成数据传递，三者关系如图2-101、图2-102所示。

图 2-101 数据传到页面模型“form_1”最终存入到数据对象



图 2-102 页面查询时从数据对象查询获取展示到组件



- editFlag为页面自定义模型。
当用户提交问卷后，单击“问卷详情”，页面on-load事件，先通过editFlag判断当前页面类型，如果不可编辑，则将查询的对象数据展示到页面中。

了解标准页面与后端数据对象的关系后，可以打开其他页面，尝试分析一下页面逻辑，也可以查看帮助信息中的视频，学习如何在应用中修改标准页面。

2.7.3 (可选) 步骤 2: 自定义调查问卷项

调查问卷页面中的问卷项由对象模型定义而来，如果需要添加或修改调查项，需要先修改对象模型“问卷记录表”。

针对Astro轻应用开发者调查问卷，新增调查项“您最常使用的功能或者您最感兴趣功能有哪些？”，需要在问卷记录表对象模型中添加字段。系统已经预置了一些闲置的自定义字段，这里以新建一个进行介绍。

操作步骤

步骤1 选中“Model”文件夹下的具体对象“命名空间_轻应用名_qR_CST”，在“自定义字段”页签，单击“新建”。

图 2-103 新增自定义字段



步骤2 选择字段类型为“文本区”，单击“下一步”。

图 2-104 选择字段类型

新建字段



步骤3 设置字段的详情信息，单击“下一步”。

图 2-105 配置详情



- 标签：字段在界面的展示名，此处配置为“questionCode10”。
- 名称：字段在系统中的唯一标识，单击输入框自动生成。
- 字段帮助：字段提示的帮助信息。
- 描述：字段的描述信息。
- 是否必填：字段是否必填，勾选表示是必填字段。
- 是否可搜：是否能在数据调试台或者脚本中，使用search语句搜索该字段。

步骤4 设置字段的访问权限，即哪些Profile可以查看或修改该字段，单击“下一步”。

图 2-106 配置权限

新建字段



步骤5 设置字段是否加入布局，保持默认配置，单击“保存”。

图 2-107 添加到页面布局



字段创建完成后，在对象的“自定义字段”页签下可查看到该字段。后续在调查问卷页面的数据模型中可关联该字段，具体操作可参见（可选）[步骤3: 自定义调查问卷页面](#)。

图 2-108 已创建的字段




---结束

2.7.4 （可选）步骤 3：自定义调查问卷页面

根据需求自定义调查问卷页面，例如将（可选）[步骤2: 自定义调查问卷项](#)中创建的自定义字段添加到页面问卷调查项中，新增调查项“您最常使用的功能或者您最感兴趣功能有哪些？”。

操作步骤

步骤1 选中“page”文件夹下的问卷调查页面“命名空间_轻应用名_questionnairePage”，如果是锁定状态，请单击页面上方的，获取锁。

步骤2 查看并修改页面预置的数据模型。

AstroZero的标准页面是通过数据模型驱动的，页面所有的逻辑都是围绕数据模型展开的。在将模型与前台页面组件（例如输入框）或者后台逻辑绑定后，开发者只需要关注模型数据的实例化和处理，不需要关注页面的渲染和展示。


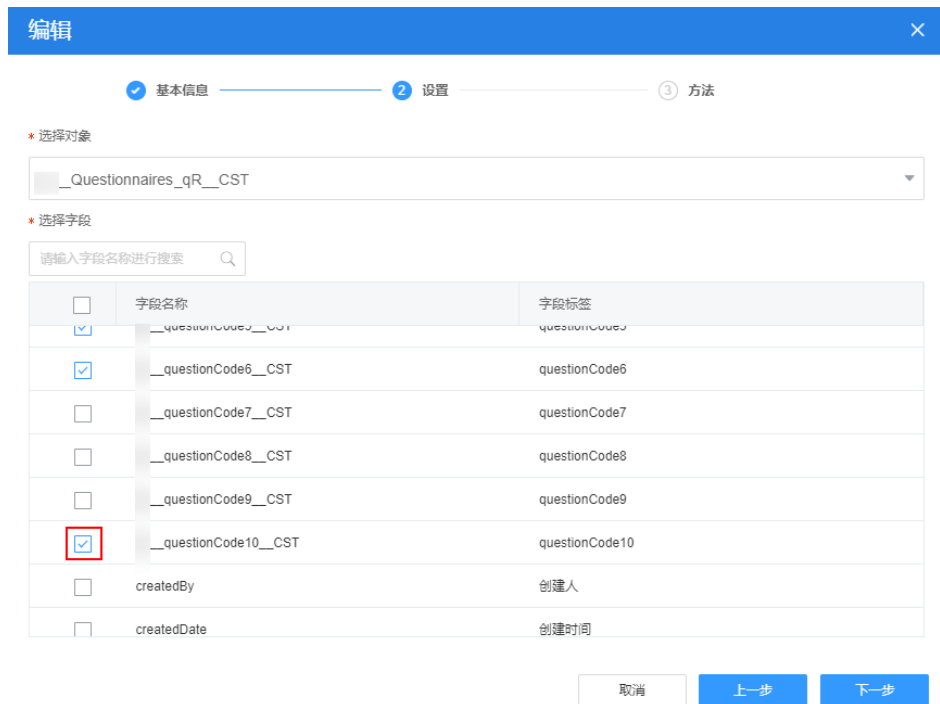
1. 如下图所示，在标准页面编辑器下侧，从“设计视图”切换到“模型视图”，单击“form_1”后的。

图 2-109 模型视图



- 勾选添加的对象字段，单击“下一步”。
该字段是在（可选）步骤2：自定义调查问卷项中，创建的自定义字段。

图 2-110 勾选字段



- 保持默认配置，单击“确定”。

步骤3 切换到“设计视图”（页面编辑区域左下角），从左侧组件区域拖拽“下拉框”组件到“设计视图”中。

图 2-111 拖入下拉框



步骤4 选中下拉框组件，在右侧配置下拉框组件属性。

图 2-112 绑定数据

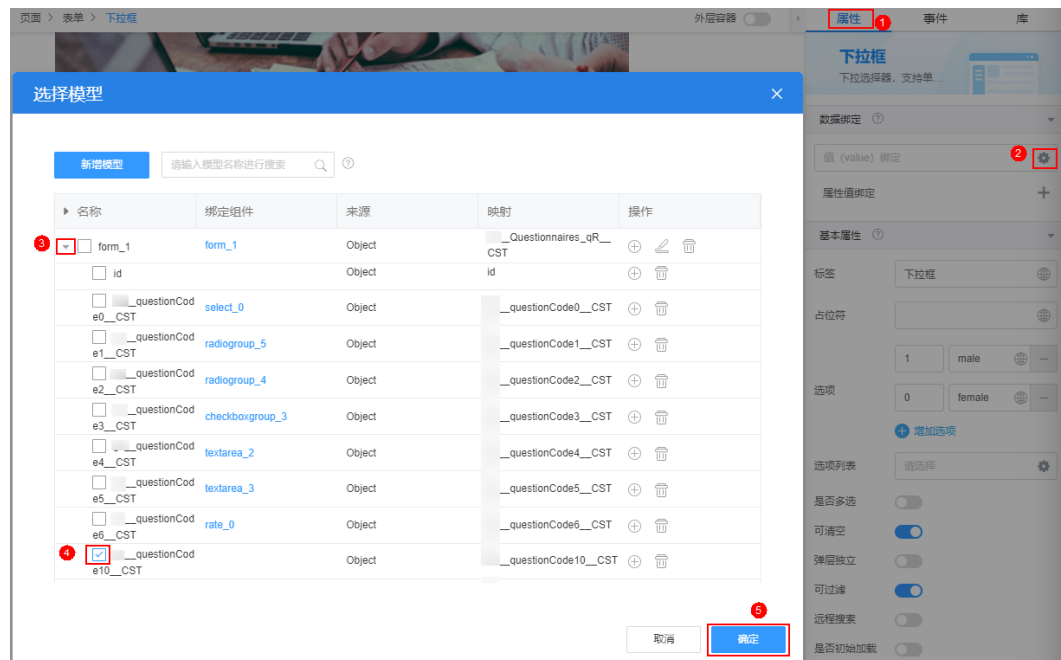


图 2-113 设置属性



图 2-114 设置标签

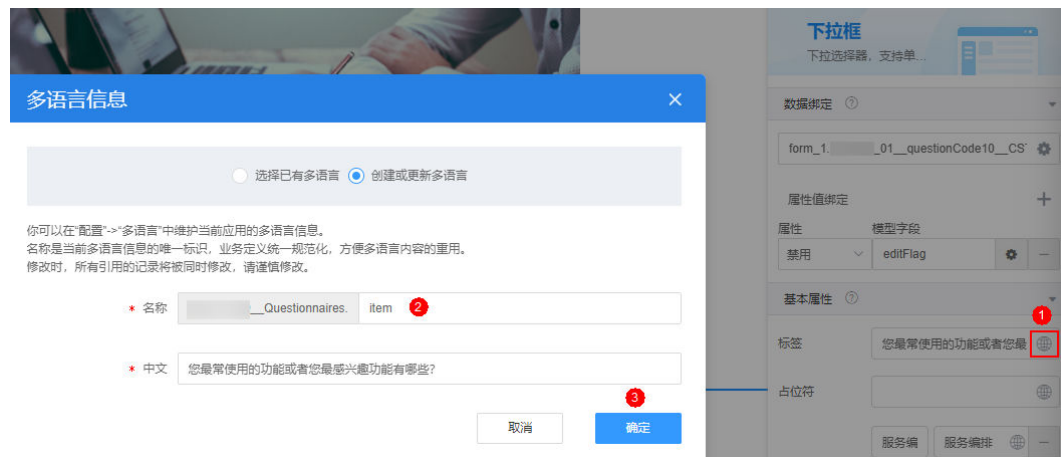





表 2-9 下拉框属性说明

属性	说明
数据绑定	<p>如图2-112所示, 设置组件绑定的数据模型, 在下拉框“属性”页签单击数据绑定后的齿轮图标, 选择步骤2中添加的模型字段“form_1.nasc_questionCode10_CST”。</p> <p>设置属性值绑定, 在属性值后单击“+”, 添加“禁用”属性, 模型字段选择“editFlag”。目的是填写问卷后, 查看调查问卷时, 不可编辑。</p>

属性	说明
标签	<p>下拉框的显示名称，只有下拉框放在表单form中才生效。</p> <p>支持国际化配置，在配置该属性时，可选择已有多语言、创建或更新多语言。此处创建的多语言会保存在租户的多语言库中。</p> <p>如图2-114所示，单击标签后的，选择“创建或更新多语言”，设置多语言名称和多语言内容，例如“名称”设置为“item”，中文设置为“您最常使用的功能或者您最感兴趣的功能有哪些？”。</p>
占位符	<p>下拉框未选择时，展示给用户的提示信息。</p> <p>支持国际化配置，在配置该属性时，可选择已有多语言、创建或更新多语言。此处创建的多语言会保存在租户的多语言库中。</p> <p>本示例不用配置。</p>
选项	<p>下拉框单击后可选择的选项内容。</p> <p>设置如图2-113所示选项。其中选项值支持国际化配置，在配置该属性时，选择“创建或更新多语言”，设置多语言名称和多语言内容。</p>
选项列表	<p>单击齿轮图标，支持选择系统中配置的选项列表。</p> <p>选项和选项列表只能配置其中一种。这里不用再配置选项列表。</p>
是否多选	<p>配置下拉框是否提供多选的能力。</p> <p>本示例选择打开此开关。</p>
可清空	<p>是否在选择下拉框显示一个删除图标，来删除选择框的内容。</p> <p>未选中“是否多选”时，才显示此配置项，本示例不用配置。</p>
弹层独立	<p>选项值区域展示不全，部分内容隐藏时，开启后，可显示完整选项值。</p> <p>本示例不用配置。</p>
可过滤	<p>配置下拉框是否提供过滤的能力。</p> <p>本示例不用配置。</p>
远程搜索	<p>是否根据输入的关键字调用自定义的接口查询数据。和事件“on-remote-search”配合使用。</p> <p>本示例不用配置。</p>
是否初始加载	<p>配置下拉框是否初始加载。</p> <p>本示例不用配置。</p>
必填	<p>是否设置为必填。</p> <p>本示例不用配置。</p>
必填错误信息	<p>在绑定了对象模型且下拉框放置于表单组件的情况下，当设置为必填时，如果输入值为空，会提示此处设置的错误提示信息。</p> <p>支持国际化配置，在配置该属性时，可选择已有多语言、创建或更新多语言。此处创建的多语言会保存在租户的多语言库中。</p> <p>本示例不用配置。</p>

步骤5 单击页面上方的，保存页面，并单击，释放锁。

步骤6 单击页面上方的，即可查看页面配置效果。

图 2-115 预览配置效果



为了持续提高您的开发体验，为您提供更好的开发平台，特邀您花几分钟反馈问卷，您的支持是我们最大的动力，衷心感谢您，祝开心每一天！

* 1.您的工作类型?

请选择

您最常使用的功能或者您最感兴趣功能有哪些?

请选择


- 全选
- 服务编排
- 脚本
- 工作流
- 标准页面
- 高级页面
- Astro大屏应用

---结束

2.7.5 步骤 4：预览并体验应用

调查问卷轻应用创建完成后，通过预览功能，可直接模拟问卷客户，预览并体验调查问卷应用的功能。

操作步骤

步骤1 在应用开发页面左侧，单击，预览调查问卷应用。

步骤2 在调查问卷页面填写问卷信息，单击“提交”。

图 2-116 填写调查问卷应用

为了持续提高您的开发体验，为您提供更好的开发平台，特邀您花几分钟反馈问卷，您的支持是我们最大的动力，衷心感谢您，祝开心每一天！

* 1.您的工作类型?

开发类

您最常使用的功能或者您最感兴趣功能有哪些?

服务编排 × 脚本 × 工作流 × 标准页面 × 高级页面 × Astro大屏应用 ×

* 2.您觉得平台的体验流畅吗?

非常流畅 一般流畅 不太流畅

* 3.您在开发过程中有困难，能否获取有效的文档/帮助?

非常有帮助 一般 没有帮助

* 4.您是从哪个途径知道数字平台的?

内外搜索 别人介绍 听过培训 其他途径

* 5.您觉得AppCube给您带来的最大价值是什么？相比其他同类产品，您为什么选择AppCube?

可快速进行应用开发交付。

* 6.您想对AppCube说点什么？可以是优点也可以是对AppCube的期许。

希望后续有些模板应用

7.您对平台的综合评分?

★★★★★

提交 重置

步骤3 在弹出的“提交成功”页面，单击“继续提交”。

如果需要查看调查问卷详情信息，请单击“查看详情”。

图 2-117 调查问卷应用提交成功页面



步骤4 在“调查问卷管理”页面，可查看到已提交的问卷记录。

单击操作列的“查看”、“删除”按钮，可查看或删除问卷记录。

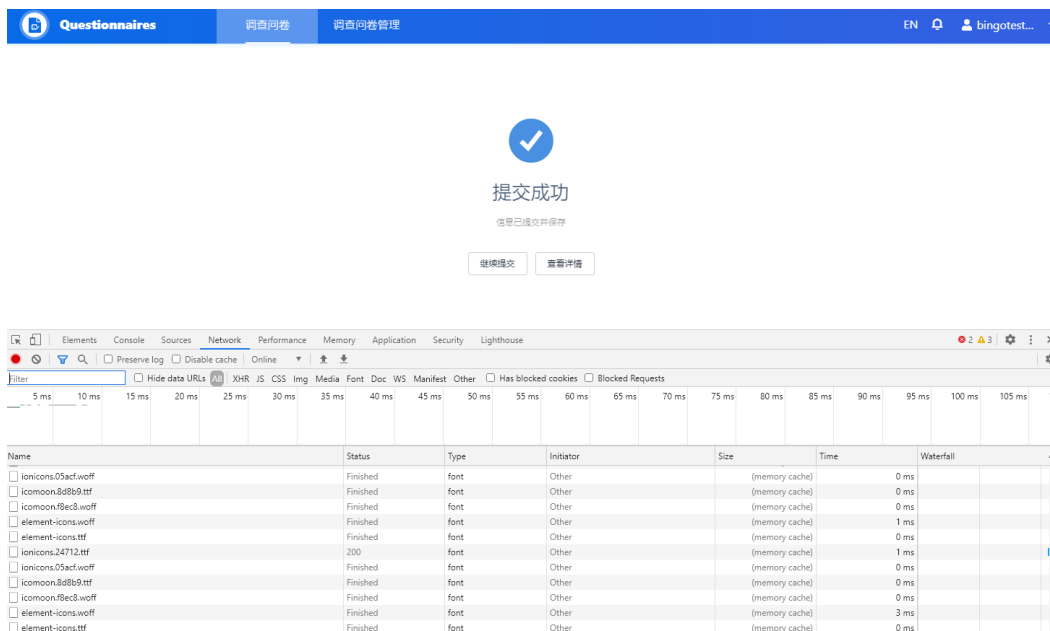
图 2-118 调查问卷管理



步骤5 在步骤3的提交成功页面，按“F12”或者“Ctrl + Shift + I”，开启调试工具。

在“Console”页签，可查看日志相关信息。在“Network”页签，可查看网络请求信息。

图 2-119 开启调试工具



----结束

2.7.6 步骤 5：编译发布应用到运行环境

应用开发完成后，最终目的是要将应用发布到运行环境中供用户使用，而在预览页面中体验的示例应用，只是在开发环境中模拟运行。因此，还需要将示例应用编译发布成一个资产包，以便在运行环境中部署应用。

说明

免费版不提供运行环境，请参考[购买商用租户](#)中操作购买专业版或专享版后，再进行打包发布操作，并在运行环境安装应用。

配置扩展权限集

在编译发布应用前，需要配置扩展权限集，即配置业务用户访问调查问卷应用所需要的权限。

说明

这里以调查问卷模板应用为例配置权限，如果配置其他应用的扩展权限，可能还需要配置接口、其他对象等相关权限，请根据实际情况进行配置。

步骤1 在AstroZero开发环境中配置扩展权限集，即配置业务用户访问调查问卷应用所需要的权限。

1. 返回AstroZero开发者页面，单击主菜单中的“管理”，进入开发环境管理中心。

图 2-120 单击管理






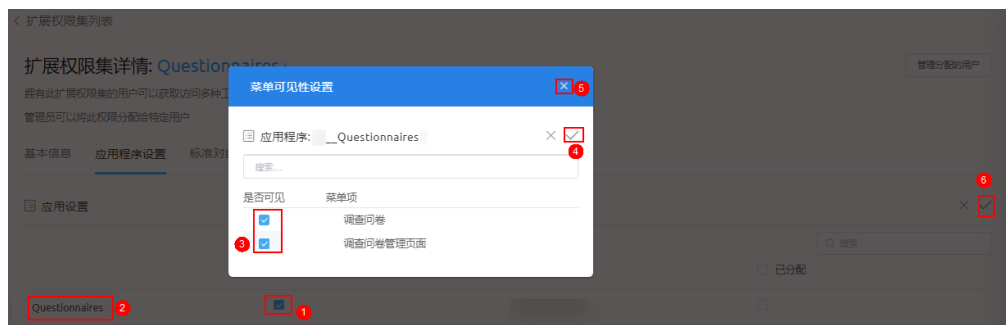
2. 在左侧导航栏中，选择“用户管理 > 扩展权限集”，单击“新建”。
3. 设置权限集标签为“调查问卷扩展权限”，名称为“Questionnaires”，单击“保存”。
4. 在扩展权限集详情页面，选择“应用程序设置”页签。
5. 单击右上角 ，选中“Questionnaires”的“已分配”复选框，单击应用名，勾选所有菜单可见，单击 ，关闭弹框。单击 ，完成应用和菜单可见性的配置。

图 2-121 设置应用可见





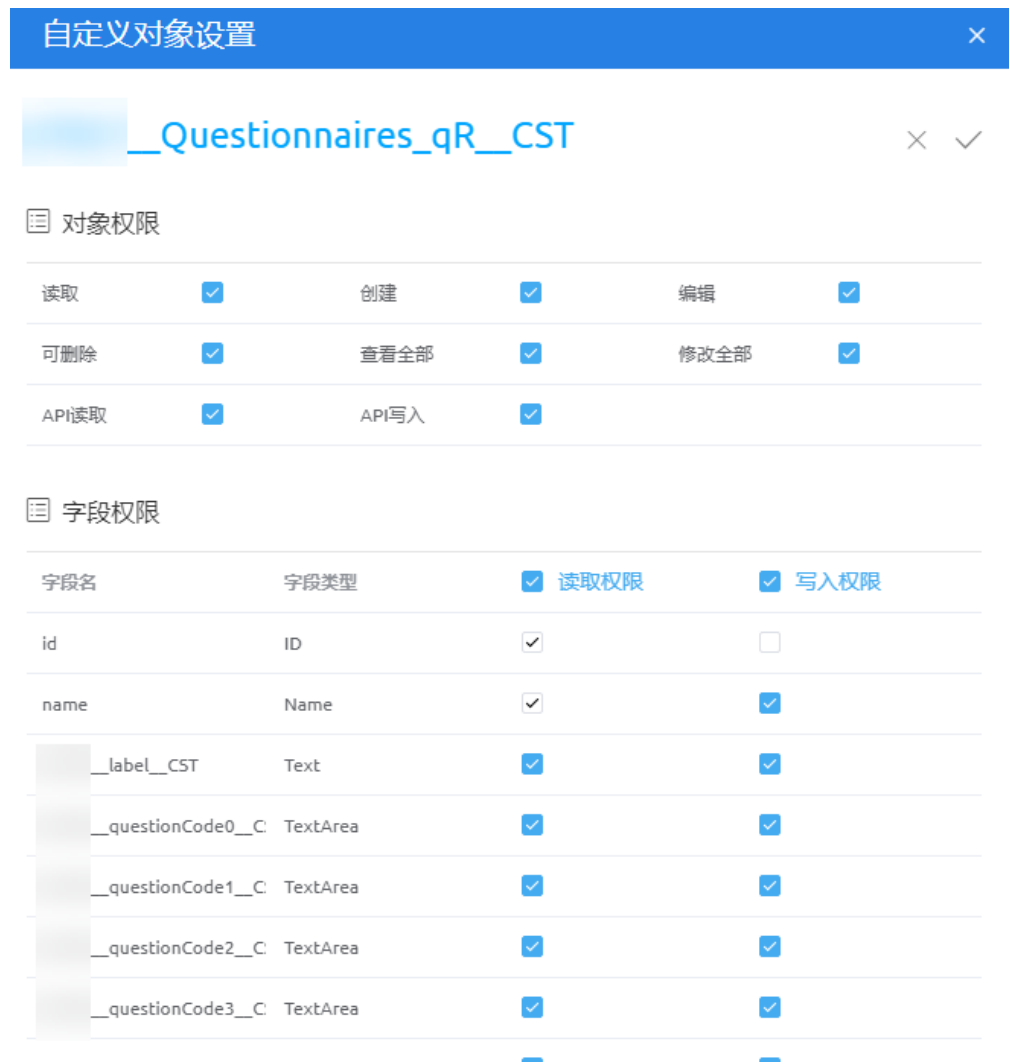
6. 选择“自定义对象设置”页签，查找并单击对象“命名空间 `__Questionnaires_qR_CST`”，单击右上角 ，勾选对象和字段所有权限，单击 。

图 2-122 勾选对象和字段所有权限



设置后，显示如图2-123所示。

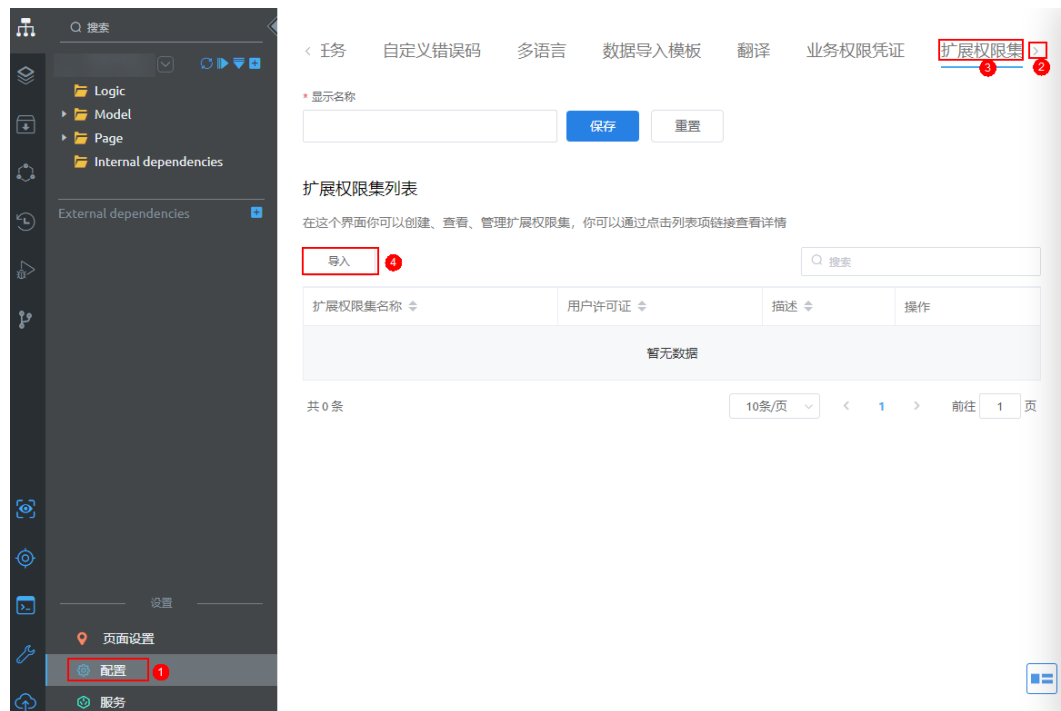
图 2-123 设置结果



步骤2 单击主菜单中的“首页”，单击“我的应用”下调查问卷应用“Questionnaires”，进入应用开发工作台。

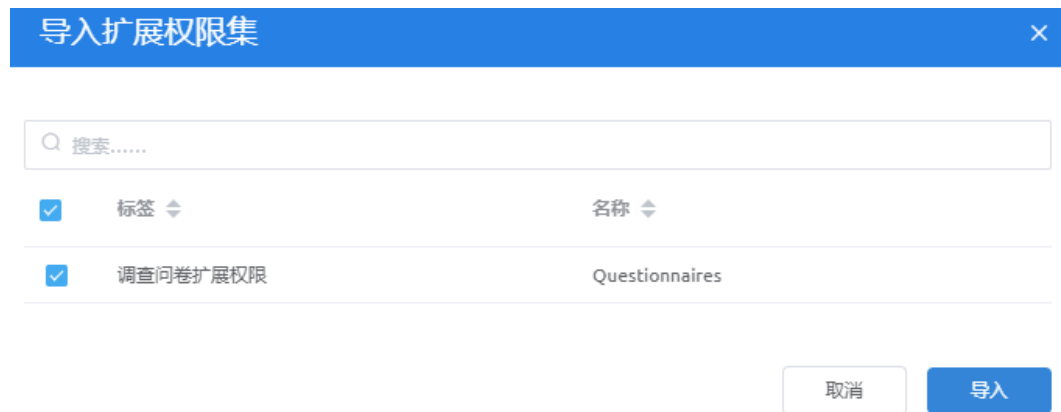
步骤3 单击页面左下角的“配置”，在右侧的“扩展权限集”页签，单击“导入”。

图 2-124 单击导入




步骤4 勾选**步骤1**中创建的扩展权限集，单击“导入”。

图 2-125 导入扩展权限集



----结束


编译发布应用

步骤1 在调查问卷应用开发页面，单击，选择“设置”。


步骤2 在包类型中，选择“资产包”，单击“保存”。

图 2-126 设置包类型



步骤3 在调查问卷应用开发页面，单击，选择“编译”。

编译成功后，关闭编译成功的窗口。如果想了解更多关于发布应用的设置，请参考[发布App](#)。

步骤4 单击左下角，在发布应用中，选择“我的仓库”。

步骤5 在发布到我的仓库页面，设置版本号和描述信息，单击“发布”。

图 2-127 发布到我的仓库



发布成功后，页面显示“程序包已经被成功上传到我的仓库”。

----结束

在运行环境部署应用

调查问卷应用发布到“我的仓库”后，会同步发布在运行环境的“我的仓库”中。因此，只需要登录到运行环境，即可安装部署应用。

步骤1 在AstroZero开发环境右上角单击用户名，选择“运行环境”。

图 2-128 切换到运行环境



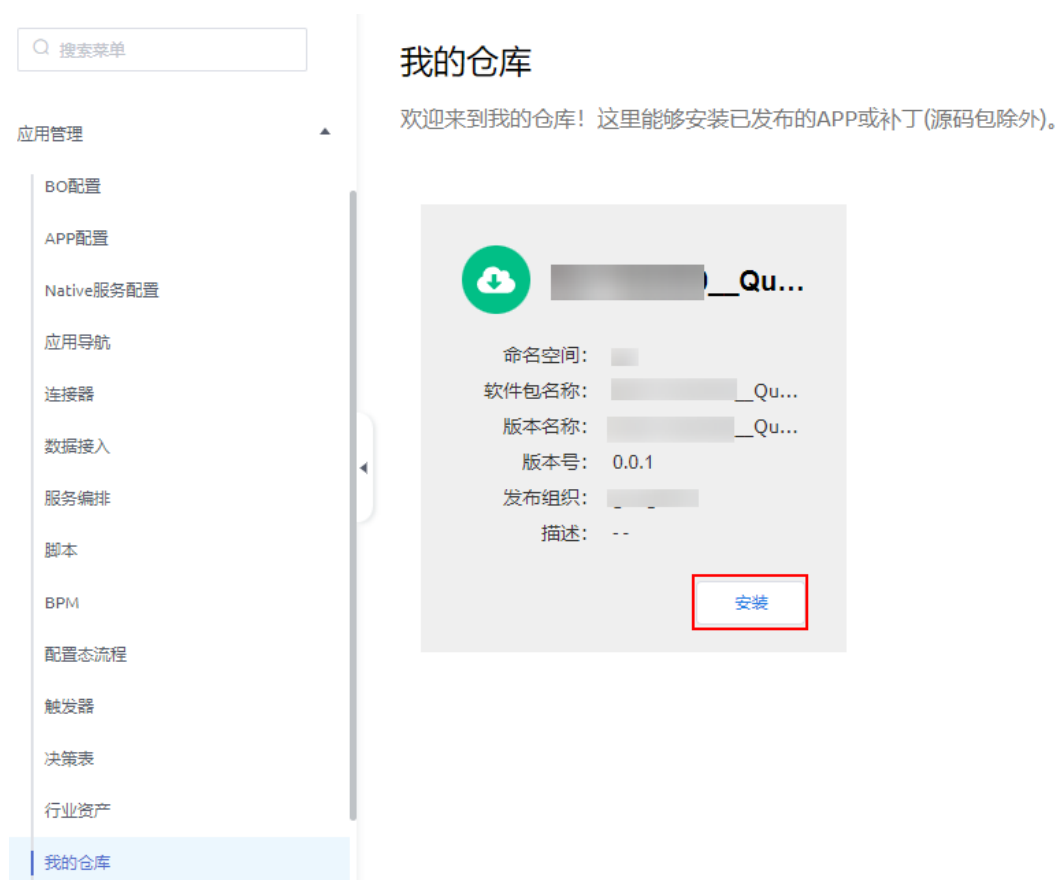
步骤2 在管理看板页面，选择“我的仓库”。

图 2-129 我的仓库



步骤3 在我的仓库中，单击调查问卷应用中的“安装”，单击“确定”。

图 2-130 安装应用



----结束

2.7.7 步骤 6: 创建业务用户并登录应用

调查问卷应用发布到运行环境后，可供业务用户登录并使用应用。

前提条件

开发者用户在运行环境管理中心的“系统管理 > 域名配置”中，查看子域名并分享给业务用户。

图 2-131 查看租户访问域名



操作步骤

步骤1 进入运行环境的业务配置中心。




1. 使用华为账号，登录[华为云网站](#)，在顶部导航栏右侧单击“控制台”，进入华为云控制台。
2. 在左侧导航栏上方，单击，选择服务实例所在的区域项目。
3. 单击，在查找框中搜索“Astro轻应用”，单击查找到的结果，进入AstroZero服务控制台。
4. 单击“进入首页”，进入应用开发页面。
5. 在“工作台”页签，单击左上角的，选择“业务配置中心”，进入运行环境业务配置中心。

图 2-132 进入运行环境业务配置中心



步骤2 创建角色权限。

说明

购买AstroZero标准版或专业版实例时，才会显示“角色权限”菜单。

业务配置中心中的角色权限都是基于权限“Portal User Profile”创建而来的。不同的业务用户权限可能有所不同，这时可通过配置扩展权限集给业务用户分配不同的权限。关于扩展权限集的更多介绍，请参见[设置扩展权限集](#)。

1. 在“业务配置中心”页面，选择“组织用户”。
2. 在左侧导航栏中，选择“角色权限”。
3. 单击“所有角色”后的“+”，设置添加的角色名称。

本示例设置为“调查问卷权限”，并配置应用为可见和默认应用。

图 2-133 添加角色



- 单击应用名，在“权限配置”页面给权限添加扩展权限集，权限集为**步骤3**在应用中导入的扩展权限集。

图 2-134 添加扩展权限集



步骤3 通过运行环境的业务配置中心创建应用的业务用户，例如创建业务用户“testQuestionnaires”。

- 在左侧导航栏中，选择“用户管理”，单击“创建用户”。
- 在“新建用户”页面填写用户信息，用户名为“testQuestionnaires”，其他参数按照需求设置。
- 在“角色”区域中，从左侧勾选**步骤2**中创建的角色“调查问卷权限”，再单击“>”赋予用户相应的权限，单击“保存”。

图 2-135 配置业务用户

基本信息

* 用户名
testQuestionnaires

* 名称
testQuestionnaires

* 邮箱

电话

* 密码

* 再次输入密码

用户设置

* 语言
中文

时区
(GMT+08:00) China Standard T...

角色

未选 1/4

已选 0/0

业务管理员

企业成员

开发者

调查问卷权限

无数据

保存 取消

步骤4 业务用户在浏览器中，输入前提条件中获取到的运行环境访问域名，输入用户名和密码，即可登录应用Portal运行页面。

图 2-136 登录应用



步骤5 业务用户首次登录需要重置密码，重置后使用新密码再次登录应用Portal运行页面。

步骤6 登录后查找应用，并体验应用。

如果应用为非默认应用，可单击应用前的图标切换应用。在调查问卷页面填写问卷信息，单击“提交”。如果需要查看调查问卷详情信息，在弹出的“提交成功”页面，单击“查看详情”。

图 2-137 填写调查问卷应用

为了持续提高您的开发体验，为您提供更好的开发平台，特邀您花几分钟反馈问卷，您的支持是我们最大的动力，衷心感谢您，祝开心每一天！

* 1.您的工作类型?

开发类

您最常使用的功能或者您最感兴趣功能有哪些?

服务编排 × 脚本 × 工作流 × 标准页面 × 高级页面 × Astro大屏应用 ×

* 2.您觉得平台的体验流畅吗?

非常流畅 一般流畅 不太流畅

* 3.您在开发过程中有困难，能否获取有效的文档帮助?

非常有帮助 一般 没有帮助

* 4.您是从哪个途径知道数字平台的?

内外搜索 别人介绍 听过培训 其他途径

* 5.您觉得AppCube给您带来的最大价值是什么？相比其他同类产品，您为什么选择AppCube?

可快速进行应用开发交付。

* 6.您想对AppCube说点什么？可以是优点也可以是缺点还可以是对AppCube的期许。

希望后续有些模板应用

7.您对平台的综合评分?

★★★★★

提交 重置

----结束

扩展知识

- 本节是通过配置扩展权限集，在业务用户的权限“调查问卷权限”中添加扩展权限集，来配置业务用户的权限。您也可以直接在业务用户的Profile权限中，配置使用调查问卷应用的具体权限，详情可参见[步骤3](#)。
- [操作步骤](#)中介绍了如何创建业务用户及业务用户如何访问调查问卷应用。如果需要实现游客身份的用户访问调查问卷应用，可参考[如何在运行环境中，实现游客访问标准页面](#)。

2.8 简易出差审批应用开发

本示例通过使用AstroZero的工作流功能，带您快速创建一个简易版的出差审批应用。

通过创建出差审批应用，您可以初步了解轻应用的基本概念并体验“工作流”的能力。

📖 说明

工作流即业务流程管理，是AstroZero的核心功能之一，工作流是一种可视化的流程编排。如果需了解更多工作流内容，请参见[什么是工作流](#)。

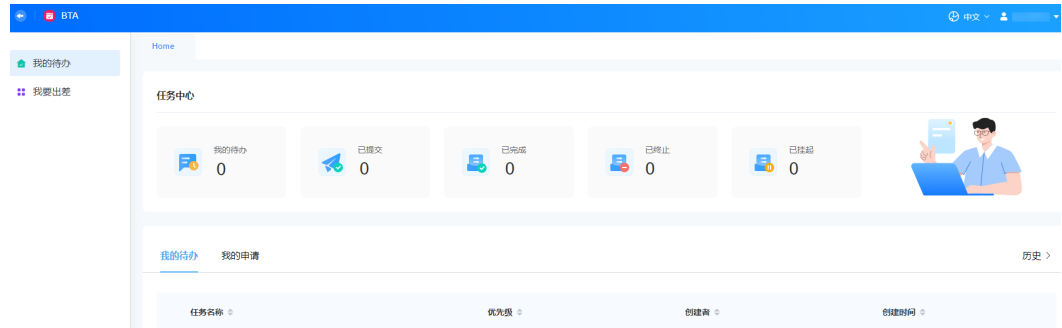
场景说明

本示例主要针对员工出差场景，即员工在出差前需要提交一个出差申请审批的电子流程，员工提交出差申请后，主管处理审批或拒绝提交人申请。

本示例中的出差审批应用主要包括如下功能：

- 基于 workflow 模板创建出差电子流。
- 发送邮件。

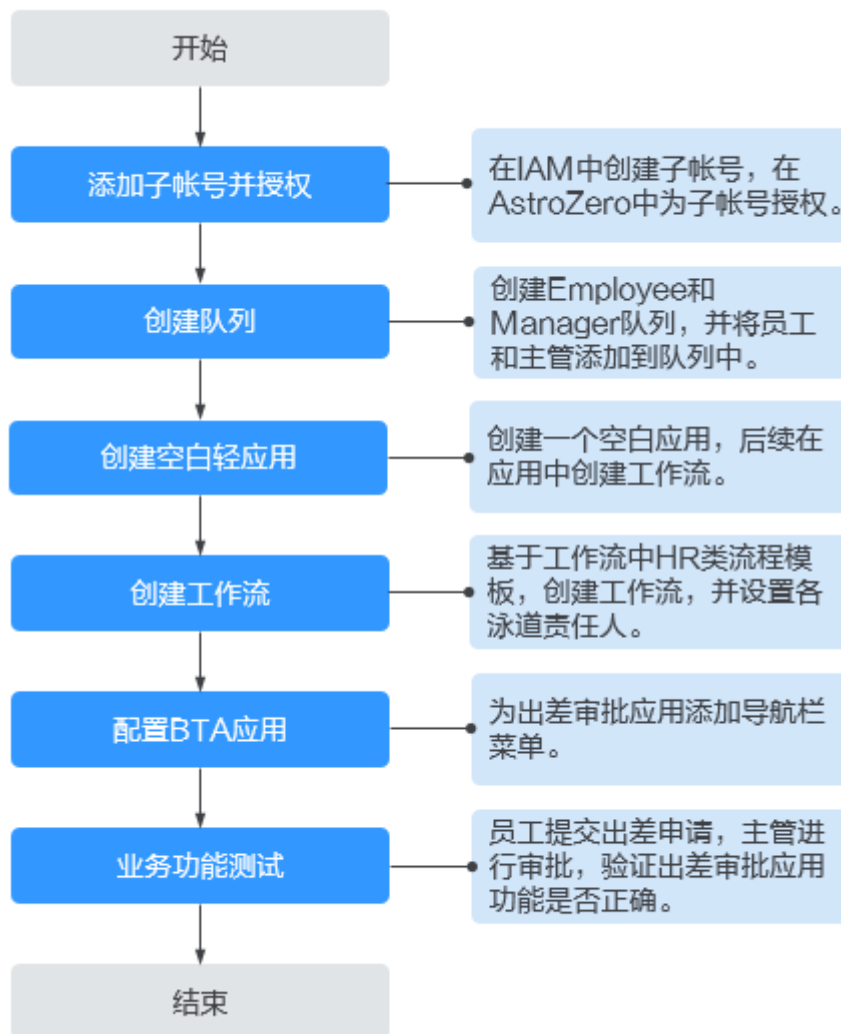
图 2-138 出差审批应用最终效果图



开发流程

在 AstroZero 中开发出差审批应用流程，如图 2-139 所示。

图 2-139 出差审批应用开发流程





前提条件

- 已注册华为账号并开通华为云，且已完成实名认证。
实名认证分为企业认证和个人认证，两种认证方式均支持新增子账号。如何申请实名认证，请参见[实名认证](#)。
- 使用已注册的华为账号，申请开通免费试用AstroZero。
如何注册并申请试用AstroZero，请参见[申请免费试用](#)。本示例中的开发操作均是使用华为账号进行的，子账号（主管、员工）仅在业务功能调测时使用。

步骤 1：添加子账号并进行授权

添加子账号的目的是为了给应用添加不同角色的业务用户，本示例中的应用包括主管和员工两种角色的业务用户。创建出差审批应用前，需要将员工、主管子账号添加到AstroZero中。

步骤1 在统一身份认证服务IAM中创建两个子账号（主管和员工）。

1. 以华为账号登录[华为云网站](#)，在顶部导航栏右侧单击“控制台”，进入华为云控制台。
2. 在左侧导航栏上方，单击，选择服务实例所在的区域项目。
3. 单击，在查找框中搜索“统一身份认证服务”，单击查找到的结果，进入IAM服务控制台。
您也可以选择“管理与监管 > 统一身份认证服务 IAM”，进入IAM服务控制台。
4. 在“用户”中，单击“创建用户”，创建主管、员工两个用户。
如何创建用户，请参见[创建IAM用户](#)。假设，本示例创建主管用户为Helen、员工用户为Mike。

步骤2 在AstroZero中添加子账号并授权。



1. 以华为账号登录华为云。
2. 单击，在查找框中搜索“Astro轻应用”，单击查找的结果，进入AstroZero服务控制台。
3. 在实例页面，单击“进入首页”。
4. 在页面左上角，单击，选择“开发环境管理 > 环境配置”，进入AstroZero开发环境的环境配置。

图 2-140 进入 AstroZero 开发环境的环境配置



5. 在左侧导航栏中，选择“用户安全 > 用户”，单击“添加IAM用户”。
6. 在全部用户列表中，选择步骤1中创建的子账号，单击“下一步”。

图 2-141 选择待添加的子账号

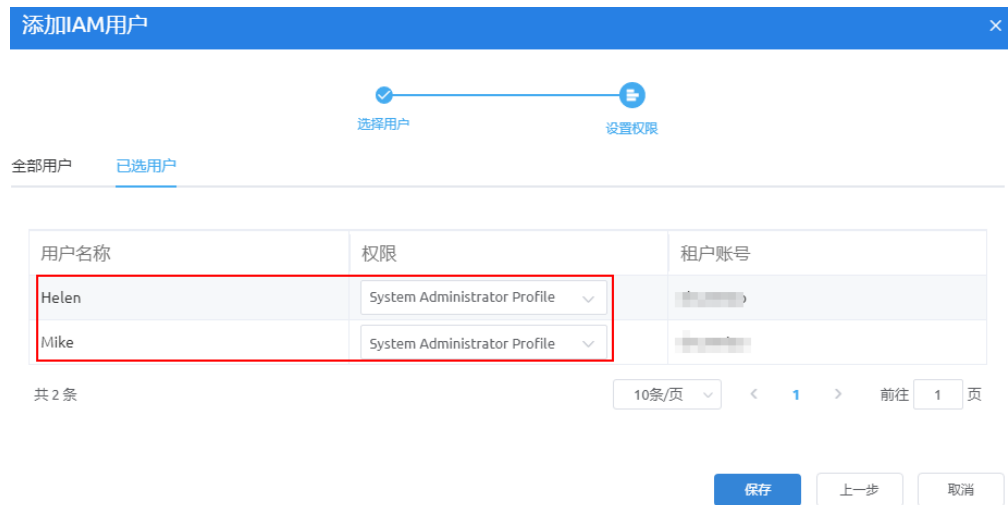


7. 赋予主管和员工System Administrator Profile权限，单击“保存”。

📖 说明

System Administrator Profile为系统管理员权限，拥有这个权限的用户拥有AstroZero全部权限。在实际的出差审批应用中，只需赋予业务用户Portal User Profile或Anonymous User Profile权限即可。此处为了验证出差审批应用的功能，赋予业务用户（员工Mike、主管Helen）System Administrator Profile权限。如何给业务用户授权，请参见[配置业务用户](#)。

图 2-142 给予账号授权



----结束

步骤 2：创建队列

队列是AstroZero中的一种成员集，即在实际业务场景中，用来记录一类具有相同权限和任务对象的成员集。

本示例中应用需要使用到表2-10中2个队列，用于在工作流中区分不同角色在流程中处理的任务。

表 2-10 队列信息

名称	说明
Employee	员工队列，添加的队列成员为普通员工，如Mike。
Manager	主管队列（负责一级审批），添加的队列成员为主管，如Helen。

步骤1 以华为账号登录AstroZero环境配置。

步骤2 在主菜单中，选择“维护”。

步骤3 在左侧导航栏中，选择“全局元素 > 队列”。

步骤4 在队列页面，单击“新建”，创建表2-10中的Employee队列。

1. 在新建队列基本信息中，设置队列标签和名称，其他参数保持默认。
 - 标签：新建队列的标签，用于在AstroZero界面上显示。本示例配置为Employee。
 - 名称：新建队列的名称，AstroZero系统中的唯一标识。本示例配置为Employee。
2. 在队列成员中，单击“添加”，为Employee工作队列添加Mike及当前租户账号为成员。

图 2-143 添加 Mike 成员

添加成员

成员类型

用户

成员列表

由于使用了懒加载, 最多显示 300 条记录, 请输入部分关键字以便缩小匹配范围。

用户: Mike 用户

取消 添加

3. 单击“保存”，进入队列详情页面。

在Employee队列详情页面，可以查看新建队列的信息。如果创建队列时未添加成员，在成员信息中单击“添加”，可为队列添加成员。

步骤5 参照**步骤4**中操作，创建Manager队列。

图 2-144 设置标签和名称

基本信息

* 标签 * 名称

Manager Manager

公共邮箱 工作队列管理者

请输入 请选择

给成员发送邮件 队列对象

图 2-145 为队列添加 Helen 成员及租户账号



----结束

步骤 3: 创建空白轻应用

创建应用是在AstroZero开发环境开发项目的第一步，也是端到端构建软件应用的入口。此处创建一个空白轻应用的目的是在应用的Logic目录下创建BPM。


- 步骤1 以华为账号登录AstroZero服务控制台。
- 步骤2 在实例列表中，单击“进入首页”，进入应用开发页面。
- 步骤3 在左侧导航栏中，单击“应用”，进入低代码应用页面。
- 步骤4 单击新建空白应用后的 ，进入新建空白应用页面。

图 2-146 进入创建轻应用入口



步骤5 设置应用标签和名称，此处均设置为BTA。

图 2-147 设置轻应用标签和名称

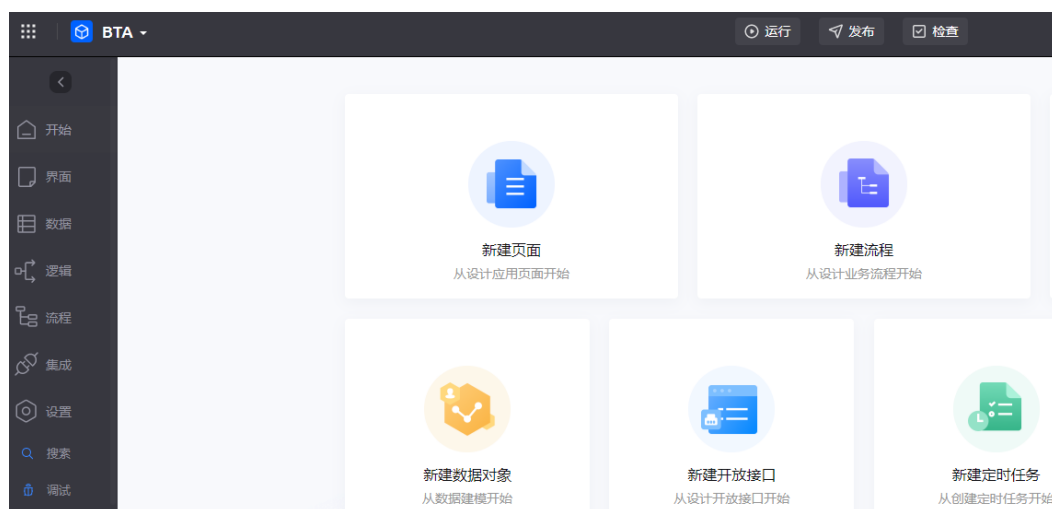


须知

图2-147名称前模糊掉的内容为命名空间，在AstroZero中为了避免不同租户间数据的重名，租户在首次创建应用时需要先定义一个命名空间。一个租户只能创建一个命名空间，创建后不支持修改，请谨慎定义。

步骤6 单击“新建”，进入BTA新版应用设计器。

图 2-148 BTA 新版应用设计器



---结束

步骤 4：创建工作流

基于工作流中HR类流程模板，创建工作流，并设置各泳道责任人。

步骤1 创建工作流。

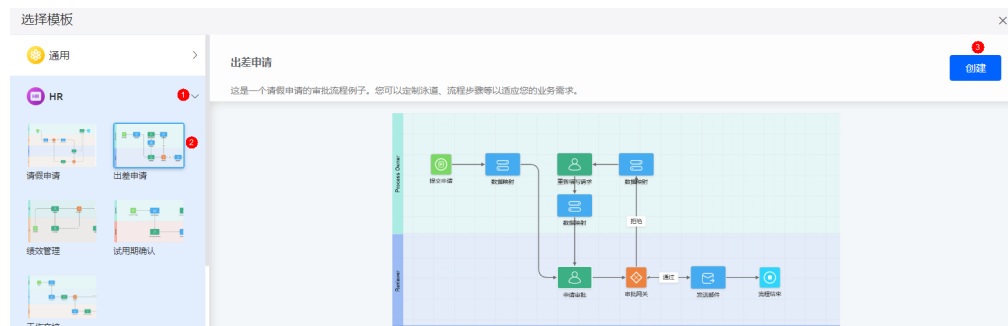
1. 在BTA新版应用设计器，选择“流程”。
2. 单击工作流后的 $+$ ，进入新建工作流页面。
3. 在添加工作流页面，单击“基于模板”，设置标签、名称和描述信息。

图 2-149 添加工作流



4. 单击“选择模板”，选择“HR”中的“出差申请”模板后，单击“创建”。

图 2-150 选择出差申请模板

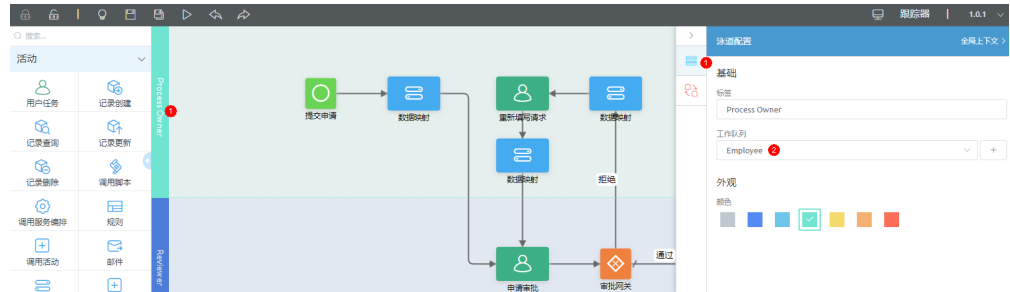


步骤2 设置工作流，配置各泳道处理人。

在进行工作流设计前，请先参照[初识工作流](#)中内容对工作流设计界面进行全面的了解。

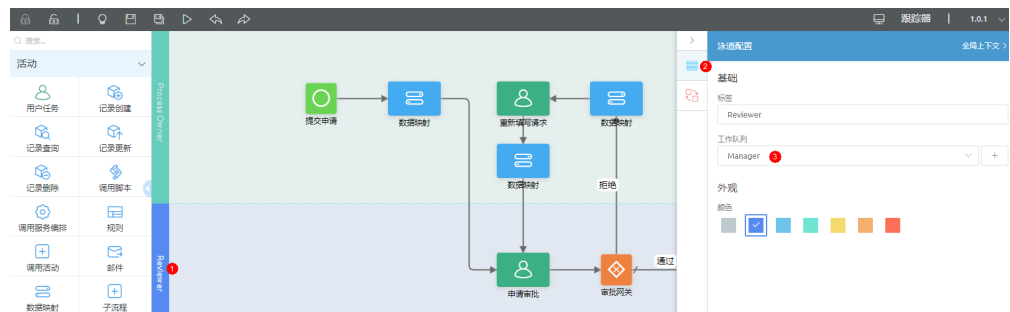
1. 在 workflow 开发页面，单击“Process Owner”泳道，配置工作队列为“Employee”。

图 2-151 配置员工为申请人



2. 单击“Reviewer”泳道，配置工作队列为“Manager”。

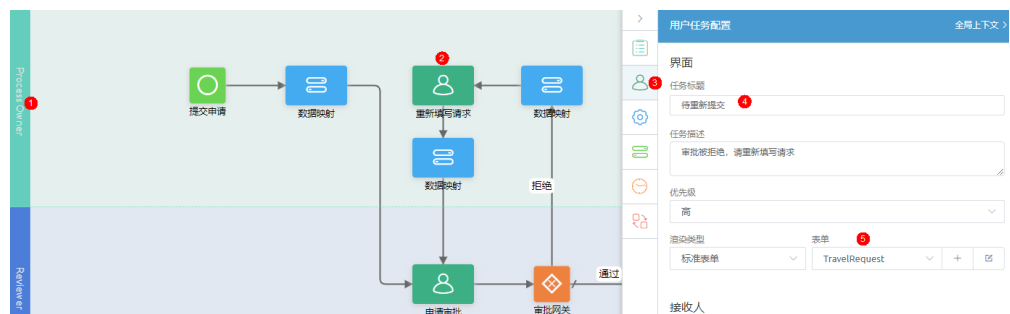
图 2-152 配置主管为审批人



3. 单击泳道“Process Owner”上的“重新填写请求”用户任务元素，设置任务标题为“待重新提交”。

执行该操作的目的是，后续在“我的待办”中可以看到“待重新提交”的任务状态。

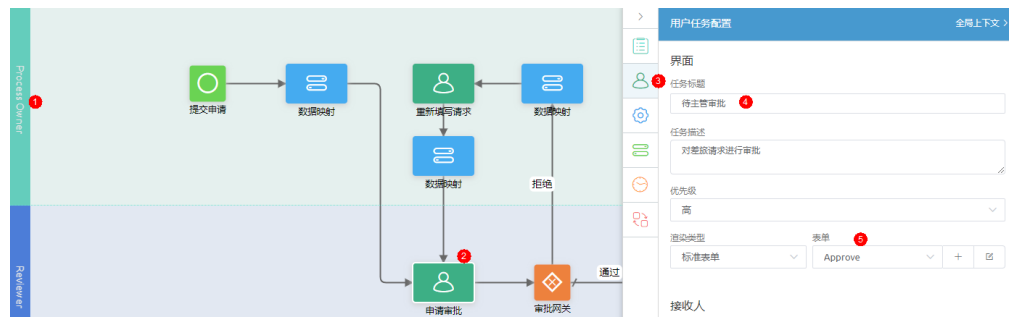
图 2-153 配置为待重新提交





4. 单击泳道“Reviewer”上的“申请审批”用户任务元素，设置任务标题为“待主管审批”。

执行此操作的目的是，后续在“我的待办”中可以看到“待主管审批”的任务状态。

图 2-154 配置为待主管审批



5. 单击 , 保存工作流。
6. 单击 , 启用工作流。

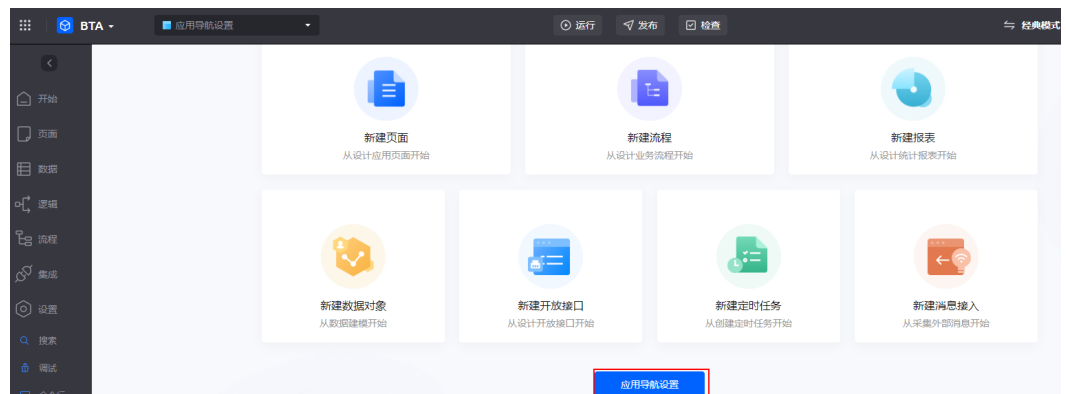
---结束

步骤 5: 配置 BTA 应用

在应用配置中, 定义出差审批应用的导航菜单栏。

步骤1 在BTA新版应用设计器的“开始”页面, 单击底部的“应用导航设置”。

图 2-155 进入应用配置



步骤2 在“主导航设置”页签, 单击“Home”, 将“菜单名称”设置为“我的待办”。

图 2-156 编辑页签



步骤3 在主导航设置中, 单击“新建”, 新建我要出差页签。

图 2-157 添加我要出差页签

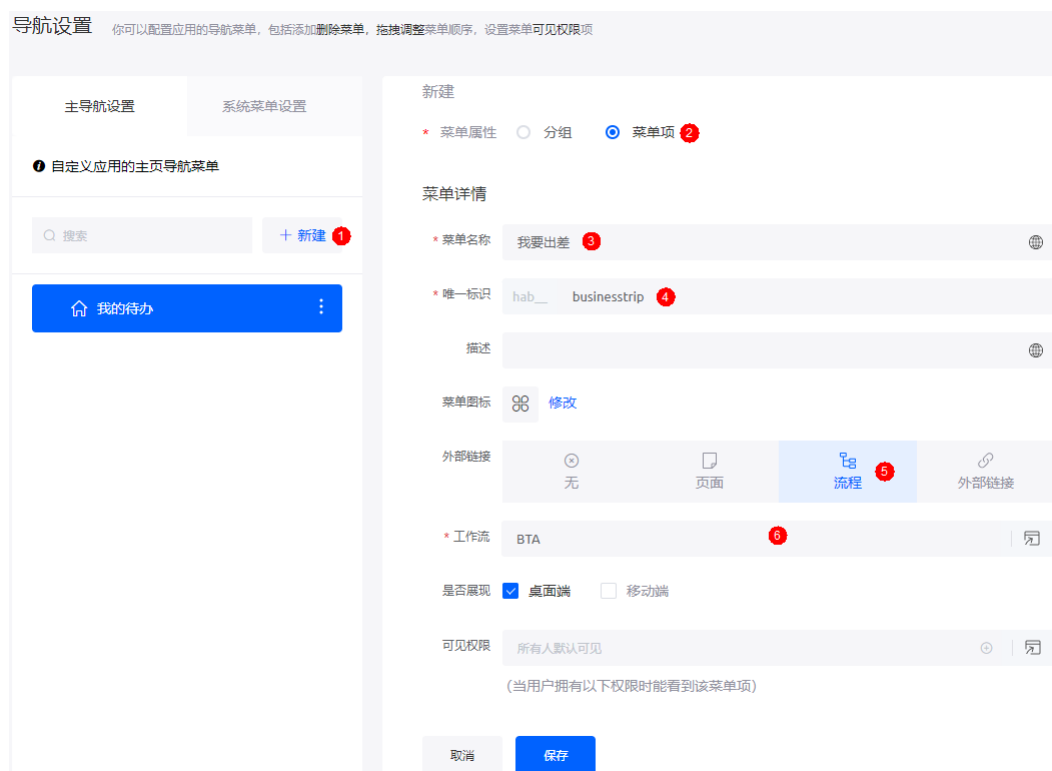
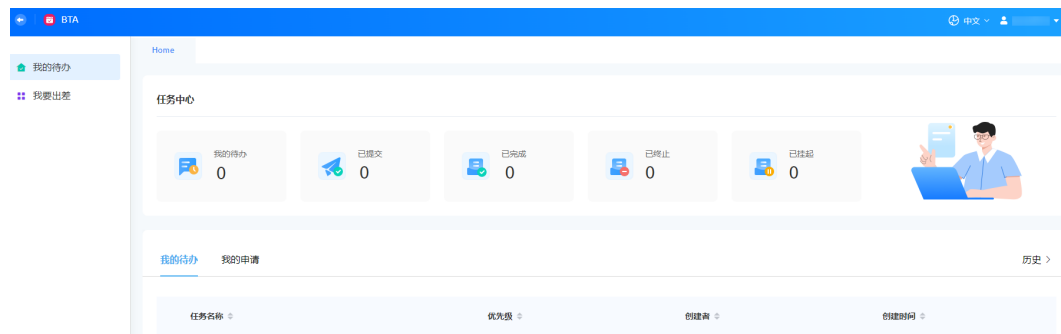


图 2-158 导航条最终效果



步骤4 在BTA新版应用设计器主菜单中，单击“运行 > 立即运行”，可预览出差审批应用。

图 2-159 出差审批应用效果图



到此，您已完成出差审批应用的开发。

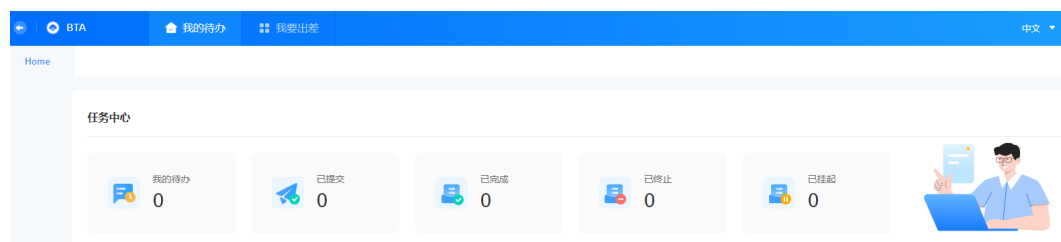
思考：如何将导航栏菜单在出差审批应用顶部显示？

在应用配置页面的“外观设置”中，可以修改菜单样式、应用图标等。

图 2-160 修改菜单样式、应用图标



图 2-161 设置后效果



----结束

步骤 6：业务功能调测

出差审批测试流程：员工提交出差申请，主管对申请进行审批，分别测试审批通过、拒绝与重填。

步骤1 我是员工，填写出差申请。

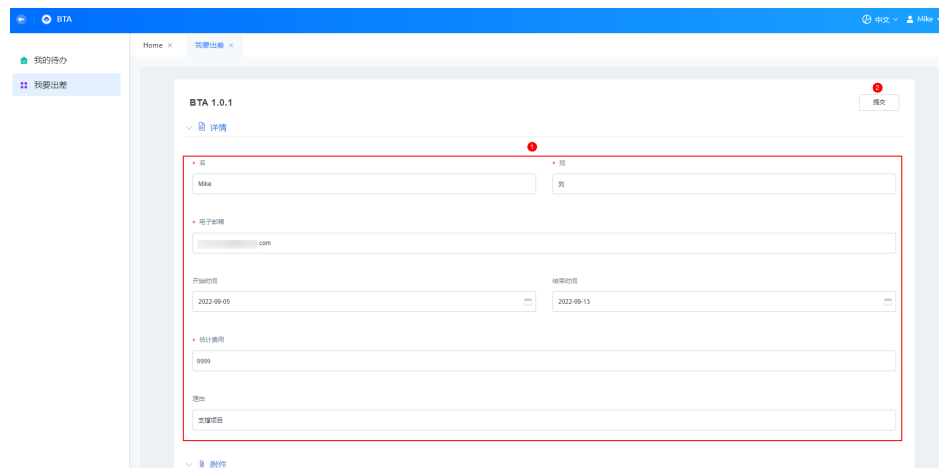
1. 以员工账号（本示例为Mike），登录AstroZero服务控制台。

图 2-162 员工登录 AstroZero



2. 在实例页面，单击“进入首页”。
3. 在左侧导航栏中，单击“应用”，进入低代码应用页面。
4. 在低代码应用列表中，单击已创建BTA应用后的“编辑”，选择“体验新版设计器”，进入BTA新版应用设计器。
5. 在主菜单中，单击“运行 > 立即运行”，进入出差审批应用预览页面。
6. 单击“我要出差”，填写出差信息，单击“提交”。

图 2-163 员工填写出差申请



7. 在“我的待办 > 我的申请”中，可查看到已提交的出差申请电子流。

图 2-164 查看我的申请



步骤2 我是主管，审批出差申请。

1. 以主管账号（本示例为Helen），登录AstroZero服务控制台。

图 2-165 主管登录 AstroZero



2. 在实例页面，单击“进入首页”。
3. 在左侧导航栏中，单击“应用”，进入低代码应用页面。
4. 在低代码应用列表中，单击已创建BTA应用后的“编辑”，选择“体验新版设计器”，进入BTA新版应用设计器。
5. 在主菜单中，单击“运行 > 立即运行”，进入出差审批应用预览页面。
6. 单击“我的待办”，在我的待办列表中，单击“待主管审批”，进入主管审批页面。

图 2-166 我的待办

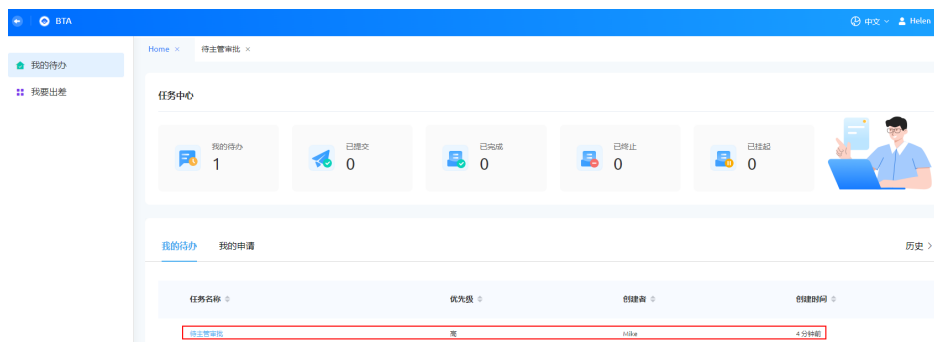
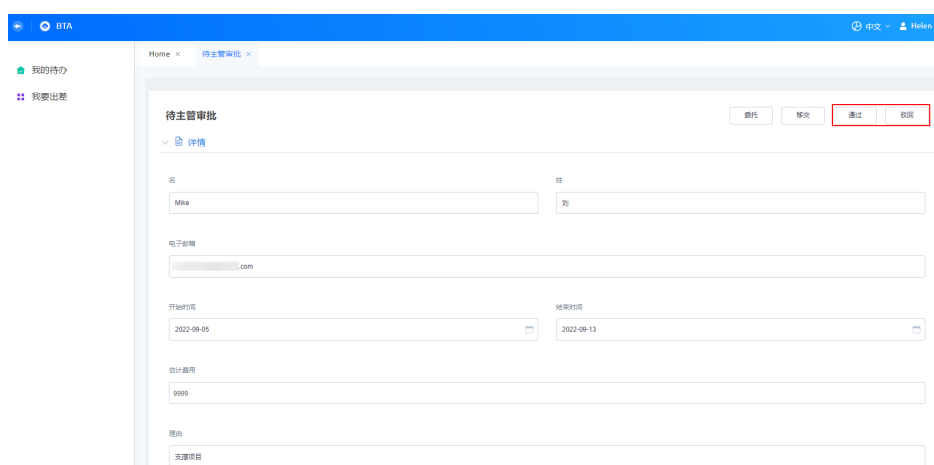
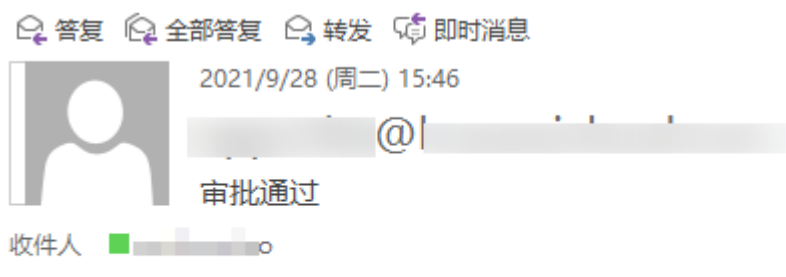


图 2-167 主管审批页面



7. 单击“通过”或“驳回”。
 - 如果主管通过了审批，会发送审批通过的邮件到图2-163中的员工电子邮箱中，如图2-168。

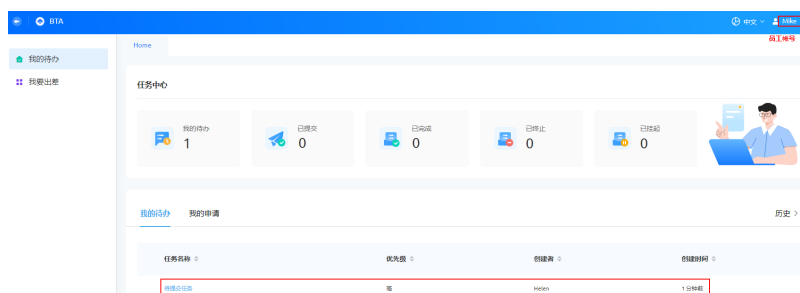
图 2-168 审批通过邮件



审批通过

- 如果主管驳回了审批，在员工“我的待办”中，可查看到被主管退回的申请记录，如图2-169。

图 2-169 查看被拒绝的记录



恭喜您，完成了出差审批应用的开发和功能测试。通过本章节的操作，相信您对 AstroZero 中的轻应用及工作流有了一定的了解，如果需更深入的学习 AstroZero，可挑战一下[进阶实战](#)和[高阶实战](#)。

📖 说明

思考：如何将出差审批应用发布到运行环境？发布到运行环境后，队列中数据会同步到运行环境中吗？

- **问题1：**免费版不提供运行环境，如果需将应用发布到运行环境，请购买专享版或专业版本，详情可参见[购买商用租户](#)。如何编译发布应用，请参见[如何一键部署应用](#)。
- **问题2：**出差审批应用发布到运行环境后，开发环境中的队列信息不会同步到运行环境中，请在运行环境中重新执行[步骤2：创建队列](#)。

----结束

2.9 创建 HelloWorld 应用

本节将带您从零开始，手动开发一个简单应用“HelloWorld”。通过创建 HelloWorld 应用，您可以轻松漫游 AstroZero，初步认识并体验 AstroZero。

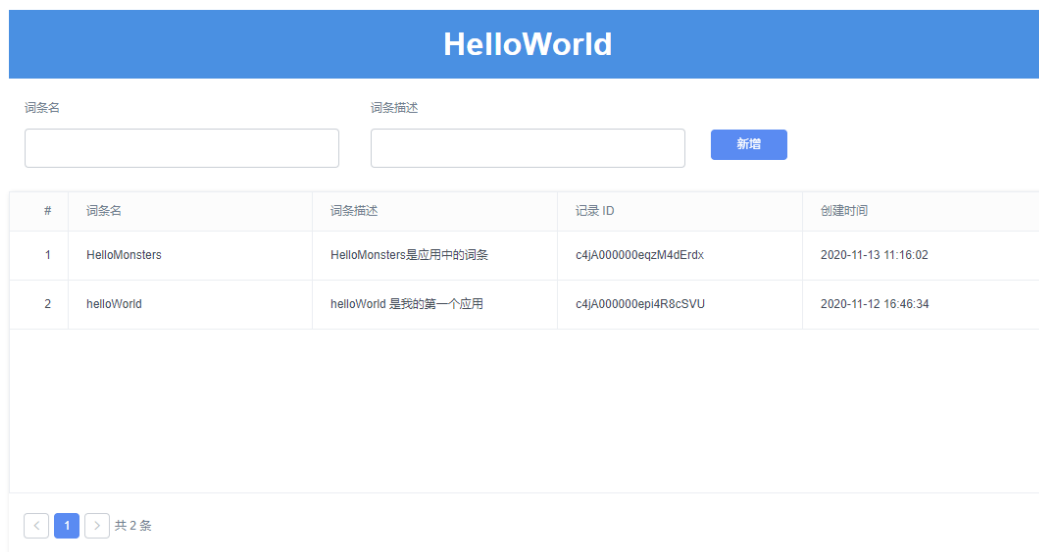
📖 说明

除了从零开始创建全新空白应用方式外，AstroZero 还支持使用应用模板快速创建应用，详细操作步骤及模板使用说明，请参见[什么是应用模板](#)。

背景信息

创建的 HelloWorld 应用主要是开发一个简单页面，在页面上的输入框中，输入一条新词条。单击“新增”，将添加的词条保存到对象模型中，并以表格形式展示出来，如[图2-170](#)所示。

图 2-170 HelloWorld 应用效果



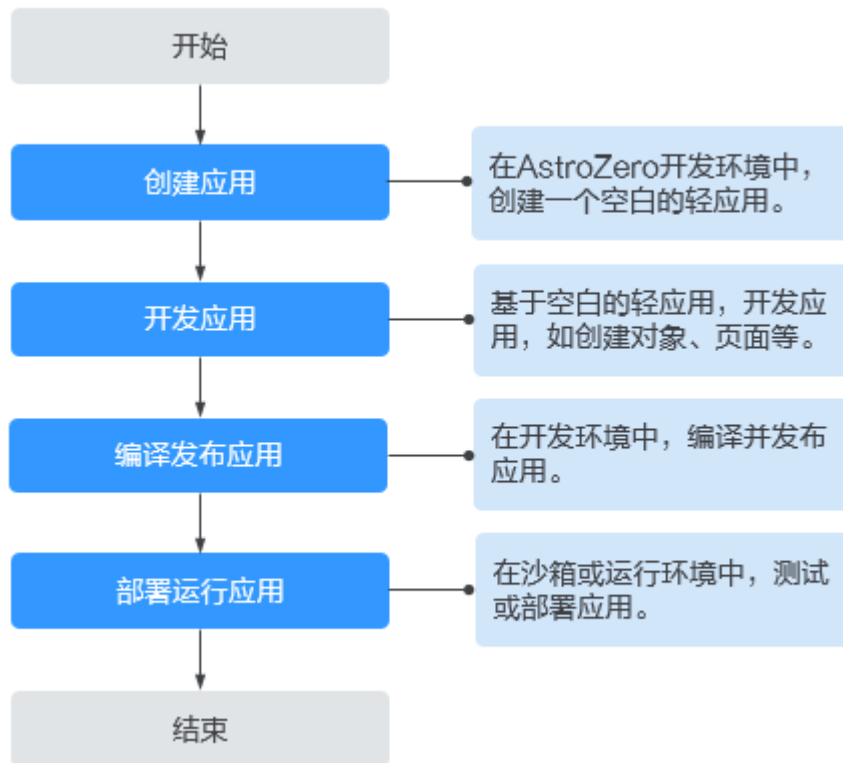
前提条件

- 已注册华为账号并开通华为云，且已完成实名认证。
- 已开通AstroZero服务，如何注册并申请试用AstroZero，请参见[注册租户](#)。

如何创建一个应用

在使用AstroZero前，您需要先了解在AstroZero创建一个应用的大致开发流程，如[图 2-171](#)所示。

图 2-171 在 AstroZero 中创建应用流程




1. 在开发环境中，**创建应用**。
如何创建应用，请参见[创建应用](#)。
2. 在开发环境中，**开发应用**。
开发应用包括开发模型、开发逻辑层以及开发前端UI层。应用开发后，您还可以根据自身需求设置应用。
如何开发应用，请参见《用户指南》中相关章节（[开发模型](#)、[服务编排](#)、[标准页面](#)和[设置应用](#)等）。
3. 在开发环境中，**编译发布应用**。
应用开发完成后，可以编译并发布应用，相关操作请参见[发布应用](#)。
4. 在沙箱或运行环境中，**部署运行应用**。
可根据自身需求，将应用发布到沙箱或运行环境。在沙箱环境中测试应用、在运行环境中部署运行应用，相关操作请参见[安装应用](#)。

步骤 1：创建应用

步骤1 使用华为账号，登录[华为云网站](#)，在顶部导航栏右侧单击“控制台”，进入华为云控制台。

步骤2 在左侧导航栏上方，单击，选择服务实例所在的区域项目。

步骤3 单击，在查找框中搜索“Astro轻应用”，单击查找到的结果，进入AstroZero服务控制台。

步骤4 在AstroZero服务控制台，单击“旧版入口”，进入AstroZero经典版开发环境。

步骤5 在AstroZero经典版开发环境首页的“项目”页签下，单击“轻应用”卡片。

图 2-172 创建轻应用



步骤6 在“轻应用”页面，单击“创建空白轻应用”。

图 2-173 创建空白轻应用



步骤7 （可选）定义命名空间。如果未提示需要创建命名空间，则直接跳过本步骤。

首次创建应用时，需要定义私有的命名空间，以区别您与其他租户的应用及应用元素，命名空间唯一且不能重复，不能修改。后续您创建的对象、脚本、服务编排、页面等的名称中，系统都会为其增加命名空间的前缀。

例如，本应用中定义的命名空间为“CNAME”，后续创建的所有对象、脚本、页面等都将以“CNAME_”开头，请您根据实际情况设置。

📖 说明

命名空间一旦设定不能修改，请慎重定义；建议您使用公司或者团队的缩写作为命名空间。

步骤8 在弹窗中，输入应用“标签”为“我的第一个应用”，“名称”为“HelloWorld”，然后单击“创建”。

应用创建完成后，自动进入应用开发页面。

📖 说明

- 如果**步骤7**中设置了命名空间，则需要先选择“应用”页签，然后再在弹窗中输入应用“标签”及“名称”。
- 在应用开发页面，如果提示“您的租户没有配置默认邮件服务器地址，将会导致在服务编排/工作流/脚本中发送邮件功能不可用”，请直接忽略，本示例不涉及。

图 2-174 创建应用

表 2-11 应用基本信息

参数名	参数说明	本示例需要填写的值
标签	应用中用于展示的文字，为了区分不同应用的描述信息，创建后可修改。	我的第一个应用。
名称	应用在系统内的唯一标识，系统会自动在该名称前添加 命名空间_ ，创建后不支持修改。 设置要求：必须以字母开头，没有连续的下划线，空格和特殊字符。	HelloWorld 说明 应用创建后，应用名称自动添加命名空间前缀 CNAME_ HelloWorld。
添加图标	可为该轻应用设置图标。	不需要设置。
分类	应用所属的分类。设置分类后，工程列表和库列表都可以根据应用或BO的分类进行筛选。	不需要设置。

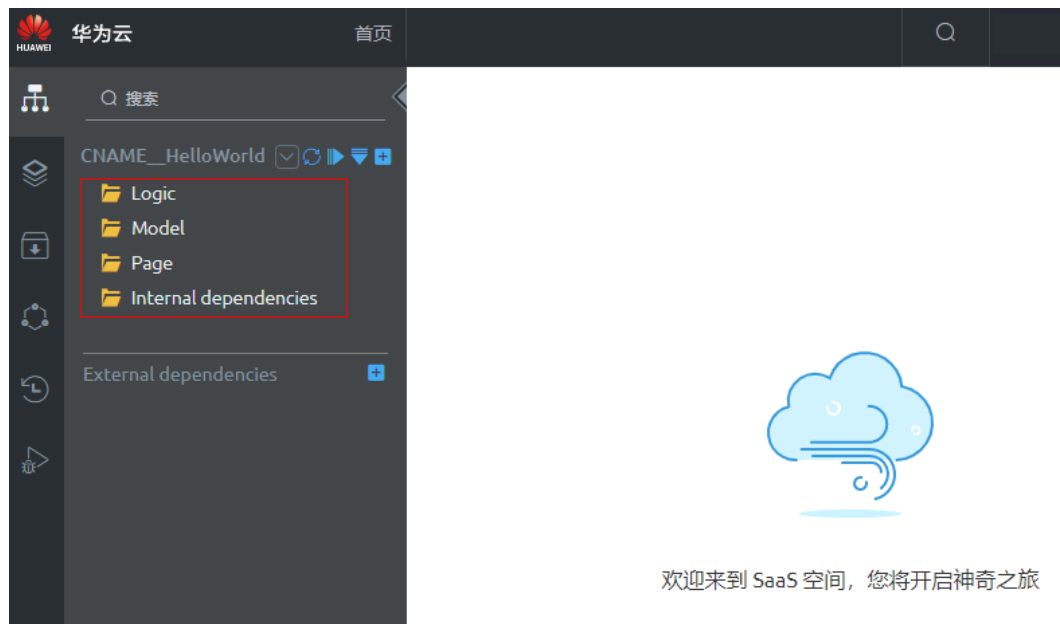
参数名	参数说明	本示例需要填写的值
描述	轻应用的描述信息。	不需要设置。
高级设置 > 运行时 版本	开发的资产包依赖所选择的运行时版本，如果线下运行版本不一致，可能产生不兼容。	保持默认。

轻应用创建完成后，将自动跳转到应用开发界面，如图2-175所示，轻应用中有4个默认目录。默认目录是AstroZero推荐使用的应用文件目录结构，后续开发的页面、对象及后台逻辑，建议按照以下对应关系进行创建，以便应用目录管理。

- Model: 数据对象
- Logic: 服务编排、脚本等
- Page: 前端页面
- Internal dependencies: 内部依赖，例如BO、原生服务

您也可以根据自己的开发习惯，单击已创建应用右侧的“+”，添加新的应用目录，自定义放置应用资源。

图 2-175 轻应用默认目录



说明

- 在应用开发界面，单击页面顶部的“首页”，可以返回开发环境首页。
- 如果想再次进入应用，请在“首页 > 项目 > 我的应用”下，单击应用名，例如“我的第一个应用”，即可进入应用开发界面。

----结束

步骤 2: 开发应用-创建对象

📖 说明

【思考】什么是对象？

对象是AstroZero的核心功能之一，对象相当于传统数据库里的一张表，用于持久化业务数据。AstroZero中有内置的标准对象，也有租户开发者创建的自定义对象。


租户角色的开发者，创建一个自定义对象后，可在前端页面使用页面模型绑定此对象，并使用表格组件，直接展现对象数据。也可以使用脚本、自定义JS代码、服务编排等，对对象中的信息进行增删改查（AstroZero中常用的使用方式）。

根据HelloWorld应用设计的功能，首先您需要先创建一个存放词条的对象 helloObject，用于保存词条名、词条描述、创建日期信息。对象helloObject中使用到的字段，如表2-12所示。

表 2-12 自定义对象 helloObject 字段详情

字段标签	字段名称	数据类型	是否唯一	是否必填	建议长度
词条名	name（预置的标准字段） 说明 自定义对象具体预置了哪些标准字段，请在 了解自定义对象中系统预置的标准字段 查看。	Name	是	是	-
创建时间	createdDate（预置的标准字段，不需要修改）	日期/时间	-	是	-
词条描述	detail	文本	否	是	255

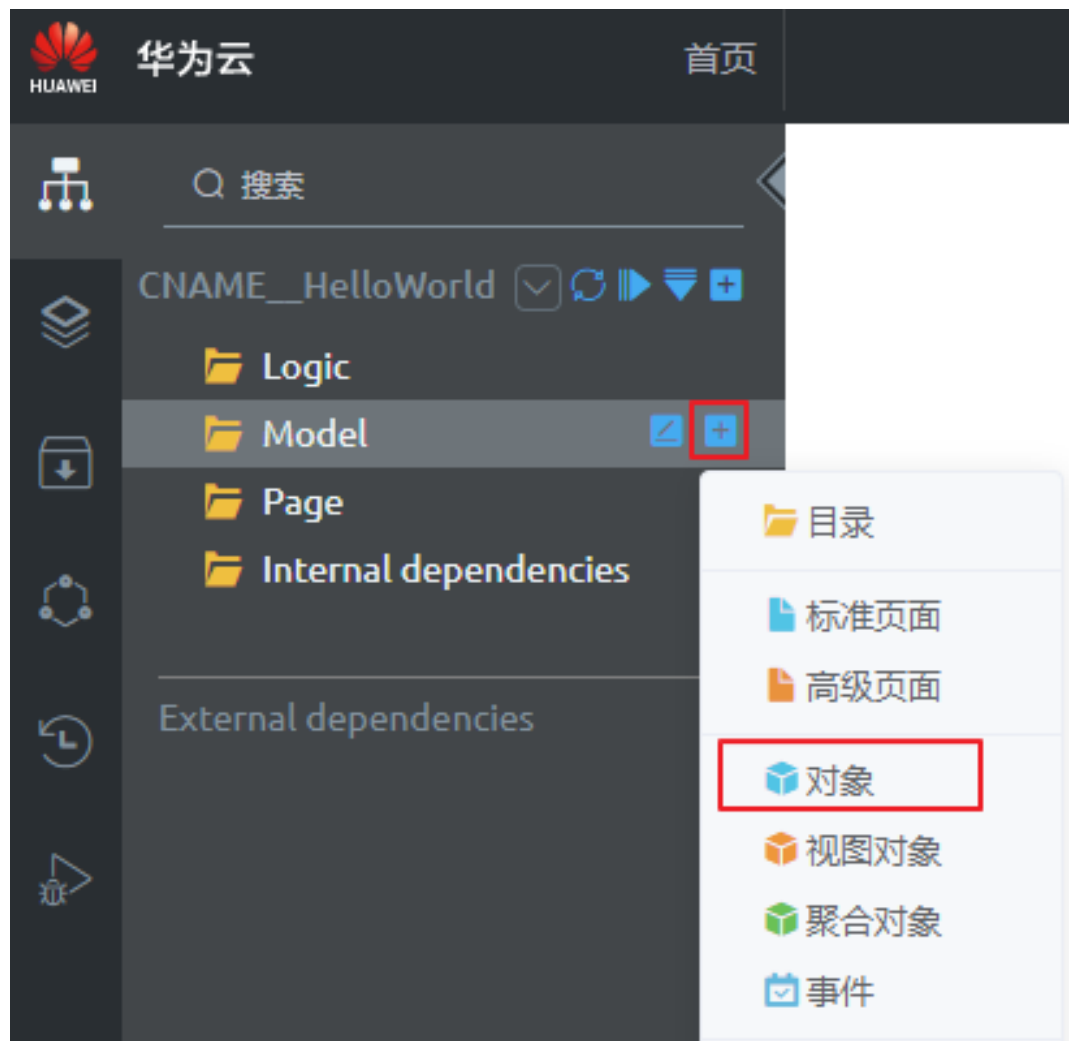
步骤1 在“HelloWorld”应用中，在规划存放对象的目录下，创建对象。

将光标放在“Model”目录上，单击，选择“对象”。

📖 说明

本示例中，使用默认“Model”目录存放对象，因此需要在“Model”目录下进行创建对象操作。对象添加完成后，系统默认会将对象放在“Model”目录下，您也可以根据个人喜好选择其他目录创建。

图 2-176 创建对象



步骤2 输入对象的“标签”为“helloObject”，单击“名称”的输入框后，系统将自动生成名称为“helloObject”，再单击“添加”。

图 2-177 添加对象

对象创建完成后，自动进入对象详情页面。

说明

【技巧】如果输入“标签”的值，首字符为英文字母，单击“名称”的输入框，系统会自动生成与“标签”对应的“名称”，也可以根据需要修改这个自动生成的“名称”。

表 2-13 创建对象的相关信息

参数名	参数说明	本示例需要填写的值
创建新对象	添加对象的方式。	保持默认，选中“创建新对象”。
标签	对象展示的名称，为了区分不同对象的描述信息，创建后可修改。	helloObject
名称	对象在系统内的唯一标识。 <ul style="list-style-type: none"> 对象创建后，系统会自动在“名称”增加租户命名空间前缀，以及增加“_CST”后缀，“_CST”是AstroZero中对象的后缀标识。 对象创建完后，“名称”不可以修改。 	helloObject 说明 对象创建后，系统自动为对象名称添加前后缀，实际创建的对象名为 CNAME_helloObject_CST。

步骤3 查看并修改对象的标准字段。

1. 查看标准字段，了解自定义对象中系统预置的标准字段。

自定义的对象创建后，系统将自动创建一些标准字段，供您使用。您可以在“自定义对象”的详情页面中，单击“标准字段”页签，查看自定义对象的标准字段。标准字段如图2-178所示，其中“name”和“createdDate”两个字段，在后续开发中将会用到。

图 2-178 标准字段

自定义对象: [CNAME_helloObject_CST](#)

基本信息 **标准字段** 自定义字段 验证规则 内嵌触发器 布局

名称	标签	数据类型	是否索引	是否必填	描述	最后修改人	最后修改时间	操作
id	记录 ID	ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	无描述信息			
name	Name	名称	<input type="checkbox"/>	<input type="checkbox"/>	无描述信息			
createdBy	创建人	查找关系 (User,PortalU...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	无描述信息			
createdDate	创建时间	日期时间	<input type="checkbox"/>	<input checked="" type="checkbox"/>	无描述信息			
lastModifiedBy	最后修改人	查找关系 (User,PortalU...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	无描述信息			
lastModifiedDate	最后修改时间	日期时间	<input type="checkbox"/>	<input type="checkbox"/>	无描述信息			
owner	所有人	查找关系 (User,PortalU...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	无描述信息			
currencyIsoCode	币种	文本	<input type="checkbox"/>	<input type="checkbox"/>	无描述信息			
installedPackage	已安装的软件包	查找关系 (PackageInstall)	<input type="checkbox"/>	<input type="checkbox"/>	无描述信息			

共 9 条 10条/页 < 1 > 前往 1 页

2. 修改标准字段“name”。

单击“name”字段右侧的，如图2-178所示，进入字段编辑页面。在“name”字段编辑页面中，修改“标签”为“词条名”，并勾选“是否唯一”，“是否必填”，然后单击“保存”，如图2-179所示。

图 2-179 修改 name 字段

字段详情: [name](#)

基本信息

* 标签	名称	字段帮助	字段描述
词条名 1	name	请输入	请输入

[展开](#)

详细信息

数据类型	* 数据长度	是否唯一	是否必填
名称	64	<input checked="" type="checkbox"/> 2	<input checked="" type="checkbox"/> 3

4 [保存](#) [取消](#)

说明

【思考】为什么要修改“name”字段，其他标准字段可以修改吗？

因为“name”的默认属性，没有完全符合HelloWorld应用对“词条名”的要求，例如“标签”，“是否唯一”这些字段属性，因此需要自定义修改一下，修改后“name”作为存放“词条名”的字段。

标准字段中目前只有“name”字段支持修改，而“createdDate”字段直接使用，不能修改。

步骤4 创建一个自定义字段“词条描述”，作为应用中存放词条描述信息的字段。

1. 在“自定义字段”页签，单击“新建”，进入创建字段页面，设置字段类型为“文本”，单击“下一步”。

图 2-180 新建自定义字段



说明

自定义字段可以手动创建，也可以批量导入自定义字段，具体操作请参见[如何使用对象的字段模板批量创建自定义字段](#)章节。

2. 输入新字段详细信息，单击“下一步”。
设置字段“标签”为“词条描述”、“名称”为“detail”、字段长度“255”，并勾选“是否必填”、“大小写敏感”、“是否可搜”。

图 2-181 定义“设备编码”详细信息

新建字段



📖 说明

字段“名称”为“detail”，实际创建的字段，系统将自动添加前后缀“CNAME_detail_CST”，“CNAME_”前缀由租户命名空间决定，“_CST”后缀代表是自定义字段。

3. 设置字段级的访问权限，即选中“读取”和“编辑”复选框，为所有预置profile配置能编辑和读取本字段的权限，单击“下一步”。

📖 说明

在AstroZero中，预置了一些权限集，并提供了自定义对象中字段级的安全控制，这些权限配置是由管理员预置的，权限集及设置的更多详细介绍，请参见[了解权限配置](#)。

4. 将字段添加到对象的页面布局：选中“添加本字段到该页面布局”，单击“保存”。

在AstroZero中，如果某个自定义字段在创建时，设置了添加到页面布局，那么在该对象的页面布局的页面表格中，将可以对该字段进行增删改查等操作。

📖 说明

【技巧】自定义对象的页面布局，可以直接操作对象，进行增删改查等操作，更为方便的是可以作为前端页面直接使用，对象页面布局更多详细介绍，请参见[编辑对象的页面布局](#)。

---结束

步骤 3：开发应用-创建服务编排

📖 说明

【思考】什么是服务编排？

服务编排功能是AstroZero的核心功能之一，即是在可视化、模板化的服务编排页面，通过拖拽图元，并配置图元，再拖拽相关参数，就可以实现特定的业务逻辑的一种开发方式。

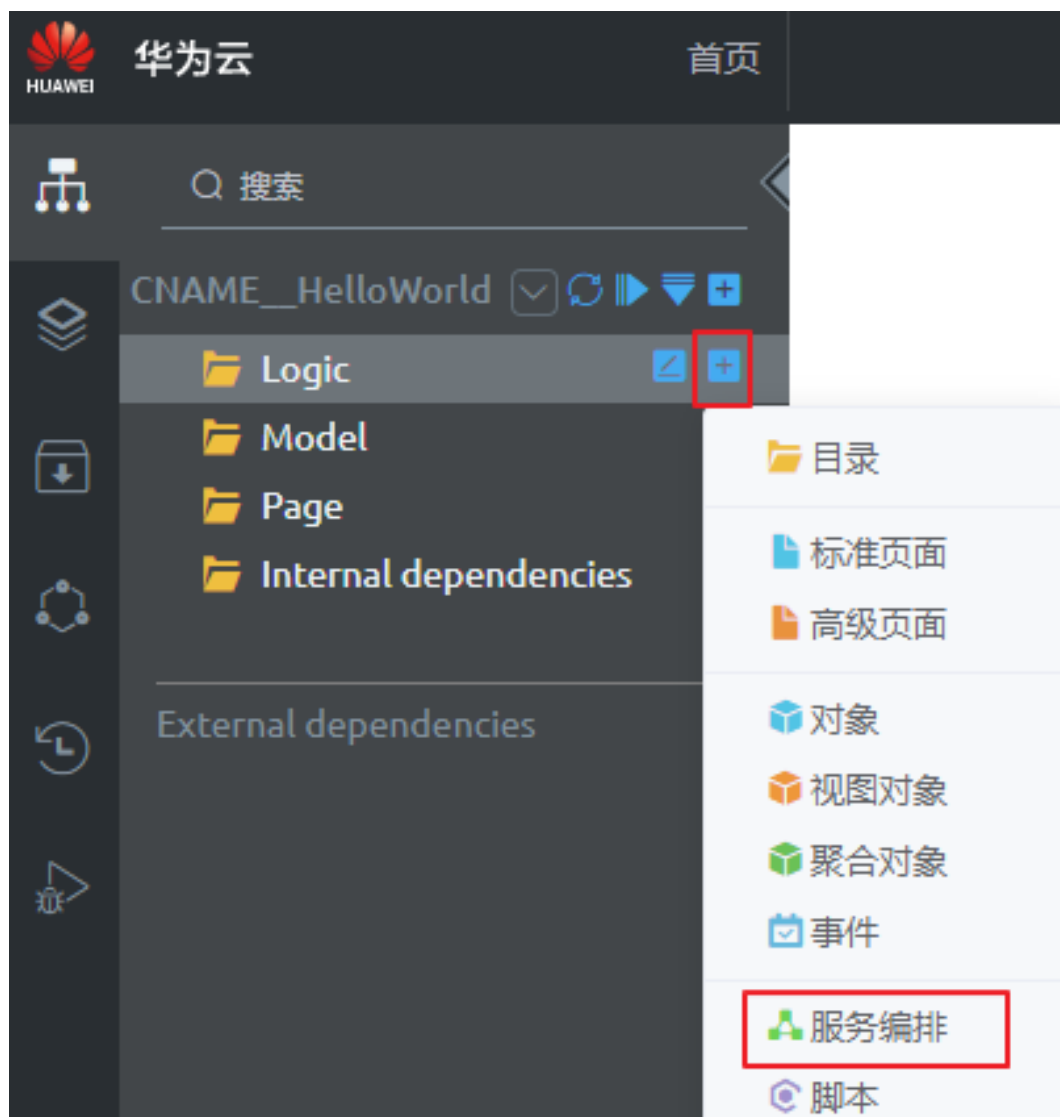
服务编排类似于编程中一段有流程、条件处理、判断逻辑的程序。这段程序有输入参数和输出参数，可以独立成为一个对外调用的方法。同时，在服务编排内部，也可以调用其他的方法。

本示例中，创建服务编排目的是操作一个[存放词条的自定义对象](#)，实现将服务编排中获得的入参信息，分别以“词条名”、“词条描述”字段写入到词条对象helloObject中。如果您还没有创建，请查阅[创建存放词条的自定义对象](#)章节，进行创建。

步骤1 在规划存放服务编排的目录“Logic”下，创建服务编排。

将光标放在“Logic”目录上，单击右侧“+”。在弹出的菜单中，选择“服务编排”。

图 2-182 创建服务编排




步骤2 在对话框中，输入服务编排的“名称”为“helloFlow”，单击“标签”的输入框后，系统将自动生成名称为“helloFlow”，再单击“添加”。

图 2-183 添加服务编排

表 2-14 服务编排基本信息

参数名	参数说明	本示例需要填写的值
创建一个新的服务编排	创建一个新的服务编排流程。	保持默认，选中“创建一个新的服务编排”。
使用已有的服务编排	基于已有的服务编排创建。方便用户快速继承已有服务编排的配置，提高效率。	不设置。
标签	服务编排用于展示的名称，创建后可修改。	helloFlow
名称	服务编排在系统内的唯一标识，创建后不可修改。 系统会自动在该名称前添加 命名空间 __。当其他功能调用服务编排时，调用的是服务编排的“名称”，而不是标签。	helloFlow 说明 实际创建的服务编排名为： CNAME__helloFlow
类型	服务编排的类型。 <ul style="list-style-type: none"> Autolaunched Flow：自启动服务编排，在接口调用后会立即执行服务编排模型定义的逻辑。 Event Trigger：事件触发的服务编排，在事件触发时才会开始执行服务编排模型定义的逻辑。选择该类型时，需要配置具体的事件以及条件规则。 	保持默认类型： Autolaunched Flow

步骤3 定义变量，作为后续获取前端页面的输入值，同时也是操作词条对象helloObject的字段。

单击，展开“全局上下文”，再单击“变量”后的“+”，修改“variable0”为“name”；再次单击“+”，创建变量“detail”。这里创建的“name”、“detail”分别对应词条对象helloObject中的“词条名”、“词条描述”字段。

说明

“全局上下文”是定义服务编排中使用到的变量列表区域，服务编排中支持定义变量、变量集、常量、结构体等参数，具体变量创建方式及变量说明请参见[配置服务编排变量以及输入输出参数](#)。

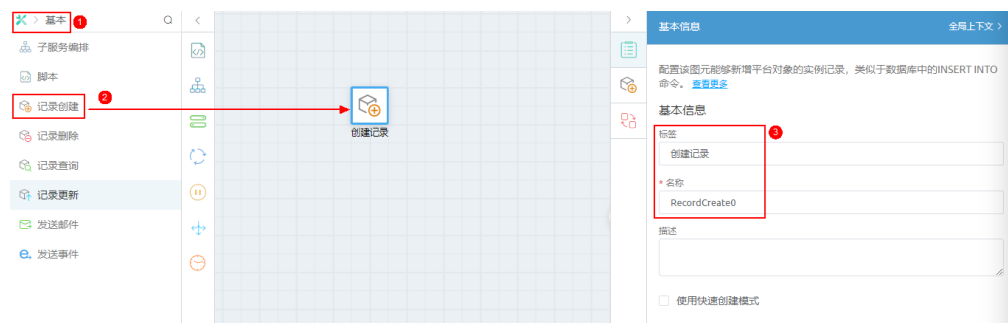
步骤4 在服务编排中，添加并配置“记录创建”图元。

说明

“记录创建”图元是通过关联当前应用（HelloWorld）中已存在的一个对象（词条对象helloObject），然后通过设置对应的操作字段（name、CNAME_detail_CST），实现向这个对象中增加记录的功能。AstroZero预置了多种逻辑图元，详细图元分类及功能介绍，请参见[图元介绍](#)。

1. 在左侧图元列表中，单击“基本”，拖拽“记录创建”图元到画布中，“标签”修改为“创建记录”，“名称”保持默认。

图 2-184 拖拽创建记录



2. 单击，配置“记录创建”图元。

先单击“对象”下拉框，选中词条对象“CNAME_helloObject_CST”。对象设置后，“赋值”下自动显示对象下可以被赋值的字段（name、CNAME_detail_CST）；再单击“全局上下文”，然后从全局上下文中，分别拖拽变量“name”、“detail”到对应“源”输入框中。关联之后，变量从前端获取到值后，将会传给对象中的字段。

图 2-185 拖拽需要修改对象字段



步骤5 定义输入参数，并保存服务编排。

这里的入参、出参指的是整个服务编排的入参、出参，具体参数需要从“全局上下文”列表中拖拽。


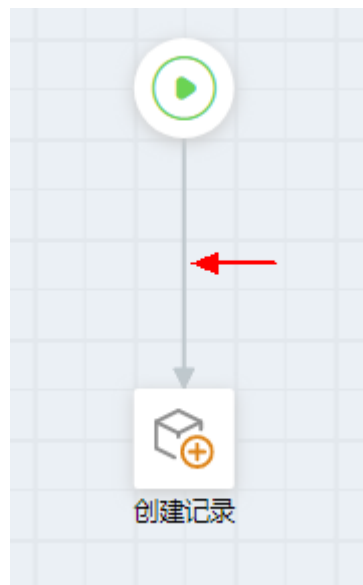
1. 在画布上，把鼠标放在起点图元上，从“+”拖动鼠标，在起点图元和“创建记录”图元间增加连线。


图 2-186 连接图元




2. 鼠标在画布空白处点一下，单击右侧，设置服务编排的输入参数。

图 2-187 拖拽服务编排的输入参数“name”、“detail”



3. 单击服务编排界面上方的, 保存服务编排。


步骤6 测试服务编排能否正常执行。

1. 单击服务编排编辑器上方的, 进入执行服务编排页面。
2. 在输入参数中, 输入测试数据, 单击“运行”。

```
{  
  "name": "helloWorld",  
  "detail": "helloWorld 是我的第一个应用"  
}
```

执行成功, 界面上会返回设备对象中的全部信息, 样例如下:

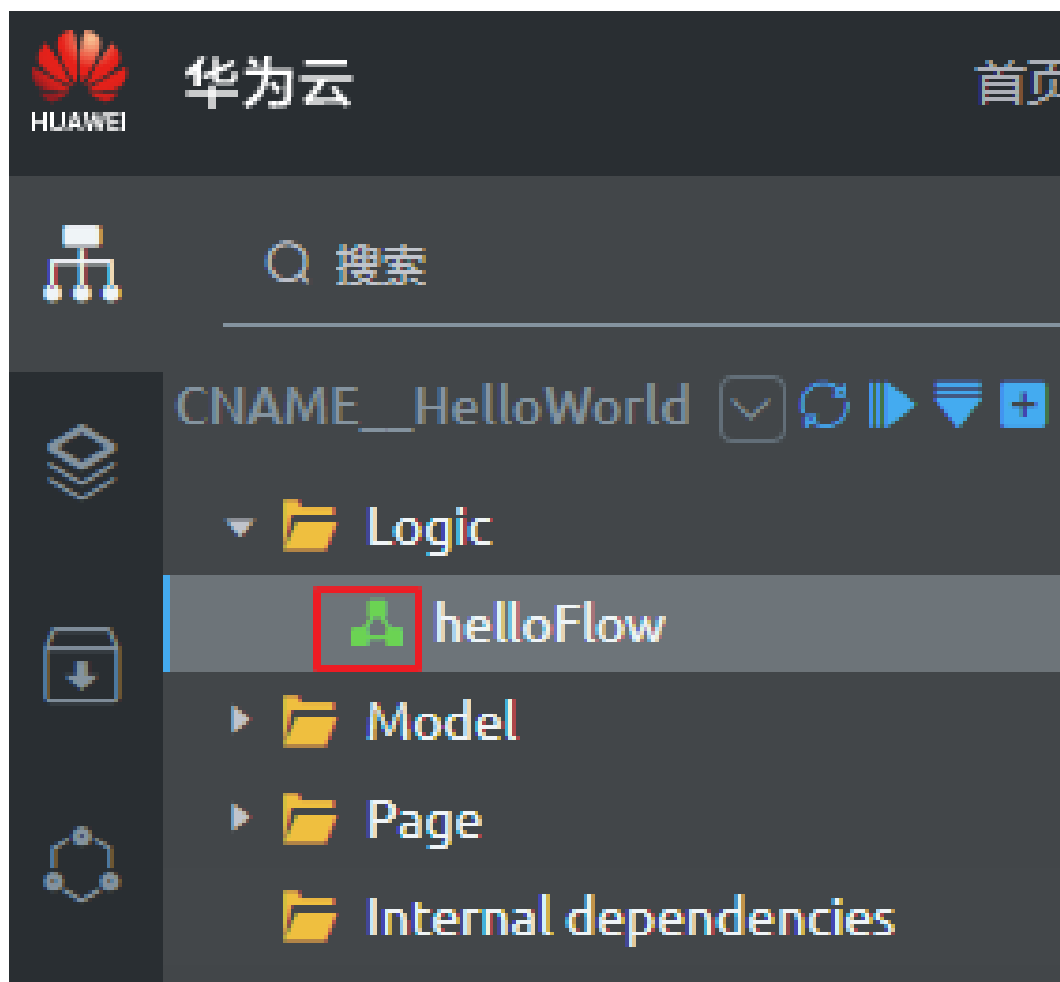
```
{  
  "interviewId": "002N000000edyfrHJiLV",  
  "outputs": {}  
}
```

步骤7 启用服务编排: 测试成功, 单击服务编排编辑器上方的, 启用服务编排。

须知

如果服务编排创建完成后, 未启用默认是属于编辑状态, 不能被页面调用。

图 2-188 服务编排启用后高亮状态



---结束

步骤 4：开发应用-创建页面

📖 说明

【思考】AstroZero中有标准页面、高级页面和大屏页面三种类型的前端页面，那么什么场景下使用标准页面？

标准页面常用于业务数据的增、删、改、查，并且对页面样式要求相对简单的场景。标准页面的优势在于，只需要拖、拉、拽页面组件，配置组件属性，绑定页面模型，再加上少量事件代码，即可快速拼装出一个完整的页面。

HelloWorld应用的业务功能主要是通过**标准页面**呈现出来的。通过使用标准页面，将对象、服务编排与页面中的组件、事件代码关联、绑定和调用，实现将前端输入的数据信息传递到对象中，并最终再通过页面呈现出来。

标准页面中使用的词条对象helloObject及服务编排helloFlow需要提前创建。如果您还没有创建，请先参考**词条对象**和**服务编排**中操作创建。

- 步骤1** 将鼠标放在规划存放标准页面的目录上（如Page），单击右侧“+”，选择“标准页面”。
- 步骤2** 在添加标准页面对话框中，输入新建标准页面的“名称”和“标签”，单击“添加”。

图 2-189 添加标准页面

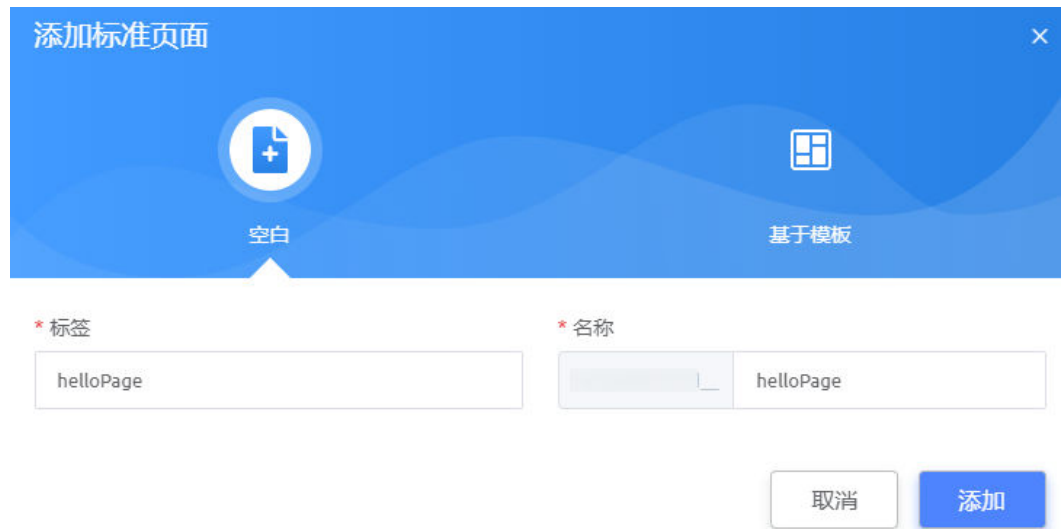


表 2-15 标准页面基本信息

参数名	参数说明	本示例需要填写的值
标签	标准页面用于展示的名称，创建后可修改。	helloPage
名称	标准页面在系统内的唯一标识，创建后不支持修改。 系统会自动在该名称前添加 命名空间 。当其他功能调用标准页面时，调用的是标准页面的“名称”，而不是标签。	helloPage 说明 实际创建的标准页面名称为CNAME_helloPage

步骤3 创建页面模型。

📖 说明

AstroZero的标准页面是通过数据模型驱动的，页面所有的逻辑都是围绕数据模型展开的，前端组件与对象及服务编排关联都需要创建页面模型。

1. 创建页面**对象模型**“userInfo”，关联自定义对象“CNAME_helloObject_CST”，后续将绑定页面表格。
 - a. 在页面底部单击“模型视图”，在“模型视图”中，单击“新增模型”。

图 2-190 新增模型



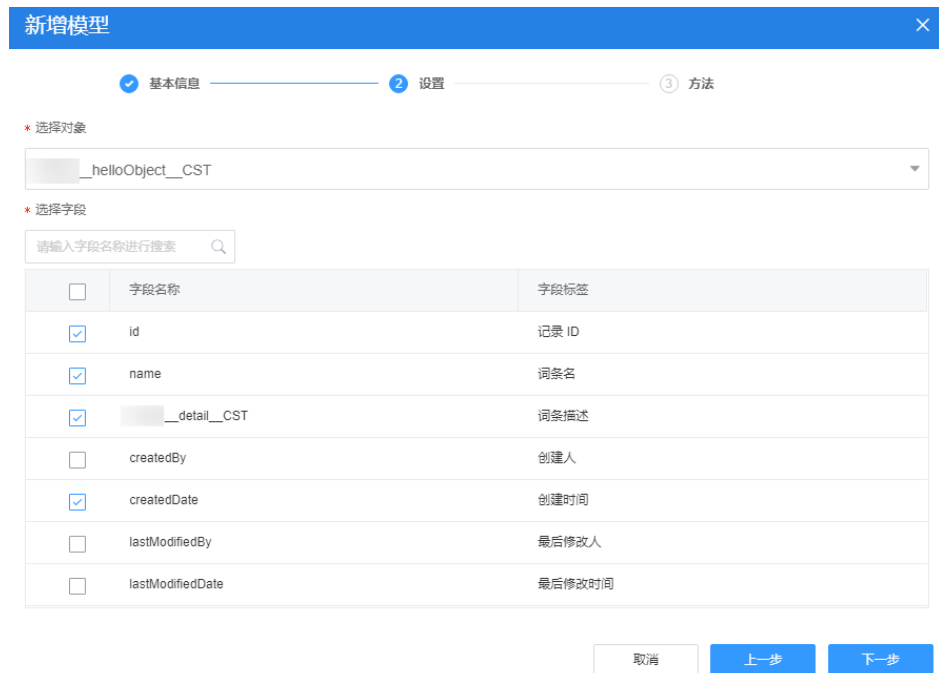
- b. 设置“模型名称”为“userInfo”，“来源”选择“对象”，单击“下一步”。

图 2-191 选择模型来源



- c. 设置“选择对象”为“CNAME_helloObject_CST”，并在“选择字段”中，勾选“name”、“id”、“CNAME_detail_CST”、“createdDate”，单击“下一步”，再单击“确定”。

图 2-192 选择字段




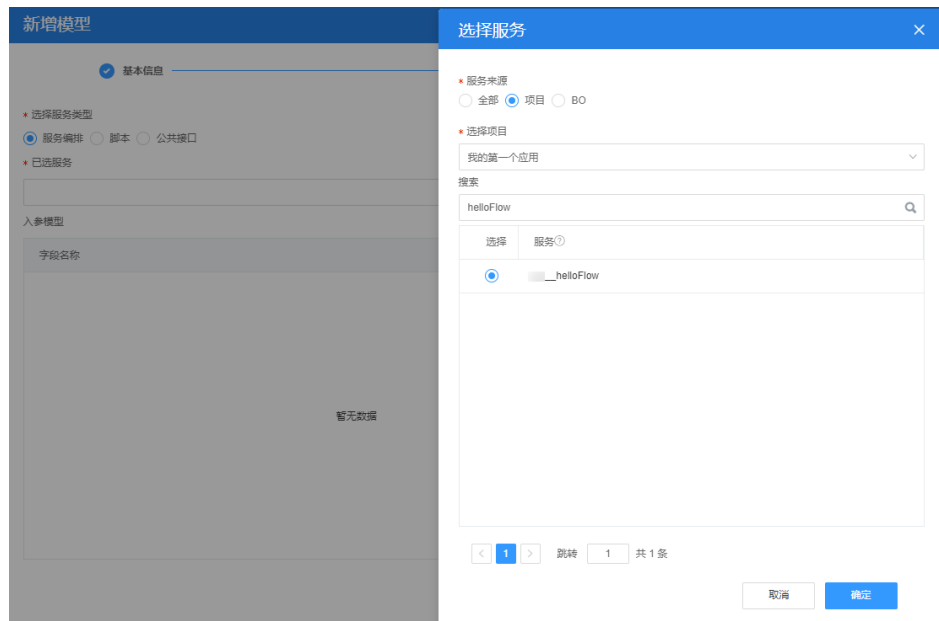
- d. 单击页面上方的，保存设置。
2. 创建页面**服务模型**“helloFlow”，关联服务编排“CNAME_helloFlow”，服务编排的输入值，后续将绑定页面上表单中的输入框。
 - a. 在“模型视图”中，单击“新增模型”。
 - b. 设置“模型名称”为“helloFlow”，“来源”选择“服务”，单击“下一步”。


图 2-193 新建 helloFlow



- c. 设置“选择服务类型”为“服务编排”，并在弹窗中，“服务来源”选择“项目”，“选择项目”设置为“我的第一个应用”，“搜索”中输入“helloFlow”，在搜索结果中选择“CNAME_helloFlow”，然后单击“确定”。如果未搜索到服务编排，请检查服务编排是否已启用。

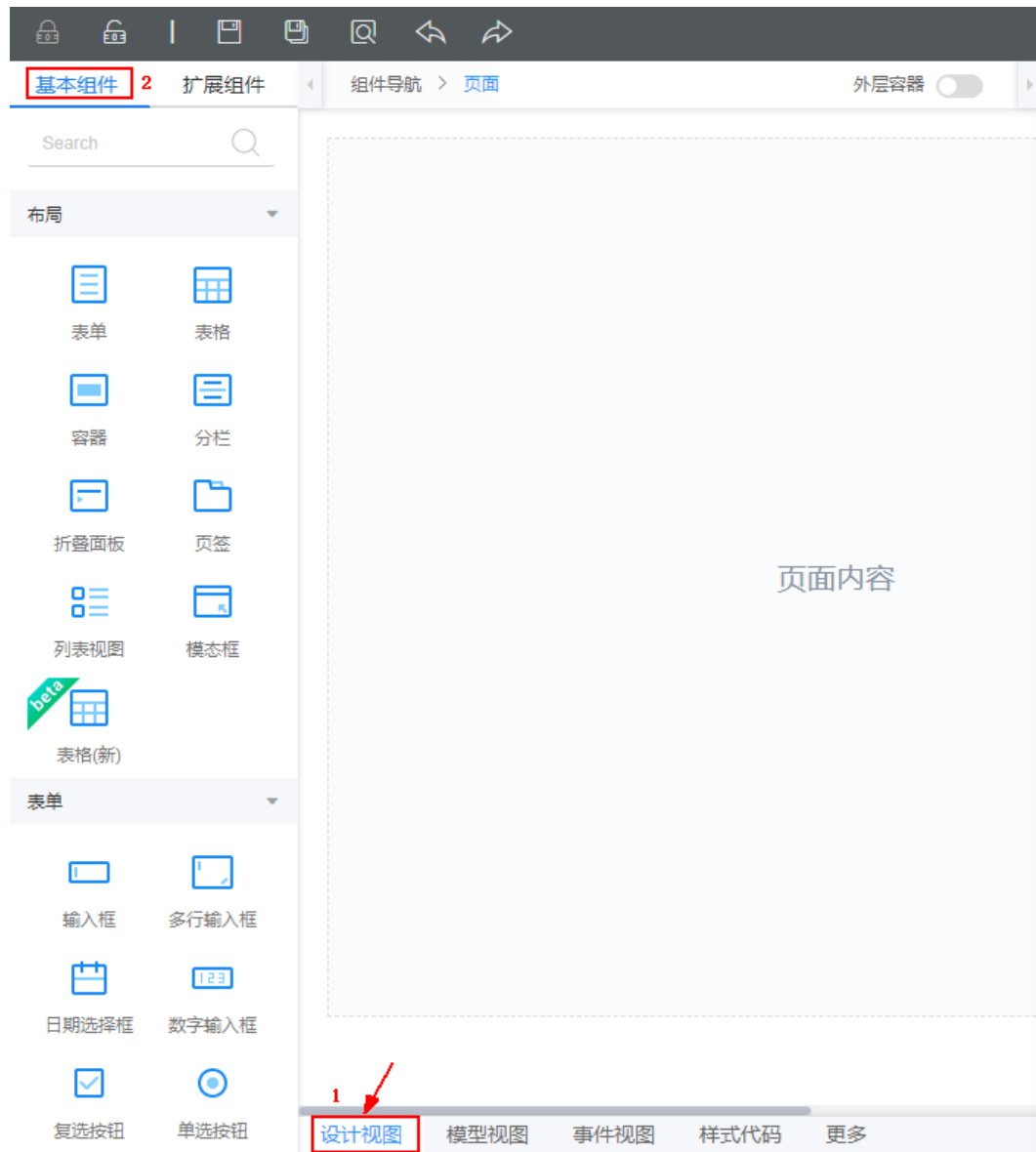
图 2-194 选择服务



- d. 单击“下一步”，再单击“确定”。
- e. 单击页面上方的, 保存设置。

步骤4 单击页面底部的“设计视图”，切换到页面设计视图，在左侧上方单击“基本组件”，展开组件列表，方便拖拽组件到页面。

图 2-195 展开组件列表



📖 说明

标准页面有设计视图、模型视图、事件视图、样式代码以及更多中的组件代码、第三方库等，详细介绍及用法请参见[标准页面](#)。

步骤5 拖拽组件，组装页面。

1. 拖拽一个“标题”组件到右侧页面中，并设置“标题”组件属性。

📖 说明

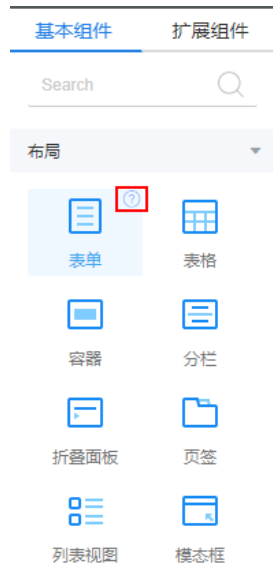
在“设计视图”中，拖拽一个页面组件到页面中，选中组件后，在右侧“属性”页签，即可设置组件的绑定数据、样式等属性。

- a. 修改“标题内容”为“HelloWorld”。
- b. 在“样式”属性中，设置标题“字体”颜色为白色、大小“25px”、居中显示，并设置标题背景为蓝色“#4A90E2”。

说明

在组件列表区，将光标放在组件上，组件右上角将显示帮助信息的问号图标。单击问号图标，即可进入组件介绍页面。在组件介绍页面，您可以了解并学习这些预置的前端组件的使用场景及参数配置方法，组件说明详细介绍，如图2-196。

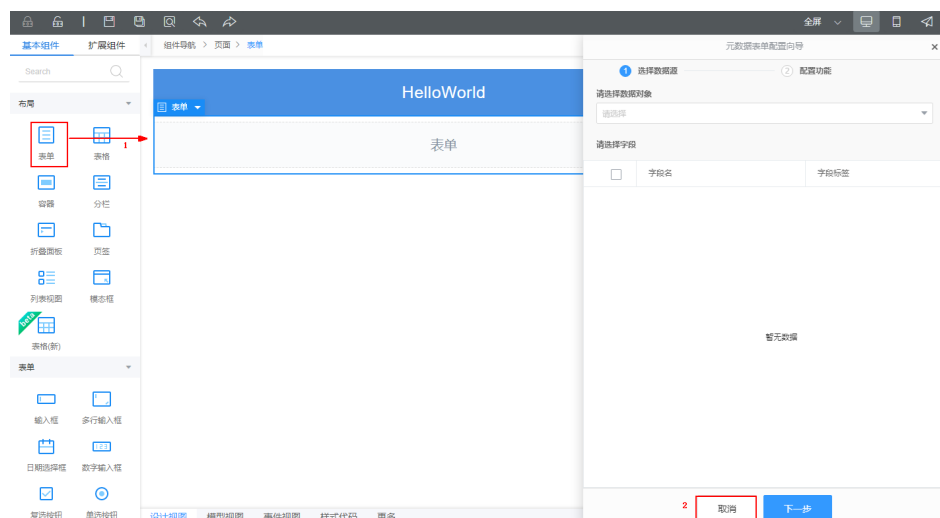
图 2-196 查看组件帮助信息



2. 拖拽一个“表单”组件到“标题”组件下，然后在数据绑定弹窗底部，单击“取消”，创建一个空表单。

因为现在还不需要定义数据源，因此需要单击“取消”，创建一个空的表单组件。

图 2-197 拖拽表单



3. 从左侧组件列表最底部“高级”中，拖拽一个“栅格容器”到“表单”。栅格容器默认是1个分栏，2个栏，选中“分栏”，然后在属性的“行布局”下，单击PC


下的 ，将分栏设置为3栏。

图 2-198 拖拽栅格容器并设置为 3 栏

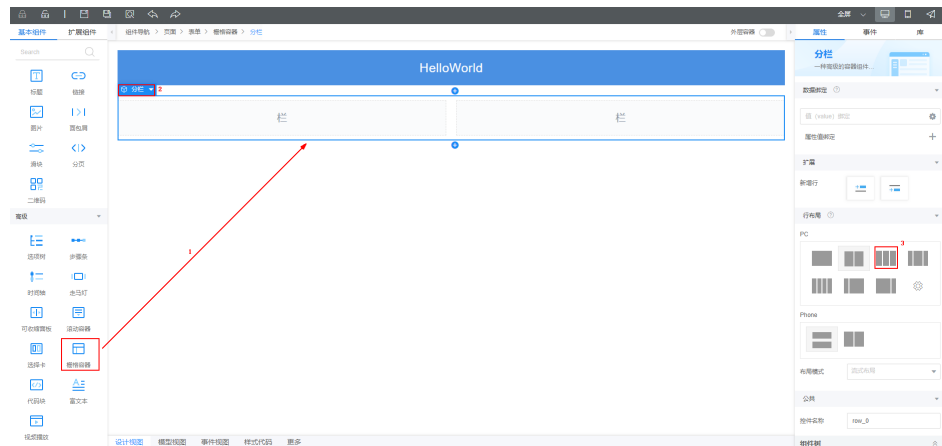
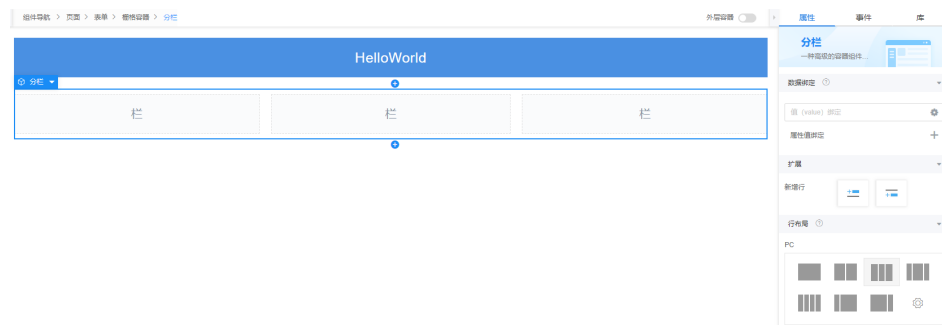


图 2-199 设置后显示为 3 栏

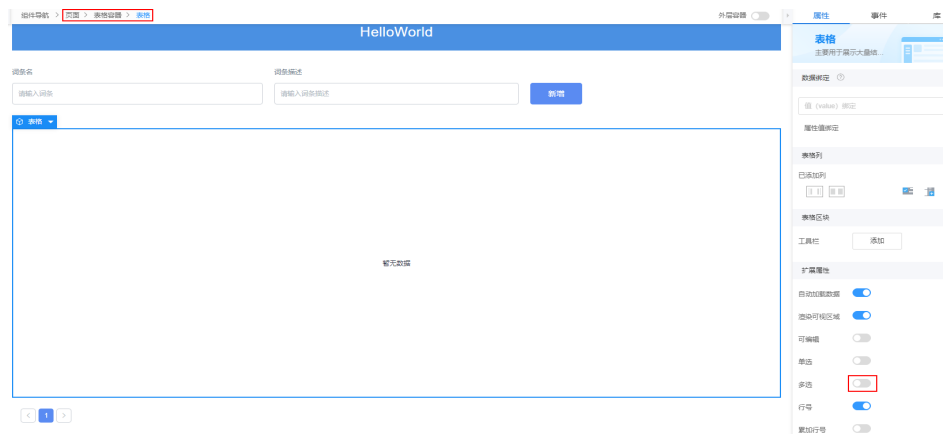



说明

栅格容器是一种灵活的容器类页面布局，可以将页面进行横向和纵向分割。

4. 向第1栏、第2栏分别拖拽一个“输入框”，第三栏中拖拽一个“按钮”组件。
5. 选中第1栏中的“输入框”，设置“标签”为“词条名”，“占位符”为“请输入词条”。
6. 选中第2栏中的“输入框”，设置“标签”为“词条描述”，“占位符”为“请输入词条描述”。
7. 选中“按钮”，设置“显示名称”为“新增”，“类型”为“主要按钮”，然后选中按钮所在“栏”，设置“样式”中“布局”上边距为“20px”。
8. 拖拽一个“表格”到“表单”下（表单之外），并选中“表格容器”下“表格”，在右侧属性最下端，关闭“多选”选项。

图 2-200 拖拽表格



9. 单击页面上方的 ，保存页面。

步骤6 绑定组件与页面模型，将前端页面与对象及服务编排关联。


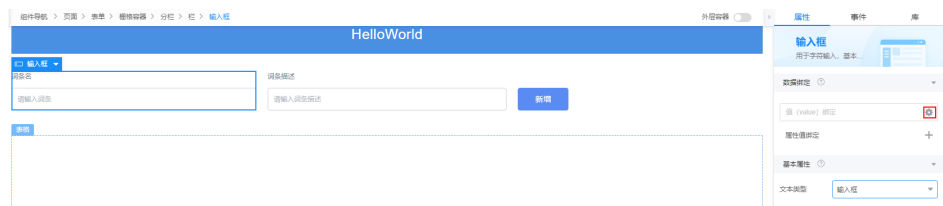
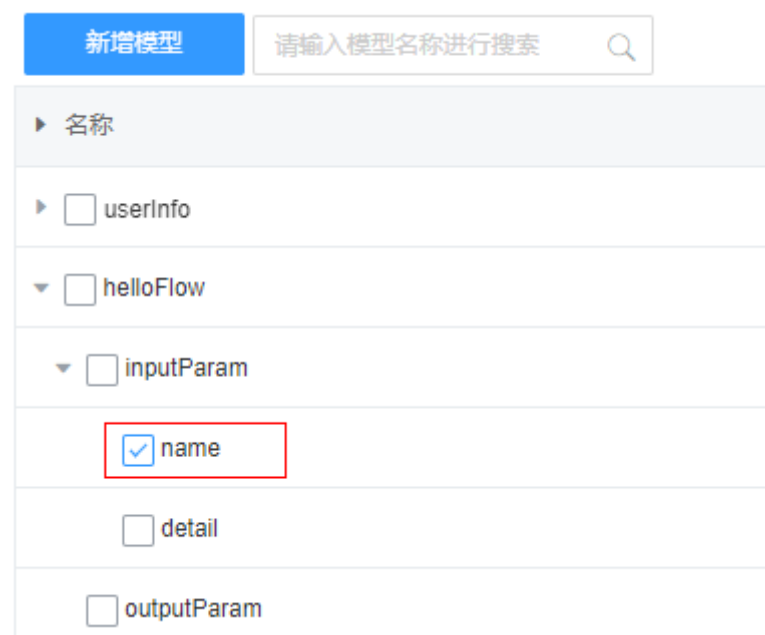
1. 选中“词条名”输入框，在右侧“数据绑定”旁，单击 ，进入选中模型弹窗。

图 2-201 单击设置图标



2. 在选择模型弹窗中，勾选“helloFlow”下“inputParam”的“name”，单击“确定”。

图 2-202 选中模型字段





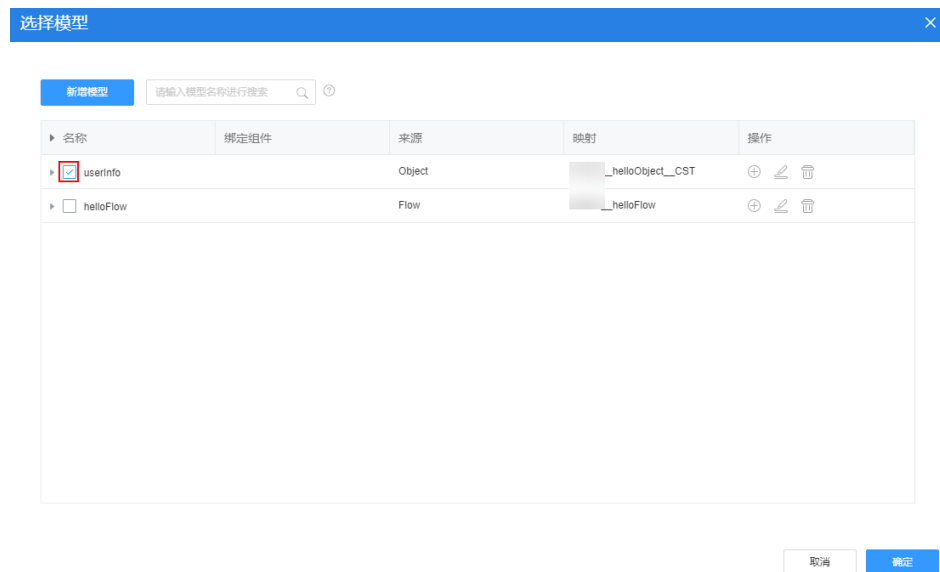
- 同样方式，为“词条描述”绑定模型字段：选中“词条描述”输入框，在右侧“数据绑定”下，单击，进入选中模型弹窗，勾选“helloFlow”下“inputParam”的“detail”，单击“确定”。
绑定之后，由页面“词条名”、“词条描述”输入框输入的值将作为服务编排的输入参数“name”、“detail”值，当页面调用服务编排时，即可将前端页面输入的值写入到词条对象helloObject中。
- 选中“表格容器”下的“表格”，在右侧“数据绑定”下，单击，然后在选择模型弹窗中，勾选“userInfo”，单击“确定”。

图 2-203 选中对象模型 helloObject



表格绑定页面对象模型后，系统自动将对象字段做为表格的列名显示。如果对象模型关联的自定义对象（这里是词条对象helloObject）中，有词条记录，将会自动查询并显示在页面表格里。在表格的属性“表格列”中，可调整列顺序。

图 2-204 绑定页面模型后



说明

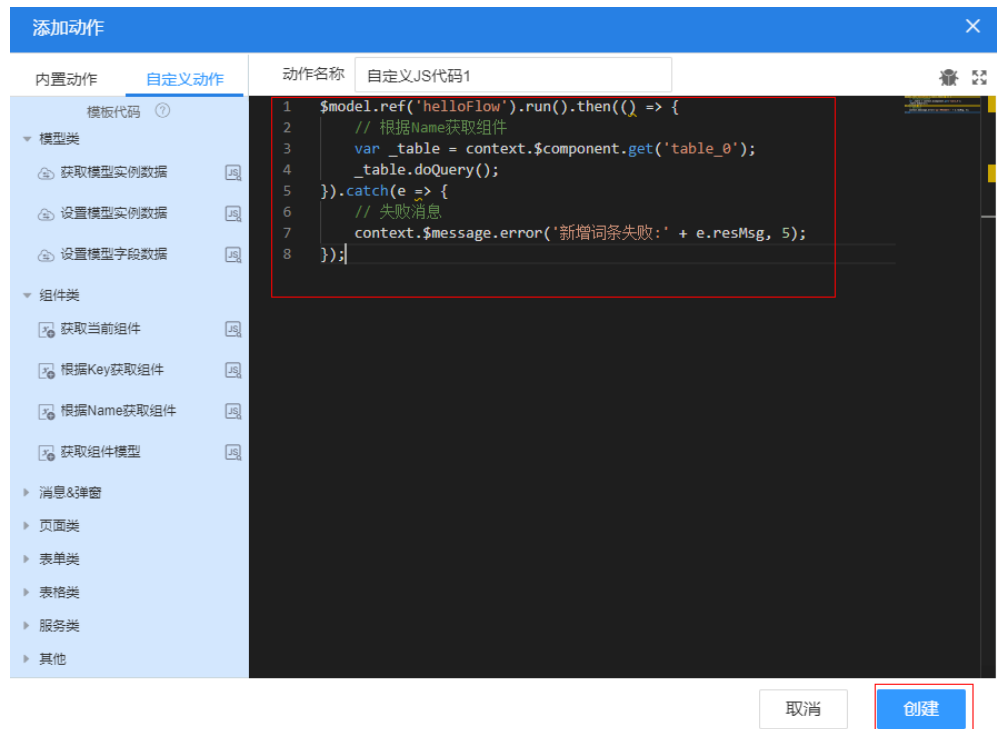
【技巧】表格组件绑定页面模型后，可以自动添加页面模型里面的字段为表格列，也可以将模型的字段作为表格列查询的条件，您可以在表格组件的属性中“表格列”及“表格区块”中设置。

步骤7 添加按钮事件。

1. 选中“新增”按钮，在右侧属性区中，单击“事件”页签。
2. 单击“点击”后的“+”，进入“添加事件”弹窗。
3. 在“自定义动作”中，输入如下事件代码，然后单击底部“创建”。


```
$model.ref('helloFlow').run().then(() => {
  // 根据Name获取组件
  var _table = context.$component.get('table_0');
  _table.doQuery();
}).catch(e => {
  // 失败消息
  context.$message.error('新增词条失败:' + e.resMsg, 5);
});
```

图 2-205 添加自定义动作代码



说明

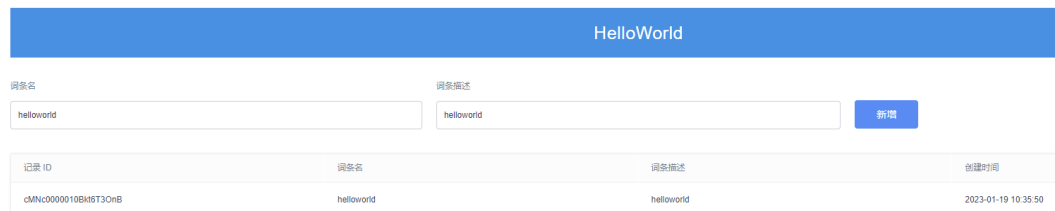
按钮事件主要是通过页面服务对象（helloFlow），调用服务编排，完成新增数据功能。

4. 单击页面上方的，保存事件。

步骤8 单击页面上方的，进入预览页面。


步骤9 在预览页面的输入框中，词条名和词条描述输入“helloworld”，单击“新增”，查看表格中新增的词条记录。

图 2-206 查看新增词条



----结束

步骤 5：开发应用-预览验证

在页面开发时，单击，可进入开发环境的预览页面，跟实际效果稍有差距。本节将介绍如何在应用的预览环境中，预览应用效果。这里预览效果跟实际在运行环境中基本一致。在应用配置下，还可以更改应用框架的样式和颜色。

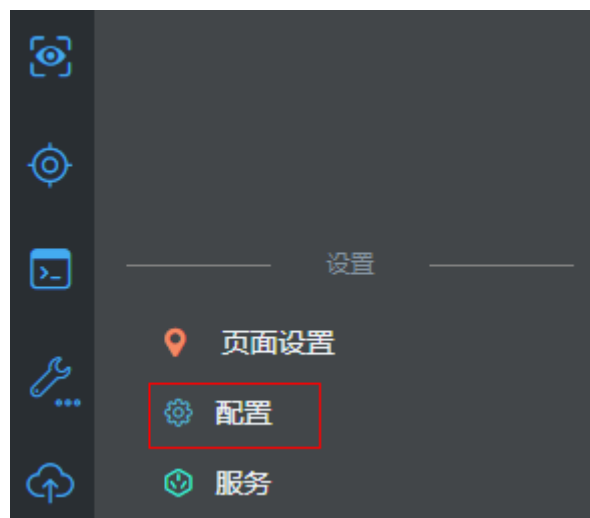
步骤1 进入“HelloWorld”应用经典设计器。

步骤2 将页面挂载到导航条上，作为应用菜单。

如果没有将页面挂载到导航条上，预览应用时，是无法找到应用的相关页面和对象等信息的。

1. 在应用左侧导航下方，单击“配置”。

图 2-207 配置



2. 在“导航条”页签，单击“菜单树”右侧的“+”，选择“添加页签”。

图 2-208 导航条添加页签



说明

在导航条上的页面较多时，您可以添加目录。添加目录可以更方便区分页面类型，方便导航使用。目录添加之后，会在应用导航中直接显示，目录下可以添加页签。另外，应用配置中还可以设置应用选项列表，系统参数等信息，您可以在[设置应用](#)中查看。

步骤3 在“添加页签”页面中，设置[表2-16](#)信息，单击“保存”。

表 2-16 页签信息

参数名	参数说明	本示例需要填写的值
页面类型	<p>当前要添加的页面类型。</p> <ul style="list-style-type: none"> • 主页页签：用于展示个人任务相关的信息。 • 对象页签：关联一个对象的布局，将展示指定对象的某个布局页面。 • 标准页面页签：用于关联并展示一个标准页面。 • 高级页面页签：用于关联并展示一个高级页面。 • BPM页签：用于关联并展示一个BPM。 • Web页签：用于关联一个URL，将展示URL对应的网页或页面。 • 状态机页签：用于关联并展示一个状态机。 • 报表页签：用于关联并展示一个报表。 	标准页面页签
显示区域	<p>导航页签的显示区域。</p> <ul style="list-style-type: none"> • 主页菜单：按照布局样式进行显示，例如显示顶栏区域或者右上角。 • 自定义菜单栏：选择该方式后，在应用的运行态Portal中，单击右上角用户名，会在下拉选项中显示导航页签。 	主页菜单
打开方式	<p>导航页签的打开方法。</p> <ul style="list-style-type: none"> • 当前窗口：选择该导航页签后，在当前窗口打开页面。 • 新窗口：选择该导航页签后，在新窗口打开页面。 	当前窗口
标签	显示在应用导航的名称。	HelloWorld
名称	用于唯一标识页签的名称。	helloPage
图标	导航页签的展示图标。	这里可以不设置，如果有，可以选择一个图标直接上传。
页面	需要添加的页面	helloPage (CNAME_helloPage) 。
描述	当前页面的描述信息，用于了解当前页面的主要内容。	不设置。

图 2-209 我的应用

添加页签 ✕

* 页签类型

显示区域

主菜单 

自定义菜单栏 

打开方式

* 标签

* 名称

图标

* 页面

描述

步骤4 在“导航条”页面，删除菜单树下的Home页签。


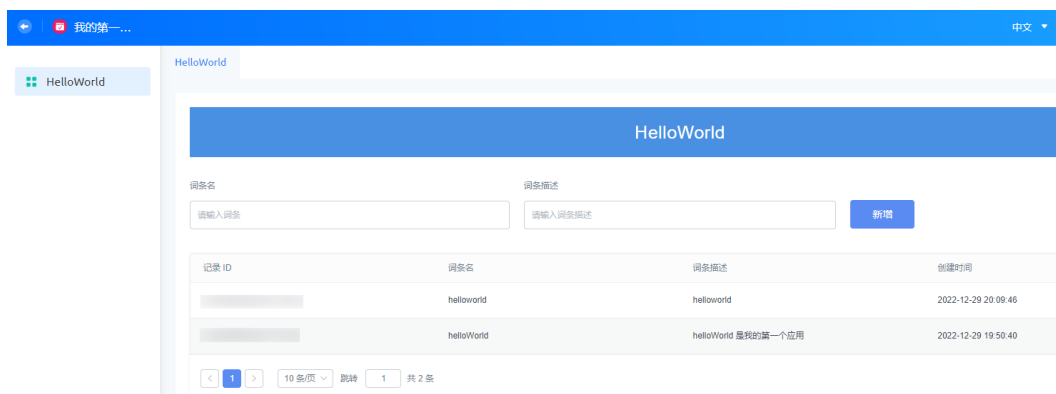
步骤5 在应用左侧导航下，单击，进入应用预览页面。

图 2-210 查看应用



步骤6 单击右侧“HelloWorld”，在右侧页面中，输入词条名和词条描述，单击“新增”，查看表格中新增的词条记录。

图 2-211 新增词条



步骤7 （可选）如果词条未能创建成功，按键盘“F12”，调测页面调试窗，重新输入词条，进行页面调试。

----结束

步骤 6：编译发布应用

在应用开发完成后，应用需要编译、打包和发布，既可以发布到当前租户的运行环境，也可以共享给其他租户，在其他租户的开发环境或运行环境下安装。

AstroZero支持多种发布方式，本示例中使用的是将应用发布到“我的仓库”进行发布，后续将会通过“我的仓库”再部署到运行环境。

• 应用包编译类型

在AstroZero经典版应用设计器中，支持编译**资产包**和**源码包**两种类型的应用包。如何在AstroZero新版设计器中，编译发布应用包，请参见[如何一键部署应用](#)。

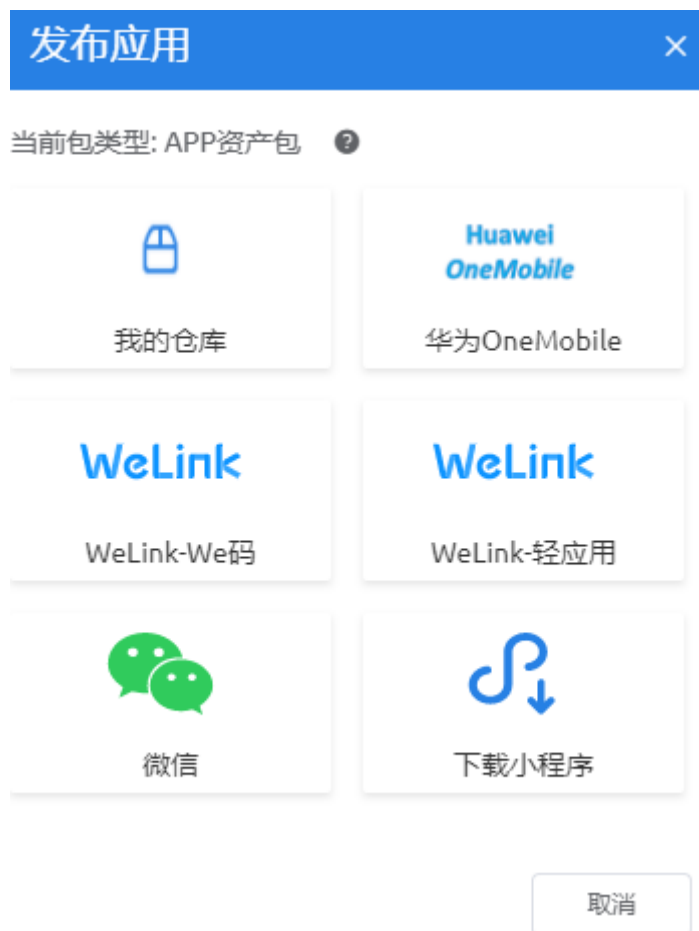
- 源码包：包中的所有组件都不受保护和限制。在其他环境安装后可编辑包中组件，即在原有基础上可进行再开发。如果后续其他用户在开发环境安装后，会显示在开发环境首页的“项目”页签下。

📖 说明

编译类型为“源码包”时，只有发布到“我的仓库”一种发布方式。

- 资产包：支持设置包中的组件是否受保护。打包时不做编译设置，默认打出的包都是资产包，包中组件都为只读保护模式，将包安装到其他环境时，只能运行和预览，不可编辑包中组件。如果后续其他用户在其他开发环境安装资产包后，应用会显示在开发环境首页的“库”页签下。
- 发布方式
AstroZero应用（资产包）发布方式，如图2-212所示。本示例中使用的是将应用发布到“我的仓库”，后续将会通过“我的仓库”再部署到运行环境。

图 2-212 应用发布方式




- 步骤1** 登录AstroZero经典版开发环境，在首页，单击“我的应用”下的“HelloWorld”，进入应用。
- 步骤2** 在应用开发页面左侧，单击，选择“设置”。
- 步骤3** 在包类型中，选择“资产包”，单击“保存”。


图 2-213 设置包类型



步骤4 在应用开发页面左侧，单击，选择“编译”，进行编译。

 说明

编译前，还可以单击进行一些自定义的编译设置，详细设置请参考[编译设置](#)。

步骤5 编译完成后，单击左下角，选择“我的仓库”，将应用程序安装包发布到我的仓库。

步骤6 填写版本信息，单击“发布”。

图 2-214 填写版本信息



发布成功后，页面显示“程序包已经被成功上传到我的仓库。”。

📖 说明

【思考】应用发布之后的包可以下载吗？其他租户可以看到吗？

在应用的“包管理”页面，可以查看已发布的应用包，也可以单击下载，如[图2-215](#)。不同租户之间，“我的仓库”里面的应用是不可见的。

图 2-215 查看发布到“我的仓库”下的应用包



----结束

步骤 7：部署运行应用

📖 说明

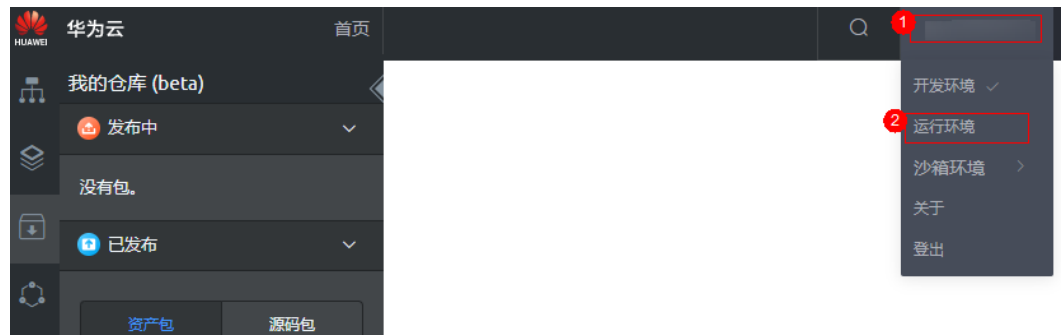
免费版不提供运行环境，请参考[购买商用租户](#)，直接购买专业版或专享版后，再进行打包发布操作，并在运行环境安装该应用。

在开发环境中，将应用发布到“我的仓库”后，AstroZero将会把该应用包同步发布到当前租户的运行环境中，以便于在运行环境中部署运行。

步骤1 使用当前租户开发者账号，访问并登录AstroZero运行环境。

您也可以直接在开发环境中，单击页面右上角的用户名，切换到运行环境，如[图2-216](#)。

图 2-216 切换到运行环境



步骤2 在运行环境的管理看板中，单击“我的仓库”，进入我的仓库。

图 2-217 我的仓库



步骤3 在“我的仓库”中，找到之前发布的“我的第一个应用”，单击“安装”，并在弹窗中单击“确定”。

步骤4 设置应用。


1. 在左侧导航栏中，选择“应用管理 > 应用导航”。
2. 在应用列表中，单击“我的第一个应用”后的 ，进入应用布局页面。

图 2-218 设置页面布局



3. 在“布局”下，选中第二种布局方式后，单击“保存”，再单击“预览”，进入应用。

图 2-219 设置布局类型并预览应用



步骤5 验证应用。

在应用页面中，词条名输入“helloWorld”，词条描述输入“helloWorld是我的第一个应用”，单击“新增”，查看表格中新增的词条记录。

图 2-220 新增词条



如果验证成功，恭喜您！您已经完成了在AstroZero上第一个应用的开发！您还可以查看快速入门中的其他实例，继续深入探索AstroZero的应用开发之旅！

----结束

（后续操作，可选）创建业务用户并配置权限

业务用户为应用的使用者，在AstroZero中开发的应用，是支持业务用户登录使用的。

在AstroZero运行环境部署应用后，可以给该应用创建业务用户，并配置业务用户使用该应用的权限。

步骤1 进入运行环境业务配置中心。




1. 使用华为账号，登录[华为云网站](#)，在顶部导航栏右侧单击“控制台”，进入华为云控制台。
2. 在左侧导航栏上方，单击，选择服务实例所在的区域项目。
3. 单击，在查找框中搜索“Astro轻应用”，单击查找到的结果，进入AstroZero服务控制台。
4. 单击“进入首页”，进入应用开发页面。
5. 在“工作台”页签中，单击页面左上角的，选择“业务配置中心”，进入运行环境业务配置中心。

图 2-221 进入运行环境业务配置中心



步骤2 （可选）新建角色。

说明

购买AstroZero标准版或专业版实例时，才会显示“角色权限”菜单。

1. 在“业务配置中心”页面的主菜单中，选择“组织用户”。
2. 在左侧导航栏中，选择“角色权限”。
3. 单击“所有角色”后的“+”，设置添加的角色名称，例如“helloWorld”，并配置应用为可见和默认应用。

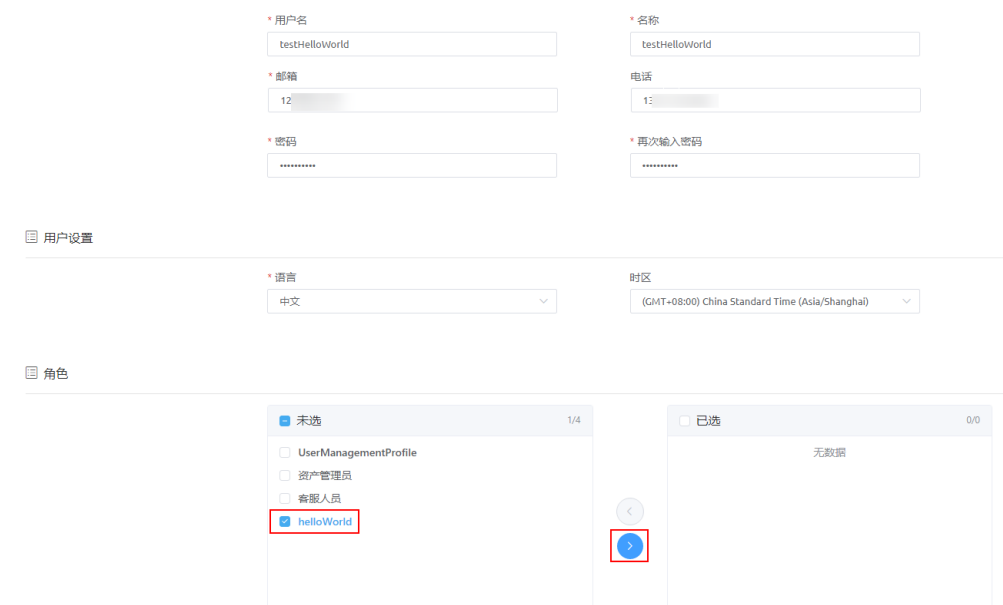
图 2-222 添加角色



步骤3 创建业务用户，例如“testHelloWorld”。

1. 在“业务配置中心”页面，选择“组织用户”。
2. 在左侧导航栏中，选择“用户管理”，单击“创建用户”。
3. 在“新建用户”页面，填写用户信息，其中用户名为“testHelloWorld”。
4. 在“角色”区域中，从左侧勾选步骤2.3创建的角色“helloWorld”再单击“>”赋予该用户相应的权限。

图 2-223 配置业务用户



5. 设置完成后，单击“保存”。

步骤4 在运行环境的环境配置中，配置业务用户权限。


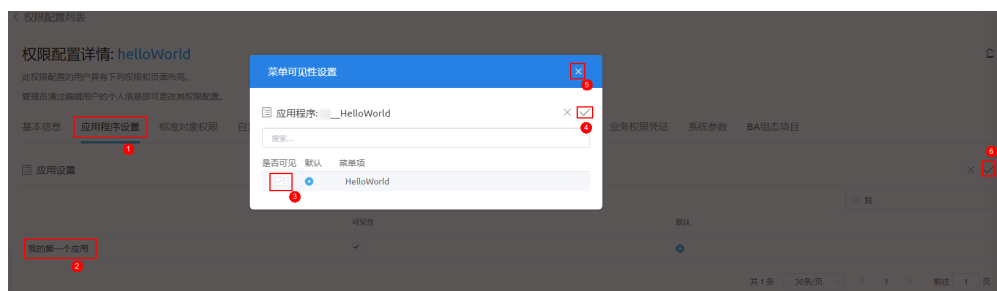
1. 在“工作台”页签中，单击页面左上角的，选择“环境配置”，进入运行环境的环境配置。


图 2-224 登录运行环境管理中心



2. 在左侧导航栏中，选择“用户安全 > 权限”。
3. 单击业务用户使用的权限“helloWorld”，进入权限详情页。
4. 在“应用程序设置”页签，设置应用和菜单可见性。

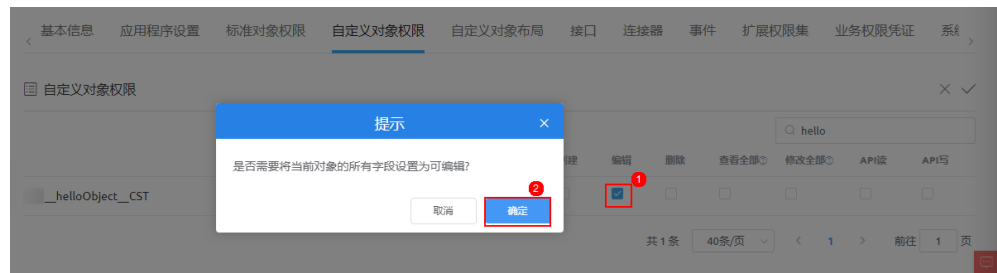
图 2-225 设置应用和菜单可见性



5. 在“自定义对象权限”页签，单击“自定义对象权限”右侧的 ，并在搜索框中输入“**CNAME_**helloObject_CST”，然后勾选“编辑”权限，并在提示弹窗中单击“确定”。

其中，“CNAME_”为命令空间，请以环境创建的实际命名空间为准。

图 2-226 赋予对象编辑权限



6. 依次勾选对象的“修改全部”、“创建”、“API读”和“API写”权限，再单击 ✓，保存设置。

图 2-227 勾选对象权限




7. 在“接口”页签，单击右侧的 ，勾选“CNAME_helloFlow”的“可执行”权限。

图 2-228 增加可执行权限



步骤5 业务用户登录应用，并使用应用。

1. 开发者用户获取运行环境的子域名，并将子域名分享给业务用户。
更多关于子域名的介绍，请参见[购买商用租户](#)。

图 2-229 获取子域名



2. 业务用户在浏览器中，输入已获取的运行环境子域名，输入用户名和密码，即可登录应用Portal运行页面。

图 2-230 登录应用



3. 业务用户首次登录需要重置密码，重置后使用新密码再次登录应用Portal运行页面。
4. 登录后查找应用，输入词条名和词条描述，新增词条，体验应用。

----结束

3 进阶实战

3.1 设备维修管理应用开发

3.1.1 注册登录指导及环境说明

3.1.1.1 注册账号及开通 AstroZero 服务

注册华为账号并实名认证

如果您已有一个华为账号，请跳到下一个任务。如果您还没有华为账号，请参考以下步骤创建。

- 步骤1** 进入[华为云](#)官网，单击页面右上角的“注册”。
- 步骤2** 参考[注册华为账号并开通华为云中](#)操作，完成注册。
- 步骤3** 注册后参考[个人账号如何完成实名认证](#)或[企业账号如何完成实名认证](#)中操作，完成个人或企业账号实名认证。

----结束

为账户充值

实名认证后，请为您的账户进行充值。AstroZero提供了免费版、标准版、专业版和专享版四种规格套餐，供您选择。为了更好的体验本示例，建议您购买专业版或专享版，否则[打包发布](#)应用到运行环境功能，将不可执行（免费版不提供运行环境）。

- AstroZero规格差异说明，请参见[产品规格差异](#)。
- AstroZero价格的详细介绍，请参见[Astro轻应用 计费说明](#)。
- 如何进行账户充值，请参见[账户充值](#)。

购买 AstroZero 实例

在使用AstroZero前，您需要创建一个AstroZero实例。AstroZero实例是一个独立的资源空间，所有的操作都是在实例内进行，不同实例间的资源相互隔离。购买AstroZero实例的详情操作，请参见[购买商用租户](#)。

📖 说明

AstroZero提供了新版和经典版两套开发环境，本入门以经典版开发环境为例进行介绍。新版开发环境的介绍，请参见[初识新版设计器](#)。

3.1.1.2 进入经典版开发环境

什么是开发环境

开发环境是开发者专门用于开发应用的环境，用户可根据需求自行开发配置各种逻辑模块化元素，从而创建各类应用。关于AstroZero中环境的更多介绍，请参见[基本概念](#)。

进入经典版开发环境

步骤1 使用华为账号，登录[华为云网站](#)，在顶部导航栏右侧单击“控制台”，进入华为云控制台。

图 3-1 进入华为云控制台



步骤2 在左侧导航栏上方，单击📍，选择服务所在的区域。

步骤3 单击☰，在查找框中搜索“Astro轻应用”，单击查找到的结果，进入AstroZero服务控制台。

图 3-2 AstroZero 服务控制台



步骤4 在主页中，单击“旧版入口”，即可进入经典版开发环境。

图 3-3 经典版开发环境



----结束

3.1.2 业务场景和流程设计介绍

AstroZero是以应用为中心的开发，本文以IoT领域电梯设备运维管理和维修的应用场景为主线，由浅入深的向您介绍如何在AstroZero中开发该应用。

业务场景

设备维修管理系统应用中，包含以下两类用户：

- 系统管理员用户：管理应用的用户，用于新增业务用户，添加用户权限、添加设备信息、管理工单和监控设备。

说明

本示例以登录AstroZero开发应用的账号，作为管理员账号。

- 业务用户：使用“设备维修管理系统”应用的用户，分别是客服人员、派单员及维修人员。

设备运维管理和维修场景的业务流程：

1. 系统管理员进行电梯信息的管理和维护，如增加电梯基本信息，修改电梯基本信息等，对业务用户的管理，如新增系统业务用户，并为其分配权限等。
2. 电梯客服人员受理用户投诉，并创建维修单。
3. 派单人员收到客服人员的维修单后，派发给维修工程师。
4. 维修工程师进行现场修理，并在处理完成后关闭维修单。
5. 系统管理员实时监控待维修设备的实时位置。

设备维修管理系统的各个角色涉及的具体业务如下：

图 3-4 业务应用管理功能

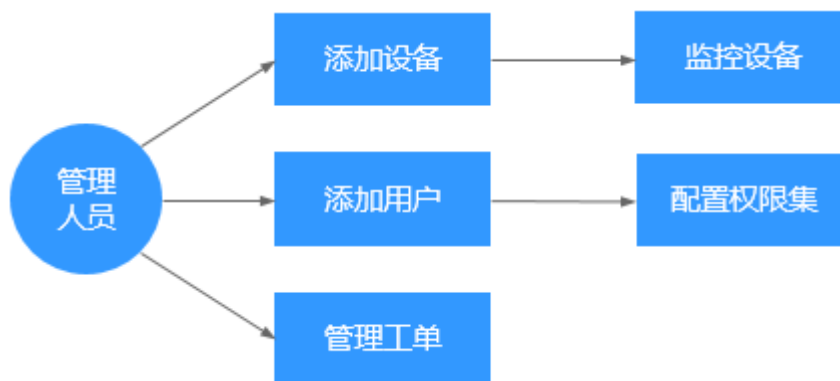
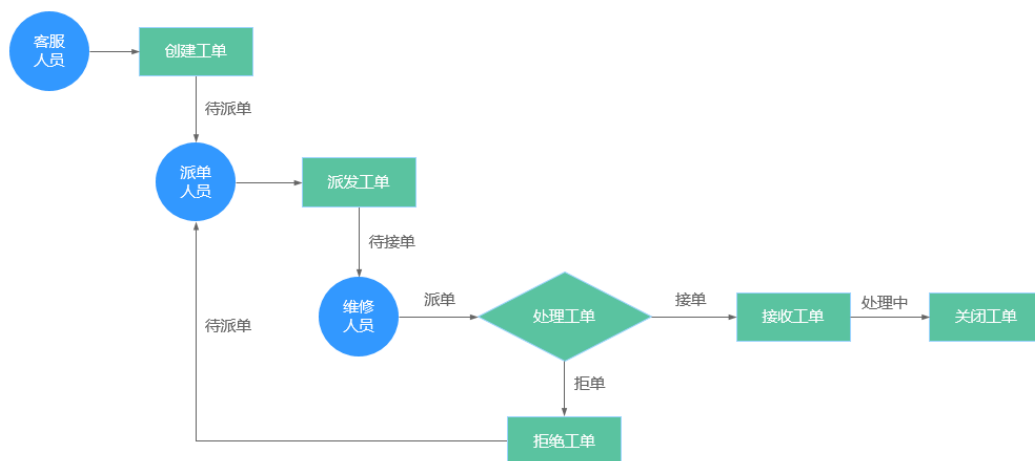


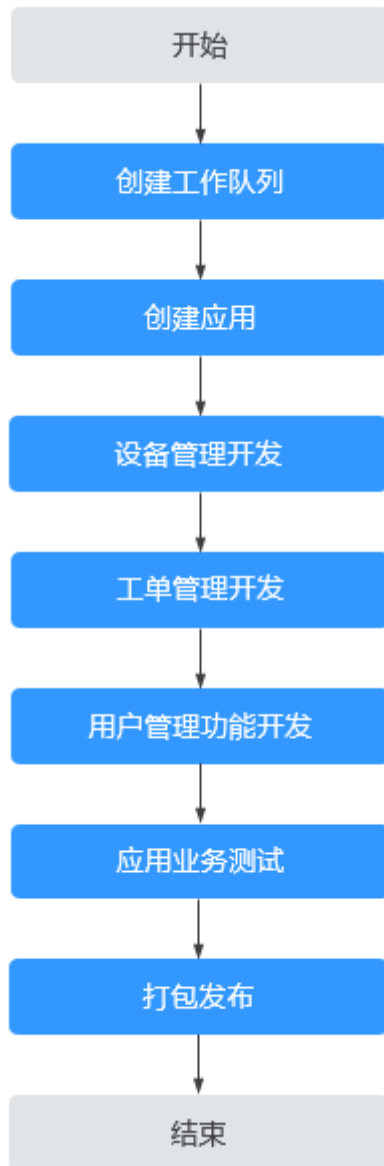
图 3-5 工单流程



开发流程

基于业务场景的应用开发流程，如图3-6所示。

图 3-6 应用开发流程



沿着上述开发线，您将在后续各章节了解到AstroZero的如下能力。

表 3-1 开发任务和平台能力的映射关系

序号	开发任务	预计开发时长	对应的平台能力	学习目标
1	创建工作队列	20分钟	创建工作队列	了解工作队列的含义。

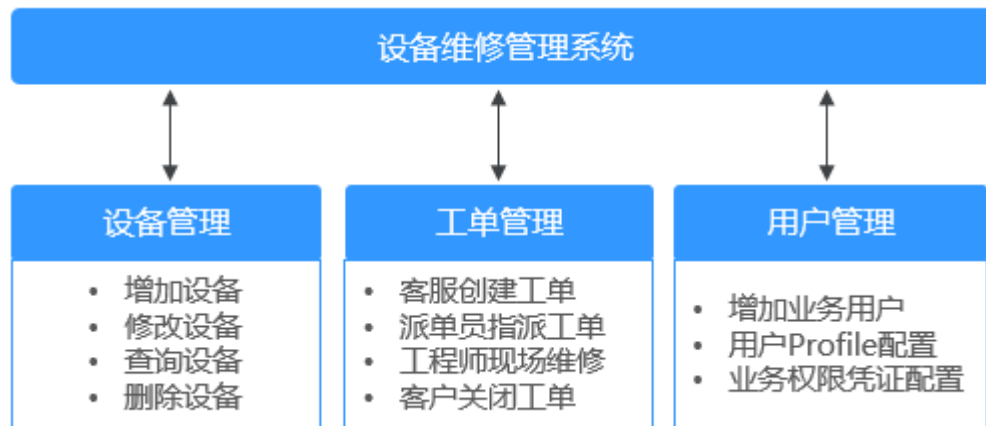
序号	开发任务	预计开发时长	对应的平台能力	学习目标
2	创建应用	1小时	<ul style="list-style-type: none"> 创建应用 创建高级页面 	<ul style="list-style-type: none"> 掌握创建应用程序的方法。 了解应用程序的目录结构。 了解高级页面的相关背景知识和组装方法。 <ul style="list-style-type: none"> - 上传widget - 使用widget拼装高级页面
3	设备管理开发	4小时	<ul style="list-style-type: none"> 创建对象 开发脚本 开发服务编排 组装标准页面 组装高级页面 	<ul style="list-style-type: none"> 了解对象相关背景知识，掌握自定义对象的方法。 了解标准页面相关背景知识，掌握使用“表格”组件组装页面、增删改查对象的方法。 了解服务编排相关背景知识，掌握开发、测试服务编排的方法。 标准页面UI组件：“表单”、“容器”、“输入框”、“下拉框”、“级联选择框”、“标题”、“按钮”的主要配置属性，为“表格”组件添加操作列。 页面模型：模型的类型和作用。 页面事件：组件事件代码的编写。 使用widget组装高级页面。

序号	开发任务	预计开发时长	对应的平台能力	学习目标
4	工单管理开发	14小时	<ul style="list-style-type: none"> 开发脚本 组装复杂的标准页面 开发服务编排 开发BPM 定义应用的导航菜单 	<ul style="list-style-type: none"> 了解脚本相关背景知识，掌握开发、测试脚本的方法。 标准页面： <ul style="list-style-type: none"> UI组件：了解表单、容器、输入框、下拉框、级联选择框、标题和按钮组件的主要配置属性，为“表格”组件添加操作列。 UI组件：了解“可折叠容器”的主要配置属性，自定义“表格”组件中某字段的显示内容。 事件：动态加载页面内容、组件事件代码的编写。 模型：模型的类型和作用。 BPM：了解BPM、掌握BPM开发流程及图元配置。
5	用户管理功能开发	4小时	<ul style="list-style-type: none"> 租户和子账号管理 权限管理 配置权限 配置业务凭证 	<ul style="list-style-type: none"> 了解租户和业务用户。 了解平台的权限控制机制：平台操作权限、应用权限、标准对象权限、自定义对象权限平台。
6	应用业务测试	30分钟	<ul style="list-style-type: none"> 应用预览环境 	<ul style="list-style-type: none"> 了解应用预览环境。 配置应用布局。
7	打包发布	10分钟	<ul style="list-style-type: none"> 打包 发布 	<ul style="list-style-type: none"> 了解应用的编译发布。 了解在运行环境安装应用，了解如何在运行环境中为应用设置自定义登录页面。 在其他租户环境安装应用包。

3.1.3 功能模块分类说明

设备维修管理系统包括三个功能模块，各功能模块分类说明，如图3-7所示。

图 3-7 功能模块说明



本文将以功能模块为单位，逐个开发，带您一步步学习如何使用AstroZero开发一个完整的应用。

3.1.4 创建工作队列

工作队列是在业务场景中，用来记录一类具有相同权限和任务对象的成员集。创建工作队列是为了给业务用户在业务工单流转过程中，区分不同的权限。

本文中的工作队列用于工单流程中的**业务用户群**，包括客服人员、派单人员和维修人员三个工作队列。

背景信息

工作队列是在业务场景中，用来记录可以受理相同具体业务的用户群体。这种记录一直保留在队列中，直到用户接受它们并进行处理，或它们被转移到另一个队列。您可以指定每个队列支持的对象集合，以及允许从队列检索记录的用户组（用户、业务用户等）。

创建工作队列主要包含两方面内容：

1. 添加队列的成员。
队列成员可以是单个用户、公共组、单个角色或带有下级角色的角色，以及业务用户；若队列的成员是角色，此队列将包含角色中所有的用户。
2. 配置队列支持的对象。
配置了支持的对象后，涉及特定数据对象的触发器、待审批任务才能进入该队列。

创建工作队列并添加成员

步骤1 使用华为账号，参考[进入经典版开发环境](#)中操作，进入AstroZero经典版开发环境。

步骤2 在经典版应用开发环境中，单击“管理”，进入AstroZero经典版开发环境管理中心。

图 3-8 管理页面入口



步骤3 在左侧导航栏中，选择“用户管理 > 工作队列”，单击“新建”。

图 3-9 工作队列入口



步骤4 新建“客服人员”队列。

1. 在新建队列页面，设置队列“标签”为“客服人员”，“名称”为“CustomerService”。
2. 在“队列成员”中，单击“添加”。

3. 设置“成员类型”为“用户”，“成员列表”选择当前租户开发者，再单击“添加”，退出弹窗。

📖 说明

当前还未创建业务用户，因此这里只添加当前开发者账号到队列中，便于后续开发测试。

4. 在新建队列页面，单击“保存”。

图 3-10 添加用户

新建队列

📄 基本信息

* 标签	* 名称
<input type="text" value="客服人员"/>	<input type="text" value="CustomerService"/>
公共邮箱 <input type="text" value="请输入"/>	工作队列管理者 <input type="text" value="请选择"/>
<input type="checkbox"/> 给成员发送邮件	<input type="checkbox"/> 队列对象

📄 队列成员

要将成员添加到此队列中，请选择一种类型的成员，然后从“可用成员”中选择组，角色或用户，并将其移动到“所选成员”。如果队列中所有对象的共享模型为“公共读/写”，则不需要将用户分配给队列，因为所有用户都可以访问这些对象的记录。

<input type="button" value="添加"/>	<input type="button" value="删除已选成员"/>	<input type="text" value="搜索"/>	
<input type="checkbox"/>	名称	成员类型	操作
暂无数据			
共 0 条	10条/页	< 1 >	前往 1 页

图 3-11 添加“用户”成员

步骤5 参考以上步骤，创建表3-2中的其他工作队列。

表 3-2 工作队列详情

标签	名称	成员列表
客服人员（上一步已创建）	CustomerService	当前应用开发者、业务用户列表成员中的客服人员。
派单员	Dispatcher	当前应用开发者、业务用户列表成员中的派单人员。
维护人员	FME	当前应用开发者、业务用户列表成员中的维护人员。

----结束

后续操作

因为业务用户是在应用创建完成后才创建的，因此当前还不能将业务用户中的“客服人员”、“派单人员”以及“维护人员”，添加到工作队列中，在[应用业务测试](#)章节，新增业务用户之后，则需要将每个业务用户添加到工作队列中。

3.1.5 创建应用

3.1.5.1 创建“设备维修管理系统”应用

应用是一个可运行的、实现特定业务功能的业务单元。您需要先创建一个应用，再在应用内创建数据对象、后台业务逻辑以及前端页面。创建应用是在AstroZero开发项目的第一步，也是端到端构建软件应用的入口。

“设备维修管理系统”应用的主要功能包括电梯设备信息管理、业务用户管理，维修工单的创建/派发和处理调度，以及整个维护情况的监控分析。

创建应用

步骤1 使用华为账号，参考[进入经典版开发环境](#)中操作，登录AstroZero经典版开发环境。

📖 说明

此处的华为账号是指购买AstroZero服务的账号。更多AstroZero中用户的介绍，请参见[AstroZero中的用户](#)。

步骤2 在AstroZero经典版开发环境首页的“项目”页签下，单击“行业应用”。

图 3-12 创建行业应用



步骤3 在行业应用页面，单击“创建行业应用”。

图 3-13 创建空白行业应用



步骤4 定义命名空间。

为保证后续您的应用发布到应用市场时，不与其他开发者冲突，首次创建应用时，需要先定义本租户的命名空间。

后续您创建的对象、脚本、服务编排、页面等的名称中，系统都会为其增加命名空间前缀。例如，您将命名空间定义为“HW”，则您后续创建的所有对象、脚本、页面等都将以“HW_”开头。

注意

- 命名空间一旦设定不能修改，请慎重定义；建议您使用公司或者团队的缩写作为命名空间。
- 本实战中以命名空间“HW_”为例进行说明，实际操作时请替换为您所创建的命名空间。

步骤5 选择“应用”页签，输入应用程序的“标签”和“名称”，单击“创建”。
应用创建完成后，自动进入应用开发页面。

图 3-14 创建应用

创建行业应用

基本信息

* 标签 [?] 设备维修管理系统

* 名称 [?] MyApp

分类 请选择

描述 请输入

高级设置

* 运行时版本 [?] 1.3.9

添加图标

取消 创建

表 3-3 应用基本信息

标签	名称
设备维修管理系统	MyApp 说明 <ul style="list-style-type: none">应用创建后，将自动添加命名空间前缀，实际创建应用名为“HW_MyApp”。应用名称在AstroZero中是应用的唯一标识符，在应用开发过程中，例如接口调用、业务用户登录等场景，会涉及拼接应用名称。在应用开发页面，若提示“您的租户没有配置默认邮件服务器地址，将会导致在服务编排/BPM/脚本中发送邮件功能不可用”，请直接忽略，本示例不涉及。

应用创建后，将会自动出现在经典版开发环境首页“项目 > 我的应用”下。再次编辑应用时，可以在“我的应用”下，单击应用名，进入应用。

图 3-15 我的应用



----结束

创建应用目录


- 步骤1** 使用华为账号，参考[进入经典版开发环境](#)中操作，进入AstroZero经典版开发环境。
- 步骤2** 在AstroZero经典版开发环境首页“项目”页签，单击“我的应用”下的“设备维修管理系统”，进入应用。
- 步骤3** 单击“设备维修管理系统”后的，再单击“目录”，在弹窗中输入“Equipment”，单击“保存”，创建一个“Equipment”目录。

图 3-16 创建应用目录

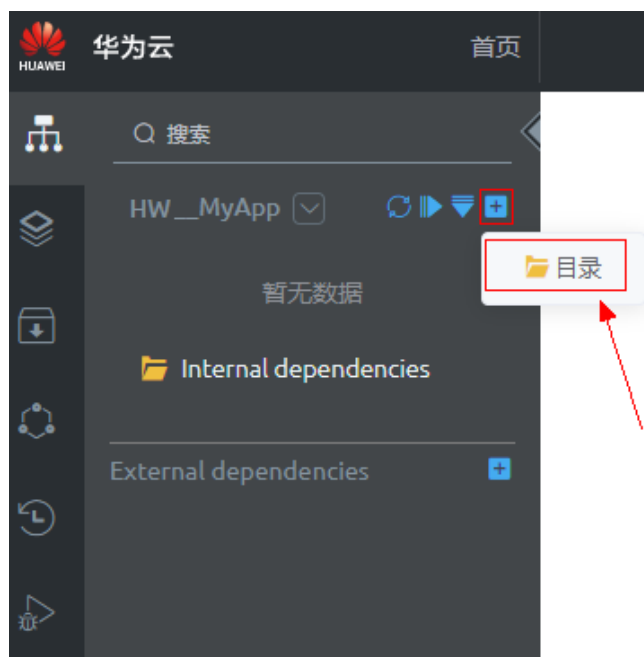
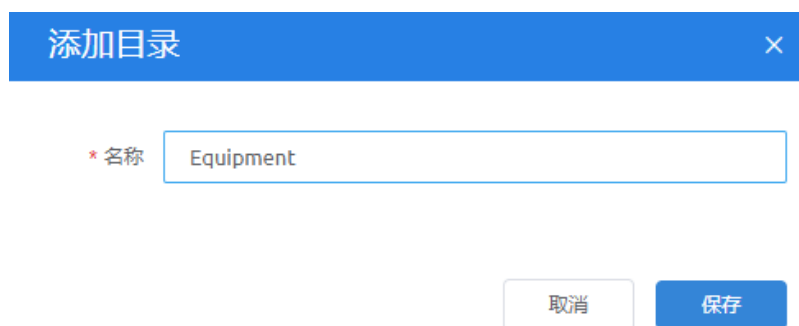


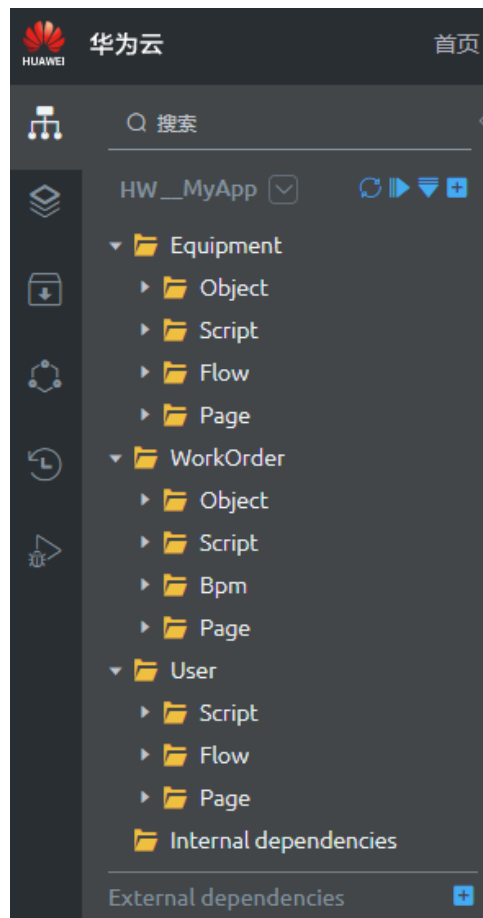
图 3-17 添加目录 Equipment



步骤4 参考上一步，按应用业务的功能模块，再创建WorkOrder和用户目录。

步骤5 在Equipment、WorkOrder和用户目录下，参考图3-18所示，分别创建Object、Script、Flow、Bpm和Page。

图 3-18 创建应用目录



----结束

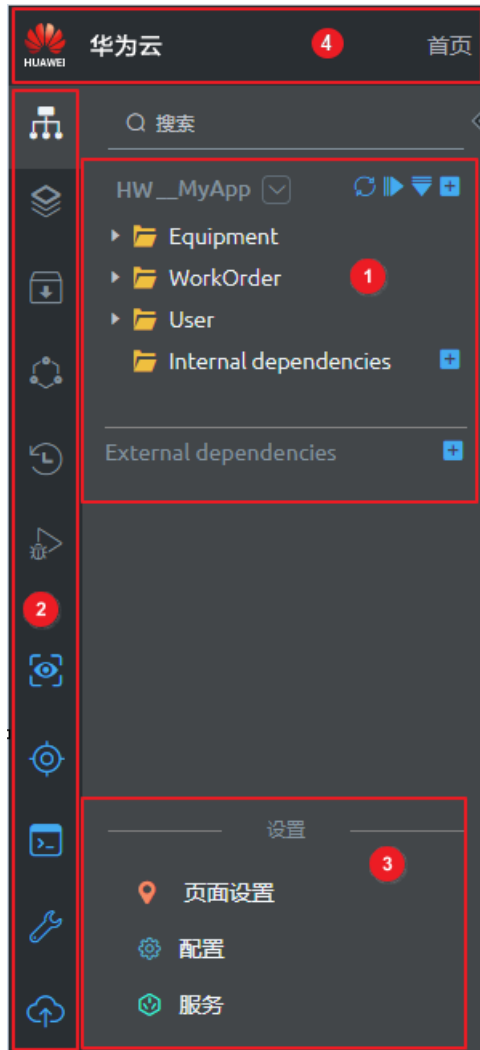
3.1.5.2 了解应用目录及菜单

一个业务系统通常包括前端页面、后台逻辑和数据库表。因此，AstroZero采用类似的结构管理应用程序。

如图3-19所示，应用管理页面包含三部分（红框1、红框2和红框3）：

- 红框1中是当前应用的页面、模型和逻辑。在后续章节中，开发的页面、对象及后台逻辑，推荐按以下规划的方式，将应用资源分别放在对应的目录下：
 - Object: 数据对象
 - Script: 脚本
 - Flow: 服务编排
 - Bpm: BPM
 - Page: 前端页面
- 红框2中是应用管理功能，例如预览应用、编译打包、发布应用等。
- 红框3中是应用设置功能，例如设置应用导航、页面设置等。
- 红框4中是AstroZero应用开发环境的导航菜单，单击“首页”可返回开发环境首页。

图 3-19 应用的管理目录



3.1.5.3 创建“业务用户登录”页面

3.1.5.3.1 背景与原理

业务应用构建登录页面时，一般情况下是通过使用AstroZero的高级页面能力实现。您可以通过本节认识高级页面，并了解登录页面的开发流程。

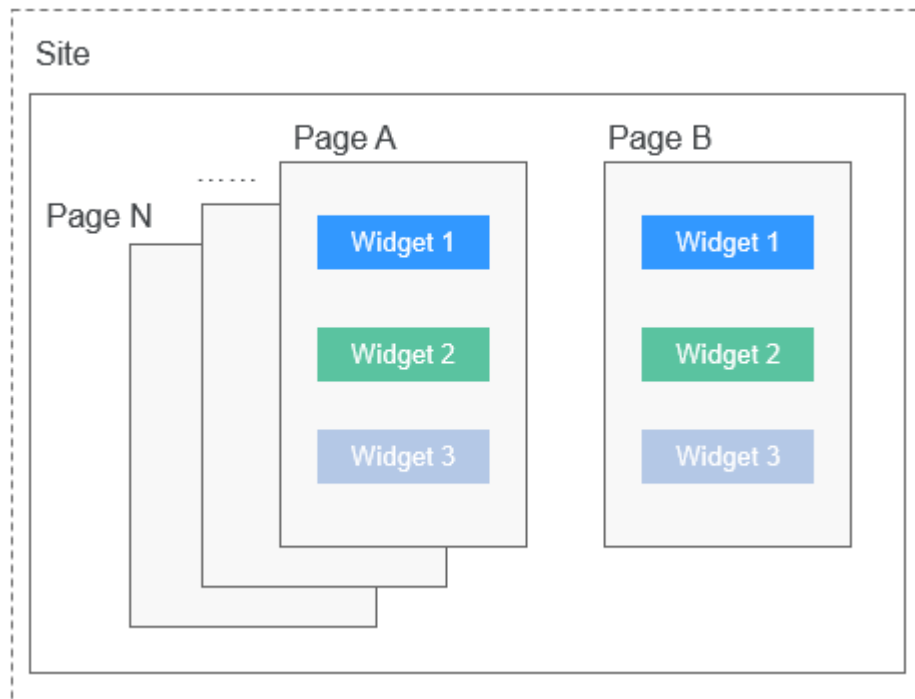
了解高级页面

AstroZero前端页面有标准页面、高级页面和业务大屏三种。本节主要带您了解、学习并使用高级页面。

- 标准页面：对于一般的业务应用系统，其功能主要是针对业务数据的增、删、改、查，前端界面的样式相对简单的页面场景，此时，推荐您使用平台提供的“标准页面”。您可以通过拖、拉、拽页面组件，再加上少量事件代码，即可拼装出所需页面，具体介绍请参见[标准页面](#)。
- 高级页面：对于一些样式比较复杂的页面，例如网站、电商、园区大屏等，您可以使用平台提供的“高级页面”。

高级页面是由一个或者多个Widget（即组件）拼装而成。如图3-20所示，组件是可复用的页面组成元素，一个页面由一个或多个组件拼装而成。如果将一个页面看成拼图游戏的完整图案，那么组件就相当于拼图的每一小块。

图 3-20 页面与 widget（组件）的关系



Widget的运行依赖Library（库），若缺少相应的Library（库），则Widget不能正常运行。因此在加载widget前，需要先加载必要的Library。

在操作前端页面时，经常会需要调用后台数据，例如即将开发的登录页面，需要获取业务用户信息。这时需要通过桥接器调用后台的服务编排、Script等获取后台数据。因此，在引用widget时，经常需要配置桥接器。

AstroZero高级页面中的组件分为系统预置组件和自定义组件：

- 系统预置组件，可以直接使用。
- 登录组件属于自定义组件。本示例中已经为您提供好了开发好的组件包，您只需要上传到站点中即可使用。自定义登录组件的开发方法，请参考[线下开发自定义登录组件及用户指南（低代码）](#)。

“业务用户登录”页面开发流程

本示例中，主要通过线下开发一个自定义登录组件，然后上传到高级页面，再进行组件配置，实现应用登录页面功能。登录页面开发流程，如图3-21所示。

图 3-21 应用登录页面开发流程



📖 说明

开发自定义组件：此步骤可以跳过，本示例中提供已经开发好的自定义登录组件；如果想要了解自定义登录组件的开发方法，请参考[线下开发自定义登录组件](#)。

业务用户登录页面即是设备维修管理系统的登录界面，大致设想如[图3-22](#)所示。

图 3-22 应用登录页面



3.1.5.3.2 上传自定义登录组件

自定义登录组件开发完成，并上传到AstroZero后，才能通过高级页面调用。

您可以直接使用文中提供的已开发完成的登录组件，上传到AstroZero中。也可以自行在AstroZero上，下载自定义组件模板，进行线下开发，然后再上传到AstroZero中。

操作步骤


- 步骤1** 单击[userLogin.zip](#)，下载自定义登录组件包到本地。
- 步骤2** 使用华为账号，参考[进入经典版开发环境](#)中操作，进入AstroZero经典版开发环境。
- 步骤3** 在AstroZero经典版开发环境首页“项目”页签的“我的应用”中，单击“设备维修管理系统”，进入应用。
- 步骤4** 在左侧列表中，单击，选择“高级页面 > 组件”，单击“提交新组件”。
- 步骤5** 单击“请选择源文件(.zip)”，选择之前下载的组件包“userLogin.zip”，设置如下参数后，单击“提交”。

图 3-23 上传 userLogin 组件



* 名字

userLogin

* 上传源文件(.zip)

请选择源文件(.zip)

t0000000000bcbtj1jzbo_userLogin.zip

文件中检测到语法警告, 点击此处 下载并查看警告信息。

组件ID

t0000000000rznuygfk_ userLogin

分类

请选择

领域


通用

场景

高级页面

* 发行说明

自定义登录组件

- 名字：设置新上传组件的名称，本示例设置为userLogin。
- 场景：设置组件的使用场景，后续在创建高级页面中会使用该自定义组件，故此处保持默认即可。
- 发行说明：设置新上传组件的发行说明，通常设置为组件的功能说明，本示例设置为“自定义登录组件”。单击发行说明后的, 可对发行说明进行国际化设置。

----结束

3.1.5.3.3 组装“业务用户登录”页面

“业务用户登录”页面是一个高级页面，主要是通过引用[上传自定义登录组件](#)中上传的自定义登录组件，再配置相关参数，来实现登录功能。

操作步骤

步骤1 创建高级页面。


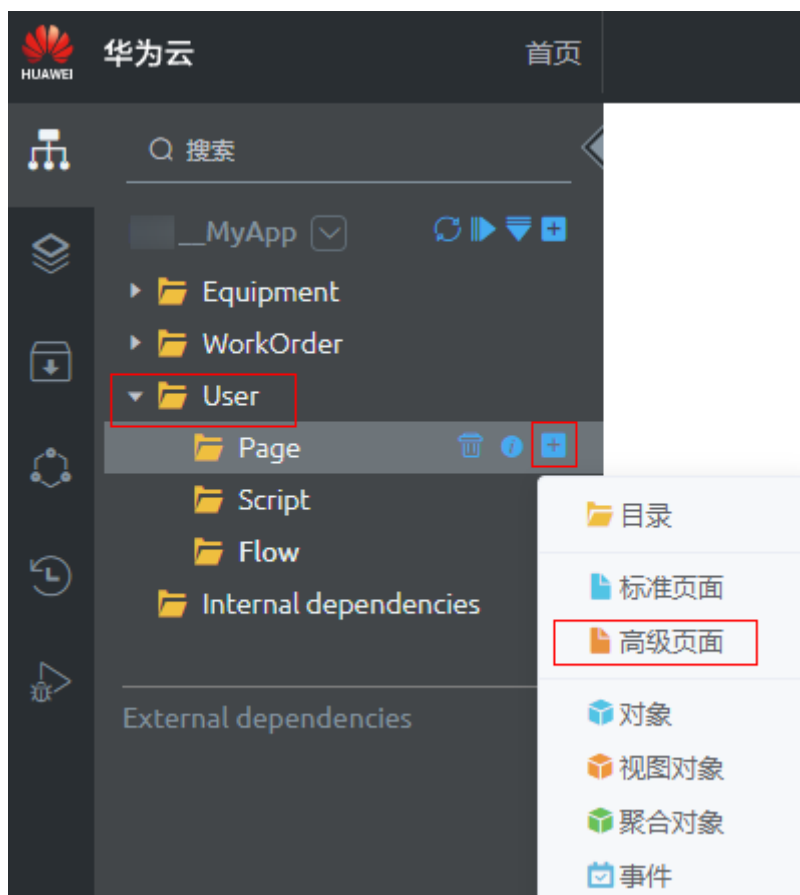
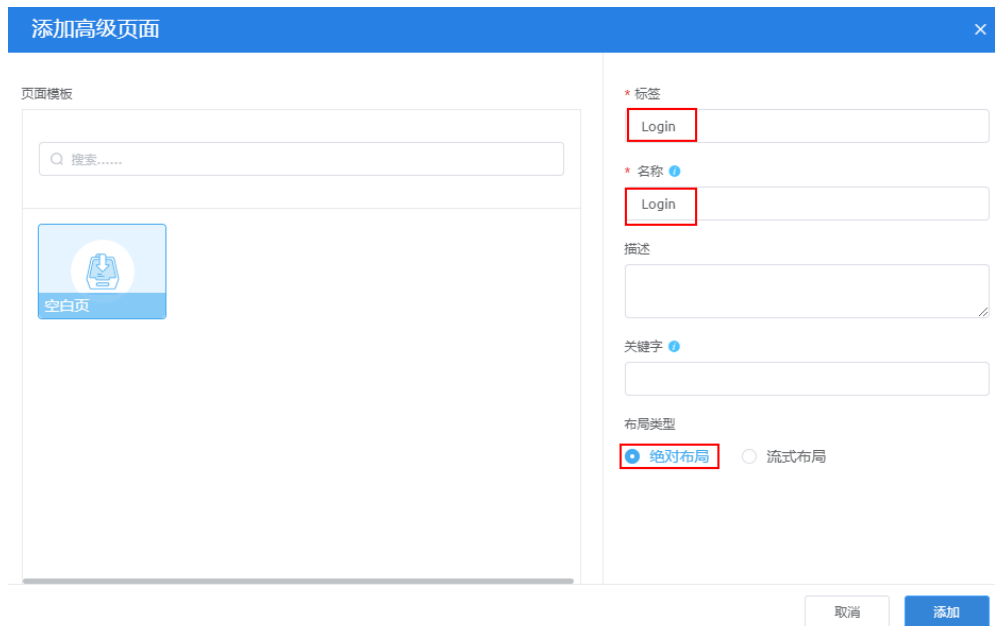
1. 进入AstroZero经典版开发环境首页，在“项目 > 我的应用”中，单击“设备维修管理系统”，进入应用。
2. 单击，进入工作目录。
3. 在“User”目录下，将鼠标放在“Page”上，单击界面上出现的“+”，在弹出菜单中选择“高级页面”。

图 3-24 创建高级页面



4. 设置“标签”和“名称”为“Login”，并选择“绝对布局”，单击“添加”。


图 3-25 添加高级页面



说明

高级页面布局有绝对布局和流式布局两种，页面布局详细介绍请参见[高级页面布局](#)。

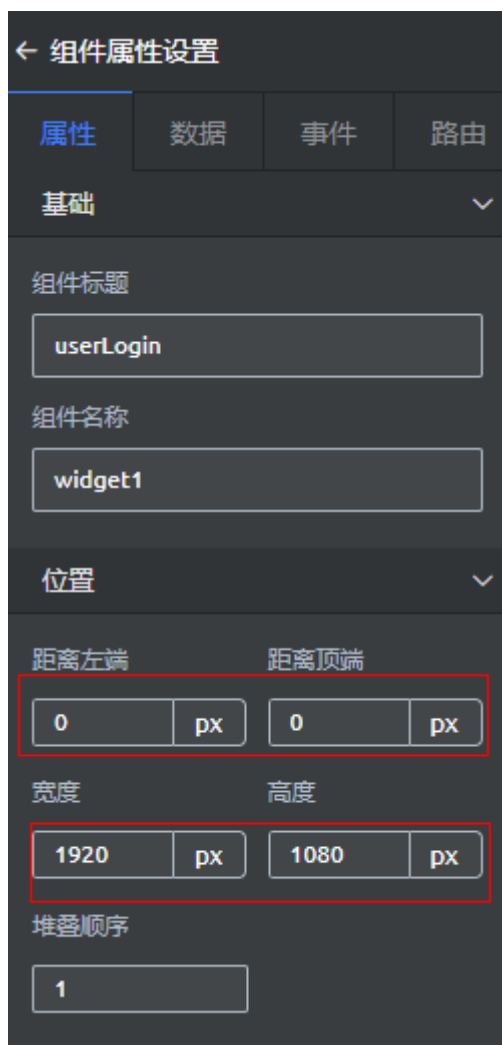
步骤2 拖拽自定义组件“userLogin”到页面。

单击左上角，打开组件列表，单击“全部”。在“自定义”页签，搜索到“userLogin”组件，将其拖进页面编辑区。“userLogin”组件即是在[上传自定义登录组件](#)中上传的自定义登录组件。

步骤3 设置自定义组件“userLogin”的位置属性。

1. 单击页面右下角的空白处，右侧显示当前视图组件列表。
2. 单击选中“userLogin”组件，会在右侧显示该组件的属性配置面板。
3. 在“位置”中，设置“距离左端”、“距离顶端”为“0”，“宽度”为“1920”，“高度”为“1080”。

图 3-26 设置组件位置





4. 单击页面上方 ，保存页面修改。
5. 在目录中，将鼠标放在“Login”上，单击 ，然后选择“设置”，在弹窗中设置页面的拉伸属性，再单击“保存”。

图 3-27 页面设置

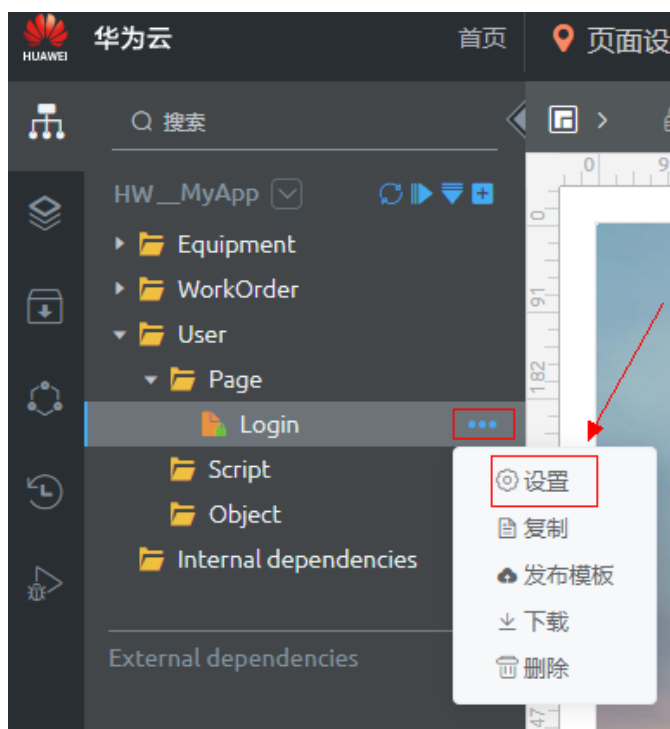
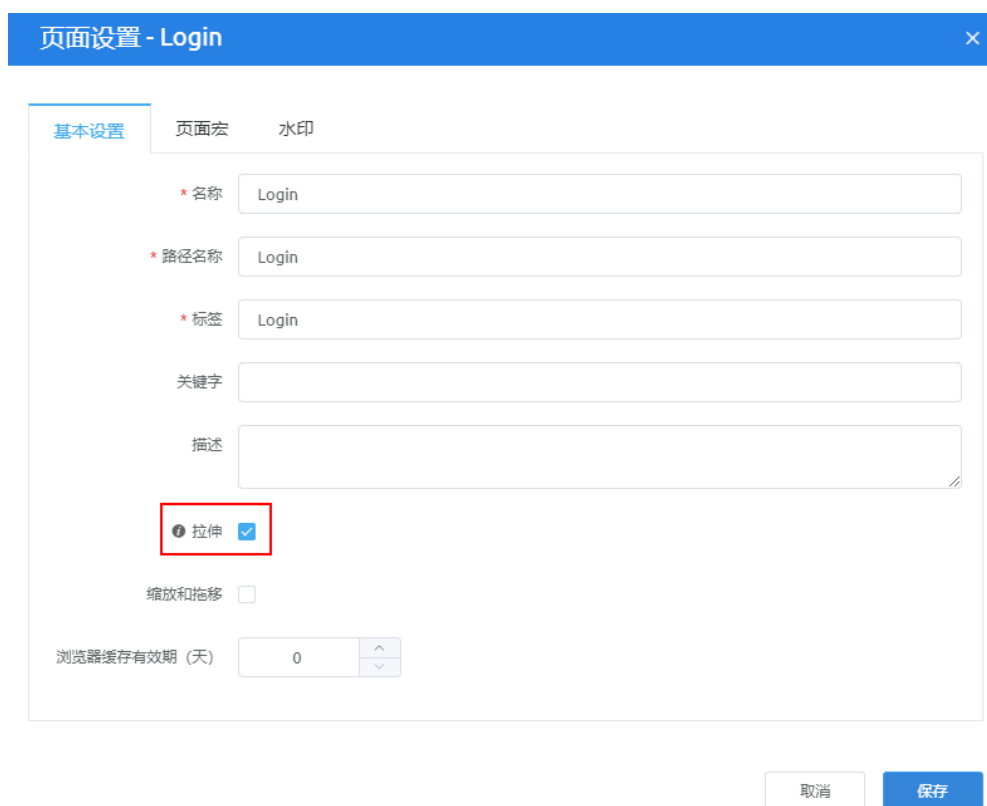


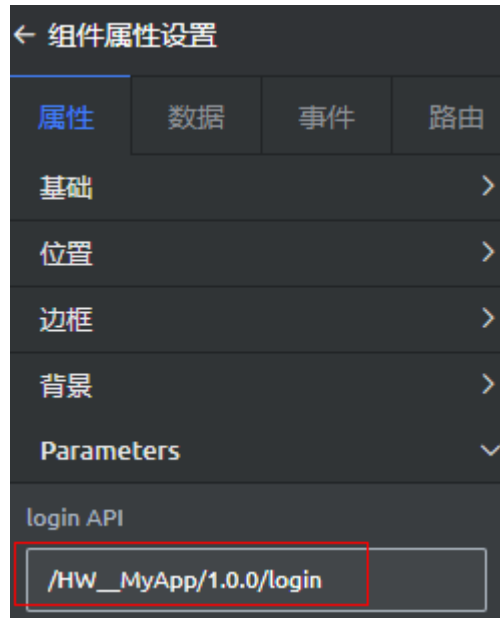
图 3-28 设置页面拉伸



步骤4 设置自定义组件“userLogin”的“数据”中的桥接器。

1. 单击选中“userLogin”组件，在右侧“组件属性设置 > 属性”页签最下端的“Parameters”配置项下，设置“login API”为“/HW_MyApp/1.0.0/login”。

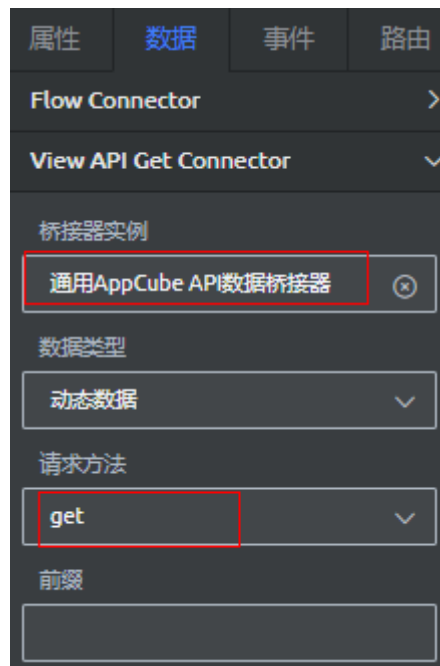
图 3-29 添加命名空间前缀



“login API”为登录接口的URL后半段，请根据实际情况修改，调用服务编排时，会在服务编排名称前面自动拼接租户命名空间的前缀，登录接口将会在[创建用户登录脚本](#)章节创建。

2. 在“数据”页签，单击“View API Get Connector”，设置“桥接器实例”为“通用AppCube API数据桥接器”，“数据类型”为“动态数据”，“请求方法”为“get”，如[图3-30](#)所示。

图 3-30 设置桥接器



自定义组件的“数据”参数说明如下：

- 桥接器实例：调用的桥接器名称。
- 请求方法：调用的方法名，如get（查询）、put（增加）、post（修改）和delete（删除）。
- 调用周期：此处不用配置。调用周期是每隔多少秒调用一次后台接口或获取静态数据，默认配置为“0”，表示只调用一次或只获取一次静态数据。

3. 参考上一步，分别设置如图3-31所示的其他桥接器实例。

图 3-31 设置其他桥接器

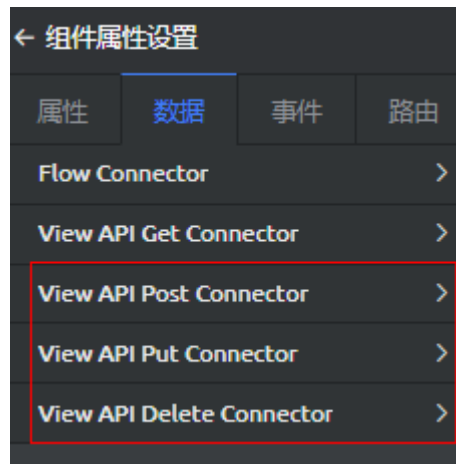


表 3-4 桥接器实例配置

数据名	桥接器实例	数据类型	Request Method
View API Get Connector (上一步已配置)	通用AppCube API数据桥接器	动态数据	get
View API Post Connector	通用AppCube API数据桥接器	动态数据	post
View API Put Connector	通用AppCube API数据桥接器	动态数据	put
View API Delete Connector	通用AppCube API数据桥接器	动态数据	delete

步骤5 设置自定义组件“userLogin”的“事件”，使自定义组件与其他页面关联。

图 3-32 需要配置的事件




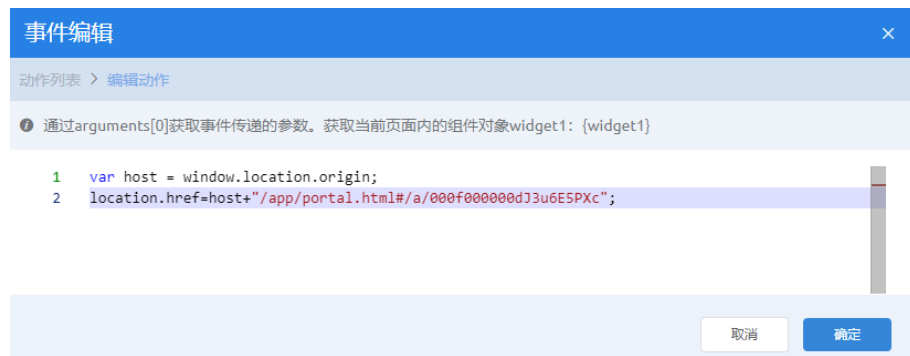
1. 单击选中“userLogin”组件，在右侧属性设置的“事件”页签下，单击“go Homepage”后的。
2. 单击“新建动作”，选择“自定义 > 自定义动作”。

图 3-33 配置事件



3. 在“事件编辑”中，输入事件代码，然后单击“确定”。

图 3-34 事件编辑



事件代码如下，其中加粗斜体代码请修改为实际运行环境主页URL：

```
var host = window.location.origin;  
location.href=host+ "/app/portal.html#/a/000f000000dJ3u6E5PXc";
```


其中，“/app/portal.html#/a/000f000000dJ3u6E5PXc”为当前应用运行环境Home页URL，可在预览应用中查看。即在左侧导航栏中，单击，进入应用预览，查看地址栏红框中URL。

图 3-35 应用预览入口

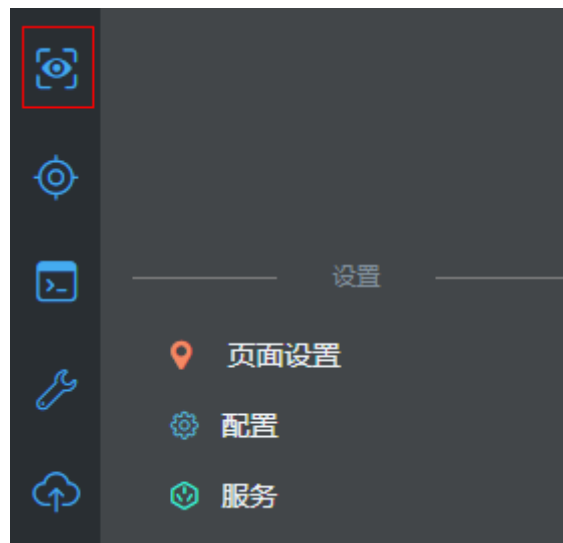
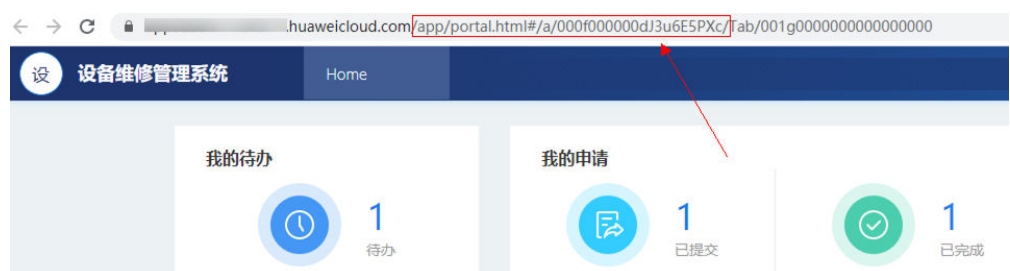


图 3-36 查看运行环境 Home 页 URL



步骤6 单击页面上部的, 保存页面。

步骤7 单击页面上部的，发布页面。

因为页面中配置的跳转目标页面、登录业务逻辑、相关账号等，当前还没有创建，因此，预览时，仅能查看登录页面的前端样式，不能进行登录操作。

说明

当前登录页中，输入业务账号及密码，单击“登录”按钮的登录逻辑是通过“自定义登录”组件，调用用户登录业务逻辑完成的。相关的业务逻辑（脚本、服务编排），将在[开发业务逻辑](#)中创建。

----结束

3.1.6 设备管理开发

3.1.6.1 开发前必读

设备管理功能主要是通过操作两个标准页面（编辑设备、设备管理）以及一个高级页面（设备监控页面），来实现电梯设备信息的新增、修改、删除、查询及监控等功能的。

设备管理界面的大致构想，如[图3-37](#)所示。界面上默认显示应用中保存的所有电梯设备信息，可以直接新增、修改或者删除电梯记录，也可以查询部分电梯记录。

- 单击“新增设备”按钮，可以在界面上插入一个空行，输入内容后单击“保存”，即可新插入一条电梯记录。
- 直接编辑表格中任意内容，单击“保存”，即可修改任意一条电梯记录。
- 选中记录，单击“删除”，即可删除任意一条电梯记录。
- 设置查询条件，单击“查询”，可以查询满足条件的电梯记录。

图 3-37 电梯设备信息管理界面



<input type="checkbox"/>	#	设备编码	设备名称	品牌	型号	地址	经度	纬度	操作
<input type="checkbox"/>	1	hw000123456	hw000123456	迅达	hw000123456	上海市虹口区...	116.466065999...	39.9099142778...	
<input type="checkbox"/>	2	test00001	1号电梯	日立	test00001	广东省深圳市龙...	114.053870303...	22.6556744647...	

通过开发设备管理功能，带您学习如何在AstroZero中，开发脚本、服务编排、标准界面和高级界面，并在开发过程中，细致解析设备管理的增加、删除和修改，以及设备位置信息监控功能开发过程及原理。

3.1.6.2 定义数据对象

3.1.6.2.1 背景和原理（对象）

AstroZero提供的数据对象（Object）定义功能，对应传统方式开发业务系统中的创建数据库表。每个Object对应一张数据库表，用于保存业务系统需要的配置数据和业务数据。

对象用于存储组织或者业务特有的数据，可理解为数据库中的数据表（逻辑表，系统实际存储时通过字段映射，统一把数据保存在数据库大宽表中）。

您可以围绕对象这一核心，定义相关的字段、字段校验规则、界面样式、字段变更时的触发事件等。如果把待开发的业务系统比作一部电影，对象就是电影中的一个角色，需要勾勒角色的外貌、性格特点、人物关系和所经历的剧情。

租户开发者可以自定义对象（Custom Object），平台允许租户增、删、改自定义对象，以及对象的字段，需要注意的是同一个租户账号下，创建的自定义对象的名称唯一不能重复。同时，平台上也有预置对象（Standard Object），您可以为这些标准对象新增字段，但不能修改、删除基线字段。

当自定义对象时，AstroZero会为这些Object自动创建一些标准字段（Standard Fields），如图3-38所示。

图 3-38 自定义对象的标准字段



字段名称	字段标签	数据类型	是否索引	操作
id	记录 ID	ID	✓	
name	Name	Name		✎
createdBy	创建人	Lookup (User, PortalUser)	✓	
createdAt	创建时间	DateTime		
lastModifiedBy	最后修改人	Lookup (User, PortalUser)	✓	
lastModifiedDate	最后修改时间	DateTime		
owner	所有人	Lookup (User, PortalUser, Queue)		
currencyIsoCode	币种	Text		
installedPackage	已安装的软件包	Lookup (PackageInstall)		
custom	自定义	CheckBox		

您自定义的字段将保存在上图中的“自定义字段”页签，更多关于对象模型的详细介绍请查看[定义对象](#)。

学习地图

如图3-39所示，通过本节的学习和实践，您可以初步了解“对象”和“标准页面”的基本概念和能力。

图 3-39 学习地图



3.1.6.2.2 方法和实践

对于设备管理功能，需要先创建一个设备对象Equipment，保存设备品牌、型号、资产编号等信息，如表3-5所示。

表 3-5 设备对象 Equipment 信息

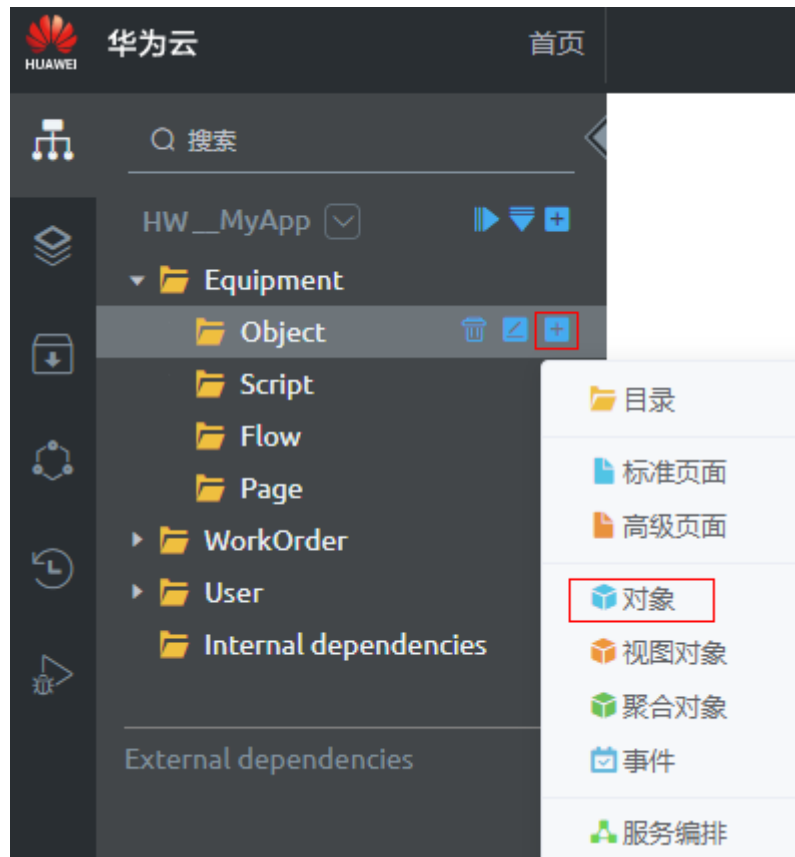
字段标签	字段名称	字段描述	数据类型	是否唯一	是否必填	建议长度
equipmentName	name (复用平台预置的标准字段)	设备名称	Name	是	NA	NA
latitude	latitude	设备位置纬度	文本	否	否	255
fullAddress	fullAddress	完整地址	文本	否	否	255
installationSiteCity	installationSiteCity	设备所在市	文本	否	否	255
installationSiteArea	installationSiteArea	设备所在区	文本	否	否	255
installationDetailAddress	installationDetailAddress	设备所在详细地址	文本	否	否	255
longitude	longitude	设备位置经度	文本	否	否	255
equipmentSN	equipmentSN	设备编码	文本, 区分大小写	是	是	255

字段标签	字段名称	字段描述	数据类型	是否唯一	是否必填	建议长度
equipmentBrand	equipmentBrand	设备品牌	选项列表 预置选项包括沃克斯、三菱、日立、迅达等。	否	否	-
equipmentModel	equipmentModel	设备型号	文本	否	否	255
installationSiteProvince	installationSiteProvince	设备所在省	文本	否	否	255

操作步骤

- 步骤1** 在AstroZero经典版开发环境首页“项目 > 我的应用”中，单击“设备维修管理系统”，进入之前创建的应用。
- 步骤2** 在“Equipment”中，将鼠标放在“Object”目录上，单击界面上出现的“+”，在弹出菜单中选择“对象”。

图 3-40 创建对象



步骤3 选中“创建新对象”，在“标签”和“名称”文本框中输入“Equipment”，单击“添加”。

系统实际创建的对象名称为“*HW_Equipment_CST*”，“*HW_*”前缀由租户命名空间namespace决定，“*_CST*”后缀代表自定义对象。在AstroZero中，同一个租户账号下，创建的自定义对象“名称”不能重复。

步骤4 将标准字段“Name”作为“设备名称”字段。


1. 在“标准字段”页签，单击“Name”字段后的编辑图标.
2. 如图3-41所示，修改字段标签为“equipmentName”，并设置字段内容唯一，单击“保存”。

图 3-41 修改 name 字段的标签

字段详情: [name](#)

基本信息

* 标签	名称	字段帮助	字段描述
equipmentNar	name	请输入	设备名称

展开

详细信息

数据类型	是否唯一	是否必填
Name	<input checked="" type="checkbox"/>	<input type="checkbox"/>

保存 取消

步骤5 按系统向导完成第1个字段“设备编码”（equipmentSN）的定义。

1. 在“自定义字段”页签，单击“新建”。

图 3-42 新建自定义字段

自定义对象: [HW_Equipment_CST](#)

基本信息 标准字段 自定义字段 验证规则 内嵌触发器 布局

新建 批量创建 导出 自定义索引

名称	标签	数据类型	是否索引	是否必填
----	----	------	------	------

共 0 条

2. 在选中字段类型中，设置字段类型为“文本”，单击“下一步”。
3. 输入字段详细信息，如图3-43所示，设置字段标签、名称为“equipmentSN”、字段长度255，并定义字段为必填字段、是唯一且大小写敏感，单击“下一步”。

注意

平台会根据字段标签自动生成字段名称，但请参照表3-5，修改字段名。对于utf-8编码，一个汉字占用三个字节。

图 3-43 定义“设备编码”详细信息

系统实际创建的字段名称为“**HW**_equipmentSN_CST”，“**HW**”前缀由租户命名空间namespace决定，“_CST”后缀代表自定义字段。

- 配置字段级安全。选中“读取”和“编辑”复选框，为所有预置profile配置能编辑和读取本字段的权限，单击“下一步”。
- 将字段添加到页面布局。选中“添加本字段到该页面布局”，单击“保存”。

步骤6 您可按系统向导完成第2个字段“设备品牌”（equipmentBrand）的定义。

- 在“自定义字段”页签，单击“新建”。
- 设置字段类型为“选项列表”，单击“下一步”。
- 如**图3-44**所示，设置字段标签、名称，输入可选值，添加选项列表的标签和值，单击“下一步”。

说明

AstroZero会根据字段标签自动生成字段名称，但请参照**表3-5**，修改字段名。

图 3-44 定义“设备品牌”详细信息

- 选中“权限集”后“读取”和“编辑”复选框，为所有预置profile配置能编辑和读取本字段的权限，单击“下一步”。

5. 选中“添加本字段到该页面布局”，单击“保存”。

步骤7 创建表3-5中剩余字段。这里有两种方法，请选择合适的方式进行创建。

- 方法一：参考上一步，分别创建表3-5中其他属性。
- 方法二：通过导入字段模板，批量快速创建剩余字段模板。
 - a. 单击import-fields-equipment.xlsm，请待导入字段模板下载到本地。
 - b. 单击当前对象的“自定义字段”，然后单击“批量创建”后的▾，选择“批量导入字段”，选择上一步下载的字模板。导入完成后，刷新页面，重新进入对象的“自定义字段”，即可查看已导入的字段。

图 3-45 导入自定义字段



图 3-46 查看已创建自定义字段

自定义对象: HW_Equipment_CST

基本信息 标准字段 自定义字段 验证规则 内嵌触发器 布局

名称	标签	数据类型	是否索引	是否必填	描述	最后修改人	最后修改时间	操作
HW_latitude_CST	latitude	文本	<input type="checkbox"/>	<input type="checkbox"/>	设备经纬度		2020-09-14 17	🗑️
HW_fullAddress_CST	fullAddress	文本	<input type="checkbox"/>	<input type="checkbox"/>	完整地址		2020-09-14 17	🗑️
HW_installationSiteCity_CST	installationSiteCity	文本	<input type="checkbox"/>	<input type="checkbox"/>	设备所在市		2020-09-14 17	🗑️
HW_installationSiteArea_CST	installationSiteArea	文本	<input type="checkbox"/>	<input type="checkbox"/>	设备所在区		2020-09-14 17	🗑️
HW_installationDetailAddress_CST	installationDetailAddress	文本	<input type="checkbox"/>	<input type="checkbox"/>	设备所在详细地址		2020-09-14 17	🗑️
HW_longitude_CST	longitude	文本	<input type="checkbox"/>	<input type="checkbox"/>	设备经纬度		2020-09-14 17	🗑️
HW_equipmentSN_CST	equipmentSN	文本	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	设备编码		2020-09-14 17	🗑️
HW_equipmentBrand_CST	equipmentBrand	选项列表	<input type="checkbox"/>	<input type="checkbox"/>	设备品牌		2020-09-14 17	🗑️
HW_equipmentModel_CST	equipmentModel	文本	<input type="checkbox"/>	<input type="checkbox"/>	设备型号		2020-09-14 17	🗑️
HW_installationSiteProvince_CST	installationSiteProvince	文本	<input type="checkbox"/>	<input type="checkbox"/>	设备所在省		2020-09-14 17	🗑️

---结束

验证

您可以用对象的页面布局Layout能力，检查对象的定义是否符合预期。

创建对象时，系统会自动为对象创建两个页面布局：

- 设备详情页面Equipment Detail：显示对象中单条记录的详细信息。如果手工创建字段时选中了“添加本字段到该页面布局”，设备详情页面包含name和添加的自定义字段。脚本创建的默认不添加，但是不影响后续的操作。

- 设备列表页面Equipment Records：显示对象中的记录列表。列表页面中默认只显示了name字段（设备名称）。因为后续章节的测试中，经常需要检查记录ID、设备编码和设备名称，所以将其他几个字段也添加到列表页面中。

步骤1 修改设备对象的列表页面布局。

1. 单击设备对象HW_Equipment_CST的“布局”页签。
2. 单击“Equipment Records”，打开页面详情。

图 3-47 对象布局

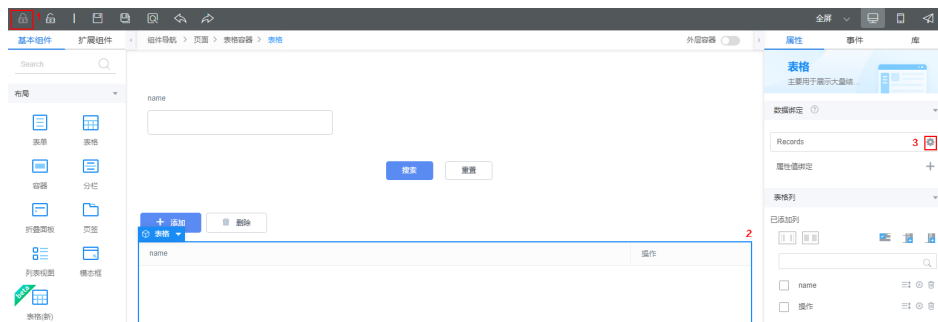


3. 选中列表“表格”，如**图3-48**所示，再单击界面右侧“属性”页签中的⚙️。

说明

如果当前页面已锁定，请先单击🔒，解锁页面。

图 3-48 绑定 Records



4. 单击Records后的编辑图标✎。

图 3-49 编辑 Records

名称	绑定组件	来源	映射	操作
Records_condition		-	-	+ ✎ 🗑️
Records	datagrid_1 Records	Object	HW__Equipment__CST	+ ✎ 🗑️
showModal	modal_0	-	-	+ ✎ 🗑️
Detail	Detail	HW__Equipment__CST	HW__Equipment__CST	+ ✎ 🗑️
showUserModal	modal_1	-	-	+ ✎ 🗑️
approvalUser	userselect	-	-	+ ✎ 🗑️

- 选中要显示在页面上的字段，如图3-50所示，单击“下一步”，再单击“确定”。

图 3-50 为 Records 添加字段

编辑

① 基本信息 ———— ② 设置 ———— ③ 方法


* 选择对象

HW__Equipment__CST


* 选择字段

请输入字段名称进行搜索

<input type="checkbox"/>	字段名称	字段标签
<input type="checkbox"/>	id	记录 ID
<input checked="" type="checkbox"/>	name	equipmentName
<input checked="" type="checkbox"/>	HW__equipmentSN__CST	equipmentSN
<input checked="" type="checkbox"/>	HW__equipmentBrand__CST	equipmentBrand
<input checked="" type="checkbox"/>	HW__equipmentModel__CST	equipmentModel
<input type="checkbox"/>	HW__installationSiteProvince__CST	installationSiteProvince
<input type="checkbox"/>	HW__installationSiteCity__CST	installationSiteCity

- 在图3-49对话框中，单击选中“Records”，再单击“确定”。
- 单击编辑器上方的，保存设置。

步骤2 预览、测试对象。

- 单击页面上的，进入预览页面。
- 单击“添加”，再输入任意测试数据，单击“保存”。

系统返回页面，并显示新插入的数据，则表示对象创建成功。

----结束

3.1.6.3 开发“编辑设备”功能

3.1.6.3.1 背景和原理

本节主要通过创建一个标准页面，调用一个具有编辑设备功能的脚本，实现编辑设备信息功能。在进行开发前，您需要先了解脚本、公共接口以及标准页面的相关知识。

学习地图

如图3-51所示，通过本章的学习和实践，您将了解“标准页面”的能力，并掌握脚本的开发方法。

图 3-51 学习地图



脚本

对应相对复杂的业务逻辑，AstroZero提供了脚本、服务编排等后台逻辑形式。本章节主要带您了解脚本的基础知识。

AstroZero的脚本引擎采用TypeScript语言。脚本执行时，TypeScript语言会被翻译成JavaScript语言，由JavaScript引擎执行。在JavaScript es5的官方标准库外，AstroZero还扩展了10+内置功能库，帮助您更高效地开发，如表3-6所示。除此之外，AstroZero还提供了代码智能补全、代码格式化、语法错误提示等功能，帮助您高效编辑代码。

脚本编辑界面如图3-52所示，您可在线编辑、测试和发布脚本。

图 3-52 脚本编辑界面

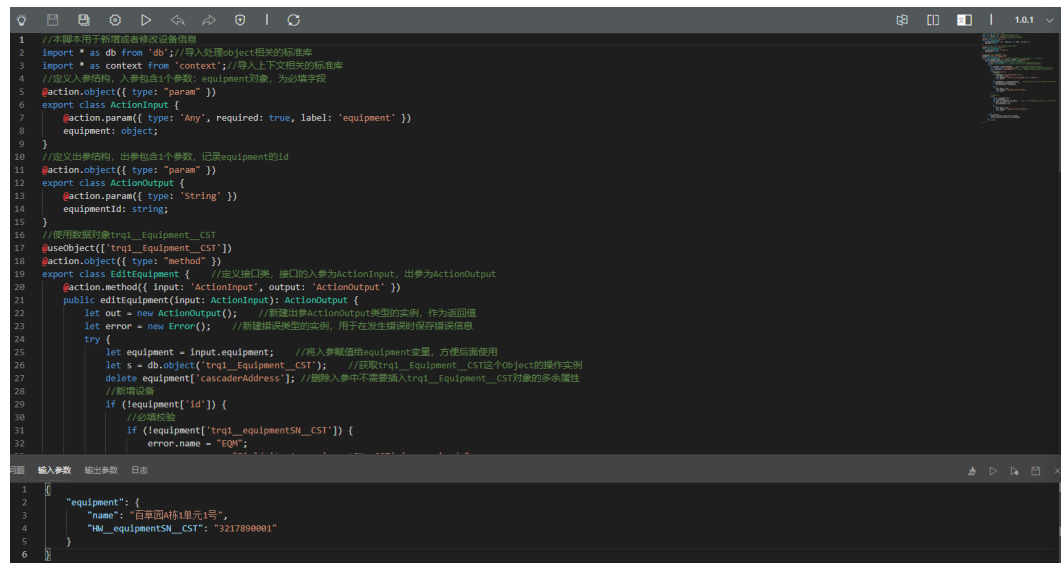


表 3-6 标准库说明

标准库名称	说明
sys	提供系统层面的操作，例如获取系统参数值。
context	脚本执行的上下文，例如获取当前租户ID、获取当前登录用户ID等。
action	定义脚本的输入输出方法，以及调用脚本的方法。
buffer	操作二进制缓存区，例如将缓冲区内容转换为字符串。
setup	对系统对象Standard Object的增、删、改、查。
db	对自定义对象Customer Object的增、删、改、查。
sql	执行sql语句，目前只支持执行select查询语句。
bo	对Business Object的增、删、改、查。
meta	操作object的元数据，例如获取字段名称、获取字段长度。
http	http或者https操作。
iconv	将字符串转为GBK、GB18030、GB2132或ISO8859_1编码格式，或者解码字符串。
crypto	加密操作，包含对哈希、HMAC、加密、解密、签名及验证的封装。
decimal	将从http请求或从数据库中查询出来的number类型值，转换成Decimal类型，并能进行加、减、乘、除等数学运算。
excel	操作Excel文件，例如生成Excel文件。
text	操作文本，例如将文本转换为二进制。

标准库名称	说明
xml	操作XML文件，例如读取XML文件内容。
uuid	生成全局唯一标识。

样例代码解读

通过下面详细的脚本代码内容解读，使您对脚本有一个更具体的认识。

一般情况下，编写脚本的大致流程为：

1. 按需引入平台标准库。

图 3-53 引入平台标准库

```
import * as db from 'db';//导入处理object相关的标准库
import * as context from 'context';//导入上下文相关的标准库
import * as decimal from 'decimal';//导入decimal数据类型相关的标准库
```

2. 定义出参、入参结构。

图 3-54 定义入参

```
//定义入参结构
@action.object({ type: "param" })
export class ActionInput {
    @action.param({ type: 'String', required: true })
    equipmentId: string;//设备ID
}
```

图 3-55 定义出参

```
//定义出参结构
@action.object({ type: "param" })
export class ActionOutput {
    @action.param({ type: 'Any', label: 'equipment' })
    equipment: object;//设备对象
}
```

3. 定义方法以及使用的对象。

图 3-56 定义方法及使用对象

```
@useObject(['HWT__Equipment__CST'])//使用数据库对象Equipment__CST
@action.object({ type: "method" })
export class QueryEquipmentDetail {
    @action.method({ input: 'ActionInput', output: 'ActionOutput' })
    public queryEquipmentDetail(input: ActionInput): ActionOutput {
```

4. 进行数据库操作。

图 3-57 数据库相关操作

```
//获取HWT__Equipment__CST这个Object的操作实例
let s = db.object('HWT__Equipment__CST');
//查询字段(全部)
let option = {};
//查询条件
let condition = {
  "conjunction": "AND",
  "conditions": [{
    "field": "id",
    "operator": "eq",
    "value": input.equipmentId
  }]
};
//调用按条件查询Equipment__CST的接口
let record = s.queryByCondition(condition, option);
```

下面我们通过解读以下脚本样例，了解一个脚本的总体结构框架、编写要求。

```
import * as decimal from 'decimal';

@action.object({type: "param"})
export class ActionInput {
  @action.param({type: 'String', required: true, label: 'your name', description: 'please input your name'})
  name: string;

  @action.param({type: 'Number', required: true, min: 1, max: 100, message: 'age must during [1, 100]'})
  age: decimal.Decimal;

  @action.param({type: 'Date', pattern: 'yyyy-MM-dd'})
  birthday: Date;

  @action.param({type: 'String', isCollection: true})
  schools: string[];

  @action.param({type: 'Boolean'})
  married: boolean;

  @action.param({type: 'MyObject'})
  obj: MyObject;
}

@action.object({type: "param"})
export class MyObject {
  @action.param({type: 'String'})
  something: string;
  @action.param({type: 'Number'})
  otherthing: decimal.Decimal;
}

@action.object({type: "param"})
export class ActionOutput {
  @action.param({type: 'String', isCollection: true})
  greets: string[];
}

@action.object({type: "method"})
export class ActionDemo {
  @action.method({ label: 'greeting something', description: 'greeting something.', input: 'ActionInput',
  output: 'ActionOutput' })
  public greet(inarg: ActionInput): ActionOutput {

    console.log('name = ', inarg.name);
    console.log('age = ', inarg.age);
    console.log('birthday = ', inarg.birthday);
    console.log('schools = ', inarg.schools);
```

```
console.log('married = ', inarg.married);
console.log('obj = ', inarg.obj);

let out = new ActionOutput();
out.greets = ['hello', 'hi', 'how are you', 'how old are you', 'long time no see'];
return out;
}
}
```

上述示例脚本包括三部分内容：

1. 导入标准库或其他模块。

示例中第1行，表示将使用平台提供的decimal库。

```
import * as decimal from 'decimal';
```

除了平台预置标准库，还可以声明对其他自定义模块的引用。例如，已经提前开发了一个脚本circle，可以用如下方式加载。

```
import * as circle from './circle';
```

2. 定义输入、输出变量。

脚本可以有多个输入、输出参数，也可以没有。所有的输入或输出参数必须封装在一个class中，作为实例成员。

本例中，脚本有6个输入参数，被封装为ActionInput。每个参数都必须定义其参数类型，同时还可以定义是否必填、标签、最大值、最小值等可选属性。

```
@action.object({type: "param"})
export class ActionInput {
  @action.param({type: 'String', required: true, label: 'your name', description: 'please input your name'})
  name: string;

  @action.param({type: 'Number', required: true, min: 1, max: 100, message: 'age must during [1, 100]'})
  age: decimal.Decimal;

  @action.param({type: 'Date', pattern: 'yyyy-MM-dd'})
  birthday: Date;

  @action.param({type: 'String', isCollection: true})
  schools: string[];

  @action.param({type: 'Boolean'})
  married: boolean;

  @action.param({type: 'MyObject'})
  obj: MyObject;
}
```

因为第6个输入参数“obj”的参数类型为自定义对象，所以还需要给出“MyObject”的定义。

```
@action.object({type: "param"})
export class MyObject {
  @action.param({type: 'String'})
  something: string;
  @action.param({type: 'Number'})
  otherthing: decimal.Decimal;
}
```

脚本有1个输出参数，被封装为ActionOutput。

```
@action.object({type: "param"})
export class ActionOutput {
  @action.param({type: 'String', isCollection: true})
  greets: string[];
}
```

3. 定义方法。

样例中，ActionDemo是外部调用的class，使用export导出。ActionDemo定义了一个action method，使用action.method装饰，表明调用脚本时从此方法入口。greet是class的实例方法，其输入、输出参数就是前面定义的ActionInput和ActionOutput。

在一个脚本文件里面，action.method只能使用一次。

```
@action.object({type: "method"})
export class ActionDemo {
  @action.method({ label: 'greeting something', description: 'greeting something.', input:
'ActionInput', output: 'ActionOutput' })
  public greet(inarg: ActionInput): ActionOutput {

    console.log('name = ', inarg.name);
    console.log('age = ', inarg.age);
    console.log('birthday = ', inarg.birthday);
    console.log('schools = ', inarg.schools);
    console.log('married = ', inarg.married);
    console.log('obj = ', inarg.obj);

    let out = new ActionOutput();
    out.greets = ['hello', 'hi', 'how are you', 'how old are you', 'long time no see'];
    return out;
  }
}
```

脚本编辑页面不支持单步调试，样例里的console.log可实现在日志里打印过程输出，方便代码调试。

公共接口

公共接口是对脚本、服务编排和对象进行再包装的一种方式。将创建的脚本、服务编排、对象包装成一个新公共服务，可以使得接口的URL地址的表达形式更规范，方便让前端页面或第三方系统进行调用。

因此，在创建完后台逻辑后（服务编排、脚本、对象），就需要先将此接口包装成标准的公共接口，才能被调用。本章节中创建了2个业务脚本，因此需要对应创建2个公共接口，供前端页面调用。

创建公共接口的入口：在APP视图下，单击下方“服务”，即可进入公共接口创建页面。

图 3-58 创建公共接口入口

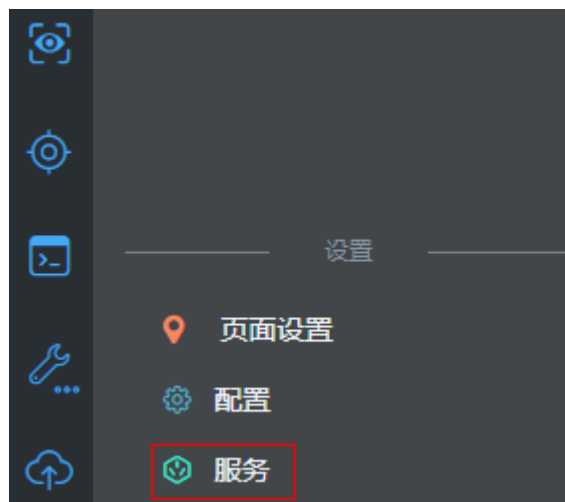


图 3-59 公共接口基本信息

新建公共接口

通过定义服务的api, 可迅速满足您定制所需要的业务接口, 并将该接口服务注册到网关, 供第三方使用。

标签

* 操作名称

* 版本

* URL 以"/"开头

内容类型

分类

描述

允许匿名访问

类型

服务编排 脚本 对象

自定义响应

* 资源

* 方法

图 3-60 公共接口 URL

公共接口

使用公共接口, 可以将flow的URL映射到发布到外部网关, 第三方可以通过OAuth2.0调用。

新建 预览

操作名称	版本	URL	方法	类型
amountPerStatus	0.0.1	/service/lgj__BeerMgtApp/0.0.1/amountPerStatus	POST	流
amountByWeekPerStatus	0.0.1	/service/lgj__BeerMgtApp/0.0.1/amountByWeekPerSt...	POST	流

标准页面

相对于高级页面, 标准页面主要用于对前端页面的样式要求相对简单的场景, 这种页面一般只是针对业务数据的增、删、改、查等基础功能。

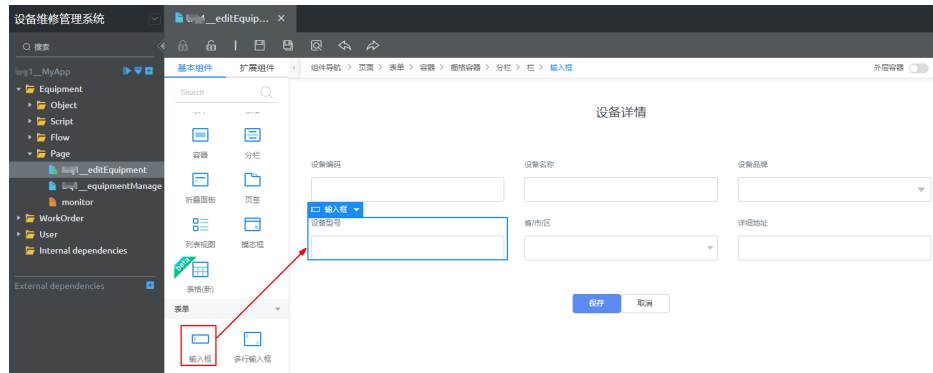
标准页面主要是通过对前端页面组件的组合编排, 以及为组件配置属性、关联事件实现业务功能的, 在[组装“编辑设备”页面](#)章节, 则会详细介绍组件、组件属性以及事件之间的调用及相关配置。

标准页面的编辑环境也称为UI Builder, UI Builder是图形化、无码化的前端页面在线开发工具。借助标准页面, 您只需要了解基本前端编码知识, 通过将页面组件拖拽至“设计视图”的画布中, 再进行必要的属性、事件配置, 就可以完成页面的开发。

设计视图

标准页面中预置了多种组件, 在开发页面时, 可以直接从左侧组件区域, 将这些组件拖拽到右侧“页面内容”中。

图 3-61 标准页面编辑界面 (UI Builder)



配置组件属性

- 了解组件使用场景及配置方法

在配置使用组件前，可以将光标放在每个组件上后，组件右上角将显示帮助信息的问号图标。单击问号图标，即可进入该组件介绍页面，了解并学习AstroZero预制前端组件的使用场景及参数配置方法。

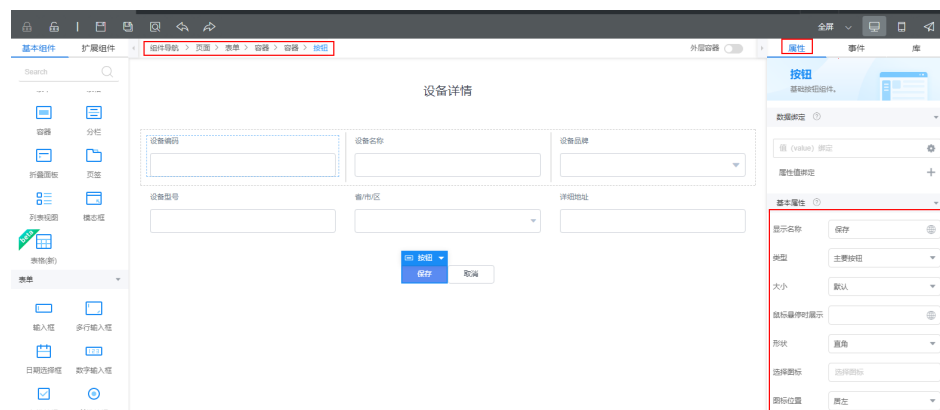
图 3-62 组件帮助信息入口



- 选中组件，查看组件属性

在“设计视图”中，选中一个页面组件，可以在右侧“属性”页签，设置该组件的绑定数据、样式等属性，如图3-63所示。在设计视图中选中一个文本框，可以在右侧修改这个文本框的标签名、样式、绑定的数据模型等。

图 3-63 设置组件的属性



- 页面及组件的事件代码

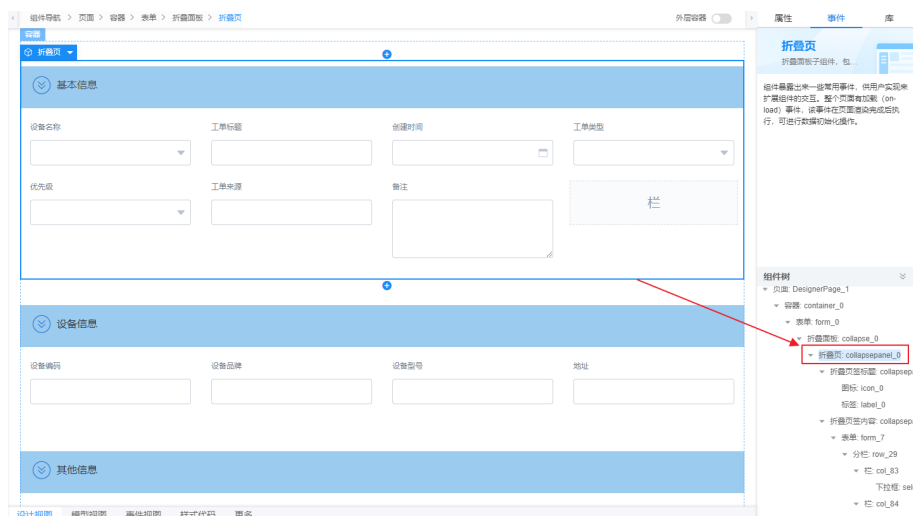
在“设计视图”中，选中任意组件，可以在右侧“事件”页签，设置组件的关联事件。例如，我们可以选择一个“按钮”，然后在按钮的“点击”事件中，添加相应的事件代码，实现在单击按钮后将界面输入保存到数据库中。

图 3-64 编辑组件的关联事件



- 利用组件导航，快速选中组件
当选中某个组件时，页面上方会在组件导航上显示它的html标签层级。因此，在组件数量比较多，位置较为紧密时，可以直接单击标签层级，快速选择组件，也可以快速切换组件。例如，在上图的组件导航中单击“页面”，可以直接选中页面上的最外层页面组件。
- 利用组件树，快速选中组件
在组件数量比较多，位置较为紧密时，也可以单击页面右下角的“组件树”，展开组件树，在组件树中，直接单击标签层级，快速选择组件，也可以快速切换组件。

图 3-65 利用组件树快速选中组件

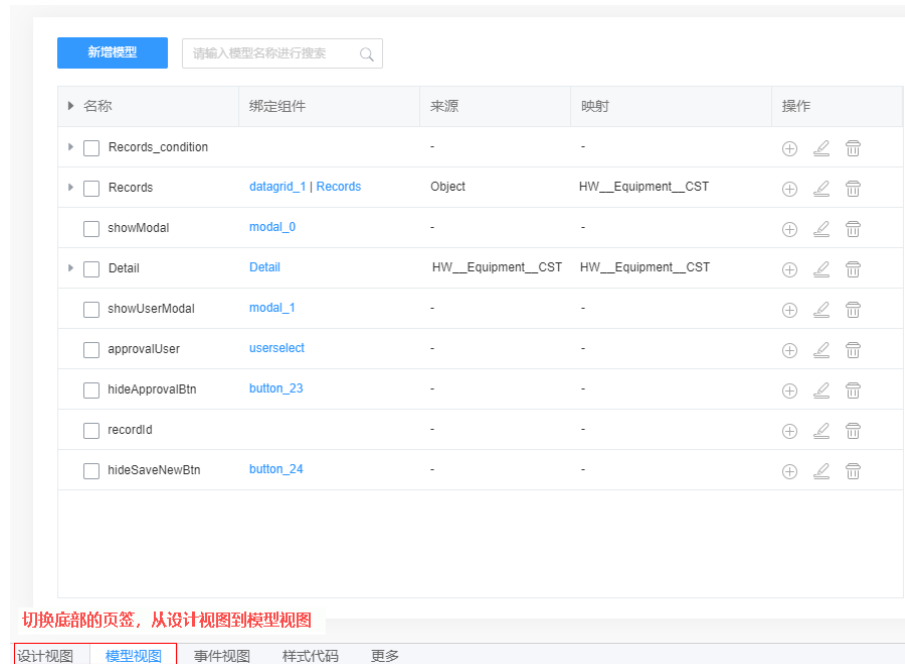


模型视图

标准页面是通过数据模型驱动的，页面所有的逻辑都是围绕数据模型展开的。在将模型与前台页面组件或者后台逻辑绑定后，开发者只需要关注模型数据的实例化和处理，不需要关注页面的渲染和展示。

在页面底部，单击“模型视图”，即可从“设计视图”切换到“模型视图”，在模型视图下，可以查看、编辑以及管理数据模型，如图3-66所示。

图 3-66 模型视图



模型定义

标准页面支持表3-7中四类模型，每类模型都包含参数定义和方法定义。方法是在模型上定义的API，通常会在前端组件关联的事件脚本（例如页面加载事件、鼠标单击事件）中调用这些API，以实现一定的逻辑。

表 3-7 模型说明

模型分类	模型说明	模型参数的定义	模型方法的定义	API调用方法
自定义模型	自定义模型是由开发者自由定义的模型。	参数由开发者自定义，可以添加子节点。	开发者自定义方法。	<code>\$model.ref("modelName").actionName();</code>
对象模型	对象模型是直接与Object对象表映射生成的。	系统自动获取Object所有的字段，开发者可以从中选择部分字段作为参数。	平台自动生成查询、保存、删除和统计4个方法。	<ul style="list-style-type: none"> <code>\$model.ref("modelName").query(param);</code> <code>\$model.ref("modelName").save();</code> <code>\$model.ref("modelName").delete();</code> <code>\$model.ref("modelName").count();</code>

模型分类	模型说明	模型参数的定义	模型方法的定义	API调用方法
服务模型	服务模型是与后台服务映射生成的，当前支持与服务编排或Script映射。	参数根据后台服务的入参、出参映射生成为inputParam和outputParam节点。	平台自动生成了run方法，用于执行模型关联的服务编排或者Script。	<code><i>\$model.ref("modelName").run();</i></code>
事件模型	事件模型是与后台事件的字段映射生成的，并且支持websocket刷新模型数据。	参数根据后台事件的字段映射生成。	平台自动生成了run方法，用于执行模型关联的服务编排或者Script。	<code><i>\$model.ref("modelName").run();</i></code>

除了在模型中定义的方法，AstroZero还提供了如下标准API：

- 获取模型数据：`$model.ref("modelName").getData();`
- 设置模型数据：`$model.ref("modelName").setData();`
- 设置模型字段值：`$model.ref("modelName").setValue(key,value);`

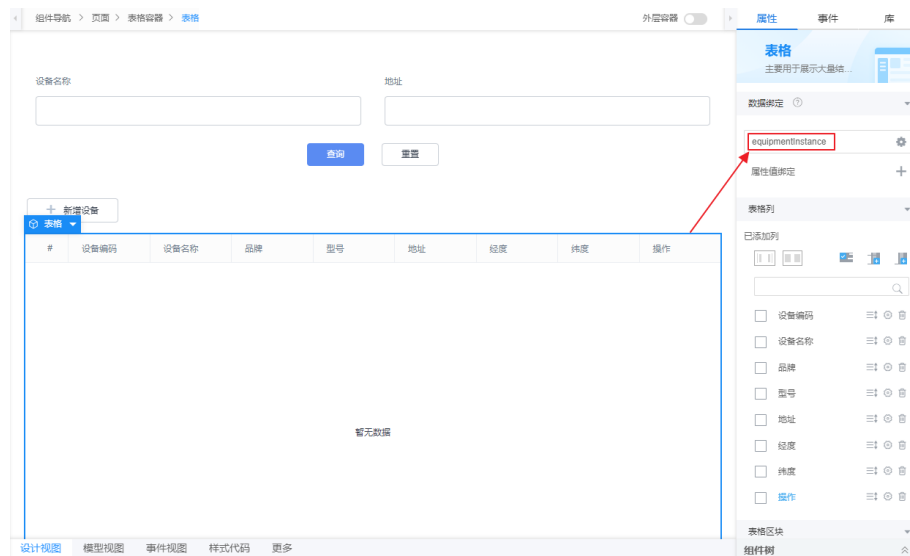
模型绑定

模型与前端组件的绑定分为值绑定和属性绑定，绑定会在指定组件上创建双向数据绑定。

- 典型的值绑定场景有：表单、表格、列表视图对应的model绑定，以及输入框、下拉框等基础组件对应的value绑定，类似Vue的v-model。
- 典型的属性绑定场景有：下拉框的选项值、步骤条的步骤值等，类似Vue的v-bind。

如图3-67所示的例子中，实现了表格DataGrid组件与值的绑定。

图 3-67 数据绑定



3.1.6.3.2 创建业务逻辑

“编辑设备”功能主要通过调用并执行一个具有“编辑设备”功能的脚本，实现编辑设备信息功能，并在修改设备信息时，能够根据设备id号，直接修改设备信息，因此我们需要在开发页面前，先创建一个“编辑设备”和“按ID查询设备详情”脚本。

创建“编辑设备”脚本

“编辑设备”脚本实现的功能是向设备对象中插入1条新记录，或者更新1条记录。

操作步骤

- 步骤1** 进入“设备维修管理系统”应用。
- 步骤2** 在“Equipment”目录中，将鼠标放在“Script”上，单击界面上出现的“+”，在弹出菜单中选择“脚本”。
- 步骤3** 在弹窗中，选中“创建一个新脚本”，在“名称”文本框中输入“editEquipment”，单击“添加”。

系统实际创建的脚本名称为“*HW*_editEquipment”，“*HW*”前缀由租户命名空间namespace决定。新建创建的脚本，默认是当前用户锁定状态，可以进行编辑保存等操作。

当编辑已有脚本时，为防止编辑时多人篡改，编辑前请单击，进行锁定。


- 步骤4** 在代码编辑器中，插入如下脚本代码。

须知

脚本中**红色内容**请替换为实际的对象名、字段名。

```
//本脚本用于新增或者修改设备信息
import * as db from 'db';//导入处理object相关的标准库
import * as context from 'context';//导入上下文相关的标准库
```

```
//定义入参结构，入参包含1个参数：equipment对象，为必填字段
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'Any', required: true, label: 'equipment' })
  equipment: object;
}
//定义出参结构，出参包含1个参数，记录equipment的id
@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'String' })
  equipmentId: string;
}
//使用数据对象HW__Equipment__CST
@useObject(['HW__Equipment__CST'])
@action.object({ type: "method" })
export class EditEquipment { //定义接口类，接口的入参为ActionInput，出参为ActionOutput
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public editEquipment(input: ActionInput): ActionOutput {
    let out = new ActionOutput(); //新建出参ActionOutput类型的实例，作为返回值
    let error = new Error(); //新建错误类型的实例，用于在发生错误时保存错误信息
    try {
      let equipment = input.equipment; //将入参赋值给equipment变量，方便后面使用
      let s = db.object('HW__Equipment__CST'); //获取HW__Equipment__CST这个Object的操作实例
      delete equipment['cascaderAddress']; //删除入参中不需要插入HW__Equipment__CST对象的多余属性
      //新增设备
      if (!equipment['id']) {
        //必填校验
        if (!equipment['HW__equipmentSN__CST']) {
          error.name = "EQM";
          error.message = "Field 'HW__equipmentSN__CST' is required.";
          throw error;
        }
        let equipmentId = s.insert(equipment); //向HW__Equipment__CST插入一条数据，返回数据的唯一标识即设备ID
        if (equipmentId && equipmentId != "") {
          out.equipmentId = equipmentId;
        }
        else {
          error.name = "EQM";
          error.message = "Equipment Cannot Be Added.";
          throw error;
        }
      }
      //编辑修改设备
      else {
        let id = equipment['id'];
        delete equipment['id'];
        let count = s.update(id, equipment); //根据设备ID，编辑更新HW__Equipment__CST的一条数据
        if (count && count == 1) {
          out.equipmentId = id;
        }
        else {
          error.name = "EQM";
          error.message = "Equipment Cannot Be Updated.";
          throw error;
        }
      }
    } catch (error) {
      console.error(error.name, error.message);
      context.setError(error.name, error.message);
    }
    return out;
  }
}
```

步骤5 单击编辑器上方的，保存脚本。

----**结束**

验证并发布

步骤1 测试新增逻辑能否正常执行。



1. 单击编辑器上方的，执行脚本。
2. 如图3-68所示，在界面底部输入测试数据，单击测试窗口右上角执行图标。

图 3-68 测试脚本



测试报文采用json格式，样例如下（报文中加粗斜体内容请替换为实际的字段名）：

```
{
  "equipment": {
    "name": "百草园A栋1单元1号",
    "HW_equipmentSN_CST": "3217890001"
  }
}
```

执行成功，会在“输出”页签返回equipmentId。请保存这个返回结果，后续的测试中会用到。

```
{
  "equipmentId": "cQue000000e1qnhgtCng"
}
```

如果执行失败，请检查之前设备对象、脚本，以及测试报文三者中的对象名、字段名是否一致。

步骤2 到设备对象布局页面（Equipment Records），预览页面，检查数据是否插入成功。



1. 在“Equipment”目录的“Object”下，单击设备对象“HW_Equipment_CST”，在“布局”页签下，单击“Equipment Records”后的预览图标。


图 3-69 对象布局页面



2. 在页面中，检查设备列表中是否包含刚插入的测试数据。
3. 如果已新增数据，为了后续测试方便，建议多创建几条数据。

步骤3 测试修改逻辑能否正常执行。

1. 单击编辑器上方的，执行脚本。


- 在界面底部输入测试数据，单击测试窗口右上角执行图标。

须知

如下样例报文中的加粗id值，请修改为图3-68的返回结果，加粗斜体字段名请替换为实际的字段名。以下报文是修改name字段。

```
{
  "equipment": {
    "id": "cQuXXXXXXXXng",
    "name": "百草园B栋2单元2号",
    "HW_equipmentSN_CST": "3217890001"
  }
}
```

- 刷新设备对象布局页面（Equipment Records）的预览页面，查看测试数据是否符合预期。

步骤4 测试成功，单击编辑器上方的，启用脚本。

----结束

创建“按 ID 查询设备详情”脚本

修改已存在的设备信息，则需要创建可以根据设备ID查询出设备详情的脚本。


操作步骤

步骤1 进入“设备维修管理系统”应用。

步骤2 在“Equipment”中，将鼠标放在“Script”目录上，单击界面上出现的“+”。

步骤3 在弹窗中，选中“创建一个新脚本”，在“名称”文本框中输入“queryEquipmentDetail”，单击“添加”。

系统实际创建的脚本名称为“HW_queryEquipmentDetail”，“HW_”前缀由租户命名空间namespace决定。新建创建的脚本，默认是当前用户锁定状态，可以进行编辑保存等操作。

当编辑已有脚本时，为防止编辑时多人篡改，编辑前请单击进行锁定。


步骤4 在代码编辑器中插入如下脚本代码。

须知

脚本中**红色内容**请替换为实际的对象名、字段名。

```
/* *****
 * 本脚本用于按设备ID查询设备详情
 * ***** */
import * as db from 'db'; // 导入处理object相关的标准库
import * as context from 'context'; // 导入上下文相关的标准库
// 定义入参结构
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'String', required: true })
  equipmentId: string; // 设备ID
```



```
}
//定义出参结构
@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'Any', label: 'equipment' })
  equipment: object;//设备对象
}
@useObject(['HW__Equipment__CST'])//使用数据库对象HW__Equipment__CST
@action.object({ type: "method" })
export class QueryEquipmentDetail {
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public queryEquipmentDetail(input: ActionInput): ActionOutput {
    let out = new ActionOutput(); //新建出参ActionOutput类型的实例，作为返回值
    let error = new Error(); //新建错误类型的实例，用于在发生错误时保存错误信息
    try {
      //必填校验
      if (!input.equipmentId || input.equipmentId == "") {
        error.name = "EQM";
        error.message = "Equipment id is required.";
        throw error;
      }
      //获取HW__Equipment__CST这个Object的操作实例
      let s = db.object('HW__Equipment__CST');
      //查询字段(全部)
      let option = {};
      //查询条件
      let condition = {
        "conjunction": "AND",
        "conditions": [{
          "field": "id",
          "operator": "eq",
          "value": input.equipmentId
        }]
      };
      //调用按条件查询Equipment__CST的接口
      let record = s.queryByCondition(condition, option);
      //如果查询到数据
      if (record && record[0]) {
        //拼接前台省市区级联选择器的数据模型
        let cascaderAddress = [];
        if (record[0].HW__installationSiteProvince__CST) {
          cascaderAddress.push(record[0].HW__installationSiteProvince__CST);
          if (record[0].HW__installationSiteCity__CST) {
            cascaderAddress.push(record[0].HW__installationSiteCity__CST);
            if (record[0].HW__installationSiteArea__CST) {
              cascaderAddress.push(record[0].HW__installationSiteArea__CST);
            }
          }
        }
        record[0].cascaderAddress = cascaderAddress;
        //将结果挂入输出对象中
        out.equipment = record[0]
      }
    } catch (error) {
      console.error(error.name, error.message);
      context.setError(error.name, error.message);
    }
    return out;
  }
}
```

步骤5 单击编辑器上方的, 保存脚本。

----结束

验证并发布

步骤1 测试新增逻辑能否正常执行。

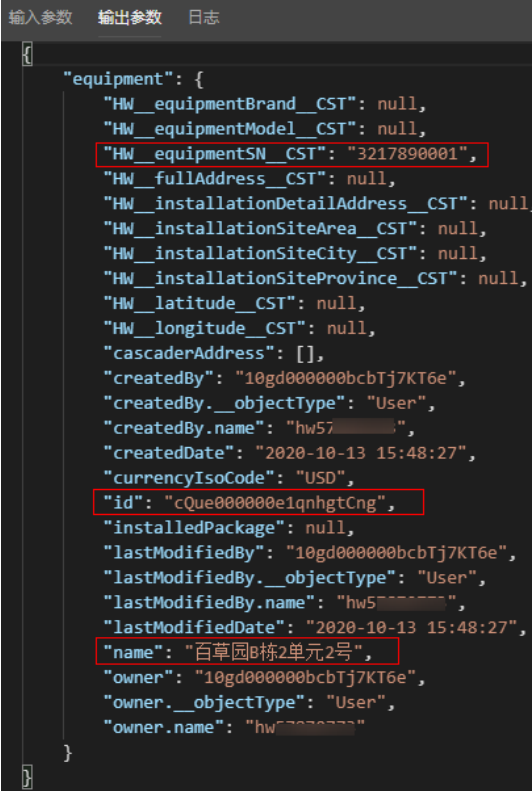
1. 单击编辑器上方的 , 执行脚本。
2. 在界面底部输入测试数据, 单击测试窗口右上角  执行图标。

测试报文样例如下, equipmentId可参考[创建“编辑设备”脚本](#)验证新增设备时生成的设备ID:


```
{  
  "equipmentId": "cQue000000e1qnhgtCng"  
}
```

执行成功, 会在“输出”页签返回查询结果, 返回结果是上一节中新增的设备信息。

图 3-70 查询返回的设备信息



```
输入参数  输出参数  日志  
{  
  "equipment": {  
    "HW_equipmentBrand_CST": null,  
    "HW_equipmentModel_CST": null,  
    "HW_equipmentSN_CST": "3217890001",  
    "HW_fullAddress_CST": null,  
    "HW_installationDetailAddress_CST": null,  
    "HW_installationSiteArea_CST": null,  
    "HW_installationSiteCity_CST": null,  
    "HW_installationSiteProvince_CST": null,  
    "HW_latitude_CST": null,  
    "HW_longitude_CST": null,  
    "cascaderAddress": [],  
    "createdBy": "10gd000000bcbTj7KT6e",  
    "createdBy.__objectType": "User",  
    "createdBy.name": "hw57...",  
    "createdDate": "2020-10-13 15:48:27",  
    "currencyIsoCode": "USD",  
    "id": "cQue000000e1qnhgtCng",  
    "installedPackage": null,  
    "lastModifiedBy": "10gd000000bcbTj7KT6e",  
    "lastModifiedBy.__objectType": "User",  
    "lastModifiedBy.name": "hw57...",  
    "lastModifiedDate": "2020-10-13 15:48:27",  
    "name": "百草园B栋2单元2号",  
    "owner": "10gd000000bcbTj7KT6e",  
    "owner.__objectType": "User",  
    "owner.name": "hw57..."  
  }  
}
```

步骤2 测试成功, 单击编辑器上方的 , 启用脚本。

----结束

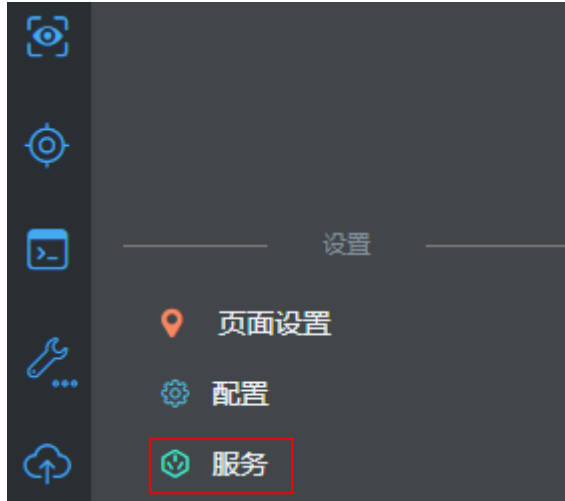
创建公共服务接口

前端页面支持通过页面创建的服务对象调用脚本, 但是为了避免各种权限之间的配置, 这里通过创建与脚本“HW_editEquipment”、“HW_queryEquipmentDetail”——对应的公共接口, 让页面直接调用这种公共接口。

操作步骤

步骤1 在应用中，单击下方“服务”，进入公共接口创建页面。

图 3-71 创建公共接口入口



步骤2 单击“新建”，进入新建公共接口页面。

图 3-72 公共接口创建

公共接口

使用公共接口，您可以将服务编排、脚本或对象的URL映射到外部网关，第三方可以通过OAuth2.0调用。



步骤3 创建“编辑设备”脚本“*HW_editEquipment*”的公共接口。

1. 设置接口参数信息，设置操作名称为“*editEquipment*”，版本为“1.0.0”，URL为“*/editEquipment*”，“类型”选择“脚本”，“资源”为“*HW_editEquipment*”，方法为“POST”，然后单击“保存”。

📖 说明

如果在“资源”下拉框中，未找到需要关联的脚本，请检查相关脚本是否已启用。

图 3-73 设置“编辑设备”脚本的公共接口参数

标签 * 操作名称

* 版本

* URL

内容类型 分类

描述

允许匿名访问

类型 服务编排 脚本 对象

自定义响应

* 资源 * 方法

2. 在应用开发页面，单击左下角的“服务”，进入公共接口页面，查看上一步中新建的自定义接口URL“/service/HW_MyApp/1.0.0/editEquipment”，后续开发页面时，会使用这个URL。

图 3-74 查看自定义接口 URL

公共接口

使用公共接口，可以将Flow的URL映射到发布到外部网关，第三方可以通过OAuth2.0调用。

操作名称	版本	URL	方法	类型	资源	最后修改人	最后修改时间	操作
editEquipment	1.0.0	/service/HW_MyApp/1.0.0/editEquipment	POST	脚本	HW_editEquipment		2020-09-08 14:12:15	

共 1 条

10条/页 < 1 > 前往 1 页

步骤4 参照上一步，创建“按ID查询设备详情”脚本“HW_queryEquipmentDetail”的公共接口，详细接口信息如图3-75所示。

操作名称为“queryEquipmentDetail”，版本为“1.0.0”，URL为“/queryEquipmentDetail”，“类型”选择“脚本”，“资源”为“HW_queryEquipmentDetail”，“方法”为“GET”。

图 3-75 “HW_queryEquipmentDetail” 的公共接口参数

The screenshot shows a configuration form for an API endpoint. The fields are as follows:

- 标签 (Tag): queryEquipmentDetail
- * 操作名称 (Operation Name): queryEquipmentDetail
- * 版本 (Version): 1.0.0
- * URL: /service/..._MyApp/1.0.0 /queryEquipmentDetail
- 内容类型 (Content Type): application/json
- 分类 (Category): 请输入 (Please enter)
- 描述 (Description): 请输入 (Please enter)
- 允许匿名访问 (Allow anonymous access):
- 类型 (Type): 服务编排, 脚本, 对象
- 自定义响应 (Custom response):
- * 资源 (Resource): ..._queryEquipmentDetail
- * 方法 (Method): GET
- Buttons: 保存 (Save), 取消 (Cancel)

----结束

3.1.6.3.3 组装“编辑设备”页面

组装页面包括拼装页面组件、定义组件的事件代码，并通过公共接口调用“编辑设备”脚本。

页面分析

如果要实现将前端页面上输入的电梯信息保存到数据库中，需要创建与前端组件绑定的自定义模型，以及与后端逻辑关联的服务模型，如图3-77所示。

图 3-76 编辑设备页面预览

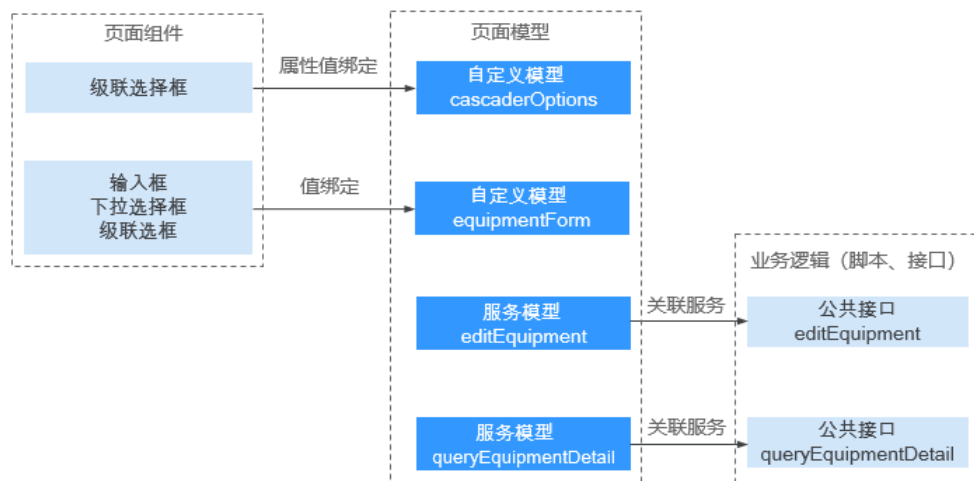
The preview shows a form titled "设备详情" (Device Details) with the following fields:

- 设备编码 (Device Code): Input field
- 设备名称 (Device Name): Input field
- 设备品牌 (Device Brand): Dropdown menu
- 设备型号 (Device Model): Input field
- 省市区 (Province/City/District): Cascading dropdown menu
- 详细地址 (Detailed Address): Input field
- Buttons: 保存 (Save), 取消 (Cancel)

页面上包含输入框（设备编码、设备名称、设备型号和地址）、选择框（设备品牌）和级联选择框（省市区）。其中，“省/市/区”级联框不仅要值绑定，还需要做属性绑定：

- 级联框的选项：以属性绑定的方式，与cascaderOptions关联。即cascaderOptions的内容，将成为级联框的可选项。
- 级联框的选择结果：以值绑定的方式，与equipmentForm关联。即级联框的选择结果，将保持到equipmentForm.cascaderAddress中。

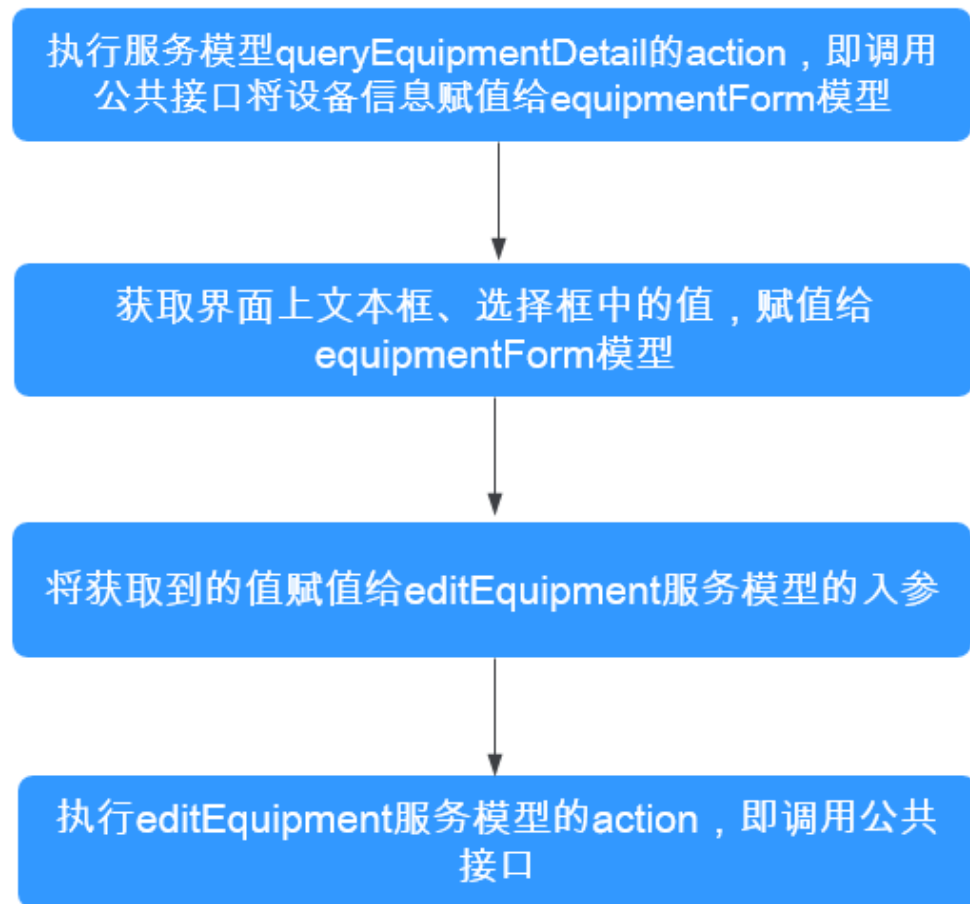
图 3-77 配置态的模型定义



同时，还需要为页面定义页面事件，为“保存”按钮定义“点击”事件，在事件的脚本代码中调用上述服务模型的API。

在应用预览下，从打开页面、在页面输入内容，到单击“保存”按钮，触发的执行逻辑如图3-78所示。

图 3-78 执行逻辑



定义模型

步骤1 进入“设备维修管理系统”应用。

步骤2 在“Equipment”中，将鼠标放在“Page”目录上，单击界面上出现的“+”，在弹出菜单中选择“标准页面”。

步骤3 在“标签”和“名称”文本框中输入“editEquipment”，单击“添加”。

平台实际创建的页面名称为“HW_editEquipment”，包含前缀“HW_”，对应首次创建应用时定义的命名空间。新建创建的页面，默认是当前用户锁定状态，可以进行编辑保存等操作。


当编辑已有标准页面时，为防止编辑时多人篡改，编辑前请单击进行锁定。

步骤4 定义与“省/市/区”级联框的可选项相关联的自定义模型。

- 在页面底部单击“模型视图”，进入模型视图页面，单击“新增模型”。
- 添加自定义模型，模型名称“cascaderOptions”，单击“下一步”，如图3-79所示。

图 3-79 定义级联框用到的自定义模型



3. 设置保持不变，单击“下一步”。
4. 方法保持不变，单击“确定”。
5. 单击页面上方的，保存设置。

步骤5 定义与页面上各个输入框、选择框相关联的自定义模型。

1. 在“模型视图”中，单击“新增模型”。
2. 添加自定义模型，模型名称“equipmentForm”，单击“下一步”，如图3-80所示。

图 3-80 定义页面组件需要关联的自定义模型




3. 单击“新增节点”，逐一添加与页面元素对应的参数（name、*HW*_equipmentSN_CST、*HW*_equipmentBrand_CST、*HW*_equipmentModel_CST、cascaderAddress、*HW*_installationDetailAddress_CST），单击“下一步”，如图3-81所示。

须知

为简化后续事件脚本，除cascaderAddress外，请确保其他5个参数的参数名与设备对象（*HW*_Equipment_CST）的字段名保持一致。注意这里的下划线是两个，要与表3-5里的字段保持一致，*HW*_需要修改为实际的命名空间前缀。

图 3-81 添加模型包含的参数



4. 方法保持不变，单击“确定”。
5. 单击页面上方的，保存设置。

步骤6 定义与API (editEquipment:1.0.0) 关联的服务模型。

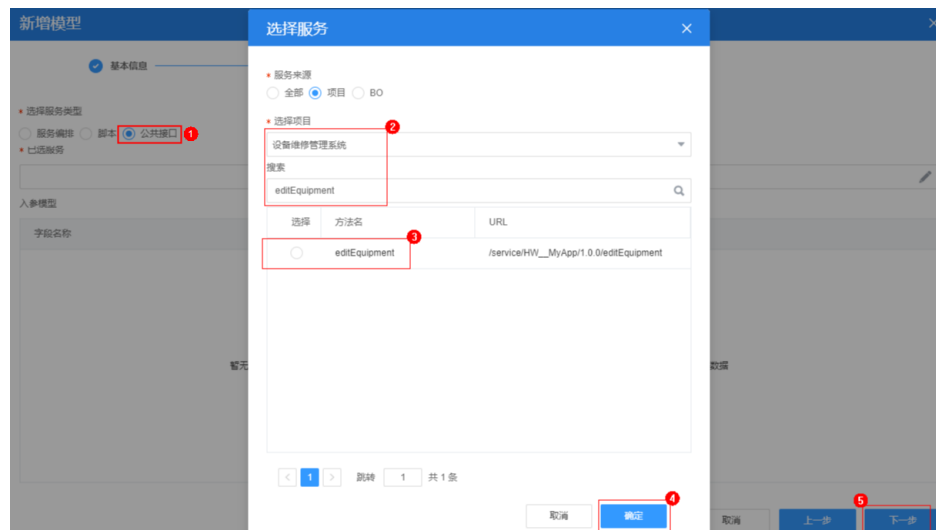
1. 在“模型视图”中，单击“新增模型”。
2. 添加服务模型，模型名称“editEquipment”，来源选择“服务”，单击“下一步”，如图3-82所示。

图 3-82 定义服务模型



3. 指定模型与API “editEquipment” 关联，单击“下一步”，如图3-83所示。关联API后，系统会自动显示API中脚本的输入、输出参数。


图 3-83 为模型关联 Script



4. 方法保持不变，单击“确定”。系统自动添加了执行的方法，如图3-84所示。未来，将在事件脚本中执行这个方法，即执行模型关联的API中的脚本。

图 3-84 为模型定义方法

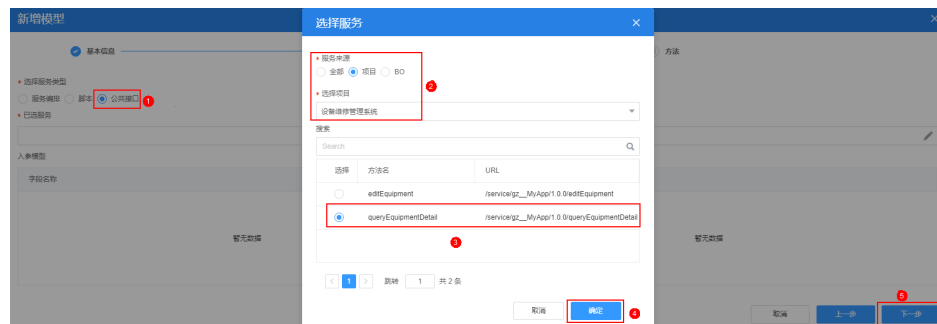


5. 单击页面上方的 , 保存设置。

步骤7 定义与API (queryEquipmentDetail) 关联的服务模型。

1. 在“模型视图”中，单击“新增模型”。
2. 添加服务模型，模型名称“queryEquipmentDetail”，“来源”选择“服务”，单击“下一步”。
3. 指定模型与API“queryEquipmentDetail”，单击“下一步”。

图 3-85 为模型关联 Script



4. 方法保持不变，单击“确定”。


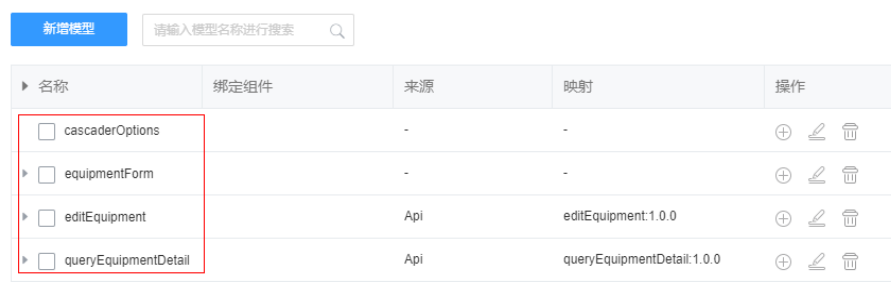
5. 单击页面上方的 , 保存设置。

图 3-86 新增的页面模型

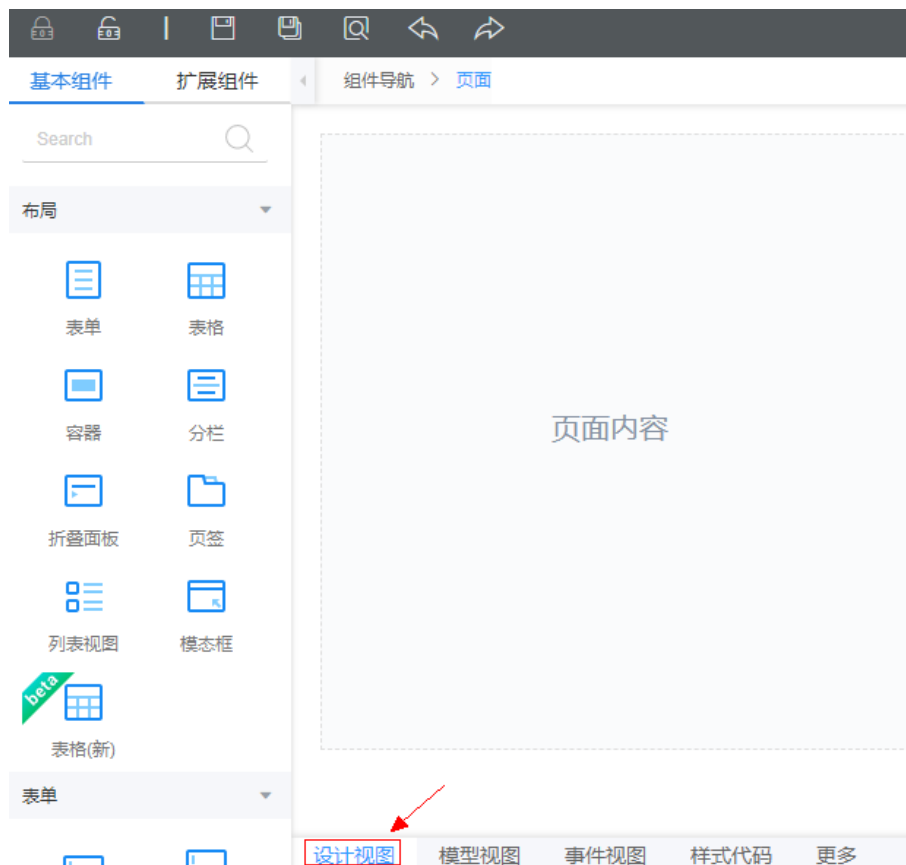


----结束

拖拽组件并关联模型

步骤1 单击“设计视图”，切换到页面设计视图。

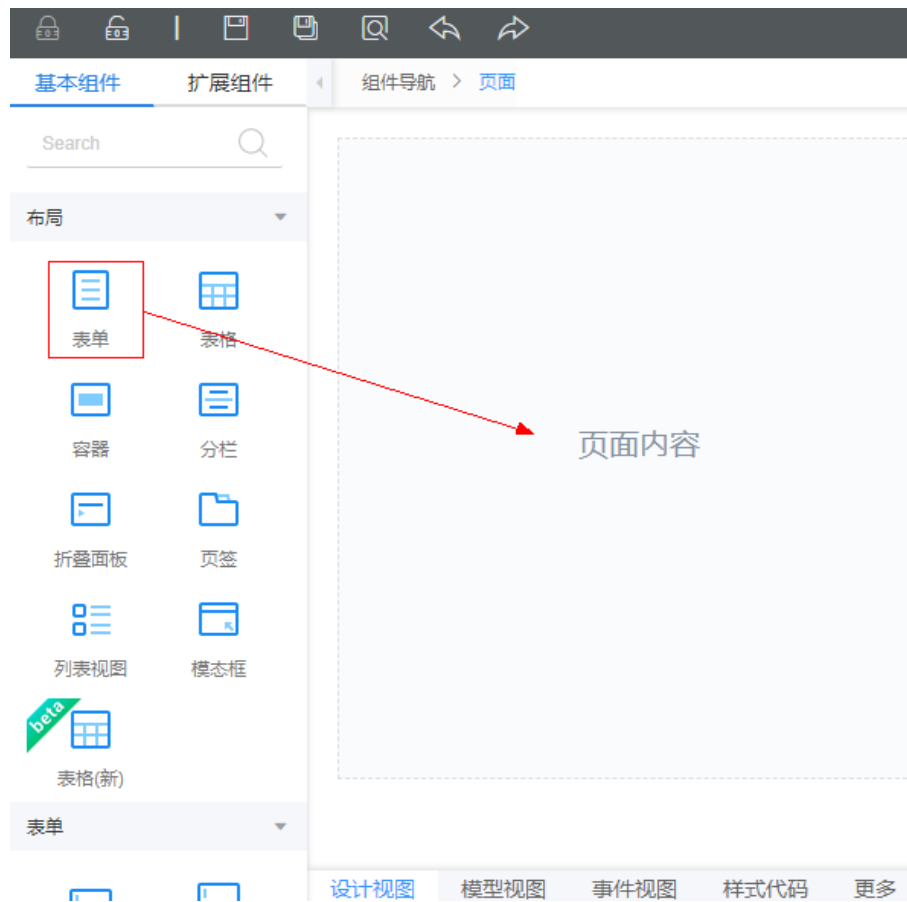
图 3-87 切换到设计视图组件列表



步骤2 将左侧组件区的“表单”拖拽到右侧“页面内容”中，在“元数据表单配置向导”弹窗底部，单击“取消”，创建一个空的表单控件。

当前不单独定义数据源，因此需要单击“取消”数据绑定。

图 3-88 拖拽表单到页面并取消数据绑定



步骤3 组装参数区域。


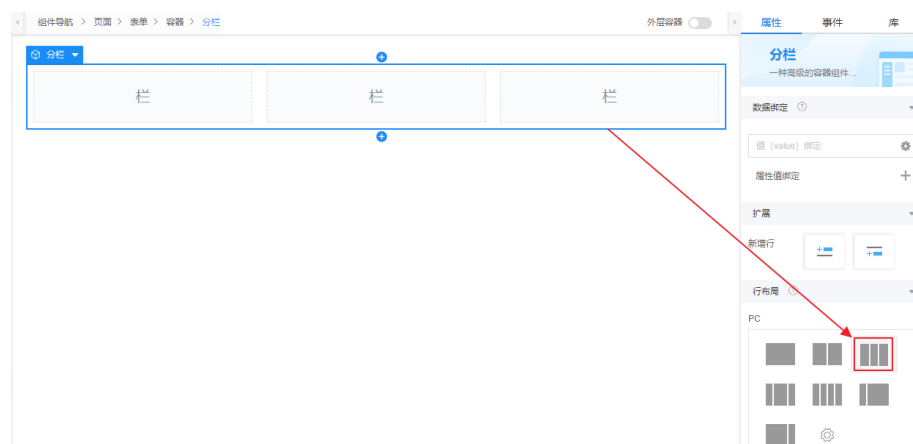
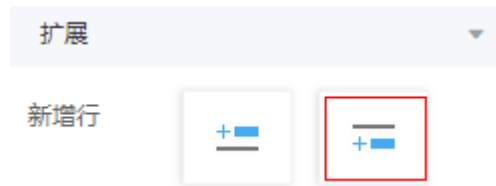
1. 从左侧组件列表中，拖一个“容器”到上一步创建的“表单”。
2. 从左侧组件列表中，拖一个“分栏”到上一步创建的“容器”中。
“分栏”默认有2个“栏”，即当前栅格中包含1行1列的区域。
3. 选中“分栏”，在右侧属性的“行布局”中，单击，将分栏设置为3栏。

图 3-89 设置分栏为 3 栏



4. 在右侧“属性”页签中，单击“新增行”后面的图标，新增一行，如图3-90所示。设置后，分栏组件被设置为2行（分栏）3列（栏）。

图 3-90 设置表格内的行列数



5. 从左侧组件区的拖一个“输入框”到分栏组件的第1行（分栏）第1栏，并在右侧“属性”页签中将“标签”修改为“设备编码”。

图 3-91 设备编码



6. 分别向第1行第2栏、第2行第1栏、第2行第3栏中拖一个“输入框”，并设置“标签”为“设备名称”、第2行第1栏“设备型号”、第2行第3栏“详细地址”。
7. 从左侧组件列表中，拖一个“下拉框”到分栏组件的第1行第3栏，并在右侧“属性”页签中将“标签”修改为“设备品牌”。
8. 从左侧组件列表中，拖一个“级联选择框”到分栏组件的第2行第2栏，并将“标签”修改为“省/市/区”。

步骤4 组装页面标题。

在左侧组件区拖拽一个“标题”组件到上一步创建的“容器”前面，并在右侧“属性”页签中将“标题内容”修改为“设备详情”，并设置“样式”的“高级设置”为“:root{text-align:center;font-size:20px;}”。

步骤5 组装按钮区域。

1. 在左侧组件区拖拽一个“容器”到3中创建的“容器”后（注意要在表单里面，两个容器在同级），并在右侧“属性”页签中将“水平对齐方式”修改为“中”，即居中对齐。

图 3-92 拖拽容器



图 3-93 设置居中对齐




2. 从左侧组件区拖拽一个“按钮”到刚创建的“容器”中，并在右侧“属性”页签中，将“显示名称”修改为“保存”，将“类型”修改为“主要按钮”。
3. 拖拽一个按钮到“保存”按钮右边，并设置为“取消”按钮。“取消”按钮的“类型”保持“默认按钮”不变。
4. 单击界面上方的，保存页面，可以在属性面板底部查看组件树。

图 3-94 组装完成后页面的组件树



步骤6 为页面组件关联模型。

1. 选中“设备名称”输入框。
2. 在右侧“属性”页签中单击⚙️，为“设备名称”输入框绑定“equipmentForm”自定义模型中的“name”参数，如图3-95所示。

数据绑定后，当在前台界面输入内容时，系统就会把输入框中的内容，赋值给“name”。

图 3-95 输入框数据绑定



3. 重复上一步，为“设备编码”、“设备型号”、“详细地址”文本输入框绑定“equipmentForm”自定义模型中的对应参数。
4. 选中“设备品牌”下拉选择框，在右侧“属性”页签中单击⚙️，为选择框绑定“equipmentForm”自定义模型中的“HW_equipmentBrand_CST”参数。
5. 选中“设备品牌”下拉选择框，在右侧“基本属性”中，单击“选项列表”的⚙️，在弹窗中输入“equipmentBrand”，在联想记录中，选择设备对象字段“HW_equipmentBrand_CST”，为选择框添加下拉选项关联的字段，如图 3-96 所示。

图 3-96 为下拉选择框定义可选项



图 3-97 输入字段搜索

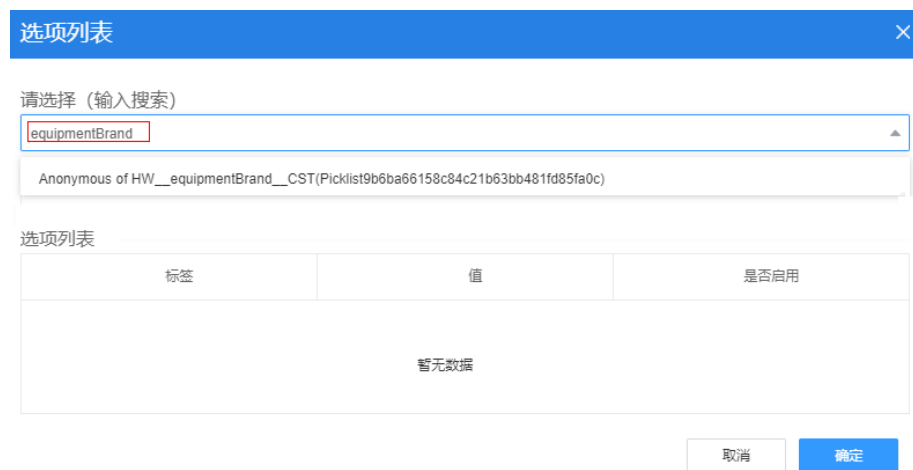


图 3-98 关联字段




- 选中“省/市/区”级联框，在右侧“属性”页签中单击⚙️，为级联框绑定“equipmentForm”自定义模型中的“cascaderAddress”参数。
- 单击属性值绑定后的“+”，将“属性”设置为“选择数据”，“模型字段”绑定到“cascaderOptions”，如图3-99所示。

数据绑定后，级联选择框的选项来自“cascaderOptions”；同时，当在前台界面选择省/市/区时，系统就会把选择结果赋值给“equipmentForm.cascaderAddress”。

图 3-99 级联框数据绑定



- 单击页面上方的，保存页面。

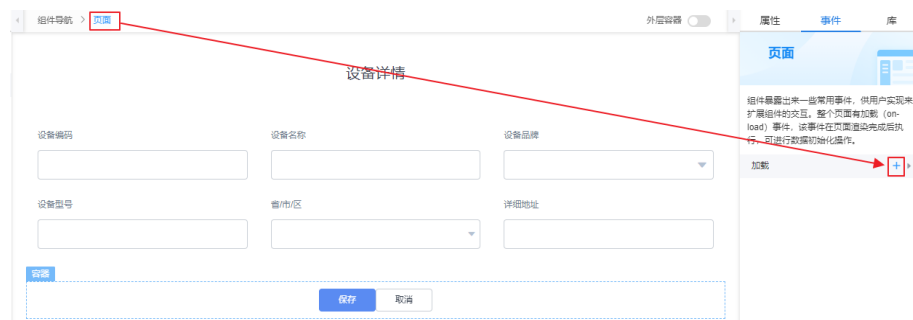
----结束

定义页面事件

步骤1 定义页面加载事件。

- 在“设计视图”中，选中最外层的“页面”，也可以直接单击组件导航，快速定位。
- 在右侧“事件”页签中，单击“加载”后的“+”，为页面添加事件代码。

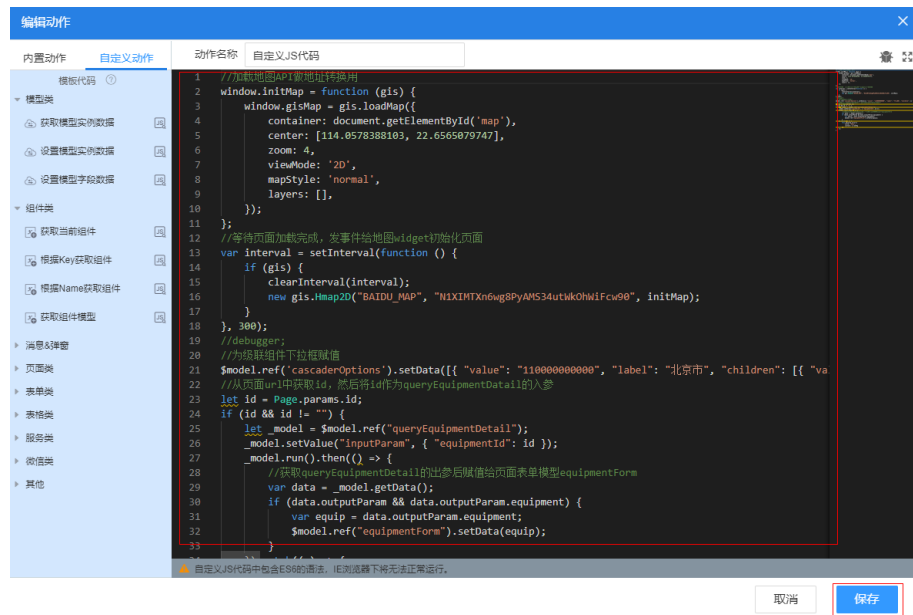
图 3-100 添加页面代码



- 在“添加动作”弹窗的“自定义动作”中，输入如下事件代码，然后单击底部“创建”。


```
//debugger;
//为级联组件下拉框赋值
$model.ref('cascaderOptions').setData([{"value":"110000000000","label":"北京市","children":
[{"value":"110100000000","label":"市辖区","children":[{"value":"110101000000","label":"东城区"},
{"value":"110102000000","label":"西城区"},{"value":"110105000000","label":"朝阳区"},
{"value":"110108000000","label":"海淀区"}]}]},{"value":"310000000000","label":"上海市","children":
[{"value":"310100000000","label":"市辖区","children":[{"value":"310101000000","label":"黄浦区"},
{"value":"310104000000","label":"徐汇区"},{"value":"310106000000","label":"静安区"},
{"value":"310115000000","label":"浦东新区"},{"value":"310110000000","label":"杨浦区"},
{"value":"310112000000","label":"闵行区"}]}]},{"value":"440000000000","label":"广东省","children":
[{"value":"440100000000","label":"广州市","children":[{"value":"440101000000","label":"市辖区"},
{"value":"440105000000","label":"海珠区"},{"value":"440106000000","label":"天河区"},
{"value":"440111000000","label":"白云区"},{"value":"440113000000","label":"番禺区"}]},
{"value":"440300000000","label":"深圳市","children":[{"value":"440301000000","label":"市辖区"},
{"value":"440303000000","label":"罗湖区"},{"value":"440304000000","label":"福田区"},
{"value":"440305000000","label":"南山区"},{"value":"440306000000","label":"宝安区"},
{"value":"440307000000","label":"龙岗区"},{"value":"440308000000","label":"盐田区"},
{"value":"440309000000","label":"龙华区"},{"value":"440310000000","label":"坪山区"}]}]}]);
//从页面url中获取id，然后将id作为queryEquipmentDatail的入参
let id =Page.params.id;
if(id && id !=''){
let _model = $model.ref("queryEquipmentDetail");
_model.setValue("inputParam",{equipmentId: id });
_model.run().then(()=>{
//获取queryEquipmentDetail的出参后赋值给页面表单模型equipmentForm
var data = _model.getData();
if(data.outputParam && data.outputParam.equipment){
var equip = data.outputParam.equipment;
$model.ref("equipmentForm").setData(equip);
}
}).catch((e)=>{
this.$dialog.error({
title:'错误',
content: e.resMsg
});
});
}
```

图 3-101 创建页面加载事件代码



```
}).catch((e) => {
  this.$dialog.error({
    title: '错误',
    content: e.resMsg
  })
});
}
else {
  console.log('data');
  //将级联框的内容转换为Script ( editEquipment ) 的省、市、区3个参数，并拼接完整地址
  let fullAddress = "";
  let cascaderOptions = $model.ref('cascaderOptions').getData();
  if (cascaderOptions.length < 1) {
    throw new Error("CascaderOptions data is wrong.");
  }
  cascaderOptions.forEach(function (province, idx) {
    if (province.value == inputs.cascaderAddress[0]) {
      //构造参数名称要与Object字段名保持一致
      inputs.HW__installationSiteProvince__CST = inputs.cascaderAddress[0];
      fullAddress += province.label;
      if (inputs.cascaderAddress.length > 1) {
        province.children.forEach(function (city, idxCity) {
          if (city.value == inputs.cascaderAddress[1]) {
            //构造参数名称要与Object字段名保持一致
            inputs.HW__installationSiteCity__CST = inputs.cascaderAddress[1];
            fullAddress += city.label;
            if (inputs.cascaderAddress.length > 2) {
              city.children.forEach(function (area, idxArea) {
                if (area.value == inputs.cascaderAddress[2]) {
                  //构造参数名称要与Object字段名保持一致
                  inputs.HW__installationSiteArea__CST = inputs.cascaderAddress[2];
                  fullAddress += area.label;
                  return false;
                }
              });
            }
          });
        }
        return false;
      }
    });
  });
  return false;
}
});
}
return false;
}
});
//界面上输入了详细地址时，拼接出fullAddress完整地址
if (inputs.HW__installationDetailAddress__CST) {
  fullAddress += inputs.HW__installationDetailAddress__CST;
}
if (fullAddress) {
  console.log('fullAddress');
  inputs.HW__fullAddress__CST = fullAddress;
  _model.setValue('inputParam', { "equipment": inputs });
  _model.run().then(() => {
    context.$page.load('/besBaas/page#/HW__equipmentManage');
  }).catch((e) => {
    this.$dialog.error({
      title: '错误',
      content: e.resMsg
    })
  });
}
else {
  _model.setValue('inputParam', { "equipment": inputs });
  _model.run().then(() => {
    //提交成功后返回设备列表页面
    context.$page.load('/besBaas/page#/HW__equipmentManage');
  }).catch((e) => {
    this.$dialog.error({
      title: '错误',
      content: e.resMsg
    })
  });
}
```


```
    })  
  });  
}  
}  
}).catch((error) => {  
});
```

4. 单击页面上方的, 保存页面, 关闭事件编排器, 返回到页面。

步骤3 定义“取消”按钮的“点击”(on-click)事件。

1. 在“设计视图”中, 选中“取消”按钮。
2. 在右侧“事件”页签中, 单击“点击”后的“+”。
3. 在“添加动作”弹窗的“自定义动作”中, 输入如下事件代码, 然后单击底部“创建”。

```
//返回设备列表页面  
context.$page.load('/besBaas/page#/HW_equipmentManage');
```

4. 单击页面上方的, 保存页面, 关闭事件编排器, 返回到页面。

----结束

验证

在页面预览中, 检查页面效果。

- 步骤1 单击界面上方的, 进入预览应用页面。

系统会弹出“HW_editEquipment”预览页面。

- 步骤2 检查“设备品牌”下拉框, “省市区”级联框的选项是否正确。

选项正确, 则说明自定义模型、页面组件与模型绑定关系是正确的。因为当前编辑完成后的保存跳转页面还没有创建, 所以此时“保存”、“取消”按钮, 还不能验证。

----结束

3.1.6.4 开发“管理设备”功能

3.1.6.4.1 背景及原理（服务编排）

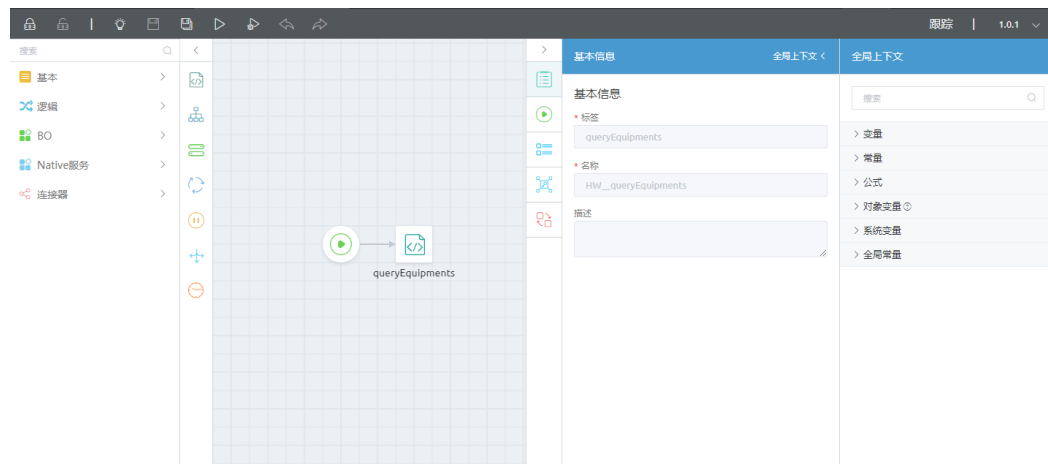
AstroZero的服务编排, 支持对逻辑判断组件、数据处理组件, 以及脚本、子服务编排、商业对象等进行可视化组合编排, 实现丰富的业务功能。

了解服务编排

在传统的开发中程序员一般是基于代码进行开发, 程序员需要学习内容较多, 开发效率相对低一些, 开发门槛也高。AstroZero的服务编排功能, 类似于编程中一段有流程、条件处理、判断逻辑的程序。这段程序有输入参数和输出参数、可以独立成为一个对外调用的方法。同时, 在程序内部, 也可以调用其他的方法。

AstroZero中的服务编排是将原来基于代码编程改变为用图形化, 拖拉拽的方式去编程。如图3-102所示, 服务编排界面是图形化、模板化的, 您甚至不需要任何编程经验, 将左侧面板区的组件拖拽到右侧画布、做必要的配置, 就可以完成服务编排的开发。

图 3-102 服务编排界面



服务编排界面中，可以编排如下组件：

- 基本组件：在服务编排引用脚本或者另一个服务编排，增/改/删/查记录，以及发送邮件、发送事件等。
- 逻辑组件：在服务编排中实现变量赋值Assignment、循环Loop、跳出循环Break、决策Decision和等待Wait。
- 商业对象：将封装好的BO能力作为服务编排中的一个节点。
- 连接器：将短信发送、支付等第3方连接器作为当前服务编排中的一个节点。

除了图形化编排，AstroZero也支持服务编排的在线测试验证，以及问题跟踪调试，方便您及时发现并解决问题。

服务编排测试通过、发布后，既可以直接被前端页面调用，也可以作为restful接口被第三方系统调用，也可以包装成公共接口后被调用。本节中主要是将服务编排包装成一个公共接口后，供页面调用，“管理设备”功能中涉及的业务逻辑，以及服务编排与脚本关系如表3-8下所示，详细操作方式及说明请参见[创建业务逻辑](#)。

表 3-8 “管理设备”功能需要创建的脚本、服务编排详情



脚本名称	主要作用	关联服务编排	关联公共接口
HW_queryEquipments	在输入查询条件后，查询设备	HW_queryEquipments	queryEquipments
HW_deleteEquipment	删除设备	不涉及	deleteEquipment
HW_equipmentSelectListQuery	查询所有设备并以选项列表的形式返回	不涉及	equipmentSelectListQuery

3.1.6.4.2 创建业务逻辑

创建“查询设备”脚本

管理设备页面中，实现输入设备的信息，查询出对应设备的全部信息功能，需要开发“查询设备”脚本和服务编排。

操作步骤

- 步骤1** 进入“设备维修管理系统”应用。
 - 步骤2** 在“Equipment”目录中，将鼠标放在“Script”上，单击界面上出现的，在弹出菜单中选择“脚本”。
 - 步骤3** 在弹窗中，选中“创建一个新脚本”，在“名称”文本框中输入“queryEquipments”，单击“添加”。
- 系统实际创建的脚本名称为“HW_queryEquipments”，“HW_”前缀由租户命名空间namespace决定。新建创建的脚本，默认是当前用户锁定状态，可以进行编辑保存等操作。
- 当编辑已有脚本时，为防止编辑时多人篡改，编辑前请单击进行锁定。
- 步骤4** 在代码编辑器中，插入如下脚本代码。


须知

脚本中**红色内容**请替换为实际的对象名、字段名。



```
/*
 * 本脚本用于按条件查询设备列表
 *
 */
import * as db from 'db'; // 导入处理object相关的标准库
import * as context from 'context'; // 导入上下文相关的标准库
import * as decimal from 'decimal'; // 导入decimal数据类型相关的标准库
// 定义入参结构
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'String' })
  name: string; // 设备名称，需要按设备名称查询时传入
  @action.param({ type: 'String' })
  fullAddress: string; // 完整地址，需要按完整地址查询时传入
  @action.param({ type: 'Number', min: 0 })
  start: decimal.Decimal; // 分页信息，表示从第几条数据开始查询
  @action.param({ type: 'Number', min: 0 })
  limit: decimal.Decimal; // 分页信息，表示一次查询几条数据
}
// 定义出参结构
@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'Any', isCollection: true, label: 'object' })
  equipments: object[]; // 设备列表
  @action.param({ type: 'String' })
  total: string; // 总共查到几条数据
}
@useObject(['HW_Equipment_CST']) // 使用数据库对象HW_Equipment_CST
@action.object({ type: "method" })
export class QueryEquipments {
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public queryEquipments(input: ActionInput): ActionOutput {
    let out = new ActionOutput(); // 新建出参ActionOutput类型的实例，作为返回值
    try {
      // 获取HW_Equipment_CST这个Object的操作实例
      let s = db.object('HW_Equipment_CST');
```


```
算等 //option是db标准库queryByCondition方法的入参（选项选项），用于对查询结果排序、分组、聚合计
//这里构造的是按创建时间逆序返回查询结果
let option = {
  "options": {
    "orderby": [
      {
        "field": "createdDate",
        "order": "desc"
      }
    ],
  },
};
//如果有分页
if (input.start && input.limit) {
  let start = decimal.toNumber(input.start);//将decimal类型转换为接口需要的number类型
  let limit = decimal.toNumber(input.limit);
  option.options['limit'] = limit;
  option.options['skip'] = start;
}
//condition是db标准库queryByCondition方法的入参（查询条件）
let condition = {
  "conjunction": "AND",
  "conditions": []
};
//基本查询条件
condition.conditions.push({
  "field": "id",
  "operator": "isnotnull",
});
condition.conditions.push({
  "field": "id",
  "operator": "ne",
  "value": ""
});
//按设备名称查询
if (input.name && input.name != "") {
  condition.conditions.push({
    "field": "name",//与对象中的字段名保持一致
    "operator": "contains",
    "value": input.name
  });
}
//按设备地址查询
if (input.fullAddress && input.fullAddress != "") {
  condition.conditions.push({
    "field": "HW__fullAddress__CST",
    "operator": "contains",
    "value": input.fullAddress
  });
}
//调用按条件查询HW__Equipment__CST的接口
out.equipments = s.queryByCondition(condition, option);
//构造实时监控页面需要的属性字段longitude__CST、latitude__CST、fullAddress__CST和
equipmentSn__CST
for (let equip of out.equipments || []) {
  equip['longitude__CST'] = equip['HW__longitude__CST'];//与对象中的字段名保持一致
  equip['latitude__CST'] = equip['HW__latitude__CST'];
  equip['fullAddress__CST'] = equip['HW__fullAddress__CST'];
  equip['equipmentSn__CST'] = equip['HW__equipmentSN__CST'];
}
//调用查询符合condition条件的数据总数的接口
out.total = s.count(condition) + "";
} catch (error) {
  console.error(error.name, error.message);
  context.setError(error.name, error.message);
}
return out;
```

```
}  
}
```

步骤5 单击编辑器上方的，保存脚本。

步骤6 测试脚本能否正常执行。

1. 单击编辑器上方的，执行脚本。
2. 在界面底部，直接单击测试窗口右上角，执行脚本。
执行成功后，会在“输出”页签返回全部设备信息。

步骤7 测试成功，单击编辑器上方的，启用脚本。


----结束

创建“查询设备”服务编排

“查询设备”脚本创建完成后，创建一个“服务编排”，并在服务编排中引用脚本，改造和配置服务编排的输入输出参数，使其更适用于前端页面调用。


操作步骤

步骤1 进入“设备维修管理系统”应用。

步骤2 在“Equipment”目录中，将鼠标放在“Flow”上，单击界面上出现的，在弹出菜单中选择“服务编排”。

步骤3 选中“创建一个新的服务编排”，在“标签”和“名称”文本框中输入“queryEquipments”，并设置类型为“Autolaunched Flow”，单击“添加”。

系统实际创建的脚本名称为“HW_queryEquipments”，“HW_”前缀由租户命名空间namespace决定。新建创建的服务编排，默认是当前用户锁定状态，可以进行编辑保存等操作。

当编辑已有服务编排时，为防止编辑时多人篡改，编辑前请单击进行锁定。

步骤4 定义服务编排用到的变量。


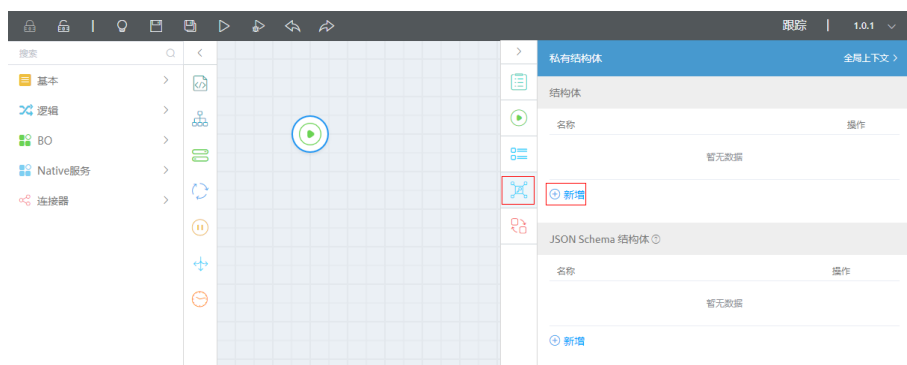
1. 单击页面右侧，再单击“新增”，在弹出窗口中输入结构体类型名称“Equipment”，单击“保存”。

图 3-103 创建私有结构体




- 单击 ，再单击“对象变量”后的“+”，在弹出窗口中定义私有结构体变量“equipments”，选中“是否为数组”（该变量将作为服务编排的输出参数，且因为查询结果可能是多条记录，所以要选）。单击“保存”，如图3-104所示。

图 3-104 新增私有结构体变量



对象变量

记录 全局结构体 私有结构体 事件

* 名称

* 私有结构体

默认值

描述

是否为数组

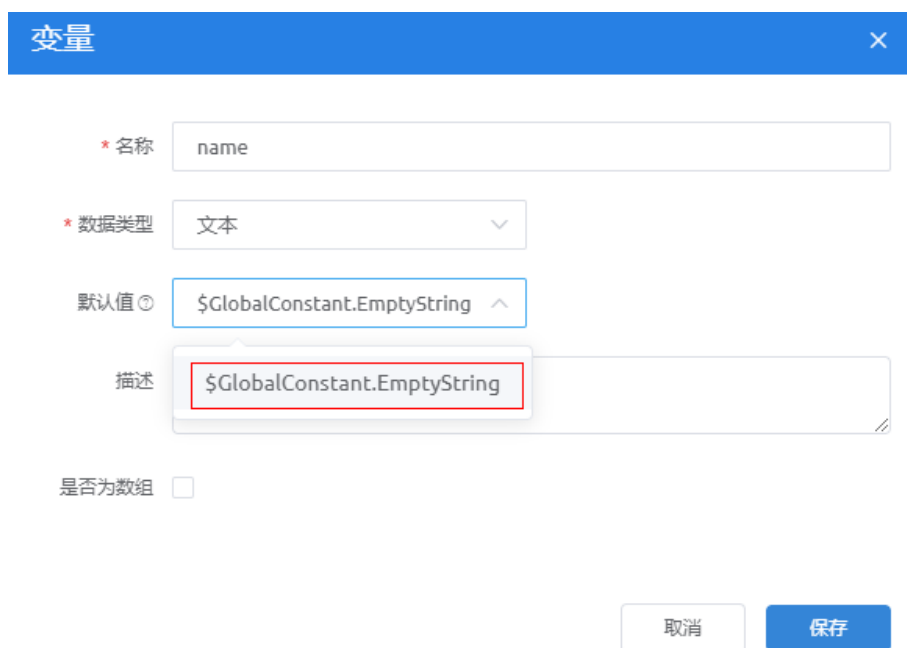
取消 保存

- 单击“变量”后的“+”，设置参数名称为“name”。单击变量后的“...”，选择“设置”，修改变量的名称、类型，选择默认值“{!\$GlobalConstant.EmptyString}”信息，如图3-106所示。

图 3-105 新增变量



图 3-106 设置变量



4. 重复上一步，定义表3-9中的其他变量。

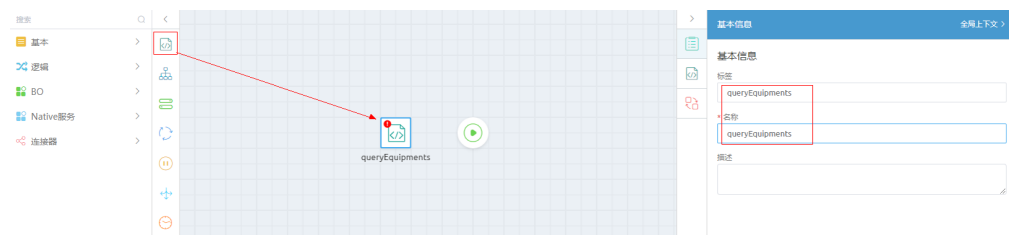
表 3-9 服务编排变量说明

变量唯一标识	描述	数据类型	默认值
name (上一步已定义)	设备名称	文本	{!\$GlobalConstant.EmptyString}
fullAddress	完整地址	文本	{!\$GlobalConstant.EmptyString}
limit	分页查询的条目数	数字	{!\$GlobalConstant.Null}
start	分页查询开始条目	数字	{!\$GlobalConstant.Null}
total	查询到的总条目数	文本	不设置

步骤5 在服务编排中添加并配置脚本图元。

1. 从左侧拖拽脚本图元到画布中。
2. 修改图元的基本信息，“标签”和“名称”都设置为“queryEquipments”。

图 3-107 设置脚本图元基本信息



3. 单击脚本图元，指定图元对应的脚本名称（HW_queryEquipments），并配置脚本的输入输出参数，如果下拉选项中没有目标脚本，请检查脚本是否已启用。
单击“全局上下文”，显示变量列表。在“输入参数”中，单击“新增行”，在下拉框中选择输入参数、依次拖拽变量到输入参数的“源”下。
在“输出参数”中，单击“新增行”，在下拉框中选择输出参数，并从变量和对象变量中拖拽“目标”，如图3-108所示。

说明

请直接从全局上下文拖拽变量到输入输出参数下的对应位置，请勿手动输入，手动输入的值系统可能不识别。

图 3-108 拖拽脚本的输入输出参数



步骤6 定义服务编排的输入、输出参数，并保存服务编排。




1. 在画布上，把鼠标放在起点图元图元上，从“+”拖动鼠标，在起点图元和脚本图元间增加连线；即将当前脚本设置为服务编排的起始节点。
2. 鼠标在画布空白处点一下，单击右侧，设置服务编排的输入输出参数，如图 3-109 所示。

图 3-109 拖拽服务编排的输入输出参数



3. 单击服务编排界面上方的，保存服务编排。
系统会弹出窗口，显示编译结果。

步骤7 测试服务编排能否正常执行。


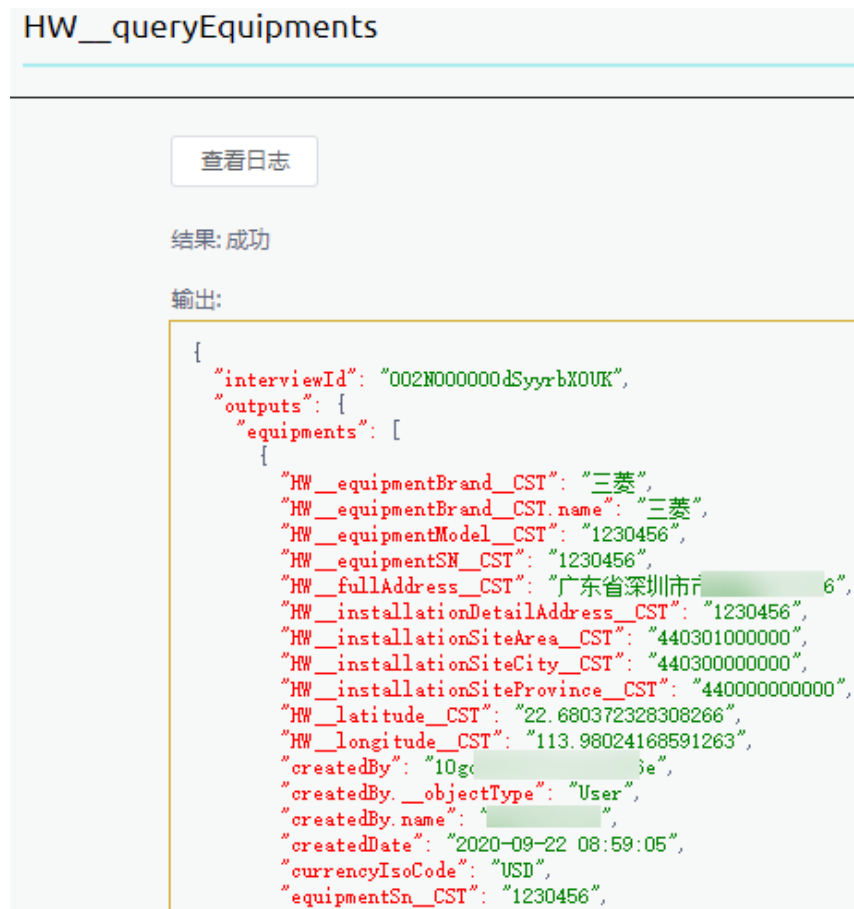
1. 单击服务编排编辑器上方的，执行服务编排。
2. 在“输入参数”中，输入测试数据，单击“运行”。
执行成功，界面上会返回设备对象中的全部信息，样例如下：

图 3-110 返回样例



步骤8 (可选) 在服务编排编辑器单击“跟踪”，可以查看到上一步的执行日志，方便定位错误。

步骤9 测试成功，单击服务编排编辑器上方的，发布服务编排。

---结束

创建“删除设备”脚本


当用户操作“删除”图标删除某条设备记录时，需要根据Id在设备对象中删除设备，因此需要创建一个根据Id“删除设备”的脚本。

操作步骤

步骤1 进入“设备维修管理系统”应用。

步骤2 在“Equipment”目录中，将鼠标放在“Script”上，单击界面上出现的“+”，在弹出菜单中选择“脚本”。

步骤3 选中“创建一个新脚本”，在“名称”文本框中输入“deleteEquipment”，单击“添加”。


当编辑已有脚本时，为防止编辑时多人篡改，编辑前请单击进行锁定。

步骤4 在代码编辑器中插入如下脚本代码。

须知

脚本中**红色内容**请替换为实际的对象名、字段名。


```
//本脚本用于删除设备
import * as db from 'db';//导入处理object相关的标准库
import * as context from 'context';//导入上下文相关的标准库
//定义入参结构，入参包含1个参数：Equipment对象，为必填字段
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'String', required: true, label: 'String' })
  id: string;
}
//定义出参结构，出参包含1个参数，Equipment的记录id
@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'String' })
  id: string;
}
//使用数据对象HW__Equipment__CST
@useObject(['HW__Equipment__CST'])
@action.object({ type: "method" })
export class DeleteEquipment { //定义接口类，接口的入参为ActionInput，出参为ActionOutput
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public deleteEquipment(input: ActionInput): ActionOutput {
    let out = new ActionOutput(); //新建出参ActionOutput类型的实例，作为返回值
    let error = new Error(); //新建错误类型的实例，用于在发生错误时保存错误信息
    try {
      let id = input.id;
      let s = db.object('HW__Equipment__CST'); //获取HW__Equipment__CST这个Object的操作实例
      //查询条件
      let condition = {
        "conjunction": "AND",
        "conditions": [{
          "field": "id",
          "operator": "eq",
          "value": id
        }]
      };
      let isDeleted = s.deleteByCondition(condition);
      if (isDeleted) {
        out.id = id;
      } else {
        error.name = "EQERROR";
        error.message = "删除设备失败! ";
        throw error;
      }
    } catch (error) {
      console.error(error.name, error.message);
      error.Error(error.name, error.message);
    }
    return out;
  }
}
```

步骤5 单击脚本编辑器上方的，保存脚本。

----**结束**

验证

步骤1 单击脚本编辑器上方的，执行脚本。


步骤2 在界面底部单击测试窗口右上角执行图标，进行测试。

如果设置输入参数（id来自的[编辑设备脚本](#)测试结果）：

```
{
  "id": "cQue000000e1qnhgtCng"
}
```

则脚本返回设备的详细信息，样例如下。

```
{
  "id": "cQue000000e1qnhgtCng"
}
```

步骤3 测试成功，单击编辑器上方启用图标，发布脚本。

----结束

创建“查询设备列表”脚本

查询设备列表脚本是为了实现将查询的所有设备并以选项列表的形式返回，并通过页面调用将查询结果展示到页面上，此脚本将会在[开发“生成工单”功能](#)章节被调用。

操作步骤

- 步骤1** 进入“设备维修管理系统”应用。
- 步骤2** 在“Equipment”目录中，将鼠标放在“Script”上，单击界面上出现的“+”，在弹出菜单中选择“脚本”。
- 步骤3** 选中“创建一个新脚本”，在“名称”文本框中输入“equipmentSelectListQuery”，单击“添加”。
- 步骤4** 在代码编辑器中插入如下脚本代码。


须知

脚本中**红色内容**请替换为实际的对象名、字段名。

```
import * as context from 'context';
import * as decimal from 'decimal';
import * as db from 'db';
//使用数据库对象HW_Equipment_CST
@useObject(['HW_Equipment_CST'])
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'String' })
  id: string;
}
@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'Any', label: 'object', isCollection: true })
  equipList: object[];
  @action.param({ type: 'Any', label: 'object' })
  equipment: object;
}
@action.object({ type: "method" })
export class EquipmentSelectListQuery {
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public equipmentSelectListQuery(input: ActionInput): ActionOutput {
    let out = new ActionOutput();
    let error = new Error();
    try {
      let objAct = db.object('HW_Equipment_CST');
```





```
if (input.id) {
  out.equipment = objAct.query(input.id);
  return out;
}
let option = {
  "options": {
    "orderby": [
      {
        "field": "lastModifiedDate",
        "order": "desc"
      }
    ],
    "skip": 0,
    "limit": 1000
  }
};
let equipList = objAct.queryByCondition(null, option);
let selectValue = equipList.map(function (v, i, a) {
  return {
    'value': v['id'],
    'display': v['name']
  }
});
out.equipList = selectValue;
} catch (error) {
  console.error(error.name, error.message);
  context.setError(error.name, error.message);
}
return out;
}
```

步骤5 单击脚本编辑器上方的, 保存脚本。

----结束

验证

步骤1 单击脚本编辑器上方的, 执行脚本。

步骤2 在界面底部单击测试窗口右上角, 进行测试。

如果不设置输入参数, 会出现类似如下样例的测试结果 (设备名称和设备id)。


```
{
  "equipList": [
    {
      "display": "百草园B栋2单元2号",
      "value": "cQue000000e1qnhgtCng"
    }
  ]
}
```

如果设置如下输入参数。

```
{
  "id": "cQue000000e1qnhgtCng"
}
```

则脚本返回设备的详细信息, 样例如下。

```
问题 输入参数 输出参数 日志
1  [
2  "equipment": {
3    "HW__equipmentBrand__CST": "迅达",
4    "HW__equipmentBrand__CST.name": "迅达",
5    "HW__equipmentModel__CST": "hw000123456",
6    "HW__equipmentSN__CST": "hw000123456",
7    "HW__fullAddress__CST": "上海市市辖区黄浦区万达广场1号电梯",
8    "HW__installationDetailAddress__CST": "万达广场1号电梯",
9    "HW__installationSiteArea__CST": "310101000000",
10   "HW__installationSiteCity__CST": "310100000000",
11   "HW__installationSiteProvince__CST": "310000000000",
12   "HW__latitude__CST": "39.90991427783537",
13   "HW__longitude__CST": "116.46606599975814",
14   "cascaderAddress": [
15     "310000000000",
16     "310100000000",
17     "310101000000"
18   ],
19   "createdBy": "10gd000000bcbTj7KT6e",
20   "createdBy.__objectType": "User",
21   "createdBy.name": "*****3",
22   "createdDate": "2020-09-17 11:09:09",
23   "currencyIsoCode": "USD",
24   "id": "cQue000000dKNL9kTBts",
25   "installedPackage": null
```

步骤3 测试成功，单击编辑器上方的，发布脚本。

----结束

创建公共接口

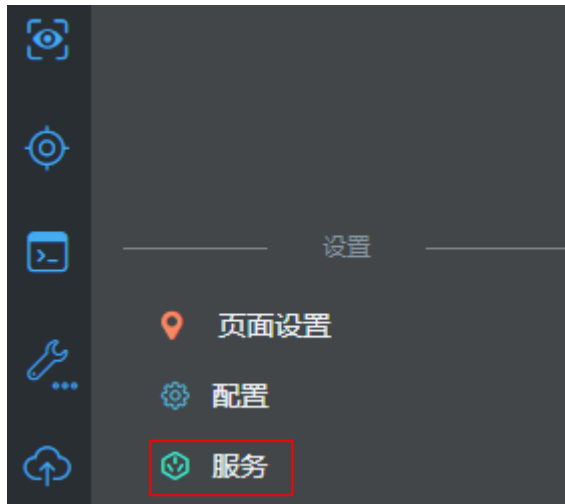
创建与脚本“*HW_deleteEquipment*”、“*HW_equipmentSelectListQuery*”以及服务编排“*HW_queryEquipments*”——对应的公共接口，让页面直接调用这种公共接口。

由于脚本“*HW_queryEquipments*”已被同名的服务编排给封装引用，因此，只需要为服务编排“*HW_queryEquipments*”创建一个关联的公共接口即可，脚本则不需要重复创建。

操作步骤

步骤1 在经典版应用开发页面，单击页面下方的“服务”，进入公共接口创建页面。

图 3-111 创建公共接口入口



步骤2 单击“新建”，进入公共接口创建页面。

图 3-112 公共接口创建

公共接口

使用公共接口，您可以将服务编排、脚本或对象的URL映射到外部网关，第三方可以通过OAuth2.0调用。



步骤3 创建“删除设备”脚本对应的“HW_deleteEquipment”的公共接口。

设置接口参数信息，设置操作名称为“deleteEquipment”，版本为“1.0.0”，URL为“/deleteEquipment”，“类型”选择“脚本”，“资源”为“HW_deleteEquipment”，方法为“DELETE”，然后单击“保存”。

说明

如果在“资源”下拉框中，未找到需要关联的脚本或服务编排，请检查相关脚本和服务编排是否已启用。

步骤4 参照上一步，创建表3-10中其他的公共接口，详细接口信息如表3-10所示。

说明

“HW_”请修改为实际命名空间前缀。

表 3-10 公共接口

设置操作	版本	URL	方法	类型	资源
deleteEquipment (上一步已创建)	1.0.0	/deleteEquipment	DELETE	脚本	HW_deleteEquipment

设置操作	版本	URL	方法	类型	资源
equipmentSelectListQuery	1.0.0	/equipmentSelectListQuery	GET	脚本	HW_equipmentSelectListQuery
queryEquipments	1.0.0	/queryEquipments	POST	服务编排	HW_queryEquipments

----结束

3.1.6.4.3 组装“设备管理”页面

通过组装设备管理的标准页面，实现对设备对象的增、删、改、查处理。

页面分析

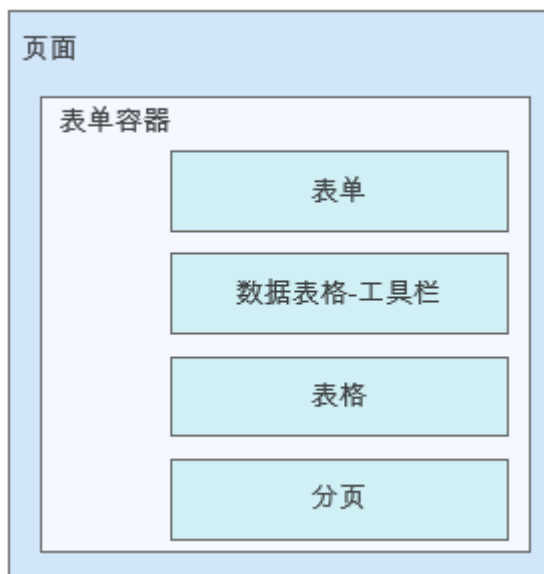
管理设备页面需要实现以下功能，如图3-113所示：

- 工具栏保留“新增设备”按钮，单击“新增设备”按钮，跳转到设备详情页面。
- 查询结果增加操作列，包含编辑和删除图标。

图 3-113 电梯信息管理界面

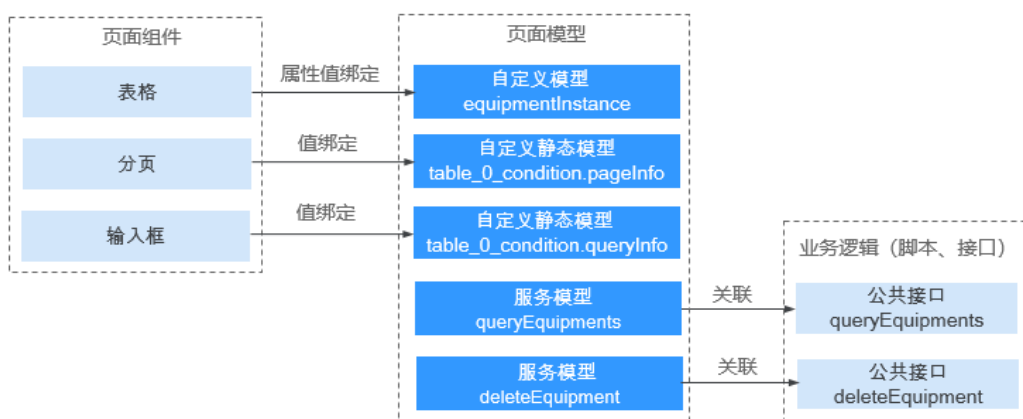
#	设备编码	设备名称	品牌	型号	地址	经度	纬度	操作
1	3217890002	睿梯	日立	3217890002	广东省深圳市福田区上步...	114.09068270407897	22.541899907145066	[编辑] [删除]
2	3217890001	百慕图B152单元2号	三菱	3217890001	广东省深圳市罗湖区地王...	114.10497208295907	22.54521930400758	[编辑] [删除]

图 3-114 前端页面组件位置关系



如果想要实现将前端页面上输入的设备信息保存到数据库中，需要将页面组件、模型、与数据对象进行关联，其关联关系如图3-115所示。

图 3-115 组件、模型、与对象的关系分析



开发页面的大致流程

1. 拖拽“表格”组件到页面上。
拖拽后，“表格”组件内默认包含“表格”和“分页”2个下层组件。
2. 创建自定义模型“equipmentInstance”，以及自节点，后续将会通过脚本（修改、查询、删除）操作对象“HW_Equipment_CST”。
3. 为“表格”绑定自定义模型“equipmentInstance”。
4. 为“表格”增加工具栏。
配置过程中，平台会自动创建自定义模型“table_0_condition”，在自定义模型“table_0_condition”下创建查询子节点“queryinfo”，后续将会与查询条件输入框、分页组件绑定。

执行流程


完成上述配置后，一个简易的增删改查电梯设备信息的页面即开发完成，页面的执行流程如表3-11所示。

表 3-11 执行流程

场景	系统执行流程
查询	<ol style="list-style-type: none"> 1. 界面查询条件传入自定义模型“table_0_condition”。 2. 将自定义模型“table_0_condition”的数据作为入参，运行服务模型“queryEquipments”（绑定“queryEquipmentes”接口），执行接口绑定的“HW_queryEquipments”脚本，从“HW_Equipment_CST”对象中获取设备信息列表。 3. 将服务模型“queryEquipments”返回的参数值绑定在自定义模型“equipmentInstance”并渲染“表格”。
新增	<ol style="list-style-type: none"> 1. “表格”中填写新设备记录后，即将填写内容传入表格绑定的自定义模型“equipmentForm”。 2. 将自定义模型“equipmentForm”的数据作为入参，运行服务模型“editEquipment”（绑定“editEquipments”接口），执行接口绑定的“HW_editEquipment”脚本，将新增记录插入到对象“HW_Equipment_CST”中。
修改	<ol style="list-style-type: none"> 1. “表格”中修改已有设备记录后，将修改内容传入表格绑定的自定义模型“equipmentForm”。 2. 将自定义模型“equipmentForm”的数据作为入参，运行服务模型“editEquipment”（绑定“editEquipments”接口），执行接口绑定的“HW_editEquipment”脚本，将修改后的电梯记录更新到对象“HW_Equipment_CST”中。
删除	<ol style="list-style-type: none"> 1. 通过自定义JS代码获取“表格”中，单击“删除”按钮的设备Id。 2. 将设备Id作为入参，运行服务模型“deleteEquipment”（绑定“deleteEquipment”接口），执行接口绑定的“HW_deleteEquipment”脚本，删除对象“HW_Equipment_CST”中的对应记录。

操作步骤

步骤1 进入“设备维修管理系统”应用。

步骤2 在“Equipment”目录中，将鼠标放在“Page”上，单击界面上出现的，在弹出菜单中选择“标准页面”。

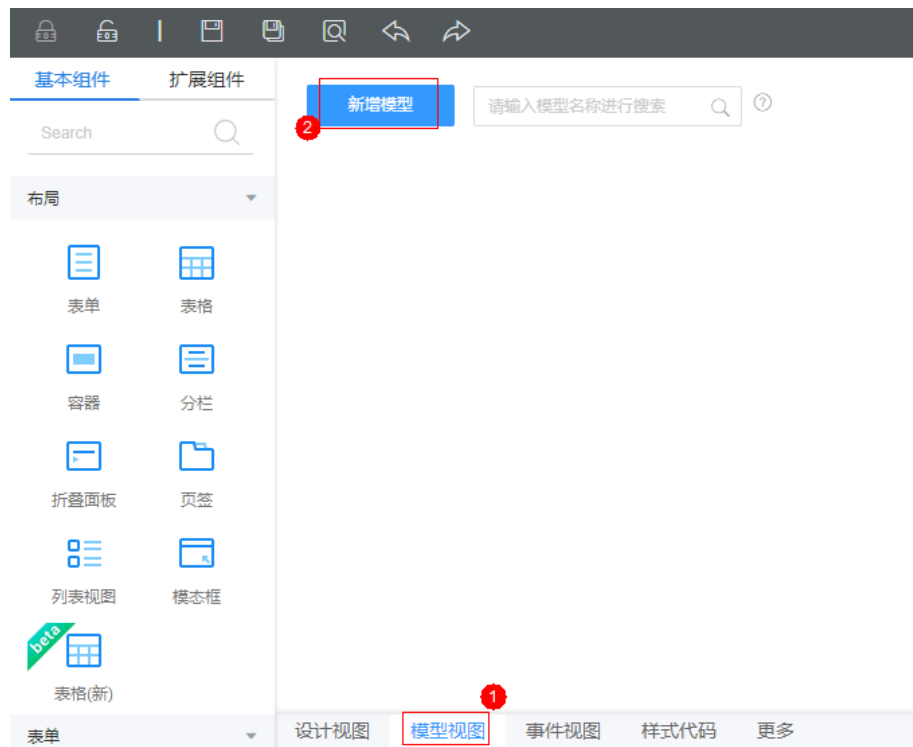
步骤3 在“标签”和“名称”文本框中输入“equipmentManage”，单击“添加”。

当编辑已有标准页面时，为防止编辑时多人篡改，编辑前请单击进行锁定。

步骤4 定义自定义模型“equipmentInstance”。

1. 在“模型视图”中，单击“新增模型”。

图 3-116 新增模型




2. 添加自定义模型，模型名称“equipmentInstance”，单击“下一步”。
3. 单击“新增节点”，创建如表3-12所示的节点。字段名称要与设备对象为“HW_Equipment_CST”一致，然后单击“下一步”，再单击“确定”。

须知

加粗斜体请替换为实际的对象名、字段名。

表 3-12 新增节点

节点名称	数据类型
name	保持默认 (Text)
<i>HW_equipmentSN_CST</i>	保持默认 (Text)
<i>HW_equipmentBrand_CST</i>	保持默认 (Text)
<i>HW_equipmentModel_CST</i>	保持默认 (Text)
<i>HW_fullAddress_CST</i>	保持默认 (Text)
<i>HW_longitude_CST</i>	保持默认 (Text)
<i>HW_latitude_CST</i>	保持默认 (Text)

4. 单击页面上方的，保存设置。

步骤5 定义服务模型“queryEquipments”，与API (queryEquipments:1.0.0) 关联。


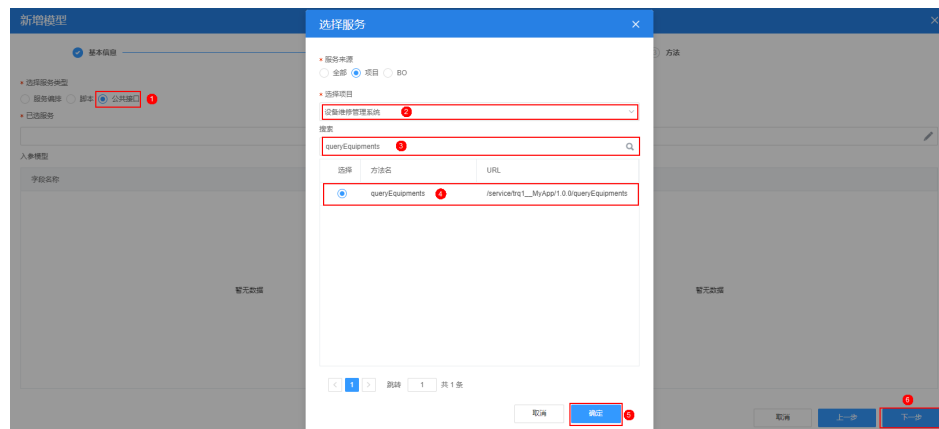

1. 在“模型视图”中，单击“新增模型”。
2. 添加服务模型，模型名称“queryEquipments”，单击“下一步”。
3. 选择服务类型为“公共接口”，并“选择项目”为“设备维修管理系统”，指定模型与API“queryEquipments”关联，单击“下一步”。
关联API后，系统会自动显示API中脚本的输入、输出参数。
4. 方法保持不变，单击“确定”。
5. 单击页面上方的，保存设置。

图 3-117 定义服务模型



步骤6 定义服务模型“deleteEquipment”，与API (deleteEquipment:1.0.0) 关联。

1. 在“模型视图”中，单击“新增模型”。
2. 添加服务模型，模型名称“deleteEquipment”，单击“下一步”。
3. 选择服务类型为“公共接口”，并“选择项目”为“设备维修管理系统”，指定模型与API“deleteEquipment”关联，单击“下一步”。
关联API后，系统会自动显示API中脚本的输入、输出参数。
4. 方法保持不变，单击“确定”。
5. 单击页面上方的，保存设置。

步骤7 切换到“设计视图”，从左侧基础组件区，拖拽一个“表格”到右侧“设计视图”中。

图 3-118 拖拽一个表格到页面



步骤8 为表格绑定模型，并设置表格查询结果区域。

1. 在“设计视图”中，选中“表格”，单击右侧“属性”页签，如图3-119所示。

图 3-119 为表格绑定对象模型



2. 在“选择模型”对话框中，选中“equipmentInstance”对象模型，单击“确定”。
绑定对象模型后，系统自动将模型的所有字段添加为表格列，如图3-120所示。

图 3-120 绑定数据模型后的表格列



3. 选中上图中“已添加列”中的任意一项，上下拖动调整列的显示顺序。
4. 单击上图中每个字段后的⚙️，修改列标题等字段属性。
以“设备编码”为例，其属性配置如图3-121所示。

图 3-121 设置“设备编码”列的属性

属性配置
✕

▼ 基本属性

字段名

列标题 🌐

列标题提示 🌐

列标题自定义渲染

▼ 功能

固定 ▼

隐藏

溢出省略

排序 ▼


筛选

单元格可编辑

显示编辑hover

表 3-13 修改各字段的显示属性

调整后的顺序	字段名	列标题	可编辑
1	HW__equipmentSN__CST	设备编码	是
2	name	设备名称	是
3	HW__equipmentBrand__CST	品牌	是
4	HW__equipmentModel__CST	型号	是
5	HW__fullAddress__CST	地址	是
6	HW__longitude__CST	经度	是
7	HW__latitude__CST	纬度	是

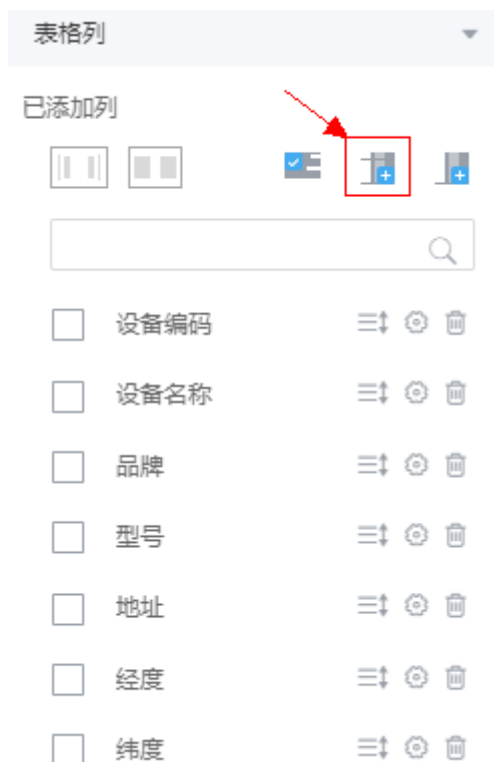
5. 单击页面上方的，保存设置。

步骤9 添加表格操作列，为表格添加编辑和删除两个操作按钮。

其中“编辑”是通过在按钮上添加自定义代码，将页面跳转到“编辑设备”页面。
“删除”则是通过自定义代码调用“删除设备”脚本实现删除功能。

1. 在表格的“表格列”下，单击，添加表格操作列“Operation1”。

图 3-122 添加操作列




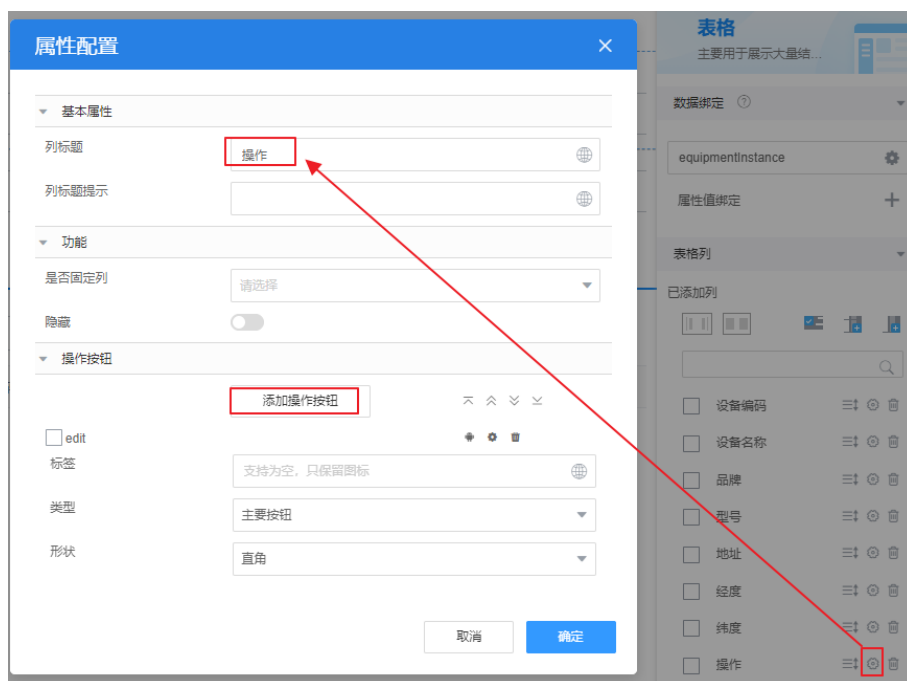

2. 单击新增的“Operation1”的，进入操作列属性配置。

图 3-123 设置操作列属性



3. 配置操作列属性。
 - a. 将“列标题”设置为“操作”。
 - b. 单击“添加操作按钮”2次，增加2个按钮。
 - c. 设置第一个按钮“类型”为“主要按钮”，“图标”为“edit”，并单击, 选择“动作列表”后的“+”，进入事件编辑，添加以下自定义代码。


须知

脚本中红色内容请替换为实际的对象名、字段名、页面名、组件ID。

```
//获取当前组件（即table）
let _component = context.$component.current;
//获取当前行row，取对象的id属性
let rowId = _component.$attrs.row.id;
//带着id跳转到设备详情页面
context.$page.load('/besBaas/page#/HW_editEquipment?id='+rowId);
```

图 3-124 设置操作列属性。



- d. 向下拖动滚动条，设置第二个按钮“类型”为“主要按钮”，“图标”为“delete”，并单击, 选择“动作列表”后的“+”，进入事件编辑，添加以下自定义代码。

须知

脚本中红色内容请替换为实际的对象名、字段名、页面名、组件ID。其中“table_0_condition”为当前表格组件的ID号，中间数字默认为“0”，如果有多次修改或创建该ID号会变化，选中表格后，在组件树上可以查看实际组件ID。

```
//获取当前组件（即table）
let _component = context.$component.current;
//获取当前行
let row = _component.$attrs.row;
// 表格组件
let _table =this.$component.table;
// 删除当前行数据
this.$dialog.confirm({
  title:'确认框',
  content:'确认是否删除? ',
  okText:"确定",
  cancelText:"取消",
  onOk:()=>{
    $model.ref('deleteEquipment').setData({inputParam:{'id': row.id}});
    $model.ref('deleteEquipment').run().then(function(data){
      let pageInfo = $model.ref('table_0_condition').getData().pageInfo;
```

```
let queryData = {
  "start": (pageInfo.curPage - 1) * pageInfo.pageSize,
  "limit": pageInfo.pageSize
};
$model.ref('queryEquipments').setData({inputParam: queryData});
$model.ref('queryEquipments').run().then(function(data) {
  $model.ref('equipmentInstance').setData(data.equipments);
  pageInfo.total = parseInt(data.total);
  $model.ref('table_0_condition').setData({"pageInfo": pageInfo});
}).catch(function(error) {
  console.log('error is', error);
});
});
}
```

步骤10 设置工具栏区域。

1. 在左侧“设计视图”中，选中“表格”，单击右侧“属性 > 表格区块”中“工具栏”后的“添加”按钮。

图 3-125 增加工具栏

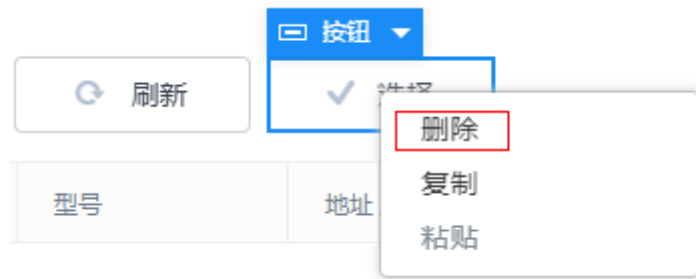


2. 只保留“新增行”按钮，删除其他工具栏中多余的按钮（单击选中待删除按钮，鼠标右键单击，**请确认已选中按钮**，然后选择“删除”，如果误删其他组件，在页面上单击返回按钮即可）。

图 3-126 工具栏中需要删除的按钮



图 3-127 选中待删除按钮




- 选中“新增行”按钮，在右侧属性面板中，修改“显示名称”为“新增设备”。在“事件”页签，再单击“新增行”的编辑按钮，在事件编排的“自定义动作”中，删除原来事件代码，输入以下事件代码，然后单击“保存”。


图 3-128 编辑事件代码



```
//跳转到编辑设备信息页面
context.$page.load('/besBaas/page#/HW_editEquipment');
```

- 单击页面上方的，保存设置。

步骤11 创建查询条件区域。

- 单击“模型视图”，切换到模型视图，在表格模型“table_0_condition”后，单击编辑图标。

说明

表格组件拖入后，页面将自动生成一个表格模型。“table_0_condition”即是在前面拖入的表格对应的表格模型，模型名可能跟拖入顺序及个数有关，一般情况下模型名称为“table_0_condition”，如果名称不一致，请根据实际情况修改。

图 3-129 编辑表格模型

名称	绑定组件	来源	映射	操作
equipmentInstance	datagrid_0 table_0	-	-	⊕ ✎ 🗑
queryEquipments		Api	queryEquipments.1.0.0	⊕ ✎ 🗑
deleteEquipment		Api	deleteEquipment.1.0.0	⊕ ✎ 🗑
table_0_condition		-	-	⊕ ✎ 🗑
pageInfo	pager_0	-	-	⊕ 🗑

- 在表格模型“table_0_condition”中，单击“新增节点”，添加一个“queryInfo”节点，“字段类型”为“Any”，然后在“queryInfo”下，再单击“新增节点”添加2个子节点“name”、“HW_fullAddress_CST”。

图 3-130 创建完成后的模型节点

名称
equipmentInstance
queryEquipments
deleteEquipment
table_0_condition
pageInfo
queryInfo
name
HW_fullAddress_CST


- 单击页面上方的 ，保存设置。
- 单击“设计视图”，切换到设计视图，从左侧组件列表中，拖一个“表单”组件到“表格容器”的最上部（表格容器内部最上部），在“元数据表单配置向导”弹窗中，单击“取消”。

图 3-131 向表格容器拖入表单




- 选中表单，在属性中“数据绑定”下，单击 ，为表单绑定“table_0_condition”中的“queryInfo”节点，并在提示弹窗中，单击“绑定并生成表单”，单击“确定”。

图 3-132 为表单绑定数据对象字段

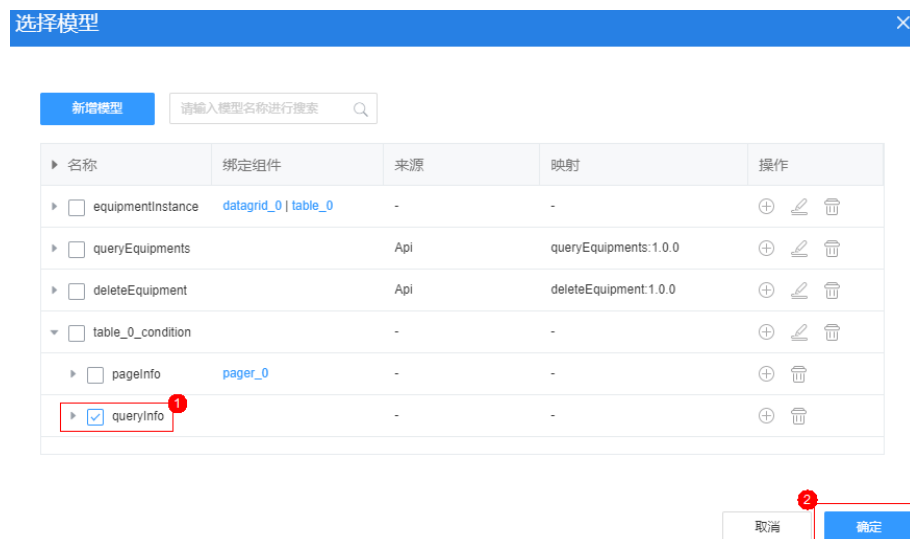


图 3-133 选择绑定并生成表单



- 在表单中，选择“name”输入框，修改“标签”为“设备名称”、选择“HW_fullAddress_CST”输入框，修改“标签”为“地址”。


- 选中“保存”按钮，修改“显示名称”为“查询”。在“事件”页签，单击，删除“提交表单”事件，然后单击“点击”后的“+”，进入事件编排，删除原有自定义代码，输入以下事件代码，单击“创建”。

图 3-134 修改按钮事件代码



须知


脚本中**红色内容**，请替换为实际的对象名、字段名、页面名、组件ID。其中“table_0_condition”为当前表格组件的ID号，中间数字默认为“0”，如果有多次修改或创建该ID号会变化，选中表格后，在组件树上可以查看实际组件ID。

```
let pageInfo = $model.ref('table_0_condition').getData().pageInfo;
let queryInfo = $model.ref('table_0_condition').getData().queryInfo;
if (!queryInfo) {
  queryInfo = {};
}
let queryData = {
  "fullAddress": queryInfo.HW__fullAddress__CST,
  "name": queryInfo.name,
  "start": 0,
  "limit": pageInfo.pageSize
};
$model.ref('queryEquipments').setData({inputParam: queryData});
$model.ref('queryEquipments').run().then(function (data) {
  $model.ref('equipmentInstance').setData(data.equipments);
  pageInfo.total = parseInt(data.total);
  $model.ref('table_0_condition').setData({"pageInfo": pageInfo});
}).catch(function (error) {
  console.log('error is', error);
});
```

查询条件区域创建后，如图3-135所示。

图 3-135 页面预览



8. 单击页面上方的, 保存设置。

步骤12 添加页面事件代码。

1. 在“设计视图”中，选中最外层的“页面”，可以直接单击组件导航，快速定位。
2. 在右侧“事件”页签中，单击“加载”后的“+”，为页面添加事件代码。
3. 在“添加动作”弹窗的“自定义动作”中，输入如下事件代码，然后单击底部“创建”。


须知

脚本中**红色内容**请替换为实际的对象名、字段名、页面名、组件ID。其中“table_0_condition”为当前表格组件的ID号，中间数字默认为“0”，如果有多次修改或创建该ID号会变化，选中表格后，在组件树上可以查看实际组件ID。

```
let pageInfo = $model.ref('table_0_condition').getData().pageInfo;
let queryData = {
  "start": (pageInfo.curPage - 1) * pageInfo.pageSize,
  "limit": pageInfo.pageSize
};
$model.ref('queryEquipments').setData({ inputParam: queryData });
$model.ref('queryEquipments').run().then(function(data) {
  $model.ref('equipmentInstance').setData(data.equipments);
  pageInfo.total = parseInt(data.total);
  $model.ref('table_0_condition').setData({"pageInfo": pageInfo });
}).catch(function(error) {
  console.log('error is', error);
});
```

---结束

验证

单击界面上方的, 进入预览页面，查看页面的展示效果，并验证以下功能：

步骤1 在界面上单击“新增设备”，查看是否跳转到“编辑设备”页面，如未跳转，请检查事件代码中页面名称前缀是否是实际空间名。

步骤2 检查“编辑设备”页面的“设备品牌”下拉框，“省市区”级联框的选项是否正确。

选项正确，则说明自定义模型、页面组件与模型绑定关系、以及页面on-load事件是正确的。

步骤3 检查“编辑设备”页面（`HW_editEquipment`）的设备录入功能是否正确。

1. 填写“设备名称”、“设备编码”等信息，单击“保存”。
2. 检查“设备管理”页面（`HW_equipmentManage`）是否包含新插入的数据。
如果查询结果页面显示新增数据，则说明新增功能以及页面的事件代码正确。
如果操作不成功，可以在页面事件脚本中增加debugger，使用Chrome开发者工具来调试定位问题。

步骤4 检查“编辑设备”页面（`HW_editEquipment`）的设备修改功能是否正确。

1. 单击一条记录后面的编辑按钮，查看是否跳转到“编辑设备”页面。
2. 修改“设备名称”、“设备编码”等信息，单击“保存”。
3. 检查“设备管理”页面（`HW_equipmentManage`）当前设备记录是否更新。
如果查询结果页面显示更新数据，则说明修改功能以及页面的事件代码正确。

步骤5 检查“设备管理”页面（`HW_equipmentManage`）的设备删除、查询、重置功能是否正确。

1. 选中空白行或其他行，单击“删除”，验证删除功能是否正常，如果不能删除，请检查操作列“删除”按钮事件代码。
2. 在查询区域分别输入“设备名称”、“地址”，单击“搜索”或“重置”验证查询电梯设备信息功能，如果不能查询或重置，请检查操作列按钮的事件代码。

----结束

3.1.6.5 开发“设备监控”功能

3.1.6.5.1 生成设备的经纬度信息

编辑设备页面可以获取到设备的经纬度信息，才能在地图上显示。因此需要先修改“编辑设备”页面，再通过调用地图API，生成经纬度数据。

在前面章节中，在编辑设备时，仅生成了设备的基本信息，没有生成精确的经纬度。如果想将设备的位置显示在地图上，则必须给设备生成一个精确的经纬度。因此在本节中，将进一步改造“编辑设备”页面，实现通过设备“详细地址”，即可以生成一个经纬度信息，然后通过查询设备的后台逻辑（脚本、服务编排及公共服务），将信息传到高级页面的自定义组件中。其中，设备的经纬度信息通过设备的详细地址生成。

操作步骤

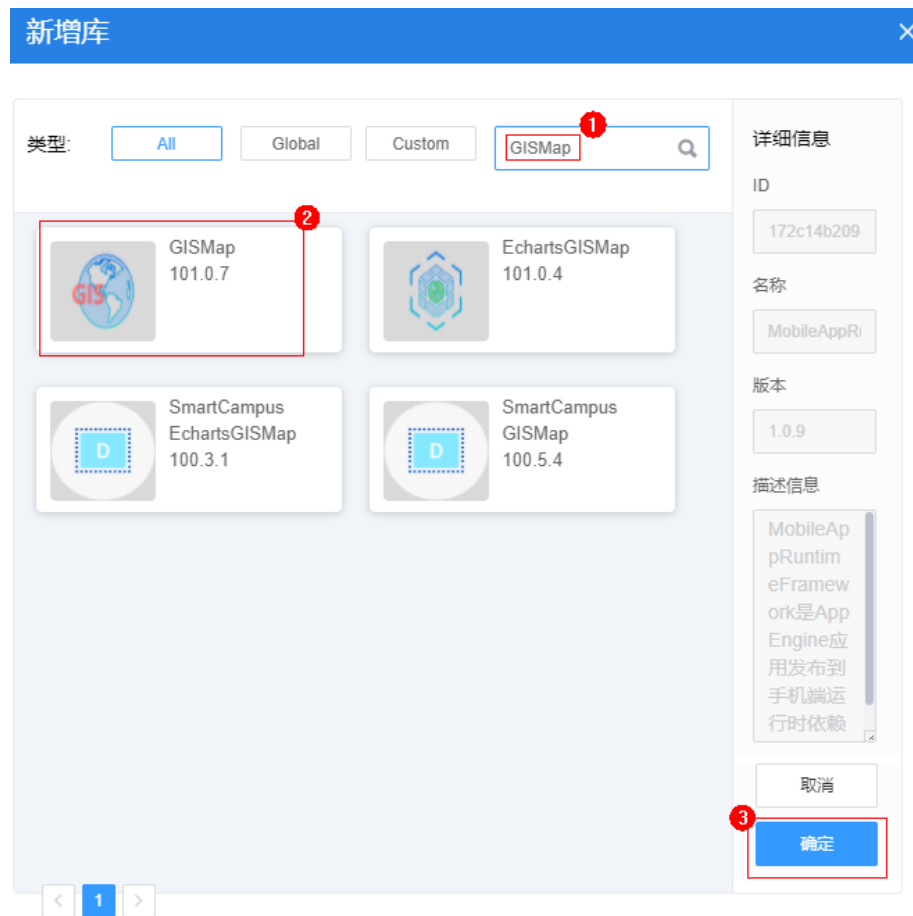
步骤1 进入“设备维修管理系统”应用。

步骤2 单击打开编辑设备“`HW_editEquipment`”页面，单击锁定页面。

步骤3 引入地图公共库。

1. 在右侧属性面板中，选择“库”页签，单击“新增库”。
2. 搜索“GISMap”，在搜索结果中，选中“GISMap”库，单击“确定”。

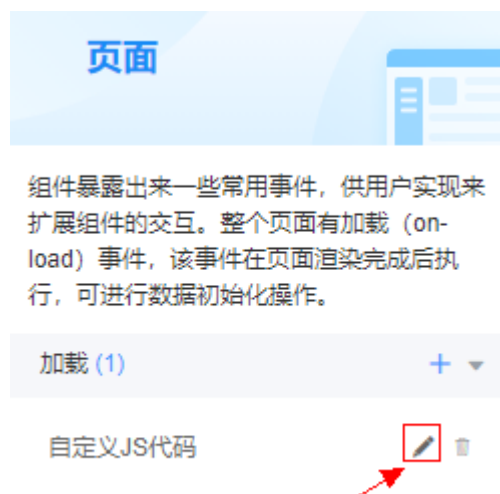
图 3-136 新增库



步骤4 修改页面事件代码。

1. 在“设计视图”中，选中最外层的“页面”，在右侧“事件”页签中，单击“自定义JS代码”后的✎。

图 3-137 添加页面事件代码




2. 在弹出的事件编排“自定义动作”中，在代码编辑区增加如下粗体脚本代码。

注意

预置的地图密钥（N1XIMTXn6wg8PyAMS34utWkOhWiFw90）仅供学习使用，如果您的商用业务系统中使用到地图组件，请重新申请自己的密钥。

```
//加载地图API做地址转换用
window.initMap = function (gis) {
  window.gisMap = gis.loadMap({
    container: document.getElementById('map'),
    center: [114.0578388103, 22.6565079747],
    zoom: 4,
    viewMode: '2D',
    mapStyle: 'normal',
    layers: [],
  });
};
//等待页面加载完成，发事件给地图widget初始化页面
var interval = setInterval(function () {
  if (gis) {
    clearInterval(interval);
    new gis.Hmap2D("BAIDU_MAP", "N1XIMTXn6wg8PyAMS34utWkOhWiFw90", initMap);
  }
}, 300);
//debugger;
//为级联组件下拉框赋值
$model.ref('cascaderOptions').setData([
  { "value": "110000000000", "label": "北京市", "children": [
    { "value": "110100000000", "label": "市辖区", "children": [
      { "value": "110101000000", "label": "东城区" },
      { "value": "110102000000", "label": "西城区" },
      { "value": "110105000000", "label": "朝阳区" },
      { "value": "110108000000", "label": "海淀区" } ] ] },
    { "value": "310000000000", "label": "上海市", "children": [
      { "value": "310100000000", "label": "市辖区", "children": [
        { "value": "310101000000", "label": "黄浦区" },
        { "value": "310104000000", "label": "徐汇区" },
        { "value": "310106000000", "label": "静安区" },
        { "value": "310115000000", "label": "浦东新区" },
        { "value": "310110000000", "label": "杨浦区" },
        { "value": "310112000000", "label": "闵行区" } ] ] },
    { "value": "440000000000", "label": "广东省", "children": [
      { "value": "440100000000", "label": "广州市", "children": [
        { "value": "440101000000", "label": "市辖区" },
        { "value": "440105000000", "label": "海珠区" },
        { "value": "440106000000", "label": "天河区" },
        { "value": "440111000000", "label": "白云区" },
        { "value": "440113000000", "label": "番禺区" } ] ],
      { "value": "440300000000", "label": "深圳市", "children": [
        { "value": "440301000000", "label": "市辖区" },
        { "value": "440303000000", "label": "罗湖区" },
        { "value": "440304000000", "label": "福田区" },
        { "value": "440305000000", "label": "南山区" },
        { "value": "440306000000", "label": "宝安区" },
        { "value": "440307000000", "label": "龙岗区" },
        { "value": "440308000000", "label": "盐田区" },
        { "value": "440309000000", "label": "龙华区" },
        { "value": "440310000000", "label": "坪山区" } ] ] } ] ] );
//从页面url中获取id，然后将id作为queryEquipmentDetail的入参
let id = Page.params.id;
if (id && id != "") {
  let _model = $model.ref("queryEquipmentDetail");
  _model.setValue("inputParam", { "equipmentId": id });
  _model.run().then(() => {
    //获取queryEquipmentDetail的出参后赋值给页面表单模型equipmentForm
    var data = _model.getData();
    if (data.outputParam && data.outputParam.equipment) {
      var equip = data.outputParam.equipment;
      $model.ref("equipmentForm").setData(equip);
    }
  }).catch((e) => {
    this.$dialog.error({
      title: '错误',
      content: e.resMsg
    });
  });
}
```

3. 单击页面上方的，保存修改并退出事件编排。

步骤5 在页面中引用地图实例。

1. 在“设计视图”中，选中“表单”，从左侧组件列表“高级”中，拖一个“代码块”到“表单”后（表单与代码块同级）。

- 选中“代码块”，在右侧“模板”文本框中填写如下代码，即在页面上创建一个DIV，存放地图实例。

```
<div id="map"></div>
```

图 3-138 在页面中引用地图实例



步骤6 修改“保存”的事件代码。

- 在“设计视图”中，选中“保存”。
- 在右侧“事件”页签中，单击“自定义JS代码”后的✎。

图 3-139 添加保存按钮事件代码

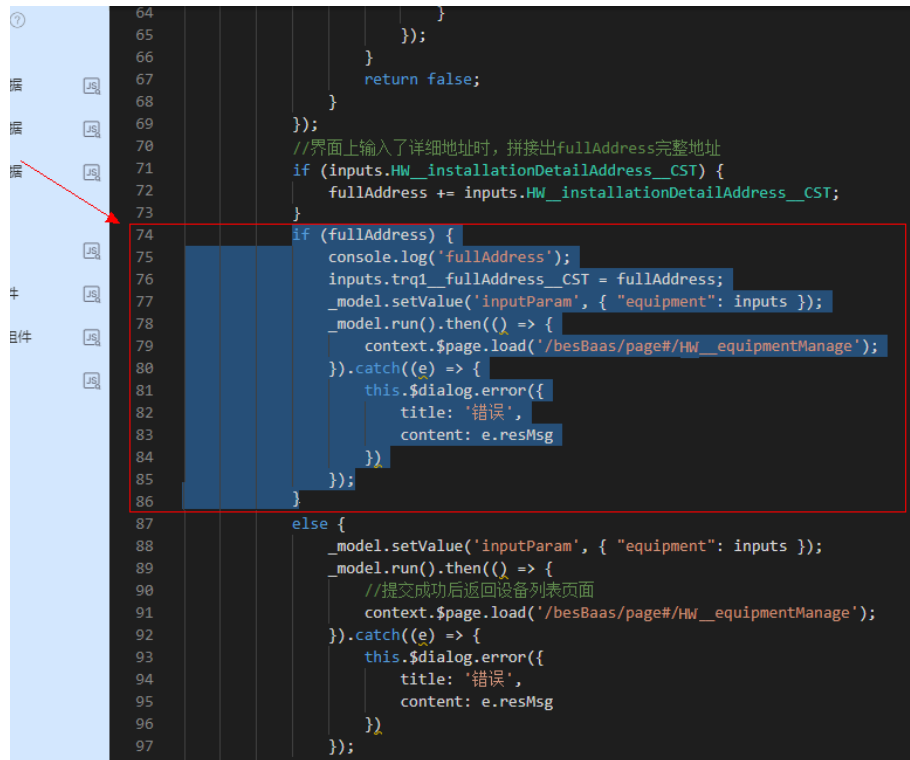


- 在事件编排的“自定义动作”中，在代码编辑区将下图中（大约74~86行）的代码，替换成如下脚本代码。

须知

脚本中**红色内容**请替换为实际的对象名、字段名、页面名。

图 3-140 替换代码



```

if (fullAddress) {
    inputs.HW_fullAddress_CST = fullAddress;
    //调用百度地图API, 根据fullAddress完整地址, 计算出经纬度坐标
    //debugger;
    var geocoder = gisMap.getGeocoder();
    geocoder.getPoint(fullAddress, function (status, result) {
        //转换完成后, 将结果存入服务编排: editEquipment的入参中
        if (result && result.lat && result.lng) {
            inputs.HW_longitude_CST = result.lng + "";
            inputs.HW_latitude_CST = result.lat + "";
        }
        _model.setValue('inputParam', { "equipment": inputs });
        _model.run().then(() => {
            context.$page.load('/besBaas/page#/HW_equipmentManage');
        }).catch((e) => {
            this.$dialog.error({
                title: '错误',
                content: e.resMsg
            });
        });
    });
}
};//增加
}

```

替换后完整代码, 如下:

```

//debugger;
// 当前组件
let _component = context.$component.current;
//校验表单
_component.getForm().formValidate().then((val) => {
    //校验成功
    //从自定义模型equipmentForm中获取界面表单数据
    let data = $model.ref('equipmentForm').getData();
    if (data) {
        console.log(data);
        //构造Script ( editEquipment ) 的其他入参
        let inputs = data;
    }
});


```

```
inputs.HW__installationSiteProvince__CST = ""; //构造参数名称要与Object字段名保持一致
inputs.HW__installationSiteCity__CST = "";
inputs.HW__installationSiteArea__CST = "";
inputs.HW__fullAddress__CST = "";
inputs.HW__longitude__CST = "";
inputs.HW__latitude__CST = "";
//获取服务模型editEquipment，执行模型的动作，即执行模型关联的Script
let _model = $model.ref('editEquipment');
if (!inputs.cascaderAddress || inputs.cascaderAddress.length < 1) {
  _model.setValue('inputParam', { "equipment": inputs });
  _model.run().then(() => {
    //提交成功后返回设备列表页面
    context.$page.load('/besBaas/page#/HW__equipmentManage');
  }).catch((e) => {
    this.$dialog.error({
      title: '错误',
      content: e.resMsg
    })
  });
}
else {
  console.log('data');
  //将级联框的内容转换为Script ( editEquipment ) 的省、市、区3个参数，并拼接完整地址
  let fullAddress = "";
  let cascaderOptions = $model.ref('cascaderOptions').getData();
  if (cascaderOptions.length < 1) {
    throw new Error("CascaderOptions data is wrong.");
  }
  cascaderOptions.forEach(function (province, idx) {
    if (province.value == inputs.cascaderAddress[0]) {
      //构造参数名称要与Object字段名保持一致
      inputs.HW__installationSiteProvince__CST = inputs.cascaderAddress[0];
      fullAddress += province.label;
      if (inputs.cascaderAddress.length > 1) {
        province.children.forEach(function (city, idxCity) {
          if (city.value == inputs.cascaderAddress[1]) {
            //构造参数名称要与Object字段名保持一致
            inputs.HW__installationSiteCity__CST = inputs.cascaderAddress[1];
            fullAddress += city.label;
            if (inputs.cascaderAddress.length > 2) {
              city.children.forEach(function (area, idxArea) {
                if (area.value == inputs.cascaderAddress[2]) {
                  //构造参数名称要与Object字段名保持一致
                  inputs.HW__installationSiteArea__CST = inputs.cascaderAddress[2];
                  fullAddress += area.label;
                  return false;
                }
              });
            }
          });
        });
      }
      return false;
    }
  });
}
return false;
});
//界面上输入了详细地址时，拼接出fullAddress完整地址
if (inputs.HW__installationDetailAddress__CST) {
  fullAddress += inputs.HW__installationDetailAddress__CST;
}
if (fullAddress) {
  inputs.HW__fullAddress__CST = fullAddress;
  //调用百度地图API，根据fullAddress完整地址，计算出经纬度坐标
  //debugger;
  var geocoder = gisMap.getGeocoder();
  geocoder.getPoint(fullAddress, function (status, result) {
    //转换完成后，将结果存入服务编排：editEquipment的入参中
    if (result && result.lat && result.lng) {
      inputs.HW__longitude__CST = result.lng + "";
    }
  });
}
```


```

        inputs.HW__latitude__CST = result.lat + "";
    }
    _model.setValue('inputParam', { "equipment": inputs });
    _model.run().then(() => {
        context.$page.load('/besBaas/page#/HW__equipmentManage');
    }).catch((e) => {
        this.$dialog.error({
            title: '错误',
            content: e.resMsg
        })
    });
    });//增加
}
else {
    _model.setValue('inputParam', { "equipment": inputs });
    _model.run().then(() => {
        //提交成功后返回设备列表页面
        context.$page.load('/besBaas/page#/HW__equipmentManage');
    }).catch((e) => {
        this.$dialog.error({
            title: '错误',
            content: e.resMsg
        })
    });
}
}
}
}).catch((error) => {
});

```

4. 单击页面上方的，保存修改并退出事件编排。

步骤7 测试能否正确生成经纬度信息：构造测试数据。

1. 单击界面上方预览图标，使用编辑设备“HW_editEquipment”页面新增如下表的设备数据。

操作成功，系统会返回管理设备“HW_equipmentManage”页面。

设备名称	设备编码	品牌	型号	地址
1号电梯	123456789	日立	123456789	广东省深圳市龙岗区四季花城
2号电梯	123456	迅达	123456	广东省深圳市坂田华为基地

2. 通过“HW_equipmentManage”页面修改另一条设备数据。

步骤8 检查测试结果：测试能否正确生成经纬度信息。

1. 预览“HW_equipmentManage”页面。
2. 查看[上一步](#)里添加的设备是否已经生成了经纬度。

图 3-141 查看生成的经纬度

设备名称	设备编码	品牌	型号	地址	Longitude	Latitude
2号电梯	123456789	迅达	123456789	广东省深圳市龙岗区四季花城	114.05244092448586	22.62720865921195
1号电梯	123456	日立	654321	广东省深圳市龙岗区坂田华为基	114.0540901540466	22.65304311238915

----结束

3.1.6.5.2 组装“设备监控”页面

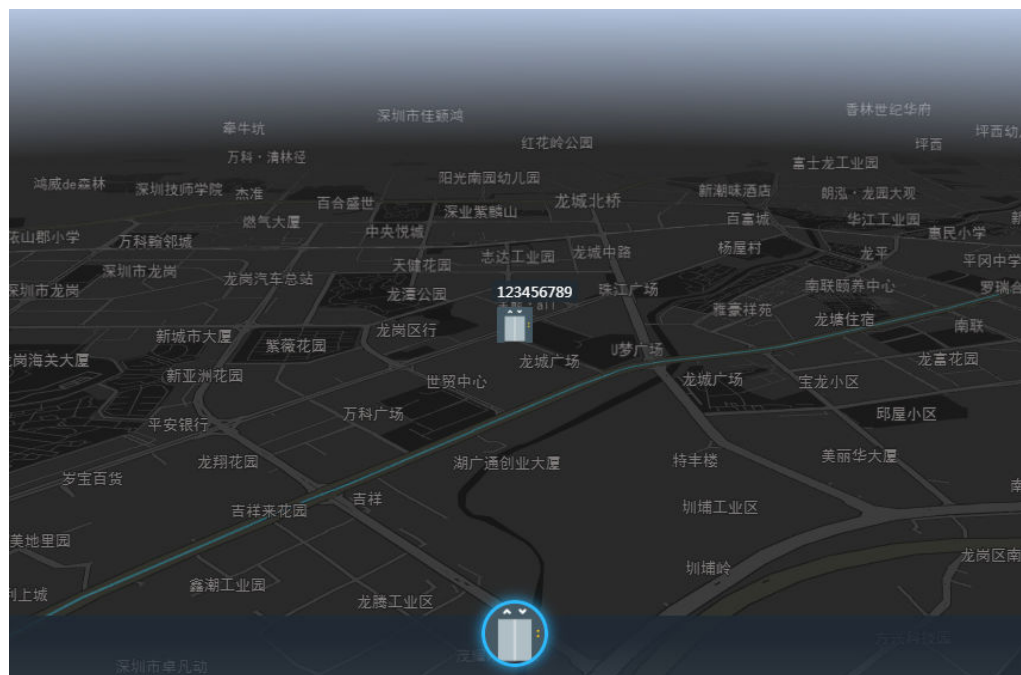
通过拖拽组件、配置组件，开发高级页面显示设备位置功能。“设备监控”是在地图上显示电梯设备的实时位置，本节我们将采用高级页面实现。

“设备监控”高级页面由地图组件和监控组件（CaseMapControllerWidget.zip）拼装而成，地图widget负责显示地图，监控widget负责根据电梯的经纬度坐标，将其显示在地图上，显示设备的页面构想如下图所示。

📖 说明

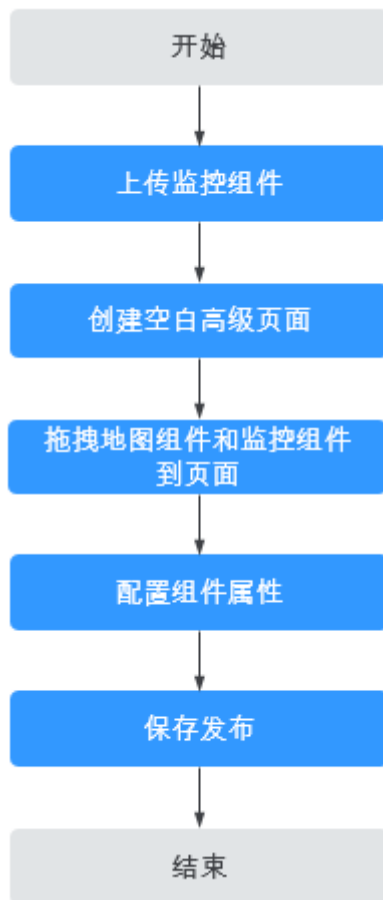
“CaseMapControllerWidget”组件已经开发完成，您可以直接在本文中获取组件包，上传到租户账号下直接使用。

图 3-142 设备监控效果




设备监控页面大致开发流程：

图 3-143 设备监控页面开发流程



操作步骤



步骤1 上传监控组件。

1. 下载监控widget文件包[CaseMapControllerWidget.zip](#)到本地。
2. 进入“设备维修管理系统”应用，在应用开发界面左侧列表单击，选择“高级页面 > 组件”，单击“提交新组件”。
3. 单击“请选择源文件(.zip)”，选择下载的“CaseMapControllerWidget.zip”包，填写基本信息（名字“CaseMapControllerWidget”和发行说明），单击“提交”进行上传。

说明

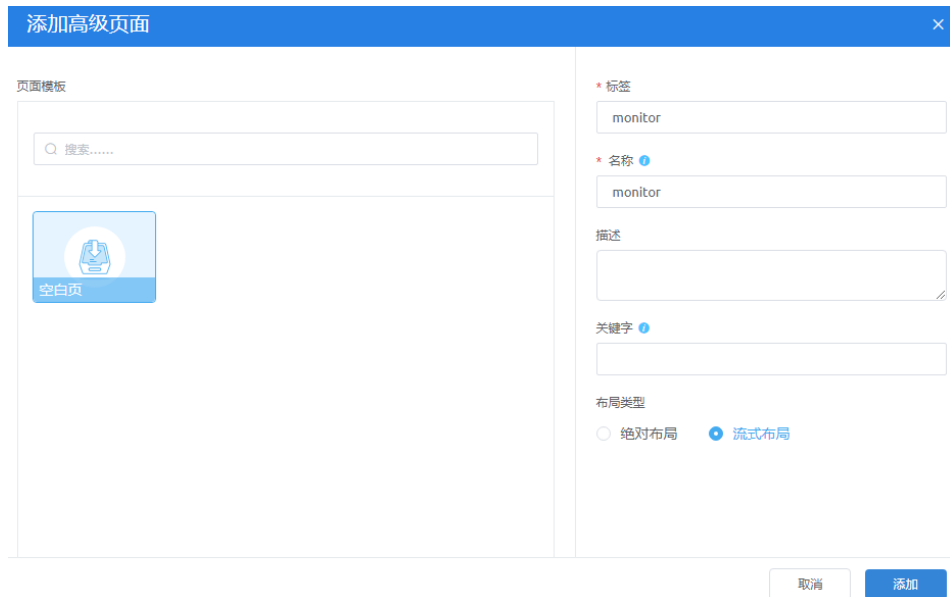
如果提示您已安装此组件，建议参考[如何更新自定义组件](#)中操作，下载文档中的组件，在AstroZero中更新此组件。

步骤2 创建空白高级页面。

1. 单击，进入工作目录。
2. 在“Equipment”目录下，将鼠标放在“Page”上，单击界面上出现的，在弹出菜单中选择“高级页面”。

3. 设置“标签”和“名称”为“monitor”，并选择“流式布局”，单击“添加”。

图 3-144 添加高级页面



步骤3 拖拽组件到页面编辑区。


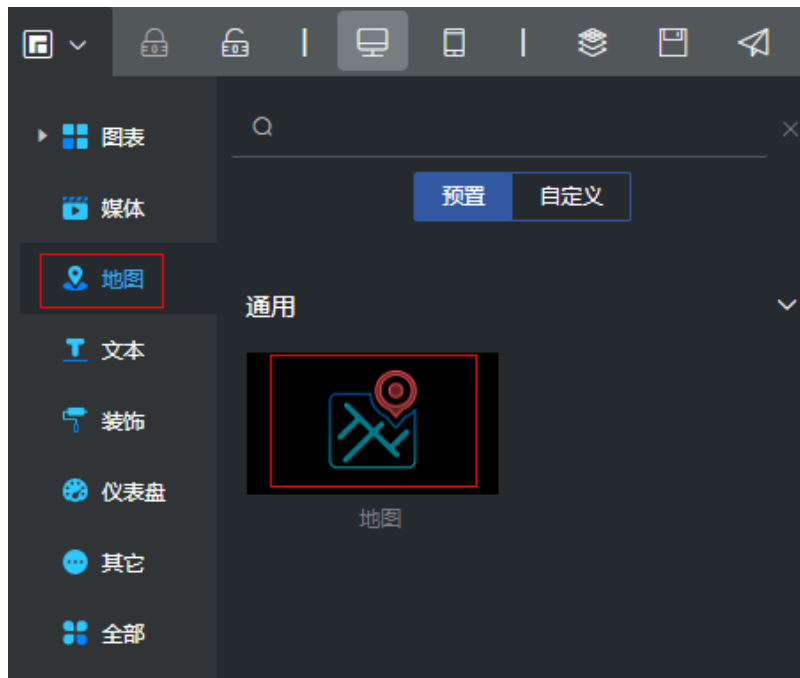
1. 单击左上角，打开组件列表。在“地图”组件下，拖拽一个通用“地图”到页面编辑区，如下图所示。

图 3-145 地图组件



2. 单击左侧列表的“全部”，然后再单击“自定义”页签，搜索“CaseMapControllerWidget”，将步骤1中上传的自定义组件拖到地图上。

步骤4 配置自定义组件CaseMapControllerWidget。

1. 将页面向下滑动，再单击页面下部的空白处，右侧显示当前视图组件列表。
2. 单击“CaseMapControllerWidget”组件，会在右侧显示该组件的属性配置面板。

图 3-146 自定义 Widget 配置图标

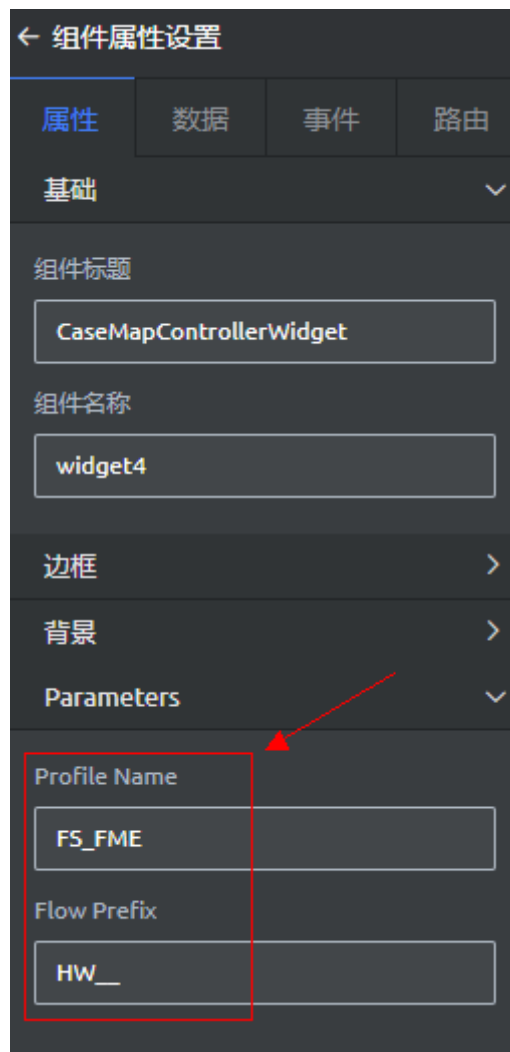


3. 修改属性参数：设置“Flow Prefix”为租户命名空间前缀“*HW_*”。

📖 说明

请注意“*HW_*”这里命名空间前缀后有2个下划线。当前组件代码中，调用的服务名称为“queryEquipments”，因此只有在此配置前缀后，才能拼装出正确的名称“*HW_*queryEquipments”。

图 3-147 设置基础参数



4. 配置桥接器的参数信息。

在“数据”页签，然后单击“通用AppCube Flow数据桥接器”，选择桥接器如下图所示。

图 3-148 设置桥接器



数据参数说明如下。


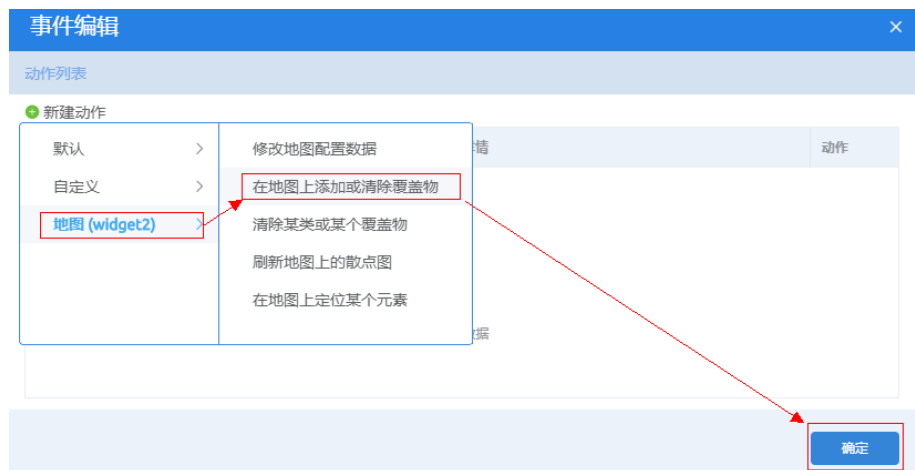
- 桥接器实例：调用的桥接器名称，选中“通用AppCube Flow数据桥接器”。
 - 请求方法：调用的方法名，如get（查询）、put（增加）、post（修改）和delete（删除），保持默认。
 - 端点：选中“Flow”。如果选API，表示调用服务编排的API。
 - 前缀：调用服务编排的版本号，此处不用配置。
5. 在“事件”页签，配置事件，使自定义widget与地图widget关联。
- a. 在“事件”页签下，单击“AddMarkersToMap”后的设置图标, 选择“新建动作 > 地图 (widget2) > 在地图上添加或清除覆盖物”，单击“确定”。

图 3-149 配置事件





- b. 再单击“centerAndZoom”后的设置图标, 选择“新建动作 > 地图 (widget2) > 修改地图配置数据”, 单击“确定”。

图 3-150 配置地图事件



步骤5 单击页面上方的, 保存页面。

步骤6 单击页面上方的, 发布页面。

----结束

验证

单击页面上方的, 进入预览页面, 查看页面演示效果。

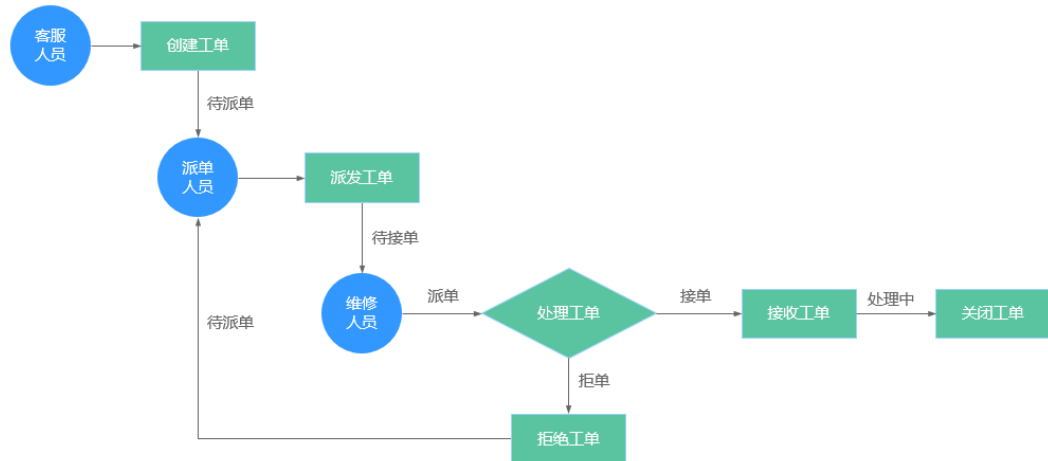
在页面上单击, 查看地图中是否显示已添加的电梯设备的实时位置。

3.1.7 工单管理开发

3.1.7.1 开发前必读

工单管理功能是通过创建脚本，实现工单创建、工单派发以及工单处理等功能。通过创建的BPM，实现工单在客服人员、派单人员、维修人员之间流转过，如图3-151。

图 3-151 工单流转大致流程



1. 电梯客服人员受理用户投诉，并创建维修工单。
2. 派单人员收到客服人员的维修工单后，派发给维修工程师。
3. 维修工程师进行现场修理，并在处理完成后关闭维修单。

3.1.7.2 定义数据对象“工单对象”

对于工单对象功能，需要先创建一个工单对象WorkOrder，保存工单、型号、资产编号等信息，如表3-14所示。

表 3-14 工单对象 WorkOrder 信息

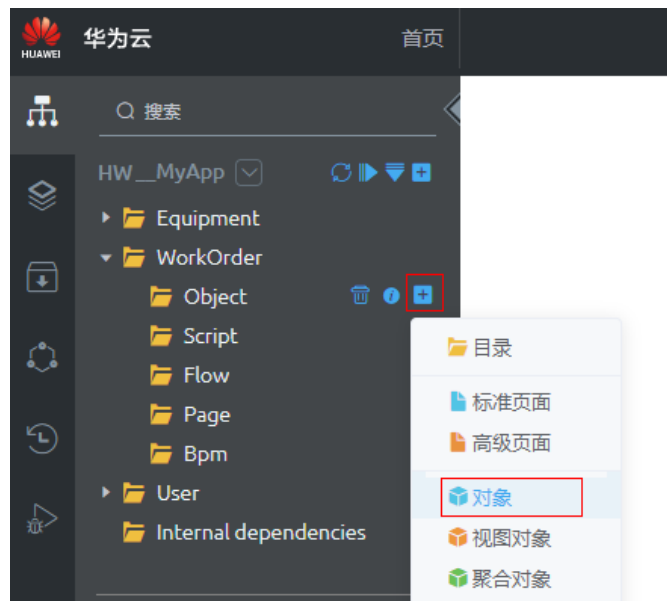
字段标签	字段名称	字段描述	数据类型	建议长度	备注
instanceId	instanceId	BPM实例Id	文本	255	-
createdBy	createdBy	工单创建人	文本	255	-
equipmentId	equipmentId	设备ID	文本	255	-
type	type	工单类型	文本	255	-
description	description	备注	文本	255	-
source	source	来源	文本	255	-
faultPhenomenon	faultPhenomenon	故障现象说明	文本	255	-
arriveTime	arriveTime	客户要求工程师到达现场时间	日期/时间	-	-

字段标签	字段名称	字段描述	数据类型	建议长度	备注
recoveryTime	recoveryTime	客户要求故障恢复时间	日期/时间	-	-
customer	customer	客户接口人	文本	255	-
assignedFme	assignedFme	当前处理人	文本	255	-
title	title	现场服务单标题	文本	255	-
status	status	服务单状态	文本	255	-
priority	priority	服务单优先级	文本	255	-
workOrderId	workOrderId	服务单ID	文本	255	区分大小写、是否必选（否），是否唯一（是）、是否可搜索（是）

操作步骤

- 步骤1** 在经典版开发环境“首页 > 我的应用”中，单击“设备维修管理系统”，进入之前创建的应用。
- 步骤2** 在“WorkOrder”中，将鼠标放在“Object”目录上，单击界面上出现的“+”，在弹出菜单中选择“对象”。

图 3-152 创建对象



步骤3 选中“创建新对象”，在“标签”和“名称”文本框中输入“WorkOrder”，单击“添加”。

系统实际创建的对象名称为“HW_WorkOrder_CST”，“HW_”前缀由租户命名空间决定，“_CST”后缀代表是自定义对象。

步骤4 按系统向导完成第1个字段“设备ID”（equipmentId）的定义。

1. 在“自定义字段”页签，单击“新建”。
2. 选中字段类型，设置字段类型为“文本”，单击“下一步”。
3. 输入字段详细信息，如图3-153所示，设置字段标签、名称为“equipmentId”、字段长度255，单击“下一步”。

须知

平台会根据字段标签自动生成字段名称，但请参照表3-14，修改字段名。对于utf-8编码，一个汉字占用三个字节。

图 3-153 定义“设备 ID”详细信息

* 标签	* 名称
<input type="text" value="equipmentid"/>	<input type="text" value="equipmentid"/> <input type="text" value="..._CST"/>
字段帮助	描述
<input type="text" value="请输入"/>	<input type="text" value="请输入"/>
* 数据长度	默认值
<input type="text" value="255"/>	<input type="text" value="请输入"/>
<input type="checkbox"/> 大小写敏感	<input type="checkbox"/> 是否必填
<input type="checkbox"/> 是否唯一	<input type="checkbox"/> 是否可搜

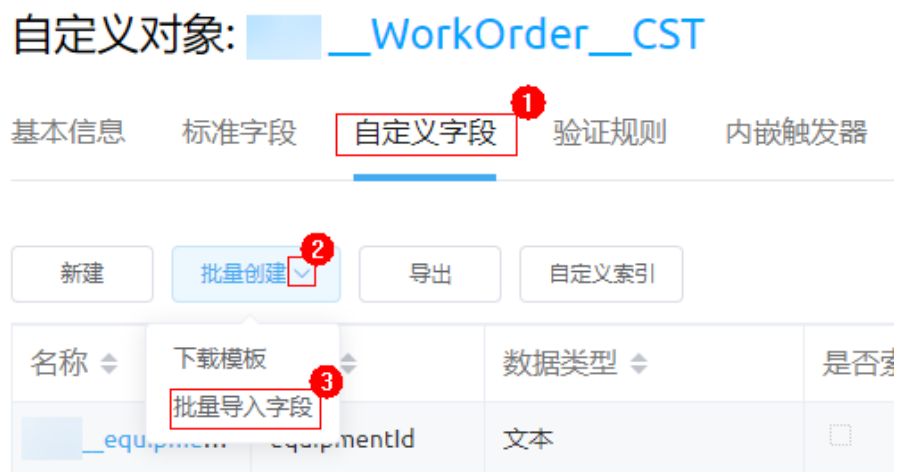
系统实际创建的字段名称为“*HW*_equipmentId_CST”，“*HW*”前缀由租户命名空间namespace决定，“_CST”后缀代表是自定义字段。

4. 配置字段级安全，选中“读取”和“编辑”复选框，为所有预置profile配置能编辑和读取本字段的权限，单击“下一步”。
5. 将字段添加到页面布局，选中“添加本字段到该页面布局”，单击“保存”。

步骤5 通过导入字段模板，创建表3-14中剩余字段。

1. 下载待导入字段模板import-fields-template-WorkOrder.xlsm到本地。
2. 单击当前对象的“自定义字段”，然后单击“批量创建”后的▾，选择“批量导入字段”，选择上一步下载的字段模板。

图 3-154 导入自定义字段



导入完成后，刷新页面，重新进入对象的“自定义字段”，即可查看已导入的字段。

图 3-155 查看已创建自定义字段

自定义对象: [HW_WorkOrder_CST](#)

基本信息 标准字段 自定义字段 验证规则 内嵌触发器 布局

新建 批量创建 导出 自定义索引 搜索

名称	标签	数据类型	是否索引	是否必填	描述	操作
HW_instanceld_CST	instanceld	文本	<input type="checkbox"/>	<input type="checkbox"/>	BPM实例id	
HW_createdBy_CST	createdBy	文本	<input type="checkbox"/>	<input type="checkbox"/>	工单创建人	
HW_equipmentId_CST	equipmentId	文本	<input type="checkbox"/>	<input type="checkbox"/>	设备Id	
HW_type_CST	type	文本	<input type="checkbox"/>	<input type="checkbox"/>	工单类型	
HW_description_CST	description	文本	<input type="checkbox"/>	<input type="checkbox"/>	备注	
HW_source_CST	source	文本	<input type="checkbox"/>	<input type="checkbox"/>	来源	
HW_faultPhenomenon_CST	faultPhenomenon	文本	<input type="checkbox"/>	<input type="checkbox"/>	故障现象说明	
HW_arriveTime_CST	arriveTime	日期/时间	<input type="checkbox"/>	<input type="checkbox"/>	客户要求工程师到达现场时间	
HW_recoveryTime_CST	recoveryTime	日期/时间	<input type="checkbox"/>	<input type="checkbox"/>	客户要求故障恢复时间	
HW_customer_CST	customer	文本	<input type="checkbox"/>	<input type="checkbox"/>	客户接口人	

共 15 条 10条/页 < 1 2 >

----结束

3.1.7.3 开发“客服人员创建工单”功能

3.1.7.3.1 本节导读

“客服人员创建工单”包含客服人员生成工单和查询工单两个功能，这两个功能我们分别通过两个标准页面实现：

- 生成工单页面：客服人员输入工单信息，生成工单页面。
- 工单列表（客服人员）页面：客服人员在工单列表页面查询工单信息，查看所有工单的列表页面。

学习地图

如图3-156所示，通过本章的学习和实践，您将进一步了解“标准页面”的能力，包括：

- 拼装复杂页面
- 了解折叠面板
- 学习表格：自定义某查询结果字段的显示内容
- 动态加载页面内容
- 了解在事件中调用脚本的API

图 3-156 学习地图



3.1.7.3.2 开发“生成工单”功能

创建“生成工单”脚本及公共接口

“生成工单”功能对应电梯管理流程的第一个节点，客服人员在录入完成维修信息、提交维修单时将会调用“生成工单”脚本及公共接口。

创建“生成工单”脚本

步骤1 在“我的应用”中，单击“设备维修管理系统”，进入应用。

步骤2 在“WorkOrder”目录中，将鼠标放在“Script”上，单击界面上出现的“+”，在弹出菜单中选择“脚本”。

步骤3 在弹窗中，选中“创建一个新脚本”，在“名称”文本框中输入“createWorkOrder”，单击“添加”。

当编辑已有脚本时，为防止编辑时多人篡改，编辑前请单击进行锁定。

步骤4 在代码编辑器中插入如下脚本代码。

须知

脚本中**红色内容**请替换为实际的对象名、字段名。



```
//本脚本用于创建工单
import * as db from 'db';//导入处理object相关的标准库
import * as context from 'context';//导入上下文相关的标准库
import * as date from 'date';
//定义入参结构，入参包含1个参数：workOrder对象，为必填字段
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'Object', required: true, label: 'object' })
  workOrderData: object;
}
//定义出参结构，出参包含1个参数，workOrder的记录id
@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'String' })
  id: string;
}
//使用数据对象HW_WorkOrder_CST
```



```
@useObject(['HW_WorkOrder_CST'])
@action.object({ type: "method" })
export class CreateWorkOrder { //定义接口类, 接口的入参为ActionInput, 出参为ActionOutput
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public createWorkOrder(input: ActionInput): ActionOutput {
    let out = new ActionOutput(); //新建出参ActionOutput类型的实例, 作为返回值
    let error = new Error(); //新建错误类型的实例, 用于在发生错误时保存错误信息
    try {
      let workOrderData = input.workOrderData; //将入参赋值给workOrderData变量, 方便后面使用
      let s = db.object('HW_WorkOrder_CST'); //获取HW_WorkOrder_CST这个Object的操作实例
      let userName = context.getUserName();
      workOrderData['HW_status_CST'] = '待派单';
      workOrderData['HW_assignedFme_CST'] = '派单员';
      workOrderData['HW_createdBy_CST'] = userName;
      let id = s.insert(workOrderData);
      if (id) {
        out.id = id;
      } else {
        error.name = "WOERROR";
        error.message = "无法创建工单! ";
        throw error;
      }
    } catch (error) {
      console.error(error.name, error.message);
      context.setError(error.name, error.message);
    }
    return out;
  }
}
```

步骤5 单击编辑器上方, 保存脚本。

步骤6 测试脚本能否正常执行。

1. 单击编辑器上方的, 执行脚本。
2. 在界面底部输入测试数据, 单击测试窗口右上角执行图标。


测试报文样例如下:

```
{
  "workOrderData":{
    "createdDate":"2020-09-30 12:00:00",
    "HW_recoveryTime_CST":"2020-09-30 12:00:00",
    "HW_arriveTime_CST":"2020-09-30 12:00:00",
    "HW_title_CST":"电梯无法关门",
    "HW_priority_CST":"高",
    "HW_workOrderId_CST":"WD0000123456",
    "HW_instanceld_CST":""
  }
}
```

执行成功, 会在“输出”页签返回查询结果。

图 3-157 输出工单 ID 号

问题	输入参数	输出参数	日志
1	{		
2		"id": "cwsC000000dbIvrNqsVs"	
3	}		

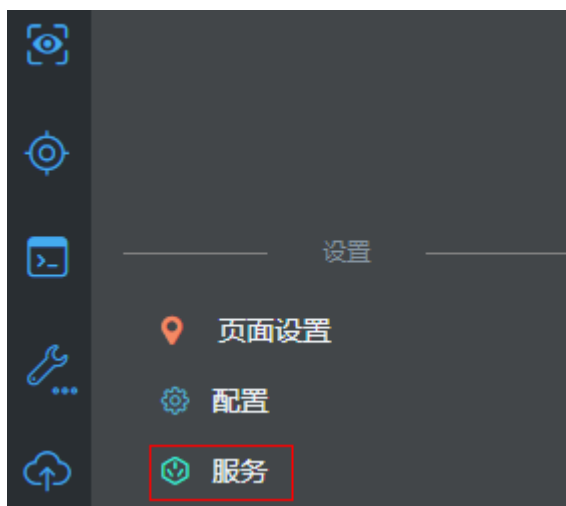
步骤7 测试成功，单击编辑器上方的，启用脚本。

----结束

创建“生成工单”公共接口

步骤1 在APP视图下，单击“服务”，进入公共接口创建页面。

图 3-158 创建公共接口入口



步骤2 单击“新建”，创建“生成工单”脚本“*HW_createWorkOrder*”的公共接口。

设置接口参数信息：操作名称为“*createWorkOrder*”，版本为“1.0.0”，URL为“/*createWorkOrder*”，“类型”选择“脚本”，“资源”为“*HW_createWorkOrder*”，方法为“POST”，然后单击“保存”。

图 3-159 新建 createWorkOrder 接口

The screenshot shows a configuration form for a new API endpoint. The fields are as follows:

- 标签 (Label): createWorkOrder
- * 操作名称 (Operation Name): createWorkOrder
- * 版本 (Version): 1.0.0
- * URL: /service/..._MyApp/1.0.0 /createWorkOrder
- 内容类型 (Content Type): application/json
- 分类 (Category): 请输入 (Please enter)
- 描述 (Description): 请输入 (Please enter)
- 允许匿名访问 (Allow Anonymous Access):
- 类型 (Type): 脚本 (Script), 服务编排 (Service编排), 对象 (Object)
- 自定义响应 (Custom Response):
- * 资源 (Resource): ..._createWorkOrder
- * 方法 (Method): POST
- Buttons: 保存 (Save), 取消 (Cancel)

----结束

组装“生成工单”页面

“生成工单”对应设备管理流程的第一个节点，客服人员使用这个页面完成维修信息录入，生成维修工单，并提交给派单人员处理。该页面使用标准页面组装实现。

页面组件分析

“生成工单”页面的组装过程如下：

1. 在页面中，拖入一个**容器**组件。
2. 在**容器**组件中，拖入一个**表单**组件和**容器**组件。
 - a. 将**表单**组件分割为**3个区域**：“基本信息”、“设备信息”、“其他信息”。
 - b. 将**2级容器**作为“按钮”区域。

组装完成后，整个页面有4个区域，如**图3-160**所示。

图 3-160 生成工单页面

The screenshot shows a form for generating a work order. It is organized into three main sections, each with a blue header and a dropdown arrow:

- 基本信息 (Basic Information):** Contains fields for 设备名称 (Device Name), 工单标题 (Work Order Title), 创建时间 (Creation Time), 工单类型 (Work Order Type), 优先级 (Priority), 工单来源 (Work Order Source), and 备注 (Remarks).
- 设备信息 (Equipment Information):** Contains fields for 设备编码 (Device Code), 设备品牌 (Device Brand), 设备型号 (Device Model), and 地址 (Address).
- 其他信息 (Other Information):** Contains a 故障记录 (Fault Record) section with a 故障现象说明 (Fault Phenomenon Description) field, and a 时间记录 (Time Record) section with 客户要求工程师到达现场时间 (Customer Request Engineer Arrival Time), 客户要求故障恢复时间 (Customer Request Fault Recovery Time), and 客户接口人 (Customer Contact Person) fields.

At the bottom of the form, there are two buttons: 提交 (Submit) and 取消 (Cancel).

其中，“基本信息”、“设备信息”是通过数据绑定页面模型直接创建的表单，“其他信息”及“按钮”区域是手动拖入的组件，然后再进行数据绑定，页面中详细组件分布如图3-161所示。

图 3-161 页面组件分析



页面模型分析

页面模型负责与页面组件交互，获取生成工单需要的数据，页面模型需要先定义，然后再与页面组件进行绑定。

对于设备维修工单，需要录入的信息包括：工单基本信息、设备信息以及工单扩展信息。

打开创单页面，有如下处理逻辑：

1. 在“基本信息”区域指定“设备名称”时，在“设备详情”区域自动显示设备信息。该能力需要开发（`equipmentSelectListQuery`）脚本实现。
2. 填写完工单，单击“提交”按钮后，会创建一个工单实例。该能力通过调用BPM实现。

结合填单页面需求，以及前端组件对应关系，我们需要创建如下模型：

表 3-15 模型分析

模型名称	作用	来源	详细定义
basicInfo	保存工单的基本信息，将与“基础信息”、“其他信息”区域中各个组件绑定	自定义模型	<p>新增的字段名称请与下面字段名称保持一致，后续将绑定页面组件。包含的节点如下，这些节点与前台页面上的工单基本信息一一对应，加粗斜体部分要替换为自己账号对应的命名空间：</p> <ul style="list-style-type: none">• <i>HW_equipmentId_CST</i>: 设备ID，字段类型SingleSelect。• <i>HW_title_CST</i>: 工单标题，字段类型Text。• createdDate: 创建时间，字段类型DateTime。• <i>HW_type_CST</i>: 工单类型，字段类型SingleSelect。• <i>HW_priority_CST</i>: 工单优先级，字段类型SingleSelect。• <i>HW_description_CST</i>: 工单备注说明，字段类型TextArea。• <i>HW_faultPhenomenon_CST</i>: 故障现象说明，字段类型Text。• <i>HW_customer_CST</i>: 客户接口人，字段类型Text。• <i>HW_recoveryTime_CST</i>: 客户要求故障恢复时间，字段类型DateTime。• <i>HW_source_CST</i>: 工单来源，字段类型Text。• <i>HW_arriveTime_CST</i>: 客户要求工程师到达现场时间，字段类型DateTime。

模型名称	作用	来源	详细定义
equipmentInfo	获取设备详细信息，将与“设备信息”区域组件绑定	自定义模型	<p>将绑定页面组件。包含的计算节点如下，这些节点与前台页面上的设备信息一一对应，加粗斜体部分要替换为自己账号对应的命名空间：</p> <ul style="list-style-type: none"> • name: 设备名称，字段类型Text。 • <i>HW_equipmentSN_CST</i>: 设备编码，字段类型Text。 • <i>HW_equipmentBrand_CST</i>: 设备品牌，字段类型Text。 • <i>HW_equipmentModel_CST</i>: 设备型号，字段类型Text。 • <i>HW_fullAddress_CST</i>: 设备完整地址，字段类型Text。 <p>须知 这些节点名称需要与设备对象的字段名保持一致。</p>
equipmentList	查询系统中的设备，设备名称下拉框的属性值绑定模型	自定义模型	与“设备名称”下拉框绑定，将从下拉框的值赋给当前下拉框值。
equipmentOptions	查询系统中的设备，设备下拉框的可选项	服务模型	与公共接口equipmentSelectListQuery关联。
setInstanceId	查询系统中的设备，设备下拉框的可选项	服务模型	与公共接口createWorkOrder关联。

操作步骤

步骤1 进入之前创建的应用。

步骤2 在“WorkOrder”目录中，鼠标放在“Page”上，单击界面上出现的“+”，在弹出菜单中选择“标准页面”。

步骤3 在“标签”和“名称”文本框中输入“createWorkOrder”，单击“添加”。

当编辑已有标准页面时，为防止编辑时多人篡改，编辑前请单击进行锁定。

步骤4 在标准页面的“设计视图”下，单击页面底部的“模型视图”页签，切换到“模型视图”。

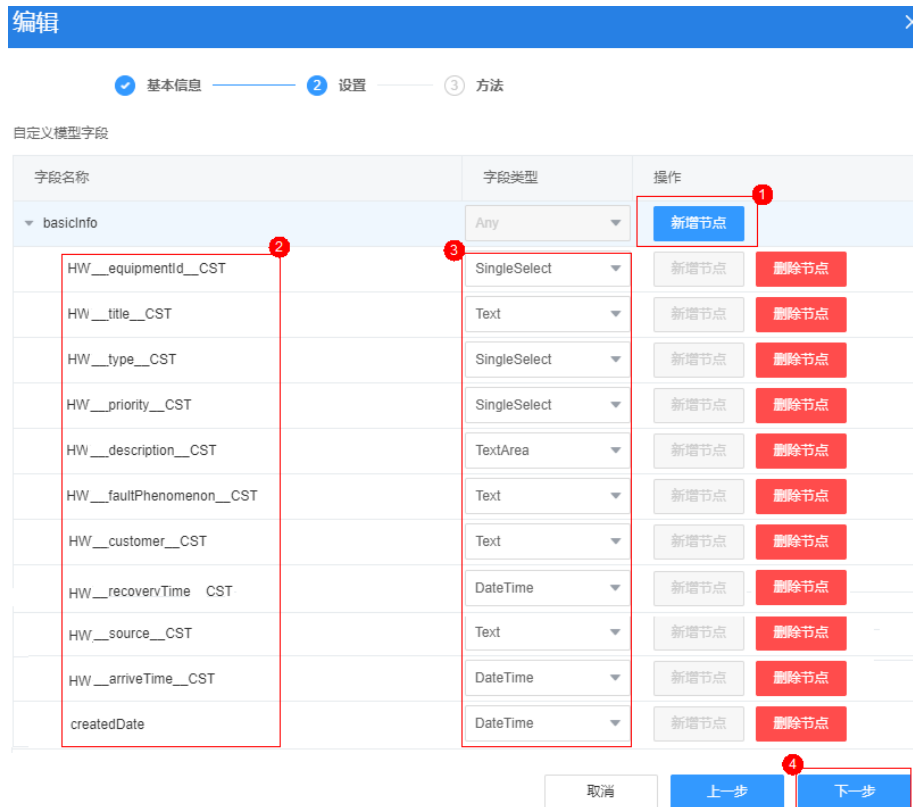
图 3-162 切换模型视图



步骤5 定义模型 “basicInfo”。

1. 在模型视图页面，单击“新增模型”。
2. 添加自定义模型，模型名称“basicInfo”，单击“下一步”。
3. 单击“新增节点”，新增设备ID字段“*HW__equipmentId__CST*”，并设置字段类型“SingleSelect”，加粗斜体内容请替换为实际命名空间前缀。
4. 依次增加表3-16中列出的字段名称，单击“下一步”，再单击“确定”。

图 3-163 新增节点字段




须知

加粗斜体内容请替换为实际命名空间前缀。


表 3-16 新增 basicInfo 模型的节点字段

字段名称	字段类型	字段描述
<i>HW__equipmentId__CST</i> (上一步已添加)	SingleSelect	设备ID
<i>HW__title__CST</i>	Text	工单标题
createDate	DateTime	创建时间
<i>HW__type__CST</i>	SingleSelect	工单类型

字段名称	字段类型	字段描述
<i>HW</i> _priority_CST	SingleSelect	工单优先级
<i>HW</i> _description_CST	TextArea	工单备注说明
<i>HW</i> _faultPhenomenon_CST	Text	故障现象说明
<i>HW</i> _customer_CST	Text	客户接口人
<i>HW</i> _recoveryTime_CST	DateTime	客户要求故障恢复时间
<i>HW</i> _source_CST	Text	工单来源
<i>HW</i> _arriveTime_CST	DateTime	客户要求工程师到达现场时间

5. 单击页面上方的, 保存设置。

步骤6 定义模型“equipmentList”。

1. 在模型视图页面, 单击“新增模型”。
2. 添加自定义模型, 模型名称“equipmentList”, 单击“下一步”。
3. 单击“下一步”, 再单击“确定”。
4. 单击页面上方的, 保存设置。

步骤7 定义模型“equipmentInfo”。


1. 在模型视图页面, 单击“新增模型”。
2. 添加自定义模型, 模型名称“equipmentInfo”, 单击“下一步”。
3. 单击“新增节点”, 依次新增表3-17中列出的字段名称, 单击“下一步”, 再单击“确定”。

须知


加粗斜体内容请替换为实际命名空间前缀。

表 3-17 新增 equipmentInfo 模型的节点字段


字段名称	字段类型	字段描述
name	Text	设备名称
<i>HW</i> _equipmentSN_CST	Text	设备编码
<i>HW</i> _equipmentBrand_CST	Text	设备品牌
<i>HW</i> _equipmentModel_CST	Text	设备型号
<i>HW</i> _fullAddress_CST	Text	设备完整地址

4. 单击页面上方的，保存设置。

步骤8 定义模型“equipmentOptions”。

1. 在模型视图页面，单击“新增模型”。
2. 添加服务模型，模型名称“equipmentOptions”，单击“下一步”。
3. 设置服务类型为“API”，“项目”选择“设备维修管理系统”，公共接口为即“equipmentSelectListQuery”，即创建“[查询设备列表](#)”脚本中定义脚本“equipmentSelectListQuery”对应的公共接口，单击“下一步”，再单击“确定”。
4. 单击页面上方的，保存设置。

步骤9 定义模型“setInstanceld”。

1. 在模型视图页面，单击“新增模型”。
2. 添加服务模型，模型名称“setInstanceld”，单击“下一步”。
3. 设置服务类型为“API”，“项目”选择“设备维修管理系统”，公共接口为“createWorkOrder”，即“生成工单”脚本“createWorkOrder”对应的公共接口，单击“下一步”，再单击“确定”。
4. 单击页面上方的，保存设置。

步骤10 拖拽页面框架组件（“基本信息”、“设备信息”、“其他信息”以及“按钮”区域）。

1. 单击页面底部的“设计视图”页签，切换到“设计视图”。
2. 从基本组件列表区拖拽“容器”到右侧“页面内容”中，然后再拖拽一个“表单”到“容器”中，并在“元数据表单配置向导”弹窗中，单击“取消”。
3. 拖拽一个“容器”到“表单”下方（**表单之外**），作为按钮区域的容器。

图 3-164 拖拽按钮区域容器



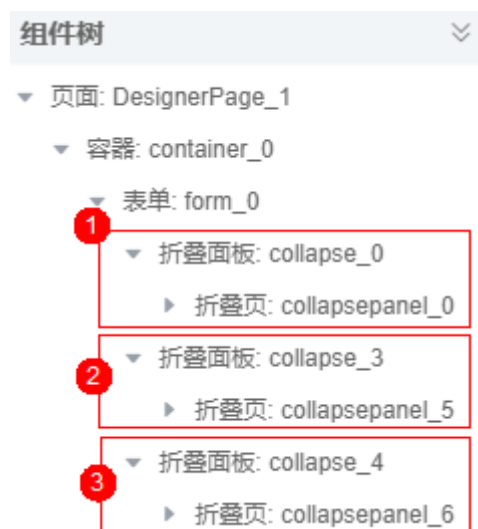
4. 选择“表单”，拖一个“折叠面板”组件到“表单”中，并页面右下角的组件树中，删除2个“折叠页”（在组件树中，选中后单击鼠标右键，再单击“删除”即可），删除后折叠面板包含1个折叠页。

图 3-165 组件层级关系及删除多余折叠页



5. 在“折叠面板”中，选中“折叠页签标题”，然后在右侧“属性”页签中进行以下设置：
 - 在“样式”下的“背景”，将背景颜色修改为浅蓝色（#9BCCEF）。
 - 选中“图标”，设置图标颜色为深蓝色。
 - 选中“标签”的“文本内容”为“基本信息”。您还可以根据需要，修改边距等其他样式，使其更美观。
6. 单击鼠标右键，并单击“复制”，再单击2次“粘贴”，在折叠面板之后，添加2个同样的折叠面板。


图 3-166 粘贴后的折叠面板



7. 分别设置第二、第三个折叠面板的“折叠页签标题”为“设备信息”、“其他信息”。

图 3-167 修改折叠页标题



8. 单击界面上方的 ，保存页面。

步骤11 组装“基本信息”区域。

1. 在“基本信息”所在的折叠面板下，从左侧组件列表中，拖一个“表单”到“折叠页签内容”中，然后在“元数据表单配置向导”弹窗中，单击“取消”。

图 3-168 拖入表单




2. 选中“表单”，在右侧属性中，单击 ，在模型选择弹窗中选中“basicInfo”，单击“确定”，并在提示弹窗中，单击“绑定并生成表单”。

图 3-169 表单数据绑定



3. 在生成的表单中，删除如表3-18中的多余字段所在组件以及按钮。

表 3-18 需要删除的组件名及组件

字段名	组件
<i>HW_faultPhenomenon_CST</i>	输入框
<i>HW_customer_CST</i>	输入框
<i>HW_recoveryTime_CST</i>	日期选择框
<i>HW_arriveTime_CST</i>	日期选择框
保存、取消	按钮


4. 分别选择表单中的2个“分栏”，在“行布局”中单击，修改分栏为4列，并按图3-171所示调整字段组件的位置（选中组件，用鼠标即可拖拽组件到目的栏），调整完成后，删除多余空白分栏。

图 3-170 调整分栏为 4 列

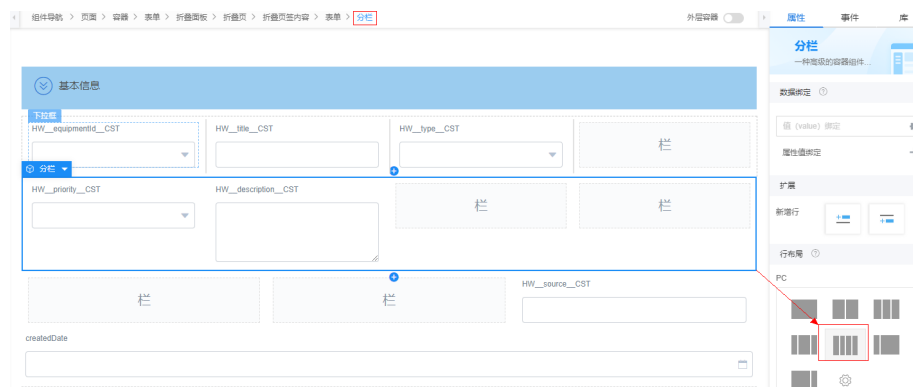
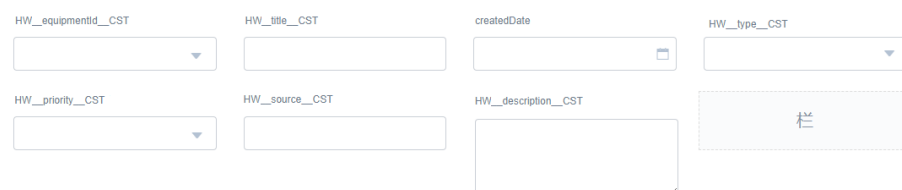


图 3-171 调整后表单



5. 选择表单中的各个组件，依次在右侧属性面板中修改各个组件标签，详细标签名如表3-19所示。

表 3-19 修改表单组件名及组件对应字段

标签名	字段名	组件类型
设备名称	<i>HW_equipmentId_CST</i>	下拉框
工单标题	<i>HW_title_CST</i>	输入框

标签名	字段名	组件类型
创建时间	createdDate	日期选择框
工单类型	HW_type_CST	下拉框
优先级	HW_priority_CST	下拉框
工单来源	HW_source_CST	输入框
备注	HW_description_CST	多行输入框

- 选中“设备名称”下拉框，在右侧属性页签，单击“属性值绑定”的“+”，“属性”设置为“选项”，然后再单击“模型字段”的设置图标，在弹窗中勾选“equipmentList”模型，对“设备名称”下拉框进行属性绑定。

图 3-172 添加属性值绑定



- 选中“工单类型”下拉框，在右侧属性页签中，设置“维修”和“保养”2个选项。


图 3-173 添加工单类型



- 选中“优先级”下拉框，在右侧属性页签中，设置“高”、“中”和“低”3个选项。

图 3-174 添加优先级



9. 单击界面上方的, 保存页面。

步骤12 组装“设备信息”区。


1. 在“设备信息”所在的“折叠面板”中，拖入一个“表单”组件到“折叠页签内容”中，并在“元数据表单配置向导”弹窗中，单击“取消”。
2. 选择“表单”，在右侧属性中，单击, 在弹窗中选中“equipmentInfo”，单击“确定”，并在提示中单击“绑定并生成表单”。

图 3-175 表单数据绑定




3. 删除“name”输入框、“保存”、“取消”按钮以及按钮所在“容器”。
4. 选择“分栏”，在“行布局”中单击, 修改分栏组件为4列，并删除空白分栏，调整后如图3-176所示。


图 3-176 调整后表单

HW_equipmentSN_CST HW_equipmentBrand_CST HW_equipmentModel_CST HW_fullAddress_CST

5. 修改各个输入框组件的标签名，并“公共”属性下，设置“只读”。

表 3-20 组件名及对应字段

标签名	字段名	其他属性
设备编码	HW_equipmentSN_CST	只读
设备品牌	HW_equipmentBrand_CST	只读
设备型号	HW_equipmentModel_CST	只读
地址	HW_fullAddress_CST	只读

6. 单击页面上方的，保存页面。

步骤13 组装“其他信息”区域，组装完成后如图3-178所示。

图 3-177 组件结构



图 3-178 其他信息区域

故障记录

故障现象说明

 栏 栏

时间记录

客户要求工程师到达现场时间 客户要求故障恢复时间 客户接口人

1. 在“其他信息”所在的“折叠面板”中，拖1个“表单”组件到“折叠页签内容”中，然后单击“取消”元数据表单配置向导弹窗。
2. 设置表单样式：选中“表单”组件，在“样式”下“高级设置”中，设置表单样式“:root{margin-left:50px;}”。

3. 在“表单”中，拖入1个“折叠面板”，并删除1个折叠页，保留2个折叠页。
4. 修改第1个折叠页“折叠页签标题”的“标签”下的“文本内容”修改为“故障记录”。
5. 在“故障记录”的“折叠页签内容”下，拖入1个分栏组件，设置为3列，并向第1列拖入“输入框”，并修改输入框的标签为“故障现象说明”。
6. 修改第2个折叠页“折叠页签标题”的“标签”下的“文本内容”修改为“时间记录”。
7. 在“时间记录”的“折叠页签内容”下，拖入1个分栏组件，并设置为3列，并分别向第1列和第2列拖入1个“日期选择框”，修改“标签”为“客户要求工程师到达现场时间”、“客户要求故障恢复时间”，然后修改“类型”为“日期时间”并设置“日期格式”为“yyyy-MM-dd HH:mm:ss”，向第3列拖入一个输入框，设置“标签”为“客户接口人”。

图 3-179 设置日期格式




8. 选择“故障现象说明”输入框，在右侧属性页签中，单击 ，为输入框绑定“basicInfo”下的“HW_faultPhenomenon_CST”字段。
9. 参考上一步，为其他组件绑定数据字段，组件名及对应字段，如表3-21所示。

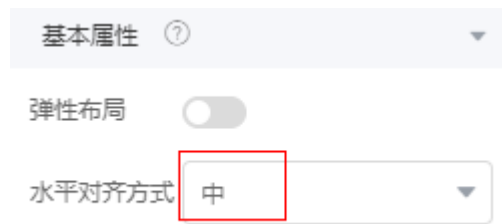
表 3-21 组件名及对应字段

标签名	绑定的字段名	组件类型
客户要求工程师到达现场时间	HW_arriveTime_CST	日期选择框
客户要求故障恢复时间	HW_recoveryTime_CST	日期选择框
客户接口人	HW_customer_CST	输入框


步骤14 组装“按钮”区域。

1. 选择页面最下方“容器”，设置“水平对齐方式”为“中”，并拖入2个“按钮”组件。

图 3-180 设置容器对齐方式



2. 修改第1个“按钮”的“显示名称”为“提交”，“类型”为“主要按钮”。
3. 修改第2个“按钮”的“显示名称”为“取消”。

步骤15 单击页面上方的, 保存页面。

----结束

添加页面事件

通过在“HW_createWorkOrder”页面上，定义页面事件，实现将工单信息存入到工单对象中。

实现“根据设备编码自动加载工单其他信息”

步骤1 在“我的应用”中，单击“设备维修管理系统”，进入应用。

步骤2 单击打开“HW_createWorkOrder”页面。

步骤3 添加页面事件代码。

1. 在“设计视图”中，选中最外层的“页面”，也可以直接单击组件导航，快速定位。
2. 在右侧“事件”页签中，单击“加载”后的“+”，为页面添加事件代码。
3. 在“添加动作”弹窗的“自定义动作”中，输入如下事件代码，然后单击底部“创建”。

```
$model.ref('equipmentOptions').setData({inputParam: {}});  
$model.ref('equipmentOptions').run().then(function(response){  
  if(response.equipList){  
    $model.ref("equipmentList").setData(response.equipList);  
  }  
});
```

4. 单击页面上方的, 并退出事件编排。

----结束

实现“显示设备详细信息”

通过定义“基本信息”区域“设备名称”下拉框的“数据改变”事件，可以实现根据所选设备，在“设备详情”区域显示设备信息。

步骤1 在“HW_createWorkOrder”页面中，选中“设备名称”下拉框。

图 3-181 添加“数据改变”事件代码



步骤2 在右侧“事件”页签中，单击“数据改变”后的“+”。

步骤3 在“添加动作”弹窗的“自定义动作”中，输入如下事件代码，然后单击底部“创建”。

须知

脚本中加粗斜体内容请替换为实际的命名空间前缀。

```
let equipment = $model.ref("basicInfo").getData();
let equipmentId = equipment.HW_equipmentId__CST;
let queryEquipParam = {"id": equipmentId};
$model.ref('equipmentOptions').setData({inputParam: queryEquipParam});
$model.ref('equipmentOptions').run().then(function (response) {
  if (response.equipment) {
    $model.ref("equipmentInfo").setData(response.equipment);
  }
});
```

步骤4 单击页面上方的, 并退出事件编排。

----结束

实现“生成工单”

通过定义“提交”按钮的“点击”事件，实现生成工单的能力。定义“取消”按钮的“点击”事件，返回工单列表页面（客服人员）。

步骤1 定义“提交”按钮“点击”事件。

1. 在“***HW***_createWorkOrder”页面的“设计视图”下，选中“提交”按钮。
2. 在右侧“事件”页签中，单击“点击”后的“+”。
3. 在“添加动作”弹窗的“自定义动作”中，输入如下事件代码，然后单击底部“创建”。

须知

脚本中加粗斜体内容请替换为实际的命名空间前缀。

```
// 配置页面的bpm参数bp.name，通过submitTask方法启动BPM并提交工单数据workOrderData到BPM
context.$page.params["bp.name"] = "HW_WorkOrderBpm";
let basicInfo = $model.get('basicInfo').getData();
basicInfo.HW_workOrderId__CST = basicInfo.HW_type__CST + "_" + new Date().getTime();
let workOrderData = {
```

```
"workOrderData": basicInfo
};
context.$bp.submitTask(workOrderData).then(function (resp) {
  context.$page.loadStdPage('HW_workOrderList');
});
```

4. 单击页面上方的, 并退出事件编排。

步骤2 定义“取消”按钮的“点击”事件。

1. 在“设计视图”中, 选中“取消”按钮。
2. 参考“提交”按钮, 给“取消”按钮定义以下代码事件


```
//返回工单列表页面
context.$page.loadStdPage('HW_workOrderList');
```

3. 单击页面上方的, 并退出事件编排。

----结束

验证

页面组装及事件添加完成后, 需要在预览页面, 初步检查页面预览效果, 验证页面的数据绑定及事件代码正确性。当前工单流转相关的BPM及工单列表尚未创建, 因此暂不验证“保存”、“取消”按钮。

步骤1 在“HW_createWorkOrder”页面中, 单击界面上方的, 进入预览页面。

系统会弹出“HW_createWorkOrder”预览页面。

步骤2 在“基本信息”区域的“设备名称”下拉框中, 选中一个设备名称, 检查“设备详情”区域是否显示了对应的设备详情。显示正确, 则说明组件与模型的绑定, 以及下拉框事件执行正确。如不显示, 则需要检查相关事件代码。

步骤3 检查“工单类型”、“优先级”下拉框中, 是否有值, 显示正确则说明下拉框属性设置正确。

----结束

3.1.7.3.3 开发“客服人员查询工单”功能

创建“查询工单”脚本及公共接口

本节中的后台逻辑主要是通过脚本的形式, 实现对工单记录的查询, 然后将查询脚本封装成一个公共接口, 供页面调用。

工单列表(客服人员)页面中, 为了实现查询并显示已有工单信息功能, 需要开发“查询工单”脚本和对应公共接口。


创建“查询工单”脚本

步骤1 在“我的应用”中, 单击“设备维修管理系统”, 进入应用。

步骤2 在“WorkOrder”目录中, 将鼠标放在“Script”上, 单击界面上出现的“+”, 在弹出菜单中选择“脚本”。

步骤3 在弹窗中, 选中“创建一个新脚本”, 在“名称”文本框中输入“queryWorkOrder”, 单击“添加”。

系统实际创建的脚本名称为“HW_queryWorkOrder”，“HW_”前缀由租户命名空间namespace决定。新建创建的脚本，默认是当前用户锁定状态，可以进行编辑保存等操作。

当编辑已有脚本时，为防止编辑时多人篡改，编辑前请单击进行锁定。

步骤4 在代码编辑器中插入如下脚本代码。

须知

脚本中**红色内容**请替换为实际的对象名、字段名。

```
import * as context from 'context';
import * as db from 'db';
import * as decimal from 'decimal';//导入decimal数据类型相关的标准库

@useObject(['HW_WorkOrder_CST'])


@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'String' })
  title: string;
  @action.param({ type: 'String' })
  status: string;
  @action.param({ type: 'String' })
  createdBy: string;
  @action.param({ type: 'Boolean' })
  isFME: boolean;
  @action.param({ type: 'Number', min: 0 })
  start: decimal.Decimal;//分页信息，表示从第几条数据开始查询
  @action.param({ type: 'Number', min: 0 })
  limit: decimal.Decimal;//分页信息，表示一次查询几条数据
}

@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'Any', label: 'object', isCollection: true })
  workOrderList: object[];
  @action.param({ type: 'String' })
  total: string;//总共查到几条数据
}

@action.object({ type: "method" })
export class QueryWorkOrder {
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public queryWorkOrder(input: ActionInput): ActionOutput {
    let out = new ActionOutput(); //新建出参ActionOutput类型的实例，作为返回值
    let error = new Error(); //新建错误类型的实例，用于在发生错误时保存错误信息
    try {
      let s = db.object('HW_WorkOrder_CST'); //获取HW_WorkOrder_CST这个Object的操作实例

      //condition是db标准库queryByCondition方法的入参（查询条件）
      let condition = {
        "conjunction": "AND",
        "conditions": []
      };
      //基本查询条件
      condition.conditions.push({
        "field": "id",
        "operator": "isnotnull",
      });
      condition.conditions.push({
        "field": "id",
        "operator": "ne",
        "value": ""
      });
    }
  }
}
```

```
});  
//按title查询  
if (input.title && input.title != "") {  
  condition.conditions.push({  
    "field": "HW__title__CST",//与对象中的字段名保持一致  
    "operator": "contains",  
    "value": input.title  
  });  
}  
//按status查询  
if (input.status && input.status != "") {  
  condition.conditions.push({  
    "field": "HW__status__CST",//与对象中的字段名保持一致  
    "operator": "eq",  
    "value": input.status  
  });  
}  
//按title查询  
if (input.createdBy && input.createdBy != "") {  
  condition.conditions.push({  
    "field": "createdBy",//与对象中的字段名保持一致  
    "operator": "eq",  
    "value": input.createdBy  
  });  
}  
if (input.isFME) {  
  condition.conditions.push({  
    "field": "HW__assignedFme__CST",  
    "operator": "eq",  
    "value": context.getUserName()  
  });  
}  
let option = {  
  "options": {  
    "orderby": [{  
      "field": "createdDate",  
      "order": "desc"  
    }]  
  }  
};  
//如果有分页  
if (input.start && input.limit) {  
  let start = decimal.toNumber(input.start);//将decimal类型转换为接口需要的number类型  
  let limit = decimal.toNumber(input.limit);  
  option.options['limit'] = limit;  
  option.options['skip'] = start;  
}  
let workOrderList = s.queryByCondition(condition, option);  
for (let i in workOrderList) {  
  if (workOrderList[i].HW__status__CST == "关闭") {  
    workOrderList[i].isDeal = true;  
  }  
}  
out.workOrderList = workOrderList;  
//调用查询符合condition条件的数据总数的接口  
out.total = s.count(condition) + "";  
} catch (error) {  
  console.error(error.name, error.message);  
  context.setError(error.name, error.message);  
}  
return out;  
}
```

步骤5 单击编辑器上方的，保存图标。

步骤6 测试脚本能否正常执行。




1. 单击编辑器上方的 ，执行脚本。
2. 在界面底部，查询脚本可以不提供输入参数，直接单击测试窗口右上角  执行图标。
执行成功，会在“输出”页签返回查询结果。

图 3-182 输出结果示例

```

{
  "total": "2",
  "workOrderList": [
    {
      "createdBy": "10gd000000bcbTj7KT6e",
      "createdBy.__objectType": "User",
      "createdBy.name": "1",
      "createdDate": "2020-09-27 15:56:45",
      "currencyIsoCode": "USD",
      "id": "cwsC000000dbIvrNqsVs",
      "installedPackage": null,
      "lastModifiedBy": "10gd000000bcbTj7KT6e",
      "lastModifiedBy.__objectType": "User",
      "lastModifiedBy.name": "3",
      "lastModifiedDate": "2020-09-29 11:21:54",
      "name": "1",
      "owner": "10gd000000bcbTj7KT6e",
      "owner.__objectType": "User",
      "owner.name": "1",
      "assignedTime_CST": "2020-09-26 12:00:00",
      "assignedTime_CST": "派单员"
    }
  ]
}

```

步骤7 测试成功，单击编辑器上方的 ，启用发布脚本。

---结束

创建“查询工单”公共接口

步骤1 在APP视图下，单击下方“服务”，进入公共接口创建页面。

步骤2 单击“新建”，创建“查询工单”脚本“*HW_queryWorkOrder*”的公共接口。

设置接口参数信息：操作名称为“*queryWorkOrder*”，版本为“1.0.0”，URL为“/*queryWorkOrder*”，“类型”选择“脚本”，“资源”为“*HW_queryWorkOrder*”，方法为“POST”，然后单击“保存”。

图 3-183 新建 queryWorkOrder 接口

The screenshot shows a form for creating a new API endpoint. The fields are as follows:

- 标签 (Label): queryWorkOrder
- * 操作名称 (Operation Name): queryWorkOrder
- * 版本 (Version): 1.0.0
- * URL: /service/..._MyApp/1.0.0 /queryWorkOrder
- 内容类型 (Content Type): application/json
- 分类 (Category): 请输入
- 描述 (Description): 请输入
- 允许匿名访问
- 类型 (Type): 脚本 (Script), 服务编排 (Service编排), 对象 (Object)
- 自定义响应
- * 资源 (Resource): ..._queryWorkOrder
- * 方法 (Method): POST
- Buttons: 保存 (Save), 取消 (Cancel)

----结束

组装“工单列表（客服人员）”页面

“工单列表（客服人员）”页面与查询设备列表页面类似，包含工单查询条件、工单列表，以及创建工单的入口，用标准页面功能实现。

如图3-184所示，工单列表（客服人员）页面主要是一个表格、“创建工单”按钮及表格查询区域构成。

图 3-184 工单列表（客服人员）

The screenshot shows the '现场工单列表' (On-site Work Order List) page. It includes search filters for '工单标题' (Work Order Title) and '状态' (Status), a '搜索' (Search) button, a '+ 创建工单' (Create Work Order) button, and a table of work orders.

<input type="checkbox"/>	#	工单id	优先级	工单标题	状态	当前处理人	创建时间	创建人
<input type="checkbox"/>	1	维修_1600074245347	高	测试	待接单	testms	2020-09-14 17:04:05	hw57i
<input type="checkbox"/>	2	维修_1600070558989	高	电梯开门延时	处理中	hw57i	2020-09-14 16:02:39	hw57i
<input type="checkbox"/>	3	维修_1599618263866	高	test	关闭	hw57i	2020-09-09 10:24:24	hw57i
<input type="checkbox"/>	4	维修_1599614211902	高	test909	处理中	hw57i	2020-09-09 09:16:53	hw57i

操作步骤


步骤1 在“我的应用”中，单击“设备维修管理系统”，进入应用。

步骤2 在“WorkOrder”中，在鼠标放在“Page”上，单击界面上出现的“+”，在弹出菜单中选择“标准页面”。

步骤3 在“标签”和“名称”文本框中输入“workOrderList”，单击“添加”。

当编辑已有标准页面时，为防止编辑时多人篡改，编辑前请单击进行锁定。

步骤4 定义模型“queryWorkOrder”。

1. 在“模型视图”中，单击“新增模型”。
2. 添加服务模型，模型名称“queryWorkOrder”，单击“下一步”。
3. “选择服务类型”为“公共接口”，“项目”为“设备维修管理系统”，“选择服务”为“queryWorkOrder”，单击“下一步”，再单击“确定”。
4. 单击页面上方的，保存模型。


步骤5 定义模型“workOrderInstance”。

1. 在“模型视图”中，单击“新增模型”。
2. 来源选择“自定义”，模型名称“workOrderInstance”，单击“下一步”。
3. 单击模型名后的“新增节点”，新增表3-22所示的节点，然后单击“下一步”，再单击“确定”。

以下字段要与“工单对象”中的字段一致，其中加粗斜体内容部分要替换为当前账号对应的命名空间前缀。

表 3-22 新增节点


字段名称	字段类型	字段描述
id	Text	记录ID
<i>HW_workOrderId_CST</i>	Text	工单Id
<i>HW_priority_CST</i>	Text	优先级
<i>HW_title_CST</i>	Text	工单标题
<i>HW_status_CST</i>	Text	状态
<i>HW_assignedFme_CST</i>	Text	当前处理人
createdDate	日期/时间	创单时间
<i>HW_createdBy_CST</i>	Text	创建人

4. 单击页面上方的，保存模型。

步骤6 单击“设计视图”，从“模型视图”切换到“设计视图”。

步骤7 拖拽页面标题。

1. 从基本组件列表区拖拽一个“标题”组件到“页面内容”中。

2. 在右侧“属性”页签中，将“标题内容”修改为“现场工单列表”。
3. 单击页面上方的，保存页面标题。

步骤8 从组件列表区拖拽一个“表格”组件到“标题”的下方。

步骤9 绑定表格数据模型，并设置查询结果区域。


1. 在“设计视图”中，选中“表格”，单击右侧“属性”页签，如图3-185所示。

图 3-185 为表格绑定数据模型




2. 在“选择模型”对话框中，选中“workOrderInstance”模型，单击“确定”。绑定模型后，系统自动将模型的所有字段添加为查询结果列。
3. 单击上图中每个字段后的，修改列标题等字段属性。
以“工单ID”为例，其属性配置如图3-186所示，各个字段及修改后的列名如表3-23所示。

图 3-186 设置“工单 ID”列的属性

属性配置

基本属性

字段名:

列标题:

列标题提示:

列标题自定义渲染:

功能

固定:

隐藏:

溢出省略:

排序:

筛选:

单元格可编辑:

显示类型:

样式


列

样式类:

表 3-23 修改各字段的显示属性

字段名	列标题	备注
id	记录ID	启用“隐藏”属性，即设置为true
HW_workOrderId_CST	工单Id	-

字段名	列标题	备注
HW_priority_CST	优先级	-
HW_title_CST	工单标题	-
HW_status_CST	状态	-
HW_assignedFme_CST	当前处理人	-
createdDate	创单时间	-
HW_createdBy_CST	创建人	-


4. 单击页面上方的，保存设置。

步骤10 添加“创建工单”按钮。


1. 选中“表格”，单击右侧“属性”页签中“表格区块”中“工具栏”后的“添加”按钮。

图 3-187 增加查询条件区域



2. 删除工具栏中多余的按钮，只保留“新增行”。
3. 修改“新增行”的“显示名称”为“创建工单”。
4. 单击页面上方的，保存设置。

步骤11 创建查询条件区域。

1. 单击“模型视图”，切换到模型视图，在表格模型“table_0_condition”后，单击编辑图标.

说明

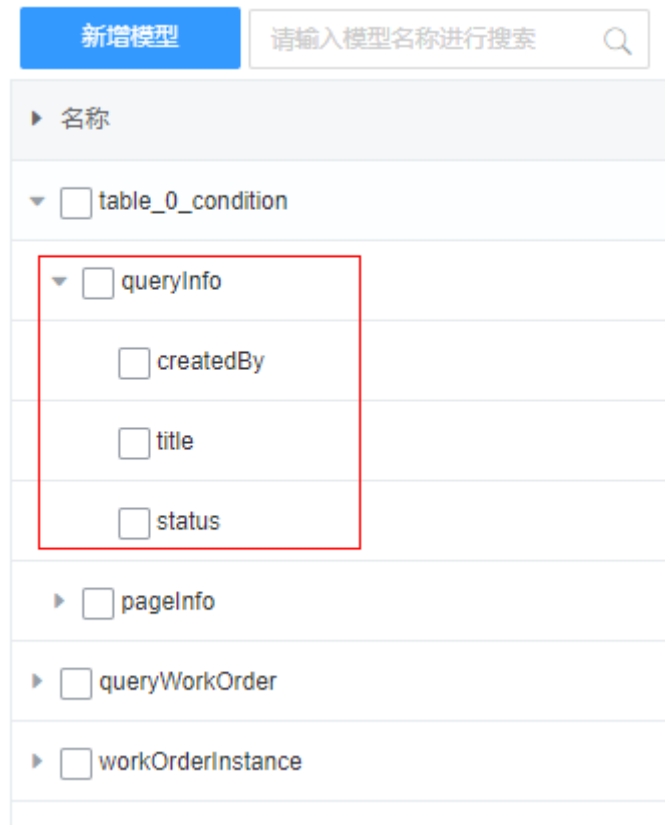
表格组件拖入后，页面将自动生成一个表格模型。“table_0_condition”即是在前面拖入的表格对应的表格模型，模型名可能跟拖入顺序及个数有关，一般情况下模型名称为“table_0_condition”，如果名称不一致，请根据实际情况修改。

图 3-188 编辑表格模型



2. 在表格模型“table_0_condition”中，单击“新增节点”，添加一个“queryInfo”节点，“数据类型”为“Any”，然后在“queryInfo”下，再单击“新增节点”添加3个子节点“createdBy”、“title”、“status”。

图 3-189 创建完成后的模型节点





3. 单击页面上方的 ，保存设置。
4. 单击“设计视图”，切换到设计视图，从左侧组件列表中，拖一个“表单”组件到“表格容器”的最上部（表格容器内部最上部），在“元数据表单配置向导”弹窗中，单击“取消”。

图 3-190 向表格容器拖入表单



- 选中表单，在属性中“数据绑定”下，单击 ，为表单绑定“table_0_condition”中的“queryInfo”节点，并在提示弹窗中，单击“绑定并生成表单”，单击“确定”。

提示



是否根据绑定模型自动生成表单（如果表单已经生成，会覆盖已有的内容）

只绑定模型

绑定并生成表单

- 在表单中，分别选中并删除“createdBy”、“重置”按钮。
- 调整“title”、“status”、“保存”为1行，并删除多余分栏及容器，调整后，如图3-191所示。

图 3-191 调整后表单




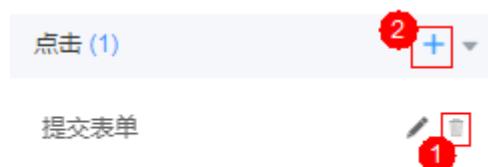
- 修改表单中输入框属性。
 - 修改“title”的“标签”为“工单标题”。
 - 修改“status”的“标签”修改为“状态”。
- 选中“保存”按钮，修改“显示名称”为“搜索”，设置单击“事件”页签，再单击 ，删除“提交表单”事件，然后单击“点击”后的“+”，进入事件编排页面，在“自定义动作”中，输入以下事件代码，单击“创建”。

图 3-192 修改按钮事件代码



须知

脚本中**红色内容**请替换为实际的对象名、字段名、页面名、组件ID。其中“table_0_condition”为当前表格组件的ID号，中间数字默认为“0”，如果有多次修改或创建该ID号会变化，选中表格后，在组件树上可以查看实际组件ID。

```
let pageInfo = $model.ref('table_0_condition').getData().pageInfo;
let queryInfo = $model.ref('table_0_condition').getData().queryInfo;
if (!queryInfo) {
  queryInfo = {};
}
let queryData = {
  "title": queryInfo.title,
  "status": queryInfo.status,
  "createdBy": queryInfo.createdBy,
  "start": 0,
  "limit": pageInfo.pageSize
};
$model.ref('queryWorkOrder').setData({ inputParam: queryData });
$model.ref('queryWorkOrder').run().then(function (data) {
  $model.ref('workOrderInstance').setData(data.workOrderList);
  pageInfo.total = parseInt(data.total);
  $model.ref('table_0_condition').setData({ "pageInfo": pageInfo });
}).catch(function (error) {
  console.log('error is', error);
});
```

10. 选中“搜索”按钮所在的栏，在右侧“样式”下的“布局”中，设置上方外边距为“30”，使“搜索”按钮上下居中显示。


图 3-193 设置搜索按钮布局样式（位置）



查询条件区域创建后，如图3-194所示。

图 3-194 查询区域



11. 单击页面上方的, 保存设置。


步骤12 添加页面事件代码。

1. 在“设计视图”中选中最外层的“页面”，在右侧“事件”页签中，单击“加载”后的“+”，进入编辑动作页面。
2. 在“自定义动作”中，输入如下脚本代码。

```
let pageInfo = $model.ref('table_0_condition').getData().pageInfo;
let queryData = {
  "start": (pageInfo.curPage - 1) * pageInfo.pageSize,
  "limit": pageInfo.pageSize
};
$model.ref('queryWorkOrder').setData({ inputParam: queryData });
$model.ref('queryWorkOrder').run().then(function (data) {
  $model.ref('workOrderInstance').setData(data.workOrderList);
  pageInfo.total = parseInt(data.total);
  $model.ref('table_0_condition').setData({ "pageInfo": pageInfo });
}).catch(function (error) {
  console.log('error is', error);
});
```

3. 单击页面上方的, 并退出事件编排。

步骤13 添加“创建工单”按钮跳转事件，跳转事件功能是将页面跳转到“生成工单”页面。

1. 选中“创建工单”按钮，在右侧“事件”页签中，单击“新增行”后的, 进入事件编排页面。
2. 在“自定义动作”中，输入以下事件代码，单击“保存”。

```
//打开创建工单页面，需要根据实际页面名称修改
context.$page.load('/besBaas/page#/HW_createWorkOrder')
```


3. 单击页面上方的, 并退出事件编排。

----结束

3.1.7.3.4 验证

“工单列表（客服人员）”和“生成工单”页面开发完成后，需要验证两个页面间的关联跳转事件，以及相关页面布局样式。

操作步骤

步骤1 在“HW_workOrderList”页面中，单击界面上方预览图标。

系统会弹出预览页面。

步骤2 查看页面中页面布局、样式是否符合预期。

步骤3 查看当前表格中的工单记录。当前表格中仅有一条测试数据，此数据是在[生成工单](#)脚本中输入的一条测试数据。

图 3-195 查看工单记录

现场工单列表

<input type="checkbox"/>	#	工单ID	优先级	工单标题	状态	当前处理人	创建时间	创建人
<input type="checkbox"/>	1	WD0000123456	高	电梯无法关门	待派单	派单员	2020-09-30 16:...	h...

步骤4 验证搜索功能。

1. 在“工单标题”中，输入“电梯维修”，进行搜索，因为当前没有该标题的工单，因此搜索结果为空，然后再输入“电梯无法关门”，如果显示该工单记录，则说明搜索功能正确。
2. 在“状态”中输入“待处理”，然后单击“搜索”，当前暂时没有“待处理”工单，因此搜索结果为空，再输入“待派单”后，单击“搜索”，则显示已存在工单，则说明搜索功能正常。

步骤5 验证页面跳转事件。

单击“创建工单”，验证是否跳转到“生成工单”页面。如果页面跳转到工单列表页面，则说明验证成功。

因当前生成工单页面相关的BPM尚未创建，这里还无法验证创建工单功能。

----结束

3.1.7.4 开发“派单员派发工单”功能

“派单员派发工单”功能包含工单列表（派单员）页面、派单对话框两个页面。

学习地图

如[图3-196](#)所示，通过本章的学习和实践，您将进一步了解“标准页面”的能力，包括：

- 弹出对话框
- 标准页面模板
- 脚本

图 3-196 学习地图



3.1.7.4.1 创建“查询维修人员”脚本


当派单员选择派单的下一步接单人时，需要先查询出，维修人员的列表，然后才能将流程走到维修人员的名下，因此我们需要创建一个“查询维修人员”的脚本逻辑。

操作步骤

步骤1 在“我的应用”中，单击“设备维修管理系统”，进入应用。

步骤2 在“WorkOrder”目录中，将鼠标放在“Script”上，单击界面上出现的“+”，在弹出菜单中选择“脚本”。

步骤3 在弹窗中，选中“创建一个新脚本”，在“名称”文本框中输入“queryWorker”，单击“添加”。

当编辑已有脚本时，为防止编辑时多人篡改，编辑前请单击进行锁定。

步骤4 在代码编辑器中插入如下脚本代码。

```
import * as context from 'context';
import * as db from 'db';

//使用数据对象PortalUser(业务用户)
@useObject(['PortalUser'])

@action.object({ type: "param" })
export class ActionInput {
}

@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'Any', label: 'object', isCollection: true })
  userList: object[];
}

@action.object({ type: "method" })
export class QueryWorker {
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public queryWorker(input: ActionInput): ActionOutput {
    let out = new ActionOutput();
    let portalUserObject = db.object('PortalUser');

    //查询用户类型为ms（维修人员）的用户。（注册用户的脚本中设置了用户类型）
    let portalUsers = portalUserObject.queryByCondition({
      "conjunction": "AND",
      "conditions": [{
        "field": "userType",
```


```
        "operator": "eq",
        "value": "ms"
    }]
});

//将查询结果转换为选项列表的形式（id和name）。
let selectValue = portalUsers.map(function (v, i, a) {
    return {
        'value': {
            "id": v['id'],
            "name": v['usrName']
        },
        'display': v['usrName']
    }
});



//当前登录的用户为平台用户（非业务用户），则加入到维修人员列表中，用于测试。
if (context.getUserType() == context.UserType.User) {
    selectValue.push({
        'value': {
            "id": context.getUserId(),
            "name": context.getUserName()
        },
        'display': context.getUserName()
    });
}

out.userList = selectValue;

return out;
}
```

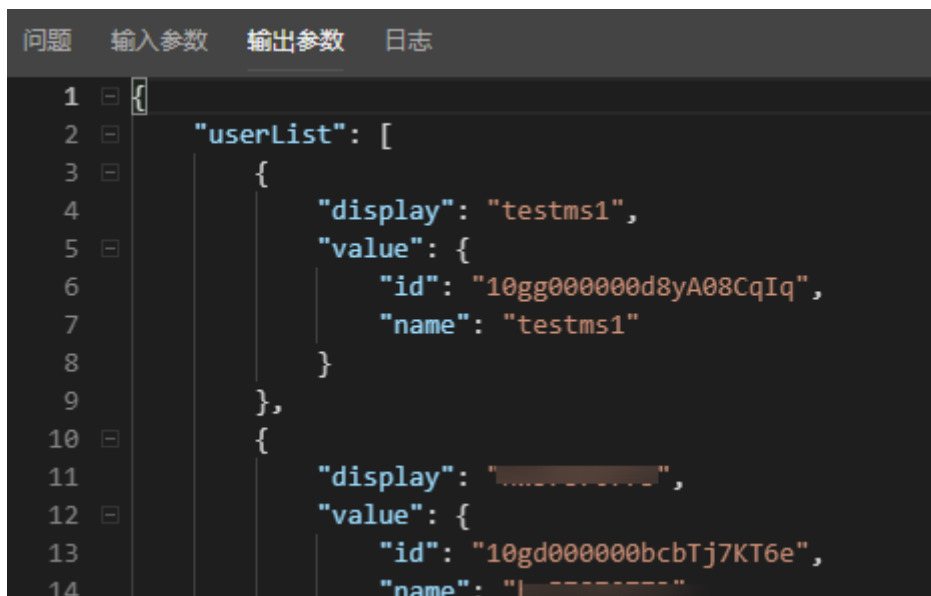
步骤5 单击编辑器上方的, 保存脚本。

步骤6 测试脚本能否正常执行。


1. 单击编辑器上方的, 执行脚本。
2. 在界面底部, 单击测试窗口右上角执行图标。

执行成功, 会在“输出”页签返回查询结果, 请记录“name”值, 此参数在[创建“派单功能”脚本](#)会作为输入参数使用。

图 3-197 输出查询出的维修人员信息



```
1  {
2    "userList": [
3      {
4        "display": "testms1",
5        "value": {
6          "id": "10gg00000d8yA08CqIq",
7          "name": "testms1"
8        }
9      },
10     {
11       "display": "testms2",
12       "value": {
13         "id": "10gd000000bcbTj7KT6e",
14         "name": "testms2"
15       }
16     }
17   ]
18 }
```

步骤7 测试成功，单击编辑器上方的，启用发布脚本。

----结束

3.1.7.4.2 创建“派单功能”脚本

当派单员选择派单的下一步接单人时，需要修改工单状态及处理人，因此我们需要创建一个实现“派单功能”的脚本逻辑，该脚本逻辑不是页面直接调用，而是通过后端 BPM 调用运行的。

操作步骤

步骤1 在“我的应用”中，单击“设备维修管理系统”，进入应用。

步骤2 在“WorkOrder”目录中，将鼠标放在“Script”上，单击界面上出现的“+”，在弹出菜单中选择“脚本”。

步骤3 在弹窗中，选中“创建一个新脚本”，在“名称”文本框中输入“dispatchWorkOrder”，单击“添加”。

当编辑已有脚本时，为防止编辑时多人篡改，编辑前请单击进行锁定。

步骤4 在代码编辑器中，插入如下脚本代码。


须知

脚本中红色内容请替换为实际的对象名、字段名。



```
//本脚本用于派发工单
import * as db from 'db';//导入处理object相关的标准库
import * as context from 'context';//导入上下文相关的标准库
import * as date from 'date';
//定义入参结构，入参包含1个参数：工单状态的修改信息，为必填字段
@action.object({ type: "param" })
export class ActionInput {
```

```
@action.param({ type: 'Any', required: true, label: 'object' })
transInfo: any;
}
//定义出参结构，出参包含2个参数，workOrder的记录id和派发后的责任人
@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'String' })
  id: string;
  @action.param({ type: 'String' })
  assignedFme: string;
}
//使用数据对象HW_WorkOrder_CST
@useObject(['HW_WorkOrder_CST'])
@action.object({ type: "method" })
export class DispatchWorkOrder { //定义接口类，接口的入参为ActionInput，出参为ActionOutput
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public dispatchWorkOrder(input: ActionInput): ActionOutput {
    let out = new ActionOutput(); //新建出参ActionOutput类型的实例，作为返回值
    let error = new Error(); //新建错误类型的实例，用于在发生错误时保存错误信息
    try {
      let transInfo = input.transInfo; //将入参赋值给transInfo变量，方便后面使用
      let s = db.object('HW_WorkOrder_CST'); //获取HW_WorkOrder_CST这个Object的操作实例

      //查询条件
      let condition = {
        "conjunction": "AND",
        "conditions": [{
          "field": "HW_workOrderId_CST",
          "operator": "eq",
          "value": transInfo['HW_workOrderId_CST']
        }]
      };
      //查找workOrderId所代表的工单信息
      let workOrder = s.queryByCondition(condition);
      workOrder[0]['HW_status_CST'] = '待接单';
      workOrder[0]['HW_assignedFme_CST'] = transInfo["HW_assignedFme_CST"].name;
      let isUpdated = s.updateByCondition(condition, workOrder[0]);
      if (isUpdated) {
        out.id = workOrder[0]['id'];
        out.assignedFme = transInfo["HW_assignedFme_CST"].id;
      } else {
        error.name = "WOERROR";
        error.message = "派发工单失败! ";
        throw error;
      }
    } catch (error) {
      console.error(error.name, error.message);
      context.setError(error.name, error.message);
    }
    return out;
  }
}
```

步骤5 单击编辑器上方的，保存脚本。

步骤6 测试脚本能否正常执行。

1. 单击编辑器上方的，执行脚本。
2. 在界面底部输入测试数据，单击测试窗口右上角执行图标。

脚本中加粗斜体内容请替换为实际的对象名、字段名。"name": "***test***"为当前租户名，可以在上一节的**输出参数**中查询。

```
{
  "transInfo": {
    "HW_workOrderId_CST": "WD0000123456",
    "HW_assignedFme_CST": {
      "id": "",


```

```
"name": "test"  
  }  
}
```

执行成功，会在“输出”页签返回查询结果。

图 3-198 返回结果



步骤7 测试成功，单击编辑器上方的，启用发布脚本。

----结束

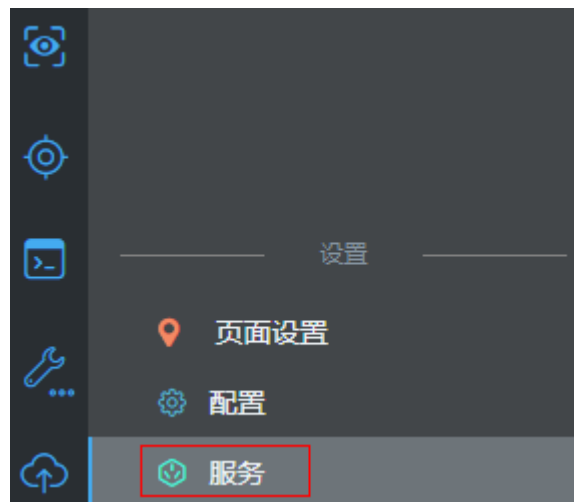
3.1.7.4.3 创建公共接口

参考之前创建公共接口的步骤，创建“查询维修人员”脚本对应的公共接口。

操作步骤

步骤1 在设计视图下，单击下方的“服务”，进入公共接口创建页面。

图 3-199 创建公共接口入口



步骤2 在公共接口页面，单击“新建”。

图 3-200 公共接口创建

公共接口

使用公共接口，可以将flow的URL映射到发布到外部网关，第三方可以通过OAuth2.0调用。



步骤3 创建“查询维修人员”、“派单功能”脚本对应的公共接口，详细接口信息如表3-24所示。

说明

加粗斜体内容请替换为实际命名空间前缀。

表 3-24 公共接口

设置操作	版本	URL	方法	类型	资源
queryWorker	1.0.0	/queryWorker	GET	脚本	<i>HW_queryWorker</i>
dispatchWorkOrder	1.0.0	/dispatchWorkOrder	POST	脚本	<i>HW_dispatchWorkOrder</i>

----结束

3.1.7.4.4 组装“派单”对话框

“派单”对话框负责指定下一环节的工单状态和处理责任人，用标准页面功能实现。

派单界面的大致构想如图3-201所示，其中“工单ID”是为了传递维修工单信息。

图 3-201 派单员派单对话框

页面模型负责与页面组件交互，传递派单需要的工单ID、工单状态、下一环节处理人等信息。结合页面需求，可知需要创建如下模型：

表 3-25 模型分析

模型名称	作用	详细定义
transInfo	保存派单的参数。	自定义模型，包含的计算节点如下，这些节点与派单接口的输入参数名称一一对应。 <ul style="list-style-type: none"> • <i>HW_workOrderId_CST</i>: 工单 ID • <i>HW_assignedFme_CST</i>: 下一环节处理人 说明 加粗斜体内容请替换为实际命名空间前缀。
workerOptions	查询系统中的工程师信息，下一环节处理的可选项。	服务模型，与公共接口queryWorker关联。

操作步骤


步骤1 在“我的应用”中，单击“设备维修管理系统”，进入应用。

步骤2 在“WorkOrder”中，鼠标放在“Page”上，单击界面上出现的“+”，在弹出菜单中选择“标准页面”。


步骤3 在“标签”和“名称”文本框中输入“workOrderDispatch”，单击“添加”。

当编辑已有标准页面时，为防止编辑时多人篡改，编辑前请单击进行锁定。

步骤4 定义模型“transInfo”。

1. 在“模型视图”中，单击“新增模型”。
2. 添加自定义模型，模型名称“transInfo”，单击“下一步”。
3. 为模型添加节点“*HW_workOrderId_CST*”（字段类型Text）、“*HW_assignedFme_CST*”（字段类型Any），单击“下一步”，再单击“确定”，加粗斜体内容请替换为实际命名空间前缀。
4. 单击页面上方的，保存模型。

步骤5 定义模型“workerOptions”。

1. 在“模型视图”中，单击“新增模型”。
2. 添加服务模型，模型名称“workerOptions”，单击“下一步”。
3. “项目”选择“设备维修管理系统”，并为模型关联API“queryWorker”，单击“下一步”，再单击“确定”。
4. 单击页面上方的，保存模型。

步骤6 拖拽页面组件。

1. 单击“设计视图”，返回页面设计。
2. 将左侧基本组件区的“表单”拖拽到右侧页面中。
因为当前还没有定义数据源，单击“取消”，创建一个空的表单控件。


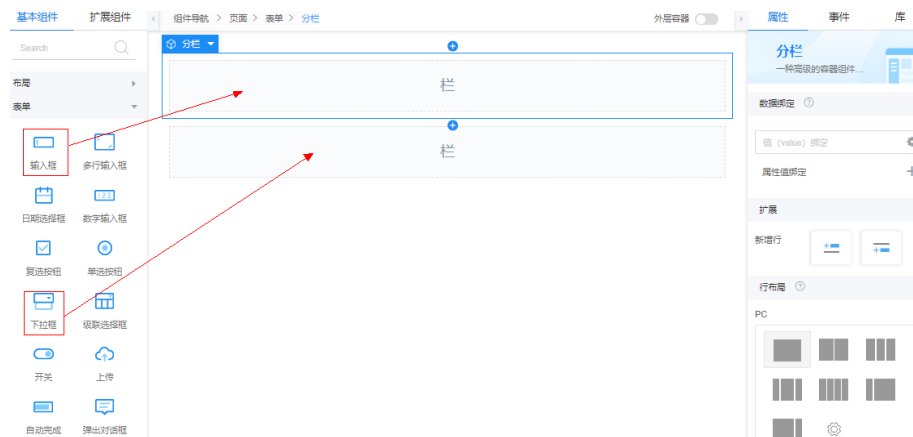
3. 拖拽一个“分栏”到表单中。
4. 选中“分栏”，在右侧单击PC下的单行图标，修改分栏为1栏，再单击“新增行”的，修改为2个分栏，每个分栏中有1栏。


图 3-202 设置分栏为 1 栏



5. 在第1个分栏中拖入一个“输入框”，在第2个分栏中拖入一个“下拉框”。

图 3-203 拖入输入框、下拉框到分栏



6. 选中“输入框”，在右侧属性面板中，单击，在选择模型弹窗中，选择“transInfo”下的“HW_workOrderId_CST”字段，并修改其“标签”为“工单ID”。
7. 选择下拉框进行以下设置。


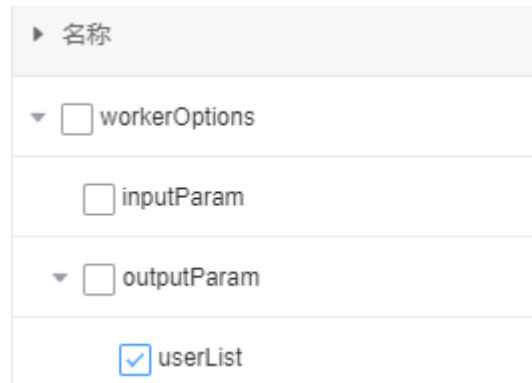
- a. 在右侧属性面板中，单击 ，在选择模型弹窗中，选择“transInfo”下的“HW_assignedFme_CST”字段，修改“标题”为“选择工程师”。
- b. 单击“属性值绑定”后的“+”，设置“属性”为“选项”，“模型字段”为“workerOptions.outputParam.userList”，即绑定属性值为服务对象的返回值。


图 3-204 绑定服务对象的返回值



- c. 开启“弹层独立”。

图 3-205 启用弹层独立



8. 单击页面上方的 ，保存设置。

步骤7 定义页面事件代码。

1. 在“设计视图”中选中最外层的“页面”，在右侧“事件”页签中，单击“加载”后的“+”，进入编辑动作页面。
2. 在“自定义动作”中，输入如下脚本代码。

须知

红色内容请替换为实际命名空间前缀。

```
let workOrderId = Page.params.workOrderId;
$model.ref("transInfo").setData({ "HW_workOrderId_CST": workOrderId });
//查询维修人员列表，作为维修人员下拉框的可选值
$model.ref('workerOptions').setData({ inputParam: {} });
$model.ref('workerOptions').run();
```

3. 单击页面上方的，并退出事件编排。

----结束

验证

单击界面上方的，进入预览页面，查看页面的展示效果。

因为这个页面需要从上一个页面获取工单信息，所以当前的预览效果只能看到空白“工单ID”以及“选择工程师”下拉框。

3.1.7.4.5 组装“工单列表（派单员）”页面



“工单列表（派单员）”页面与“工单列表（客服人员）”页面相比，大致功能一致，仅多了一个操作列，操作列中包含了派单员需要使用的派单图标。

因为两个页面很相似，所以为了快速创建，可以将“工单列表（客服人员）”页面保持为一个模板，再引用这个模板创建新的页面。

操作步骤

步骤1 在“我的应用”中，单击“设备维修管理系统”，进入应用。

步骤2 使用“工单列表（客服人员）”页面创建新页面。

1. 打开“HW_workOrderList”，单击锁定页面。
2. 单击页面上方的，另存为页面。
3. 如图3-206所示，在弹出对话框中设置模板标签和名称为“dispatchWorkOrder”，单击“创建”。

您可以看到新创建页面继承了原“工单列表（客服人员）”页面中的组件、模型和事件代码。

图 3-206 保存页面模板



另存为 ...

* 标签

dispatchWorkOrder

* 名称

dispatchWorkOrder

另存为模板?

* 缩略图

取消 创建

步骤3 删除“创建工单”功能。

选中“创建工单”按钮所在的“数据表格-工具栏”，单击鼠标右键，单击“删除”，删除前请确认已选中“创建工单”按钮，以免误删其他组件。

步骤4 为查询结果增加操作列。


1. 在左侧“设计视图”中，选中“表格”，单击右侧“属性”页签“表格列”中的 ，添加操作列。


图 3-207 增加操作列



2. 单击上图中Operation1后的⚙️。
3. 在“属性配置”对话框中进行以下配置。
 - a. 修改“列标题”为“操作”，单击“添加操作按钮”。
 - b. 设置按钮“标签”为“派单”，“图标”选择“shuffle”，并在“禁用”中输入“`$row.HW_status_CST!="待派单"`”，即是当工单状态为“待派单”时，该按钮才可以使用。“`HW_`”请修改为实际命名空间前缀。

图 3-208 为操作列增加派单按钮



- 单击图3-208中“派单”按钮的 ，再单击“添加动作”的“+”，进入事件编排中，在事件编排页面，添加如下事件代码，然后单击“创建”。

须知

脚本中加红色内容请替换为实际的页面名。

```
let _component = context.$component.current;
// 配置页面的bpm参数
context.$page.params["bp.name"] = "HW_WorkOrderBpm";
let currentRow = _component.$attrs.row;
let workOrderId = currentRow.HW_workOrderId_CST;
let instanceId = currentRow.HW_instanceId_CST;
let taskId = "";
let _model = $model.ref("workOrderInstance");
context.$dialog.popup(
  {
    title: '处理工单', //弹出框标题
    page: 'HW_workOrderDispatch', // 弹出页面名称
    footerHide: false,
    width: 20, //宽, 小于100为百分比, 大于100为px
    height: 250, // 高px
    okText: '确定',
    cancelText: '取消',
```



```

let pageInfo = $model.ref('table_0_condition').getData().pageInfo;
let queryData = {
  "start": (pageInfo.curPage - 1) * pageInfo.pageSize,
  "limit": pageInfo.pageSize
};
$model.ref('queryWorkOrder').setData({ inputParam: queryData });
$model.ref('queryWorkOrder').run().then(function(data) {
  $model.ref('workOrderInstance').setData(data.workOrderList);
  pageInfo.total = parseInt(data.total);
  $model.ref('table_0_condition').setData({"pageInfo": pageInfo });
}).catch(function(error) {
  console.log('error is', error);
});


```

3. 单击页面上方的, 并退出事件编排。

----结束

3.1.7.4.6 验证

因当前还未创建工单状态流转的BPM，因此派单功能还不能正常测试运行。本节只验证页面显示及跳转相关内容。

步骤1 在“HW_dispatchWorkOrder”页面中，单击界面上方的, 进入页面预览，在页面预览中进行以下验证。

步骤2 查看页面显示：正常情况下，系统会显示客服人员创建的工单，且每条记录后都有派单图标，如果工单状态为“待派单”，则派单按钮为高亮可用状态，如果不是“待派单”状态，则按钮灰度不可用。

如果当前工单列表中没有“待派单”状态的工单，可以在“生成工单”脚本中输入一条测试数据，生成一条工单。

图 3-209 页面预览

现场工单列表

工单标题		状态							
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="搜索"/>									
<input type="button" value="+ 创建工单"/>									
<input type="checkbox"/>	#	工单ID	优先级	工单标题	状态	当前处理人	创建时间	创建人	操作
<input type="checkbox"/>	1	WD0000123457	高	电梯无法关门	待派单	派单员	2020-09-30 16:40...	hw5	<input type="button" value="派单"/>
<input type="checkbox"/>	2	WD0000123456	高		待派单	派单员	2020-09-30 09:50...	hw5	<input type="button" value="派单"/>

步骤3 查看页面跳转。

选择一条“工单状态”为“待派单”的工单记录，单击“派单”按钮，查看是否弹出“处理工单对话框”，如果未跳转，请检查操作列“派单”按钮上的事件。

步骤4 在“处理工单对话框”中，查看是否显示“工单ID”，“选择工程师”下拉框是否显示正常，如果不正常，需要检查“派单”对话框相关事件代码及属性值绑定。

图 3-210 处理工单对话框



----结束


3.1.7.5 开发“维修工程师处理工单”功能

“维修工程师处理工单”功能包含“工程师查看待处理工单列表页面”和“处理工单对话框”两个页面。

3.1.7.5.1 创建“处理工单”脚本

“维修工程师处理工单”功能包含两个页面，一个是工程师查看待处理工单列表页面，一个是处理工单对话框页面。首先需要创建“处理工单”脚本。

操作步骤


- 步骤1** 在经典版开发环境“首页 > 我的应用”中，单击“设备维修管理系统”，进入应用。
- 步骤2** 在“WorkOrder”目录中，将鼠标放在“Script”上，单击界面上出现的“+”，在弹出菜单中选择“脚本”。
- 步骤3** 在弹窗中，选中“创建一个新脚本”，在“名称”文本框中输入“modifyOrderStatus”，单击“添加”。
当编辑已有脚本时，为防止编辑时多人篡改，编辑前请单击进行锁定。
- 步骤4** 在代码编辑器中插入如下脚本代码。

须知



脚本中加红色内容请替换为实际的对象名、字段名。

```
//脚本用于派发工单
import * as db from 'db';//导入处理object相关的标准库
import * as context from 'context';//导入上下文相关的标准库
import * as date from 'date';
//定义入参结构，入参包含1个参数：工单的状态信息，为必填字段
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'Object', required: true, label: 'object' })
  statusInfo: object;
}
//定义出参结构，出参包含2个参数，workOrder的记录id和当前状态
@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'String' })
  id: string;
  @action.param({ type: 'String' })
  status: string;
}
//使用数据对象HW_WorkOrder_CST
@useObject(['HW_WorkOrder_CST'])
@action.object({ type: "method" })
export class ModifyOrderStatus { //定义接口类，接口的入参为ActionInput，出参为ActionOutput
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public modifyOrderStatus(input: ActionInput): ActionOutput {
    let out = new ActionOutput(); //新建出参ActionOutput类型的实例，作为返回值
    let error = new Error(); //新建错误类型的实例，用于在发生错误时保存错误信息
    try {
      let statusInfo = input.statusInfo; //将入参赋值给statusInfo变量，方便后面使用
      let s = db.object('HW_WorkOrder_CST'); //获取HW_WorkOrder_CST这个Object的操作实例

      //查询条件
      let condition = {
        "conjunction": "AND",
        "conditions": [{
          "field": "id",
          "operator": "eq",
          "value": statusInfo['id']
        }]
      };
      //查找workOrderId所代表的工单信息
      let workOrder = s.queryByCondition(condition);
      if (statusInfo['HW_status_CST'] == "接单") {
        workOrder[0]['HW_status_CST'] = "处理中";
      } else if (statusInfo['HW_status_CST'] == "拒单") {
        workOrder[0]['HW_status_CST'] = "待派单";
        workOrder[0]['HW_assignedFme_CST'] = "派单员";
      } else if (statusInfo['HW_status_CST'] == "关单") {
        workOrder[0]['HW_status_CST'] = "关闭";
      }
      let isUpdated = s.updateByCondition(condition, workOrder[0]);
      if (isUpdated) {
        out.id = workOrder[0]['id'];
        out.status = statusInfo['HW_status_CST'];
      } else {
        error.name = "WOERROR";
        error.message = "修改工单状态失败! ";
        throw error;
      }
    } catch (error) {
      console.error(error.name, error.message);
      context.setError(error.name, error.message);
    }
    return out;
  }
}
```

步骤5 单击编辑器上方的，保存脚本。

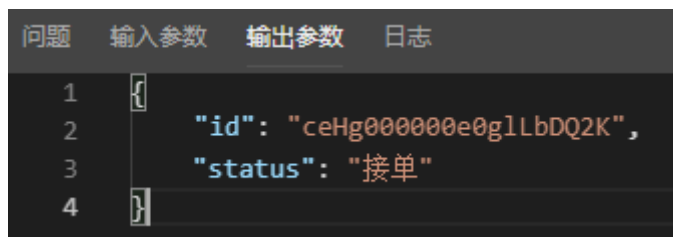
步骤6 测试脚本能否正常执行。


1. 单击编辑器上方的, 执行脚本。
2. 在界面底部, 输入如下示例, 单击测试窗口右上角执行图标。
“ceHg000000e0gLLbDQ2K” 则是派单员操作的当前工单记录的工单id, 您可以在派单功能脚本的测试结果中获取一个id值。

```
{
  "statusInfo": {
    "id": "ceHg000000e0gLLbDQ2K",
    "HW_status_CST": "接单"
  }
}
```

执行成功, 会在“输出”页签返回查询结果。

图 3-211 输出处理后工单信息




步骤7 测试成功, 单击编辑器上方的, 启用发布脚本。

----结束

3.1.7.5.2 创建“判断下一步状态”脚本

- 步骤1 在“我的应用”中, 单击“设备维修管理系统”, 进入应用。
- 步骤2 在“WorkOrder”目录中, 将鼠标放在“Script”上, 单击界面上出现的“+”, 在弹出菜单中选择“脚本”。
- 步骤3 在弹窗中, 选中“创建一个新脚本”, 在“名称”文本框中输入“judgeNextStatus”, 单击“添加”。

当编辑已有脚本时, 为防止编辑时多人篡改, 编辑前请单击进行锁定。


步骤4 在代码编辑器中插入如下脚本代码。

须知



脚本中红色内容请替换为实际的对象名、字段名。

```
//本脚本用于判断下一步状态变化
import * as db from 'db';//导入处理object相关的标准库
import * as context from 'context';//导入上下文相关的标准库
//定义入参结构, 入参包含1个参数: workOrder对象, 为必填字段
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'String', required: true, label: 'String' })
  id: string;
}
//定义出参结构, 出参包含1个参数, workOrder的记录id
@action.object({ type: "param" })
export class ActionOutput {
```

```
@action.param({ type: 'Any', label: 'Object', isCollection: true })
statusList: object[];
}
//使用数据对象HW_WorkOrder_CST
@useObject(['HW_WorkOrder_CST'])
@action.object({ type: "method" })
export class JudgeNextStatus { //定义接口类，接口的入参为ActionInput，出参为ActionOutput
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public judgeNextStatus(input: ActionInput): ActionOutput {
    let out = new ActionOutput(); //新建出参ActionOutput类型的实例，作为返回值
    let error = new Error(); //新建错误类型的实例，用于在发生错误时保存错误信息
    try {
      let id = input.id;
      let s = db.object('HW_WorkOrder_CST'); //获取HW_WorkOrder_CST这个Object的操作实例
      //查询条件
      let condition = {
        "conjunction": "AND",
        "conditions": [{
          "field": "id",
          "operator": "eq",
          "value": id
        }]
      };
      let workOrder = s.queryByCondition(condition);
      if (workOrder) {
        if (workOrder[0].HW__status__CST == "待接单") {
          out.statusList = [
            {
              'value': "接单",
              'display': "接单"
            }, {
              'value': "拒单",
              'display': "拒单"
            }
          ];
        }
        if (workOrder[0].HW__status__CST == "处理中") {
          out.statusList = [{
            'value': "关单",
            'display': "关单"
          }];
        }
      }
    } catch (error) {
      console.error(error.name, error.message);
      context.setError(error.name, error.message);
    }
    return out;
  }
}
```

步骤5 单击编辑器上方的，保存脚本。

步骤6 测试脚本能否正常执行。


1. 单击编辑器上方的，执行脚本。
2. 在界面底部，输入如下输出参数报文，单击测试窗口右上角执行图标，“ceHg000000e0glLbDQ2K”则是派单员操作的当前工单记录的工单id，您可以在[处理工单](#)脚本的测试结果中获取一个id值。

```
{
  "id": "ceHg000000e0glLbDQ2K"
}
```

执行成功，会在“输出”页签返回查询结果。

图 3-212 返回下一步状态



步骤7 测试成功，单击编辑器上方的，启用发布脚本。

---结束

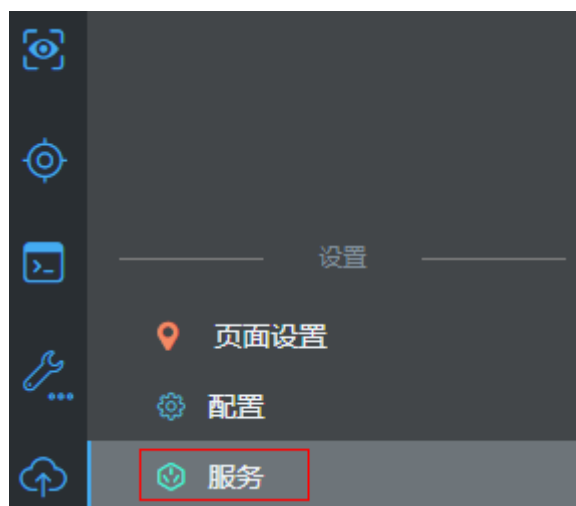
3.1.7.5.3 创建公共接口

参考之前创建公共接口的步骤，创建“查询维修人员”脚本对应的公共接口。

操作步骤

步骤1 在经典版应用开发页面的设计视图下，单击“服务”，进入公共接口创建页面。

图 3-213 服务入口



步骤2 单击“新建”，如下图所示。

图 3-214 公共接口创建

公共接口

使用公共接口，可以将flow的URL映射到发布到外部网关，第三方可以通过OAuth2.0调用。



步骤3 创建“处理工单”、“判断下一步状态”脚本对应的公共接口，详细接口信息如表 3-26所示。

表 3-26 公共接口

设置操作	版本	URL	方法	类型	资源
modifyOrderStatus	1.0.0	/modifyOrderStatus	POST	脚本	<i>HW_modifyOrderStatus</i>
judgeNextStatus	1.0.0	/judgeNextStatus	POST	脚本	<i>HW_judgeNextStatus</i>

----结束

3.1.7.5.4 组装“处理工单”对话框

“维修工程师处理工单”功能包含两个页面，一个是工程师查看待处理工单列表页面，一个是处理工单对话框。

“处理工单”对话框是一个手机端的Html5页面，负责指定下一环节的工单状态，是“待处理工单列表”页面中引用的对话框。

按照工程师在对话框中选定下一步流程状态，如图3-215所示。

图 3-215 现场工程师处理工单对话框

选择下一步操作
请选择 >

页面模型分析

页面模型负责与页面组件交互，传递处理工单需要的工单ID、工单状态等信息。结合页面需求，“处理工单”对话框页面，需要创建如下模型：

表 3-27 模型分析


模型名称	作用	详细定义
statusInfo	保存派单的参数。	自定义模型，包含的计算节点如下，这些节点与派单接口的输入参数名称一一对应。 <ul style="list-style-type: none"> id: 工单ID <i>HW_status_CST</i>: 下一环节状态 须知 加粗斜体内容请替换为实际命名空间前缀。
statusOptions	判断下一步状态。	服务模型，绑定公共接口“judgeNextStatus”，调用“judgeNextStatus”脚本，判断下一步状态。

操作步骤

步骤1 在“我的应用”中，单击“设备维修管理系统”，进入应用。

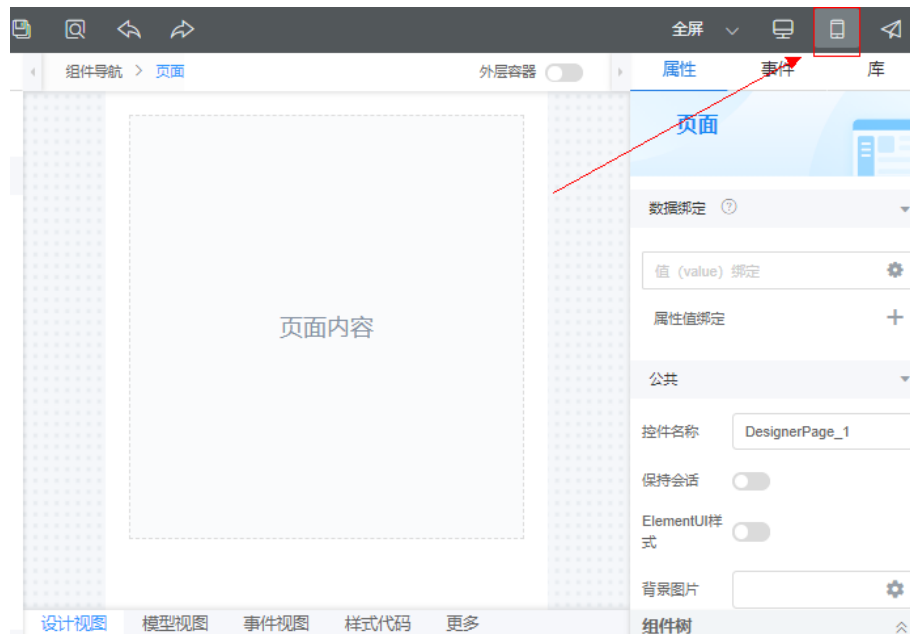
步骤2 鼠标放在“Page”上，单击界面上出现的“+”，在弹出菜单中选择“标准页面”。

步骤3 在“标签”和“名称”文本框中输入“workOrderProcess”，单击“添加”。


当编辑已有标准页面时，为防止编辑时多人篡改，编辑前请单击进行锁定。

步骤4 单击页面右上部的，切换到手机端设计模式。


图 3-216 切换手机端设计模式



步骤5 定义模型“statusInfo”。

1. 在“模型视图”中，单击“新增模型”。
2. 添加自定义模型，模型名称“statusInfo”，单击“下一步”。
3. 单击“新增节点”，为模型添加节点“id”和“HW_status_CST”字段，字段类型都采用默认的Text。单击“下一步”，再单击“确定”，加粗斜体内容请替换为实际命名空间前缀。
4. 单击页面上方的，保存模型。

步骤6 定义模型“statusOptions”。

1. 在“模型视图”中，单击“新增模型”。
2. 添加服务模型，模型名称“statusOptions”，单击“下一步”，
3. “服务类型”选择“API”，“选择项目”为“设备维修管理系统”，“搜索”中的API，选择“judgeNextStatus”，单击“下一步”，再单击“确定”。
4. 单击页面上方的，保存模型。

步骤7 拖拽页面组件。

1. 从“模型视图”切换到“设计视图”。
2. 从左侧基本组件列表区中，拖拽1个“表单”到“页面内容”中。
因为当前还没有定义数据源，单击“取消”元数据表单配置向导弹窗，创建一个空的表单控件。
3. 拖拽1个“分栏”到“表单”。
4. 修改“分栏”为1栏（1行1列）。

图 3-217 设置分栏为 1 行 1 列




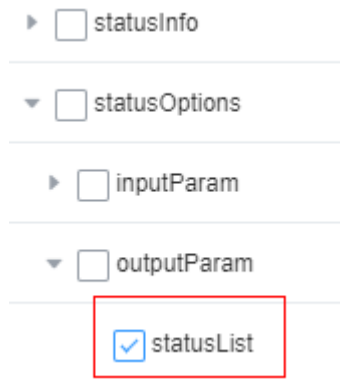
5. 拖拽1个“下拉框”到分栏的“栏”中，然后选中下拉框，在右侧属性面板中，对下拉框进行以下配置。
 - a. 添加值绑定。
单击“数据绑定”下“值绑定”的 ，在弹窗中选择“statusInfo”下的“HW_status_CST”字段。
 - b. 添加属性值绑定。
单击“属性值绑定”的“+”，设置“属性”为“选项”，“模型字段”为“statusOptions”下“outputParam”节点的“statusList”。


图 3-218 属性值绑定



- c. 修改“标签”为“选择下一步操作”，并设置“选项”为“user1, user1”、“user2, user2”。

图 3-219 设置选项值



- d. 单击页面上方的, 保存设置。

步骤8 定义页面事件代码。


1. 在“设计视图”中，选中最外层的“页面”。
2. 在右侧“事件”页签中，单击“加载”后的“+”。
3. 在添加动作弹窗中，输入如下事件代码，然后单击“创建”。

```
let id = Page.params.id;  
$model.ref("statusInfo").setData({ "id": id });  
$model.ref('statusOptions').setData({ inputParam: { "id": id } });  
$model.ref('statusOptions').run();
```

4. 单击页面上方的, 并退出事件编排。

----结束

验证

单击界面上方的, 进入预览页面，查看页面的展示效果。

因为这个页面需要从上一个页面获取工单信息，所以当前的预览效果只能看到“选择下一步”下拉框。

3.1.7.5.5 组装“待处理工单”页面

“待处理工单”页面是一个手机端的页面，用于显示现场工程师名下的待处理工单，将使用标准页面功能实现。

如图3-220所示工程师在手机端可以看到自己名下所有的待处理工单。系统默认优先显示待处理的工单。

图 3-220 “待处理工单”页面



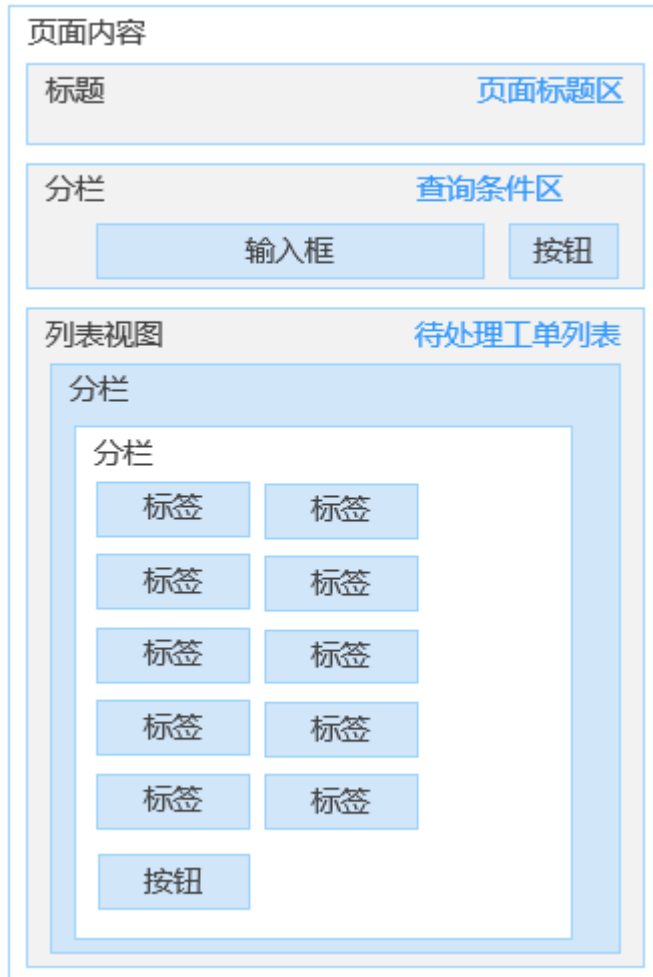
因当前页面组件较多，因此我们分3个大步骤开发页面，先拖拽页面的前端组件，再定义页面模型，进行模型与前端组件绑定，最后为页面添加相应的事件代码。

页面组件分析

分析图3-220的组成，可以将整个页面分割为3个区域：页面标题、查询条件和待处理工单列表区域，如图3-221所示。

待处理工单列表通过在列表视图中，使用2个“分栏”的嵌套，组装出待处理工单列表区。

图 3-221 页面组件分析




操作步骤


以下操作步骤只是给出的大致的拖拽方法，您可以根据需要，修改各组件的颜色、对齐方式、边距显示样式，使页面更美观。

步骤1 在“我的应用”中，单击“设备维修管理系统”，进入应用。

步骤2 鼠标放在“Page”上，单击界面上出现的“+”，在弹出菜单中选择“标准页面”。

步骤3 在“标签”和“名称”文本框中输入“workOrderListM”，单击“添加”。

当编辑已有标准页面时，为防止编辑时多人篡改，编辑前请单击进行锁定。

步骤4 单击页面右上部的 Mobile图标，切换到手机端设计模式。

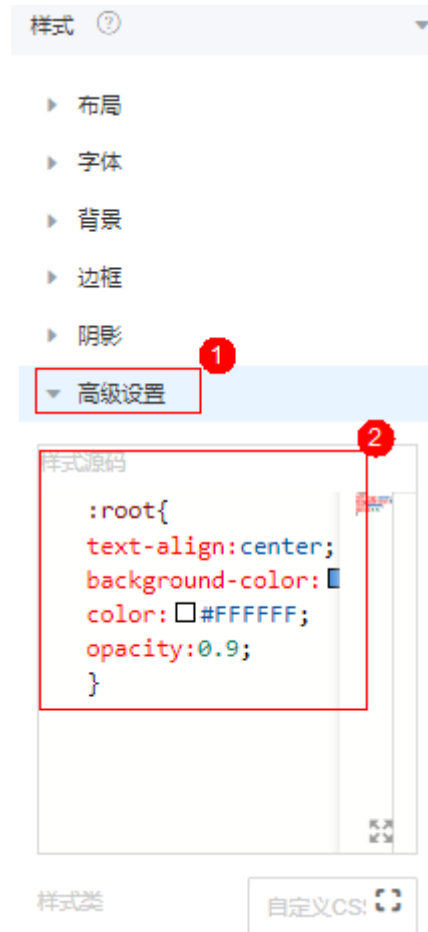
步骤5 组装“页面标题区”。

1. 从基本组件列表区，拖拽1个“标题”组件到“页面内容”。
2. 选中“标题”组件，在右侧“属性”页签，配置标题组件属性：

- a. 修改“标题内容”为“工单列表”。
- b. 在“样式”下的“高级设置”中，输入如下样式代码，即设置标题组件的背景色、水平居中对齐，文字颜色为白色。

```
:root{
text-align:center;
background-color:#4a90e2;
color:#FFFFFF;
opacity:0.9;
}
```

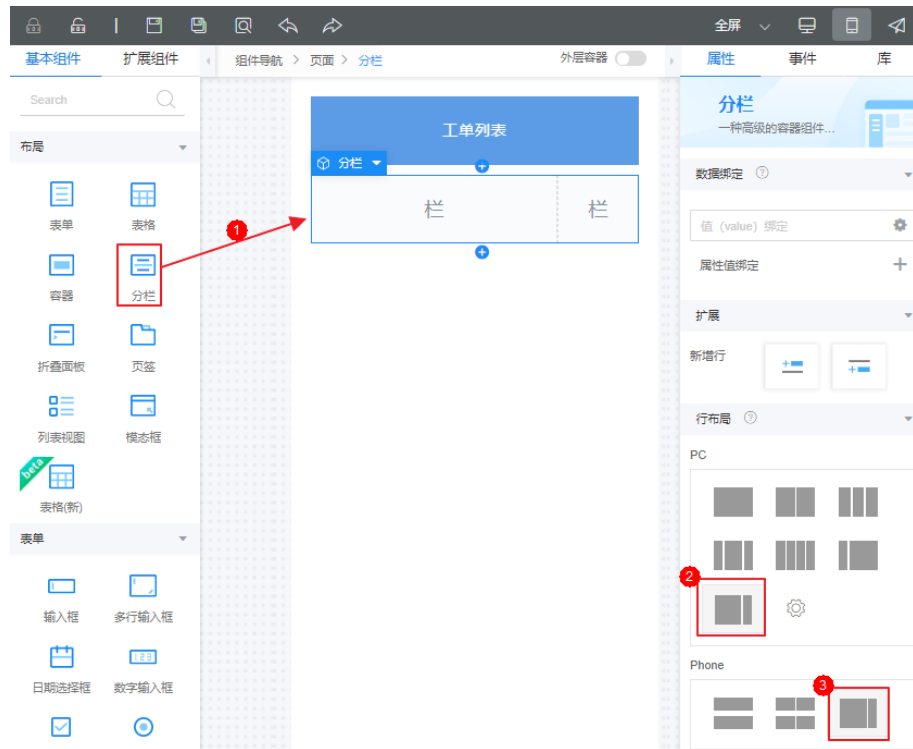
图 3-222 高级设置



步骤6 组装“查询条件区”。

1. 拖拽1个“分栏”到“工单列表”之后，在右侧“行布局”下先单击PC中的2栏图标，再单击Phone下的2栏图标，将分栏设置为左右不均分的2栏，如下图所示。

图 3-223 拖入分栏组件并设置为 2 栏



2. 选中左侧“栏”，在“样式 > 高级设置”下输入样式代码，设置宽度为75%，高度为40px。

图 3-224 修改列宽度



```
:root{  
width:75%;  
height:40px;  
}
```

- 选中右侧“栏”，同样方式设置宽度为25%，高度为40px。

```
:root{  
width:25%;  
height:40px;  
}
```

- 拖拽1个“输入框”组件到“分栏”的左栏，修改其“占位符”为“请输入工单完整标题”。
- 拖拽1个“按钮”组件到“分栏”的右栏，修改其“显示名称”为“查询”、“类型”为“主要按钮”。
- 选择“请输入工单完整标题”所在的“栏”，设置“弹性布局”，并修改内边距为“10px”。

图 3-225 设置栏属性



步骤7 组装“待处理工单列表区”。

1. 在“分栏”之后，拖入1个“列表视图”组件。

图 3-226 拖入列表视图



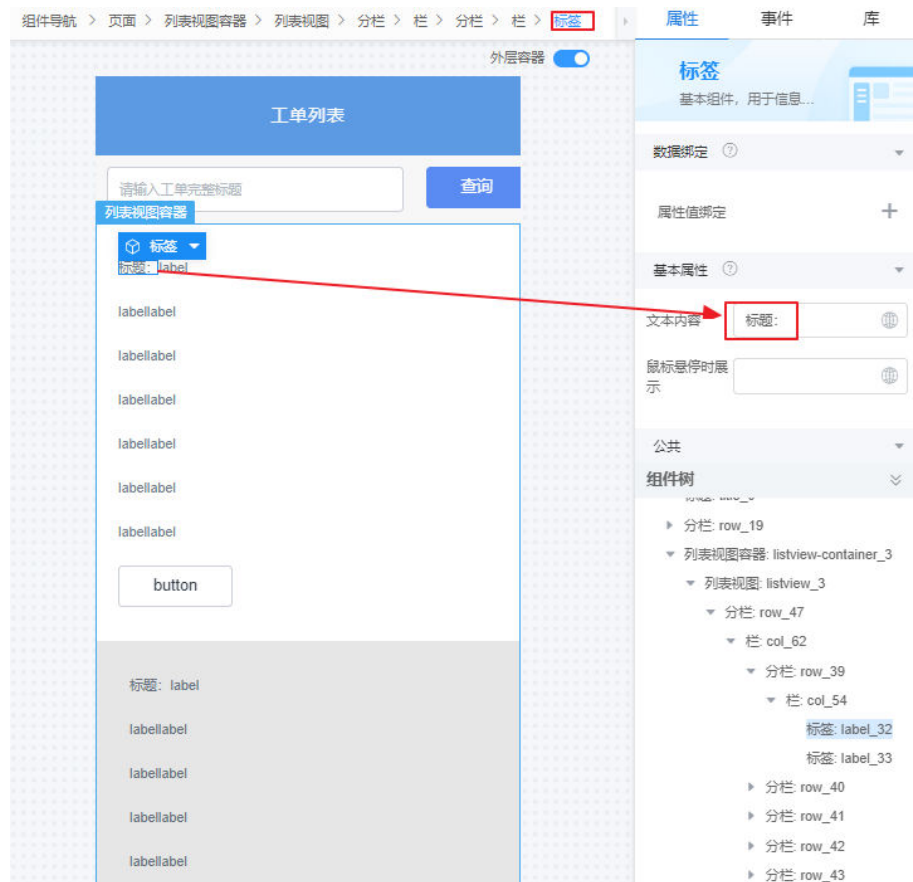
2. 在“列表视图容器”下层的“列表视图”中，拖入1个“分栏”，在右侧修改为1个栏（Phone），并设置分栏内边距为“10px”。

图 3-227 修改分栏为 1 栏及内边距



3. 在“分栏”中，再拖入1个“分栏”，并参考上图修改为1个栏（Phone），然后单击分栏的+7次，新增7个分栏，新增后，变为8个分栏。
4. 向第1~7分栏中，分别拖拽2个“标签”组件（并排），然后在第8分栏中，拖入一个“按钮”组件。
5. 修改第1分栏中的第1个标签的“文本内容”修改为“标题：”。

图 3-228 修改第 1 个标签标题



6. 重复上一步，修改第2~7分栏中的第1个标签的“文本内容”为“单号：”、“状态：”、“时间：”和“优先级：”。

表 3-28 标签的文本内容

所在分栏	第一栏	第二栏
1	标题：	保持默认（label）
2	单号：	保持默认（label）
3	状态：	保持默认（label）
4	要求到达时间：	保持默认（label）
5	要求解决时间：	保持默认（label）
6	现场故障说明：	保持默认（label）
7	优先级：	保持默认（label）

📖 说明

正常情况下，第二栏中的内容是动态获取的，在表3-30中，设置的第二栏中的“文本内容”，仅会在未获取相应工单信息时显示，您也可以根据实际需要自定义“标题”组件默认显示内容。

7. 在“样式”属性下，自定义修改各个标签组件的样式，例如加粗文本。

图 3-229 自定义修改样式



8. 在第8行内，选中“按钮”组件，设置按钮“显示名称”为“处理”、“类型”为“主要按钮”，您也可以在属性中为按钮添加图标，修改按钮背景色为“#4a90e2”。

组装完成效果，如[图3-230](#)所示。

图 3-230 组装工单信息展示区域

工单列表

请输入工单完整标题

标题: label

单号: label


状态: label

要求到达时间: label

要求解决时间: label

现场故障说明: label

优先级: label

步骤8 单击界面上方的, 保存页面。

----结束


定义模型并绑定页面组件

页面模型负责与页面组件交互，获取显示工单需要的数据。结合页面展示需求，可知需要创建如下模型：


表 3-29 模型分析

模型名称	作用	详细定义
workOrderList	保存工单的基本信息。	<p>自定义模型，包含的字段节点如下，这些节点与工单对象的字段名称相同，加粗斜体内容请替换为实际命名空间前缀：</p> <ul style="list-style-type: none"> • <i>HW_title_CST</i>: 工单标题，字段类型Text。 • <i>HW_workOrderId_CST</i>: 工单ID，字段类型Text。 • <i>HW_priority_CST</i>: 工单优先级，字段类型Text。 • id: 记录ID，字段类型Text。 • <i>HW_recoveryTime_CST</i>: 要求解决时间，字段类型Text。 • <i>HW_faultPhenomenon_CST</i>: 现场故障说明，字段类型Text。 • <i>HW_status_CST</i>: 工单状态，字段类型Text。 • <i>HW_instanceld_CST</i>: 工单绑定的BPM实例的id，字段类型Text。 • <i>HW_arriveTime_CST</i>: 要求到达时间，字段类型Text。 • isDeal: 判断工单状态字段，字段类型Bool。
queryCondition	查询条件。	自定义模型，包含字段节点: title, 字段类型Text, 与页面上的查询条件对应。
queryWorkOrder	查询工单。	服务模型，绑定公共接口“queryWorkOrder”，调用查询工单逻辑。
total	保存满足查询条件的记录数，作为Scroll组件滚动显示时的累计记录数。	<p>自定义模型，包含的字段节点如下：</p> <ul style="list-style-type: none"> • currentTotal: 当前页显示的数量，字段类型Text。 • actualTotal: 实际总数量，字段类型Text。


步骤1 定义模型“workOrderList”。

1. 单击“模型视图”，切换到模型视图。
2. 在模型视图中，单击“新增模型”。
3. 添加自定义模型，模型名称“workOrderList”，单击“下一步”。
4. 单击“新增节点”，依次增加表3-29中列出的字段名称，单击“下一步”，再单击“确定”。
5. 单击页面上方的，保存模型。


步骤2 定义模型“queryCondition”。

1. 在“模型视图”中，单击“新增模型”。
2. 添加自定义模型，模型名称“queryCondition”，单击“下一步”。
3. 单击“新增节点”，依次增加表3-29中列出的字段名称，单击“下一步”，再单击“确定”。
4. 单击页面上方的，保存模型。

步骤3 定义模型“total”。

1. 在“模型视图”中，单击“新增模型”。
2. 添加自定义模型，模型名称“total”，单击“下一步”。
3. 单击“新增节点”，依次增加表3-29中列出的字段名称，单击“下一步”，再单击“确定”。
4. 单击页面上方的，保存模型。

步骤4 定义模型“queryWorkOrder”。

1. 在“模型视图”中，单击“新增模型”。
2. 添加服务模型，模型名称“queryWorkOrder”，单击“下一步”，
3. “服务类型”选择“API”，“选择项目”为“设备维修管理系统”，“搜索”中的API，选择“queryWorkOrder”，单击“下一步”，再单击“确定”。
4. 单击页面上方的，保存模型。

步骤5 绑定模型“workOrderList”。

1. 单击“设计视图”，从“模型视图”切换到“设计视图”。
2. 选中“列表视图容器”标签中的“列表视图”标签（不要选择列表视图容器），为其绑定“workOrderList”。

图 3-231 绑定数据模型



3. 如图3-232所示，选中“标题：”后的标签组件label，在右侧“属性”页签，单击“+”，增加属性值绑定，即设置“属性”为“文本内容”，“模型字段”为“workOrderList”下的“HW_title_CST”。

图 3-232 为标签绑定模型



4. 依次为其他标签添加属性值绑定，具体绑定的字段如表3-30所示。


表 3-30 标签的属性值绑定

所在分栏	第一栏	第二栏	第二栏标签绑定的字段
1	标题:	label	workOrderList.HW_title_CST
2	单号:	label	workOrderList.HW_workOrderId_CST
3	状态:	label	workOrderList.HW_status_CST
4	要求到达时间:	label	workOrderList.HW_arriveTime_CST
5	要求解决时间:	label	workOrderList.HW_recoveryTime_CST
6	现场故障说明:	label	workOrderList.HW_faultPhenomenon_CST
7	优先级:	label	workOrderList.HW_priority_CST

5. 为“处理”按钮添加属性值绑定，“属性”设置为“禁用”，绑定“workOrderList”的“isDeal”字段，绑定后为“workOrderList.isDeal”。

图 3-233 添加属性值绑定




6. 单击页面上方的，保存设置。

步骤6 绑定模型“queryCondition”。

1. 选中“请输入工单完整标题”输入框，为其进行值绑定，绑定字段为“queryCondition.title”。

图 3-234 绑定工单标题



2. 单击页面上方的，保存设置。

----结束


定义页面中的事件代码

通过定义页面的on-load事件，可以实现打开页面即自动展示当前工程师的待处理工单。

步骤1 定义页面事件代码。

1. 在“设计视图”中，选中最外层的“页面”。

2. 在右侧“事件”页签中，单击“加载”后的“+”。
3. 在添加事件弹窗中，输入如下脚本代码。

```
$model.ref('queryWorkOrder').setData({inputParam: {"isFME": true}});  
$model.ref('queryWorkOrder').run().then(function (data) {  
  if (data && data.workOrderList) {  
    //将查询到的数据赋值给页面模型来展示  
    $model.ref('workOrderList').setData(data.workOrderList);  
  }  
}).catch(function (error) {  
  console.log('error is', error);  
});
```
4. 单击页面上方的，并退出事件编排。

----结束

实现“按标题查询工单”

通过定义“查询”按钮的“点击”事件，可以实现根据工单标题查询工单的能力。

- 步骤1 在“设计视图”中，选中“查询”按钮。
- 步骤2 在右侧“事件”页签中，单击“点击”后的“+”。
- 步骤3 在添加事件弹窗中，输入如下脚本代码。

```
let queryTitle = $model.ref('queryCondition').getData().title;  
$model.ref('queryWorkOrder').setData({ inputParam: { "title": queryTitle, "isFME": true } });  
$model.ref('queryWorkOrder').run().then(function (data) {  
  if (data && data.workOrderList) {  
    //将查询到的数据赋值给页面模型来展示  
    $model.ref('workOrderList').setData(data.workOrderList);  
  }  
}).catch(function (error) {  
  console.log('error is', error);  
});
```

- 步骤4 单击页面上方的，并退出事件编排。

----结束

实现“处理工单”

通过定义“处理”按钮的“点击”事件，可以实现接单、处理工单的能力。

- 步骤1 在“设计视图”中，选中“处理”按钮。
- 步骤2 在右侧“事件”页签中，单击“点击”后的“+”。
- 步骤3 在添加事件弹窗中，输入如下脚本代码。

须知

脚本中**红色内容**请替换为实际的页面名。

```
//获取当前组件（即button）  
let _component = context.$component.current;  
// 配置页面的bpm参数  
context.$page.params["bp.name"] = "HW_WorkOrderBpm";  
//获取当前行id  
let id = _component.$attrs.row.id;
```



```
let taskId = "";
let instanceld = "";
//遍历获取当前行内容
let row = {};
let workorders = $model.ref('workOrderList').getData();
workorders.forEach(function (wo, idx) {
  if (wo.id == id) {
    row = wo;
    instanceld = wo.HW__instanceld__CST;
  }
});
//打开处理弹框
context.$dialog.popup({
  title: '处理工单',
  page: 'HW__workOrderProcess',
  footerHide: false,
  width: 20,
  height: 180,
  okText: '提交',
  cancelText: '取消',
  params: { id: id, row: row },
  onCancel: function () {
  },
  onOk: function () {
    let statusInfo = { "statusInfo": $model.ref("statusInfo").getData() };
    let taskData = {
      "action": "complete",
      "variables": statusInfo
    };

    // 获取csrf token
    context.$utils.getCSRFToken().then(function (token) {
      let url = '/u-route/baas/bp/v2.0/query/tasks?flag=activeTask&rootID=' + instanceld;
      fetch(url, {
        method: 'GET',
        headers: {
          'Content-Type': 'application/json',
          'CSRF-Token': token
        }
      }).then(function (resp) {
        resp.json().then(function (data) {
          taskId = data.result.Recs[0].id;
          let url2 = '/u-route/baas/bp/v2.0/runtime/tasks/' + taskId;
          fetch(url2, {
            method: 'PUT',
            headers: {
              'Content-Type': 'application/json'
            },
            body: JSON.stringify(taskData)
          }).then(function (resp) {
            context.$message.success('处理成功');
            $model.ref('queryWorkOrder').setData({ inputParam: { "isFME": true } });
            $model.ref('queryWorkOrder').run().then(function (data) {
              if (data && data.workOrderList) {
                //将查询到的数据赋值给页面模型来展示
                $model.ref('workOrderList').setData(data.workOrderList);
              }
            }).catch(function (error) {
              console.log('error is', error);
            });
          }).catch(function (error) {
            console.log('error is', error);
          });
        });
      }).catch(function (error) {
        console.log('error is', error);
      });
    });
  });
});
```

```
}  
});
```

步骤4 单击页面上方的，并退出事件编排。

----结束

验证

通过PC端预览页面的方式，访问“待处理工单”页面进行测试。

说明

请在PC端预览或测试“待处理工单”标准页面，当前版本不建议在移动端预览页面。

步骤1 打开“待处理工单”标准页面。

步骤2 单击界面上方的，进入预览页面。

步骤3 查看页面的展示效果。

图 3-235 工单列表页面



如果当前租户开发者名下没有工单，所以当前的预览效果只能看到页面框架，没有工单。

图 3-236 无工单信息



----结束

3.1.7.6 开发“管理员管理工单”功能

3.1.7.6.1 创建“删除工单”脚本及公共接口

管理员的“工单管理”页面与“工单列表（派单）”页面相比，功能大致相同，仅多了一个创建工单功能、操作列中的删除按钮，因此需要多创建一个实现删除工单功能的后台逻辑。

创建脚本

步骤1 在“我的应用”中，单击“设备维修管理系统”，进入应用。

步骤2 在“WorkOrder”目录中，将鼠标放在“Script”上，单击界面上出现的“+”，在弹出菜单中选择“脚本”。

步骤3 在弹窗中，选中“创建一个新脚本”，在“名称”文本框中输入“deleteWorkOrder”，单击“添加”。

当编辑已有脚本时，为防止编辑时多人篡改，编辑前请单击进行锁定。


步骤4 在代码编辑器中插入如下脚本代码。

须知



脚本中**红色内容**请替换为实际的对象名、字段名。

```
//本脚本用于删除工单
import * as db from 'db';//导入处理object相关的标准库
import * as context from 'context';//导入上下文相关的标准库
//定义入参结构，入参包含1个参数：workOrder对象，为必填字段
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'String', required: true, label: 'String' })
  id: string;
}
//定义出参结构，出参包含1个参数，workOrder的记录id
@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'String' })
  id: string;
}
```

```
//使用数据对象HW_WorkOrder_CST
@useObject(['HW_WorkOrder_CST'])
@action.object({ type: "method" })
export class DeleteWorkOrder { //定义接口类，接口的入参为ActionInput，出参为ActionOutput
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public deleteWorkOrder(input: ActionInput): ActionOutput {
    let out = new ActionOutput(); //新建出参ActionOutput类型的实例，作为返回值
    let error = new Error(); //新建错误类型的实例，用于在发生错误时保存错误信息
    try {
      let id = input.id;
      let s = db.object('HW_WorkOrder_CST'); //获取HW_WorkOrder_CST这个Object的操作实例
      //查询条件
      let condition = {
        "conjunction": "AND",
        "conditions": [{
          "field": "id",
          "operator": "eq",
          "value": id
        }]
      };
      let isDeleted = s.deleteByCondition(condition);
      if (isDeleted) {
        out.id = id;
      } else {
        error.name = "WOERROR";
        error.message = "删除工单失败! ";
        throw error;
      }
    } catch (error) {
      console.error(error.name, error.message);
      context.setError(error.name, error.message);
    }
    return out;
  }
}
```


步骤5 单击编辑器上方的，保存脚本。

步骤6 测试脚本能否正常执行。

1. 单击编辑器上方的，执行脚本。
2. 在界面底部，输入以下报文作为“输入参数”，然后单击测试窗口右上角执行图标。“ceHg000000e0gLLbDQ2K”则是派单员操作的当前工单记录的工单id，您可以在[派单功能](#)脚本的测试结果中获取一个id值。

```
{
  "id": "ceHg000000e0gLLbDQ2K"
}
```

执行成功，会在“输出”页签返回查询结果。

步骤7 测试成功，单击编辑器上方的，启用发布脚本。

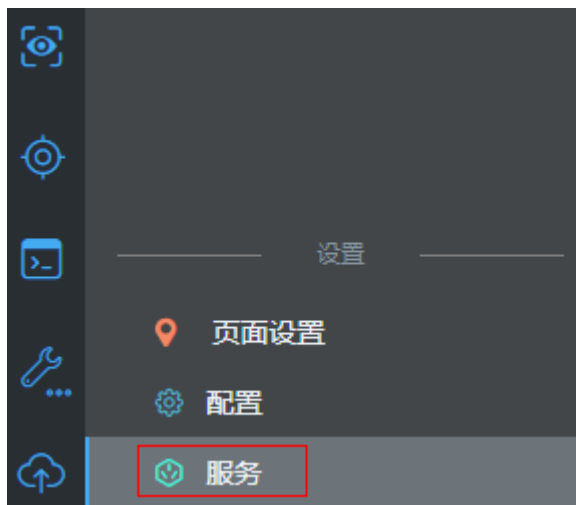
----结束

创建公共接口

参考之前创建公共接口的步骤，创建“删除工单”脚本对应的公共接口。

步骤1 在设计视图下，单击下方“服务”，进入公共接口创建页面。

图 3-237 创建公共接口入口



步骤2 在公共接口中，单击“新建”。

图 3-238 公共接口创建

公共接口

使用公共接口，可以将flow的URL映射到发布到外部网关，第三方可以通过OAuth2.0调用。



步骤3 创建“删除工单”脚本对应的公共接口，详细接口信息如表3-31所示。

表 3-31 公共接口

设置操作	版本	URL	方法	类型	资源
deleteWorkOrder	1.0.0	/deleteWorkOrder	DELETE	脚本	HW_deleteWorkOrder

---结束

3.1.7.6.2 组装“工单管理”页面

管理员的“工单管理”页面与“工单列表（派单员）”页面相比，大致功能一致，仅多了一个创建工单功能，和操作列中的删除按钮。因为两个页面很相似，所以为了快速创建，我们将通过“工单列表（派单员）”页面另存一个页面，然后再改造这个另存后的页面成为“管理员管理工单”页面。

“工单管理”页面是在“工单列表（派单员）”基础上创建的，因此该页面创建完成后，就已经具备了“查询工单”、“派发工单”的功能，而在本节开发步骤中，主要为该页面添加“创建工单”、“删除工单”功能即可。其中，“创建工单”功能与“工单列表（客服人员）”页面中的创建方法基本一致。

操作步骤

步骤1 在经典版开发环境“首页 > 我的应用”中，单击“设备维修管理系统”，进入应用。

步骤2 使用“工单列表（派单员）”页面创建模板。



1. 打开“*HW_dispatchWorkOrder*”，单击锁定页面。
2. 单击页面上方的，保存模板。
3. 如**图3-239**所示，在弹出对话框中设置页面标签和名称为“*manageWorkOrder*”，单击“创建”。

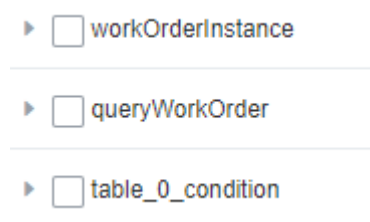
图 3-239 保存页面模板



步骤3 检查及创建页面模型。

1. 一般情况下，另存方式创建的页面，会保留原页面中的页面模型，单击“模型视图”，查看当前页面是否存在以下模型。

图 3-240 页面原有模型




2. 创建服务模型“deleteWorkOrder”。该模型用于绑定删除工单公共接口。
 - a. 在“模型视图”，单击“新增模型”。
 - b. 添加服务模型，模型名称“deleteWorkOrder”，单击“下一步”，
 - c. “服务类型”选择“API”，“选择项目”为“设备维修管理系统”，“搜索”中的API，选择“deleteWorkOrder”，单击“下一步”，再单击“确定”。
 - d. 单击页面上方的，保存模型。

步骤4 添加“创建工单”按钮。

1. 单击“设计视图”，从“模型视图”切换到“设计视图”。
2. 在页面中，选中“表格”，单击右侧“属性 > 表格区块”中“工具栏”后的“添加”按钮。

图 3-241 增加查询条件区域



3. 删除工具栏中多余的按钮，只保留“新增行”。
4. 修改“新增行”的“显示名称”为“创建工单”。
5. 单击页面上方的，保存设置。

步骤5 为操作列增加“删除”功能。


1. 选中“表格”，在右侧“属性”页签的“表格列”中，单击上图中“操作”列后的.

图 3-242 编辑操作列



2. 在“属性配置”对话框中，单击“添加操作按钮”，新增后按钮在“派单”按钮下方，然后对新增按钮进行以下设置。
 - a. 向下拖拽拖动条，找到新增的按钮区域，设置按钮“标签”为“删除”，“图标”选择“delete”。

图 3-243 为操作列增加派单按钮




- b. 单击图3-243中“删除”按钮的 \oplus ，再单击“添加动作”的“+”，进入事件编排中，在事件编排页面，添加如下事件代码，然后单击“创建”。

须知


脚本中红色内容请替换为实际的页面名、组件名。

```
let _component = context.$component.current
let currentRow = _component.$attrs.row;
let id = currentRow.id;
let _model = $model.ref("workOrderInstance");
context.$dialog.confirm(
  {
    title: '确认',
    content: '确认删除该工单吗？',
    okText: '确定',
    cancelText: '取消',
    onOk: () => {
      $model.ref('deleteWorkOrder').setData({ inputParam: { "id": id } });
      $model.ref('deleteWorkOrder').run().then(function (data) {
        let pageInfo = $model.ref('table_0_condition').getData().pageInfo;
        let queryData = {
          "start": (pageInfo.curPage - 1) * pageInfo.pageSize,
          "limit": pageInfo.pageSize
        }
      })
    }
  }
);
```

```
});  
$model.ref('queryWorkOrder').setData({ inputParam: queryData });  
$model.ref('queryWorkOrder').run().then(function (data) {  
  $model.ref('workOrderInstance').setData(data.workOrderList);  
  pageInfo.total = parseInt(data.total);  
  $model.ref('table_0_condition').setData({ "pageInfo": pageInfo });  
}).catch(function (error) {  
  console.log('error is', error);  
});  
});  
},  
onCancel: () => {}  
}  
);
```

- c. 单击“确定”，关闭“属性配置”对话框。
- d. 单击页面上方的，保存设置。

步骤6 检查页面加载事件是否跟随页面复制过来。


1. 选中最外层的“页面”，在右侧“事件”页签中，单击“自定义JS代码”后的。

2. 查看自定义动作中是否已有如下脚本代码，如果没有，请输入。

```
let pageInfo = $model.ref('table_0_condition').getData().pageInfo;  
let queryData = {  
  "start": (pageInfo.curPage - 1) * pageInfo.pageSize,  
  "limit": pageInfo.pageSize  
};  
$model.ref('queryWorkOrder').setData({ inputParam: queryData });  
$model.ref('queryWorkOrder').run().then(function (data) {  
  $model.ref('workOrderInstance').setData(data.workOrderList);  
  pageInfo.total = parseInt(data.total);  
  $model.ref('table_0_condition').setData({ "pageInfo": pageInfo });  
}).catch(function (error) {  
  console.log('error is', error);  
});
```

3. 单击页面上方的，并退出事件编排。

步骤7 添加“创建工单”按钮跳转事件，跳转事件功能是将页面跳转到“生成工单”页面。

1. 选中“创建工单”按钮，在右侧“事件”页签中，单击“新增行”后的，进入事件编排页面。

2. 在“自定义动作”中，输入以下事件代码，单击“创建”，加粗斜体内容请替换为实际命名空间前缀。

```
//打开创建工单页面，需要根据实际页面名称修改  
context.$page.load('/besBaas/page#/HW_createWorkOrder')
```

3. 单击页面上方的，并退出事件编排。

----结束

3.1.7.7 定义工单流转 BPM

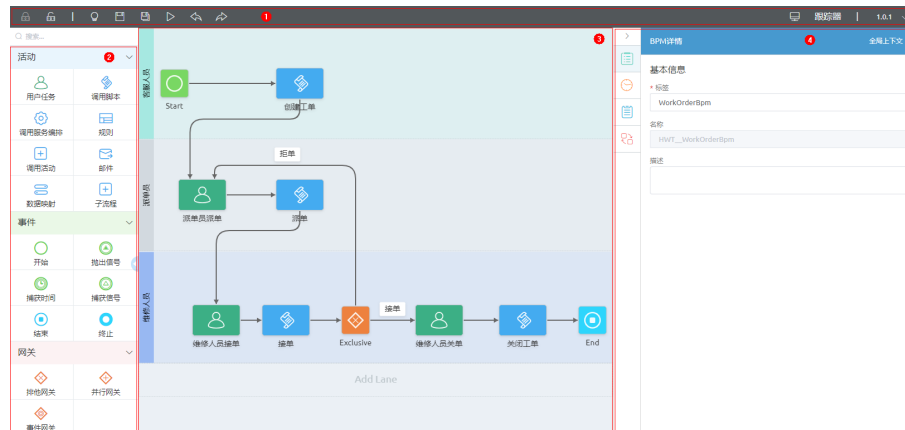
3.1.7.7.1 背景与原理（BPM）

工单管理模块中的工单场景业务编排是通过AstroZero的流程编排BPM（Business Process Management）功能实现的，通过在前端页面调用BPM完成工单流转，即客服人员创单，派单员派发工单，维修工程师处理工单的全过程。

开发BPM即是对前端页面及后端逻辑（脚本、服务编排等）进行编排的过程。AstroZero提供的BPM作为商业流的配置工具，可以通过模板化、图形化实现对商业流业务流程的编排和执行。

了解 BPM 设计界面

图 3-244 BPM 设计界面



整体编辑器页面由上方按钮区域、左侧面板图元区域、中间画布工作区域、右侧属性配置区域四部分组成。

表 3-32 BPM 设计界面说明

编号区域名称	功能说明
1	功能按钮区域，包括锁定、解锁、启用（或者禁用）、保存、另存为新版本或者新BPM、运行、操作回退、撤销回退、启用流跟踪器以及切换版本的操作。支持快捷键操作，即可脱离鼠标直接用键盘操作。
2	BPM的组成图元，一个BPM业务流程由以下几个部分组成： <ul style="list-style-type: none"> 事件图元（Events）：用来表明BPM的生命周期中发生的事件，例如开始、捕获信号等。 网关图元（Gateways）：网关用来控制流程的执行流向，可理解为决策、判断。 活动图元（Activities）：是BPM的核心图元，可理解为节点或者步骤，例如调用脚本、用户需要做的任务。
3	BPM设计操作区域。在该区域可对BPM进行具体流程设计、组件放置。 <ul style="list-style-type: none"> 不同色块的表示不同的泳道，BPM由一个或多个泳道组成，泳道中包括了实现不同功能逻辑的图元。 选中泳道或者某个图元，可以在右侧属性配置区域进行各种配置操作。

编号区域名称	功能说明
4	<p>整个BPM、泳道或者图元属性设置区域。</p> <ul style="list-style-type: none">• 当选择BPM中图元时，右侧配置区域为该图元的属性设置区域。• 当选择BPM中空白区域时，右侧配置区域为该BPM的设置区域。• 当选择泳道左侧标签时，右侧配置区域为该泳道的属性设置区域。

BPM 能力

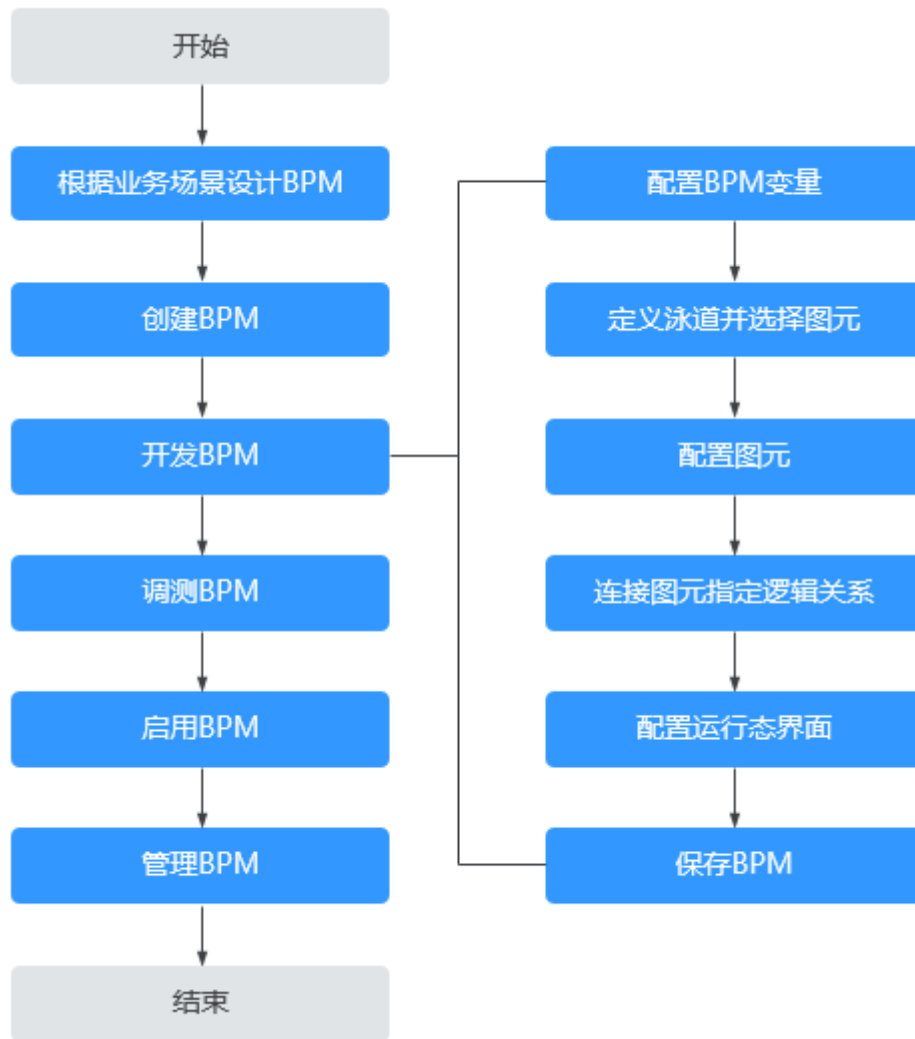
BPM可以提供如下能力，本示例中仅使用到前三种能力：

- 端到端页面流编排
将用户交互的前端页面与各种任务之间进行编排，形成一个完整的商业流。
- 支持长流程
步骤之间可以是立即执行，也可以是小时、天、甚至更长时间的间隔，支持SLA期限管理。
- 跨人员的工作流
支持每一步由不同的用户、组串行或并行处理。
- BPM内部进行调用
BPM可以作为子BPM被其他BPM进行内部调用。在总的BPM中使用“调用活动”元素，可嵌套使用子BPM。

BPM 开发流程

BPM开发的一般流程如[图3-245](#)所示。

图 3-245 BPM 开发配置大致流程



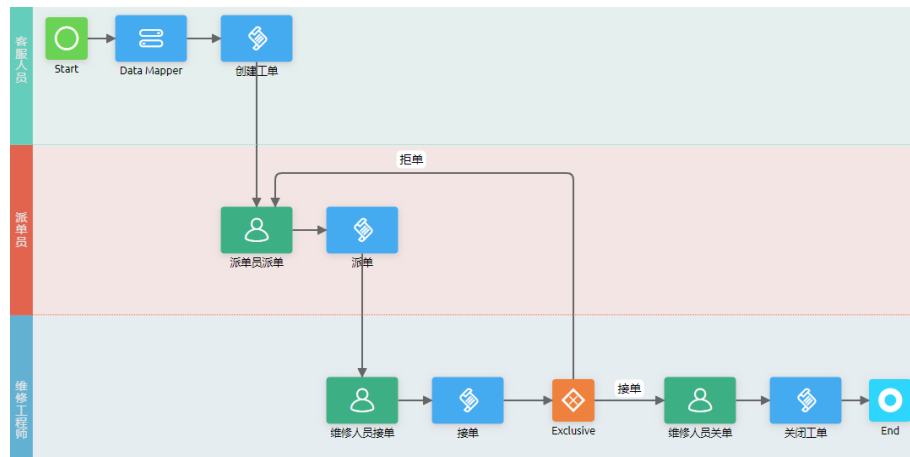
更多BPM详细介绍，请参见[Astro轻应用产品文档](#)中相关描述。

3.1.7.7.2 创建并开发 BPM

工单管理开发中，工单流转BPM设计大致如下。

1. 根据业务角色不同，使用不同的工作队列，创建客服人员、派单员以及维修人员3个泳道。

图 3-246 工单 BPM 设计界面



通过在不同的泳道中，拖拽功能图元，然后通过配置图元属性，以及需要调用关键脚本逻辑，将前端页面中获取到的数据传回到后端的工单对象中，以完成工单流转的全过程。图元之间调用关系及工单流转过程，如表3-33所示。

表 3-33 图元调用关系及工单流程过程

序号	图元名称	图元类型	图元作用	下一步流转
1	Start	开始	用户在创建工单页面，填写工单信息并单击“提交”按钮后，触发该图元，并将工单信息传入。	携带工单信息流转到“Data Mapper”图元。
2	Data Mapper	数据映射	将当前BPM的ID赋值给工单对象中BPM的ID字段。	携带工单信息、BPM的信息流转到“创建工单”图元。
3	创建工单	调用脚本	调用“创建工单”脚本，根据传入的工单信息，完成工单创建。	工单创建完毕后，流转“派单员派单”图元。
4	派单员派单	用户任务	将创建成功的工单，流转到派单员的派单页面（派单员队列中的用户均可进行派单）。	单击“派单”按钮，选择需要指定的“维修人员”后，将派单信息流转到“派单”图元。
5	派单	调用脚本	调用“派发工单”脚本，根据传入的派单信息进行工单信息（当前处理人/状态）的修改。	“派发工单”脚本执行后，会返回维修人员的用户Id，工单将流转到“维修人员”图元。

序号	图元名称	图元类型	图元作用	下一步流转
6	维修人员接单	用户任务	将派发成功的工单流转到指定的维修人员的工单处理界面。	单击“处理”按钮，进行工单的处理，可选择“接单”或“拒单”，并将处理信息流转到“接单”图元。
7	接单	调用脚本	调用“处理工单”脚本，根据传入的处理信息进行工单信息（当前处理人/状态）的修改。	“处理工单”脚本执行后返回处理结果，并将处理结果流转到“Exclusive” 排他网关图元，进行下一步判断。
8	Exclusive	排他网关	根据脚本返回的处理结果进行判断，执行不同的流程。	<ul style="list-style-type: none"> • 如果脚本返回结果为“接单”，将流转到“维修人员接单”图元。 • 如果脚本返回结果为“拒单”，将重新流转回“派单员派单”图元，重新进行派单。
9	维修人员关单	用户任务	将接单成功的工单流转到接单的维修人员的工单处理界面。	对于处理中状态的工单，单击“处理”按钮，进行工单的处理，即进行关闭操作，将处理信息流转到“关闭工单”图元。
10	关闭工单	调用脚本	调用处理工单脚本，根据传入的处理信息，将工单状态修改为关闭。	关闭工单成功后，将流转到结束图元。
11	End	结束	整个流程执行完毕。	-

操作步骤

创建流程大致是先创建3个泳道，再拖拽泳道图元，然后配置相关图元，保存并启用。

步骤1 在“我的应用”中，单击“设备维修管理系统”，进入应用。

步骤2 在“WorkOrder”目录中，将鼠标放在“Bpm”上，单击界面上出现的“+”，在弹出菜单中选择“BPM”。

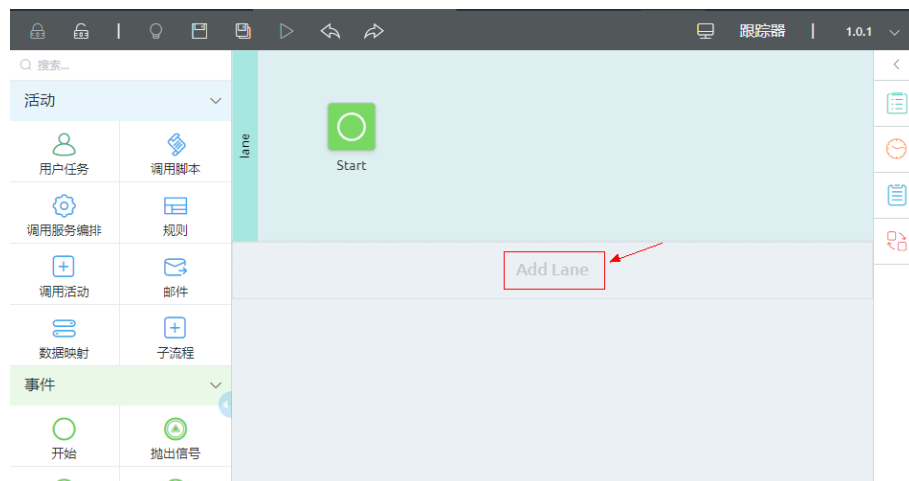
步骤3 在弹窗中，设置“标签”、“名称”为“WorkOrderBpm”，单击“添加”。

图 3-247 添加 BPM



步骤4 在设计区域，单击2次“AddLane”，为BPM添加2个角色泳道。

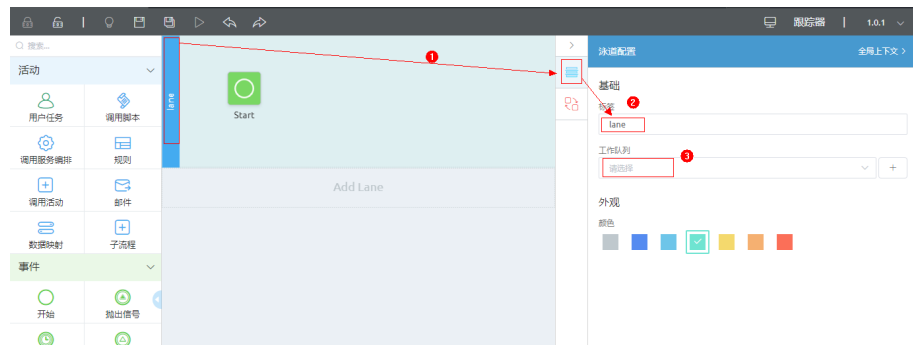
图 3-248 添加泳道



步骤5 修改泳道基础属性。

1. 选中“Lane”，在右侧属性配置区域，设置泳道标签为“客服人员”，“工作队列”设置为**创建工作队列**中创建的“客服人员”。

图 3-249 修改客服人员泳道基本信息



2. 参考上一步，分别设置其他两个泳道的“标签”及“工作队列”。

表 3-34 泳道标签及工作队列

泳道初始标签	泳道修改后标签	工作队列
Lane (上一步已配置)	客服人员	客服人员
lane1	派单员	派单员
lane2	维修人员	维修人员

步骤6 创建BPM中用的变量及对象变量。


1. 在右侧属性面板区域，单击 ，展开全局上下文。

图 3-250 展开全局上下文





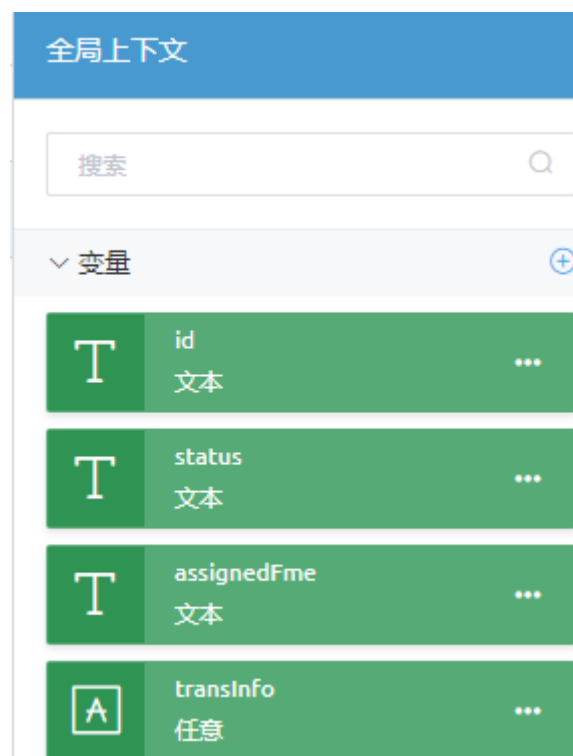
2. 单击“变量”后的  图标4次，分别创建以下变量，单击变量后的 ，可以修改字段类型，创建完成后，如图3-251所示。

表 3-35 BPM 变量

变量名	字段类型
id	文本
status	文本
assignedFme	文本
transInfo	任意

图 3-251 创建变量



- 单击“对象变量”的 \oplus ，在弹窗中设置“名称”为“workOrderData”，“对象”设置为“***HW_WorkOrder_CST***”（工单对象），加粗斜体内容请以实际命名空间前缀为准，然后单击“保存”。

图 3-252 新增对象变量

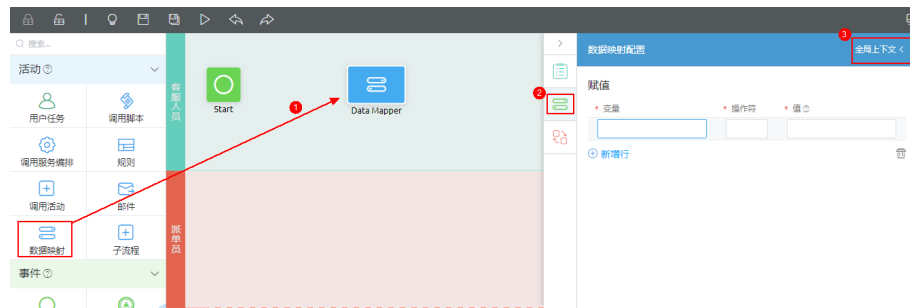


- 再次单击“对象变量”的⁺，在弹窗中设置“名称”为“statusInfo”，“对象”设置为“HW_WorkOrder_CST”（工单对象），然后单击“保存”。

步骤7 拖拽并配置“客服人员”泳道。

- 从左侧“活动”图元区域，拖拽一个“数据映射”图元到“客服人员”泳道中。
- 选中“Data Mapper”图元，然后在右侧导航菜单上单击[≡]，进入数据映射配置，再单击[🔄]，展开全局上下文。

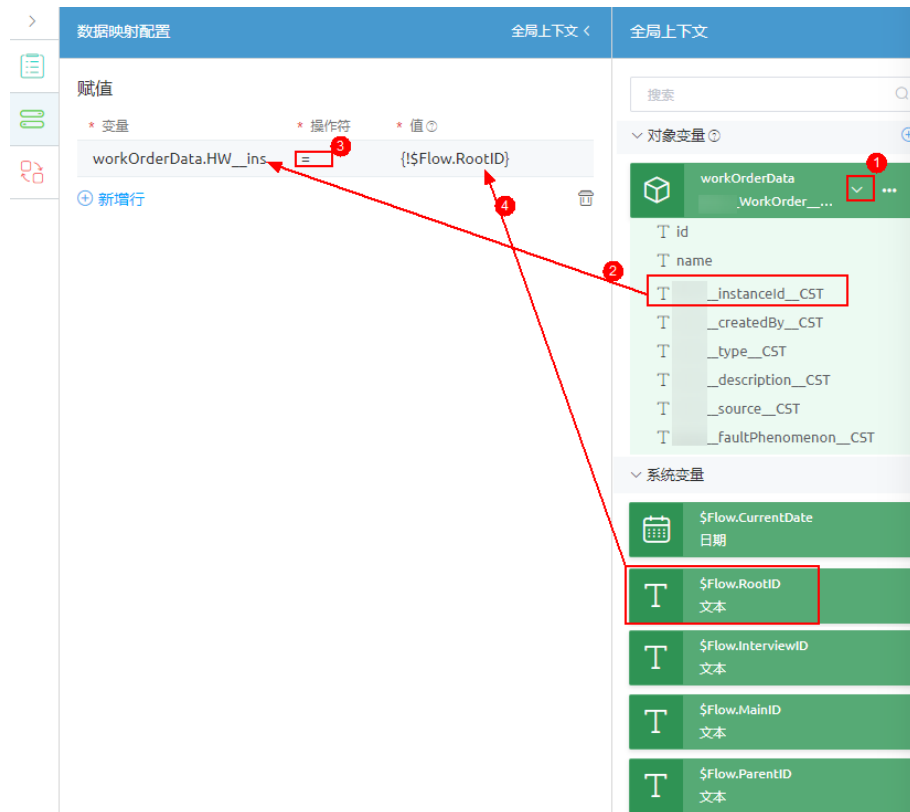
图 3-253 拖拽并设置“数据映射”图元



- 分别从全局上下文的“对象变量”、“系统变量”中拖拽参数到数据映射配置下：

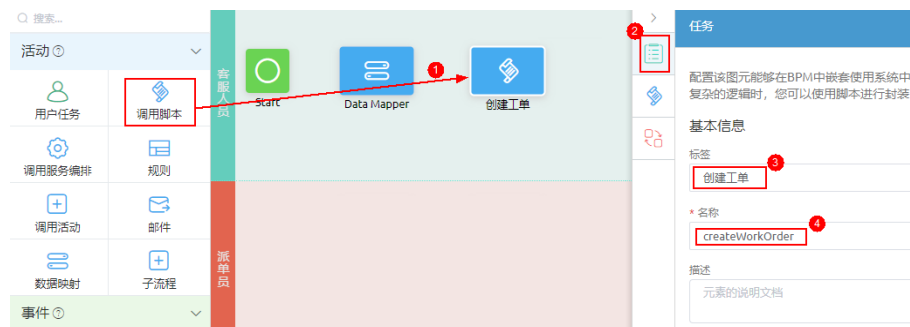
拖拽对象变量“workOrderData”下的“HW_instancelid_CST”字段到“变量”下，“操作符”为“=”，拖拽“系统变量”下的“{!\$Flow.RootID}”做为“值”。

图 3-254 拖拽赋值变量



4. 从左侧“活动”图元区域，拖拽一个“调用脚本”图元到“客服人员”泳道中，并设置“标签”为“创建工单”、“名称”为“createWorkOrder”。

图 3-255 设置创建工单图元




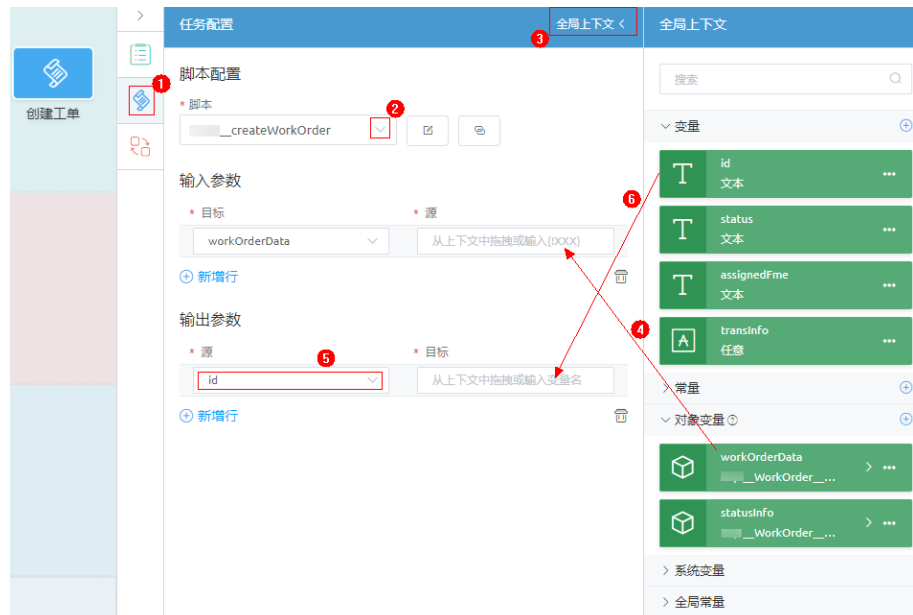
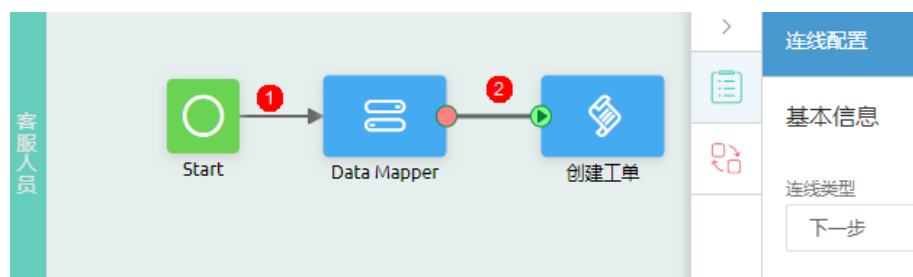
5. 选中“创建工单”图元，单击 ，再单击“全局上下文”，配置“创建工单”图元绑定的脚本以及输入输出参数。
 - a. 在“脚本配置”下选择“生成工单”脚本“HW_createWorkOrder”。
 - b. 从全局上下文中拖拽对象变量“workOrderData”到“输入参数”的“源”输入框中，作为输入参数的值。
 - c. 设置“输出参数”为“id”，并拖拽变量“id”作为“目标”输入框中的值。

图 3-256 配置“创建工单”图元



6. 从“Start”图元中拖拽一条连线到“Data Mapper”图元，然后从“Data Mapper”图元拖拽一条连线到“创建工单”图元，默认“连线类型”为“下一步”。

图 3-257 拖拽连线



步骤8 拖拽并配置“派单员”泳道。


1. 从左侧“活动”图元区域，依次拖拽一个“用户任务”、“调用脚本”图标到“派单员”泳道中。
2. 设置“用户任务”图元的“标签”为“派单员派单”、“名称”为“dispatchWorkOrderList”。
3. 设置“调用脚本”图元的“标签”为“派单”、“名称”为“dispatch”。
4. 选中“派单员派单”图元，单击，设置“任务标题”、“任务描述”为“派单员派单”，“渲染类型”为“标准页面”，“页面”设置为“HW_dispatchWorkOrder”。

图 3-258 用户任务配置

The screenshot displays the 'User Task Configuration' (用户任务配置) page. The left sidebar contains navigation icons for home, user, settings, and other functions. The main content area is titled '用户任务配置' and includes a '全局上下文' (Global Context) link. The configuration fields are as follows:

- 界面 (Interface):**
 - 任务标题 (Task Title): 派单员派单
 - 任务描述 (Task Description): 派单员派单
 - 优先级 (Priority): 正常
 - 渲染类型 (Render Type): 标准页面 (Standard Page)
 - 页面 (Page): HW_dispatchWorkOr
- 接收人 (Receiver):**
 - * 类型 (Type): 当前泳道 (Current Lane)
 - Options:
 - 当前泳道角色的任意成员 (Any member of the current lane role)
 - 当前泳道中的上一个任务被分配的人员 (Person assigned the previous task in the current lane)
 - 审批类型 (Approval Type): 任何人均可审批 (Anyone can approve)


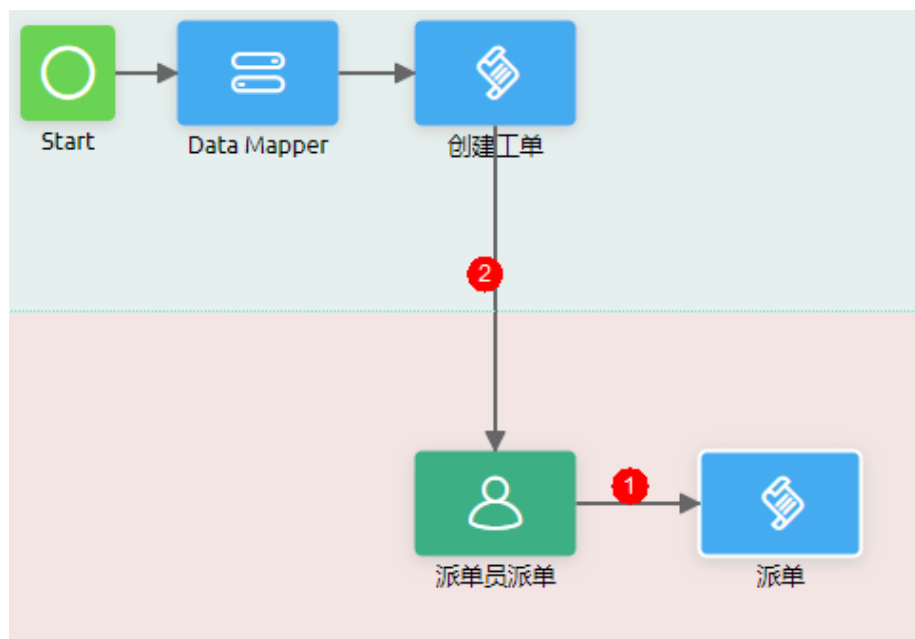
5. 选中“派单”图元，单击 ，再单击“全局上下文”，配置“派单”图元绑定的脚本以及输入输出参数。
 - a. 在“脚本配置”下选择“生成工单”脚本“HW_dispatchWorkOrder”。
 - b. 从全局上下文中，拖拽变量“transInfo”到“输入参数”的“源”输入框中，作为输入参数的值。
 - c. 在“输出参数”下，单击“新增行”，设置“输入参数”为“id”、“assignedFme”，并从全局上下文中，拖拽变量“id”、“assignedFme”作为“目标”输入框中的值。

图 3-259 配置“派单”图元



- d. 从“派单员派单”图元中拖拽一条连线到“派单”图元，然后再从“创建工作单”图元中拖拽一条连线到“派单员派单”图元，默认“连线类型”均为“下一步”。

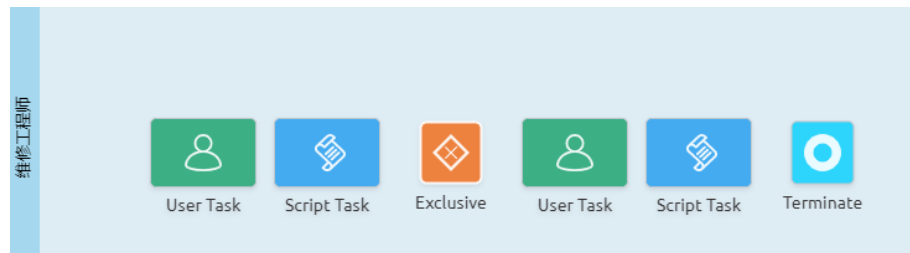
图 3-260 拖拽连线



步骤9 拖拽并配置“维修人员”泳道。

- 1. 从左侧“活动”图标区域，拖拽2个“用户任务”、2个“调用脚本”、1个“排他网关”以及“终止”图元到“维修人员”泳道中，然后调整图元顺序。

图 3-261 拖拽并调整图元顺序

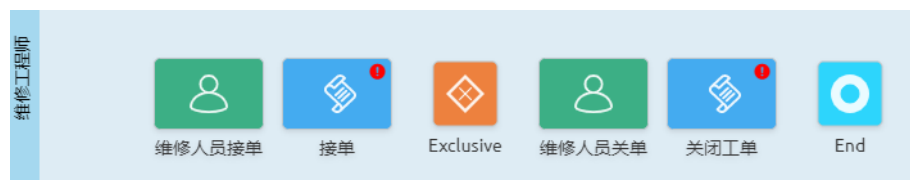


2. 从左到右，分别设置2个“用户任务”、2个“调用脚本”图元的基本属性，具体如表3-36所示，设置完成后如图3-262所示。

表 3-36 图元基本属性设置

图元（从左向右）	标签	名称
用户任务1	维修人员接单	takeWorkOrder
调用脚本1	接单	takeWorkOrder1
用户任务2	维修人员关单	dealWorkOrder
调用脚本2	关闭工单	closeWorkOrder
终止	End	end

图 3-262 图元基本属性设置



3. 配置“维修人员接单”图元。


选中“维修人员接单”图元，单击，并展开全局上下文，配置“维修人员接单”图元，详细配置如表3-37所示。


图 3-263 用户任务配置



表 3-37 用户任务配置

参数项	值
任务标题	维修人员接单
任务描述	维修人员任务列表
渲染类型	标准页面
页面	<i>HW_workOrderListM</i>
接收人下“类型”	名称和表达式
参与者下“类型”	表达式
参与者下“取值”	“变量”下的“assignedFme” 说明 请直接从全局上下文拖拽“assignedFme”到“取值”下，请勿手动输入，手动输入的值系统可能不识别。

4. 配置“接单”图元。

选中“接单”图元，单击，再单击“全局上下文”，并展开全局上下文，配置“维修人员接单”图元，详细配置如表3-37所示。

说明

请直接从全局上下文拖拽“statusInfo”、“id”、“status”到输入输出参数对应“源”或“目标”下，请勿手动输入，手动输入的值系统可能不识别。

图 3-264 配置“接单”图元

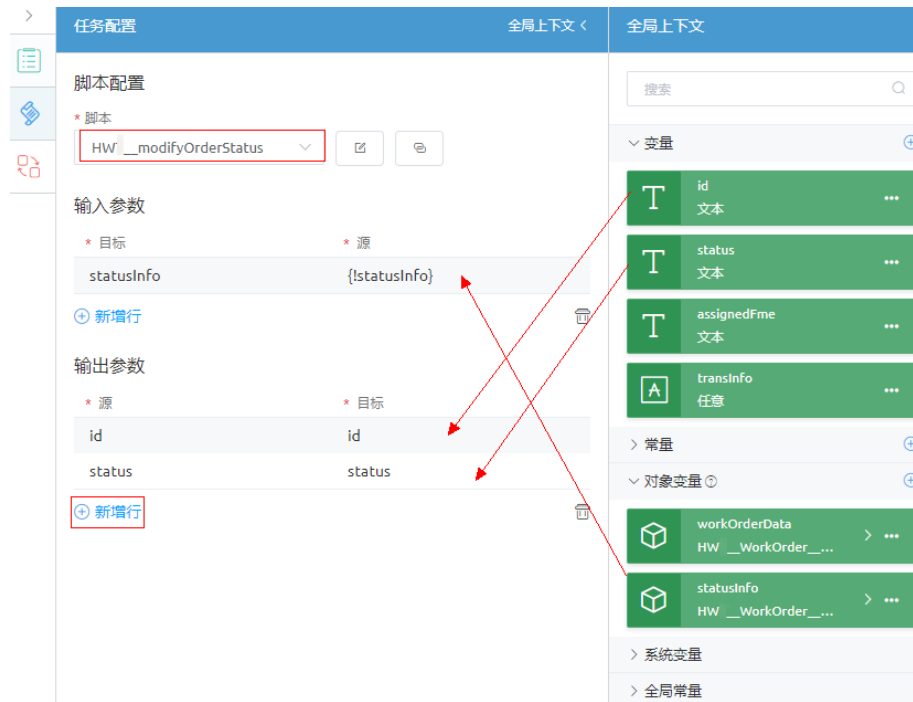


表 3-38 接单图元配置

参数项	值
脚本	<i>HW__modifyOrderStatus</i>
输入参数“statusInfo”	对象变量下“statusInfo”
输出参数“id”	变量下“id”
输出参数“status”	变量下“status”

5. 配置“维修人员关单”图元。


选中“维修人员关单”图元，单击，设置“任务标题”为“维修人员关单”，“任务描述”为“维修人员任务列表”，“渲染类型”为“标准页面”，“页面”设置为“*HW_workOrderListM*”，并设置“当前泳道中的上一个任务被分配的人员”，如图3-265所示。

图 3-265 配置“维修人员关单”图元

用户任务配置 全局上下文 <

界面

任务标题
维修人员关单

任务描述
维修人员任务列表

优先级
正常

渲染类型
标准页面

页面
HW_workOrderListM


接收人

* 类型
当前泳道

当前泳道角色的任意成员

当前泳道中的上一个任务被分配的人员

6. 配置“关闭工单”图元。

选中“关闭工单”图元，单击，再单击“全局上下文”，并展开全局上下文，配置“关闭工单”图元，详细配置如表3-37所示。

说明

请直接从全局上下文拖拽“statusInfo”、“id”、“status”到输入输出参数对应“源”或“目标”下，请勿手动输入，手动输入的值系统可能不识别。

图 3-266 拖拽关闭工单输入输出值

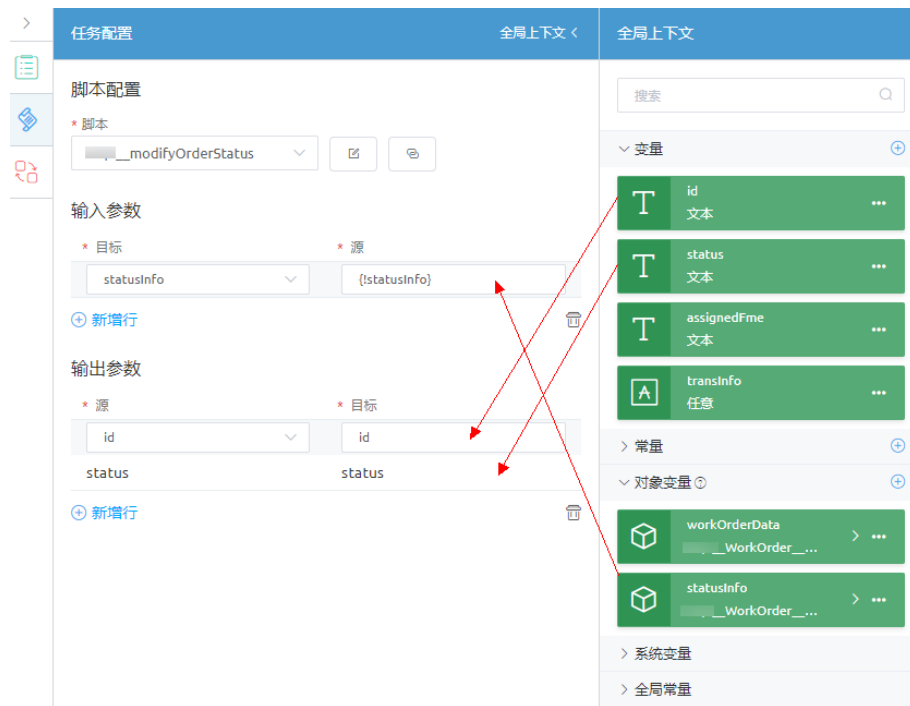
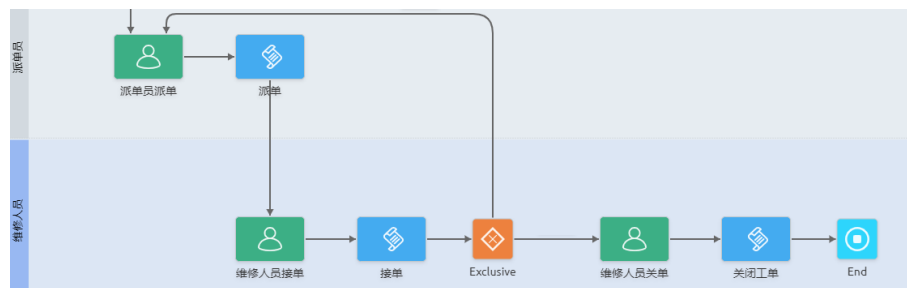


表 3-39 接单图元配置

参数项	值
脚本	HW_modifyOrderStatus
输入参数“statusInfo”	对象变量下“statusInfo”
输出参数“id”	变量下“id”
输出参数“status”	变量下“status”

步骤10 拖拽图元直接的关系连线，具体如图3-267所示。

图 3-267 添加图元之间连线



步骤11 配置“排他网关”图元连线属性。

- 选中“派单员派单”与“排他网关”之间的连线，单击鼠标右键，选择“配置”，设置“标签”为“拒单”，再单击“新增行”，拖拽变量“status”到“资源”下，设置“比较符”为“==”，“值”为“拒单”。

图 3-268 选中排他网关图元连线

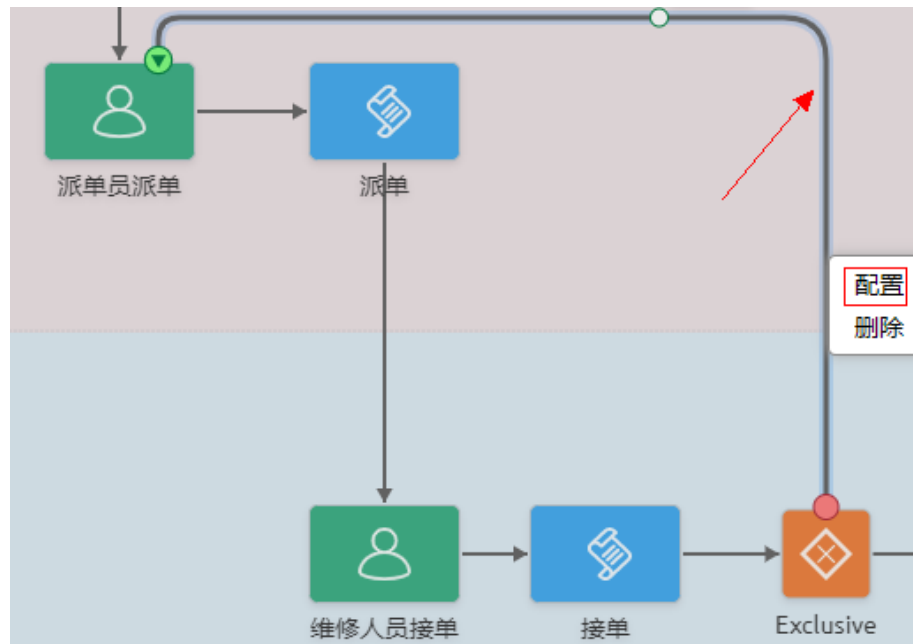


图 3-269 配置“拒单”连线



2. 选择“排他网关”图元与“维修人员关单”图元之间的连线，参考上一步，设置连线，设置“标签”为“接单”，再单击“新增行”，拖拽变量“status”到“资源”下，设置“比较符”为“==”，“值”为“接单”。

图 3-270 配置“接单”连线



步骤12 单击页面上方的, 再单击, 启用BPM。

----结束

3.1.7.8 验证工单管理功能

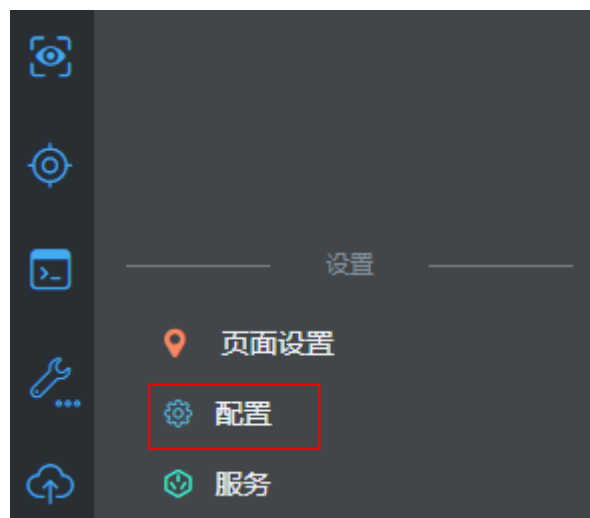
3.1.7.8.1 挂载前端页面

应用开发完成后, 要将页面挂载到导航条上, 作为应用菜单, 在应用预览时, 查看应用相关页面。应用菜单仅支持开发者号的管理员查看, 业务号只能查看对应权限的页面。

操作步骤

步骤1 在经典版应用开发页面, 单击左侧导航栏下方的“配置”。

图 3-271 应用配置入口



步骤2 在“导航条”页签，单击“菜单树”右侧的“+”，选择“添加页签”。

图 3-272 导航条添加页签



步骤3 定义“设备管理”页签。

在“添加页签”弹窗中，设置以下信息，然后单击“保存”。

- 页面类型：设置为“标准页面页签”。
- 标签：设置为“设备管理”。
- 名称：设置为“equipmentManage”。
- 页面：设置为“*HW*_equipmentManage”。

图 3-273 设备管理页签

编辑页签
✕

* 页签类型 ▼

标准页面页签

显示区域



主页菜单

✓



自定义菜单栏

打开方式 ▼

当前窗口

* 标签

设备管理

名称

equipmentManage

图标



* 页面 ▼

equipmentManage

描述

取消

保存

步骤4 请按照以上方式，创建表3-40的导航菜单。

表 3-40 导航菜单

页签类型	标签	名称	页面
标准页面页签	工单管理	manageWorkOrderList	HW_manageWorkOrder
标准页面页签	工单列表（客服人员）	workOrderList	HW_workOrderList
标准页面页签	工单列表（派单员）	dispatchWorkOrder	HW_dispatchWorkOrder

页签类型	标签	名称	页面
标准页面页签	工单列表（维修人员）	workOrderListM	HW_workOrderListM
高级页面页签	设备监控	monitor	HW_monitor

步骤5 （可选）设置导航的布局及颜色。

您可以根据自己的喜好，设置导航的布局及颜色，默认导航菜单是靠左，蓝色。

如**图3-274**所示，单击红框中的布局，即可将导航菜单设置为横向菜单，然后单击“保存”。

图 3-274 切换为横向导航



----结束

3.1.7.8.2 验证

本节主要是通过不同的页面上操作工单流程，验证工单管理功能中的各个功能。

操作步骤

步骤1 在经典版应用开发页面，单击左侧导航栏下方的，进入应用预览页面。

图 3-275 查看应用



步骤2 验证创建工单。

1. 选择“工单列表（客服人员）”，然后在工单列表（客服人员）页面，单击“创建工单”，进入创建工单页面。
2. 在“设备名称”下拉框中选择一个设备，检查“设备详情”区域显示了对应的设备详情。显示正确，则说明组件与模型的绑定，以及下拉框“数据改变”事件执行正确。

说明

如果“设备名称”下，没有设备名，请先在“设备管理”页面，单击“新增行”进行添加。

3. 检查是否正常提交工单。
填写工单信息，单击“提交”。如果页面跳转到工单列表页面，且在查询结果中，显示新创建的工单，则说明提交按钮的“点击”事件执行正确。
如果页面跳转到工单列表页面，且在查询结果中显示新创建的工单，则说明验证成功。

图 3-276 验证结果样例



步骤3 验证派单功能。

1. 单击“工单列表（派单员）”，进入“工单列表（派单员）”页面，查看页面是否显示上一步创建的工单，且每条记录后都有派单图标。
2. 选择上一步新建的一条工单记录，并单击该记录的 **派发**，弹出处理工单弹窗。

图 3-277 状态为“待派单”工单记录



3. 在处理工单弹窗中，设置下拉框“选择工程师”为当前租户账号，然后单击“确定”，返回“工单列表（派单员）”页面。

查看工单记录的“状态”、“当前处理人”是否已更新为“待接单”、当前租户名。如果已更新，说明派单流程正常。如未改变请检查“派单”按钮上事件代码以及“处理工单”弹窗中数据绑定及事件代码。

步骤4 验证待接工单。

1. 进入“工单列表（维修人员）”页面，检查系统显示派单员刚刚派的工单，如图 3-278 所示。

如未正常显示请检查当前页面标签组件的属性值绑定及页面事件代码。

图 3-278 待处理工单



2. 单击“处理”按钮，进入处理工单弹窗，“选择下一步操作”设置为“接单”，单击“提交”按钮。

处理完成后，返回“待处理工单”页面，查看“状态”是否已经更新为“处理中”。

如果有多条工单，“待处理”状态的工单，优先显示，“处理中”的工单可能会显示在页面的下面，工单状态改变后，可以拖动滚动条查找该工单。

说明

如果在“选择下一步操作”中，选择了“拒单”，流程将返回“派单员”处，状态将变为“待派单”。

图 3-279 处理工单-关单



图 3-280 查看处理中的工单



- 单击“处理中”工单的“处理”按钮，选择“关单”，单击“提交”，返回页面后，查看该条工单的状态是否变成“关闭”，如果已关闭，则说明关单流程正常。

图 3-281 关闭工单



步骤5 参考以上步骤，验证工单管理页面。

1. 在应用菜单中，选择“工单管理”页面，进入工单管理，查看页面工单列表中的工单信息。
2. 进行“创建工单”、派单及删除工单操作，验证工单管理功能。

----结束

3.1.8 用户管理功能开发

3.1.8.1 背景与知识

“用户管理”功能包括业务新增业务用户、查看删除业务用户、添加业务用户权限集三部分。

在业务场景中，会区分不同业务用户，业务用户对应了不同的用户权限，本示例应用中包含的业务用户，即是使用设备管理应用的用户，分别是客服人员、派单员及维修人员。

在AstroZero开发的的应用的所有业务用户，最终都会存储到平台的业务用户表中，即PortalUser表。业务用户在登录应用后的访问权限是通过Profile进行控制管理的，可基于平台预置的Portal User Profile进行自定义业务用户权限。

图 3-282 业务用户



3.1.8.2 开发业务逻辑

3.1.8.2.1 创建用户注册脚本

通过创建用户注册脚本，完成添加业务用户账户的业务逻辑，以供“业务用户注册”页面调用。

前提条件

参考[如何登录经典版环境配置](#)中操作，登录经典版环境配置，将“系统管理 > 系统参数 > 内置系统参数”中的参数“bingo.security.sensitive.data”取值修改为“是”（若为否，通过脚本创建出来的业务用户无法正常登录）。

图 3-283 修改“bingo.security.sensitive.data”的值为“是”



操作步骤

- 步骤1** 在“我的应用”中，单击“设备维修管理系统”，进入应用。
- 步骤2** 在“User”目录中，将鼠标放在“Script”上，单击界面上出现的“+”，在弹出菜单中选择“脚本”。
- 步骤3** 在弹窗中，选中“创建一个新脚本”，在“名称”文本框中输入“registerPortalUser”，单击“添加”。
- 步骤4** 在代码编辑器中，插入如下脚本代码。

```
import * as buffer from "buffer";
import * as crypto from "crypto";
import * as db from "db";
```

```
import * as context from 'context';
import * as http from 'http';
import * as permission from 'permission';

//定义入参结构, 包括注册账号的用户名、密码和角色, 为必填字段
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'String', required: true, label: 'string' })
  username: string;
  @action.param({ type: 'String', required: true, label: 'string' })
  password: string;
  @action.param({ type: 'String', required: true, label: 'string' })
  role: string;
}

//定义出参结构, 出参包含1个参数, portaluser的记录id
@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'String' })
  msg: string;
}

//使用数据对象PortalUser
@useObject(['PortalUser'])
@action.object({ type: "method" })
export class RegisterPortalUser { //定义接口类, 接口的入参为ActionInput, 出参为ActionOutput
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public registerPortalUser(input: ActionInput): ActionOutput {
    let out = new ActionOutput(); //新建出参ActionOutput类型的实例, 作pu为返回值
    let error = new Error(); //新建错误类型的实例, 用于在发生错误时保存错误信息
    try {
      let s = db.object('PortalUser');

      let saltedPassword = salt(input.password);
      let userMsg = {
        "usrName": input.username,
        "name": input.username,
        "userPassword": saltedPassword['saltedPassword'],
        "passwordSalt": saltedPassword['salt'],
        "userType": input.role
      };


      let userId = s.insert(userMsg);
      if (userId) {
        out.msg = "注册成功! ";
      } else {
        error.name = "USERERROR";
        error.message = "注册失败! ";
        throw error;
      }
    } catch (error) {
      if (error.name == "405230618") {
        error.message = "该用户名已注册! "
      }
      console.error(error.name, error.message);
      context.setError(error.name, error.message);
    }
    return out;
  }
}

function _salt(password: string, saltBuf: buffer.Buffer, encoding: buffer.Encoding = buffer.Encoding.Base64):
string {
  const passwordBuf = buffer.from(password)
  const crypt = crypto.pbkdf2(passwordBuf, saltBuf, 1000, 32, crypto.Hashs.SHA1)
  return crypt.toString(encoding)
}



function salt(password: string, encoding: buffer.Encoding = buffer.Encoding.Base64): object {
  const saltBuf = crypto.randomBytes(6)
```



```
const saltedPassword = _salt(password, saltBuf, encoding)
return {
  salt: saltBuf.toString(encoding),
  saltedPassword: saltedPassword
}
```

步骤5 单击编辑器上方的, 保存脚本。


步骤6 测试脚本能否正常执行。

1. 单击编辑器上方的, 执行脚本。
2. 在界面底部输入测试数据, 单击测试窗口右上角执行图标。

```
{
  "username": "test_cs",
  "password": "****",
  "role": "cs"
}
```

执行成功, 会在“输出”页签返回查询结果。

```
{
  "msg": "注册成功!"
}
```

步骤7 测试成功, 单击编辑器上方的, 启用发布脚本。

----结束

3.1.8.2.2 创建用户登录脚本

用户登录脚本是[组装“业务用户登录”页面](#)中, 自定义登录组件调用的业务逻辑, 之前创建的登录页只有前端页面, 本节将创建完整的登录业务逻辑。

操作步骤

步骤1 在“我的应用”中, 单击“设备维修管理系统”, 进入应用。

步骤2 在“User”目录中, 将鼠标放在“Script”上, 单击界面上出现的“+”, 在弹出菜单中选择“脚本”。

步骤3 在弹窗中, 选中“创建一个新脚本”, 在“名称”文本框中输入“login”, 单击“添加”。


步骤4 在代码编辑器中, 插入如下脚本代码。

```
import * as buffer from "buffer";
import * as crypto from "crypto";
import * as db from "db";
import * as context from 'context';
//定义入参结构, 包括账号的用户名、密码为必填字段,验证码为非必填字段
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'String', required: true, label: 'string' })
  username: string;
  @action.param({ type: 'String', required: true, label: 'string' })
  password: string;
  @action.param({ type: 'String', required: true, label: 'string' })
  captcha: string;
}
//定义出参结构, 出参包含5个参数, 登录结果和用户角色
@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'String' })
```



```
msg: string;
@action.param({ type: 'String' })
username: string;
@action.param({ type: 'String' })
userId: string;
@action.param({ type: 'String' })
captcha: string;
@action.param({ type: 'String' })
profile: string;
}
//使用数据对象PortalUser
@useObject(['PortalUser'])
@action.object({ type: "method" })
export class Login { //定义接口类, 接口的入参为ActionInput, 出参为ActionOutput
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public login(input: ActionInput): ActionOutput {
    let out = new ActionOutput(); //新建出参ActionOutput类型的实例, 作为返回值
    let error = new Error(); //新建错误类型的实例, 用于在发生错误时保存错误信息
    try {
      out.captcha = input.captcha;
      let s = db.object('PortalUser');
      let condition = {
        "conjunction": "AND",
        "conditions": [{
          "field": "usrName",
          "operator": "eq",
          "value": input.username
        }]
      };
      let user = s.queryByCondition(condition);
      if (user && user.length == 1) {
        if (validate(user[0].passwordSalt, user[0].userPassword, input.password)) {
          out.msg = "登录成功! ";
          out.username = user[0].usrName;
          out.userId = user[0].id;
          out.profile = user[0].userType;
        } else {
          out.msg = "密码错误! ";
        }
      } else {
        out.msg = "用户不存在! ";
      }
    } catch (error) {
      console.error(error.name, error.message);
      context.setError(error.name, error.message);
      out.msg = error.message;
    }
    return out;
  }
}

function _salt(password: string, saltBuf: buffer.Buffer, encoding: buffer.Encoding = buffer.Encoding.Base64):
string {
  const passwordBuf = buffer.from(password)
  const crypt = crypto.pbkdf2(passwordBuf, saltBuf, 1000, 32, crypto.Hashes.SHA1)
  return crypt.toString(encoding)
}

function validate(salt: string, userSaltedPassword: string, password: string, encoding: buffer.Encoding =
buffer.Encoding.Base64): boolean {
  const saltBuf = buffer.from(salt, encoding);
  const saltedPassword = _salt(password, saltBuf, encoding);
  return saltedPassword === userSaltedPassword
}
```

步骤5 单击编辑器上方的, 保存脚本。


步骤6 测试脚本能否正常执行。

1. 单击编辑器上方的 ，执行脚本。
2. 在界面底部输入测试数据，单击测试窗口右上角  执行图标，其中“test_cs”、变量“{XXXXXXXX}”为用户注册脚本中测试数据。

```
{
  "username": "test_cs",
  "password": "{XXXXXXXX}",
  "captcha": ""
}
```

执行成功，会在“输出”页签返回查询结果。

```
{
  "captcha": "",
  "msg": "登录成功!",
  "profile": "cs",
  "userId": "10gg0XXXXXXXXXXXX",
  "username": "test_cs"
}
```

步骤7 测试成功，单击编辑器上方的 ，启动发布脚本。

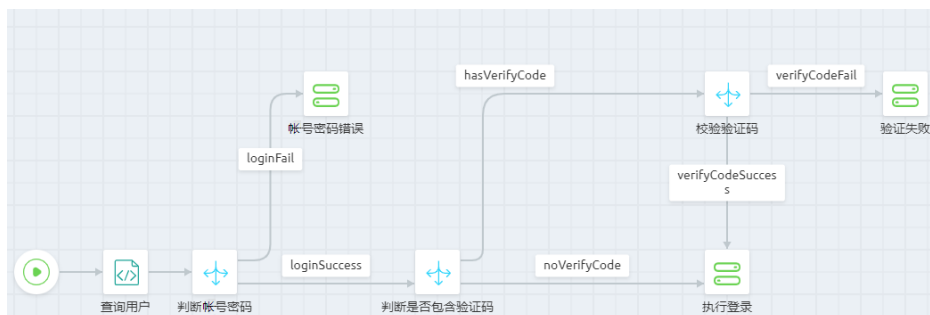
---结束

3.1.8.2.3 创建用户登录服务编排

用户登录服务编排通过调用脚本、编排图元等操作实现用户登录完整逻辑。

实现原理

图 3-284 用户登录服务编排大致设想



如图3-284所示，用户登录服务编排业务逻辑实现过程如下：

1. 通过调用“用户登录”脚本，查询登录账户密码，然后使用“决策”图元进行判断，判断当前登录的账号密码是否正确。
2. 如果判断账户密码错误，直接执行“账户密码错误”，判断账户密码正确，则使用“决策”图元，继续判断是否有验证码。
3. 如果判断当前登录没有验证码，则直接执行登录，判断当前有验证码，则继续判断验证码是否正确。
4. 如果判断验证码正确，则执行登录操作，判断验证码错误，则执行验证失败。

操作步骤

“用户登录”服务编排开发的大致过程：先拖拽1个脚本图元，3个决策图元以及3个赋值图元，再分别配置各个图元属性，然后配置各个图元之间连线类型，最后保存启用。



- 步骤1** 在“我的应用”中，单击“设备维修管理系统”，进入应用。
- 步骤2** 在“User”目录中，将鼠标放在“Flow”上，单击界面上出现的“+”，在弹出菜单中选择“服务编排”。
- 步骤3** 选中“创建一个新的服务编排”，在“标签”和“名称”文本框中输入“login”，并设置类型为“Autolaunched Flow”，单击“添加”。
- 步骤4** 定义服务编排用到的变量。
- 单击 ，展开全局上下文，再单击“变量”后的 ，设置参数名称为“username”，如 [图3-285](#)所示。

图 3-285 新增变量



- 重复上一步，定义其他变量。
需要定义的变量如 [表3-41](#)所示。

表 3-41 服务编排变量说明

变量名称（唯一标识）	数据类型
username（上一步已创建）	文本
password	文本
msg	文本
userId	文本
loginMsg	文本
captcha	文本
profile	文本


- 单击“公式”后的 ，在左侧公式弹窗中，设置“名称”为“portalUserLogin”，“表达式”为“PORTALUSERLOGIN({!username})”，单击“保存”。

图 3-286 添加公式变量 “portalUserLogin”



4. 参考上一步，创建公式变量 “verifyCode”，需要定义的变量如表3-42所示。

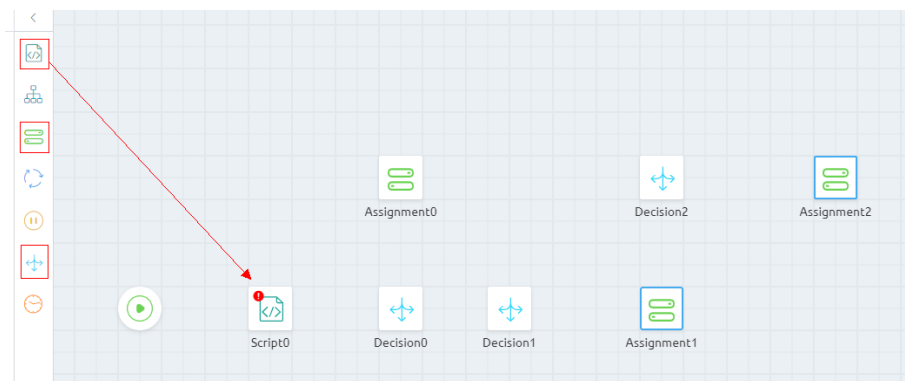
表 3-42 公式变量说明

名称	表达式
verifyCode	VERIFYCODEWITHTYPE(!captcha,"login")

步骤5 拖拽图元到服务编排画布，并配置图元的基本属性。

1. 从图元区分别拖拽脚本（1个）、决策（3个）、赋值（3个）图元到画布中，排列，如图3-287所示。

图 3-287 拖拽脚本（1个）、决策（3个）、赋值（3个）图元



2. 选中“Script0”图元，在右侧基本信息中，设置“标签”为“查询用户”。
3. 参考步骤5.2，设置其他图元的“标签”属性，具体值如表3-43所示。


表 3-43 图元基本属性

名称（变量唯一标识，不需要修改）	标签
Script0（上一步已设置）	查询用户
Decision0	判断账号密码
Decision1	判断是否包含验证码
Decision2	校验验证码
Assignment0	账号密码错误
Assignment1	执行登录
Assignment2	验证失败

图 3-288 修改后图元



步骤6 配置“查询用户”脚本图元。

1. 单击 ，指定图元对应的脚本名称（HW_login），并配置脚本的输入输出参数。
2. 单击“全局上下文”，显示变量列表，从“变量”中，拖拽“username”、“password”、“captcha”到“输入参数”下对应的“源”输入框中。
3. 在“输出参数”下，单击5次“新增行”，依次添加下拉选项中的输出参数字段，并从“变量”中拖拽相应的字段到“目标”输入框下，字段与变量对应关系如图3-289所示。

说明

请直接从全局上下文拖拽“变量”到对应的输入输出参数下，请勿手动输入，手动输入的值系统可能不识别。

图 3-289 拖拽脚本的输入输出参数



步骤7 配置“判断账号密码”决策图元。


1. 选择“判断账号密码”图元，在右侧单击  图标，修改“默认”的“名称”为“loginFail”。

图 3-290 修改“默认”结果名称



2. 单击“新增”，增加一个可编辑的结果，修改结果为“loginSuccess”，在“可视”下单击“新增行”，并拖拽变量中的“msg”到“资源”下，设置“比较符”为“==”，“值”为““登录成功!””。

📖 说明

- 请直接从全局上下文拖拽变量“msg”到“资源”下，请勿手动输入，手动输入的值系统可能不识别。
- “登录成功!”需要与“login”脚本中的输出参数一致。

图 3-291 修改可编辑的结果



步骤8 配置“判断是否包含验证码”决策图元。


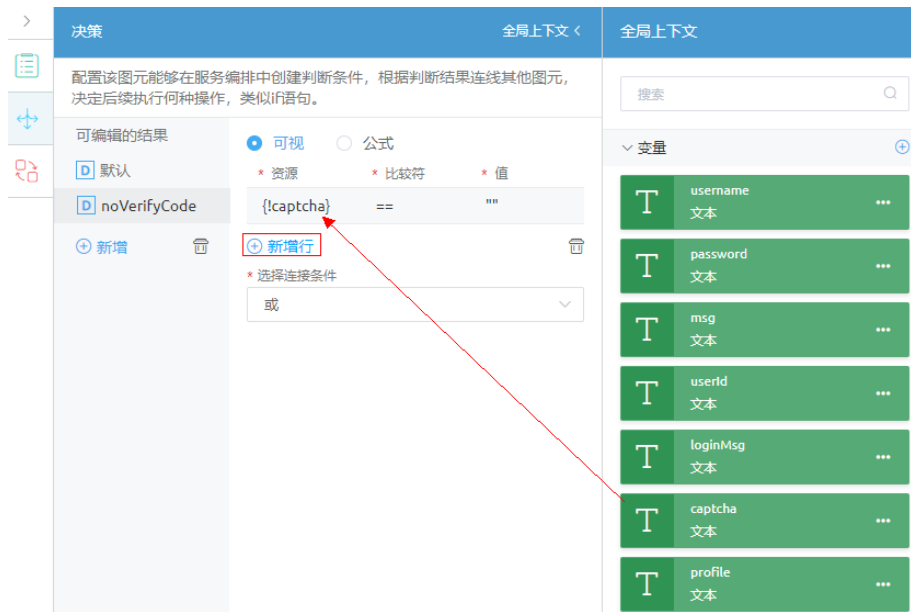
1. 选择“判断是否包含验证码”图元，在右侧单击  图标，修改“默认”的“名称”为“hasVerifyCode”。

图 3-292 修改“默认”结果名称



2. 单击“新增”，增加一个可编辑的结果，修改结果为“noVerifyCode”，在“可视”下单击“新增行”，并拖拽变量中的“captcha”到“资源”下，设置“比较符”为“==”，“值”为“”。

图 3-293 修改可编辑的结果



步骤9 配置“校验验证码”决策图元。

1. 选择“校验验证码”图元，在右侧单击图标，修改“默认”的“名称”为“verifyCodeFail”。

图 3-294 修改“默认”名称




2. 单击“新增”，增加一个可编辑的结果，修改结果为“verifyCodeSuccess”，在右侧选择“公式”，并从全局上下文中，拖拽“verifyCode”到“公式”下。

图 3-295 修改可编辑的结果



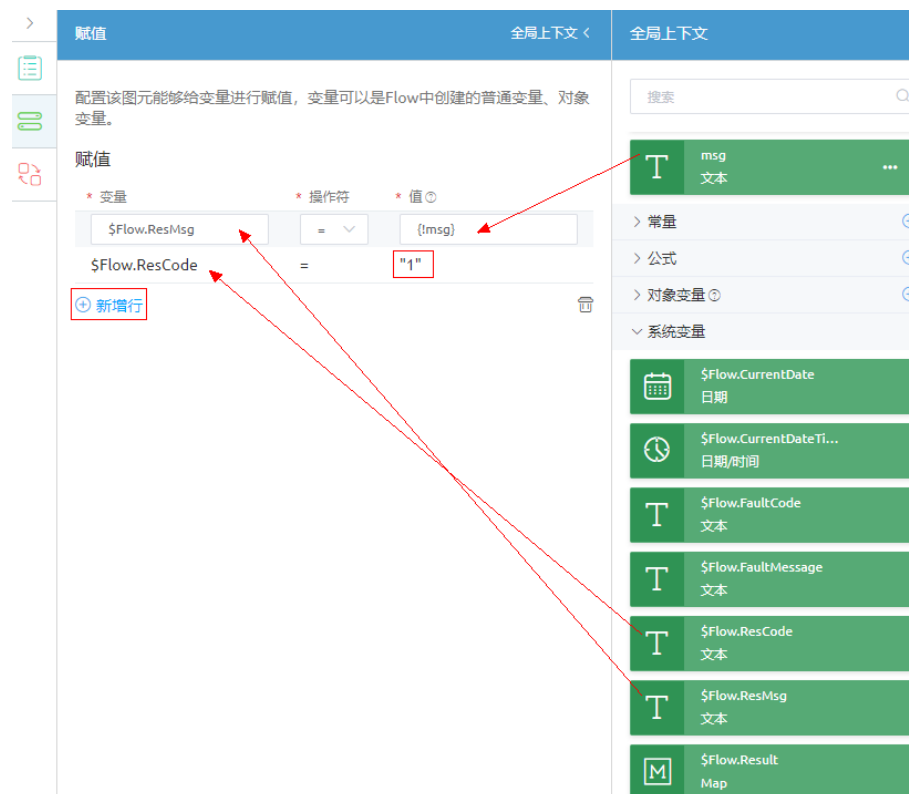
步骤10 配置“账号密码错误”赋值图元。

选择“账号密码错误”图元，在右侧单击  图标，单击“新增行”，从全局上下文的“系统变量”中，拖拽“\$Flow.ResMsg”到“赋值”下，并设置“操作符”为“=”，拖拽“msg”到“值”；然后再拖拽“系统变量”下的“\$Flow.ResCode”到“赋值”的“变量”下，设置“操作符”为“=”，设置“值”为“1”。

 说明

请直接从全局上下文拖拽变量到“赋值”下的对应位置，请勿手动输入，手动输入的值系统可能不记账。

图 3-296 配置“账号密码错误”图元



步骤11 配置“执行登录”赋值图元。


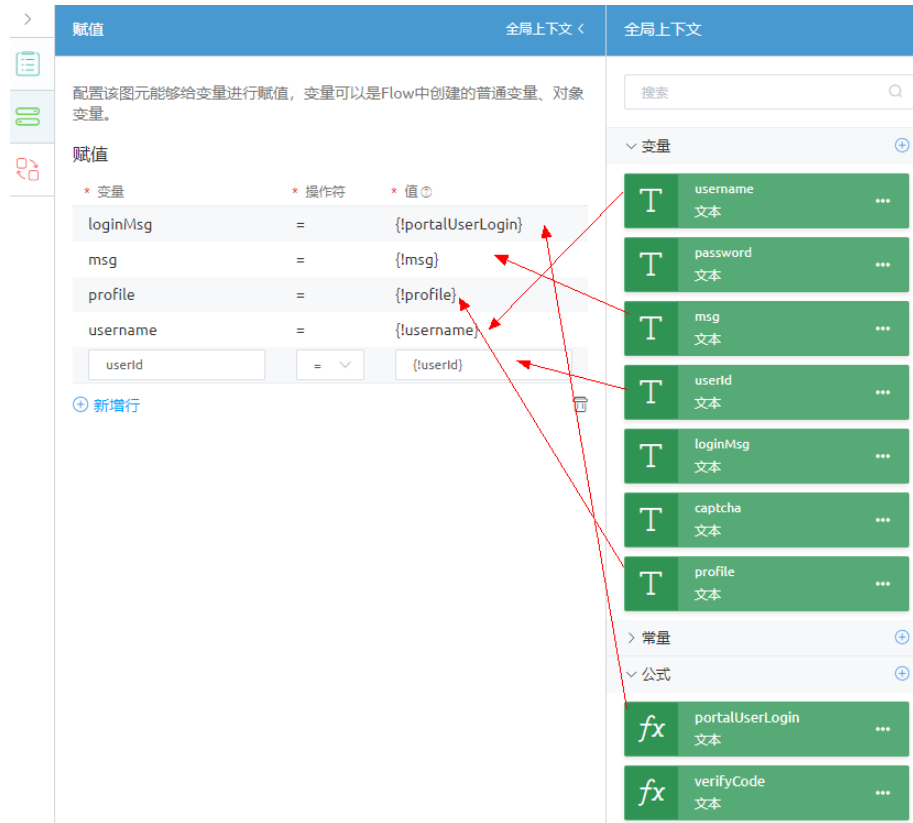
1. 选择“执行登录”图元，在右侧单击图标，单击5次“新增行”。
2. 从全局上下文，拖拽“msg”等字段到“赋值”的“变量”下，并设置“操作符”为“=”，然后再拖拽“值”下的各个值，具体字段对应关系，如图3-297所示。

图 3-297 拖拽“执行登录”赋值的变量及值



说明

请直接从全局上下文拖拽变量到“值”下的对应位置，请勿手动输入，手动输入的值系统可能不识别。

表 3-44 变量与值对应关系

变量	操作符	值
loginMsg	=	portalUserLogin
msg	=	msg
profile	=	profile
username	=	username
userId	=	userId

步骤12 配置“验证失败”赋值图元。


选择“验证失败”图元，在右侧单击图标，单击“新增行”，从全局上下文“系统变量”，拖拽“\$Flow.ResMsg”、“\$Flow.ResCode”到“赋值”下，并设置操作符为“=”，分别设置“值”为“"验证码错误! ""、“"1"”。

表 3-45 赋值

变量	操作符	值
\$Flow.ResMsg	=	"验证码错误! "
\$Flow.ResCode	=	"1"

图 3-298 配置“验证失败”赋值图元



步骤13 拖拽图元连线，并配置连线属性。


1. 在画布上，把鼠标放在起点图元图元上，从“+”拖动鼠标，在起点图元和“查询用户”图元间增加连线；即将当前脚本设置为服务编排的起始节点。
2. 依次在“查询用户”、“判断账号密码”、“判断是否包含验证码”、“执行登录”图元直接拖拽连线。

图 3-299 拖拽连线




- 单击“判断账号密码”与“判断是否包含验证码”图元之间的连线，再右侧属性单击，在“连线”中修改“连线类型”为“loginSuccess”。

图 3-300 选中连线

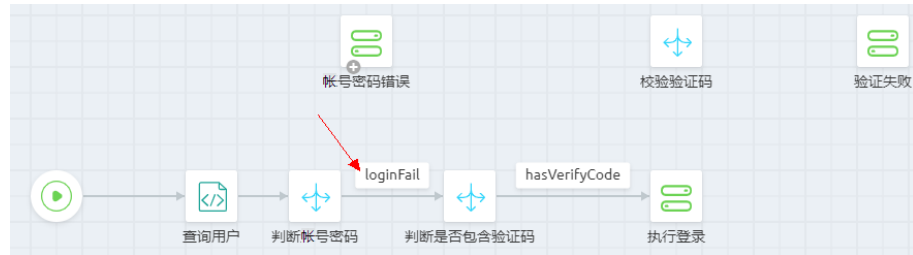



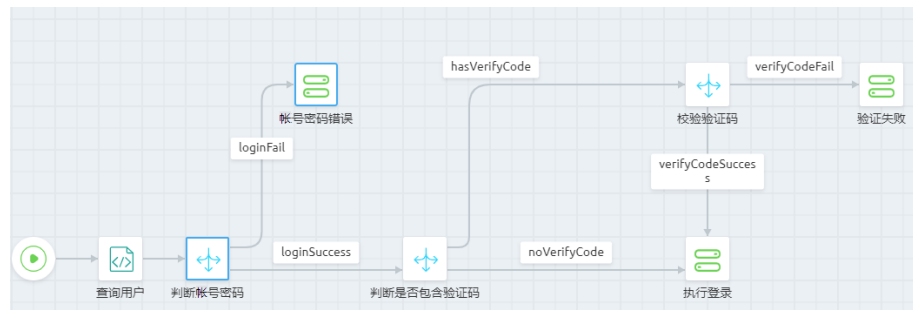
图 3-301 修改连线类型



- 单击“判断是否包含验证码”与“执行登录”图元之间的连线，再右侧属性单击，在“连线”中修改“连线类型”为“noVerifyCode”
- 从“判断账号密码”图元上拖拽一条连线到“账号密码错误”图元。
- 从“判断是否包含验证码”图元上拖拽一条连线到“校验验证码”图元。
- 从“校验验证码”图元上拖拽一条连线到“验证失败”图元。
- 从“校验验证码”图元上拖拽一条连线到“执行登录”图元，并设置该连线的“连线类型”为“verifyCodeSuccess”。

连线拖拽完成，如图3-302所示。

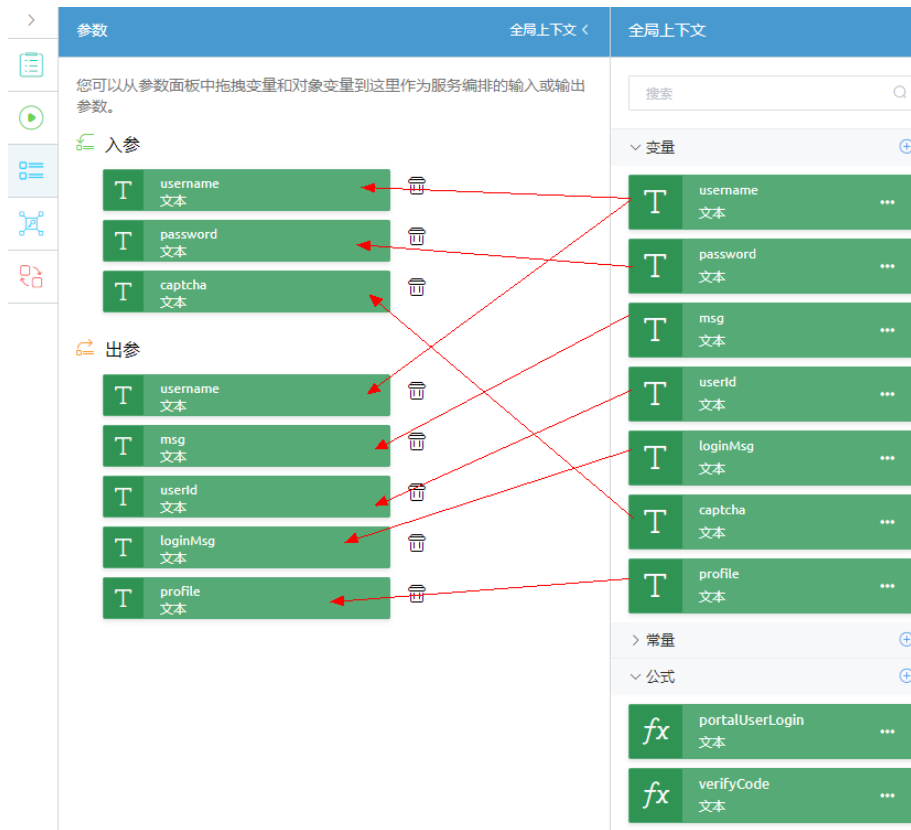
图 3-302 拖拽图元连线




步骤14 定义服务编排的输入、输出参数，并保存服务编排。


- 鼠标在画布空白处点一下，单击右侧，设置服务编排的输入输出参数，如图3-303所示。

图 3-303 拖拽服务编排的输入输出参数



2. 单击服务编排页面上方的 ，保存服务编排。系统会弹出窗口，显示编译结果。

步骤15 测试服务编排能否正常执行。


1. 单击服务编排编辑器上方的 ，执行服务编排。
2. 在“Flow Run”界面中输入测试数据，单击“运行”。其中，“test_cs”、“{XXXXXXXX}”为用户注册脚本中测试数据。

```
{  
  "username": "test_cs",  
  "password": "{XXXXXXXX}",  
  "captcha": ""  
}
```

执行成功，界面上会返回设备对象中的全部信息，样例如下：

```
{  
  "interviewId": "002N000000jjQ95dKbCK",  
  "outputs": {  
    "loginMsg": "null",  
    "msg": "登录成功!",  
    "profile": "cs",  
    "userId": "10gg0XXXXXXXXXXXXX",  
    "username": "test_cs"  
  }  
}
```

- 步骤16** (可选) 在服务编排编辑器单击“跟踪”，可以查看到上一步的执行日志，方便定位错误。

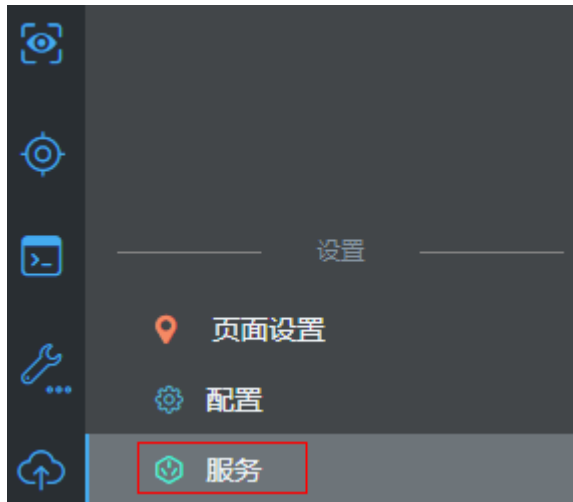
步骤17 测试成功，单击服务编排编辑器上方的，启用发布服务编排。

----结束

3.1.8.2.4 创建公共接口

步骤1 在经典版应用开发页面，单击左侧导航栏下方的“服务”，进入公共接口创建页面。

图 3-304 服务入口



步骤2 在公共接口中，单击“新建”。

图 3-305 公共接口创建

公共接口

使用公共接口，可以将flow的URL映射到发布到外部网关，第三方可以通过OAuth2.0调用。



步骤3 创建“用户登录”、“用户注册”脚本对应的公共接口，详细接口信息如表3-46所示。

说明

如果在“资源”下拉框中，未找到需要关联的脚本或服务编排，请检查相关脚本和服务编排是否已启用，**加粗斜体内容**以实际命名空间前缀为准。

表 3-46 公共接口

设置操作	版本	URL	方法	类型	资源
login	1.0.0	/login	POST	服务编排	<i>HW_login</i>

设置操作	版本	URL	方法	类型	资源
registerPortalUser	1.0.0	/registerPortalUser	POST	脚本	HW_registerPortalUser

----结束

3.1.8.3 组装页面

3.1.8.3.1 组装“业务用户注册”页面

“业务用户注册”页面实际上是一个对话框，是在管理员操作“新增用户”时，弹出的注册对话框，详细如图3-306所示。

图 3-306 业务用户注册对话框

用户名

密码

角色

结合页面需求，页面需要创建如下模型：

表 3-47 模型分析


模型名称	作用	详细定义
userInfo	保存注册用户信息的参数。	自定义模型，包含的计算节点如下： <ul style="list-style-type: none"> username：用户名，与输入框绑定。 password：密码，与输入框绑定。 role：角色，与下拉框绑定。

操作步骤

步骤1 在“我的应用”中，单击“设备维修管理系统”，进入应用。

步骤2 在“User”目录中，鼠标放在“Page”上，单击界面上出现的“+”，在弹出菜单中选择“标准页面”。

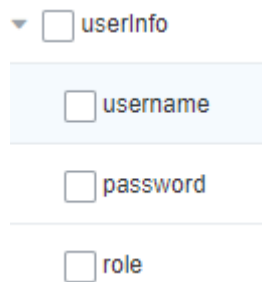
步骤3 在“标签”和“名称”文本框中输入“registerPortalUser”，单击“添加”。


当编辑已有标准页面时，为防止编辑时多人篡改，编辑前请单击进行锁定。

步骤4 定义模型“userInfo”。

1. 在页面底部单击“模型视图”，然后在“模型视图”中，单击“新增模型”。
2. 添加自定义模型，模型名称“userInfo”，单击“下一步”。
3. 单击“新增节点”，为模型添加节点“username”、“password”、“role”字段，字段类型都采用默认的Text。单击“下一步”，再单击“确定”。

图 3-307 新增节点



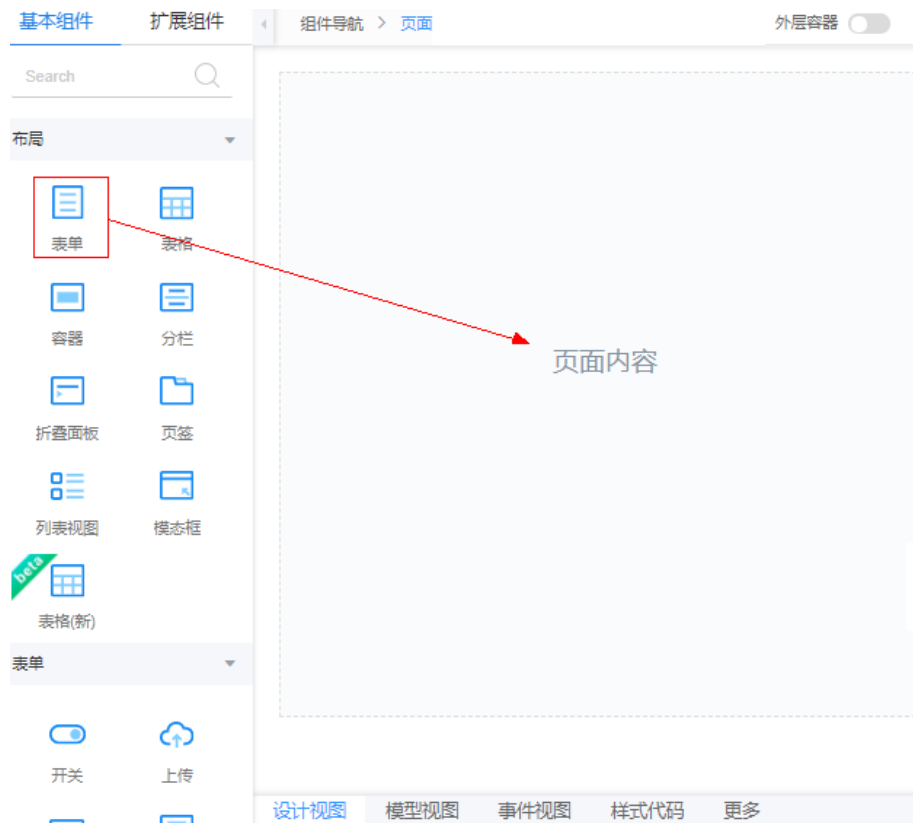
4. 单击页面上方的，保存模型。

步骤5 拖拽页面组件。

1. 单击“设计视图”，从“模型视图”切换到“设计视图”。
2. 从左侧基本组件列表区中，拖拽1个“表单”到“页面内容”中。

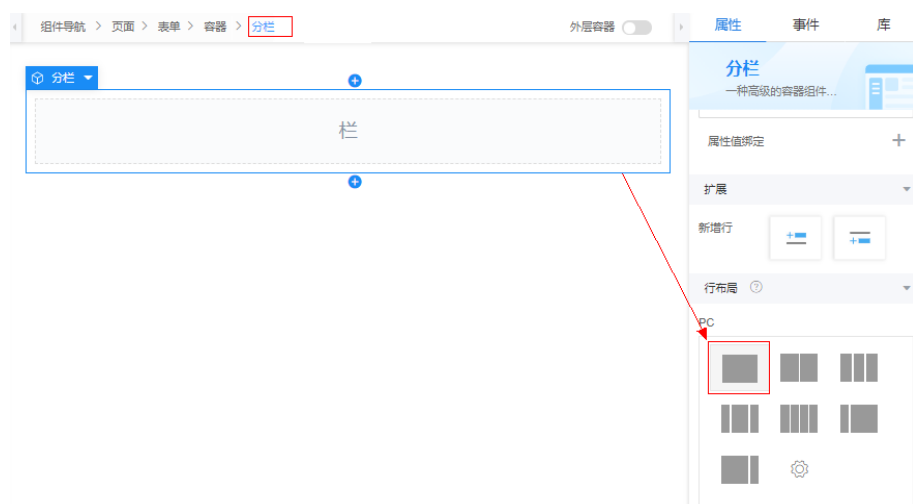
因为当前还没有定义数据源，在“元数据表单配置向导”弹窗底部，单击“取消”，创建一个空的表单控件。

图 3-308 拖拽一个空表单



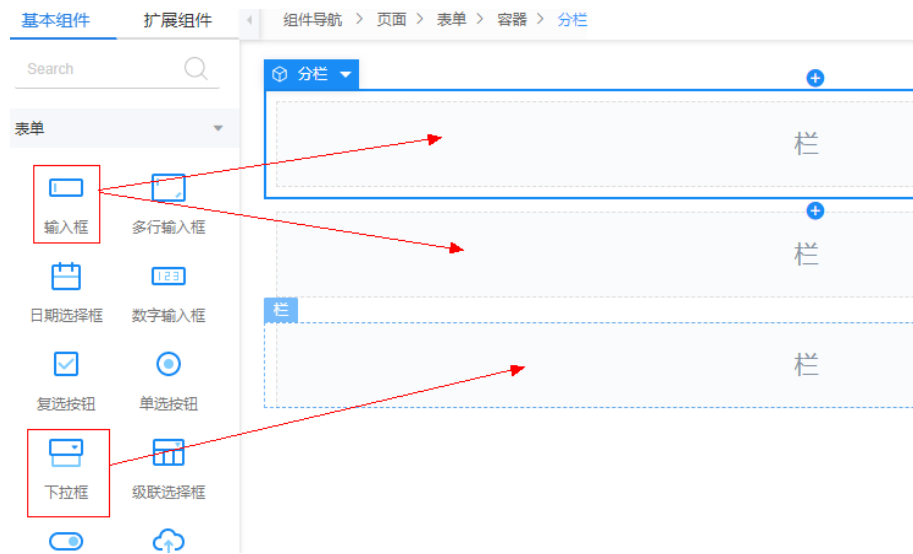
3. 拖拽1个“容器”到“表单”。
4. 拖拽1个“分栏”到“容器”。
5. 修改“分栏”为1栏（1行1列）。

图 3-309 设置分栏为 1 行 1 列



6. 选择“分栏”下的 + 2次，新增2个分栏。
7. 向前两个分栏，分别拖拽1个“输入框”组件，向第三个分栏中，拖拽一个“下拉框”组件。

图 3-310 拖拽输入框、下拉框



步骤6 配置组件属性，及数据绑定。

1. 设置第一个“输入框”属性。
 - a. 添加值绑定。


选中“输入框”组件，单击“数据绑定”下“值绑定”的⚙️，在弹窗中选择“userInfo”下的“username”字段。
 - b. 修改“标签”为“用户名”，“占位符”为“请输入用户名”。
2. 设置第二个“输入框”属性。
 - a. 添加值绑定。

选中“输入框”组件，单击“数据绑定”下“值绑定”的⚙️，在弹窗中选择“userInfo”下的“password”字段。
 - b. 修改“标签”为“密码”，“占位符”为“请输入密码”。
3. 设置“下拉框”属性。
 - a. 添加值绑定。

单击“数据绑定”下“值绑定”的⚙️，在弹窗中选择“userInfo”下的“role”字段。
 - b. 修改“标签”为“角色”，并设置“选项”为“cs, 客服人员”，“ds, 派单人员”、“ms, 维修人员”。

图 3-311 设置选项值



步骤7 单击页面上方的，保存设置。


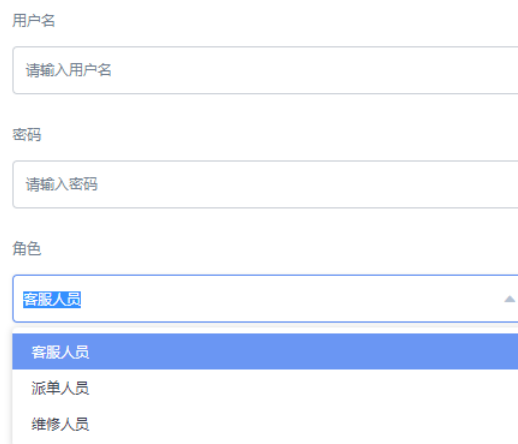
步骤8 单击页面上方的，进入预览页面，查看页面是否符合预期，并单击查看是否有下拉框选项。

图 3-312 页面预览



----结束

3.1.8.3.2 组装“业务用户管理”页面

“业务用户管理”页面主要是通过页面对象模型与“PortalUser”标准对象绑定，并在前端页面保存新增业务用户时，调用“用户注册”逻辑，完成对该对象进行增删查的操作，实现业务用户管理功能。

结合页面需求，页面需要创建如下模型：

表 3-48 模型分析

模型名称	作用	详细定义
portalUser	与后台标准对象PortalUser绑定。	对象模型，绑定对象以下字段。 <ul style="list-style-type: none"> • userName: 用户名 • profile: 权限 • userType: 用户类型 • createdBy: 创建者 • createDate: 创建日期
registerUser	与registerPortalUser公共接口绑定。	在页面获取到用户注册信息时，调用registerPortalUser公共接口，对PortalUser进行新增操作。

操作步骤

步骤1 在“我的应用”中，单击“设备维修管理系统”，进入应用。

步骤2 在“User”目录中，鼠标放在“Page”上，单击界面上出现的“+”，在弹出菜单中选择“标准页面”。

步骤3 在“标签”和“名称”文本框中输入“portalUserList”，单击“添加”。

当编辑已有标准页面时，为防止编辑时多人篡改，编辑前请单击进行锁定。

步骤4 定义对象模型“portalUser”。

1. 在页面底部单击“模型视图”，然后在“模型视图”中，单击“新增模型”。
2. 添加对象模型，模型名称“portalUser”，单击“下一步”。
3. 在“选择对象”下，选择标准对象“PortalUser”，并勾选以下字段，然后单击“下一步”，再单击“确定”。

表 3-49 选择字段

节点名称	数据类型
userName	保持默认
profile	保持默认
userType	保持默认
createdBy	保持默认
createDate	保持默认

图 3-313 选择对象

编辑

1 基本信息 2 设置 3 方法


* 选择对象

PortalUser

* 选择字段

<input type="checkbox"/>	字段名称	字段标签
<input type="checkbox"/>	id	记录 ID
<input checked="" type="checkbox"/>	usrName	User Name
<input type="checkbox"/>	userPassword	User Password
<input type="checkbox"/>	passwordSalt	passwordSalt
<input checked="" type="checkbox"/>	profile	Profile
<input type="checkbox"/>	languageLocaleKey	Language Locale Key
<input type="checkbox"/>	userId	Userid

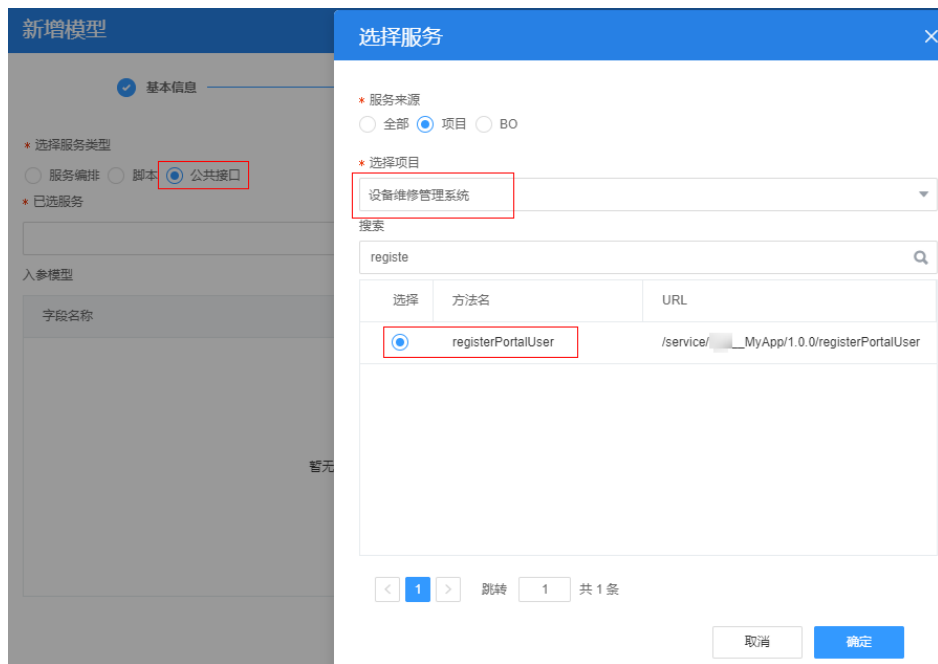
取消 上一步 下一步

4. 单击页面上方的，保存模型。


步骤5 定义与API (registerPortalUser:1.0.0) 关联的服务模型。

1. 在“模型视图”下，单击“新增模型”。
2. 设置模型名称为“registerUser”，来源选择“服务”，单击“下一步”。
3. “选择服务类型”为“公共接口”，“选择项目”为“设备维修管理系统”，指定模型与API“registerPortalUser”关联，单击“确定”，再单击“下一步”。

图 3-314 选择服务类型为 API

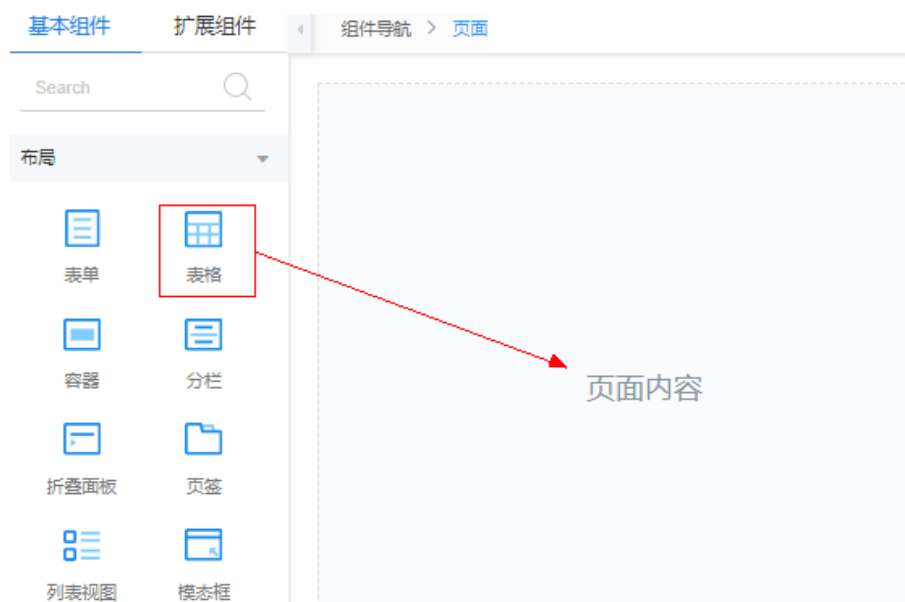


关联API后，系统会自动显示API中脚本的输入、输出参数。

4. 方法保持不变，单击“确定”。
5. 单击页面上方的, 保存设置。

步骤6 切换到“设计视图”，从左侧基础组件区，拖拽一个“表格”到右侧“页面内容”中。

图 3-315 拖拽一个表格到页面



步骤7 为表格绑定模型，并设置表格查询结果区域。


1. 在“设计视图”中，选中“表格”，单击右侧“属性”页签, 如图3-316所示。

图 3-316 为表格绑定对象模型

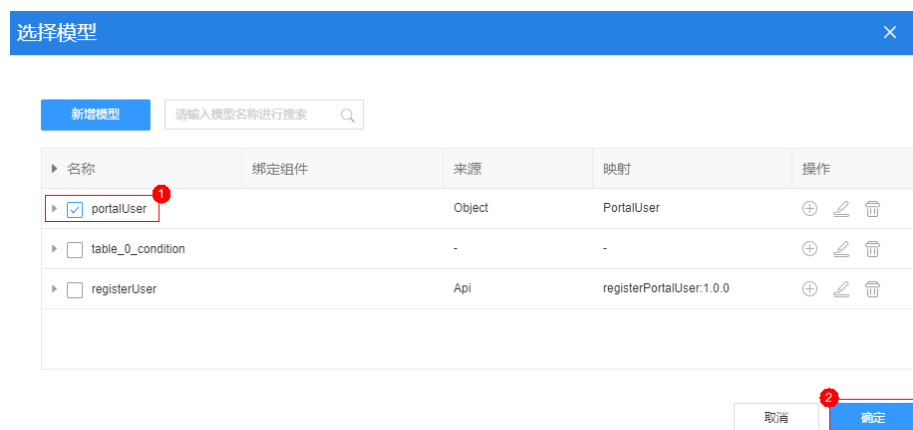


2. 在“选择模型”对话框中，勾选“portalUser”对象模型，再单击“确定”。

须知

勾选“portalUser”后，可以将该节点下全部字段自动添加为表格的列。

图 3-317 绑定模型



绑定对象模型后，系统自动将模型的所有字段添加为表格列，如图3-318所示。

图 3-318 绑定数据模型后的表格列

#	User Name	Profile	UserType	创建人	创建时间
暂无数据					


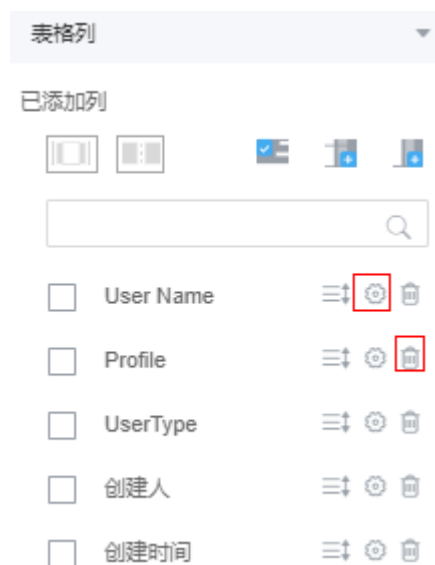
- 选中“表格”，在右侧“已添加列”中，单击“Profile”后的, 删除“Profile”字段。

图 3-319 删除字段



- 单击“User Name”后的, 修改“列标题”为“用户名”。

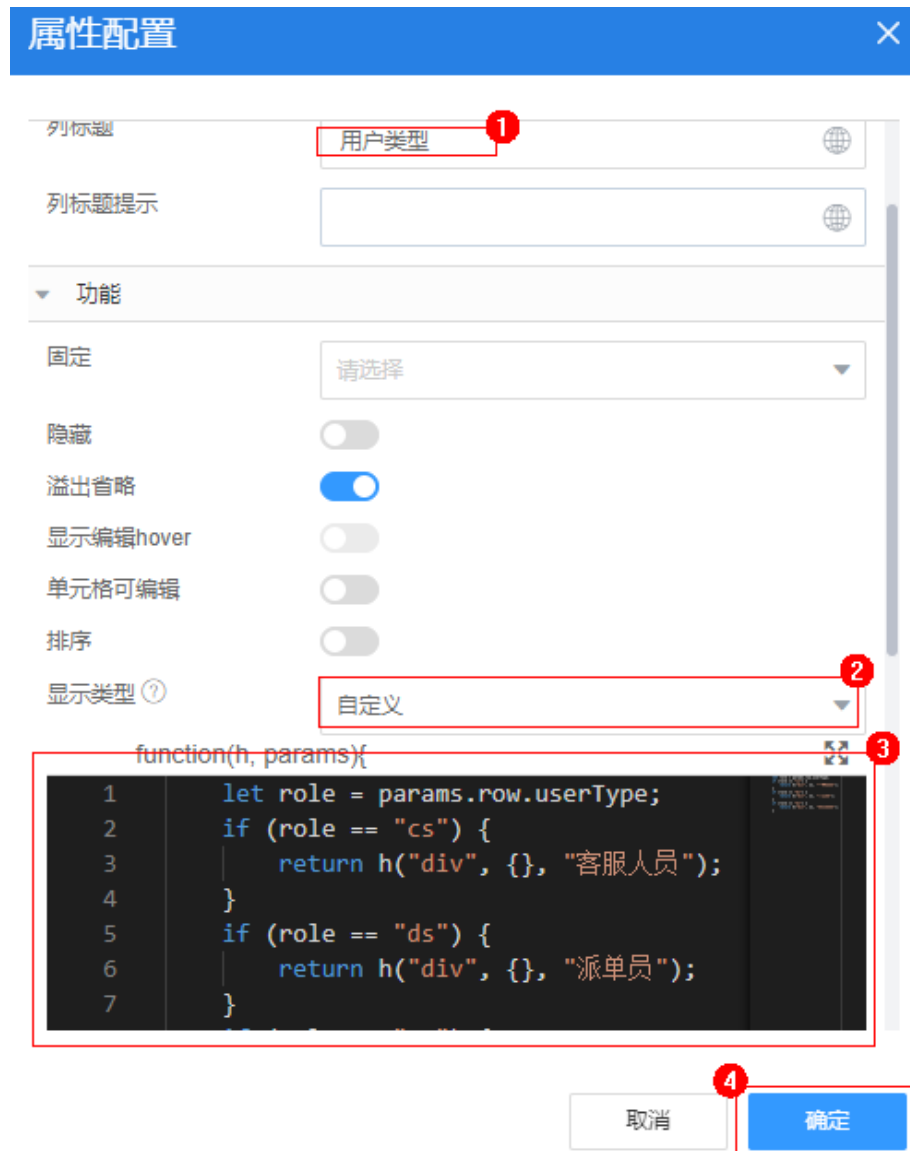
图 3-320 设置 “User Name” 列标题




5. 单击 “UserType” 后的 ⚙️，修改 “列标题” 为 “用户类型”，“显示类型” 设置为 “自定义”，并在输入以下自定义事件代码，然后单击 “确定”。

```
let role = params.row.userType;
if (role == "cs") {
  return h("div", {}, "客服人员");
}
if (role == "ds") {
  return h("div", {}, "派单员");
}
if (role == "ms") {
  return h("div", {}, "维修人员");
}
```

图 3-321 修改“UserType”列属性



6. 单击页面上方的, 保存设置。

步骤8 设置工具栏区域。

1. 在左侧“设计视图”中选中“表格”，单击右侧“属性”页签“表格区块”中“工具栏”后的“添加”按钮。

图 3-322 增加工具栏



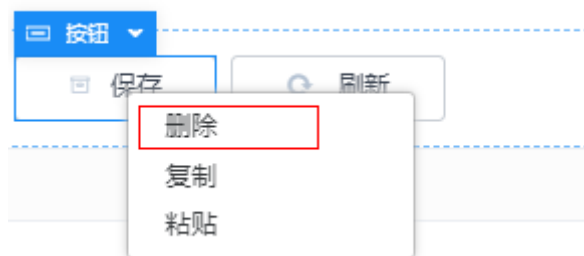
2. 只保留“新增行”、“删除”按钮，删除其他工具栏中多余的按钮。


单击选中待删除按钮，鼠标右键单击，**请确认已选中按钮**，然后选择“删除”，如果误删其他组件，在页面上单击返回按钮即可。

图 3-323 工具栏中需要删除的按钮



图 3-324 选中待删除按钮



- 选中“新增行”按钮，在右侧属性面板中，修改“显示名称”为“新增用户”。
- 单击页面上方的, 保存设置。

步骤9 创建查询条件区域。

- 选中“表格”，单击右侧“属性”页签“表格区块”中“查询”后的“添加”按钮。

图 3-325 增加查询



- 在弹窗中勾选“usrName”、“userType”，单击“确定”，表格上方自动生成查询条件区域。

图 3-326 添加查询条件




3. 拖拽一个“下拉框”组件到“用户类型”输入框下方，然后选择“用户类型”输入框，右键单击删除。
4. 配置“下拉框”组件属性。
 - a. 选择“下拉框”组件，在右侧属性面板中，单击“数据绑定”，在数据绑定弹窗中，勾选“table_0_condition”的“queryInfo”下的“userType”。

图 3-327 添加数据绑定



- b. 修改“下拉框”的“标签”为“用户类型”，并设置“选项”为“cs, 客服人员”，“ds, 派单人员”、“ms, 维修人员”。

图 3-328 设置选项值



步骤10 添加“新增用户”按钮的“点击”事件代码。


单击“事件”页签，再单击“新增行”的 ，在“自定义动作”中，删除原有事件代码，输入以下事件代码，然后单击“保存”。

图 3-329 编辑事件代码





```
let _table = context.$component.table;
context.$dialog.popup(
  {
    title: '新增用户', //弹出框标题
    page: 'HW_registerPortalUser', // 弹出页面名称
    footerHide:false,
    width:40, //宽, 小于100为百分比, 大于100为px
    height:450, // 高px
    okText: '确定',
    cancelText: '取消',
    onCancel: () => {
      $model.ref("userInfo").setData({});
    }, // 取消时回调
    onOk: () => {
```

```

// 确定时回调
//debugger;
let userInfo = $model.ref("userInfo").getData();
$model.ref('registerUser').setData({inputParam: userInfo});
$model.ref('registerUser').run().then(function (data) {
  if (data && data.msg){
    context.$message.success(data.msg);
    _table.doQuery();
  }
}).catch(function (error) {
  console.log('error is', error);
  context.$message.error(error.resMsg);
});
}
)

```

步骤11 单击页面上方的, 保存设置。

步骤12 单击页面上方的, 进入预览页面, 查看页面是否符合预期。

1. 查看页面样式是否符合预期, 即查看“用户类型”下拉框中, 选项值是否正常。
2. 查看用户列表中是否有测试数据。

图 3-330 业务用户管理



图 3-330 展示了“业务用户管理”的界面。顶部包含“用户名”输入框和“用户类型”下拉选择框。下拉框已打开，显示了“客服人员”、“派单人员”和“维修人员”三个选项。下方有两个操作按钮：“新增用户”和“删除”。再下方是一个表格，列出了用户信息。表格下方有分页控件，显示当前为第 1 页，共 10 条记录。

	#	用户名	用户类型	创建人	创建时间
<input type="checkbox"/>	1	test_cs	客服人员		2020-10-15 15:15:31

3. 单击“新增用户”, 查看是否弹出“业务用户注册”页面, 如果正常显示, 则说明按钮事件代码正确, 如果不显示弹窗, 请检查按钮事件代码。
4. 在“业务用户注册”弹窗中, 输入“用户名”、“密码”, 并设置“角色”, 单击“确定”, 是否提示“注册成功”, 注册成功后, 返回“业务用户管理”, 查看业务用户列表中, 是否显示新增用户信息。如果新增用户失败, 请检查“业务用户管理”页面相关事件代码。

----结束

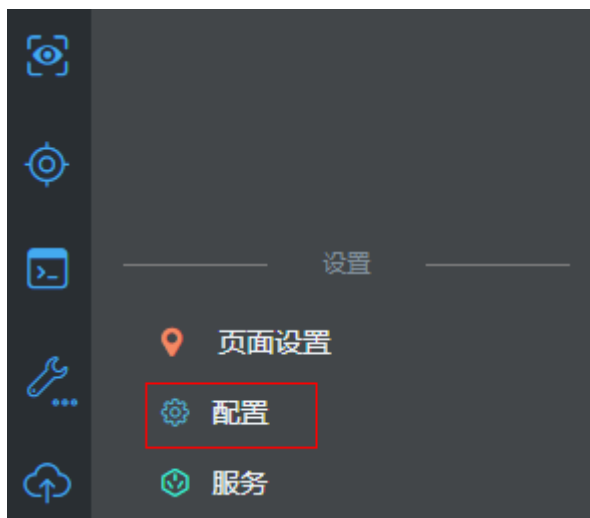
3.1.8.3.3 验证

验证用户管理功能之前, 我们需要先将“业务用户管理”页面挂载到导航条上, 应用菜单仅支持开发者账号的管理员查看, 业务账号只能查看对应权限的页面。

操作步骤

步骤1 在经典版应用开发页面, 单击左侧导航栏下方的“配置”。

图 3-331 应用配置入口



步骤2 在“导航条”页签单击“菜单树”右侧的“+”，选择“添加页签”。

图 3-332 添加页签



步骤3 定义“新增用户”页签。在“添加页签”弹窗中，设置以下信息，然后单击“保存”。

- 页面类型：设置为“标准页面页签”。
- 标签：设置为“新增用户”。
- 名称：设置为“addUser”。
- 页面：设置为“*HW_portalUserList*”。

图 3-333 添加“新增用户”页面

编辑页签 ✕

* 页签类型

显示区域

主页菜单

自定义菜单栏

打开方式

* 标签

名称

图标

* 页面

描述


步骤4 在应用左侧导航下，单击，进入应用预览页面。

图 3-334 查看应用



步骤5 在应用菜单上，单击“新增用户”，进入用户管理页面。

图 3-335 新增用户



步骤6 在用户管理页面，单击“新增用户”，在弹窗中设置用户名、密码及角色。

图 3-336 添加新用户

The image shows a '新增用户' (Add New User) dialog box. It has a title bar with the text '新增用户' and a close button (X). Below the title bar, there are three input fields: '用户名' (Username) with the value 'test_ds', '密码' (Password) with the value 'test_ds', and '角色' (Role) with a dropdown menu showing '派单人员'. At the bottom of the dialog, there are two buttons: '取消' (Cancel) and '确定' (Confirm).

步骤7 创建完成后，在用户列表中，查看是否已存在刚刚添加的用户，

如果有，则说明页面及事件代码设置正确。如果页面有报错，请根据页面报错定位问题。

----结束

3.1.8.4 创建业务凭证

业务权限凭证是业务用户在登录AstroZero开发的应用后，所拥有的访问后端业务接口的凭证，后端业务接口以CustomAPI的形式进行开放时，可指定归属业务权限凭证。根据本示例的业务场景，创建4个业务凭证，即“cs”客服人员、“ds”派单人员、“ms”维修人员、“Login”业务用户登录专用。

操作步骤

步骤1 使用华为账号，登录AstroZero经典版开发环境。

步骤2 在AstroZero经典版开发环境首页，单击“管理”，进入AstroZero经典版开发环境管理中心。

步骤3 选择“用户管理 > 业务权限凭证”，在右侧“业务权限凭证”中，单击“新建”。

步骤4 在弹窗中，设置“标签”、“名称”为“cs”，单击“保存&新建”。

图 3-337 创建业务凭证。

* 标签
CS

* 名称
CS

目录
请选择或输入新目录

保存 保存&新建 取消

步骤5 分别创建业务凭证“ds”、“ms”、“Login”。

----结束

3.1.8.5 创建权限配置

业务用户登录应用后的访问权限，是通过AstroZero的Profile进行控制管理的。

背景信息

本示例中创建了3种业务用户的业务权限，主要是在AstroZero预置的Portal User Profile权限基础上，进行自定义业务用户权限配置和拓展实现的。在AstroZero的权限配置功能中，基于某个权限配置的新创建的Profile，将会继承原Profile的全部权限。

在后续有新的业务用户注册时，只需要将业务用户添加到对应的权限配置中，即可获得该权限配置中的权限。

权限配置创建的大致流程：

1. 基于Portal User Profile创建客服人员csProfile。
2. 配置客服人员csProfile workflow权限、自定义对象权限以及业务凭证权限。
3. 在客服人员csProfile的基础上，创建派单人员dsProfile、维修人员msProfile，并修改对应的业务凭证权限。
4. 配置匿名用户登录权限配置的业务凭证。

操作步骤

步骤1 使用华为账号，登录AstroZero经典版开发环境。

步骤2 在AstroZero经典版开发环境首页，单击“管理”，进入AstroZero经典版开发环境管理中心。

步骤3 修改内置系统参数“bingo.permission.resource.default.switch”默认值。

1. 在左侧导航栏中，选择“系统配置 > 系统参数 > 内置系统参数”。
2. 在搜索框中，搜索“bingo.permission.resource.default.switch”，然后单击参数名。

图 3-338 修改内置系统参数默认值开发接口权限




3. 单击“值”右侧的, 进入编辑页面, 设置“值”为“是”, 单击“保存”, 返回参数详情页面。

图 3-339 编辑“值”



图 3-340 将值修改为“是”



修改当前参数是为了开启“Portal User Profile”权限配置中接口权限。

步骤4 创建“客服人员”的权限配置“csProfile”。

1. 选择“用户管理 > 权限配置”，在右侧“权限配置列表”中，单击“新建”。

图 3-341 新建权限配置



2. 在新建权限配置弹窗中，选择“现有权限配置”为“Portal User Profile”，设置要新增的“权限配置名称”为“csProfile”，单击“保存”，返回权限配置列表。

图 3-342 基于 Portal User Profile 新增权限配置



说明

- 普通克隆：权限配置创建后，不仅继承了原模板权限配置的权限，且后续可以根据实际业务在权限配置详情页进行修改。
 - 继承克隆：除基本信息和业务权限凭证，其余权限屏蔽编辑按钮，不能修改。
3. 在权限配置列表中，单击“csProfile”，进入权限配置详情。
 4. 配置“应用程序设置”下的权限。
 - a. 在搜索框中搜索“设备维修管理系统”，设置当前应用“可见性”及“默认”。

图 3-343 设置应用可见性及默认






- b. 单击“设备维修管理系统”，进入菜单可见性设置。
- c. 单击右上角，设置客服人员的默认菜单“工单列表（客服人员）”、及其他可见菜单，例如“Home”、“工单管理”，然后单击，返回应用列表，再单击，保存设置。

图 3-344 设置默认菜单为“工单列表（客服人员）”




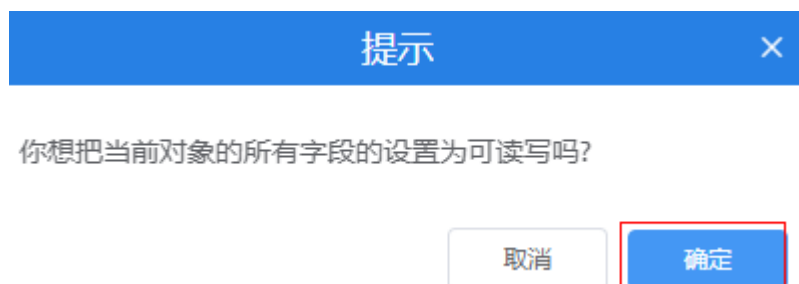
5. 配置“自定义对象权限”下的对象权限。
 - a. 在“自定义对象权限”页签，单击“自定义对象权限”右侧的 ，并在搜索框中输入“HW_WorkOrder_CST”，然后勾选“HW_WorkOrder_CST”对象的“编辑”权限，并在提示弹窗中单击“确定”。“HW_”请以实际命名空间前缀为准。

图 3-345 勾选自定义对象“编辑”权限



图 3-346 设置自定义对象所有字段可读写



- b. 勾选“HW_WorkOrder_CST”对象的“修改全部”、“创建”权限，再单击✓，保存设置。

图 3-347 勾选“修改全部”



6. 单击“业务权限凭证”，在“业务权限凭证”下，勾选“cs”业务凭证，再单击✓，保存设置。

图 3-348 配置业务凭证

权限配置详情: [csProfile](#)

此权限配置的用户具有下列权限和页面布局。

管理员通过编辑用户的个人信息即可更改其权限配置。



步骤5 创建“派单人员”的权限配置“dsProfile”。

1. 选择“用户管理 > 权限配置”，在右侧“权限配置列表”中，单击“新建”。
2. 在新建权限配置弹窗中，选择“现有权限配置”为“csProfile”，设置要新增的“权限配置名称”为“dsProfile”，单击“保存”，返回权限配置列表。

图 3-349 基于 csProfile 新增权限配置

新建权限配置

现有权限配置 csProfile

用户许可证 Baa5

* 权限配置名称 dsProfile

描述 请输入

取消 保存

3. 在权限配置列表中，单击“dsProfile”，进入权限配置详情。单击“应用程序设置”，单击“设备维修管理系统”，在弹窗中设置派单人员的默认菜单，及可见菜单如图3-350所示。

图 3-350 设置派单人员默认菜单



4. 单击“业务权限凭证”，在“业务权限凭证”下，取消勾选“cs”，勾选“ds”业务凭证，再单击✓，保存设置。

图 3-351 修改业务权限凭证

权限配置详情: dsProfile

此权限配置的用户具有下列权限和页面布局。

管理员通过编辑用户的个人信息即可更改其权限配置。



步骤6 创建“维修人员”的权限配置“msProfile”。

1. 选择“用户管理 > 权限配置”，在右侧“权限配置列表”中，单击“新建”。
2. 在新建权限配置弹窗中，选择“现有权限配置”为“csProfile”，设置要新增的“权限配置名称”为“msProfile”，单击“保存”，返回权限配置列表。
3. 在权限配置列表中，单击“msProfile”，进入权限配置详情。在“应用程序设置”页签，单击“设备维修管理系统”，在弹窗中设置维修人员的默认菜单，及可见菜单如图3-352所示。

图 3-352 设置维修人员默认菜单



4. 单击“业务权限凭证”页签，在“业务权限凭证”下，取消勾选“cs”，勾选“ms”业务凭证，再单击✓，保存设置。

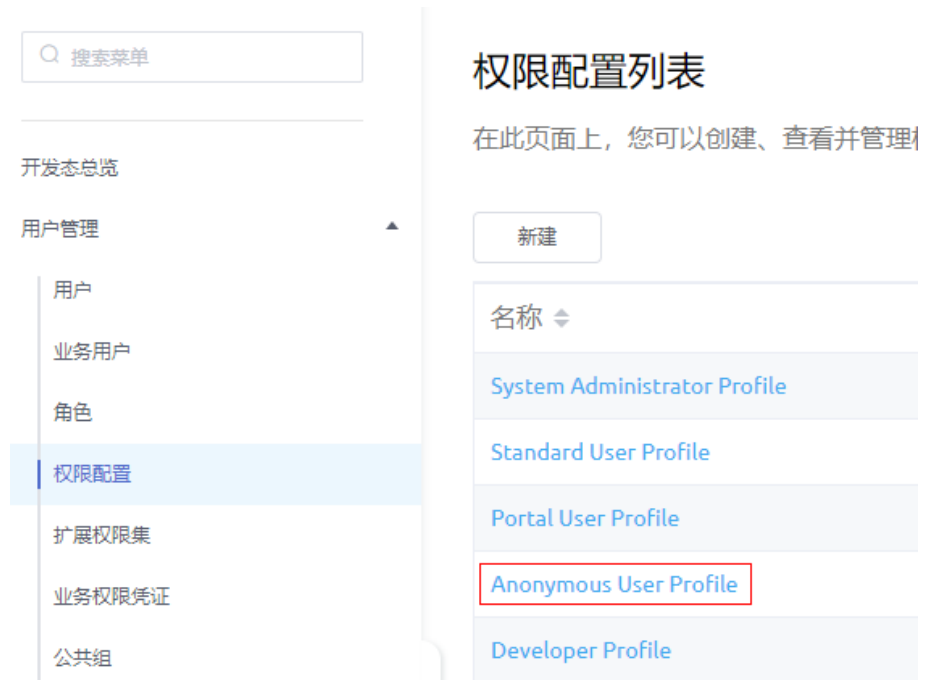
图 3-353 修改业务权限凭证



步骤7 配置“Anonymous User Profile”的业务凭证。

1. 选择“用户管理 > 权限配置”，在右侧“权限配置列表”中。
2. 在权限配置列表中，单击“Anonymous User Profile”，进入权限配置详情。

图 3-354 单击 Anonymous User Profile



3. 单击“业务权限凭证”，在“业务权限凭证”下，勾选“Login”业务凭证，再单击✓，保存设置。

图 3-355 添加业务权限凭证



----结束

3.1.8.6 添加接口级业务权限凭证

业务用户的权限配置创建完成后，还需要为应用中创建的公共接口，添加接口级权限配置，本示例中业务用户需要使用的公共接口，主要是在“工单管理”及“用户管理”2个功能模块创建的。

操作步骤

- 步骤1 在“我的应用”中，单击“设备维修管理系统”，进入应用。
- 步骤2 在应用中，单击下方“配置”，进入应用配置。

图 3-356 应用配置入口



- 步骤3 在应用配置页面，单击“业务权限凭证”页签，进入业务权限凭证。

图 3-357 业务权限凭证



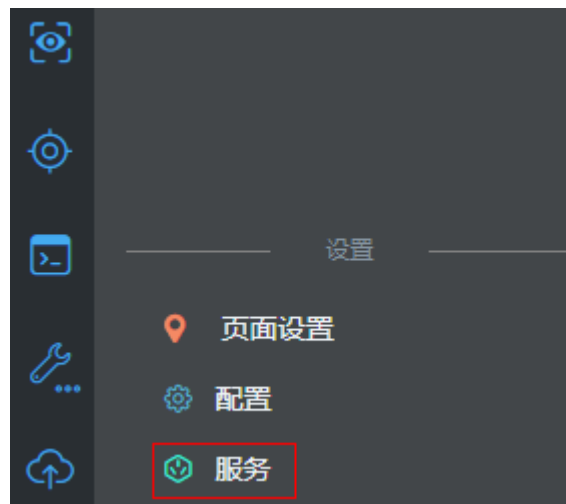
步骤4 单击“导入”，在弹窗中勾选“cs”、“ds”、“ms”、“Login”，然后单击“导入”。

图 3-358 导入业务权限凭证



步骤5 在页面下方，单击“服务”，进入公共接口页面。

图 3-359 创建公共接口入口



步骤6 为“createWorkOrder”公共接口，添加客服人员业务凭证“cs”。

1. 在接口列表中，单击“createWorkOrder”接口，进入接口详情。
2. 在接口详情页面，单击“业务权限凭证”下的“编辑”。

图 3-360 编辑业务权限凭证




3. 在编辑业务权限凭证弹窗中，勾选“cs”，再单击 ，然后单击“保存”。

图 3-361 选择客服人员业务凭证 cs



步骤7 参考**步骤6**，给其他公共接口添加业务权限凭证，详细接口名及业务凭证如**表3-50**所示。

表 3-50 需要添加业务凭证的公共接口

编号	公共接口名	需要添加的业务凭证	涉及业务用户
1	createWorkOrder (上一步已添加)	cs	客服人员
2	equipmentSelectListQuery	cs	客服人员
3	judgeNextStatus	ms	维修人员
4	modifyOrderStatus	ms	维修人员
5	queryWorkOrder	cs、ds、ms	客服人员、派单人员、维修人员

编号	公共接口名	需要添加的业务凭证	涉及业务用户
6	queryWorker	ds	派单人员
7	login	Login	客服人员、派单人员、维修人员

----结束

3.1.9 应用业务测试

3.1.9.1 本节导读

应用开发完成后，需要根据业务流程，通过使用不同角色账号登录，进行应用业务测试。

业务测试主要分为**管理业务用户**、**添加设备及设备管理**和**处理工单**三部分。其中，前两部分是管理员权限进行的操作（这里管理员账号，均使用应用开发者账号进行）。处理工单部分，需要根据工单流程顺序，切换客服人员、派单人员、维修人员以及管理员号进行测试。

- **管理员用户**：当前登录AstroZero开发应用使用的账号，用于新增业务用户、添加用户权限、添加设备信息、管理工单，监控设备。
- **业务用户**：使用“设备维修管理系统”应用的用户，分别是客服人员、派单员及维修人员。

表 3-51 业务测试详情

角色	测试页面	涉及功能
管理员用户	新增用户页面、平台管理用户页面、设备管理页面、工单管理页面、设备监控页面	新增业务用户，配置权限集、添加设备信息、管理工单，监控设备
业务用户	客服人员	App登录页面，工单列表（客服人员）页面
	派单员	App登录页面，工单列表（派单员）页面
	维修人员	App登录页面，工单列表（维修人员）页面
		创建工单
		接单、拒单、关单
		派单、查看工单状态

3.1.9.2 管理业务用户

管理业务用户模块主要包括新增业务用户、配置业务用户权限集两大功能。

- **新增业务用户**
管理员用户登录应用预览页面后，在“新增用户”页面，新增业务用户。新增后，管理员用户可以在用户列表中，查看或删除用户。

- 配置业务用户权限集
业务用户添加完成后，按照各自的角色功能，对每个业务用户添加业务凭证、设置权限配置，并在对应的工作队列中，添加业务用户。
配置权限集后，业务用户将获得BPM、服务编排、自定义对象的操作和查看权限。

操作步骤

步骤1 使用应用开发者账号，访问并登录AstroZero经典版开发环境。


步骤2 在“项目 > 我的应用”中，单击“设备维修管理系统”应用的 ，进入应用预览页面。

图 3-362 查看应用



步骤3 新增业务用户。

1. 在应用菜单中，选择“新增用户”，进入用户管理页面。
2. 单击“新增用户”，在弹窗中输入用户名及密码，并设置当前新增业务用户的“角色”，单击“确定”。

图 3-363 新增用户



新增用户

用户名

testcs

密码

.....

角色

客服人员

取消 确定

表 3-52 角色对应权限集

用户名	角色	对应权限集
testcs	客服人员	csProfile
testds	派单员	dsProfile
testms	维修人员	msProfile

步骤4 完成新增后，返回用户管理页面，查看是否已存在刚刚提交的用户信息。

图 3-364 查看用户



Home X 新增用户 X EN 用户头像 hw

用户名 用户类型 搜索

+ 新增用户 删除

<input type="checkbox"/>	#	用户名	用户类型	创建时间	创建人
<input type="checkbox"/>	1	testcs	客服人员	2020-09-11 09:46:44	hw

步骤5 增加业务用户的权限集。

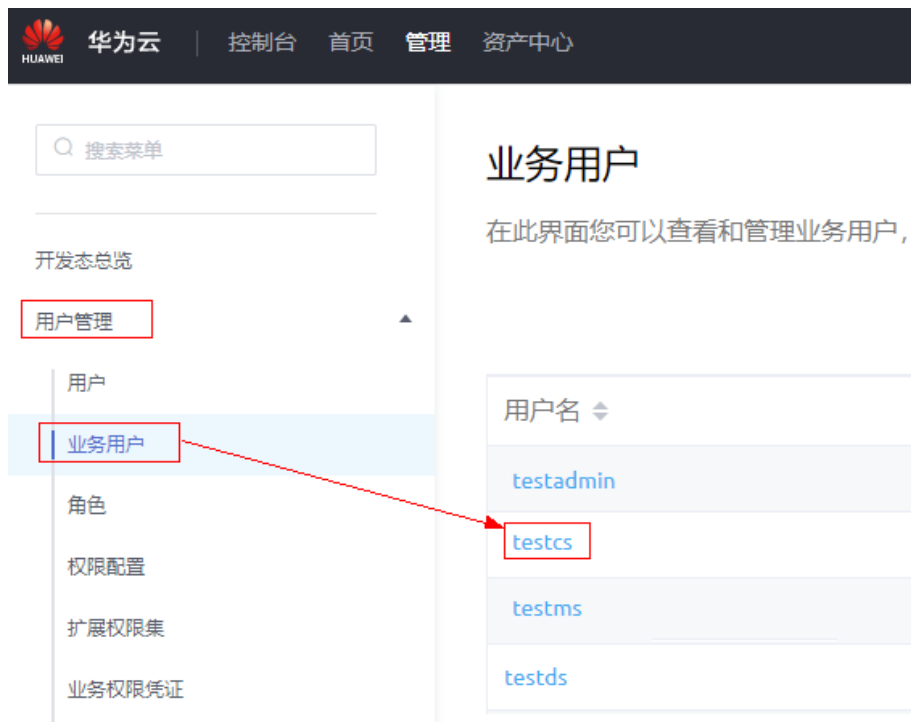
1. 在浏览器中打开新的页签，使用当前应用开发者账号访问并登录AstroZero经典版开发环境。
2. 在经典版开发环境首页，单击“管理”，进入AstroZero经典版开发环境管理中心。

图 3-365 管理页签



3. 在左侧导航栏中，选择“用户管理 > 业务用户”，进入业务用户列表。
4. 在用户列表中，单击一个需要添加权限集的用户名，进入该用户信息详情。

图 3-366 选择业务用户名



5. 单击“覆盖业务用户权限”后的, 勾选后，单击“保存”。

业务用户详情: testcs

☰ 基本信息

名称	用户名	时区	覆盖业务用户权限
testcs	testcs	<input type="text" value="请选择"/>	<input checked="" type="checkbox"/>

6. 单击“权限集”下的“编辑”，在弹窗中设置对应权限集，然后单击“保存”，返回用户详情查看已添加的权限集。
不同角色对应的权限集，如表3-53所示。

表 3-53 角色对应权限集

角色	对应权限集
客服人员	csProfile
派单员	dsProfile
维修人员	msProfile

图 3-367 业务用户详情

< 业务用户列表

业务用户详情

☰ 基本信息

用户名	时区
testcs	Local

☰ 权限集

图 3-368 编辑业务用户权限列表



步骤6 在工作队列中，添加业务用户。

1. 在经典版开发环境首页，单击“管理”，进入经典版管理中心。
2. 在左侧导航栏中，选择“用户 > 工作队列”，单击客服人员队列“CustomerService”，进入队列详情。
3. 在队列详情中，单击“成员信息”下的“添加”，在弹窗中设置“成员类型”为“业务用户”，在“成员列表”下拉框中选中需要添加为客服人员的业务用户账号，单击“添加”。

图 3-369 在队列中添加业务用户



----结束

3.1.9.3 管理设备信息

应用开发完成后，设备信息中仅存在一些在创建过程中的测试数据，需要管理员对设备信息进行新增、编辑、删除等操作。

操作步骤


- 步骤1** 使用应用开发者账号，访问并登录AstroZero经典版开发环境。
- 步骤2** 在“项目 > 我的应用”中，单击“设备维修管理系统”应用的 ，进入应用预览页面。
- 步骤3** 在应用菜单中，选择“设备管理”，在设备管理页面中，单击“新增设备”，在“设备详情”页，设置需要新增的设备信息，然后单击“保存”。

图 3-370 设备管理

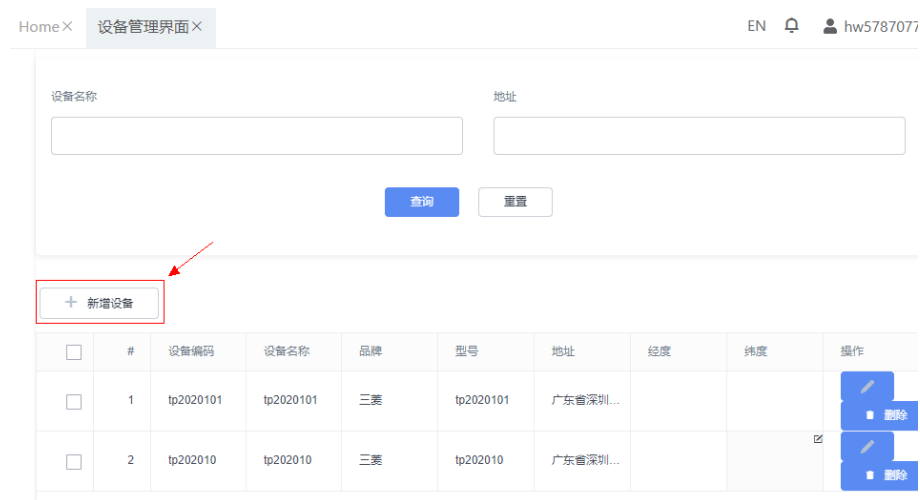


图 3-371 设置设备详细



- 步骤4** 设备保存后，返回到设备管理页面，查看设备列表中是否已存在上一步添加的设备。
- 步骤5** 选择一条设备记录，单击“操作”中的“编辑”及“删除”按钮，验证当前信息是否可以编辑及删除。

----结束

3.1.9.4 处理工单

处理工单需要使用不同的业务用户账号，通过登录对应功能页面，对工单流转进行处理。工单流转的流程顺序如下：

1. 客服人员创单。
2. 派单员派单。
3. 维修人员接单。
4. 管理员监控设备。

进入应用登录页面

使用业务人员账号验证时，需要登录应用的登录页面。在开发环境中，您可以按照以下方式，进入应用登录页面。

步骤1 使用应用开发者账号，访问并登录AstroZero经典版开发环境。

步骤2 在“项目 > 我的应用”中，单击“设备运维管理”，进入应用。


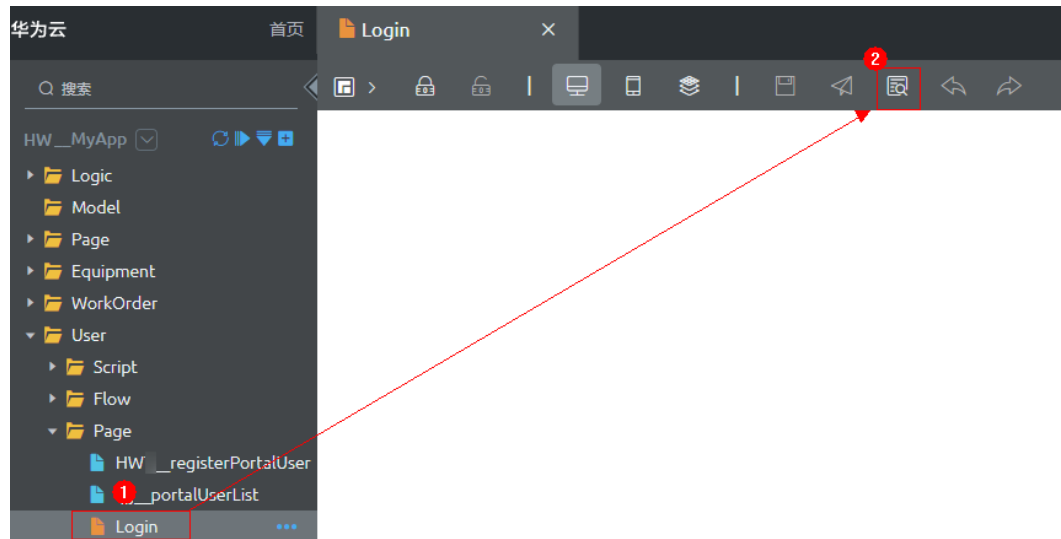
步骤3 在应用菜单中的“User > Page”下，单击“Login”页面，再单击预览图标。

图 3-372 预览登录页



预览之后进入即可登录页，您可以在登录页使用对应的业务用户（例如testcs）登录。

图 3-373 进入 Login 页面的预览



📖 说明

- 在开发环境中，业务用户需要在开发者账号登录的状态下，才能正常登录“设备运维管理”应用。
- 在开发环境中，使用业务用户账号在图3-373所示Login预览页，登录应用时，业务用户与当前预览页面的开发者账号共用同一个token，因此业务用户登录应用后，会导致开发者账号下线。当您要切换其他业务用户或返回开发环境时，需要重新使用开发者账号登录AstroZero开发环境。
- 如果在开发者账号未登录或已下线的状态下，直接使用业务用户账号登录应用，则属于匿名登录，如果出现以下错误属于正常现象。

```
{"responseCode":401,"responseMessage":"Not allowed."}
```
- 当应用发布到运行环境后，业务用户则可以直接通过Login预览页URL登录应用，详细操作请参考[在运行环境安装应用](#)，发布应用到运行环境。

----结束

验证处理工单

步骤1 验证创建工单。

- 访问图3-373中的应用登录页，使用客服人员账号登录（testcs）。
- 登录后，进入“工单列表（客服人员）”页面，单击“创建工单”，进入创建工单页面。
- 单击“选择工单方案”下拉框，设置“基本信息”区域的“设备名称”，检查“设备详情”区域显示了对应的设备详情。显示正确，则说明组件与模型的绑定，以及下拉框on-change事件执行正确。
- 检查是否正常提交工单。

填写工单的要素，单击“提交”。如果页面跳转到工单列表页面，且在查询结果中，显示新创建的工单，则说明提交按钮的“点击”事件执行正确。

如果页面跳转到工单列表页面，且在查询结果中显示新创建的工单，则说明验证成功。

图 3-374 验证结果样例

现场工单列表

工单标题	状态	操作						
<input type="text"/>	<input type="text"/>	<input type="button" value="搜索"/>						
<input type="button" value="+ 创建工单"/>								
#	工单id	优先级	工单标题	状态	当前处理人	创建时间	创建人	
<input type="checkbox"/>	1	维修_160007055...	高	电梯开门延时	待派单	派单员	2020-09-14 16:0...	h100000000

步骤2 验证派单功能。


- 访问图3-373中的应用登录页，使用派单人员账号登录（testds）。
- 登录后，进入“工单列表（派单员）”页面。
正常情况下，系统会显示客服人员创建的工单，且每条记录后都有派单图标。
- 选择上一步新建的一条工单记录，并单击该记录的  派单，弹出处理工单弹窗。

图 3-375 状态为“待派单”工单记录



4. 在处理工单弹窗中，设置下拉框“选择工程师”为“testms”，然后单击“确定”，返回“工单列表（派单员）”页面。

查看工单记录的“状态”、“当前处理人”是否已更新为“待接单”、“testms”。如果已更新，说明派单流程正常。

图 3-376 处理工单-派单



步骤3 验证待接单单。

1. 访问图3-373中的应用登录页，使用派单人员账号登录（testms）。
2. 登录后，进入“工单列表（维修人员）”页面。
正常情况下，系统会显示派单员在上一步中派发的工单，如图3-377所示。

图 3-377 待处理工单



3. 单击“处理”，进入处理工单弹窗，“选择下一步操作”设置为“接单”，单击“提交”。

处理完成后，返回“待处理工单”页面，查看“状态”是否已经更新为“处理中”。

如果有多条工单，“待处理”状态的工单，优先显示，“处理中”的工单可能会显示在页面的下面，工单状态改变后，可以拖动滚动条查找该工单。

📖 说明

如果在“选择下一步操作”中，选择了“拒单”，流程将返回“派单员”处，状态将变为“待派单”。

图 3-378 处理工单-关单



图 3-379 查看处理中的工单



4. 单击“处理中”工单的“处理”，选择“关单”，单击“提交”，返回页面后，查看该条工单的状态是否变成“关闭”，如果已关闭，则说明关单流程正常。

图 3-380 关闭工单



步骤4 管理工单及设备监控。


1. 使用应用开发者账号，访问并登录AstroZero经典版开发环境。
2. 在“项目 > 我的应用”中，单击“设备维修管理系统”应用的 ，进入应用预览页面。

图 3-381 查看应用



- 单击“工单管理”，进入工单管理页面，进行“创建工单”、派单及删除工单操作，验证工单管理功能。

步骤5 验证设备监控功能。


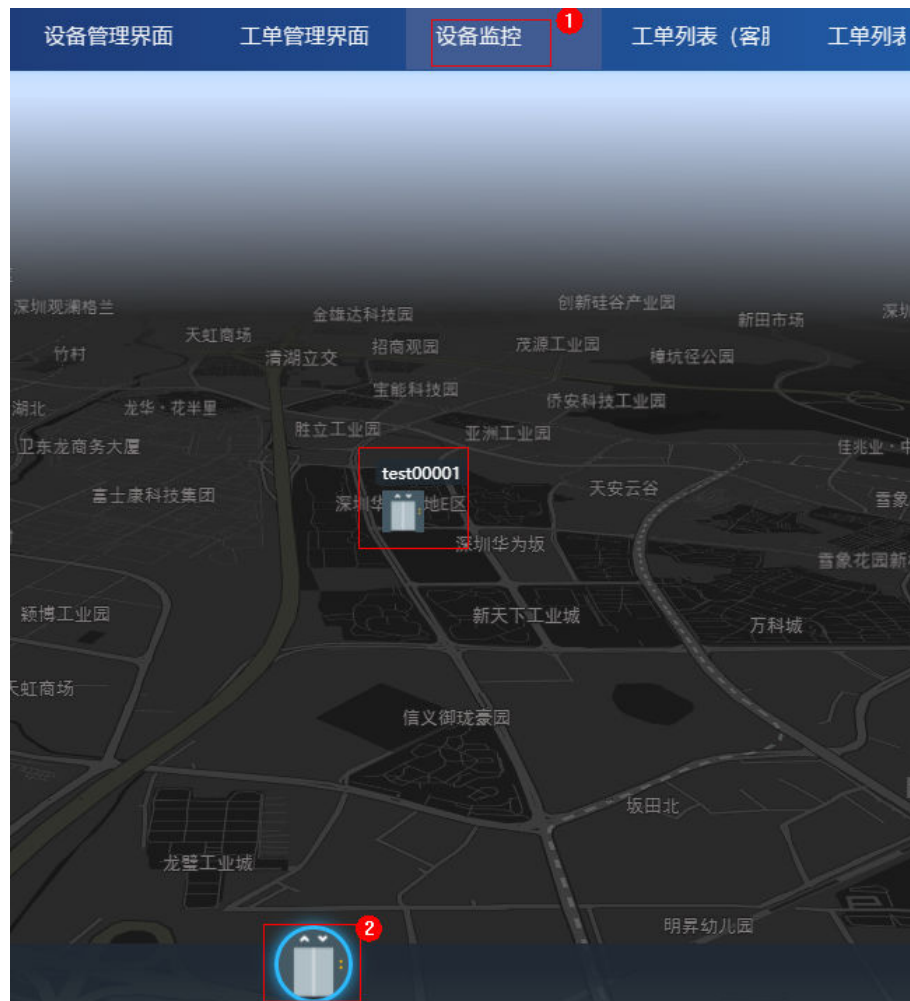
单击“设备监控”，在页面上单击图标，查看地图中是否显示已添加的电梯设备的实时位置，显示实时位置则说明页面配置正确。

图 3-382 实时监控效果



----结束

3.1.10 打包发布

⚠ 注意

免费试用版本未开通运行环境权限，可能无法正常体验打包发布功能。建议您[升级为专业版](#)后，继续体验本示例。

在应用开发完成后，应用需要编译、打包、发布，既可以发布到当前租户的运行环境，也可以共享给其他租户，在其他租户的开发环境或运行环境下安装。

发布须知

- 应用包编译类型

在AstroZero经典版应用设计器中，支持编译资产包和源码包两种的应用包。

- 资产包：资产包中组件默认是设置为受保护状态，可以在运行环境中安装后，进行预览，运行使用。也可以分享给其他租户安装，在其他租户的开发环境安装后，将显示在“库”页签下，资产包安装后仅能浏览不支持修改。

- 源码包：源码包中的组件不受保护和限制。源码包可以分享给其他租户，其他租户安装在开发环境中后，可以进行二次开发，安装后将显示在“项目”页签下，详细操作请参见[在其他租户环境安装应用包](#)，另外，运行环境中不能安装源码包。
- **发布方式**
AstroZero应用（资产包）发布有多种发布方式：我的仓库、Welink、华为OneMobile、微信、下载小程序，详细介绍请参考[发布App](#)。
本示例中，主要介绍把应用发布到“我的仓库”，并通过“我的仓库”再部署到运行环境。

📖 说明

编译类型为“源码包”时，只有发布到“我的仓库”一种发布方式。

编译与发布




- 步骤1** 使用华为账号，访问并登录AstroZero经典版开发环境。
- 步骤2** 在“项目 > 我的应用”中，单击“设备维修管理系统”，进入应用。
- 步骤3** 单击左下角的，选择“设置”。
- 步骤4** 在包类型中，选择“资产包”，单击“保存”。

图 3-383 设置包类型



- 步骤5** 单击，选择“编译”，进行资产包编译。
若想了解更多关于发布应用的设置，请参考[如何编译发布应用](#)。
- 步骤6** 编译完成后，单击左下角，选择“我的仓库”，将应用程序安装包发布到当前租户的私有仓库。
- 步骤7** 填写版本信息，单击“发布”。

发布成功后，页面显示“程序包已经被成功上传到我的仓库。”。

----结束

在运行环境安装应用

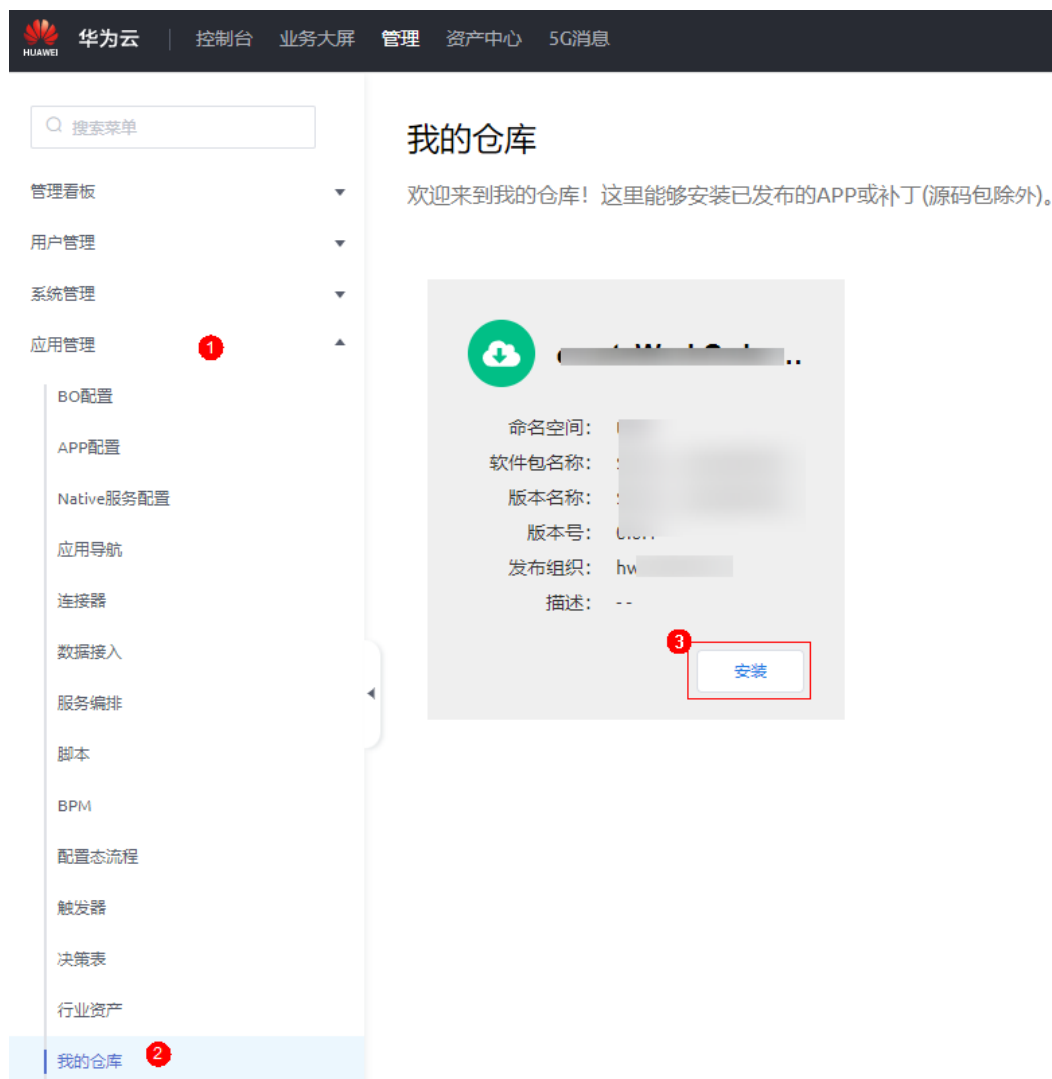
步骤1 使用开发者账号，访问并登录AstroZero经典版运行环境。

步骤2 在经典版运行环境首页，单击“我的仓库”或者在右侧菜单中，选择“应用管理 > 我的仓库”，在右侧中，找到**编译与发布**中发布的“设备维修管理系统”应用，单击“安装”。

说明

如果“我的仓库”下没有待安装的“设备维修管理系统”应用，您需要参考**编译与发布**中的操作方式，编译一个“资产包”应用，并选择发布到“我的仓库”。

图 3-384 在“我的仓库”安装设备运行管理应用



步骤3 预览应用。



1. 在左侧导航栏中，选择“应用管理 > 应用导航”。
2. 在右侧中，找到**编译与发布**中发布的“设备运维管理”应用，单击 ，进入应用布局页面，设置布局后，单击“保存”。
3. 单击 ，预览管理员的应用页面。

图 3-385 应用程序列表



步骤4 (可选) 获取登录页面URL，并分享登录页给应用业务用户。

说明

如果需要运行应用，则需要运行环境“管理”中[创建工作队列](#)、[创建权限配置](#)，再参考[应用业务测试](#)中的步骤运行应用。


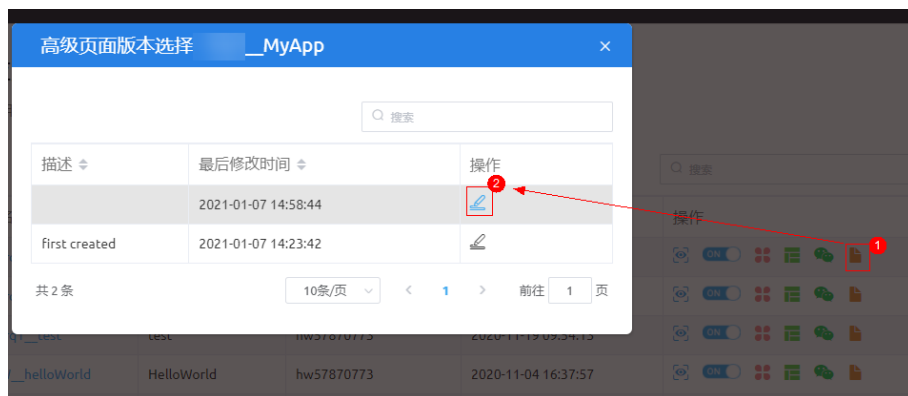
1. 在“应用程序列表”的“设备运维管理”应用中，单击，进入应用，查看应用的登录页面。

图 3-386 查看高级页面入口




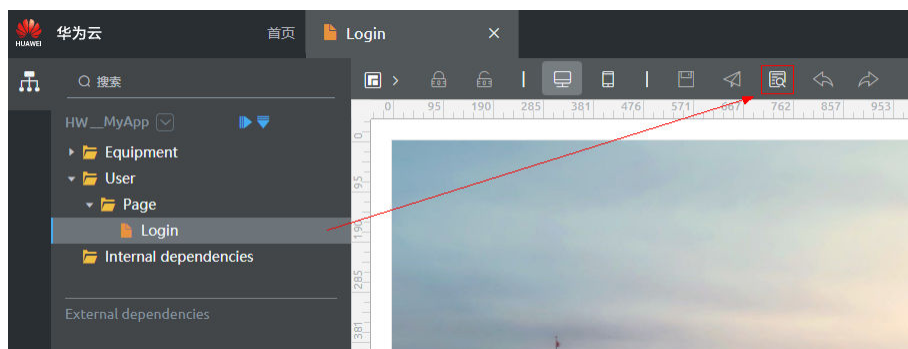
2. 在“Page”中，单击“Login”页面，在页面中单击预览登录页面。

图 3-387 预览查看登录页 URL

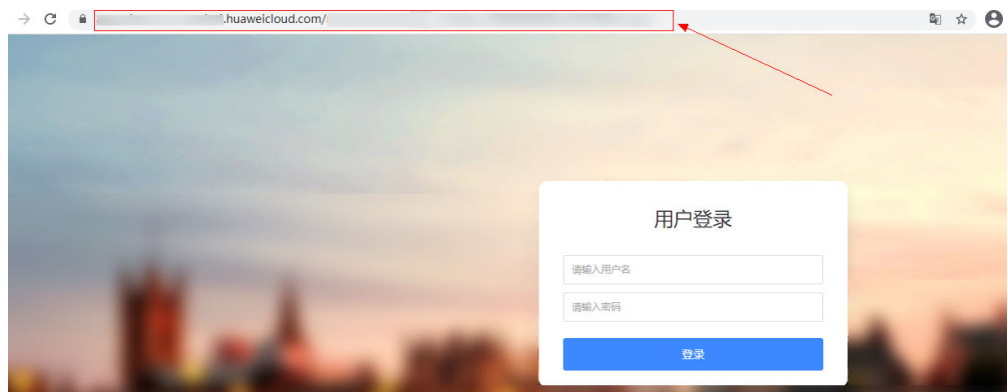


3. 预览之后进入登录页，复制登录页URL，并记录此地址。
将登录页URL分享给业务用户，业务用户即可使用账号及密码进行登录。

📖 说明

如果您还没有业务用户，您需要参考[管理业务用户](#)章节进行添加。

图 3-388 复制应用登录页 URL




----结束

在其他租户环境安装应用包

当其他租户需要安装此编译后的资产包或源码包时，请参考以下步骤进行安装。

步骤1 当前租户下载资产包或源码包。

在当前华为账号开发环境中，进入设备管理应用，在页面左侧单击，在“已发布”下，单击“资产包”或“源码包”，查看已发布的APP，在右侧页面单击“下载”。

步骤2 将下载后的应用包分享给其他租户。

步骤3 在其他华为账号下，以导入方式，安装应用包。

1. 使用华为账号，登录AstroZero经典版开发环境。
2. 在经典版开发环境首页，单击“管理”，进入AstroZero经典版开发环境管理中心。
3. 在左侧导航栏中，选择“应用管理 > 软件包管理 > 软件包安装”。
4. 单击“新建”，在“软件包安装”页面，拖入待安装的应用包，单击“安装”。
如果安装的是资产包，请在“库”页签下查看。如果安装的是源码包，请在“项目”页签下的“我的应用”中查看。

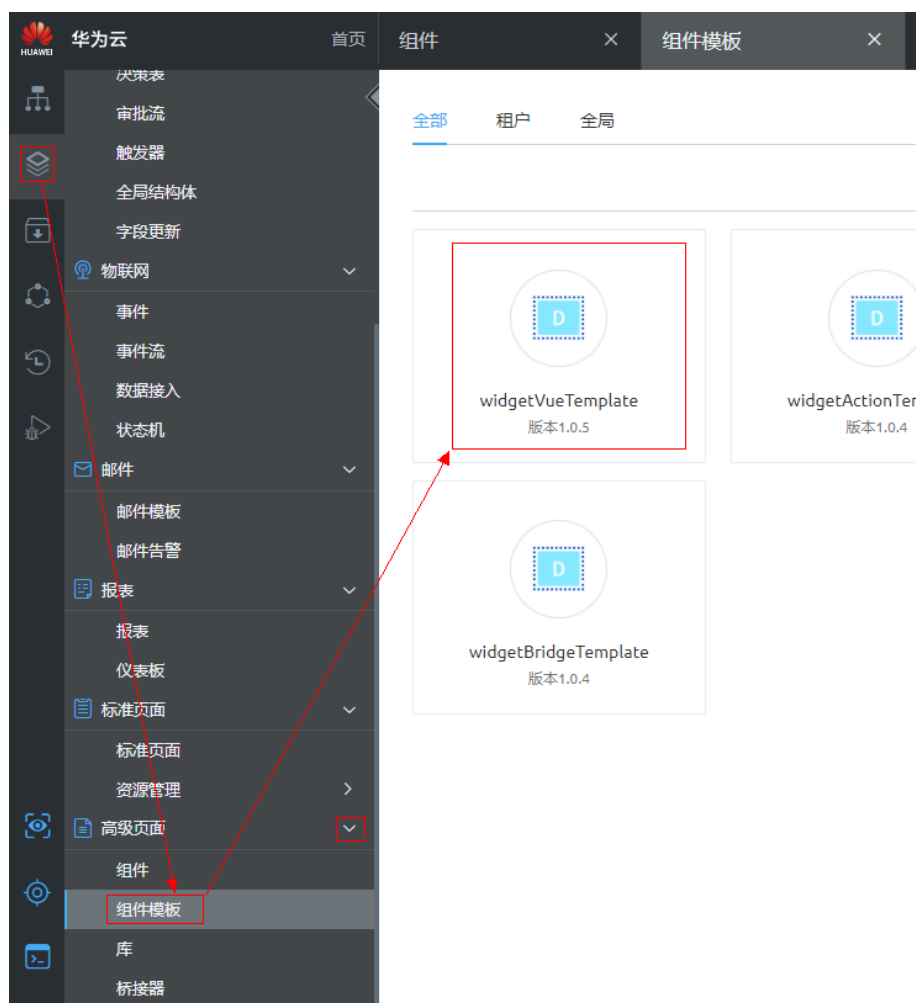
----结束

3.1.11 FAQ

3.1.11.1 线下开发自定义登录组件

步骤1 在左侧资产下的组件模板中，单击widgetVueTemplate，再单击“下载”。

图 3-389 下载组件模板



步骤2 在弹出的窗口中，输入组件名称，并单击“保存”，将组件模板保存到本地，并解压。

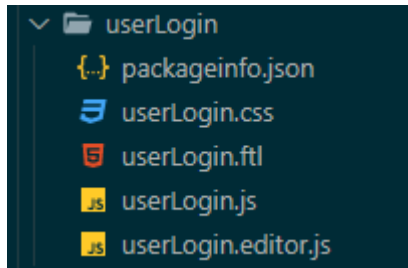
图 3-390 输入组件名称



步骤3 查看解压后的组件目录。

目录结构如**图3-391**所示，其中userLogin.js文件是写vue业务逻辑的代码，userLogin.ftl用于写html代码，userLogin.css用于写样式代码，userLogin.editor.js以及packageinfo.json是配置文件。

图 3-391 目录结构



步骤4 在本地编辑器中打开文件夹，把userLogin.editor.js文件中的config代码改为如下代码，用于配置桥接器。

```
config: [
  {
    type: 'connectorV2',
    name: 'FlowConnector',
    label: 'Flow Connector',
    model: 'ViewModel'
  },
  {
    type: 'connectorV2',
    name: 'common.GetConnector',
    label: 'View API Get Connector',
    model: 'ViewModel'
  },
  {
    type: 'connectorV2',
    name: 'common.PostConnector',
    label: 'View API Post Connector',
    model: 'ViewModel'
  },
  {
    type: 'connectorV2',
    name: 'common.PutConnector',
    label: 'View API Put Connector',
    model: 'ViewModel'
  },
  {
    type: 'connectorV2',
    name: 'common.DeleteConnector',
    label: 'View API Delete Connector',
    model: 'ViewModel'
  }
]
```

步骤5 把packageinfo.json文件中加入如下**加粗内容**。

```
{
  "widgetApi": [
    {
      "name": "userLogin"
    }
  ],
  "widgetDescription": "",
  "authorName": "",
  "localFileBasePath": "",
  "requires": [
    {
```

```
    "name": "global_Vue",  
    "version": "100.8.3"  
  },  
  {  
    "name": "global_Vue18n",  
    "version": "100.7.3"  
  },  
  {  
    "name": "global_Element",  
    "version": "101.0.3"  
  }  
]  
}
```

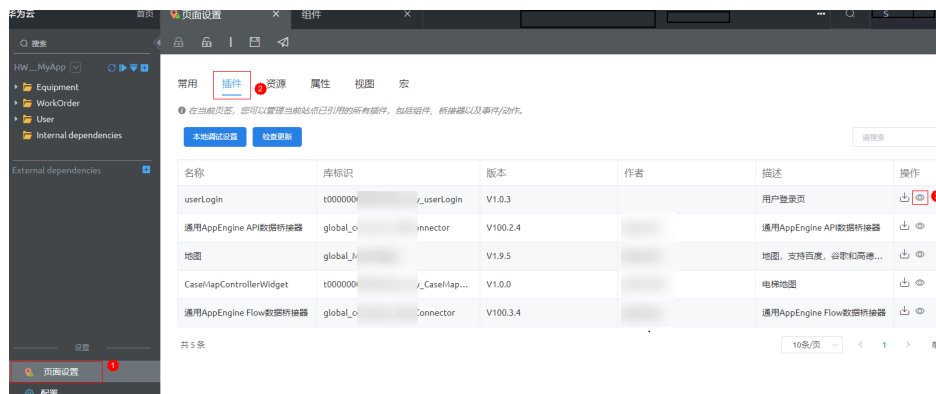
步骤6 将修改后的组件文件压缩成一个zip包。压缩后，即可根据需要上传到AstroZero，供高级页面使用。

----结束

3.1.11.2 如何更新自定义组件

步骤1 在应用中，单击左下方的“页面设置”，再选择“插件”页签，找到需要更新的组件（例如userLogin），单击组件所在行右侧“查看详情”，进入组件详情页。

图 3-392 页面设置下的组件列表



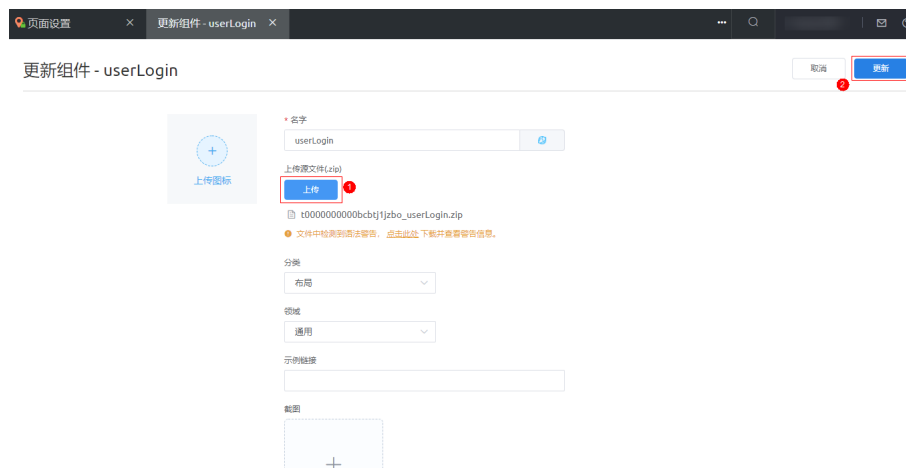
说明





如果页面  图标高亮，则需要先单击  解锁页面。

步骤2 单击“更新”按钮进入组件更新页面。



步骤3 单击“上传”，上传本地的组件zip包，再单击“更新”。



步骤4 返回“插件”页签，单击 获取锁，再单击刚刚上传组件（userLogin）的升级按钮，然后单击 保存升级，最后单击 发布即可。



----结束

3.1.11.3 派单不成功，报错“无法获取 rootID”

问题描述

派单页面进行派单时，派单状态不改变，并报以下错误：

```
resCode: "405233002"
```

```
resMsg: "无法从请求的URL中获取正确的参数: 'rootID'"
```

可能原因

当前操作的工单信息是在“生成工单”（createWorkOrder）脚本中，使用测试数据创建的，脚本中测试数据创建的工单不支持在BPM状态流转。

解决方法

删除所有使用脚本创建的测试数据。使用创建工单页面（workOrderList）进行创建。

3.2 告警监控大屏应用开发

3.2.1 场景说明

当用户需要开发业务大屏页面，使用可视化大屏进行项目运营管理、业务监控、风险预警时，则需创建业务大屏应用来进行大屏页面的开发。本章节以开发告警监控大屏应用为例，介绍典型的大屏开发流程。

告警监控大屏应用场景：产品经理、运营人员需要使用大屏，来查看全国范围的告警监控。大屏页面构思，如图3-393所示。

图 3-393 大屏页面构思图



页面由下列组件拼装而成，组件介绍顺序为从上到下、从左到右。

表 3-54 组件说明

组件	说明
文本编辑	文本编辑组件，用于展示标题，例如告警监控大屏、各区域未处理告警处理情况
基本柱图	分状态进行告警处理情况统计，状态分：待派单、已派单、处理中、已关闭
区域图	统计告警处理平均时长
地图	分地区统计告警数量
基本饼图	分类型统计告警数量
区域图	统计告警数量趋势

3.2.2 前提条件

使用说明

大屏页面中的基本柱图、区域图和基本饼图组件显示的数据来自于后台接口，您需要在该开发页面配置所有组件用到的桥接器，配置前，需要获取所有组件调用后台的自定义接口URL。

配置数据来源并导入告警数据


- 步骤1 单击[下载链接](#)，获取应用包“Mock.zip”。
- 步骤2 在AstroZero服务控制台，单击“进入首页”，进入新版应用开发页面。
- 步骤3 单击左上角的，选择“开发环境管理 > 环境配置”，进入AstroZero开发环境的环境配置。

图 3-394 进入 AstroZero 开发环境的环境配置



步骤4 在左侧导航栏中，选择“应用管理 > 安装管理 > 包安装”，单击“新建”。

步骤5 拖入已下载的应用包，单击“安装”。

如何安装软件包，请参见[安装软件包](#)。

步骤6 单击[下载链接](#)，获取告警数据包“hab_INF_WARN_CST.zip”，在“维护 > 应用维护 > 数据导入 > 数据导入”中，单击“新建”，将告警数据包拖入进行导入。


等待一分钟后，单击操作列，数据导入页面中会更新导入状态，“Completed”表示导入成功。如果数据导入失败，您可以从邮件内容或导入页面“详情”中获知失败原因。

图 3-395 数据导入



步骤7 在左侧导航栏中，选择“全局元素 > 自定义接口”，查看自定义接口URL。

请记录该接口URL，在本示例中创建大屏应用时需要使用，如[表3-55](#)所示。

图 3-396 自定义接口 URL



表 3-55 组件桥接器配置说明

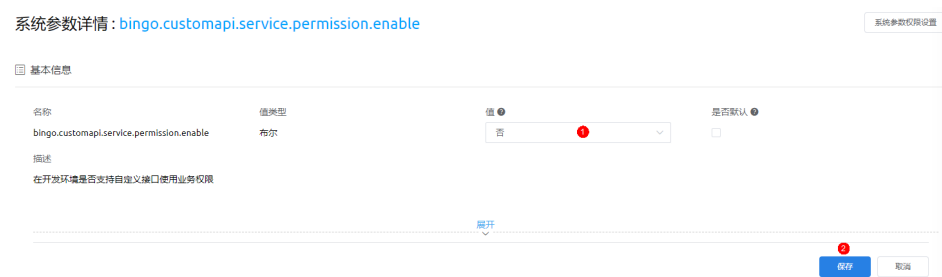
组件	桥接器	自定义接口URL	功能说明
基本柱图	柱状图和折线图数据桥接器	/service/hab_mock/1.0.1/WarnAmountPerStatus	分状态进行告警处理情况统计，状态分：待派单、已派单、处理中、已关闭。

组件	桥接器	自定义接口URL	功能说明
区域图	柱状图和折线图数据桥接器	/service/hab__mock/1.0.1/WarnCostPerDay	统计告警处理平均时长。
基本饼图	饼图数据桥接器	/service/hab__mock/1.0.1/WarnAmountPerType	分类型告警数量统计接口。
区域图	柱状图和折线图数据桥接器	/service/hab__mock/1.0.1/WarnAmountPerDay	统计告警数量趋势。

步骤8 在主菜单中，选择“配置”，返回配置页面。

步骤9 在左侧导航栏中，选择“系统设置 > 系统参数 > 内置系统参数”，配置系统参数“bingo.customapi.service.permission.enable”值为“否”，关闭开发环境中对公共接口业务权限凭证的校验。

图 3-397 设置 bingo.customapi.service.permission.enable 为否



----结束

3.2.3 开发大屏项目


前提条件

已参考**前提条件**中操作，申请百度地图密钥，导入告警数据。

开发大屏页面

步骤1 使用华为账号，登录[华为云网站](#)，在顶部导航栏右侧单击“控制台”，进入华为云控制台。

步骤2 在左侧导航栏上方，单击，选择服务实例所在的区域项目。

步骤3 单击，在查找框中搜索“Astro轻应用”，单击查找到的结果，进入AstroZero服务控制台。

步骤4 在AstroZero服务控制台，单击“进入首页”，进入AstroZero新版应用开发页面。


步骤5 在主页中，单击“业务大屏”中的，即可进入新建项目页面。

图 3-398 进入新建项目页面

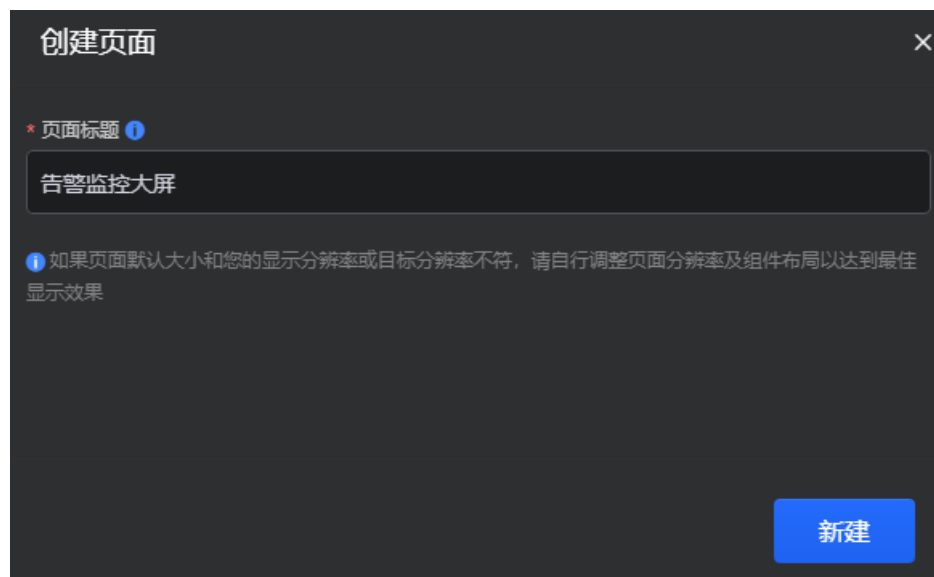


步骤6 根据需要选择项目类型，这里选择“大屏&PC端项目”，输入不重复的项目名称，例如“Alarm”，单击“新建”，进入项目开发界面。

步骤7 在大屏&PC端项目页面中，单击“+ 新建页面”，选择“新建空白页面”。

步骤8 输入页面标题“告警监控大屏”，单击“新建”。

图 3-399 创建页面




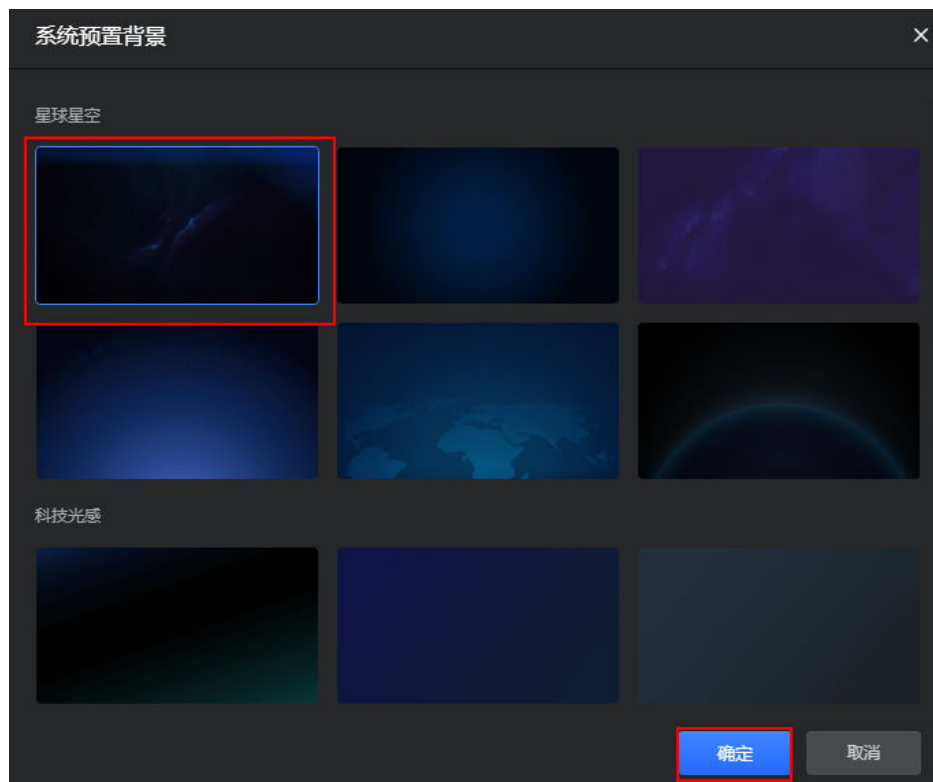
步骤9 在页面上方单击，根据实际需要设置背景图片。

图 3-400 设置背景图片



图 3-401 选择背景图



步骤10 从全部组件中，分别拖拽文本编辑、基本柱图、地图、基本饼图和两个区域图到画布区域，布局如图3-403所示。

图 3-402 拖拽文本编辑组件到画布

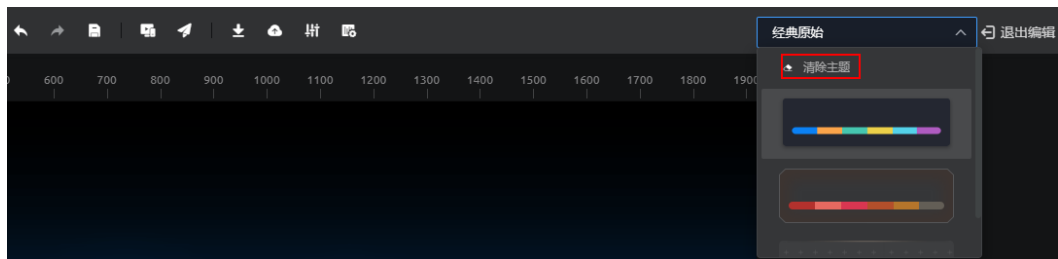


图 3-403 组件布局



步骤11 在页面右上角，根据需要选择所需的主题（本示例设置为经典原始，即单击“清除主题”）。

图 3-404 设置页面主题



步骤12 配置文本编辑组件，设置“告警监控大屏”标题。

1. 选中上方文本编辑组件，单击上方 ，设置文本内容为“告警监控大屏”，字体设置为“60”，字体颜色设置为“rgba(255, 255, 255, 1)”。

图 3-405 设置文本编辑组件



2. 在段落设置中，设置标题对齐方式为“居中”。

图 3-406 设置对齐方式



步骤13 配置基本柱图组件样式，配置数据源。


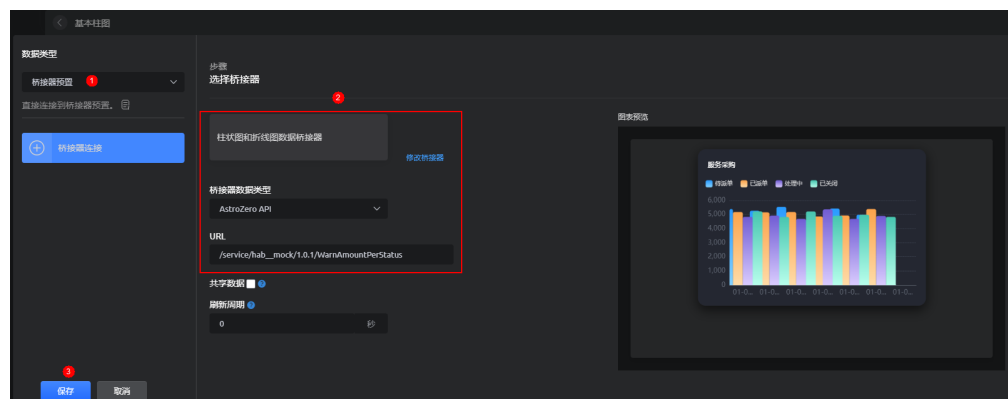
1. 选中基本柱图组件，在组件上方单击，配置组件数据源。

图 3-407 配置桥接器



桥接器参数说明，如表3-56所示。

表 3-56 参数说明

参数	说明
数据配置	设置组件在页面呈现数据的来源类型，本示例请选择“桥接器预置”。 更多关于组件数据配置的介绍，请参见 页面数据来源介绍 。
选择桥接器	调用的桥接器名称，从下拉框选择系统提供的预置桥接器“柱状图和折线图数据桥接器”。
桥接器数据类型	桥接器的数据源类型，支持“静态数据”和“AstroZero API”。 本示例选择“AstroZero API”。

参数	说明
URL	<p>“桥接器数据类型”设置为“AstroZero API”时，该参数才会显示。</p> <p>请配置为自定义接口URL“/service/hab_mock/1.0.1/WarnAmountPerStatus”，即表3-55中公共接口URL。</p> <p>须知 在拷贝复制参数取值样例时，请检查复制出来的数据是否准确，不能有多余空格。</p>
刷新周期	<p>“数据源类型”设置为“桥接器预置”时，该参数才会显示。</p> <p>每隔多少秒调用一次后台接口，默认配置为“0”，表示只调用一次。</p> <p>配置为“30”。</p>
共享数据	<p>是否共享数据，“桥接器数据类型”设置为“AstroZero API”时，该参数才会显示。当项目里多个组件通过一个公共请求调用同一个接口的数据时，为了避免多次调接口，可以选择开启共享数据。</p> <p>本示例不勾选。</p>

- 选中基本柱图组件，在组件上方单击，设置模板为“蓝色渐变不排序”。

图 3-408 设置模板




- 选中基本柱图组件，在组件上方单击，将每个数据系列里“数据标签”设置为“上”，让数据显示在柱形图上方。

图 3-409 数据标签设置为上



4. 打开“水平堆叠”与“横向柱条”效果。

图 3-410 设置堆叠显示



5. 选中基本柱图组件，在组件上方单击，隐藏y轴网格线。

图 3-411 隐藏 y 轴网格线




- 选中基本柱图组件，在组件上方单击，设置组件标题内容为“告警处理情况统计”。

图 3-412 设置组件标题



图 3-413 设置后效果



步骤14 参考步骤13，配置左下角区域图组件。


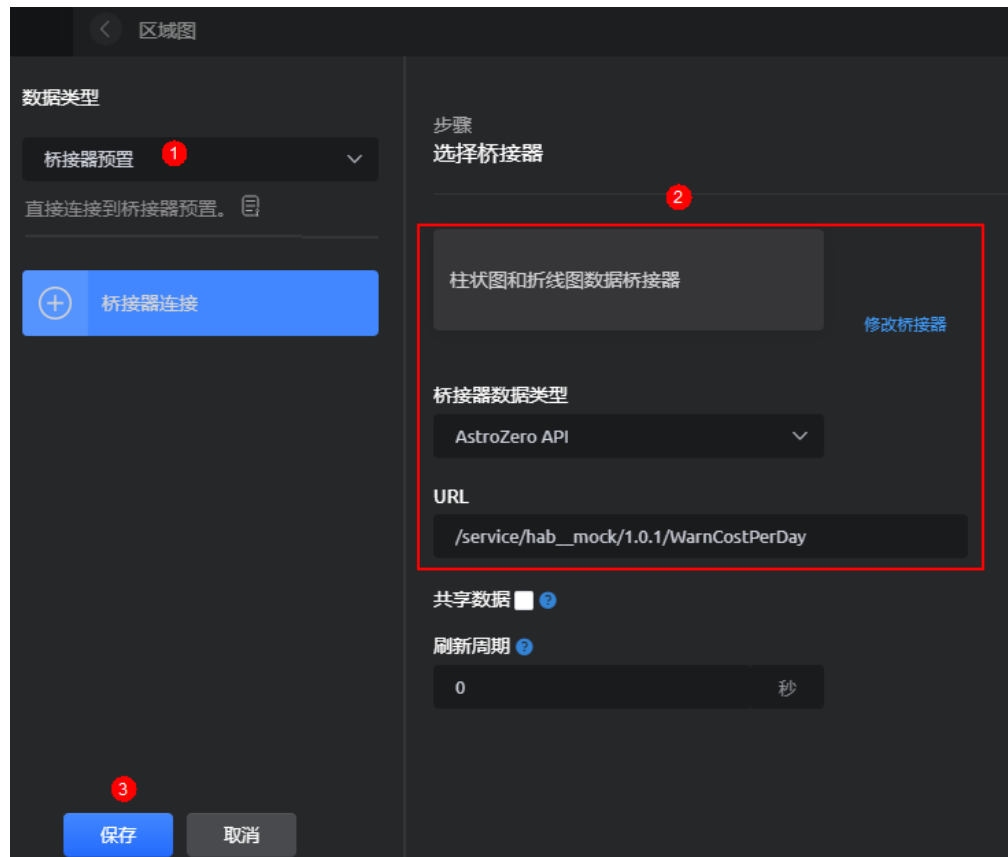
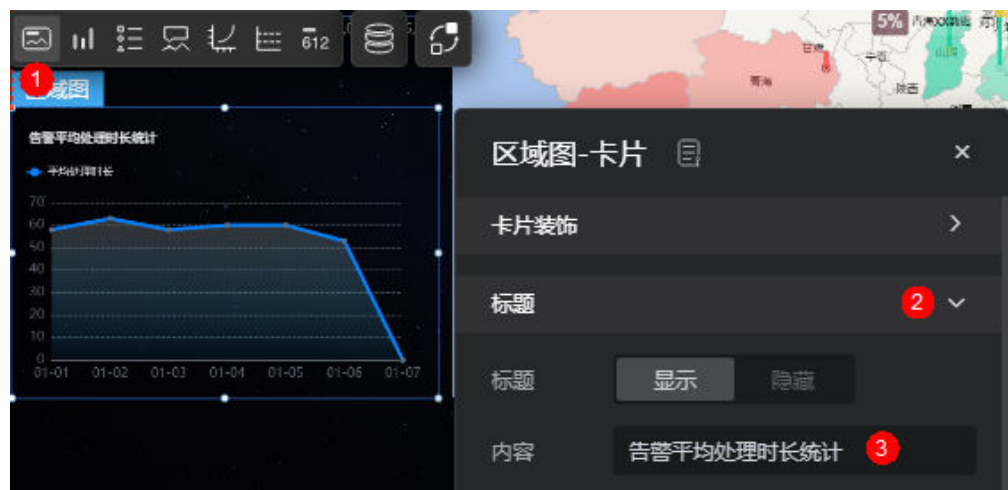
1. 选中左下角区域图组件，在组件上方单击，配置组件数据源。
数据源选择“桥接器预置 > 柱状图和折线图数据桥接器”，“桥接器数据类型”配置为“AstroZero API”，URL配置为“/service/hab_mock/1.0.1/WarnCostPerDay”（在拷贝复制参数取值样例时，请检查复制出来的数据是否准确，不能有多余空格）。

图 3-414 设置数据源



- 选中基本折线图组件，单击组件上方的，设置标题“内容”为“告警平均处理时长统计”。

图 3-415 设置标题




- 单击组件上方的，设置Y轴的轴线为“显示”。

图 3-416 显示轴线



步骤15 参考步骤13，配置地图。



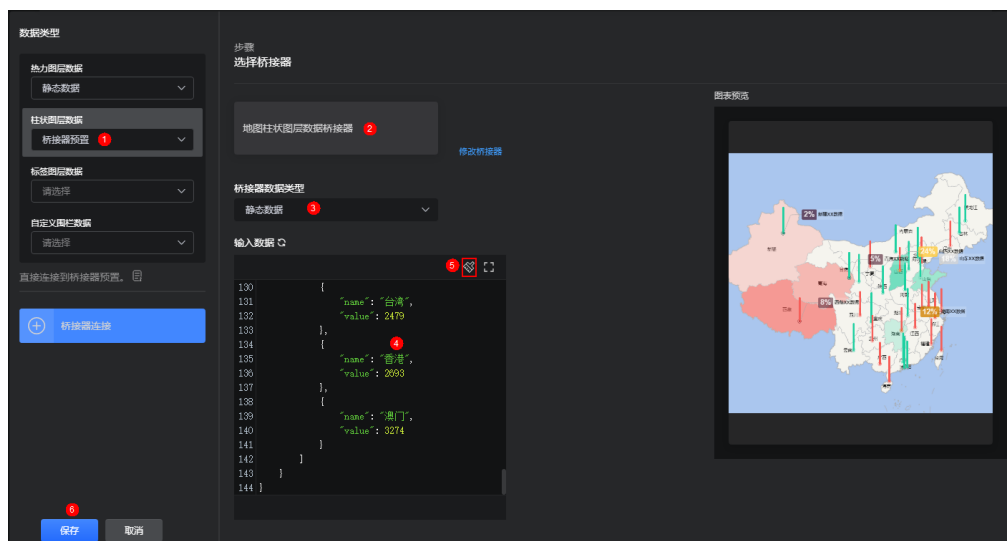
- 选中地图组件，在组件上方单击，配置组件数据源。
配置“柱状图数据/桥接器预置 > 地图柱状图数据桥接器”实例，“桥接器数据类型”配置为“静态数据”，输入示例数据后，单击格式化数据。

图 3-417 配置桥接器数据



```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "name": "柱状图层数据",
    "data": [
      {
        "name": "北京",
        "value": 3146
      },
      {
        "name": "天津",
        "value": 2641
      },
      {
        "name": "河北",
        "value": 2880
      },
      {
        "name": "山西",
        "value": 2357
      },
      {
        "name": "内蒙古",
        "value": 2810
      },
      {
        "name": "辽宁",
        "value": 2540
      },
      {
        "name": "吉林",
        "value": 2665
      },
      {
        "name": "黑龙江",
        "value": 2748
      },
      {
        "name": "上海",
        "value": 2372
      },
      {
        "name": "江苏",
        "value": 2613
      },
      {
        "name": "浙江",
        "value": 2554
      },
      {
        "name": "安徽",
        "value": 2427
      },
      {
        "name": "福建",
        "value": 2587
      },
      {
        "name": "江西",
        "value": 2501
      },
      {
        "name": "山东",
        "value": 2810
      },
      {
        "name": "河南",
        "value": 2773
      }
    ]
  }
}
```

```
},  
{  
  "name": "湖北",  
  "value": 2814  
},  
{  
  "name": "湖南",  
  "value": 2384  
},  
{  
  "name": "广东",  
  "value": 3014  
},  
{  
  "name": "广西",  
  "value": 2232  
},  
{  
  "name": "海南",  
  "value": 2600  
},  
{  
  "name": "重庆",  
  "value": 3056  
},  
{  
  "name": "四川",  
  "value": 2439  
},  
{  
  "name": "贵州",  
  "value": 2609  
},  
{  
  "name": "云南",  
  "value": 2755  
},  
{  
  "name": "西藏",  
  "value": 2451  
},  
{  
  "name": "陕西",  
  "value": 2791  
},  
{  
  "name": "甘肃",  
  "value": 2817  
},  
{  
  "name": "青海",  
  "value": 2651  
},  
{  
  "name": "宁夏",  
  "value": 2639  
},  
{  
  "name": "新疆",  
  "value": 2690  
},  
{  
  "name": "台湾",  
  "value": 2479  
},  
{  
  "name": "香港",  
  "value": 2693  
},  
}
```

```

    {
      "name": "澳门",
      "value": 3274
    }
  ]
}

```


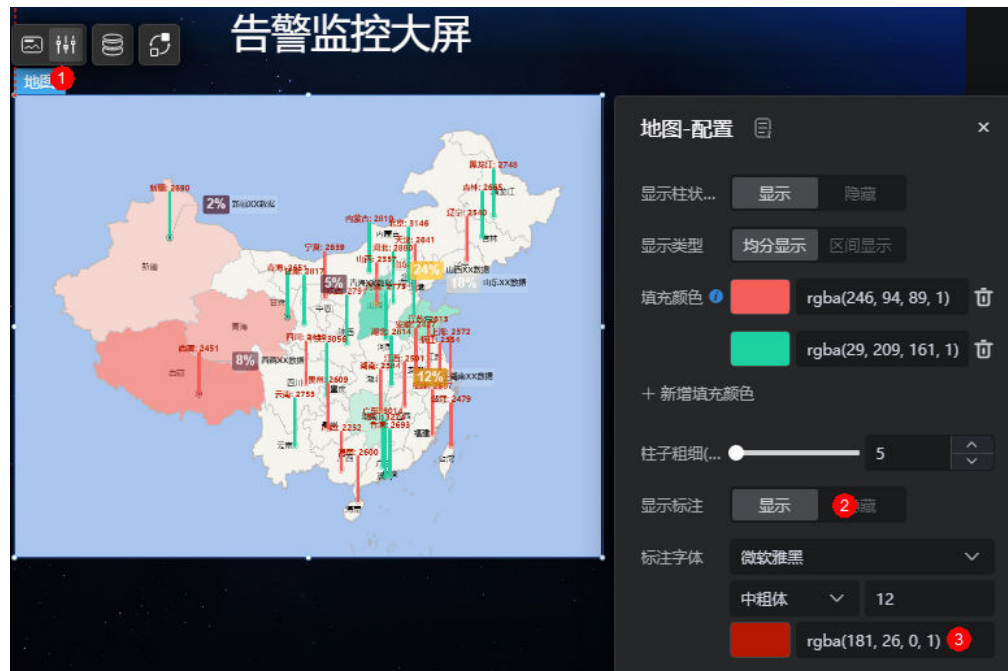
- 选中地图组件，在组件上方单击，将“柱状图层配置”中的“显示标注”设置为“显示”，标注字体颜色为“rgba(181, 26, 0, 1)”。

图 3-418 设置标注字体颜色



步骤16 参考**步骤13**，配置基本饼图。


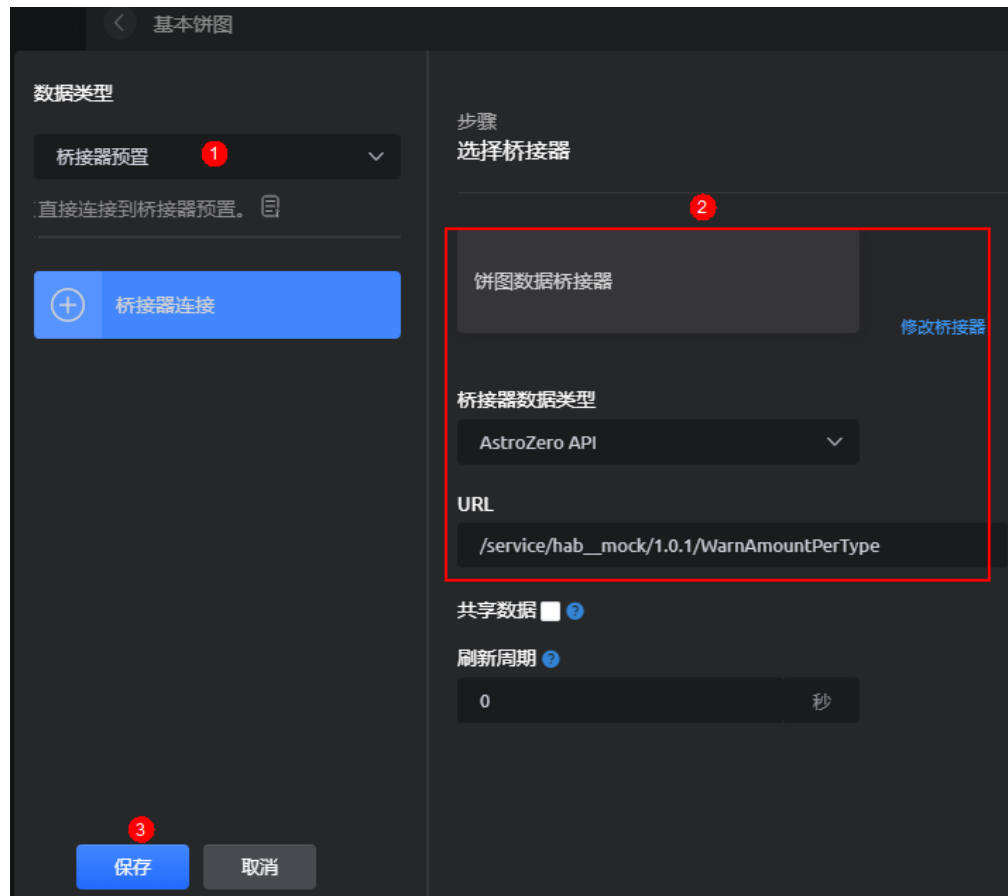
- 选中右上方基本饼图组件，在组件上方单击，配置组件数据源。
数据源选择“桥接器预置 > 饼图数据桥接器”，“桥接器数据类型”配置为“AstroZero API”，URL配置为“/service/hab_mock/1.0.1/WarnAmountPerType”（在拷贝复制参数取值样例时，请检查复制出来的数据是否准确，不能有多余空格）。

图 3-419 设置基本饼图数据源



- 选中基本饼图组件，单击组件上方的，设置标题“内容”为“告警分类统计”，参考模板配置为“带标签”。

图 3-420 设置标题



图 3-421 设置参考模板为带标签




- 选中基本饼图组件，单击组件上方的 ，在统一配置中隐藏“系列名”。

图 3-422 隐藏“系列名”




- 选中基本饼图组件，单击组件上方的，隐藏图例。

图 3-423 关闭图例



步骤17 参考步骤13，配置区域图。


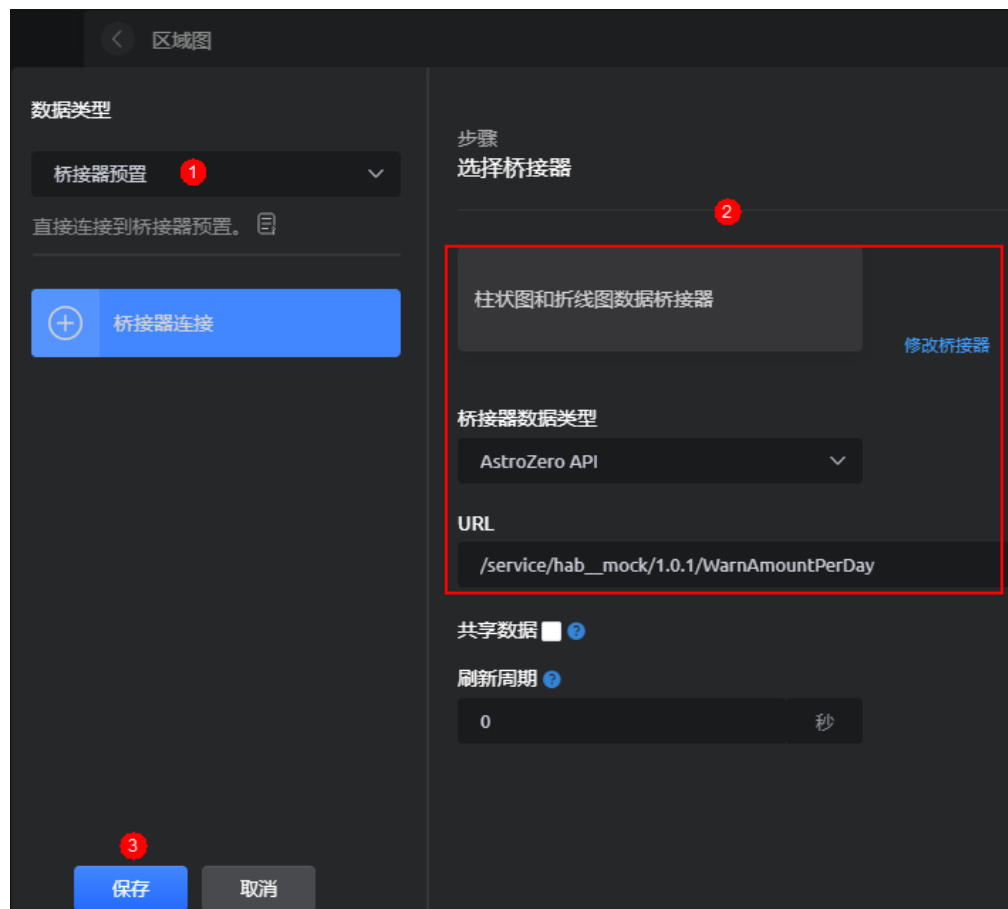
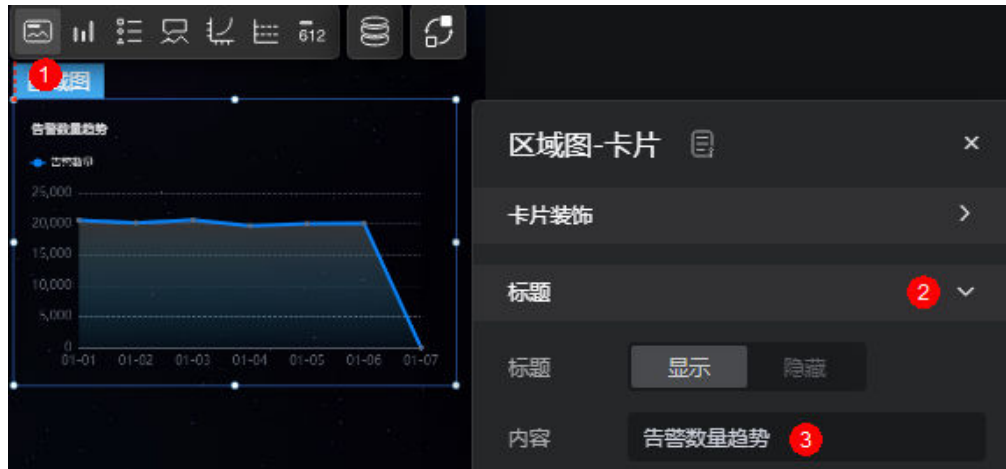
- 选中右下方区域图组件，在组件上方单击，配置组件数据源。
数据源选择“桥接器预置 > 柱状图和折线图数据桥接器”，“桥接器数据类型”配置为“AstroZero API”，URL配置为“/service/hab_mock/1.0.1/WarnAmountPerDay”（在拷贝复制参数取值样例时，请检查复制出来的数据是否准确，不能有多余空格）。

图 3-424 配置组件数据来源



- 选中区域图组件，单击组件上方的，设置标题“内容”为“告警数量趋势”。

图 3-425 设置标题内容




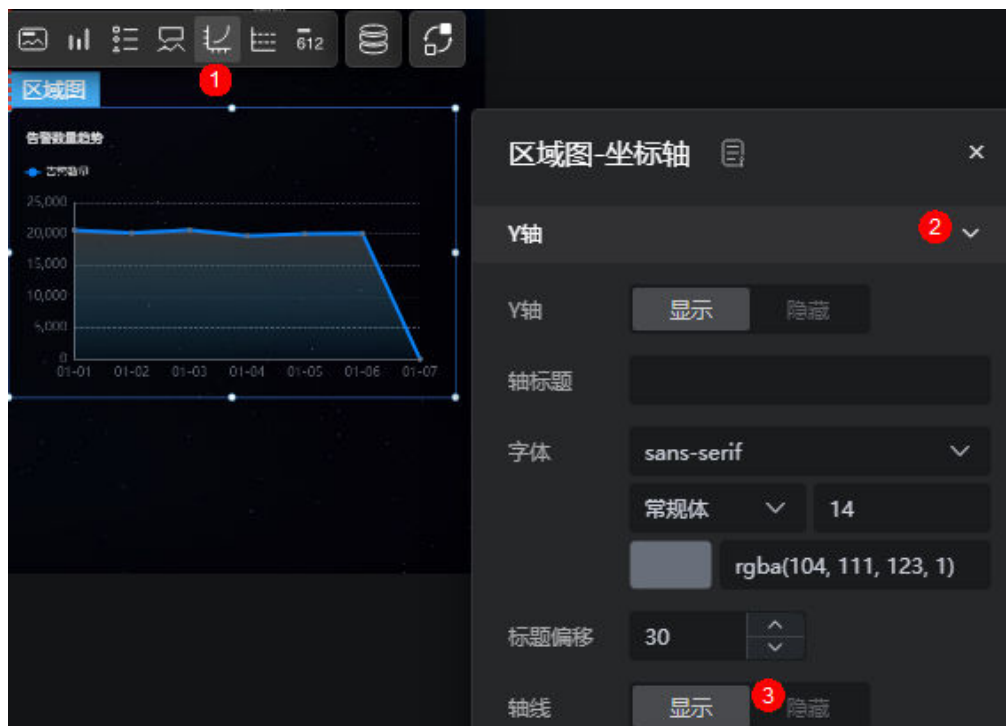
- 单击组件上方的，设置Y轴的轴线为“显示”。

图 3-426 设置 Y 轴的轴线为显示



步骤18 单击页面上方，保存后单击.

步骤19 打开“发布链接”开关，生成发布后的页面网址，记录页面成功发布网址。

步骤20 打开页面成功发布网址，检查发布效果是否符合预期。

图 3-427 查看发布效果



---结束

3.2.4 迁移大屏项目

大屏项目可以从开发环境导出，从运行环境导入，从而实现了不同环境之间的大屏资产快速迁移。

前提条件

由于大屏页面的某些数据来源于“告警服务Mock应用”中定义的后台接口，您需要将应用部署到运行环境，并导入告警数据。

步骤1 在AstroZero服务控制台，单击“进入首页”，进入新版应用开发页面。

步骤2 在左侧导航栏中，选择“应用”。

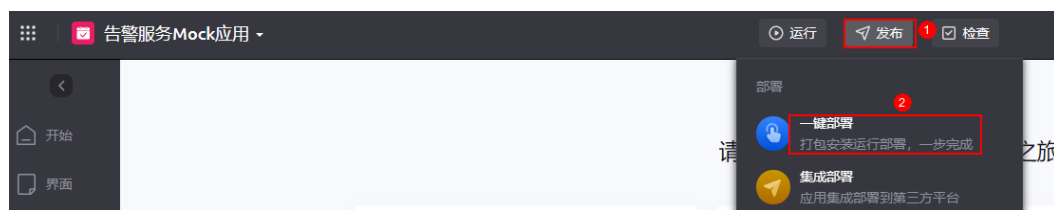
步骤3 在低代码应用列表中，单击“告警服务Mock应用”后的“编辑”，选择“体验新版设计器”。

图 3-428 选择体验新版设计器



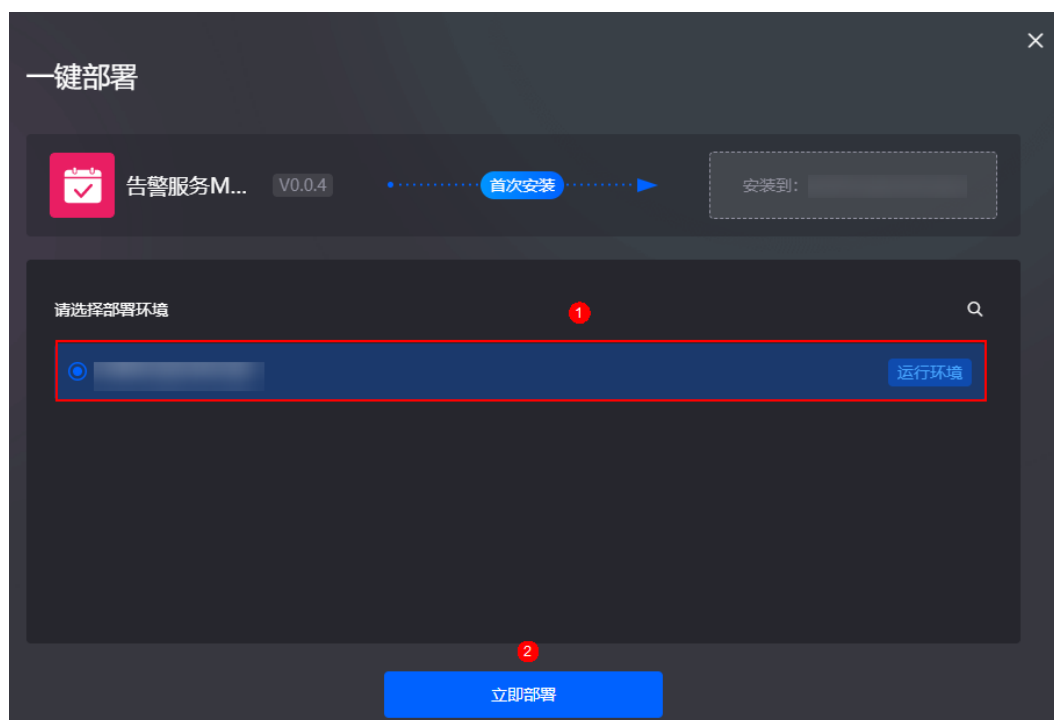
步骤4 在告警服务Mock应用的新版设计器中，单击主菜单的“发布”，选择“一键部署”。

图 3-429 一键部署应用



步骤5 在一键部署页面，选择目标运行环境，单击“立即部署”。

图 3-430 一键部署



步骤6 在弹出的确认部署页面，单击“立即部署”。

在页面左下角，可查看到当前的任务部署进度。部署成功后，显示如图3-432所示。

图 3-431 查看部署进度



图 3-432 部署成功



步骤7 单击[下载链接](#)，获取告警数据包“hab_INF_WARN_CST.zip”。

步骤8 参考[如何登录新版环境配置](#)中操作，登录AstroZero新版运行环境的环境配置。

步骤9 在“维护 > 应用维护 > 数据导入 > 数据导入”中，单击“新建”。

步骤10 拖入[步骤7](#)中获取的告警数据包，单击“导入”。


等待一分钟后，单击操作列，数据导入页面中会更新导入状态，“Completed”表示导入成功。如果数据导入失败，您可以从邮件内容或导入页面“详情”中获知失败原因。

图 3-433 数据导入

数据导入

当您准备好数据文件后，按照导入向导的提示，完成数据的导入。

文件名	状态	描述	结果	导入详情	创建人	创建日期	最后修改日期	所有者	操作
hab_INF_WARN_CST.zip	Completed	成功	-	详情		2022-01-06 09:14:52	2022-01-06 09:15:18		

----结束

操作步骤

参考[发布及安装大屏项目](#)中操作，迁移大屏项目到运行环境。

4 高阶实战

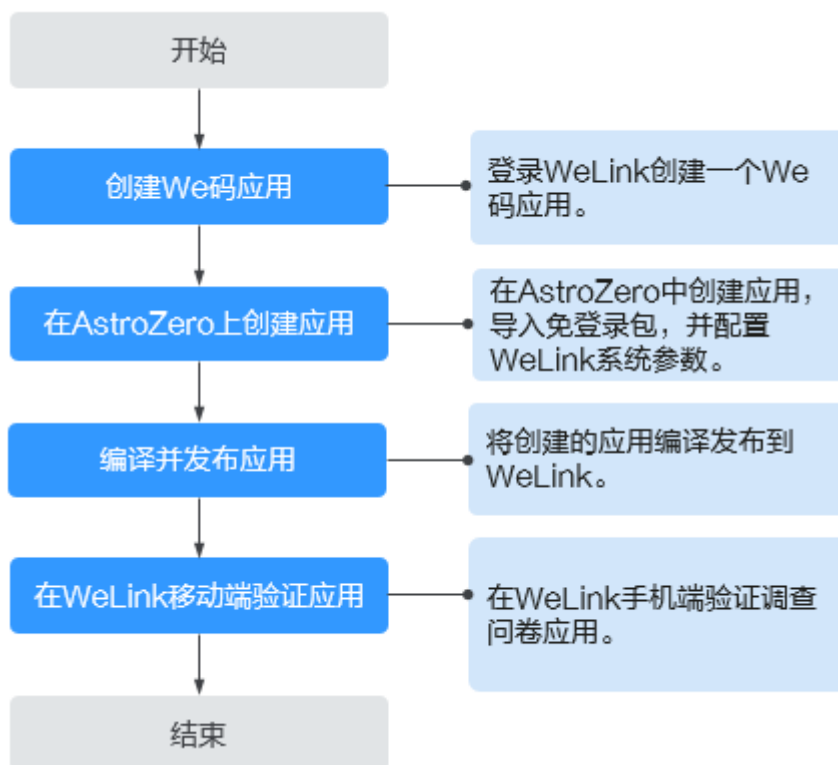
4.1 发布 WeLink We 码应用

目的

AstroZero上开发的应用可以发布成WeLink（蓝标）We码应用。We码是将前端的静态资源放入WeLink，都是本地访问，适用于对页面加载速度要求比较高的场景。关于WeLink We码应用的详细介绍，可参见[WeLink开发之旅](#)。

以创建“调查问卷应用”为例，向您介绍如何发布We码应用，发布流程如[图4-1](#)所示。

图 4-1 We 码应用发布流程



前提条件

- 已注册华为账号并开通华为云，且已完成实名认证。
- 已开通AstroZero服务，否则请[注册租户](#)中操作注册并申请试用。
- 您需要有一个WeLink小程序开发者账号，并已下载WeLink移动端。

如果没有这样的账号，请先登录华为云申请开通[WeLink](#)服务。开发者账号由对应的租户管理员分配。如何分配或想了解WeLink开放平台配置We码小程序的更多详细操作，请参考[We码小程序参考文档](#)。

步骤 1：创建 We 码应用

在WeLink端发布小程序之前，需要以WeLink管理员账号登录WeLink开发平台，创建一个We码应用。

步骤1 打开浏览器，访问[WeLink开放平台](#)。

图 4-2 访问 WeLink 开放平台



步骤2 在图4-2中单击右上角用户名后的 ▾，选择“进入开发者后台”。

📖 说明

如果使用已有账号直接登录WeLink开放平台，此处请选择“登录开发者后台”，并通过WeLink移动端扫码或输入WeLink开发者账号密码登录。

图 4-3 登录开发者后台



步骤3 在“We码应用”下，单击“立即开发”，进入创建We码应用页面。

图 4-4 We 码管理



步骤4 在创建We码应用页面，设置应用相关信息，单击“提交”。

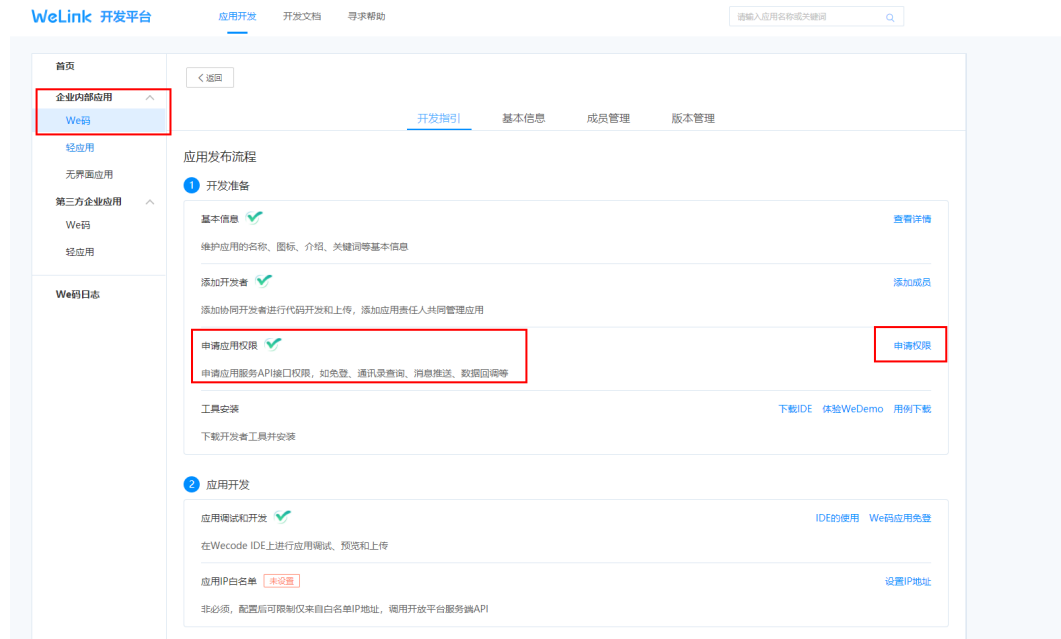
图 4-5 创建 We 码



- 图标：上传新建We码应用的图标，也可直接使用推荐的图标。
- 小程序中文名称：本示例配置为“开发者调查问卷”。
- 小程序英文名称：本示例配置为“MyWelinkApp”。

步骤5 选择“企业内部应用 > We码 > 开发指引”，单击“开发准备”中的“申请权限”。

图 4-6 登录 WeLink 开发平台



步骤6 分别单击“用户登录信息”、“轻应用鉴权”、“租户详细信息”、“组织架构信息”和“用户详细信息”后的“申请权限”，申请相关权限。

图 4-7 申请相关权限

身份与鉴权			
权限名称	权限描述	状态	操作
用户登录信息	获取用户的基本信息，用于登录系统和应用	未开通	申请授权
轻应用鉴权	轻应用鉴权，用于调用JS-API	未开通	申请授权

通讯录			
权限名称	权限描述	状态	操作
租户详细信息	获取租户详细信息，如租户名称、许可、规模等	未开通	申请授权
组织架构信息	按userid或部门id，以部门编码方式获取组织架构	未开通	申请授权
用户简单信息	按部门id获取用户简单信息，如userid、工号、中文名、英文名	未开通	申请授权
用户详细信息	按userid或部门id获取用户更多详细信息，如工号、手机号、邮箱等详细信息	未开通	申请授权
通讯录同步	企业向WeLink同步通讯录信息	未开通	申请授权
用户邮箱信息	获取用户信息，包括邮箱等敏感信息	未开通	申请授权
部门人员信息	根据部门编码查询部门人员信息	未开通	申请授权
用户手机信息	获取用户信息，包含手机号、邮箱等敏感信息	未开通	申请授权
用户权限信息	获取用户权限信息，包括分域管理员所管理的部门范围等	未开通	申请授权

📖 说明

如果要发布的应用调用了WeLink的其他接口，请添加其他相关权限。示例中的应用只开启上述权限。

步骤7 在“基本信息”页签，记录应用的唯一标识“client_id”和应用密钥“client_secret”，用于接口鉴权。

图 4-8 查看应用的唯一标识和密钥



----结束

步骤 2: 在 AstroZero 上创建应用

We码应用创建完成后，需要在AstroZero租户账号中，创建一个应用，导入一个免登录App包，再配置企业WeLink相关系统参数。

步骤1 快速创建一个应用。

1. 使用华为账号，登录AstroZero经典版开发环境。
2. 在AstroZero经典版开发环境首页的“项目”页签下，单击“轻应用”。

图 4-9 创建轻应用



- 在“轻应用”页面的“系统模板库”下，向下滑动滚动条，找到“调查问卷”模板，并单击“使用模板”。

图 4-10 使用应用模板创建应用



- (可选) 定义命名空间。如果未提示需要创建命名空间，则直接跳过本步骤。首次创建应用时，需要定义私有的命名空间，以区别您与其他租户的应用及应用元素，命名空间唯一且不能重复，不能修改。后续您创建的对象、脚本、服务编排、页面等的名称中，系统都会为其增加命名空间的前缀。例如，本应用中定义的命名空间为“CNAME”，后续创建的所有对象、脚本、页面等都将以“CNAME_”开头，请您根据实际情况设置。

注意

命名空间一旦设定不能修改，请慎重定义。建议您使用公司或者团队的缩写作为命名空间。

- 在“创建轻应用”页面输入应用“标签”为“Astro轻应用开发者调查问卷”，单击“创建”。

图 4-11 应用开发工作台



说明

在应用开发页面，如果提示“您的租户没有配置默认邮件服务器地址，将会导致在服务编排/BPM/脚本中发送邮件功能不可用”，请直接忽略，本示例不涉及。



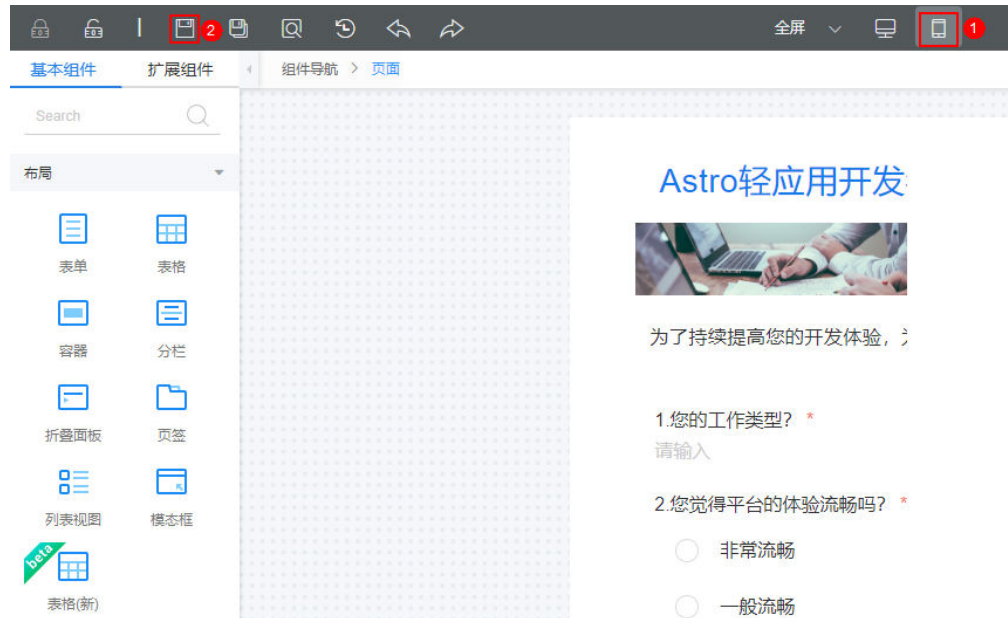

- 在应用开发页面，单击“Page”目录下的“CNAME__Questionnaires_questionnairePage”页面，单击页面右上方的，将页面布局模式切换为手机模式，单击。

图 4-12 修改页面为手机模式



- 同样方式，修改页面“CNAME__Questionnaires_submitPage”的布局模式为手机模式。
- 在左侧导航栏中，单击，即可预览应用。
在调查问卷应用预览页面，输入问卷信息后，单击“提交”。提交完成后，在“调查问卷管理”中可查看到已提交的问卷信息。

步骤2 导入免登录应用包。

📖 说明

免登录应用包仅为样例包，用于调查问卷应用发布到WeLink We码应用场景，实现WeLink用户在使用We码应用时，可通过免登逻辑获取到AstroZero的认证信息。开发者可根据实际业务需求自定义开发免登逻辑，入参出参需要和样例保持一致，确保免登逻辑正常。

- 单击[下载链接](#)，下载免登录应用包。
- 在AstroZero经典版开发环境中，单击“管理”，进入AstroZero经典版开发环境管理中心。
- 在左侧导航中，选择“应用管理 > 软件包管理 > 软件包安装”。
- 单击“新建”，将免登录包拖拽到上传文件夹处，单击“安装”。

图 4-13 导入软件包



说明

应用包上传完成后, 在经典版开发环境首页“项目 > 我的应用”下, 可查看到“WeLink”应用。

步骤3 配置WeLink相关系统参数及匿名用户权限。

1. 在经典版开发环境管理中心的左侧导航栏中, 选择“系统管理 > 系统参数”。
2. 在系统参数列表中, 查找参数“welinkClientId”, 单击查找到的“welinkClientId”, 进入参数详情页面。
3. 单击“值”后的✎, 将“值”修改为步骤7中获取的应用唯一标识“client_id”取值, 单击“保存”。

图 4-14 修改“值”为应用标识



- 按照上述方法，将参数“welinkClientSecret”的“值”修改为步骤7中，获取的应用密钥“client_secret”取值。

说明

“welinkClientId”和“welinkClientSecret”参数只需要在开发第一个We码应用时进行配置，后续开发其他We码应用不需要再次配置。

- 修改WeLink小程序回调参数。

在“内置系统参数”页签，搜索“token_url_in_bluewelink”，修改“值”为“/service/Tong_Welink/1.0.0/welinkLogin”。

图 4-15 修改内置系统参数值用于小程序回调



说明

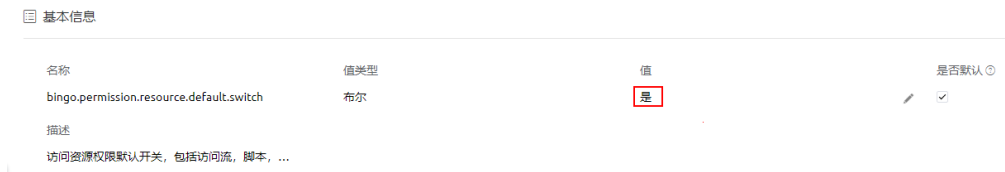
“/service/Tong_Welink/1.0.0/welinkLogin”是免登录应用WeLink中“welinkLogin”接口的URL。设置该“值”，后续将应用发布到WeLink后，AstroZero自动将WeLink用户添加到AstroZero业务用户列表中，并归属到Portal User Profile权限集。

- 开启权限配置接口权限。

在“内置系统参数”页签，搜索“bingo.permission.resource.default.switch”，将参数的值修改为“是”。

图 4-16 开启接口权限

系统参数详情: [bingo.permission.resource.default.switch](#)




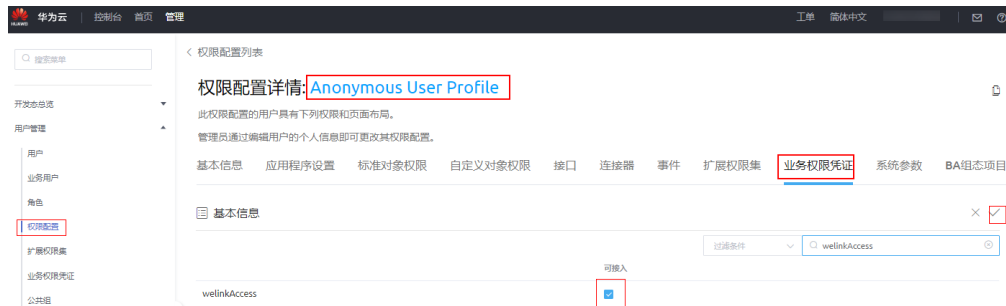
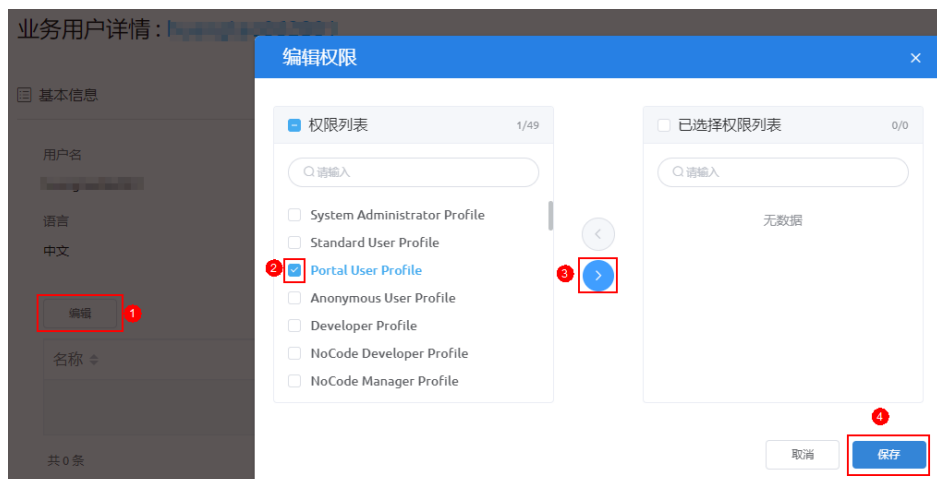
- 配置允许WeLink用户访问小程序权限。
选择“用户管理 > 权限配置”，在“权限配置列表”中，单击“Anonymous User Profile”，在“业务权限凭证”页签，单击 ，勾选“welinkAccess”后的“可接入”。

图 4-17 匿名用户权限配置



- 给WeLink用户配置权限。
选择“用户管理 > 业务用户”，在业务用户列表中，单击WeLink用户名，参考图 4-18在业务用户详情中，单击“编辑”，赋予WeLink用户“Portal User Profile”权限。

图 4-18 配置权限





- 配置应用可见性。
选择“用户管理 > 权限配置”，在“权限配置列表”中，单击“Portal User Profile”。在权限配置详情中，单击“应用程序设置”页签。单击 ，选中应用的“可见性”复选框，单击 ，配置应用可见性。

图 4-19 配置应用可见性



10. 配置允许WeLink用户操作当前应用中的自定义对象权限。


选择“用户管理 > 权限配置”，在“权限配置列表”中，单击“Portal User Profile”。在“自定义对象权限”页签，搜索“Questionnaires_qR_CST”对象，在搜索结果中找到“CNAME__Questionnaires_qR_CST”，单击，先勾选“编辑”，保存后再勾选其他全部权限。

图 4-20 配置自定义对象的操作权限



说明

这里以“调查问卷”模板应用为例配置“Portal User Profile”权限，WeLink用户使用该应用只要配置相关对象权限即可。使用其他应用，可能还需要配置接口、脚本等相关权限，请根据实际情况配置权限。

----结束

步骤 3：编译并发布应用

下面介绍如何将**步骤2：创建应用**中创建的应用编译发布到WeLink上，并设置AstroZero端及WeLink端的应用发布参数。

步骤1 在AstroZero上编译发布应用。




1. 在AstroZero经典版开发环境首页“项目 > 我的应用”下，单击“Astro轻应用开发者调查问卷”，进入应用开发页面。
2. 单击导航栏左下角的，选择“设置”，包类型选择“资产包”，单击“保存”。
3. 单击左下角，选择“编译”，启动应用编译。
4. 编译完成后，单击，选择蓝标“WeLink-We码”，跳转到发布至WeLink页面。

图 4-21 选择 WeLink



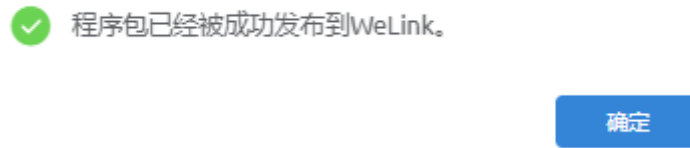
说明

如果当前未登录WeLink，需要先扫码或输入账号、密码登录。登录后，直接跳转到应用创建设置页。

步骤2 设置WeLink应用信息。

1. 在“发布至WeLink”页面，设置待发布应用的以下信息。
 - 小程序名称：从下拉框选择要发布到WeLink上的应用名。本示例选择“开发者调查问卷”。
 - 版本号：要发布的版本号。注意是数字递增的，新发布的版本号应高于之前的版本。本示例设置为“1.0.0”。
 - 主页：选择一个发布的页面作为首页，即显示的默认页面。本示例设置为“命名空间_Questionnaires_questionnairePage”。
 - 环境类型：承载应用程序服务的AstroZero环境类型。本示例选择“开发环境”。
 - 开启调试：开启后，在手机端测试小程序时会展示VConsole，用于调试。建议在测试阶段开启，正式使用时关闭。
2. 设置完成后，单击“发布”。
3. 参数确认无误后，单击“确定”。

图 4-22 确定发布后跳转至 WeLink 应用管理页



步骤3 在WeLink上提交发布应用申请。

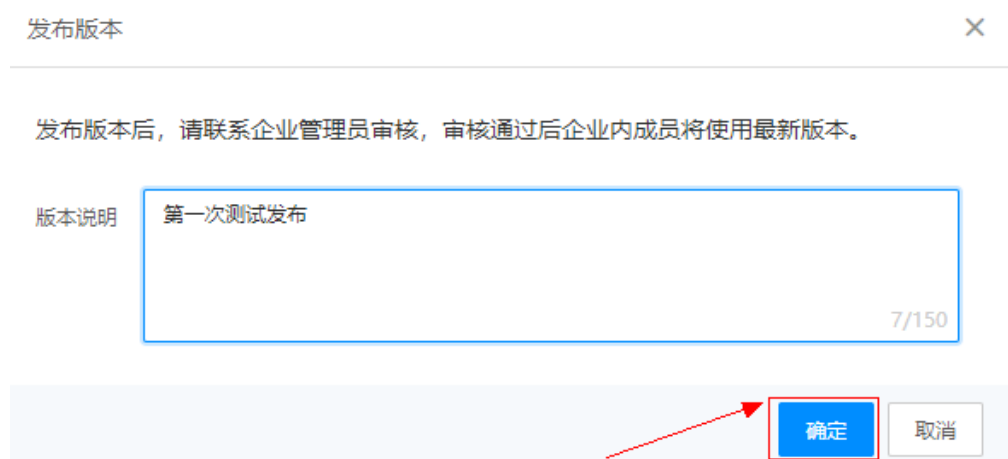
1. 在“版本管理”中，单击“发布版本”。

图 4-23 在 WeLink 上发布版本



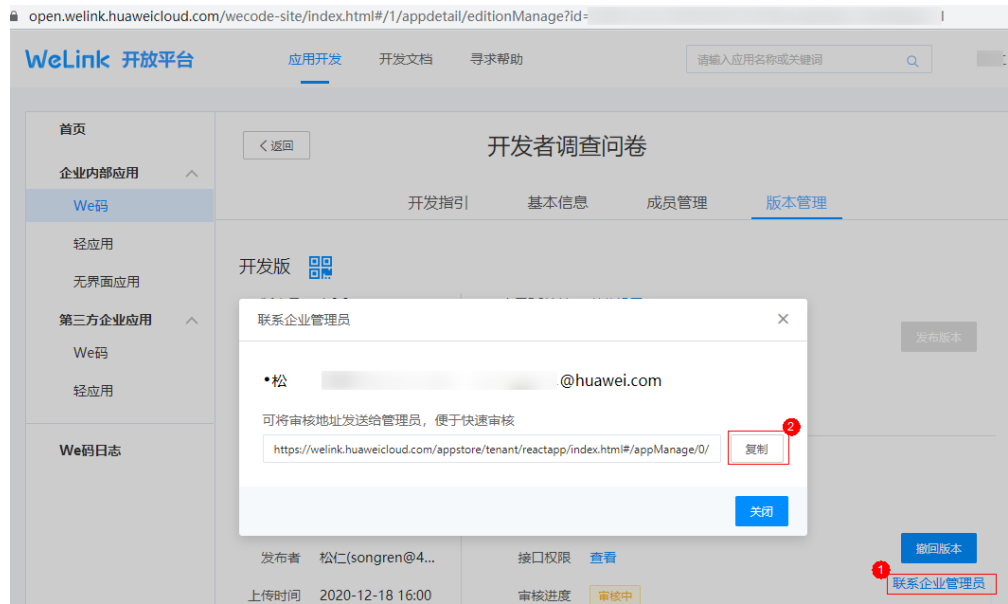
2. 在“版本说明”中输入相关说明信息，如“第一次测试版本”，单击“确定”，提交发布应用申请给企业管理员审核。

图 4-24 输入版本说明



3. 提交审核后，页面将显示“审核版”，单击“联系企业管理员”，在弹窗中单击“复制”，将应用发布审核链接发送给管理员审核发布版本。

图 4-25 联系管理员审核版本



步骤4 联系管理员，审核发布应用版本。

1. 管理员打开应用发布审核链接，登录后，在“审核新版本”中单击“同意”。

📖 说明

如果没有应用发布审核链接，请访问[WeLink管理后台](#)，选择“应用 > 应用管理 > 全部应用”，搜索到小程序，单击应用名即可进入应用审核页面。

2. 在“同意新版本”中，输入审核意见，单击“确定”，完成审核。
审核完成后，应用会发布到企业WeLink的开发环境中。

---结束

步骤 4：在 WeLink 移动端测试应用

应用发布后，可以在WeLink移动端搜索并验证已发布的小程序应用。

步骤1 在WeLink手机端，单击“业务”，搜索小程序“开发者”。

步骤2 单击搜索结果列表中的应用名，进入调查问卷页面。

如果无法正常登录应用，且提示用户名已注册，请按照[常见问题](#)中操作处理。

图 4-26 搜索小程序“开发者调查问卷”



步骤3 输入问卷信息，单击“提交”，成功后跳转到“提交成功”页面。

图 4-27 进入小程序页面

Astro轻应用开发



为了持续提高您的开发体验，

1.您的工作类型? *

请选择

2.您觉得平台的体验流畅吗? *

非常流畅

一般流畅

不太流畅

3.您在开发过程中要困难, 能否获取有效的文档/帮助? *

非常有帮助

一般

没有帮助

步骤4 单击“查看详情”或“继续提交”，测试验证小程序的功能。

图 4-28 提交成功




问卷调查完成后，切回到“开发者调查问卷”的应用开发页面，单击预览应用。在“调查问卷管理”页签中，可查看或删除相关数据。

图 4-29 管理调查问卷记录



恭喜您，到此您已完成WeLink小程序的开发。

----结束

相关主题

如果对导入的免登录应用有兴趣，可以查看以下内容，更深入的了解免登录应用。

- 免登录应用所在位置
将免登录应用包（[下载链接](#)）导入到AstroZero后，在开发环境首页“项目 > 我的应用”下会显示导入的应用包。

📖 说明

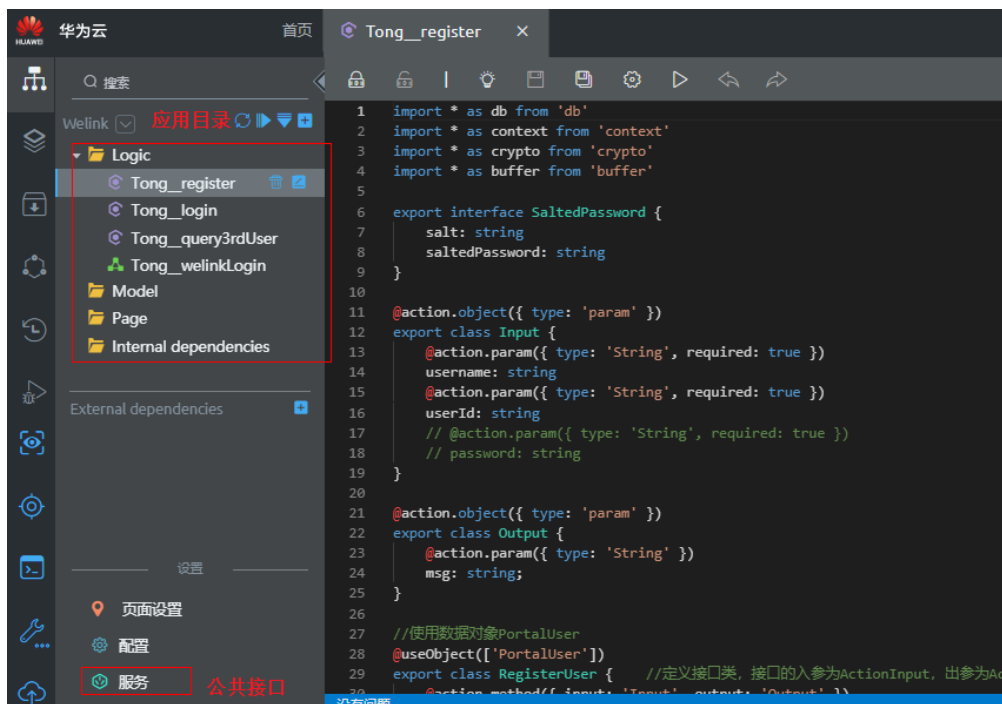
本示例中提供的免登录包仅供学习使用，如果需要商用，建议参考样例包进行优化。

图 4-30 免登录应用位置



- 免登录应用开发工作台介绍

图 4-31 应用目录



- 脚本、服务编排和公共接口等应用资源介绍

表 4-1 脚本及接口等应用资源用途

资源名称	所在位置	作用
Tong__register	应用目录Logic	业务用户注册脚本，用于WeLink新用户接入。 说明 “Tong__”为免登录应用包开发时，开发者使用的命名空间。此命名空间不能修改。
Tong__registerquery3rdUser	应用目录Logic	查询WeLink用户信息脚本，用于WeLink用户接入验证。
Tong__welinkLogin	应用目录Logic	登录服务编排，用于WeLink用户登录业务应用。
welinkAccess	配置 > 业务权限凭证	业务权限凭证，用于管理外部调用免登录接口的权限。
welinkLogin	服务 > 公共接口	公共接口，用于对外暴露服务，并编辑业务权限凭证，归入welinkAccess。
token_url_in_bluewelink	管理 > 系统参数 > 内置系统参数	AstroZero内置系统参数，用于小程序回调。

- AstroZero应用与企业WeLink互通涉及的接口

应用发布到WeLink后，WeLink企业中的用户在We码中使用该应用时，AstroZero需要实现WeLink企业用户到应用业务用户的转换，获取用户身份相关接口说明如表4-2所示。

表 4-2 获取用户身份相关接口

序号	描述	接口	接口类型
1	We码小程序获取免登授权码	HWH5.getAuthCode	前端JS API
2	服务端获取access_token	/auth/v2/tickets	服务端API
3	服务端根据免登授权码获取userId	/auth/v2/userid	服务端API
4	服务端获取用户详细信息	/contact/v1/users	服务端API

常见问题

- 问题1

问题描述：在WeLink移动端无法登录应用，且提示“用户名已注册”。

图 4-32 用户名已注册



可能原因：

可能是因为你拥有多个WeLink账号导致的。

处理方法：

- 在AstroZero经典版开发环境中，单击“管理”，进入AstroZero经典版开发环境管理中心。
- 在左侧导航中，选择“用户管理 > 业务用户”。

- c. 在业务用户列表中，检查同步过来的Welink账号是否均已自动分配了Portal User Profile权限集。
- d. 如果权限集为空，请单击用户名，为此账号添加Portal User Profile权限集，添加完成后，重新打包发布。

图 4-33 多个 welink 账号手动添加 Portal User Profile 权限集



图 4-34 编辑用户权限集



图 4-35 编辑用户权限集



图 4-36 添加 Portal User Profile 权限



• 问题2

问题描述:

WeLink导入的业务用户，在PC端无法正常登录已发布的应用。

可能原因:

在AstroZero中导入业务用户时，未同步WeLink中的业务用户密码。

处理方法:

请参照[扫码登录第三方网站集成指导](#)中操作处理。

后续操作


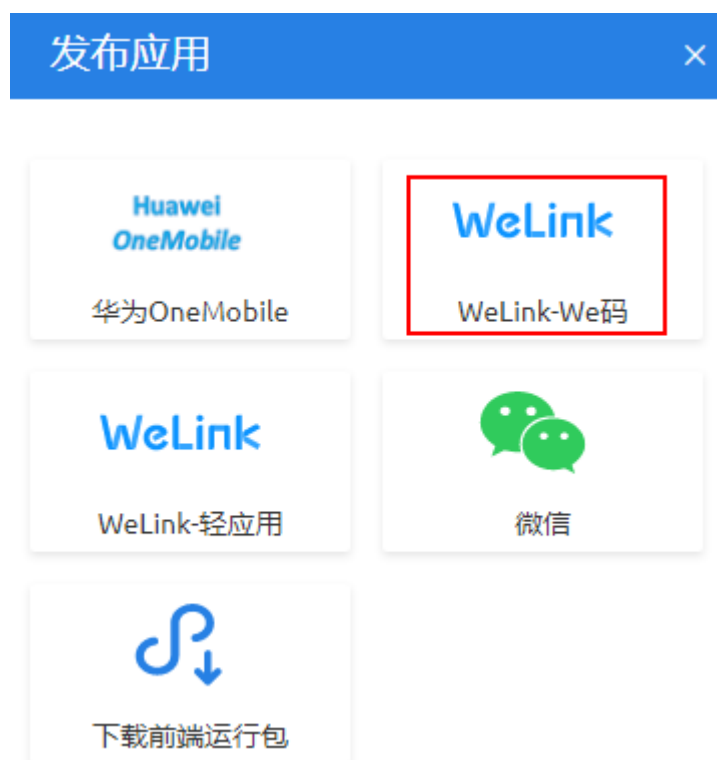
AstroZero支持在沙箱或运行环境中发布WeLink小程序，发布流程和开发环境中类似（[步骤3：编译并发布应用](#)）。即在AstroZero沙箱或运行环境管理平台选择“应用管理 > 应用导航”，在应用程序列表中单击待发布应用后的，选择“WeLink-We码”，进行发布。

图 4-37 AstroZero 沙箱或运行环境中发布 WeLink 小程序



4.2 发布 WeLink 轻应用

业务场景

当前支持通过扫码绑定AstroZero与WeLink后，将AstroZero中开发的应用快速发布到WeLink上，实现企业业务的高效率、低成本创新。

发布WeLink轻应用是将已开发好的应用或全新开发的Web页面发布到WeLink，用户在WeLink APP中即可打开应用。发布WeLink轻应用适用于简单的应用场景。

本示例以创建“调查问卷轻应用”为例，介绍如何把AstroZero上开发的应用发布成WeLink（蓝标）轻应用。

调查问卷模板为企业和个人提供调查问卷应用模板，用于收集资料或管理问题记录。该模板以对AstroZero开发者展开调查问卷为例，可基于该应用模板快速自定义调查问卷内容。

图 4-38 开发者调查问卷



The image shows a survey form for 'Astro轻应用开发'. It includes a header with the title, a sub-header with a photo of people working, and three questions with radio button options.

Astro轻应用开发

为了持续提高您的开发体验，

1.您的工作类型? *

请选择

2.您觉得平台的体验流畅吗? *

- 非常流畅
- 一般流畅
- 不太流畅

3.您在开发过程中要困难，能否获取有效的文档/帮助? *

- 非常有帮助
- 一般
- 没有帮助

使用模板创建调查问卷时，会自动创建如下目录：

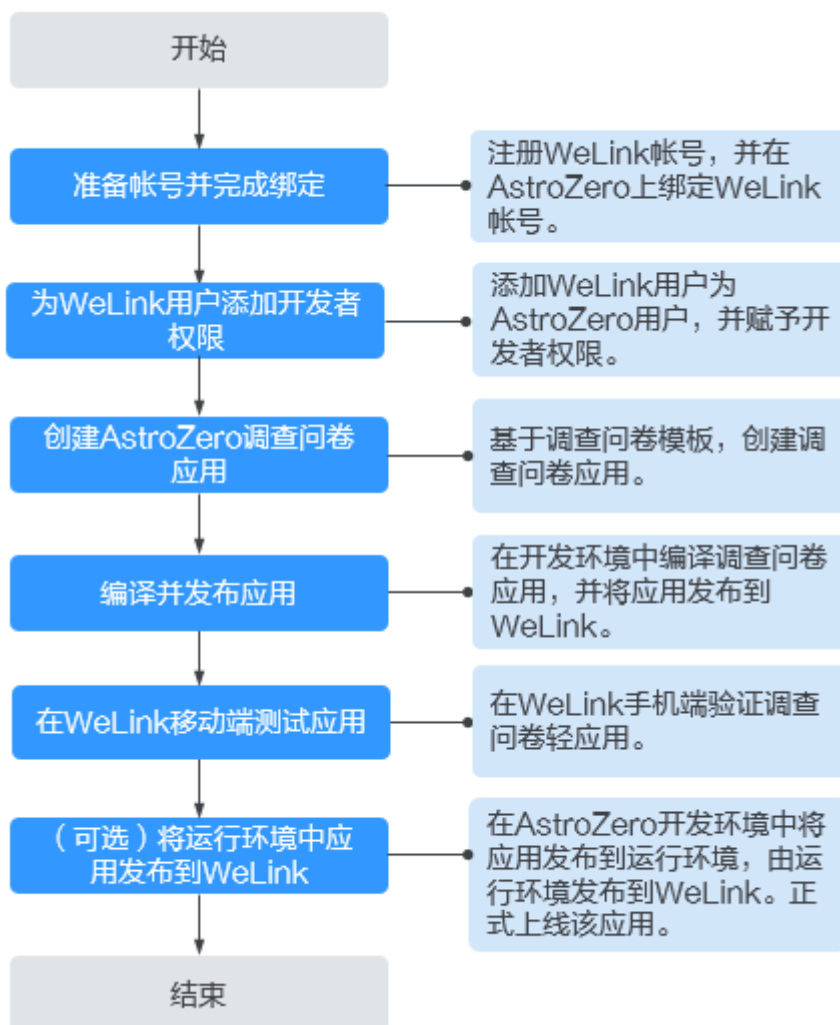
- Logic：用于存放业务中的后台逻辑，例如服务编排、脚本等。本示例不涉及。
- Model：用于存放数据对象。示例应用中会自动创建“命名空间__MyWelinkApp_qR_CST”，用于存放调查问卷数据。
- Page：用于存放前端页面。示例应用中会自动创建如下3个前端页面
 - 命名空间__MyWelinkApp_questionnairePage：问卷提交页。
 - 命名空间__MyWelinkApp_ManagementPage：问卷记录管理页。

- 命名空间_MyWeLinkApp_submitPage: 问卷提交成功页。
- Internal dependencies: 存放内部依赖文件。本示例不涉及。

开发流程

开发WeLink轻应用的流程，如图4-39所示。

图 4-39 开发流程



前提条件

- 已注册华为账号并开通华为云，且已完成实名认证。
- 已开通AstroZero服务，否则请参考[注册租户](#)中操作注册并申请试用。

步骤 1: 准备账号并完成绑定

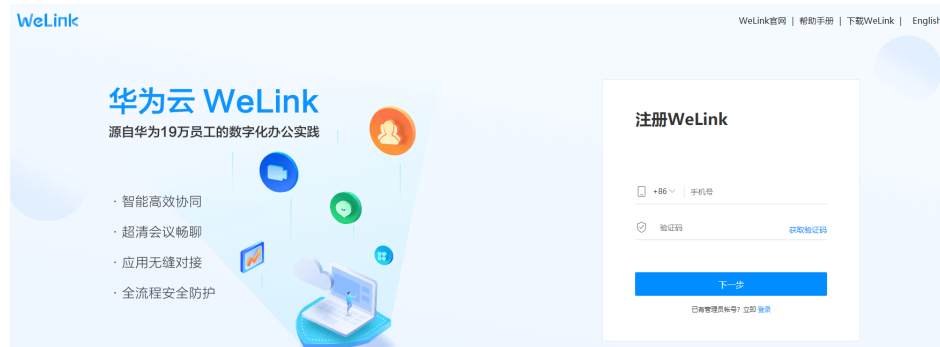
在AstroZero上完成和WeLink账号的绑定操作。

步骤1 (可选) 注册WeLink账号。

📖 说明

- 如果已有WeLink账号，请直接执行[步骤2](#)。
 - 本步骤以在WeLink上注册账号为例进行说明，您也可以在AstroZero经典版开发环境管理中心的“系统管理 > 统一身份认证设置”，单击“WeLink”，在页面右侧单击“注册WeLink”，进行WeLink账号的注册。
1. 登录[WeLink注册页面](#)，输入用户名和验证码，单击“下一步”。

图 4-40 WeLink 登录页面



2. 选择待创建WeLink账号的角色，请根据实际需求设置，本示例选择“我是管理员”。

图 4-41 选择待创建 WeLink 账号的角色

< 请选择

欢迎使用WeLink

我是成员

如果您的企业已开通WeLink账号或有同事已经使用，请选此项

我是管理员

如您想为企业或组织创建WeLink账号，请选此项


📖 说明


如果注册的手机号已加入企业或者组织，则不会显示[图4-41](#)页面。


3. 在“创建企业/组织”页面，设置如下信息后，单击“创建”。



图 4-42 创建企业/组织

< 创建企业/组织

 企业/组织名

 联系人姓名

 联系人邮箱

 行业 

创建

点击创建，即代表您已阅读并同意 [《隐私政策声明》](#) 与 [《用户服务协议》](#)

- 企业/组织名：新建企业或组织名称。
- 联系人姓名：新建企业或组织的联系人。
- 联系人邮箱：新建企业或组织的联系人的邮箱地址。
- 行业：新建企业或组织所属的行业。

WeLink账号注册成功后，您可以登录企业管理后台，也可以下载PC客户端。

图 4-43 登录企业管理后台/下载 PC 客户端



步骤2 以账号登录AstroZero经典版开发环境，在“管理”下，绑定WeLink并同步WeLink信息。

说明

此处的账号是指购买AstroZero服务的华为账号。更多AstroZero中用户的介绍，请参见[AstroZero中的用户](#)。

1. 参考[如何登录经典版环境配置](#)中操作，登录经典版环境配置。
2. 在左侧导航栏中，选择“系统管理 > 统一身份认证设置”，单击“WeLink”。

图 4-44 进入绑定 WeLink 页面



3. 在弹出页面，单击“绑定WeLink”。
4. 在弹出的“扫码认证”框中，打开移动端WeLink，在右上角单击“+”，扫描图 4-45 中所示二维码。

图 4-45 扫描二维码



5. 勾选待同步用户所在的WeLink部门，单击“同步”。

图 4-46 同步用户信息



说明

- 绑定WeLink会同步WeLink用户为AstroZero业务用户，用户数不计入License用户数限额。只有业务用户登录当前AstroZero环境时，才会计入License用户数限额。
- AstroZero绑定WeLink时，扫码绑定的WeLink用户会自动添加为AstroZero开发者用户和业务用户。
- 同步WeLink部门时，会自动同步WeLink部门主管为AstroZero业务用户。
- 同步的WeLink用户包括子部门的用户。

扫码认证后，自动完成AstroZero和WeLink绑定，并同步WeLink所有部门信息到AstroZero，同步后的WeLink部门信息显示在AstroZero管理中心的“用户管理 > 角色”中。

勾选要同步用户所在的WeLink部门，系统还会同步所选部门的WeLink用户，将WeLink用户同步到AstroZero当前环境的业务用户列表中，并赋予“Portal User Profile”权限。业务用户是访问在AstroZero中开发的业务应用（例如调查问卷应用）的用户账号。在AstroZero管理中心的“用户管理 > 业务用户”中，可查看同步来的WeLink用户信息。

思考：如果在图4-46中未同步用户信息，即单击“跳过”，完成WeLink账号绑定后，后续该如何同步用户信息呢？

在AstroZero管理中心的“系统管理 > 统一身份认证设置”，单击“WeLink”，在“同步WeLink用户”区域下单击“立即同步”，即可完成用户信息的同步。

----结束

步骤 2：添加 WeLink 用户为 AstroZero 开发者

如果需要WeLink用户在AstroZero开发环境中具备开发者权限，请添加WeLink用户为AstroZero开发者用户，并赋予开发者权限。

步骤1 以华为账号登录AstroZero经典版开发环境，在首页单击“管理”，进入AstroZero经典版开发环境管理中心。

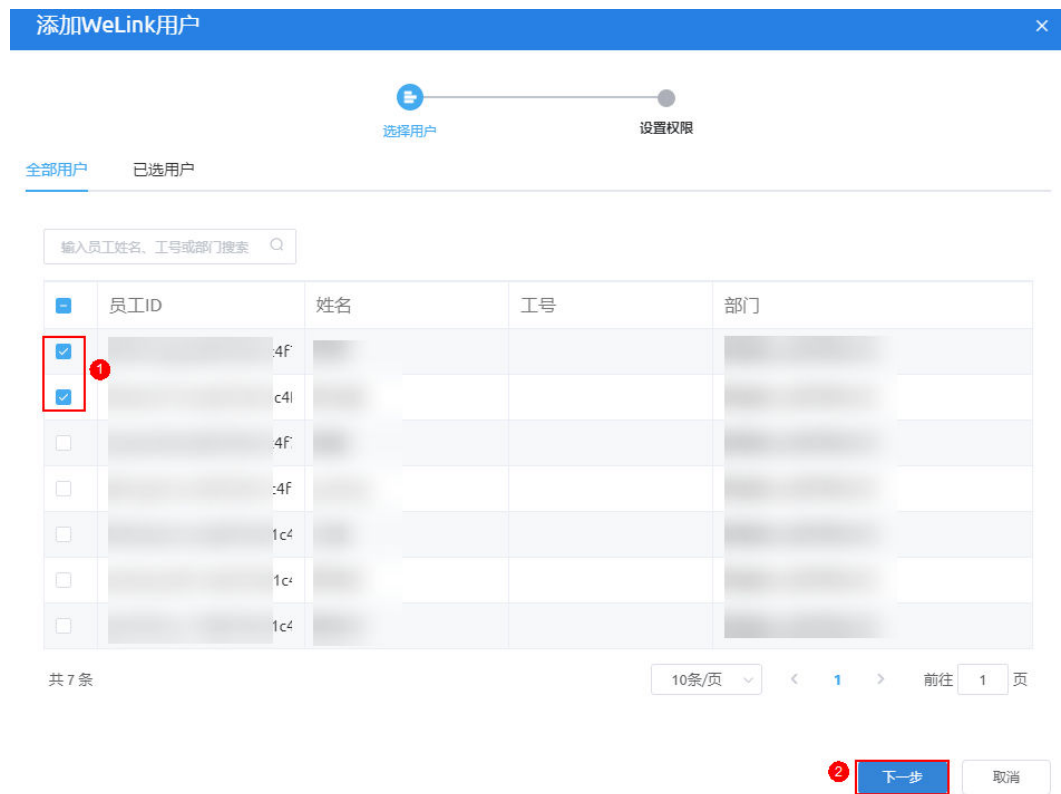
步骤2 在左侧导航栏中，选择“用户管理 > 用户”，单击“添加WeLink用户”。

图 4-47 添加 WeLink 用户



步骤3 勾选需要添加的WeLink用户，单击“下一步”。

图 4-48 勾选 WeLink 用户



步骤4 权限配置为“Developer Profile”，单击“保存”。

添加完成后，在用户列表中可查看到已添加的WeLink用户。

WeLink用户添加了开发者权限后，即可在AstroZero产品页使用WeLink扫码登录AstroZero开发环境，具体操作如下：


1. 登录[AstroZero产品页](#)，单击“WeLink扫码登录”。

图 4-49 WeLink 扫码登录入口



图 4-50 登录页



2. 打开移动端WeLink，在右上角单击“+”，扫描二维码进行登录，也可单击图 4-50 上的 ，使用WeLink用户的账号密码登录。

----结束

步骤 3: 创建 AstroZero 调查问卷应用

在AstroZero经典版开发环境中创建“调查问卷应用”，设置应用在移动端显示效果，并为WeLink用户设置业务访问权限。

步骤1 基于调查问卷模板，创建调查问卷应用。

1. 参考[如何进入经典开发环境](#)中操作，使用具备开发者权限的AstroZero账号，登录AstroZero经典版开发环境。

📖 说明

- 具备开发者权限的AstroZero账号：AstroZero租户及具备开发者权限的WeLink用户，都具有进入开发环境开发应用的权限。
 - 租户请使用华为账号登录AstroZero，具备开发者权限的WeLink用户则需要访问 [AstroZero产品页](#) 扫码登录。
2. 在AstroZero经典版开发环境首页的“项目”页签下，单击“轻应用”。

图 4-51 创建轻应用



3. 在“轻应用”页面，找到“调查问卷”模板，并单击“使用模板”。

图 4-52 使用应用模板创建应用



4. （可选）定义命名空间。如果未提示需要创建命名空间，则直接跳过本步骤。
- 首次创建应用时，需要定义私有的命名空间，以区别您与其他租户的应用及应用元素，命名空间唯一且不能重复，不能修改。后续您创建的对象、脚本、服务编排、页面等的名称中，系统都会为其增加命名空间的前缀。
- 例如，本应用中定义的命名空间为“CNAME”，后续创建的所有对象、脚本、页面等都将以“CNAME_”开头，请您根据实际情况设置。

注意

命名空间一旦设定不能修改，请慎重定义。建议您使用公司或者团队的缩写作为命名空间。

5. 在“创建轻应用”页面，输入应用“标签”为“Astro轻应用开发者调查问卷”，单击“创建”。

说明

在应用开发页面，如果提示“您的租户没有配置默认邮件服务器地址，将会导致在服务编排/BPM/脚本中发送邮件功能不可用”，请直接忽略，本示例不涉及。



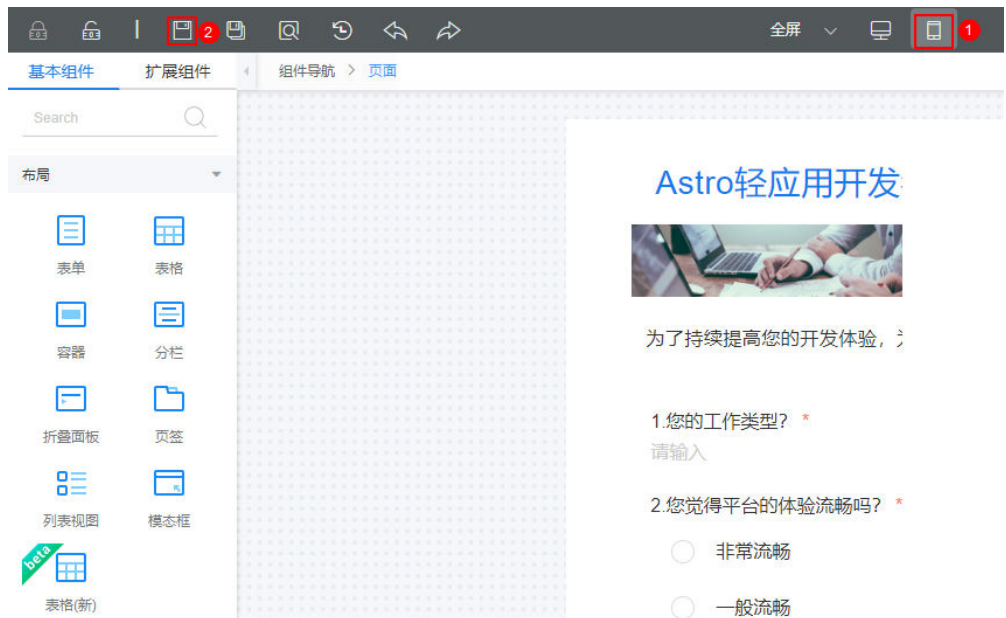

6. 修改要发布的页面模式，单击“Page”目录下的“命名空间__Questionnaires_questionnairePage”页面，单击页面，切换页面布局模式为手机模式，单击。

图 4-53 修改页面为手机模式



7. 按照同样的方式，修改页面“命名空间__Questionnaires_submitPage”的布局模式为手机模式。
8. 在右侧导航栏中单击，可预览调查问卷应用。

步骤2 设置调查问卷应用在移动端显示效果。

1. 单击左侧菜单栏下方的“配置”，在“移动应用设置”页签，设置如下参数。

图 4-54 移动应用设置




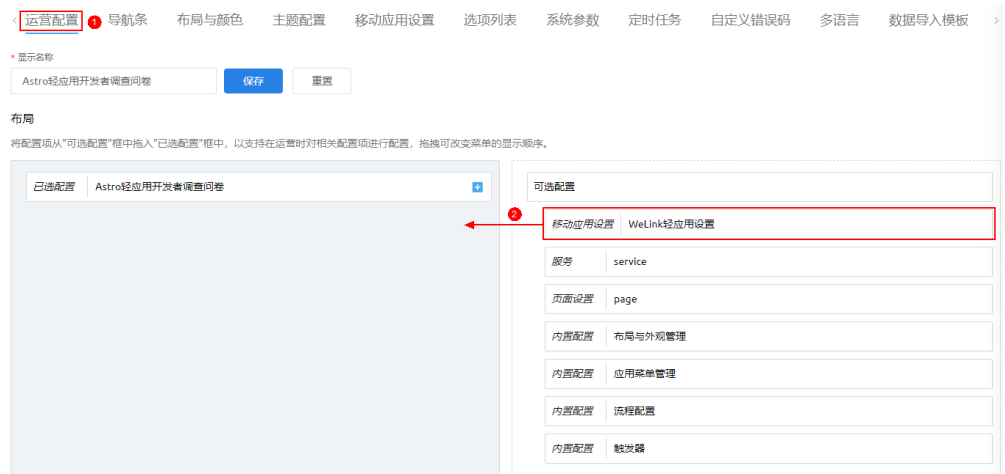
- 显示名称：调查问卷应用在移动端显示的名称，本示例配置为“WeLink轻应用设置”。设置后，单击“保存”。
 - 移动端首页：选择“调查问卷”页面作为WeLink H5轻应用的主页。单击，使用WeLink扫一扫可体验在WeLink上打开该页面的效果。
 - PC端首页：可选配置，选择页面作为WeLink开放平台上轻应用的PC端首页。
 - 应用管理后台设置：可选配置，作为应用后台免登地址。
2. (可选) 在“运营配置”页签中，将配置项添加到左侧“已选配置”中。
执行该操作的目的是：调查问卷发布到运行环境后，可在经典版运行环境管理中心“应用管理 > APP配置”中，修改移动应用设置。

图 4-55 添加运营配置



步骤3 配置允许WeLink用户使用业务应用权限。





1. 在AstroZero经典版开发环境上方选择“管理”，进入经典版开发环境管理中心。
2. 在左侧导航栏中，选择“用户管理 > 权限配置”。
3. 在“权限配置列表”中，单击“Portal User Profile”。
4. 在权限配置详情中，单击“应用程序设置”页签，单击，选中该应用的“可见性”复选框，单击。配置应用可见性。

图 4-56 配置应用可见性



5. 在权限配置详情中，单击“自定义对象权限”页签，进入编辑页面。搜索“Questionnaires_qR_CST”对象，在搜索结果中找到“命名空间__Questionnaires_qR_CST”，单击，先勾选“修改全部”，再勾选“创建”权限，单击保存配置。
执行才操作的目的是，允许WeLink用户操作当前应用中的自定义对象。

说明

实际的对象名，可在“Astro轻应用开发者调查问卷”应用开发页面的“Model”目录查看。

图 4-57 配置自定义对象的操作权限



----结束

步骤 4：编译发布应用

本节介绍如何将将在AstroZero上创建的应用发布到WeLink上。

步骤1 在AstroZero上编译发布应用，将创建的应用编译发布到WeLink上。

1. 在AstroZero经典版开发环境首页的“项目 > 我的应用”下，单击“Astro轻应用开发者调查问卷”，进入应用开发页面。
2. 在左侧导航栏上，单击左下角，选择“设置”，包类型选择“资产包”，单击“保存”。
3. 在左侧导航栏上，单击左下角，选择“编译”，启动应用编译。
4. 编译完成后，单击，选择“WeLink-轻应用”。

图 4-58 选择 WeLink-轻应用



如果是WeLink用户的开发者，还可以直接在“We”下，单击“WeLink”图标一键发布。

图 4-59 一键发布至 WeLink



5. 在发布轻应用页面，单击“创建轻应用”。

📖 说明

和AstroZero绑定的WeLink上没有创建过轻应用，才会显示“打开WeLink开放平台”链接。本示例以通过AstroZero入口创建WeLink上轻应用为例。

图 4-60 创建轻应用



6. 输入待创建轻应用的名称，单击“完成”。本示例名称设置为“开发者调查问卷轻应用”。

图 4-61 输入轻应用名称



执行上述操作后，在WeLink上会创建名称为“开发者调查问卷轻应用”的轻应用。

7. 应用创建完成后，系统会自动在“轻应用名称”中填入刚创建的应用，单击“发布”。

图 4-62 发布轻应用



8. 单击“确定”，在WeLink扫描登录框中，打开WeLink扫描或使用账号密码登录WeLink开放平台。

步骤2 在WeLink上提交发布应用申请。

1. 在“版本管理”中，单击“发布版本”。

图 4-63 在 WeLink 上发布版本



2. 选择审核员，在“版本说明”中输入说明信息，如“第一次测试发布”，单击“确定”，提交发布应用申请给审核员审核。

图 4-64 输入版本说明



- 提交审核后，页面将显示“审核版”，单击“联系企业管理员”，在弹窗中单击“复制”，可将应用发布审核链接发送给审核员审核发布版本，单击“关闭”。

图 4-65 联系管理员审核版本



步骤3 联系审核员，审核发布应用版本。

- 审核员打开应用发布审核链接，登录后，单击“审核新版本”。如果是应用测试阶段，可设置部分测试人员可见，再单击“同意”。如果需要直接使用开发环境发布到WeLink中的轻应用作为业务应用正式上线，可设置全部人员可见。

图 4-66 同意发布



2. 在“同意新版本”中，输入审核意见，单击“确定”，完成审核。
审核完成后，应用会发布到企业WeLink中。

----结束

步骤 5: 在 WeLink 移动端测试应用

应用发布后，可以在WeLink手机端搜索并验证已发布的轻应用，也可以在企业WeLink管理员界面直接扫码进入轻应用。

步骤1 在WeLink手机端，单击“业务”，搜索应用“开发者调查问卷轻应用”。

步骤2 搜索到后，单击应用名，进入调查问卷页面。

图 4-67 搜索轻应用“开发者调查问卷轻应用”



步骤3 输入问卷信息，单击“提交”，成功后跳转到“提交成功”页面。

图 4-68 进入轻应用页面



Astro轻应用开发

为了持续提高您的开发体验，

1. 您的工作类型? *

请选择

2. 您觉得平台的体验流畅吗? *

非常流畅

一般流畅

不太流畅

3. 您在开发过程中要困难，能否获取有效的文档/帮助? *

非常有帮助

一般

没有帮助

图 4-69 提交成功




问卷调查完成后，切回到“开发者调查问卷”的应用开发页面，单击预览应用。在“调查问卷管理”页签中，可查看或删除相关数据。

图 4-70 管理调查问卷记录



恭喜您，到此您已完成WeLink轻应用的开发。

AstroZero提供了很多模板应用供您直接使用，方便您快速将应用发布到WeLink中，省去开发环节。WeLink用户可直接在WeLink移动端使用该应用。如果租户购买的AstroZero规格为专业版或专享版，拥有运行环境，您可继续执行**步骤6：（可选）将运行环境中应用发布到WeLink**，使用运行环境将应用发布到WeLink，WeLink用户在WeLink移动端使用该应用时，调用的后台接口服务都承载在运行环境中。

----结束

步骤 6：（可选）将运行环境中应用发布到 WeLink

在AstroZero开发环境中将应用发布到运行环境，由运行环境发布到WeLink。开发环境、运行环境的介绍，请参见[开发环境介绍](#)。

说明

免费版不提供运行环境，请参考[购买商用租户](#)中操作购买专业版或专享版后，再进行打包发布操作，并在运行环境安装应用。

步骤1 将应用发布到运行环境。





1. 在AstroZero经典版开发环境首页“项目 > 我的应用”下，单击“Astro轻应用开发者调查问卷”，进入应用开发页面。
2. 在左侧导航栏上，单击左下角，选择“设置”，在“编译设置”页面，选择“资产包”，单击“保存”。
3. 在左侧导航栏上，单击左下角，选择“编译”，启动应用编译。
4. 编译完成后，单击，选择“我的仓库”。
5. 设置版本号和描述信息，单击“发布”。

图 4-71 发布到我的仓库



发布成功后，页面显示“程序包已经被成功上传到我的仓库”。

思考：应用发布后，在哪里可以查看已发布的应用呢？

- 方式一：在经典版应用开发页面，单击左侧的。在“已发布”下，可看到已发布的应用。
- 方式二：在经典版开发环境首页，单击“管理”，在“应用管理 > 我的仓库”中，可查看到已发布的应用。

步骤2 在AstroZero经典版运行环境部署应用。

1. 在AstroZero经典版开发环境右上角，单击用户名，选择“运行环境”，进入经典版运行环境。

图 4-72 切换到运行环境



2. 在管理看板页面，选择“我的仓库”。

图 4-73 我的仓库



3. 在我的仓库中，单击调查问卷应用中的“安装”，单击“确定”。

步骤3 在WeLink上创建正式的轻应用，并参考**步骤2**在AstroZero经典版运行环境管理中心的“系统管理 > 统一身份认证设置”，单击“WeLink”，再单击“绑定WeLink”。

步骤4 配置同步到AstroZero的WeLink用户允许使用该业务应用权限。




1. 在AstroZero**经典版运行环境**左侧导航栏中，选择“用户管理 > 权限配置”。
2. 在“权限配置列表”中，单击“Portal User Profile”。
3. 在权限配置详情中，单击“应用程序设置”页签，单击，选中该应用的“可见性”复选框，单击。配置应用可见性。

图 4-74 配置应用可见性



4. 在权限配置详情中，单击“自定义对象权限”页签，进入编辑页面。搜索“Questionnaires_qR_CST”对象，在搜索结果中找到“命名空间__Questionnaires_qR_CST”，单击，先勾选“修改全部”，再勾选“创建”

权限，单击 ✓，保存配置。执行此操作的目录是，允许WeLink用户操作当前应用中的自定义对象。

图 4-75 配置自定义对象的操作权限



说明

实际的对象名，可在“Astro轻应用开发者调查问卷”应用开发页面的“Model”目录下查看。

步骤5 在经典版运行环境将应用发布到WeLink。


1. 在经典版运行环境管理中心选择“应用管理 > 应用导航”，在应用程序列表中单击待发布应用“命名空间__Questionnaires”后的 ，选择“WeLink-轻应用”，发布应用到WeLink上创建的正式应用中。

图 4-76 选择 WeLink-轻应用



2. 参考步骤2~步骤3，在WeLink上发布版本并审核，审核时设置使用该应用的所有WeLink用户可见。

应用发布后，使用该应用的所有WeLink用户可以在WeLink手机端搜索并使用已发布的轻应用。第三方网站也可通过WeLink移动端扫码登录应用，详情请参见[扫码登录第三方网站集成指导](#)。

----结束

常见问题

• 问题1

问题描述：

将AstroZero和WeLink账号A绑定后，如果需要绑定另外的WeLink账号B，怎么处理？

处理方法：

需要先将AstroZero和WeLink账号A解绑，再绑定其他WeLink账号。

- a. 在AstroZero当前环境管理中心选择“系统管理 > 统一身份认证设置”，单击“WeLink”，再单击“解绑”。

图 4-77 解绑 WeLink



- b. 在弹出的提示框，根据情况勾选“删除部门和用户信息”，表示是否删除已同步的部门和用户信息，单击“确定”。
 - c. 选择“系统管理 > 统一身份认证设置”，单击“WeLink”，再单击“绑定 WeLink”，绑定其他WeLink账号。
- **问题2**
问题描述：

在AstroZero中将应用发布到WeLink时，显示如图4-78所示错误提示，怎么处理？

图 4-78 报错



可能原因：

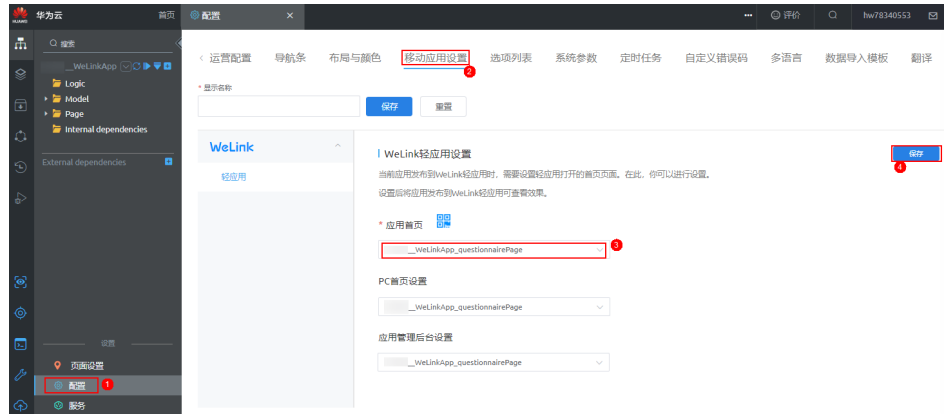
在AstroZero中选择了相同的页面作为WeLink中不同轻应用的主页。

处理方法:

在AstroZero发布应用前，更改主页设置。或者选择和之前一样的WeLink轻应用发布。请根据实际场景选择处理方式。

- 方式1: 在经典版应用开发工作台左侧菜单栏下方，选择“配置”，右侧选择“移动应用设置”页签，更改“应用首页”设置。

图 4-79 更改应用首页




- 方式2: 编译应用完成后，单击，选择“WeLink-轻应用”时，选择和之前相同的WeLink轻应用发布。

图 4-80 发布 WeLink 轻应用



- **问题3**

问题描述:

WeLink导入的业务用户，在PC端无法正常登录已发布的应用。

可能原因:

在AstroZero中导入业务用户时，未同步WeLink中的业务用户密码。

处理方法:

请参考[扫码登录第三方网站集成指导](#)中操作处理。

4.3 发布微信小程序

AstroZero允许将标准页面或高级页面发布到微信小程序。本节将介绍在AstroZero上发布应用到微信小程序的基本操作，包括微信公众平台小程序注册、小程序登录鉴权、发布配置、并验证小程序是否发布成功。

步骤1 准备微信公众平台账号，即在微信公众平台注册小程序。

1. 访问[微信公众平台](#)，单击右上角“立即注册”。

图 4-81 立即注册



2. 选择注册的账号类型为“小程序”。
3. 请填写邮箱（该邮箱未被微信公众平台注册，未被微信开放平台注册，未被个人微信号绑定），设置登录密码，填写验证码，并同意相关协议，单击“注册”。填写的邮箱后续会作为管理员账号进行使用。
4. 登录邮箱，查收激活邮件，单击激活链接。
5. 单击激活链接后，按照界面提示继续下一步的注册流程。
请选择主体类型，由于该发布功能面向的微信公众账号主体类型只能属于企业、政府、媒体或者其他组织，“主体类型”要选择非个人类型，并完善主体信息和管理员信息。
 - 企业类型账号可选择两种主体验证方式：
 - 方式一：需要用公司的对公账户向腾讯公司打款来验证主体身份。打款信息在提交主体信息后可以查看到。
 - 方式二：通过微信认证验证主体身份，需支付300元认证费。认证通过前，小程序部分功能暂无法使用。
 - 政府、媒体、其他组织类型账号，必须通过微信认证验证主体身份。认证通过前，小程序部分功能暂无法使用。
6. 单击“确定”，完成注册流程。
 - 选择对公打款的用户，请根据页面提示，向指定的收款账号汇入指定金额。**请在10天内完成汇款，否则将注册失败。**
 - 选择通过微信认证验证主体身份的用户，完成注册流程后请尽快进行微信认证，认证完成之前部分功能暂不可使用。

步骤2（可选）使用管理员账号（即注册过程中填写的邮箱）登录[微信公众平台](#)，选择“管理 > 成员管理”，在“项目成员”和“体验成员”区域可新增开发者和体验者。

步骤3 使用管理员账号登录[微信公众平台](#)，获取如下关键信息并配置业务域名和request合法域名。

图 4-82 获取关键信息

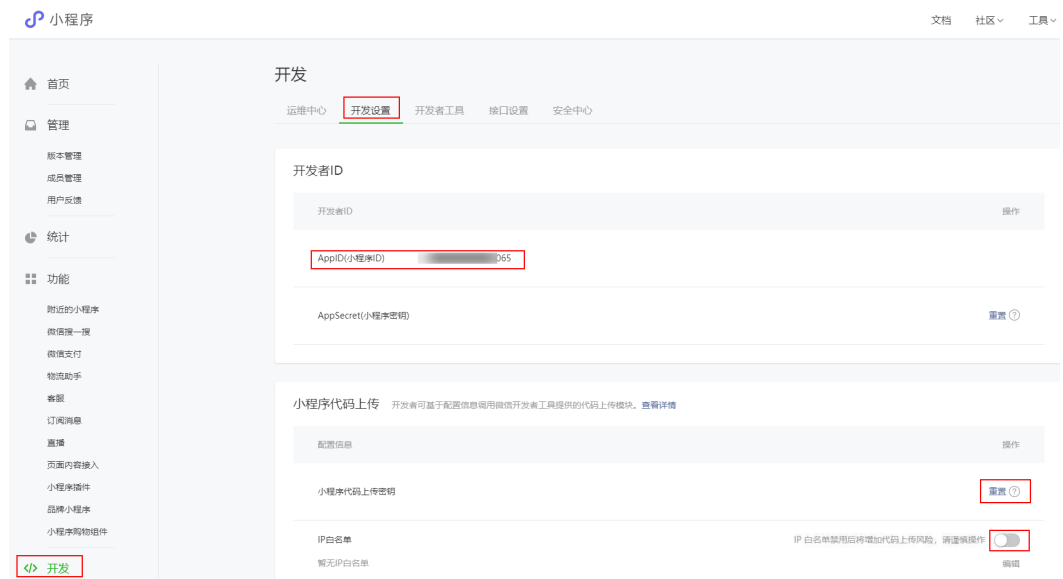


图 4-83 配置业务域名 1



图 4-84 配置业务域名 2



图 4-85 配置 “request 合法域名”

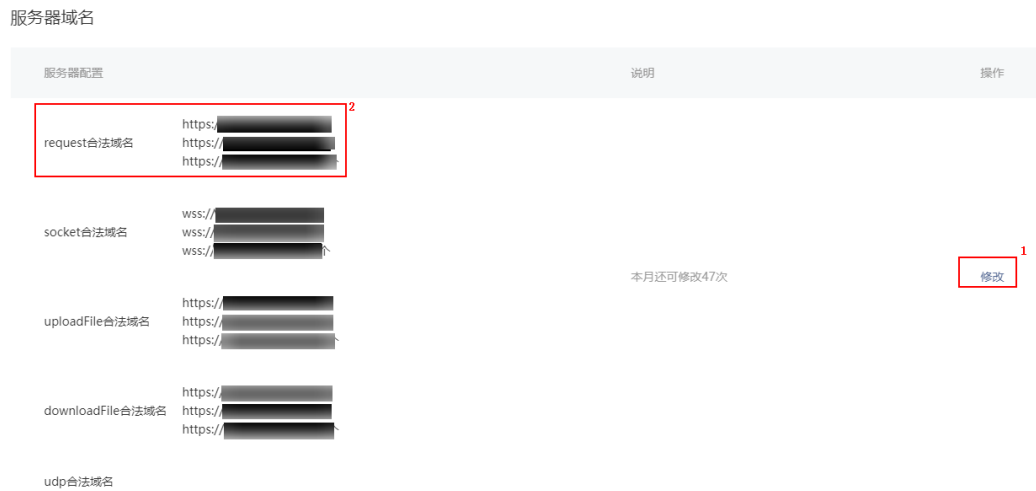


表 4-3 关键信息说明

参数	说明
AppID	小程序的AppID，小程序的注册类型为非个人的微信小程序。 获取方法：使用管理员账号登录 微信公众平台 ，在微信公众平台“开发 > 开发设置”中，可查看“AppID（小程序ID）”的值。
小程序代码上传密钥	用于上传小程序代码到微信中的密钥。 获取方法：使用管理员账号登录 微信公众平台 ，在微信公众平台“开发 > 开发设置”中，单击“小程序代码上传密钥”后的“重置”，进行获取。
IP白名单	在AstroZero开发环境发布微信小程序前，需要在微信公众平台“开发 > 开发设置”页面的“小程序代码上传”区域，关闭“IP白名单”。
业务域名	使用管理员账号登录 微信公众平台 ，在微信公众平台“开发 > 开发设置”页面的“业务域名”区域，单击“修改”。在弹出的“配置业务域名”页面，将承载服务的AstroZero环境访问地址（如华北-北京四AstroZero开发环境域名“appcube.cn-north-4.huaweicloud.com”、运行环境域名为“appcuberun.cn-north-4.huaweicloud.com”），在页面中调用的请求URL（即前端页面中调用的公共接口URL，包括iframe中的请求URL）都加入业务域名中。 须知 您需要在“配置业务域名”页面，单击“下载校验文件”，提交工单联系运维人员将校验文件放置在域名根目录下。

参数	说明
request合法域名	<p>将AstroZero运行环境的域名，配置到微信公众平台的合法域名中。</p> <p>配置方法：在微信公众平台“开发 > 开发设置”页面的“服务器域名”区域，单击“修改”。在“request合法域名”中，添加AstroZero运行环境的域名。</p> <p>AstroZero运行环境是指正式发布上线的真实环境。测试完成后，将应用发布至生产环境，即运行环境，供业务用户使用。华北-北京四AstroZero运行环境域名为“appcuberun.cn-north-4.huaweicloud.com”。免费版AstroZero不提供运行环境，您需要参考购买商用租户中操作，购买专业版或专享版AstroZero，才会拥有AstroZero运行环境。</p>

步骤4 使用账号登录AstroZero经典版开发环境，开发免登BO，用于用户通过微信小程序使用发布的应用时，无需用户输入用户名密码，即可获取当前用户身份。

单击[这里](#)，可获取免登录开发样例。此样例仅作为参考，实际使用时请根据具体场景进行修改，但必须确保入参和出参是一致的。同时需要将服务的请求地址配置到系统参数“token_url_in_wechat”中，且入参、出参设置如下：

- 入参为：
 - code: wx.login后获取的code。
 - appid: 当前小程序id。
- 出参为AstroZero用户登录后，生成的token。

注意



同个租户创建不同的应用，发布到微信小程序前，开发免登BO时，为了防止不同小程序的业务用户出现混淆（即为了防止小程序A的业务用户也可登录小程序B），需要对业务逻辑做处理。建议业务用户数据保存到PortalUser表中时，可使用字段“id”代替“usrName”，或者“usrName”前加上应用“id”加以区分。

步骤5 在AstroZero经典版开发环境，创建要发布的小程序应用，在应用内根据需求创建对象、页面、服务编排和脚本等组件。

步骤6 登录AstroZero发布小程序。

支持在开发环境、沙箱环境以及运行环境中发布小程序，小程序中调用的后台接口服务所承载的环境，取决于在哪个环境中发布小程序。

- 在经典版开发环境中，发布小程序步骤如下：
 - a. 在经典版应用开发页面右下角，单击，选择“编译”。
 - b. 编译成功后，页面会显示“编译XXX成功”，关闭编译成功的窗口。
 - c. 在应用开发页面左侧，单击，选择“微信”。在“发布微信小程序”页面，配置如下参数。
 - 小程序ID: 小程序的AppID，参数获取方法请参见[表4-3](#)。

- 版本号：发布的版本号。每次发布时需要使用新的版本号，使用之前请确认版本号是否已被使用过。
- 主页：选择一个页面作为首页，即显示的默认页面。
- 小程序代码上传密钥：用于上传小程序代码到微信中的密钥，参数获取方法请参见[表4-3](#)。
- d. （可选）单击“预览”，可生成小程序预览页面的二维码，开发者使用微信扫一扫功能，可预览小程序页面。
- e. 设置完成后，单击“发布”。
- 在沙箱环境及经典版运行环境中，发布小程序步骤如下（需要提前在沙箱环境或经典版运行环境中，安装业务应用，具体操作可参考[如何安装应用](#)）：
 - a. 在经典版运行环境管理中心，选择“应用管理 > 应用导航”。
 - b. 在应用程序列表中，单击待发布应用后的，再选择.
 - c. 在“发布微信小程序”页面，配置如下参数。
 - 小程序ID：小程序的AppID，参数获取方法请参见[表4-3](#)。
 - 版本号：发布的版本号。每次发布时需要使用新的版本号，使用之前请确认版本号是否已被使用过。
 - 主页：选择一个页面作为首页，即显示的默认页面。
 - 小程序代码上传密钥：用于上传小程序代码到微信中的密钥，参数获取方法请参见[表4-3](#)。
 - d. （可选）单击“预览”，可生成小程序预览页面的二维码，开发者使用微信扫一扫功能，可预览小程序页面。
 - e. 设置完成后，单击“发布”。

步骤7 使用小程序开发者账号登录[微信公众平台](#)，选择“管理 > 版本管理”，在上传的小程序后单击“提交审核”。

您也可以在“版本管理”页面，单击小程序后的下拉图标，将小程序设置为体验版，生成小程序预览页面的二维码，体验者用户可通过扫一扫功能预览小程序。

图 4-86 上传小程序



审核通过后（审核时长一般为一周内），在微信小程序中可搜索到上传的应用。

----结束

5 入门实践

当您购买了Astro轻应用实例后，可以根据业务需要搭建低代码、零代码和大屏应用。本文介绍Astro轻应用常见的搭建实践，帮助您更好的使用Astro轻应用。

表 5-1 常用最佳实践

实践	描述
零代码应用开发示例	本最佳实践通过使用AstroZero的零代码应用模板快速创建培训报名应用为例，带您了解在AstroZero中开发一个零代码应用的大致流程，并认识相关概念。
创建Astro调查问卷	本最佳实践通过使用AstroZero的轻应用模板创建一个问卷调查应用为例，带您了解在AstroZero中开发一个低代码应用的大致流程，并认识应用开发中的相关概念。
交通管理大屏开发示例	本最佳实践以使用Astro Canvas中预置模板构建一个交通管理大屏为例，向您介绍如何在Astro Canvas中快速构建可视化页面。
简易出差审批应用开发	本最佳实践通过使用AstroZero的工作流功能，带您快速创建一个简易版的出差审批应用。