

华为云 Astro 轻应用

最佳实践

文档版本 01
发布日期 2025-09-04



版权所有 © 华为云计算技术有限公司 2025。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 华为云 Astro 轻应用最佳实践汇总	1
2 对象专项	5
2.1 使用华为云 Astro 轻应用将客户与订单数据关联并同步修改	5
2.2 使用华为云 Astro 轻应用在前端表格中增删改对象数据	15
3 脚本专项	27
3.1 通过华为云 Astro 轻应用中的脚本实现表单的提交限制功能	27
3.2 通过华为云 Astro 轻应用中的脚本实现表格数据的增加和删除	36
4 模板专项	51
4.1 使用华为云 Astro 轻应用文件模板生成合同文档	51
4.2 使用华为云 Astro 轻应用邮件模板发送邮件通知	90
5 工作流专项	111
5.1 通过华为云 Astro 轻应用流程模板创建出差审批电子流	111
5.2 通过华为云 Astro 轻应用工作流中的用户任务实现优先审批功能	128
6 标准页面专项	172
6.1 为华为云 Astro 轻应用标准页面中表格的数据增加链接	172
6.2 为华为云 Astro 轻应用标准页面中的表格增加求和等计算能力	179
6.3 在华为云 Astro 轻应用标准页面的表格中显示图片	188
6.4 如何在审批页面直接复用提交的表单页面	194
6.5 如何自定义华为云 Astro 轻应用标准页面组件包	213
6.5.1 自定义标准页面组件方案概述	213
6.5.2 自定义标准页面组件操作流程	213
6.5.3 自定义标准页面组件操作步骤	214
7 高级页面专项	233
7.1 使用华为云 Astro 轻应用自定义组件在页面中的属性	233
7.2 使用华为云 Astro 轻应用为组件配置中英文语言属性	239
7.3 使用华为云 Astro 轻应用创建高级页面适配多终端显示	246
7.4 使用华为云 Astro 轻应用开发高级页面时如何引用第三方库	253
7.5 在华为云 Astro 轻应用高级页面中使用花瓣图展示订单数据	259
7.6 在华为云 Astro 轻应用高级页面中使用轮播组件实现图片展示和 URL 跳转	269
8 连接器专项	276

8.1 通过华为云 Astro 轻应用中的连接器上传并识别身份证图片.....	276
8.2 通过华为云 Astro 轻应用中的连接器实现文件上传功能.....	288
8.3 通过华为云 Astro 轻应用连接器对接第三方数据库.....	293
9 业务用户专项.....	310
9.1 通过华为云 Astro 轻应用开发业务用户登录页.....	310
9.1.1 业务用户登录页方案概述.....	310
9.1.2 业务用户登录页后端逻辑开发实施步骤.....	311
9.1.3 业务用户登录页前台开发实施步骤.....	330
10 智能化专项.....	351
10.1 通过华为云 Astro 轻应用智能助手构建知识库.....	351
10.2 通过华为云 Astro 轻应用调用大模型连接器实现智能助手聊天对话.....	364
11 零代码专项.....	373
11.1 如何设置选项关联.....	373

1

华为云 Astro 轻应用最佳实践汇总

本文汇总了基于华为云Astro轻应用（Huawei Cloud Astro Zero）常见应用场景的操作实践，为每个实践提供详细的方案描述和操作指导，帮助用户深入了解华为云Astro轻应用的各个功能。

表 1-1 华为云 Astro 轻应用最佳实践一览表

最佳实践	说明
使用华为云Astro轻应用将客户与订单数据关联并同步修改	在某些订单系统中，通常需要将客户信息和订单数据进行关联，用于处理订单、扣减库存等。本实践主要向您介绍如何将两个对象进行关联，实现对象数据的同步修改功能。
使用华为云Astro轻应用在前端表格中增删改对象数据	本实践主要向您介绍如何通过增加一个工具栏，在前端页面实现对象数据的增加、删除和修改。
通过华为云Astro轻应用中的脚本实现表单的提交限制功能	在开发前端页面时，可以在脚本中为表单添加一些提交限制，来提升用户体验和数据的安全。本实践向您介绍，如何在脚本中定义一个延迟时间，在规定的时间内提交表单时，提示“Submission failed: Not PortalUser!”；超出规定的时间，则提示“Submission failed: Submitted too late”。
通过华为云Astro轻应用中的脚本实现表格数据的增加和删除	本实践向您介绍如何通过脚本，实现表格数据的增加和删除。
使用华为云Astro轻应用文件模板生成合同文档	华为云Astro轻应用低代码平台提供了文档类模板创建能力，支持快速创建“Word文档”、“Excel文档”、“邮件”和“短信”四种类型的模板，以满足多样化的业务文档需求，提升文档的制作效率与规范性。 本实践以Word文档为例进行介绍，基于该功能开发者可以根据不同客户设置不同类型的打印模板。

最佳实践	说明
使用华为云Astro轻应用邮件模板发送邮件通知	华为云Astro轻应用低代码平台提供了文档类模板创建能力，支持快速创建“Word文档”、“Excel文档”、“邮件”和“短信”四种类型的模板，以满足多样化的业务文档需求，提升文档的制作效率与规范性。 本实践以邮件模板为例进行介绍，基于该功能开发者可以预先设置好邮件格式和框架，避免每次都重复的编写。
通过华为云Astro轻应用流程模板创建出差审批电子流	华为云Astro轻应用低代码平台基于业界BPMN 2.0标准，实现了自己的业务流程管理系统，即工作流。本实践通过创建一个出差审批应用，帮助您快速熟悉华为云Astro轻应用中的工作流。
通过华为云Astro轻应用工作流中的用户任务实现优先审批功能	本实践将为您介绍在华为云Astro轻应用工作流中，如何通过配置用户任务的审批动作和终止动作，在会签过程中实现优先级审批功能。
为华为云Astro轻应用标准页面中表格的数据增加链接	在标准页面中，支持为表格中的数据增加超链接，来提升用户体验和数据交互的便捷性。本实践向您介绍，如何在表格的webName列中，将鼠标移动至WEB A上在页面的左下角可查看到对应的链接地址，单击会跳转到对应的页面。
为华为云Astro轻应用标准页面中的表格增加求和等计算能力	在标准页面中，支持为表格增加求和、求积等计算能力，来提升数据的处理效率。本实践主要向您介绍，如何将表格中“商品花费”列的值设置为“商品数*价格+其他成本”。
在华为云Astro轻应用标准页面的表格中显示图片	在表格中显示图片可增强信息的表达效果，信息更直观、生动和易于理解。本实践向您介绍，如何在标准页面中，通过自定义列的显示类型，将图片显示在表格中。
如何在审批页面直接复用提交的表单页面	在标准页面中，通过表格创建数据提交页面和通过表格创建数据审批页面是两个常用功能，如员工出差申请、请假申请、报销单据审批及商品出入库审批等。当数据提交页面与数据审批页面内容差异不大时，可以考虑将两个功能复用至同一页面，避免重复开发，提高开发效率。
如何自定义华为云Astro轻应用标准页面组件包	华为云Astro轻应用提供了自定义低代码页面组件的能力，当预置的组件不能满足您的业务需求时，您可以按照本实践提供的组件规范开发符合自己业务的组件，以此来灵活地扩展平台功能，提高应用构建的效率和灵活性。 本实践以开发一个imgButton组件为例，向您介绍华为云Astro轻应用自定义组件的规范。
使用华为云Astro轻应用自定义组件在页面中的属性	组件预置的属性不能满足您的业务需求时，支持为组件自定义属性。本实践向您介绍，如何为组件widget_demo_property自定义Text Property、Checkbox Property和Select Property三个属性。

最佳实践	说明
使用华为云Astro轻应用为组件配置中英文语言属性	为组件配置多语言属性，实现组件在不同语种环境下都可正常显示。在华为云Astro轻应用中对组件进行国际化配置，主要是修改国际化资源文件（i18n）。本实践向您介绍，如何为组件widget_demo_i18n配置中文和英文两种语言属性。
使用华为云Astro轻应用创建高级页面适配多终端显示	当用户开发的高级页面应用于多种设备时，如何才能保证在不同大小的设备上，能够呈现同样的网页？为此，华为云Astro轻应用提供了高级页面的电脑端和移动端两种终端视图、流式布局的响应式布局，并为绝对布局提供“拉伸”功能辅助自适应。 本实践以开发一个满足响应式布局的商品列表组件为例，向您介绍如何适配多终端。
使用华为云Astro轻应用开发高级页面时如何引入第三方库	库是支撑高级页面组件运行的第三方依赖，如果缺少相应的库，则高级页面组件不能正常运行。本实践向您介绍，如何引入第三方库MintUI，来降低组件开发复杂度、丰富组件的功能。
在华为云Astro轻应用高级页面中使用轮播组件实现图片展示和URL跳转	高级页面中的轮播组件主要用于多个图片的自动循环切换，您也可以为图片添加超链接，即单击图片，跳转到指定的网站。本实践向您介绍，如何为轮播组件中图片添加一个超链接，跳转到华为云Astro轻应用帮助文档。
在华为云Astro轻应用高级页面中使用花瓣图展示订单数据	高级页面中组件展示的数据除了系统预置的静态数据之外，还支持动态数据，即通过调用脚本、服务编排或对象等接口动态生成的数据。本实践向您介绍如何在高级页面中，通过玫瑰花饼图展示订单数据。
通过华为云Astro轻应用中的连接器上传并识别身份证图片	华为云Astro轻应用封装了不同类型的连接器用于对接其他服务，对接后即可在应用中使用该服务。本实践向您介绍如何通过OCR连接器，实现身份证信息的识别和存储能力。
通过华为云Astro轻应用中的连接器实现文件上传功能	华为云Astro轻应用封装了不同类型的连接器用于对接其他服务，对接后即可在应用中使用该服务。本实践向您介绍如何通过OBS连接器，将前端页面中上传的文件存储到OBS桶中。
通过华为云Astro轻应用连接器对接第三方数据库	本实践将为您介绍如何通过自定义连接器将第三方数据库集成到华为云Astro轻应用中。
通过华为云Astro轻应用开发业务用户登录页	华为云Astro轻应用为每个应用预置了一个默认的登录页，业务用户可通过默认的登录页登录应用。本实践向您介绍，如何为应用自定义一个登录页。
通过AstroZero华为云Astro轻应用智能助手构建知识库	对于查询文献资料的场景，当用户自身资料文档数量庞大，使用资料不方便、查询效率低下时，华为云Astro轻应用标准页面中的智能助手组件能够协助用户调用大模型接口获取答案。同时，通过RAG这个技术，结合知识库，为用户提供更为智能、自然的对话体验，提升信息检索的准确性和相关性。

最佳实践	说明
通过AstroZero华为云Astro轻应用调用大模型连接器实现智能助手聊天对话	开发者使用华为云Astro轻应用开发标准页面时，如果需要通过智能助手组件对接外部大模型接口，可以先在大模型连接器中配置大模型对接信息，然后再在智能助手组件中选取对应的大模型连接器，即可实现智能助手对话应用。 本实践将为您介绍，如何配置大模型连接器，以及在智能助手组件中使用大模型连接器，实现智能助手聊天对话应用。
如何设置选项关联	在数据录入、数据查询和流程审批等类型应用中，经常会使用联动筛选、限定可选范围，确保数据逻辑一致，减少用户操作失误。本实践向您介绍在华为云Astro轻应用零代码平台中如何通过“选项关联”来实现此功能。

2 对象专项

2.1 使用华为云 Astro 轻应用将客户与订单数据关联并同步修改

期望实现效果

在某些订单系统中，通常需要将客户信息和订单数据进行关联，用于处理订单、扣减库存等。例如，订单应用“A”中存在客户对象customerList和订单对象orderList两个对象，在华为云Astro轻应用中通过为两个对象建立关联关系，可轻松实现如下功能：

- 客户对象页面可以新增、展示客户对象数据

图 2-1 新增客户对象数据

创建 ×

customerList0

名称

customerid

customerName

2

2

客户2

取消

保存

保存并新建

图 2-2 展示客户对象数据

名称

customerid

customerName

操作

新增

重置

+

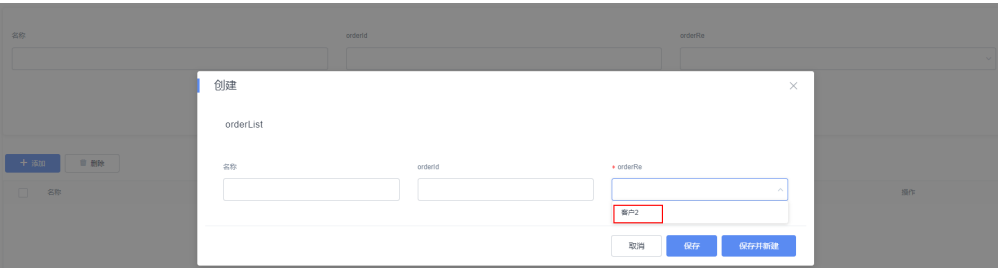
清除

删除

名称	customerid	customerName	操作
☐	2	客户2	修改删除

- 订单对象页面可以新增订单数据，关联已有客户信息

图 2-3 新增订单数据、关联已有客户信息

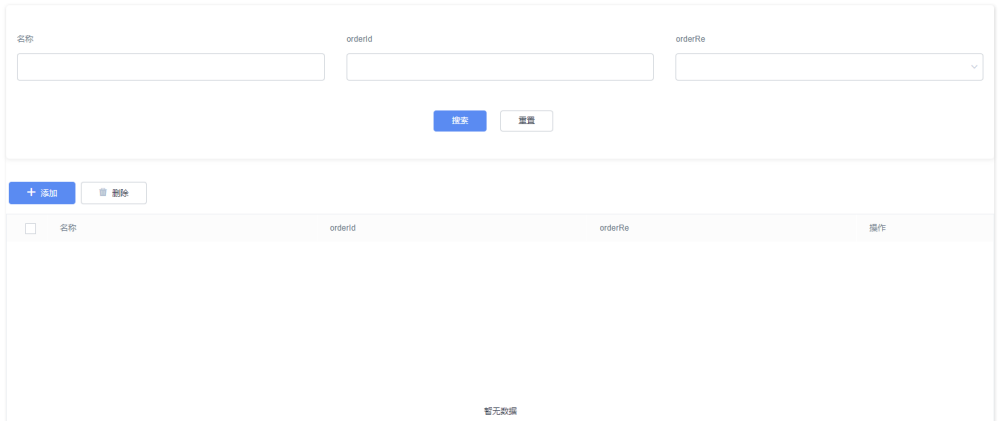


- 删除客户数据后，订单数据也随之删除

图 2-4 删除客户数据



图 2-5 订单数据同时被删除



功能实现方法

- 步骤1** 创建一个低代码应用。
1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
 2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
 3. 在“应用”中，单击“新建低代码应用”或单击 ，进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。
 4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。

5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 2-6 创建一个空白应用

新建 空白应用

基本信息

添加图标

* 标签

我的第一个应用

分类

请选择

* 名称

A

描述

请输入

高级设置

取消

新建

表 2-1 新建空白应用参数说明

参数	说明	示例
标签	新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。	我的第一个应用
名称	新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下： - 长度不能超过31个字符，包括前缀命名空间的长度。 - 名称前的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。	A

步骤2 创建客户对象customerList和订单对象orderList，并为对象添加字段。

1. 在应用设计器的左侧导航栏中，选择“数据”，单击对象中的“+”。
2. 设置对象的名称和唯一标识，单击“确定”。

图 2-7 创建对象 customerList

创建新对象

您可以创建用来保存持久化数据的实体对象，对象里的数据存储在平台里面

从空白创建

从模板创建

您将创建单个对象，从零开始设计对象的字段，并添加测试数据

* 对象名称

customerList

* 唯一标识

customerList

取消

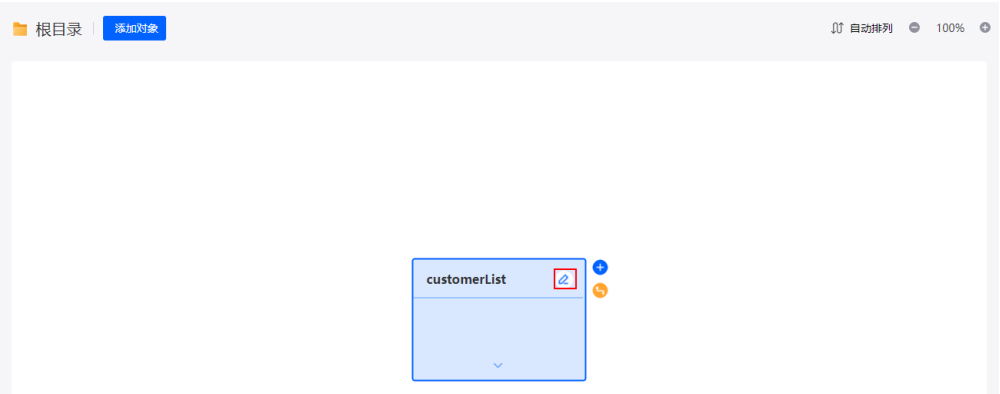
确定

表 2-2 新建 customerList 对象参数说明

参数	说明	示例
对象名称	新建对象的名称，创建后可修改。 取值范围：1~80个字符。	customerList
唯一标识	新建对象在系统中的标识，创建后不支持修改。命名要求如下： - 长度不能超过63个字符，包括前缀命名空间的长度。标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母，数字和下划线组成，且不能以下划线结尾。	customerList

3. 在已创建的对象中，单击，进入对象详情页面。

图 2-8 选择编辑按钮



4. 在“字段”页签，单击“添加”，为对象添加customerId字段。

图 2-9 添加 customerId 字段

添加字段

*

显示名称

customerId

*

唯一标识

customerId

*

字段类型

文本

...

*

数据长度

64

描述

取消

确认

表 2-3 添加 customerId 字段参数说明

参数	说明	示例
显示名称	新建字段的名称，创建后可修改。 取值范围：1~63个字符。	customerId

参数	说明	示例
唯一标识	新建字段在系统中的标识，创建后不支持修改。命名要求如下： - 长度不能超过63个字符，包括前缀命名空间的长度。 - 必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。	customerId
字段类型	单击 *** ，在弹出的页面中，根据页面提供的参数解释，选择新建字段所属的类型。	文本
数据长度	允许输入字段的长度。	64

5. 在“字段”页签，再次单击“添加”按钮，添加customerName字段。

图 2-10 添加 customerName 字段

添加字段

*

显示名称

customerName

*

唯一标识

customerName

*

字段类型

文本

*

数据长度

64

描述

取消

确认

6. 按照上述操作，创建订单对象orderList，并为其添加orderId字段。

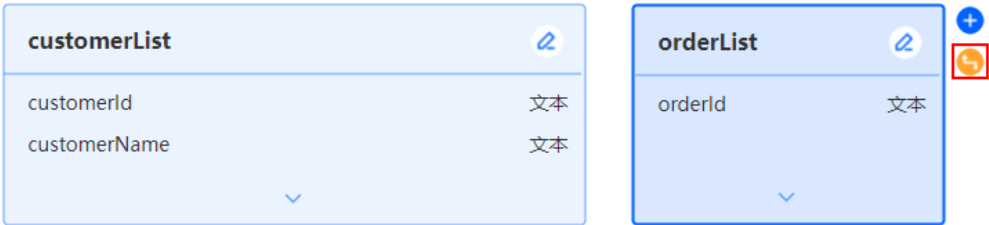
图 2-11 创建 orderList 对象并添加字段



步骤3 选择订单对象，添加关联。

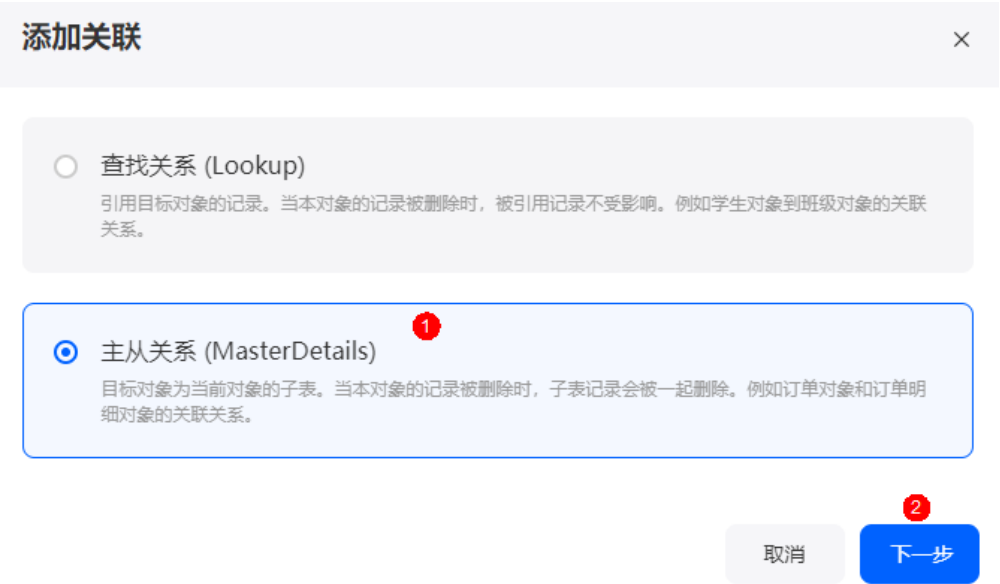
1. 单击对象上的，进入添加关联页面。

图 2-12 选择关联图标



2. 选择“主从关系”，单击“下一步”。
- 主从关系是指目标对象为当前对象的子表，通过关联当前字段与另一对象的ID字段，创建本对象与另一对象的主从关系。定义了主从关系后，本字段的取值只能来源于关联主对象。当本对象的记录被删除时，子表记录会被一起删除。

图 2-13 选择主从关系



3. 添加关联关系，单击“确定”。

图 2-14 添加关联

添加关联

*

显示名称

orderRe

*

唯一标识

orderRe

*

关联对象

customerList (__customerList__CST)

取消

上一步

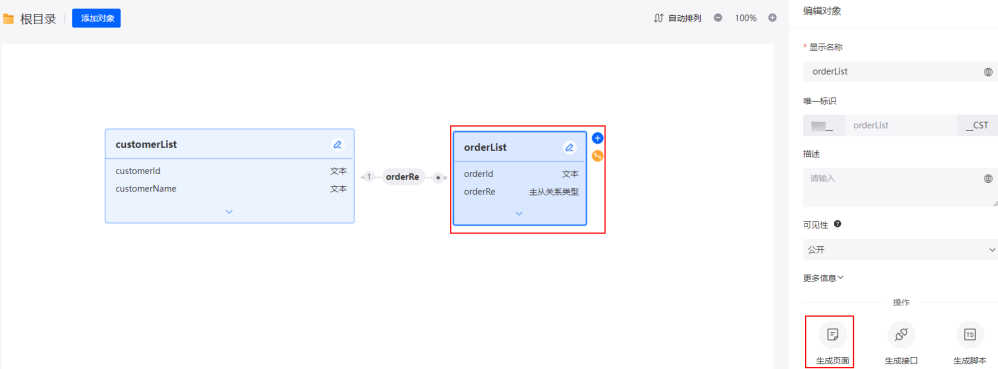
确定

表 2-4 添加 orderRe 关联参数说明

参数	说明	示例
显示名称	关联关系在页面显示的名称，创建后可修改。 取值范围：1~80个字符。	orderRe
唯一标识	关联关系在系统中的唯一标识，创建后不可修改。 命名要求如下： - 长度不能超过63个字符，包括前缀命名空间的长度。 - 必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。	orderRe
关联对象	选择关联目标，下拉框展示当前应用内所有对象的显示名称。	选择 步骤2 中创建的客户对象customerList。

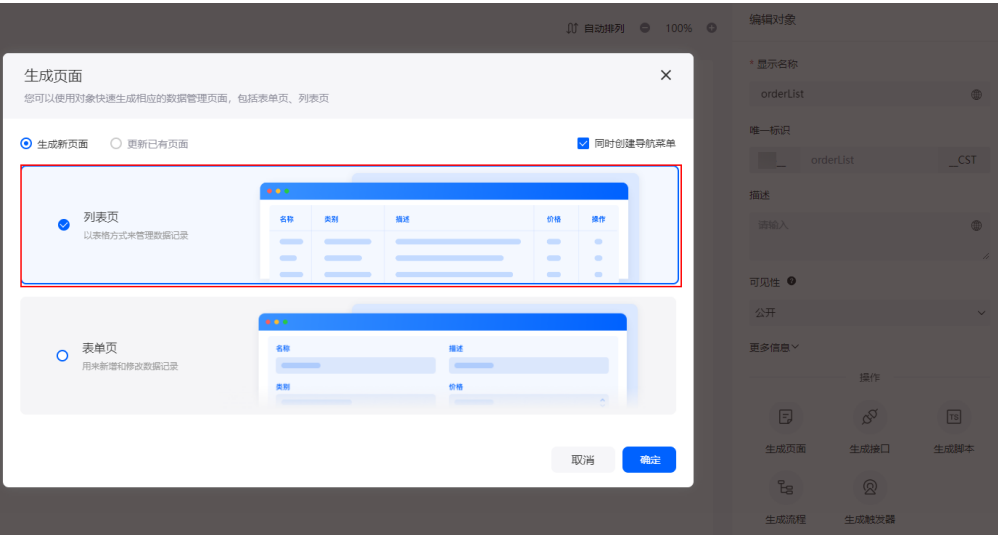
- 步骤4** 分别将客户和订单对象生成页面。
- 选中订单对象orderList，在右侧编辑对象中，选择“操作 > 生成页面”。

图 2-15 选择生成页面



2. 取消选中“表单页”，仅保留“列表页”，单击“确定”。

图 2-16 将 orderList 对象生成列表页



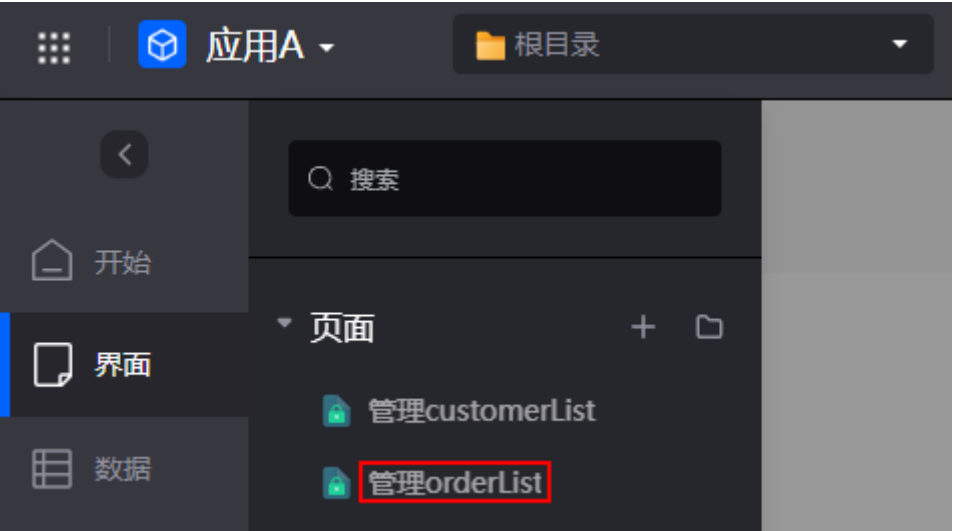
创建成功后，页面会显示“页面生成成功”的提示。进入“界面 > 页面”中，可查看到生成的列表页。

3. 按照同样的方法，将客户对象customerList也生成列表页。

步骤5 在订单对象页面设置父对象显示的字段。

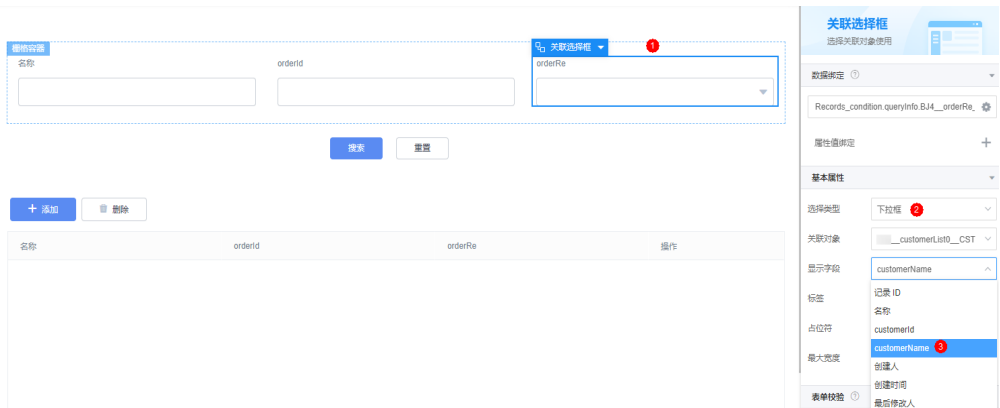
1. 在应用设计器的左侧导航栏中，选择“界面”。
2. 在“页面”中，单击“管理orderList”。

图 2-17 单击订单对象页面 orderList



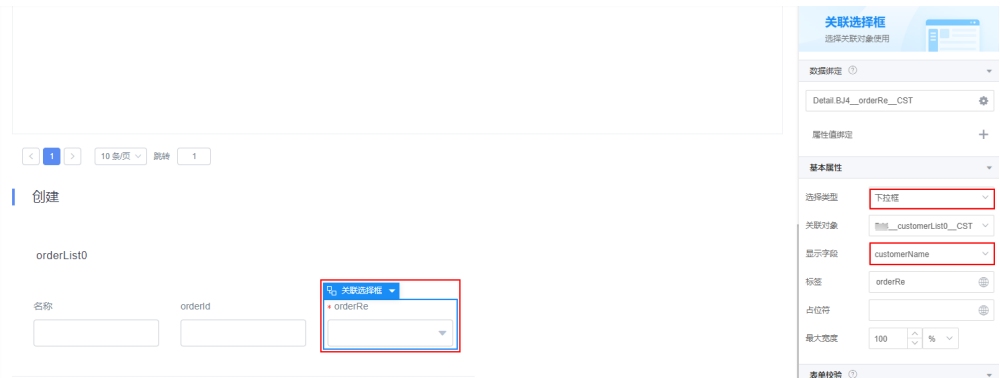
3. 选中“orderRe”字段，将“属性 > 基本属性”中的“选择类型”设置为“下拉框”，“显示字段”设置为“customerName”。

图 2-18 设置父对象显示字段



4. 在下方的创建中，同样选中“orderRe”字段，将“属性 > 基本属性”中的“选择类型”设置为“下拉框”，“显示字段”设置为“customerName”。

图 2-19 为创建页面设置父对象显示的字段



步骤6 单击页面上方的，保存页面。

步骤7 保存成功后，单击页面上方的，查看页面配置效果。

----结束

2.2 使用华为云 Astro 轻应用在前端表格中增删改对象数据

期望实现效果

通过添加一个工具栏，在前端页面实现对象数据的增加、删除和修改。例如，在标准页面的表格中，增加、删除和修改数据时，页面关联的后台对象数据也会随之更改。

最终实现效果：双击demoName中的数据栏，可以对数据进行编辑。编辑后，单击“保存”按钮，可发现表格中数据已经被更新，同时后台对象也已被更新。选择对应的表格列，单击“删除”按钮，确认删除。删除成功后，后台对象中的数据也会随之删除。

图 2-20 在页面更新数据



图 2-21 对象中数据也随之更新

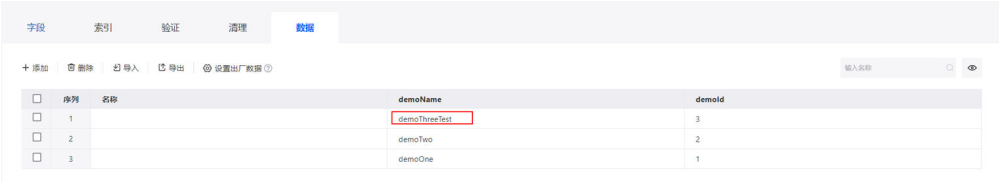


图 2-22 执行删除操作



图 2-23 页面中数据被删除

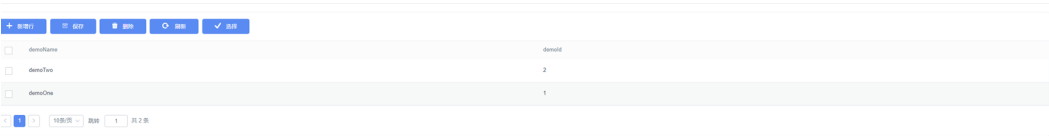


图 2-24 对象中数据也随之删除

字段索引验证清理数据				
+ 添加 - 删除 2 导入 1 导出 设置出厂数据				
输入名称				
☐	序列	名称	demoName	demoId
☐	1		demoTwo	2
☐	2		demoOne	1

功能实现方法

步骤1 创建一个低代码应用。

1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
3. 在“应用”中，单击“新建低代码应用”或单击，进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。
4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。
5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 2-25 创建一个空白应用

新建 空白应用

基本信息


添加图标

* 标签

我的第一个应用

分类

请选择

高级设置

* 名称

A

描述

请输入

取消

新建

表 2-5 新建空白应用参数说明

参数	说明	示例
标签	新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。	我的第一个应用

参数	说明	示例
名称	新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下： - 长度不能超过31个字符，包括前缀命名空间的长度。 - 名称前的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。	A

步骤2 创建对象“demoData”，并为对象添加字段和数据。

1. 在应用设计器的左侧导航栏中，选择“数据”，单击对象中的“+”。

2. 设置对象的名称和唯一标识为“demoData”，单击“确定”。

图 2-26 创建对象 demoData



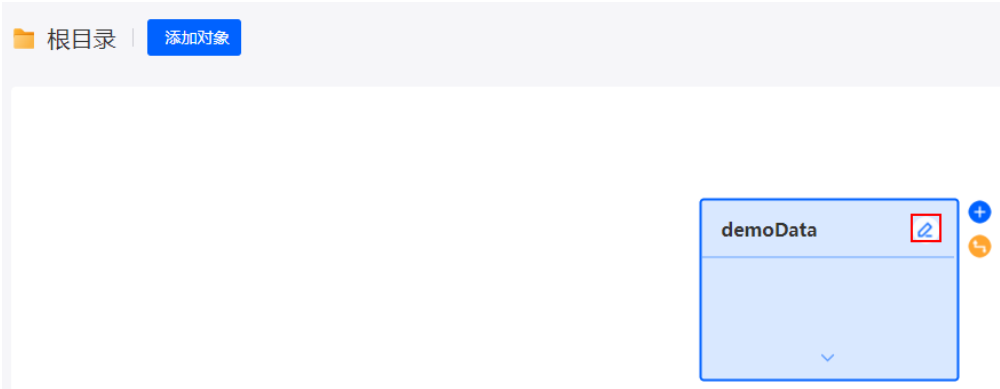
表 2-6 新建 demoData 对象参数说明

参数	说明	示例
对象名称	新建对象的名称，创建后可修改。 取值范围：1~80个字符。	demoData

参数	说明	示例
唯一标识	新建对象在系统中的标识，创建后不支持修改。命名要求如下： - 长度不能超过63个字符，包括前缀命名空间的长度。标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母，数字和下划线组成，且不能以下划线结尾。	demoData

3. 在已创建的对象中，单击，进入对象详情页面。

图 2-27 选择编辑按钮



4. 在“字段”页签，单击“添加”，为对象添加demoName字段。

图 2-28 添加 demoName 字段

添加字段

*

显示名称

demoName

*

唯一标识

demoName

*

字段类型

文本

*

数据长度

64

描述

取消

确认

表 2-7 添加 demoName 字段参数说明

参数	说明	示例
显示名称	新建字段的名称，创建后可修改。 取值范围：1~63个字符。	demoName
唯一标识	新建字段在系统中的标识，创建后不支持修改。命名要求如下： - 长度不能超过63个字符，包括前缀命名空间的长度。 - 必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。	demoName
字段类型	单击***，在弹出的页面中，根据页面提供的参数解释，选择新建字段所属的类型。	文本

5. 在“字段”页签，再次单击“添加”按钮，添加demoId字段。

图 2-29 添加 demold 字段

添加字段

*

显示名称

demold

*

唯一标识

demold

*

字段类型

文本

...

*

数据长度

64

描述

取消

确认

6. 选择“数据”页签，单击“添加”，为对象添加数据。

图 2-30 为对象添加数据

字段索引验证清理数据

+ 添加

删除

导入

导出

设置出厂数据 ⑦

输入名称

☐	序列	名称	demoName	demold
☐	1		demoThree	3
☐	2		demoTwo	2
☐	3		demoOne	1

步骤3 新建对象模型。

1. 在应用设计器的左侧导航栏中，选择“界面”，单击页面后的“+”。

2. 输入页面的标签和名称，单击“添加”，新建一个标准页面。

图 2-31 新建一个标准页面



表 2-8 标准页面参数说明

参数	说明	示例
标签	标准页面的标签名，创建后可修改。 取值范围：1~64个字符。	page01
名称	标准页面的名称，名称是标准页面在系统中的唯一标识，创建后不可修改。命名要求如下： - 长度不能超过64个字符，包括前缀命名空间的长度。 - 必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。	page01

3. 在标准页面的底部，单击“模型视图”，从设计视图切换到模型视图。

图 2-32 单击模型视图



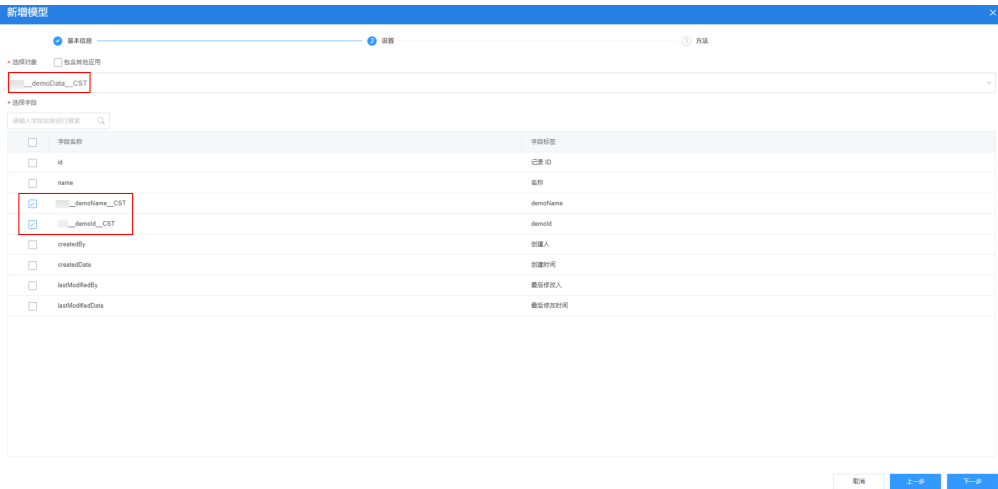
4. 单击“新增模型”，输入模型名称（如demo）、“来源”选择“对象”，单击“下一步”。

图 2-33 新建模型



5. 选择步骤3中创建的对象和添加的字段，单击“下一步”。

图 2-34 选择对象和字段



6. 单击“确定”，完成模型的创建。

步骤4 返回设计视图页面，新建表格关联模型。

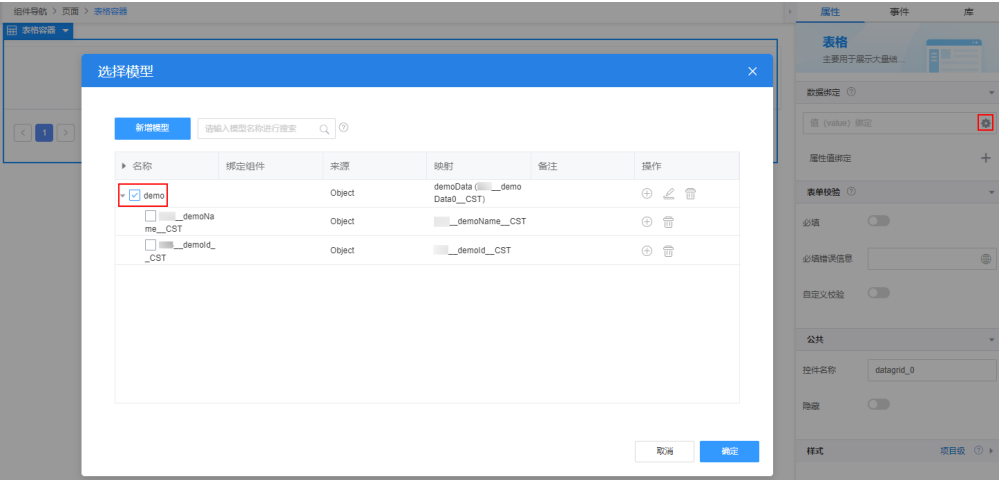
- 1. 在标准页面的底部，单击“设计视图”，从模型视图切换回设计视图。
- 2. 在标准页面中，拖入一个表格组件。

图 2-35 拖入表格组件



- 3. 选中表格组件，在“属性 > 数据绑定 > 值绑定”中，单击 .
- 4. 选中步骤3中创建的模型，单击“确定”。

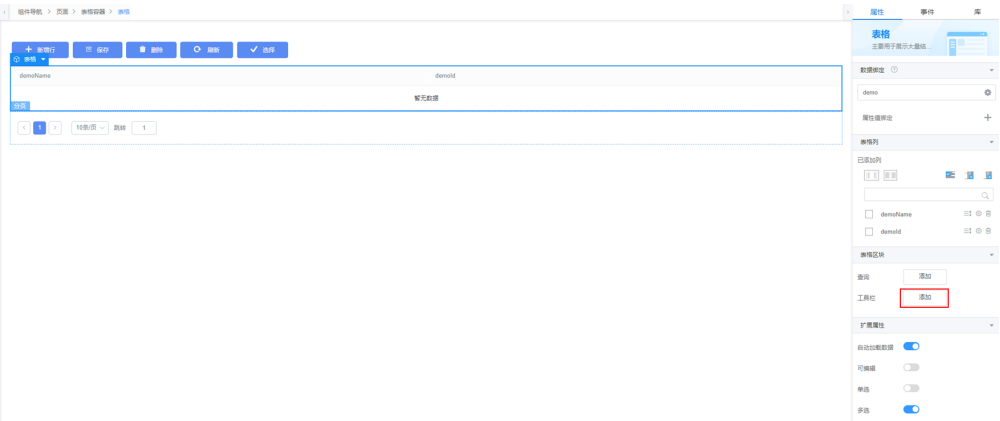
图 2-36 选择模型



步骤5 添加工具栏。

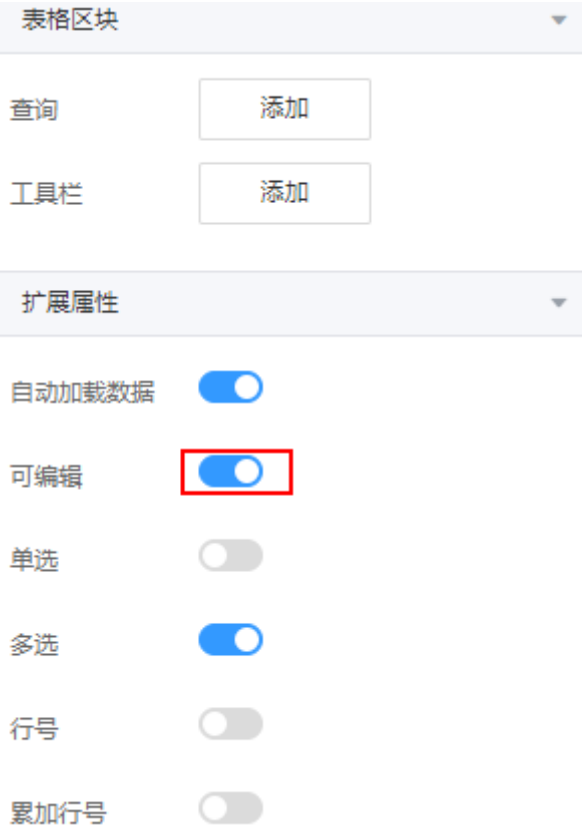
1. 选中表格组件，在“属性 > 表格区块”中，单击工具栏后的“添加”，添加一个工具栏。

图 2-37 添加工具栏



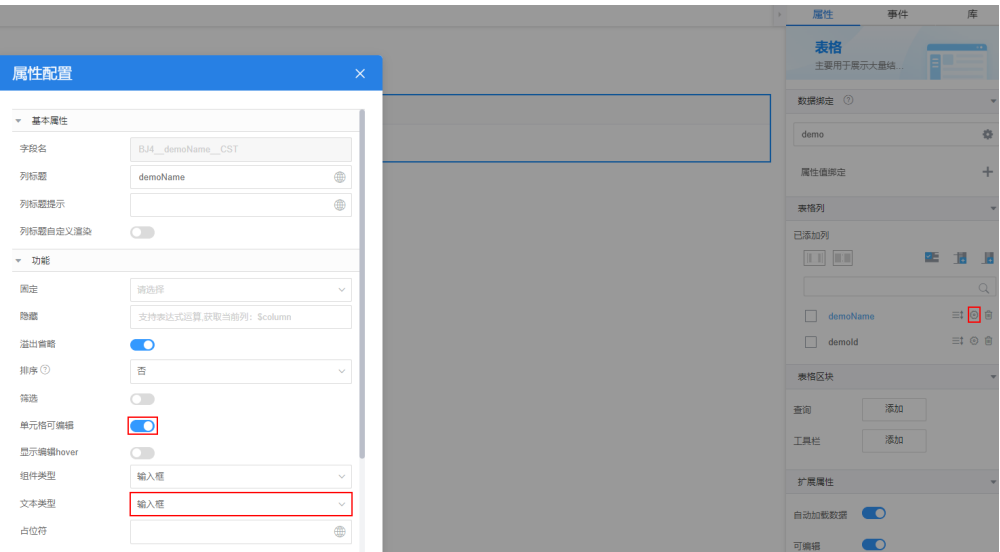
2. 在扩展属性中，开启“可编辑”。

图 2-38 开启可编辑



3. 将demoName、demoId列设为可编辑。
- 在已添加列中，单击demoName后的⚙️，将列设置为可编辑。按照上述操作，将demoId列也设置为可编辑。

图 2-39 将列 demoName 设置为可编辑



步骤6 单击页面上方的，保存页面。

步骤7 保存成功后，单击页面上方的，查看页面配置效果。

----结束

3 脚本专项

3.1 通过华为云 Astro 轻应用中的脚本实现表单的提交限制功能

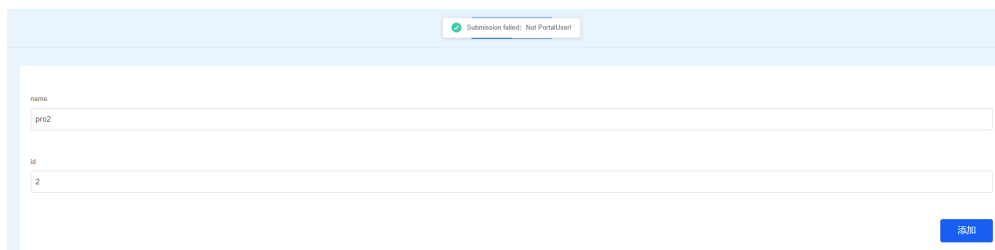
期望实现效果

在开发前端页面时，可以在脚本中为表单添加一些提交限制，来提升用户体验和数据的安全。例如，在脚本中定义一个延迟时间（如[图3-1](#)），在规定的时间内提交表单时，提示“Submission failed: Not PortalUser!”（如[图3-2](#)）；超出规定的时间，则提示“Submission failed: Submitted too late”（如[图3-3](#)）。

图 3-1 通过脚本限定

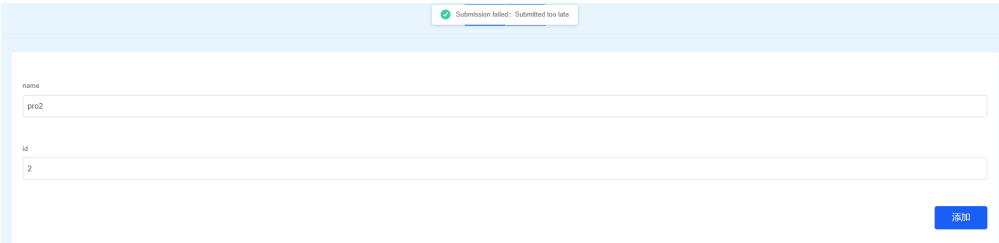
```
try {  
  let currentTime = now();  
  let date = toDate('2024-04-08 20:08:08', 'yyyy-MM-dd HH:mm:ss');  
  if (date.getTime() < currentTime.getTime()) { // 限制提交时间  
    error.name = "WOERROR";  
    error.message = "Submitted too late";  
    throw error;  
  }  
}
```

图 3-2 非业务用户无法提交



The screenshot shows a web form with three input fields labeled 'name', 'pro2', and 'id'. The 'id' field contains the value '2'. At the top of the form, there is a green error message: 'Submission failed: Not PortalUser!'. At the bottom right of the form, there is a blue button labeled '添加' (Add).

图 3-3 提示提交过晚



功能实现方法

- 步骤1** 创建一个低代码应用。
1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
 2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
 3. 在“应用”中，单击“新建低代码应用”或单击，进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。
 4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。
 5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 3-4 创建一个空白应用



表 3-1 新建空白应用参数说明

参数	说明	示例
标签	新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。	我的第一个应用

参数	说明	示例
名称	新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间_。命名要求如下： - 长度不能超过31个字符，包括前缀命名空间的长度。 - 名称前的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。	A

步骤2 创建对象“product”，并为对象添加字段。

1. 在应用设计器的左侧导航栏中，选择“数据”，单击对象中的“+”。
2. 设置对象的名称和唯一标识为“product”，单击“确定”。

图 3-5 创建对象 product



表 3-2 新建 product 对象参数说明

参数	说明	示例
对象名称	新建对象的名称，创建后可修改。 取值范围：1~80个字符。	product

参数	说明	示例
唯一标识	新建对象在系统中的标识，创建后不支持修改。命名要求如下： - 长度不能超过63个字符，包括前缀命名空间的长度。标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母，数字和下划线组成，且不能以下划线结尾。	product

3. 在已创建的对象中，单击，进入对象详情页面。
4. 在“字段”页签，单击“添加”，为对象添加ProName字段。

图 3-6 添加 ProName 字段

添加字段

*

显示名称

proName



*

唯一标识



proName

*

字段类型

文本

...

*

数据长度

64

描述



取消

确认

表 3-3 添加 proName 字段参数说明

参数	说明	示例
显示名称	新建字段的名称，创建后可修改。 取值范围：1~63个字符。	proName
唯一标识	新建字段在系统中的标识，创建后不支持修改。命名要求如下： - 长度不能超过63个字符，包括前缀命名空间的长度。 - 必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。	proName
字段类型	单击***，在弹出的页面中，根据页面提供的参数解释，选择新建字段所属的类型。	文本

5. 在“字段”页签，再次单击“添加”按钮，添加prold字段。

图 3-7 添加 prold 字段

添加字段

* 显示名称

prold

* 唯一标识

_

prold

* 字段类型

文本

* 数据长度

64

描述

取消

确认

步骤3 创建一个脚本。

1. 在应用设计器中，选择“逻辑”，单击脚本后的“+”。

2. 新建一个空白的脚本，名称设置为“submitLimit”，单击“添加”。

图 3-8 创建脚本 submitLimit

新增脚本

创建一个新脚本

使用已有脚本

* 名称

submitLimit

模板

空脚本

描述

取消

添加

- 在脚本编辑器中，输入示例代码。

本示例代码主要实现的功能为：获取前端页面输入，对入参进行校验，并调用对象接口完成对象实例增加。示例中的“命名空间__product__CST”为[步骤2](#)中创建的对象。

```
//本脚本用于提交表单以及限制提交时间、用户类型
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 { now } from 'date';
import { toDate } from 'date';

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

//使用数据对象命名空间__product__CST
@useObject(['命名空间__product__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 currentTime = now();
      let date = toDate('2024-04-08 20:08:08', 'yyyy-MM-dd HH:mm:ss');
      if (date.getTime() < currentTime.getTime()) { // 限制提交时间
        error.name = "WOERROR";
        error.message = "Submitted too late";
        throw error;
      }
    }
  }
}
```



```

    }
    let user = context.getUserType();
    if (user != "PortalUser") { // 限制提交用户类型，只有业务用户PortalUser可以提交
        error.name = "WOERROR";
        error.message = "Not PortalUser!";
        throw error;
    }

    let productData = new Object();
    productData['命名空间__proName__CST'] = input.proName; //入参赋值
    productData['命名空间__proId__CST'] = input.proId;
    let s = db.object('命名空间__product__CST'); //获取命名空间__product__CST这个Object的操作

实例
    let id = s.insert(productData);
    if (id) {
        out.id = id;
    } else {
        error.name = "WOERROR";
        error.message = "Unable to create product!";
        throw error;
    }
} catch (error) {
    console.error(error.name, error.message);
    context.setError(error.name, error.message);
}
return out;
}
}

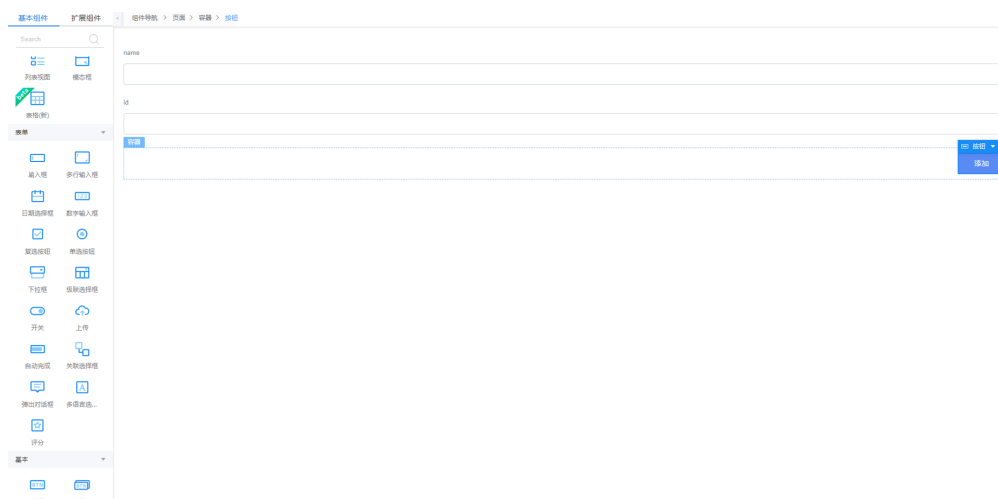
```

4. 单击，保存脚本，保存成功后单击，激活脚本。

步骤4 创建表单页面，用于提交表单数据。

1. 在应用设计器中，选择“界面”，单击页面后的“+”。
2. 输入页面的标签和名称，单击“添加”，新建一个标准页面。
3. 从基本组件中，拖拽2个输入框组件和1个按钮组件到右侧画布中。
4. 选中第一个输入框，将标签修改为“name”，将第二个输入框的标签修改为“id”，将按钮显示名称设置为“添加”。

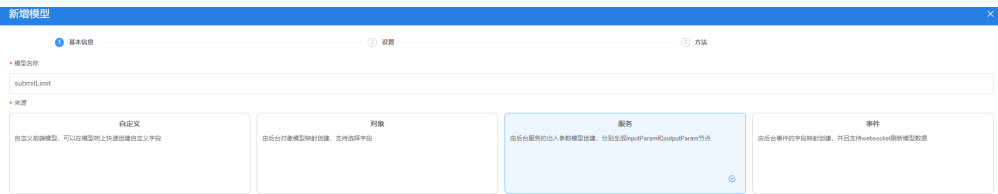
图 3-9 设计表单页面



步骤5 新建对象模型。

1. 在标准页面的底部，单击“模型视图”，从设计视图切换到模型视图。
2. 单击“新增模型”，输入模型名称（如submitLimit）、“来源”选择“服务”，单击“下一步”。

图 3-10 新建模型



3. “选择服务类型”设置为“脚本”，在弹出的选择服务页面选择步骤3中创建的脚本，单击“确定”。

图 3-11 选择脚本



4. 单击“下一步”，再单击“确定”，完成模型的创建。

步骤6 返回设计视图页面，新建表格关联模型。

1. 在标准页面底部，单击“设计视图”，从模型视图切换回设计视图。
2. 选中name输入框，在“属性 > 数据绑定 > 值绑定”中，单击。
3. 选择步骤5中创建的模型（proName），单击“确定”。

图 3-12 选择模型

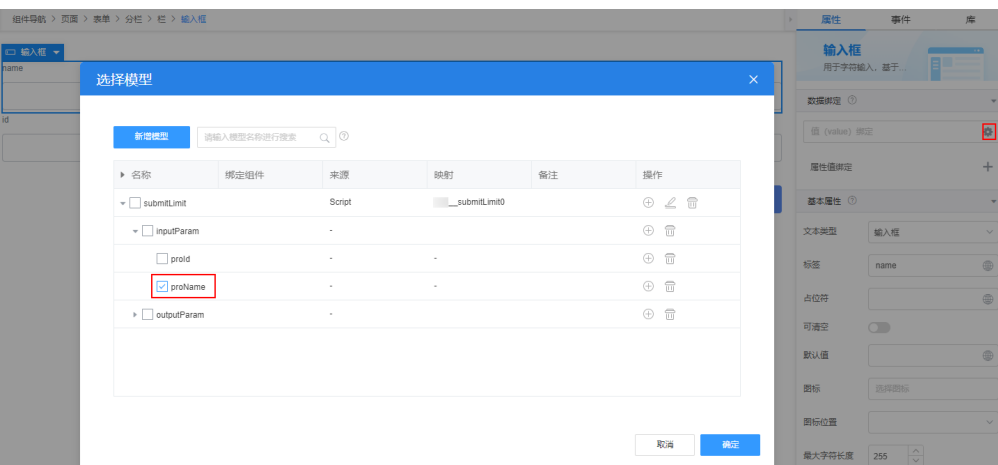
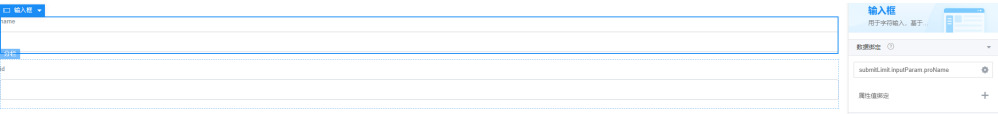
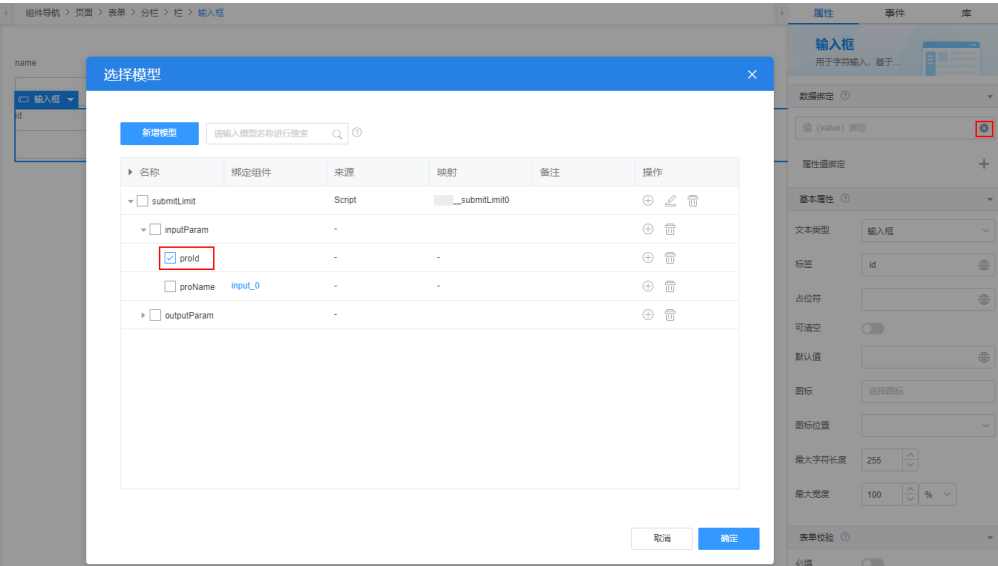


图 3-13 绑定后效果



4. 按照上述操作，为id输入框绑定[步骤5](#)中创建的模型（prold）。

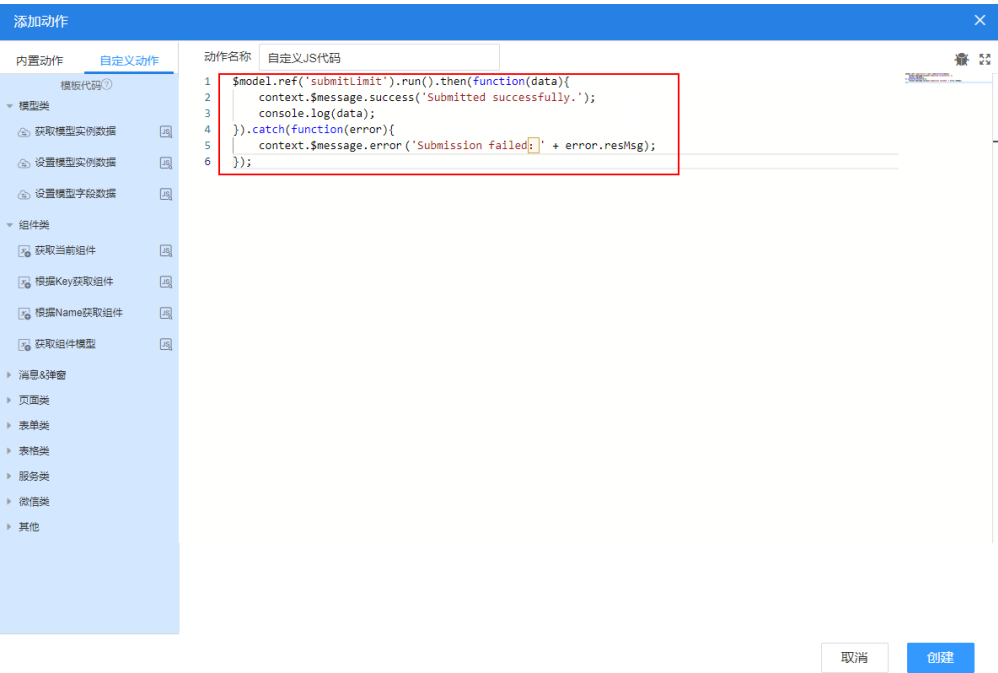
图 3-14 为 id 输入框绑定模型



步骤7 为添加按钮，添加事件。

1. 选中添加按钮组件，在页面右侧选择“事件”页签。
2. 单击“点击”后的“+”，进入添加动作页面。
3. 在自定义动作中，输入自定义代码，单击“创建”。

图 3-15 自定义动作



本示例中自定义的JS代码主要实现的功能为：单击提交按钮后，获取脚本抛出的异常信息并在页面展示。

```
$model.ref('submitLimit').run().then(function(data){
  context.$message.success('Submitted successfully. ');
  console.log(data);
}).catch(function(error){
  context.$message.error('Submission failed: ' + error.resMsg);
});
```

- 步骤8 单击页面上方的, 保存页面。
- 步骤9 保存成功后，单击页面上方的, 预览效果。
- 结束

3.2 通过华为云 Astro 轻应用中的脚本实现表格数据的增加和删除

期望实现效果

通过脚本，在前端页面实现对象数据的增加和删除。例如，在标准页面中增加或删除一条数据时，标准页面关联的对象中，数据也会随之添加或删除。

图 3-16 在页面添加数据

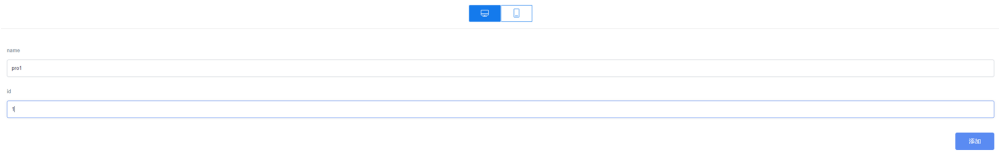


图 3-17 对象中同步新增数据

字段 索引 验证 清理 数据			
+ 添加 删除 导入 导出 设置出厂数据			
☐	序列	名称	proName
☐	1		pro1
			proId
			1

图 3-18 对象中有两条数据

字段 索引 验证 清理 数据			
+ 添加 删除 导入 导出 设置出厂数据			
☐	序列	名称	proName
☐	1		pro2
☐	2		pro1
			proId
			2
			1

图 3-19 在前端页面删除 proId 为 1 的数据



图 3-20 对象中 proId 为 1 的数据同步删除

字段	索引	验证	清理	数据
+ 添加 删除 导入 导出 设置出厂数据				
	序列	名称	proName	proId
	1		pro2	2

功能实现方法（增加功能实现）

步骤1 创建一个低代码应用。

1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
3. 在“应用”中，单击“新建低代码应用”或单击，进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。
4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。
5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 3-21 创建一个空白应用

新建 空白应用

基本信息



添加图标

标签

我的第一个应用

名称

A

分类

请选择

描述

请输入

高级设置

取消

新建

表 3-4 新建空白应用参数说明

参数	说明	示例
标签	新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。	我的第一个应用

文档版本 01 (2025-09-04)

版权所有 © 华为云计算技术有限公司

37

参数	说明	示例
名称	新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间_。命名要求如下： - 长度不能超过31个字符，包括前缀命名空间的长度。 - 名称前的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。	A

步骤2 创建对象“product”，并为对象添加字段。

1. 在应用设计器的左侧导航栏中，选择“数据”，单击对象中的“+”。
2. 设置对象的名称和唯一标识为“product”，单击“确定”。

图 3-22 创建对象 product



表 3-5 新建 product 对象参数说明

参数	说明	示例
对象名称	新建对象的名称，创建后可修改。 取值范围：1~80个字符。	product

参数	说明	示例
唯一标识	新建对象在系统中的标识，创建后不支持修改。命名要求如下： - 长度不能超过63个字符，包括前缀命名空间的长度。标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母，数字和下划线组成，且不能以下划线结尾。	product

3. 在已创建的对象中，单击，进入对象详情页面。
4. 在“字段”页签，单击“添加”，为对象添加ProName字段。

图 3-23 添加 ProName 字段

添加字段

*

显示名称

proName



*

唯一标识



proName

*

字段类型

文本

...

*

数据长度

64

描述



取消

确认

表 3-6 添加 proName 字段参数说明

参数	说明	示例
显示名称	新建字段的名称，创建后可修改。 取值范围：1~63个字符。	proName
唯一标识	新建字段在系统中的标识，创建后不支持修改。命名要求如下： - 长度不能超过63个字符，包括前缀命名空间的长度。 - 必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。	proName
字段类型	单击***，在弹出的页面中，根据页面提供的参数解释，选择新建字段所属的类型。	文本

5. 在“字段”页签，再次单击“添加”按钮，添加proId字段。

图 3-24 添加 proId 字段

添加字段

* 显示名称

proId

* 唯一标识

_

proId

* 字段类型

文本

* 数据长度

64

描述

取消

确认

步骤3 创建一个脚本。

1. 在应用设计器中，选择“逻辑”，单击脚本后的“+”。

2. 新建一个空白的脚本，名称设置为“sqlAdd”，单击“添加”。

图 3-25 创建脚本 sqlAdd

新增脚本

● 创建一个新脚本

使用已有脚本

* 名称

sqlAdd

模板

空脚本

描述

取消

添加

3. 在脚本编辑器中，输入示例代码。
- 本示例代码主要实现的功能为：在单击添加按钮时，获取前端输入的姓名、ID等数据，插入到对象中。如果插入失败，则记录失败信息。示例中的“命名空间__product__CST”为[步骤2](#)中创建的对象。

```
//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

//定义入参结构
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'String', required: true, label: 'String' })
  proId: string;
  @action.param({ type: 'String', required: true, label: 'String' })
  proName: 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;
}
//使用数据对象命名空间__product__CST
@useObject(['命名空间__product__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 productData = new Object();
      productData['命名空间__proName__CST'] = input.proName; //将入参赋值给productData变量，方便后面使用
      productData['命名空间__proId__CST'] = input.proId;
      let s = db.object('命名空间__product__CST'); //获取命名空间__product__CST这个Object的操作实例
      let id = s.insert(productData);
      if (id) {
```

```
        out.id = id;
      } else {
        error.name = "WOERROR";
        error.message = "Unable to create product!";
        throw error;
      }
    } catch (error) {
      console.error(error.name, error.message);
      context.setError(error.name, error.message);
    }
  }
  return out;
}
```

4. 单击，保存脚本，保存成功后单击，激活脚本。

步骤4 新建对象模型。

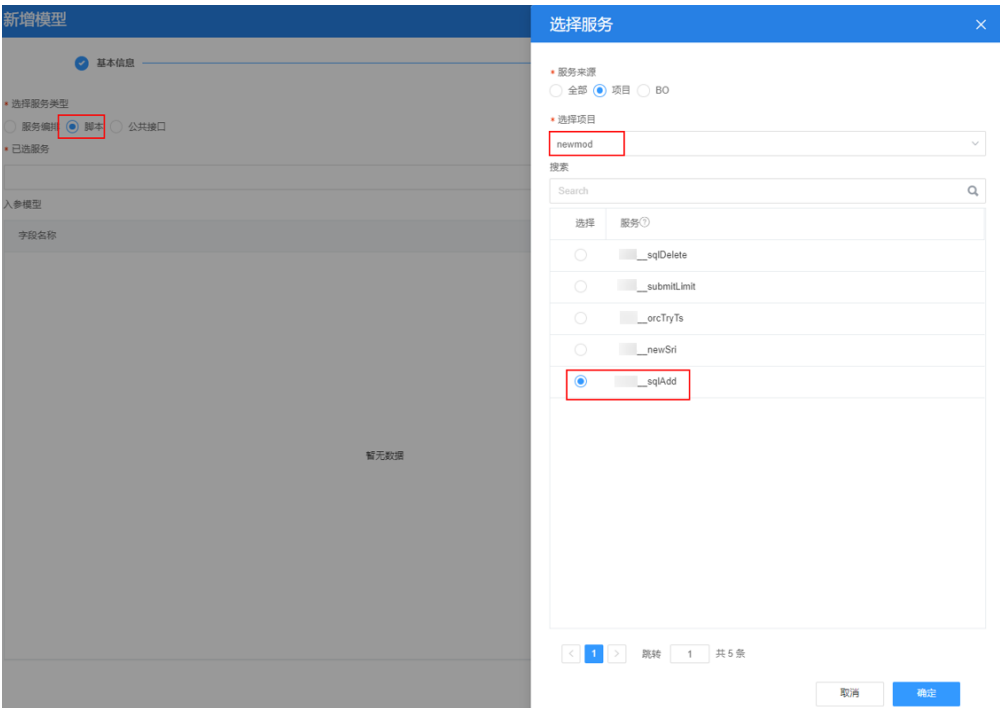
1. 在应用设计器中，选择“界面”，单击页面后的“+”，新建一个标准页面。
2. 在标准页面底部，单击“模型视图”。
3. 单击“新增模型”，输入模型名称（如sqlAdd）、“来源”选择“服务”，单击“下一步”。

图 3-26 新建模型



4. 选择步骤3中创建的脚本，单击“确定”。

图 3-27 选择脚本

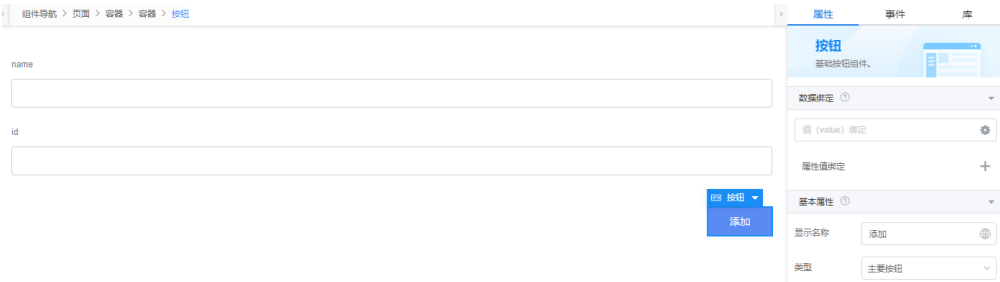


5. 单击“下一步”，再单击“确定”，完成模型的创建。

步骤5 返回设计视图页面，绑定关联模型。

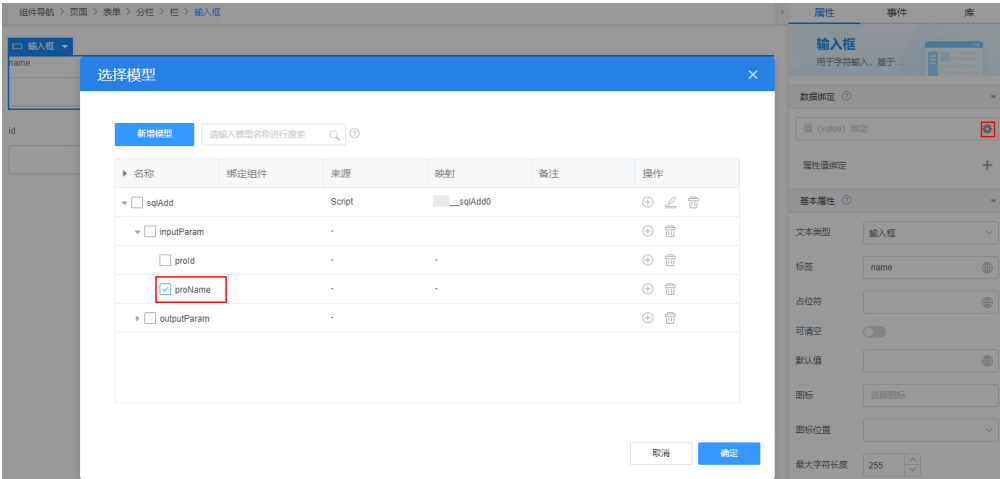
1. 在标准页面中，拖入2个输入框组件和1个按钮组件，将输入框标签修改为“name”和“id”，将按钮显示名称设置为“添加”。

图 3-28 页面最终设置效果



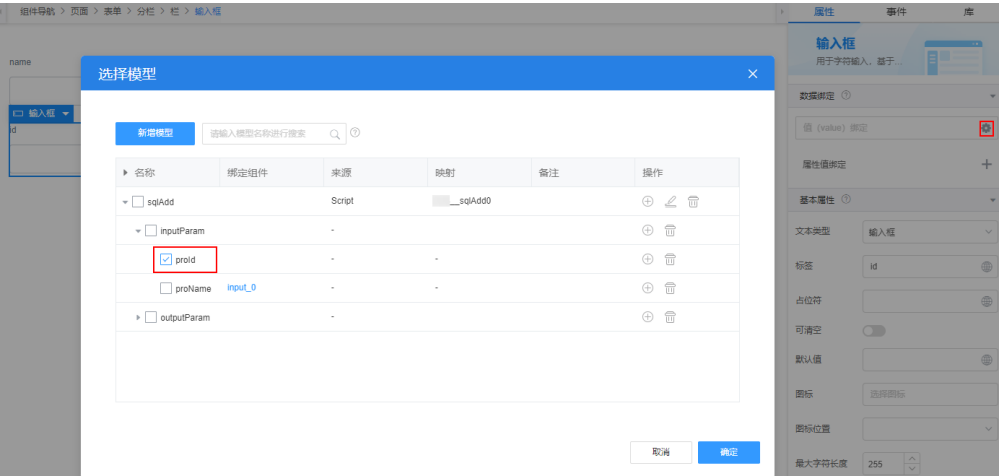
2. 选中name输入框，在“属性 > 数据绑定 > 值绑定”中，单击 。
3. 选中**步骤4**中创建的模型（proName），将输入框和脚本中的数据做绑定。

图 3-29 选择模型



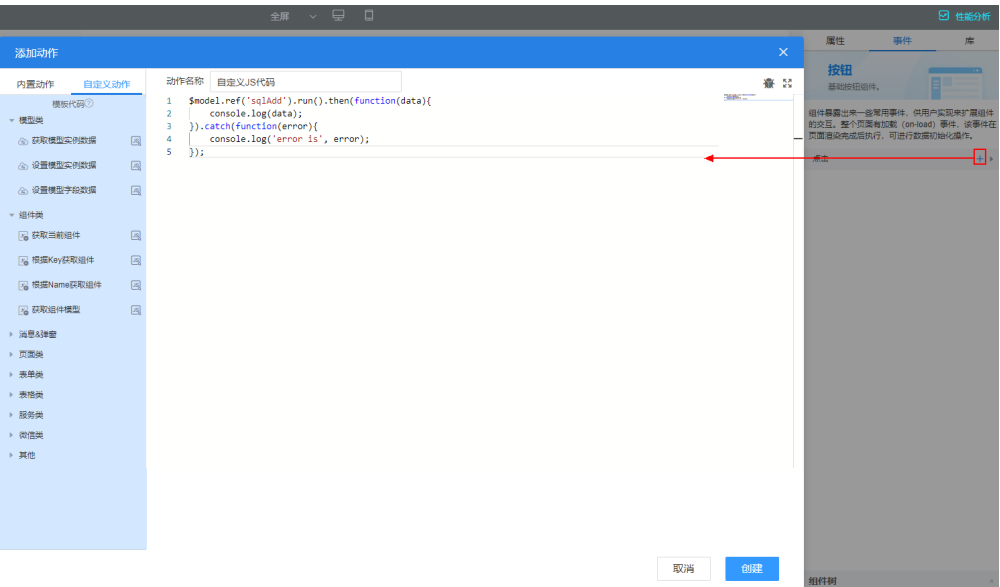
4. 按照上述操作，为id输入框绑定**步骤4**中创建的模型（prold）。

图 3-30 为 id 输入框绑定模型



- 步骤6 为添加按钮，添加事件。
1. 选中添加按钮组件，选择“事件”页签。
 2. 单击“点击”后的“+”，进入添加动作页面。
 3. 在自定义动作中，输入自定义代码，单击“创建”。

图 3-31 自定义动作



本示例中自定义的JS代码主要实现的功能为：单击提交按钮后，调用服务，服务调用脚本进行记录新增。

```
$model.ref('sqlAdd').run().then(function(data){
  console.log(data);
}).catch(function(error){
  console.log('error is', error);
});
```

- 步骤7 返回标准页面，单击，保存页面，保存成功后单击，预览效果。

----结束

功能实现方法（删除功能实现）

- 步骤1 创建一个低代码应用。
- 1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
 - 2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
 - 3. 在“应用”中，单击“新建低代码应用”或单击，进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。
 - 4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。
 - 5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 3-32 创建一个空白应用

新建 空白应用

基本信息



添加图标

标签

我的第一个应用

名称

A

分类

请选择

描述

请输入

高级设置

取消

新建

表 3-7 新建空白应用参数说明

参数	说明	示例
标签	新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。	我的第一个应用

文档版本 01 (2025-09-04)

版权所有 © 华为云计算技术有限公司

45

参数	说明	示例
名称	新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下： - 长度不能超过31个字符，包括前缀命名空间的长度。 - 名称前的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。	A

步骤2 创建一个删除脚本。

1.

在应用设计器中，选择“逻辑”，单击脚本后的“+”。
2.

新建一个空白的脚本（如sqlDelete），单击“添加”。

图 3-33 创建脚本 sqlDelete

新增脚本

创建一个新脚本

使用已有脚本

* 名称

sqlDelete

模板

空脚本

描述

取消

添加

3.

在脚本编辑器中，输入示例代码。

本示例脚本主要实现的功能为：根据页面输入的ID参数，使用接口，根据ID按条件删除一条数据记录。如果报错，则记录错误信息。示例中的“命名空间__product__CST”为**步骤2**中创建的对象。

```
//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
```

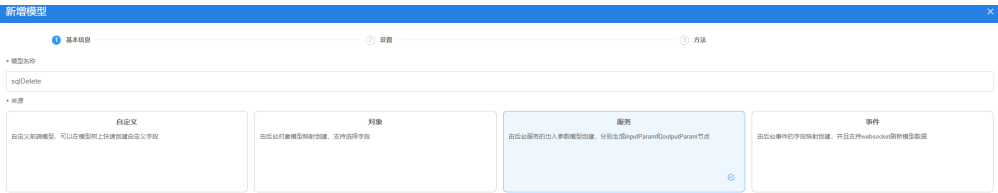
```
//定义入参结构
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'String', required: true, label: 'String' })
  id: string;
}
//定义出参结构，出参包含1个参数，workOrder的记录id
@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'String' })
  id: string;
}
//使用数据对象命名空间__product__CST
@useObject(['命名空间__product__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('命名空间__product__CST'); //获取命名空间__product__CST这个Object的操作实例
      //查询条件
      let condition = {
        "conjunction": "AND",
        "conditions": [{
          "field": "命名空间__prold__CST",
          "operator": "eq",
          "value": id
        }]
      };
      let isDeleted = s.deleteByCondition(condition);
      if (isDeleted) {
        out.id = id;
      } else {
        error.name = "WOERROR";
        error.message = "Failed to delete the work order!";
        throw error;
      }
    } catch (error) {
      console.error(error.name, error.message);
      context.setError(error.name, error.message);
    }
    return out;
  }
}
```

4. 单击，保存脚本，保存成功后单击，激活脚本。

步骤3 新建对象模型。

1. 在应用设计器中，选择“界面”，单击页面后的“+”。
2. 输入页面的标签和名称，单击“添加”，新建一个标准页面。
3. 在标准页面底部，单击“模型视图”。
4. 单击“新增模型”，输入模型名称（如sqlDelete）、“来源”选择“服务”，单击“下一步”。

图 3-34 新建模型



5. “选择服务类型” 设置为 “脚本”，在弹出的页面选择步骤2中创建的脚本，单击“确定”

图 3-35 选择脚本

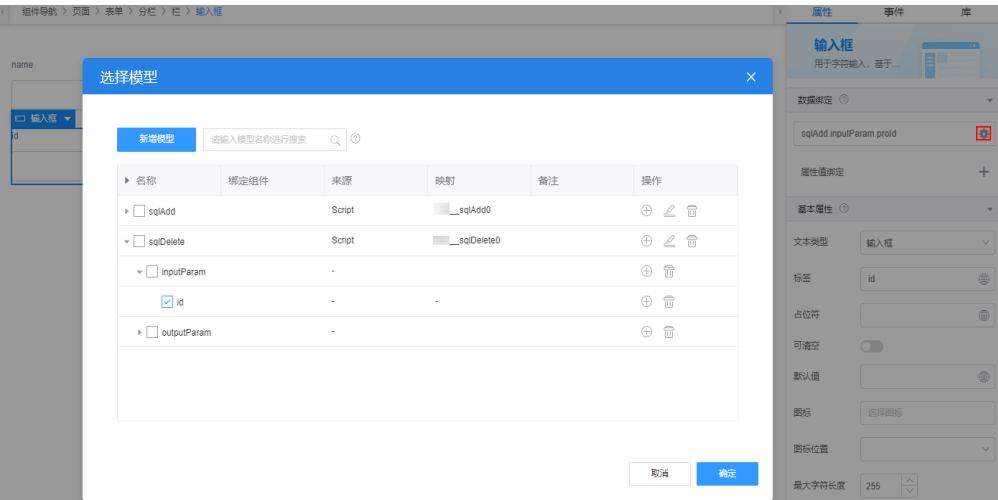


6. 单击“下一步”，再单击“确定”，完成模型的创建。

步骤4 返回设计视图页面，绑定关联模型。

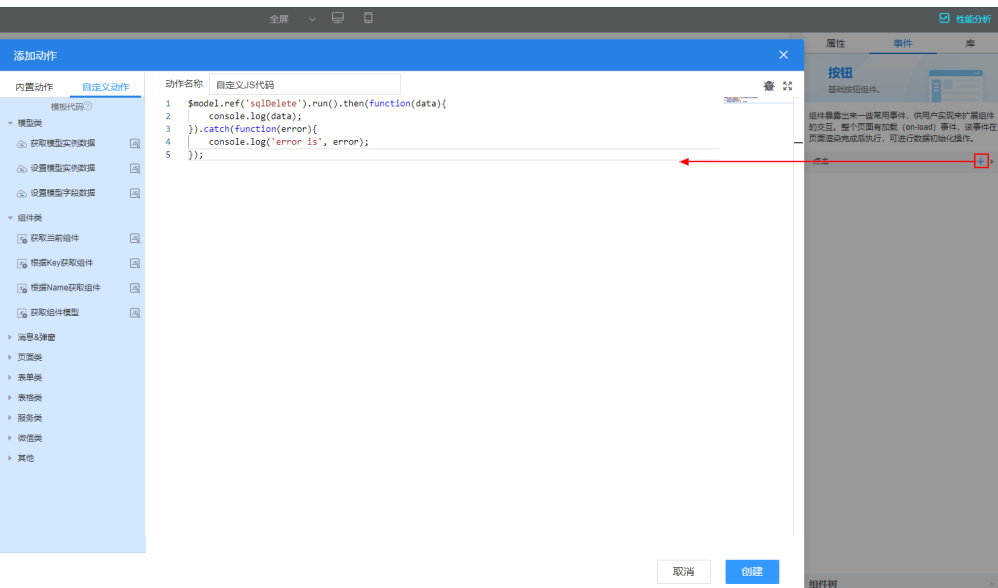
1. 在标准页面中，拖入一个输入框组件和一个按钮组件，将输入框组件的“标签”设置为“id”，按钮组件的“显示名称”设置为“删除”。
2. 选中输入框组件，在“属性 > 数据绑定 > 值绑定”中，单击 。
3. 选择步骤3中创建的模型，输入框数据绑定选择sqlDelete模型中的入参“id”，单击“确定”。

图 3-36 选择模型



- 步骤5** 为删除按钮，添加事件。
- 1. 选中删除按钮组件，选择“事件”页签。
 - 2. 单击“点击”后的“+”，进入添加动作页面。
 - 3. 在自定义动作中，输入自定义代码，单击“创建”。

图 3-37 自定义动作



示例中自定义的JS代码主要实现的功能为：利用JS调用后端服务，后端服务调用脚本完成删除功能。

```
$model.ref('sqlDelete').run().then(function(data){
  console.log(data);
}).catch(function(error){
  console.log('error is', error);
});
```

- 步骤6** 单击页面上方的，保存页面。

步骤7 保存成功后，单击页面上方的，预览效果。

----结束

4 模板专项

4.1 使用华为云 Astro 轻应用文件模板生成合同文档

应用场景

华为云Astro轻应用低代码平台提供了文档类模板创建能力，支持快速创建“Word文档”、“Excel文档”、“邮件”和“短信”四种类型的模板，以满足多样化的业务文档需求，提升文档的制作效率与规范性。

本实践以**Word文档**为例进行介绍，基于该功能开发者可以根据不同客户设置不同类型的打印模板。例如，用户在商品订单管理系统中，基于商品、价格等信息，自动生成了商品订单列表。在处理完自己的订单后，可以使用模板功能将订单信息同步到合同模板中，生成合同文档，用于签订线下合同。同样，在财务领域，可以使用文档模板来生成定制化的发票和收据。对于需要发送正式邀请的场合，文档模板可以用来创建和打印专业的商务信函。

方案优势

用户可以在服务编排中，通过拖、拉、拽的方式调用文档模板节点、配置模板的输入输出参数，用于生成具体的文档。同时可以将该服务编排包装成开放接口供第三方使用，也可以在华为云Astro轻应用的标准页面中直接调用供业务用户下载。

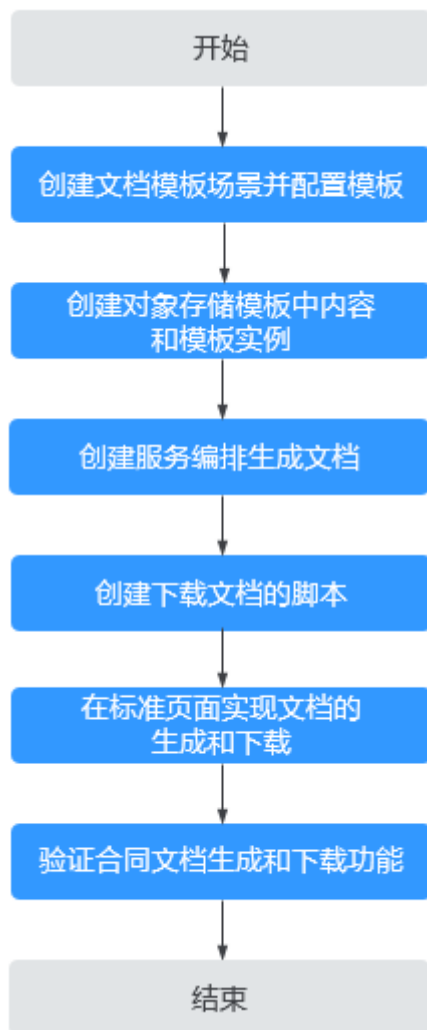
约束与限制

- 用户上传Word文档后（文档中包含使用\${参数}定义的文本类型的变量），只能预览该文档模板或生成具体的文档，不能直接在界面上对其进行修改。如果需要，只能在本地更改Word模板样式后重新上传。
- 生成的docx类型的文档模板，只支持文本类的变量替换，无法动态替换二维码、图片等内容。

操作流程

在华为云Astro轻应用中，通过文件模板生成具体合同文档的操作流程，如图4-1所示。

图 4-1 使用文件模板生成合同文档操作流程



步骤一：创建文档模板场景并配置模板

创建一个模板场景并在场景中添加一个合同模板。在模板场景的配置中，添加文档模板中需要替换的参数。模板场景是一个业务场景的集合，在模板场景中支持创建多个模板，模板之间共享数据结构。

步骤1 创建一个低代码应用。

1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
3. 在“应用”中，单击“新建低代码应用”或单击, 进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。
4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。
5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 4-2 创建一个空白应用

新建 空白应用

基本信息

添加图标

* 标签

我的第一个应用

* 名称

A

分类

请选择

描述

请输入

高级设置

取消

新建

表 4-1 新建空白应用参数说明

参数	说明	示例
标签	新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。	我的第一个应用
名称	新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下： - 长度不能超过31个字符，包括前缀命名空间的长度。 - 名称前的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。	A

- 步骤2 创建文件模板场景。
1. 在应用设计器的左侧导航栏中，选择“逻辑”，单击“更多 > 模板”。

图 4-3 单击模板



- 单击“添加模板场景”，设置模板场景的标签和名称，单击“确定”。

图 4-4 创建模板场景



表 4-2 新建模板场景参数说明

参数	说明	示例
标签	新建模板场景的名称，创建后可修改。 取值范围：1~80个字符。	文档模板
名称	新建模板场景在系统中的标识，创建后不支持修改。命名要求如下： - 长度不能超过64个字符，包括前缀命名空间的长度。 - 标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母，数字和下划线组成，且不能以下划线结尾。	docModel

步骤3 在“配置”页签，设置模板文件的存储位置，单击“保存”。

图 4-5 设置文件存储

文档模板

您可以配置多种模板，这些模板可以预置在应用中，用于发送通知或生成文档。

标签

名称

状态

文档模板

_docModel

禁用

启用

删除

基本

模板

数据

配置

生成的文件名

随机字符串

自定义名称

文件存储

连接器类型

连接器名称

连接器桶名

文件存储位置

OBS

_OBS_Connector

template-obs

docTemplate

邮件发送

连接器名称

请选择连接器

短信发送

连接器名称

请选择连接器

取消

保存

表 4-3 文件存储参数说明

参数	说明	示例
连接器类型	文件存储的连接器类型，当前仅支持“OBS”和“MINIO”两种。在华为云Astro轻应用中，通过创建OBS、MINIO连接器，可将数据存储到OBS和MINIO中，详细介绍请参见 对接OBS云对象存储实例 、 对接MINIO存储华为云Astro轻应用对象或资产 。	OBS
连接器名称	在华为云Astro轻应用中创建OBS或MINIO连接器的名称，可在“集成 > 连接器 > 连接器实例 > 存储 > OBS/MINIO”中查看。	命名空间 _OBS_Connector
连接器桶名	创建连接器时，配置的OBS或MINIO桶名称。	template-obs
文件存储位置	指定文件在OBS或MINIO桶中的存储路径。	docTemplate

步骤4 在“模板”页签，单击“添加”，创建合同模板。

图 4-6 新建合同模板

新建模板

* 标签

合同文档模板

* 名称

contractModel

描述

模板类型

Word文档

Excel文档

邮件

短信

取消

确定

表 4-4 新建模板参数说明

参数	说明	示例
标签	新建模板的名称，创建后可修改。 取值范围：1~80个字符。	合同文档模板
名称	新建模板在系统中的标识，创建后不支持修改。命名要求如下： - 长度不能超过63个字符，包括前缀命名空间的长度。 标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母，数字和下划线组成，且不能以下划线结尾。	contractModel

参数	说明	示例
模板类型	新建文档模板的类型。 - Word文档：支持用户上传Word格式的文档模板，文档的大小建议小于10M。 - Excel文档：支持用户上传Excel类型的文档模板。上传的Excel文档不能超过10个Sheet页，每个Sheet页中列不能超过200个，行不能超过1000个，单元格内字符串的长度不超过500个字符。 - 邮件：支持用户在模板编辑界面编写邮件模板。 - 短信：支持用户在模板编辑界面编写短信模板。	Word文档

步骤5 在“数据”页签，为合同文档模板添加表4-5中参数。

此处添加的模板参数，对应文档模板中需要替换的内容。

图 4-7 为合同模板添加公司名称参数



表 4-5 待添加参数

名称	唯一标识	数据类型
公司名称	companyName	文本
合同金额	amount	数字
订单数目	orderNum	数字
合同签订人	person	文本

名称	唯一标识	数据类型
合同日期	date	日期
合同名称	contractName	文本
乙方公司名称	otherCompanyName	文本

步骤6 上传文档模板。

1. 在“模板”页签，单击步骤4中创建模板后的，进入合同文档模板页面。

图 4-8 进入模板编辑页面



2. 单击“点击上传”，选择本地待上传的Word文档，查看上传后效果。
上传操作执行成功后，在步骤3中配置的OBS桶中可查看到已上传的Word文档，如图4-11所示。

图 4-9 上传 Word 文档

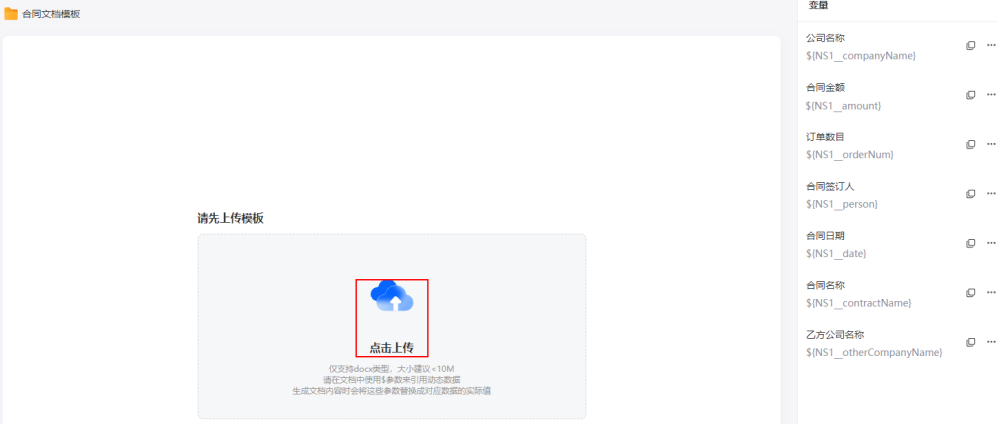


图 4-10 查看文档上传后效果



图 4-11 在 OBS 桶中查看文档是否上传



本实践中待上传的Word文档内容如下，实际使用时请根据自身的业务需求进行上传。

```

    ${命名空间__contractName}

第一条合同目的
本合同旨在规定${命名空间__companyName} 向乙方购买商品/服务的具体条款和条件。

第二条商品/服务描述
商品/服务的详细描述，包括但不限于型号、规格、数量、单价等。
订单数目：${命名空间__orderNum}

第三条价格条款
商品/服务的总额为 ${命名空间__amount} 元。

第四条质量保证
乙方保证所提供的商品/服务符合约定的质量标准

第五条违约责任
如一方违反合同条款，违约方应赔偿对方因此遭受的所有损失。

第六条生效条件
本合同自双方授权代表签字盖章之日起生效。

甲方：${命名空间__companyName}
乙方：${命名空间__otherCompanyName}

合同签订人：${命名空间__person}
日期：${命名空间__date}

```

3. 单击页面右上方的“预览”，输入参数，预览效果。
- 例如，在“输入参数”中输入如下模板参数，查看合同中对应参数是否被替换。
- {
 "命名空间__contractName": "买卖合同",

```
"命名空间__companyName": "xx有限公司"
}
```

图 4-12 合同中参数已被替换

买卖合同

第一条 合同目的
本合同旨在规定xx有限公司向乙方购买商品/服务的具体条款和条件。

第二条 商品/服务描述
商品/服务的详细描述，包括但不限于型号、规格、数量、单价等。
订单数目：\${NS1__orderNum}

第三条 价格条款
商品/服务的总额为 \${NS1__amount} 元。

第四条 质量保证
乙方保证所提供的商品/服务符合约定的质量标准

第五条 违约责任
如一方违反合同条款，违约方应赔偿对方因此遭受的所有损失。

第六条 生效条件
本合同自双方授权代表签字盖章之日起生效。

甲方：xx有限公司
乙方：\${NS1__otherCompanyName}

合同签订人：\${NS1__person}
日期：\${NS1__date}

4. 预览符合预期后，单击页面右上方的“启用”，启用文档模板。

步骤7 返回文档模板场景中，单击“启用”，启用模板场景。

图 4-13 启用模板场景



----结束

步骤二：创建对象存储模板中内容和模板实例

创建一个对象并在对象中添加字段，用于存储文档模板中的内容和后续生成的模板实例。

步骤1 在应用设计器的左侧导航栏中，选择“数据”，单击对象中的“+”。

步骤2 设置对象的名称和唯一标识，单击“确定”。

图 4-14 创建对象 docObject

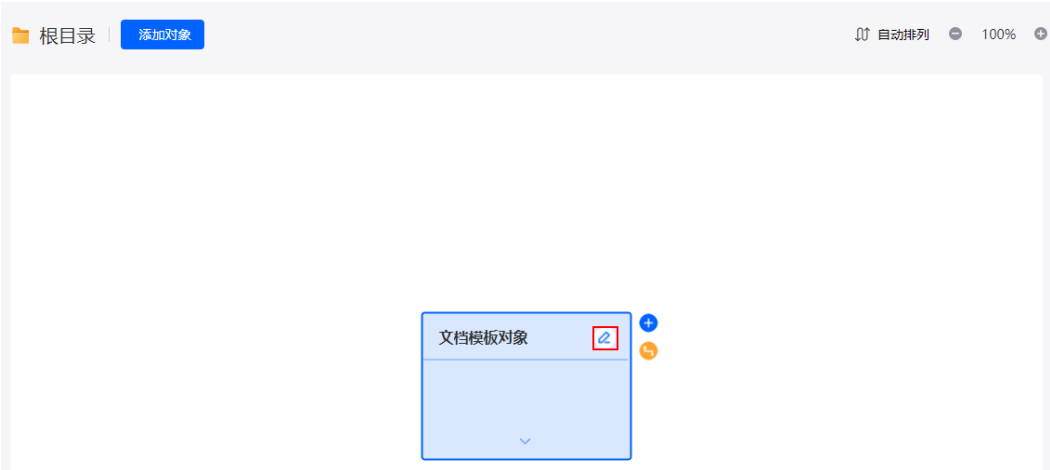


表 4-6 新建 docObject 对象参数说明

参数	说明	示例
对象名称	新建对象的名称，创建后可修改。 取值范围：1~80个字符。	文档模板对象
唯一标识	新建对象在系统中的标识，创建后不支持修改。命名要求如下： - 长度不能超过63个字符，包括前缀命名空间的长度。 标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母，数字和下划线组成，且不能以下划线结尾。	docObject

步骤3 在已创建的对象中，单击，进入对象详情页面。

图 4-15 选择编辑按钮



步骤4 在“字段”页签，单击“添加”，为对象添加companyName字段。

图 4-16 添加 companyName 字段

添加字段

*

显示名称

公司名称

*

唯一标识

companyName

*

字段类型

文本

...

*

数据长度

64

描述

取消

确认

表 4-7 添加 companyName 字段参数说明

参数	说明	示例
显示名称	新建字段的名称，创建后可修改。 取值范围：1~63个字符。	公司名称

参数	说明	示例
唯一标识	新建字段在系统中的标识，创建后不支持修改。命名要求如下： - 长度不能超过63个字符，包括前缀命名空间的长度。 - 必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。	companyName
字段类型	单击***，在弹出的页面中，根据页面提供的参数解释，选择新建字段所属的类型。	文本
数据长度	允许输入字段的长度。	64

步骤5 按照上述操作，为对象继续添加表4-8中字段。

图 4-17 查看对象中字段



表 4-8 待添加对象字段

名称	唯一标识	数据类型
公司名称（已添加）	companyName	文本
合同金额	amount	数字

名称	唯一标识	数据类型
订单数目	orderNum	数字
合同签订人	person	文本
合同日期	date	日期
合同名称	contractName	文本
乙方公司名称	otherCompanyName	文本
合同模板实例	modedoc	文本（数据长度设置为255）

----结束

步骤三：创建服务编排生成文档

创建一个服务编排，添加“调用模板”和“记录创建”两个图元，用于根据合同中参数创建具体的文档。

- 步骤1 在应用设计器的左侧导航栏中，选择“逻辑”，单击编排后的“+”。
- 步骤2 设置服务编排的标签和名称，单击“添加”。

图 4-18 新建服务编排

添加服务编排

● 创建一个新的服务编排

☐ 使用已有的服务编排

* 标签

根据合同参数创建具体文档实例

* 名称

docFlow

* 类型

Autolaunched Flow

描述

请输入

取消

添加

文档版本 01 (2025-09-04)

版权所有 © 华为云计算技术有限公司

65

表 4-9 新建服务编排参数说明

参数	说明	示例
标签	服务编排的标签名，用于在界面展示，创建后可修改。 取值范围：1~64个字符。	根据合同模板创建具体文档实例
名称	服务编排在系统中的唯一标识，创建后不支持修改。命名要求如下： - 长度不能超过64个字符，包括前缀命名空间的长度。 标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。 - 名称必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不能以下划线结尾。	docFlow

步骤3 创建全局上下文变量。

1. 在服务编排设计页面，选中开始节点，单击 。
2. 在全局上下文中，单击变量后的 ，新建变量variable0。
3. 单击variable0变量后的 ，选择“设置”。
4. 将变量的“名称”设置为“companyName”，单击“保存”。

图 4-19 新建 companyName 变量

变量

* 名称

companyName

变量名是用于变量在流程中引用的唯一标识。修改变量名不会改变图元中的引用，可能导致流程不可用。

* 数据类型

文本

默认值

请输入常量或从列表中选择

描述

请输入

是否为数组

外部使用

取消

保存

5. 按照上述操作，创建表4-10中的变量。

图 4-20 查看已创建的变量



表 4-10 需要创建的全局上下文变量

名称	数据类型
companyName（已创建）	文本
amount	数字
orderNum	数字
person	文本
date	日期
contractName	文本
otherCompanyName	文本
modedoc	文本

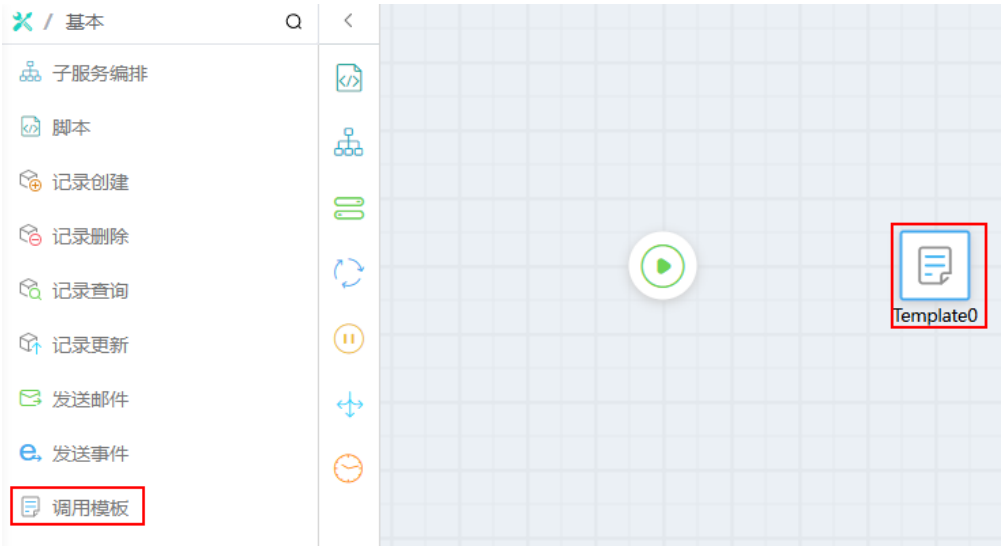
6. 选中开始节点，设置节点的输入参数和输出参数。

图 4-21 设置入参和出参



- 步骤4 添加调用模板节点。
1. 在“基本”图元中，拖拽“调用模板”图元到开始图元后。

图 4-22 拖拽调用模板图元到画布中



2. 选中调用模板图元，单击，设置调用模板。

图 4-23 设置调用模板图元

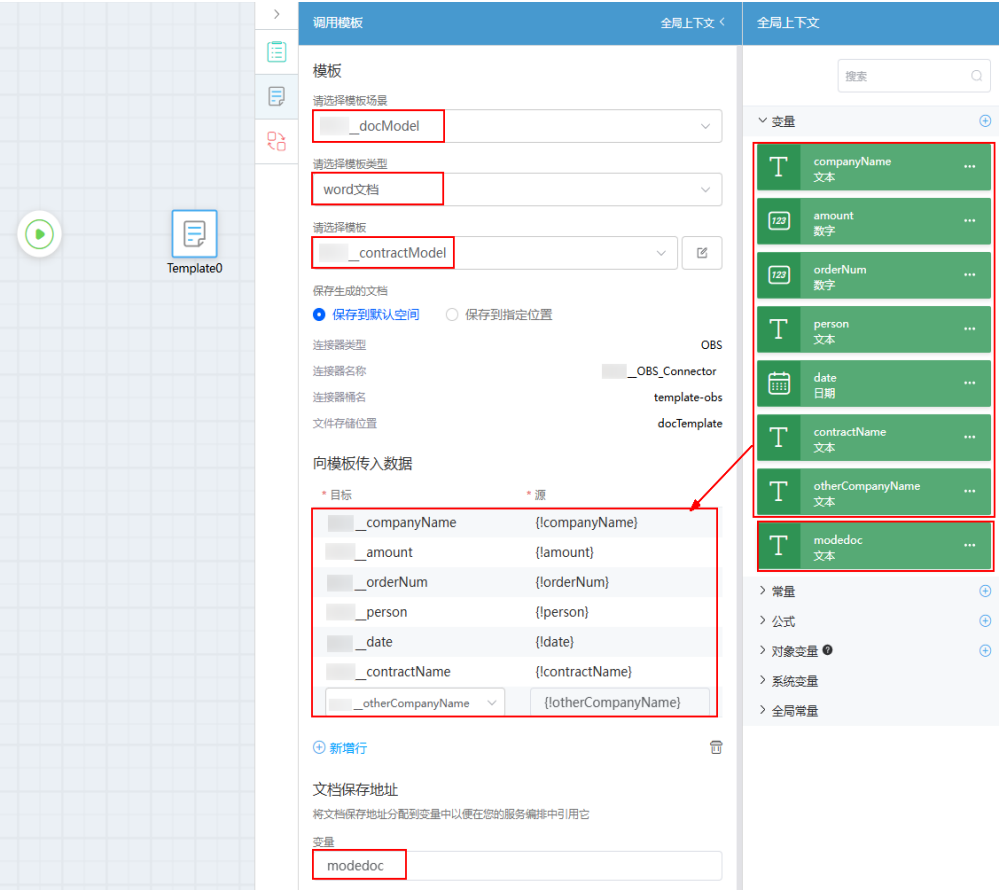


表 4-11 调用模板图元设置说明

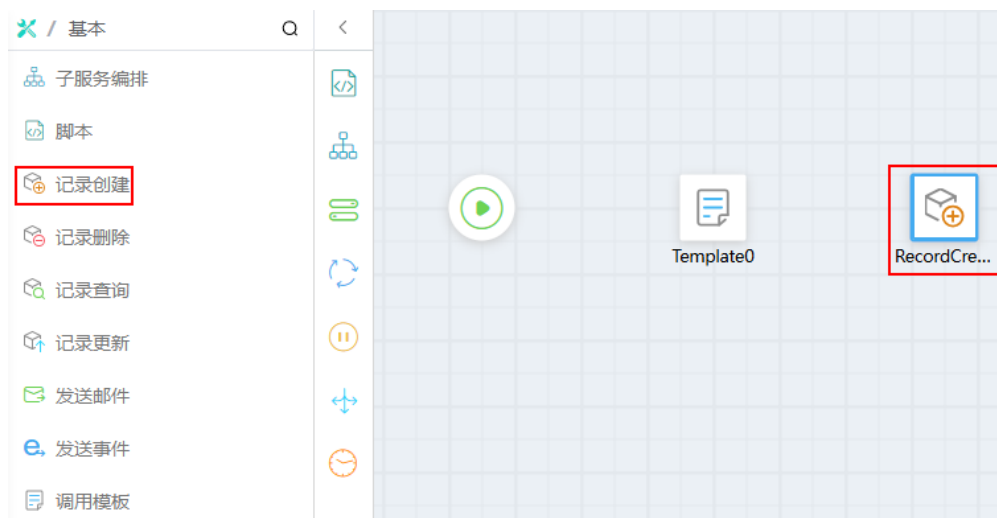
参数	说明	示例
请选择模板场景	选择调用模板图元关联的模板场景，即步骤2中创建的。	命名空间_docModel
请选择模板类型	选择待调用模板的类型。	Word文档
请选择文档模板	选择模板场景中创建的文档模板，即步骤4中创建的。	命名空间__contractModel
连接器类型	根据选择的文档模板自动进行关联。	OBS
连接器名称	根据选择的文档模板自动进行关联。	命名空间__OBS_Connector
连接器桶名	根据选择的文档模板自动进行关联。	template-obs
文件存储位置	根据选择的文档模板自动进行关联。	docTemplate

参数	说明	示例
向模板传入数据	向模板中传入数据，将入参变量一次赋值给对应的模板参数。	步骤3 中创建的变量
文档保存地址	将出参modelDoc变量放入文档保存地址，用于存放生成的文档名。	modedoc

步骤5 添加创建记录节点。

1. 在“基本”图元中，拖拽“记录创建”图元到“调用模板”图元后。

图 4-24 添加记录创建图元



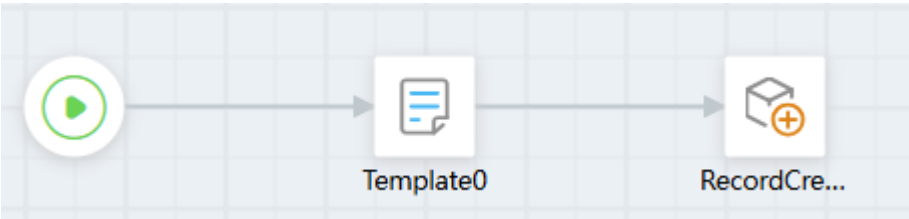
2. 选中记录创建图元，单击 ，将入参、出参变量都存入[步骤二：创建对象存储模板中内容和模板实例](#)中创建的对象中。

图 4-25 记录创建设置



步骤6 连接图元指定逻辑关系，即从开始图元连向调用模板图元，从调用模板图元连向记录创建图元。

图 4-26 指定图元逻辑关系



步骤7 单击页面上方的 ，保存服务编排。

步骤8 单击 ，进入服务编排调试页面。

在输入参数中，输入如下内容，并单击“运行”。

```
{
  "companyName": "A公司",
  "amount": "10",
  "orderNum": "2",
  "person": "张三",
  "date": "2024-11-05",
  "contractName": "新采购合同",
}
```



```
"otherCompanyName": "B公司"
}
```

执行成功后，提示如下信息。登录OBS服务控制台，在[步骤3](#)中配置的路径下会生成[图4-27](#)中的合同文档。将合同文档下载到本地，查看合同中内容，如[图4-29](#)所示，可发现合同中的内容已被替换。

图 4-27 服务编排执行成功

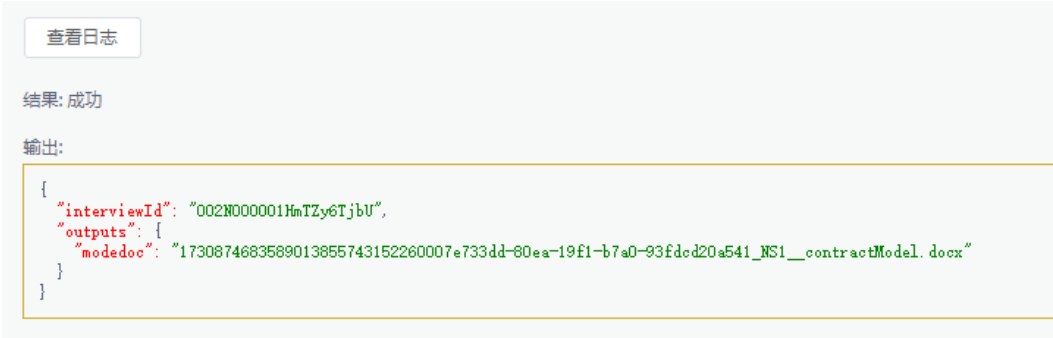


图 4-28 查看新生成的合同文档



图 4-29 查看合同内容

↵

新采购合同↵

↵

第一条 合同目的↵
本合同旨在规定 A公司 向乙方购买商品/服务的具体条款和条件。↵

↵

第二条 商品/服务描述↵
商品/服务的详细描述，包括但不限于型号、规格、数量、单价等。↵
订单数目：2↵

↵

第三条 价格条款↵
商品/服务的总额为 10 元。↵

↵

第四条 质量保证↵
乙方保证所提供的商品/服务符合约定的质量标准↵

↵

第五条 违约责任↵
如一方违反合同条款，违约方应赔偿对方因此遭受的所有损失。↵

↵

第六条 生效条件↵
本合同自双方授权代表签字盖章之日起生效。↵

↵

甲方：A公司↵
乙方：B公司↵

↵

合同签订人：张三↵
日期：2024-11-05↵

步骤9 单击页面上方的，启用服务编排。

----结束

步骤四：创建下载文档的脚本

创建一个脚本，用于根据文档名下载OBS桶中生成的合同文档。

步骤1 在应用设计器的左侧导航栏中，选择“逻辑”，单击脚本后的“+”。

步骤2 新建一个空白的脚本，名称设置为“docScript”，单击“添加”。

图 4-30 创建脚本 docScript

新增脚本

创建一个新脚本

使用已有脚本

* 名称

docScript

模板

空脚本

描述

取消

添加

步骤3 在脚本编辑器中，输入示例代码。

本示例代码主要用于下载文档，示例中的“命名空间_OBS_Connector”为步骤3中配置的连接器名称，“docTemplate”为步骤3中配置的文件存储位置。

```
import * as context from 'context';//导入上下文相关的标准库
import * as objectstorage from 'objectstorage';

//定义入参结构
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'String', required: true, label: 'String' })
  docName: string;
}
//定义出参结构
@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'Any' })
  buf: any;
}

@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 {
      // OBS桶路径，和模板配置中的一致
      let path = "docTemplate/";
      // 调用连接器下载，命名空间_OBS_Connector为调用连接器名称
      let obsCli = objectstorage.newClient(objectstorage.StoreType.OBS, "命名空间_OBS_Connector");
      let data = obsCli.getObject(path + input.docName);
      out.buf = data;
    } catch (error) {
      console.error(error.name, error.message);
      context.setError(error.name, error.message);
    }
  }
}
```

```
}
    return out;
}
}
```

步骤4 单击，保存脚本。

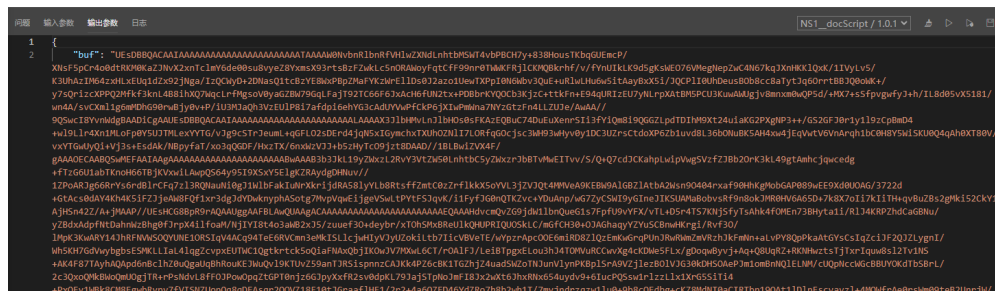
步骤5 运行测试脚本。

1. 单击编辑器上方的，执行脚本。
2. 设置输入参数，单击测试窗口右上角的，查看返回消息。

```
{
  "docName": "1730874683589013855743152260007e733dd-80ea-19f1-b7a0-93fdcd20a541_命名空间_contractModel.docx"
}
```

其中，“1730874683589013855743152260007e733dd-80ea-19f1-b7a0-93fdcd20a541_命名空间_contractModel.docx”为**步骤三：创建服务编排生成文档**中生成的文档，即通过脚本根据文档名下载OBS桶中的实际合同文档。

图 4-31 查看输出参数



步骤6 脚本测试完成后，单击页面上方的，启用该脚本。

----结束

步骤 5：通过标准页面调用服务编排和脚本实现文档的生成和下载

设计一个标准页面，通过调用服务编排和脚本，在前端页面实现合同文档的生成和下载等功能。

步骤1 新建一个标准页面。

1. 在应用设计器中，选择“界面”，单击页面后的“+”，新建一个标准页面。
2. 设置标准页面的标签和名称，单击“添加”。

图 4-32 添加标准页面

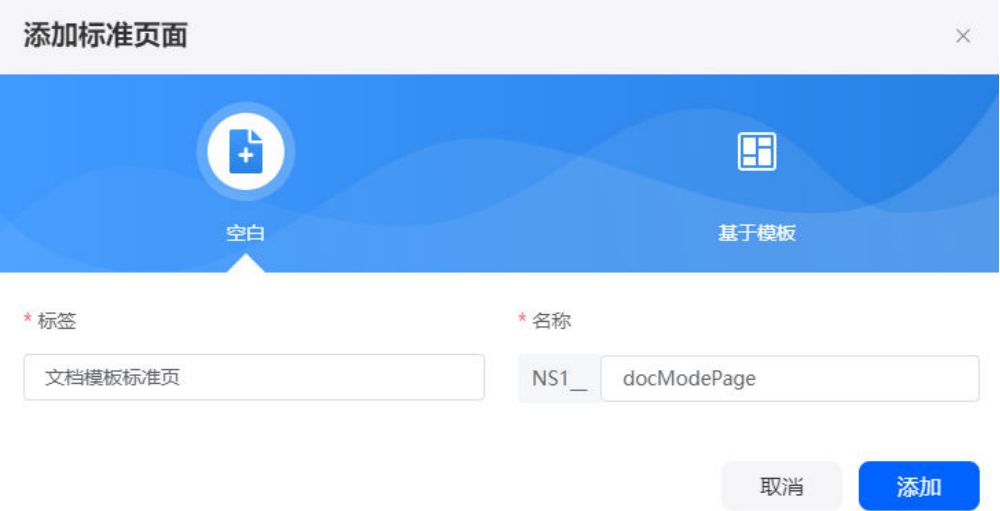


表 4-12 新建标准页面参数说明

参数	说明	示例
标签	输入标准页面的标签名，用于在页面显示，创建后可修改。 取值范围：1~64个字符。	文档模板标准页
名称	输入标准页面的名称，名称是标准页面在系统中的唯一标识，创建后不可修改。命名要求如下： - 长度不能超过64个字符，包括前缀命名空间的长度。 - 标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不能以下划线结尾。	docModePge

- 步骤2** 新增对象模型。
- 在标准页面底部，单击“模型视图”，将页面从设计视图切换到模型视图。
 - 单击“新增模型”，输入模型名称（flowDemo）、“来源”选择“服务”，单击“下一步”。

图 4-33 新增服务编排对象模型



3. 选择**步骤三：创建服务编排生成文档**中创建的服务编排，单击“下一步”，再单击“确定”，完成模型的创建。

图 4-34 选择目标服务编排



4. 再次单击“新增模型”，输入模型名称（scriptDemo）、“来源”选择“服务”，单击“下一步”。

图 4-35 新增脚本对象模型



5. 选择**步骤四：创建下载文档的脚本**中创建的脚本，单击“下一步”，再单击“确定”，完成模型的创建。

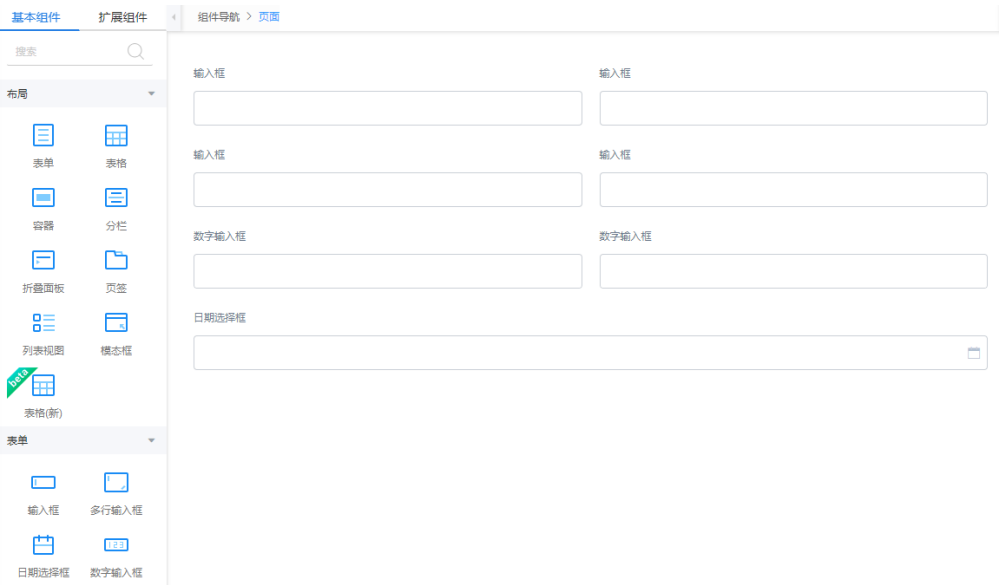
图 4-36 选择目标脚本



步骤3 设计一个标准页面，用于生成和下载合同文档。

1. 在标准页面底部，单击“设计视图”，从模型视图切换到设计视图。
2. 从基本组件中，拖拽输入框（4个）、数字输入框（2个）和日期选择器到标准页面的画布中，布局如**图4-37**。

图 4-37 拖拽组件到画布



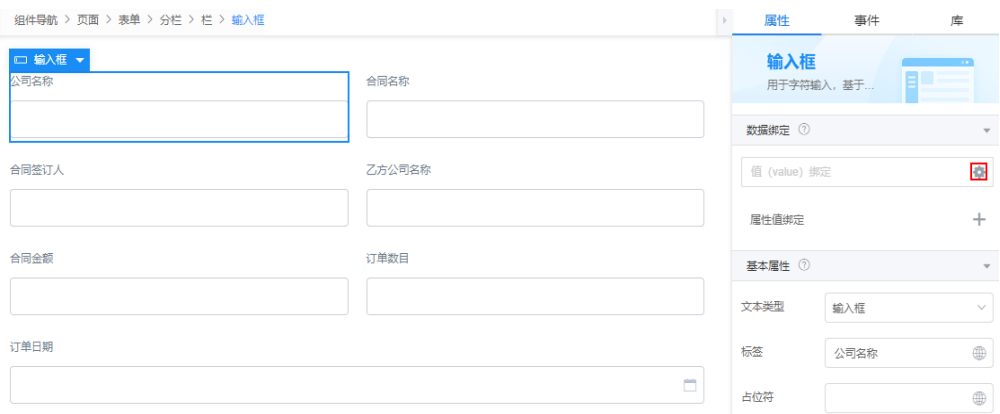
3. 选中组件，修改组件的标签，修改后效果如图4-38所示。

图 4-38 修改组件标签



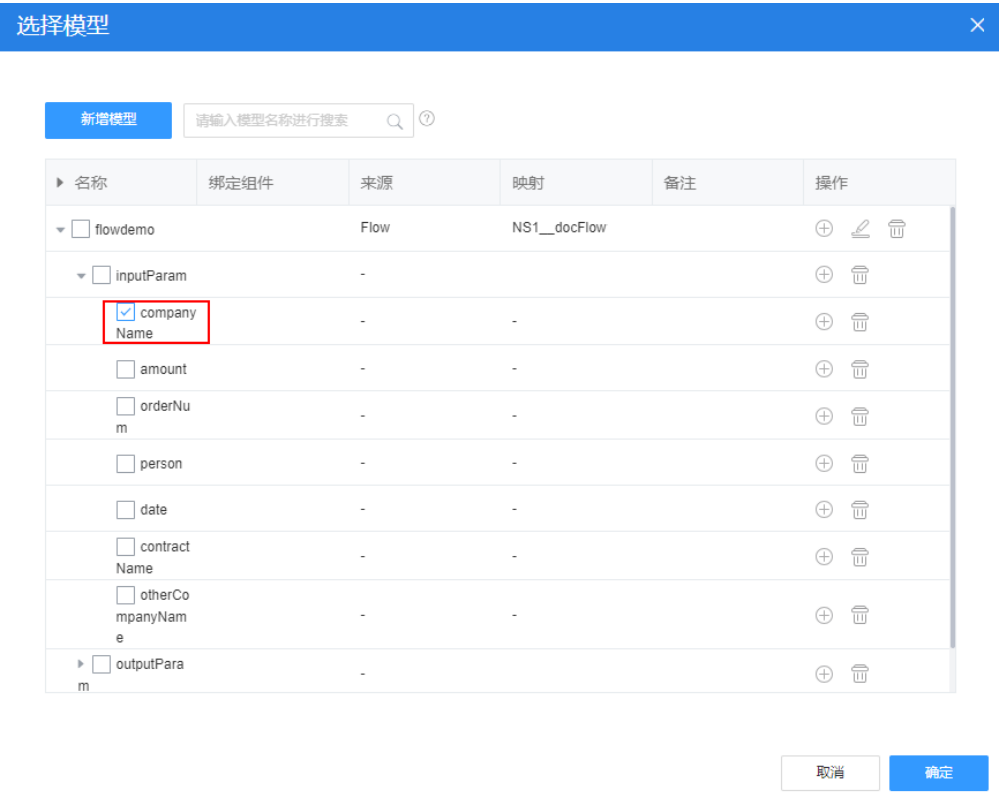
4. 选中公司名称输入框，在“属性 > 数据绑定”中，单击“值（value）绑定”后的 。

图 4-39 选择值绑定



5. 选择步骤2.2模型中的“companyName”，单击“确定”。

图 4-40 绑定 companyName

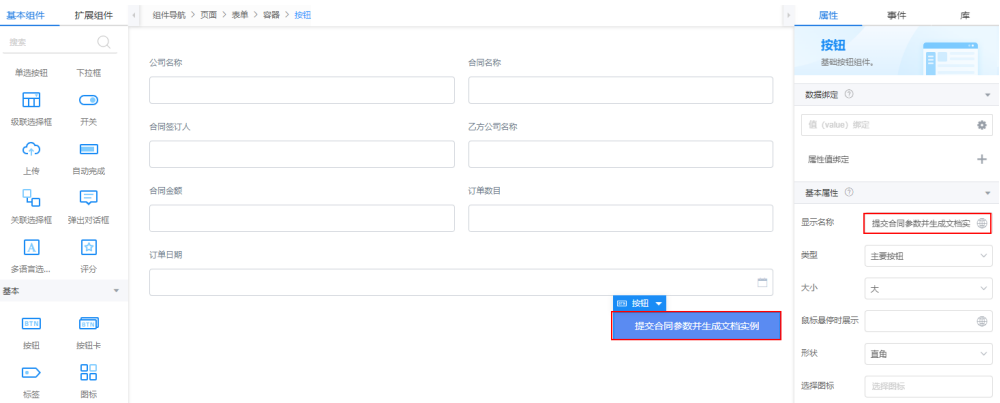


6. 按照上述操作，分别为其他组件绑定模型。
- “合同名称”绑定“flowdemo.inputParam.contractName”，“合同签订人”绑定“flowdemo.inputParam.person”，“乙方公司名称”绑定“flowdemo.inputParam.otherCompanyName”，“合同金额”绑定“flowdemo.inputParam.amount”，“订单数目”绑定“flowdemo.inputParam.orderNum”，“订单日期”绑定“flowdemo.inputParam.date”。

步骤4 为标准页面添加一个按钮组件。

1. 从“基本组件 > 基本”中，拖拽一个按钮组件到订单日期组件下方，并将按钮的“显示名称”修改为“提交合同参数并生成文档实例”。

图 4-41 添加按钮组件



2. 选中按钮组件，在“事件”页签中，单击“点击”后的 +，进入添加动作页面。
3. 在自定义动作中，输入如下示例代码，单击“创建”。

本示例代码实现的功能：根据服务编排模型中设置的合同参数，生成对应的合同文档。其中，“flowDemo”为[步骤2.2](#)中创建的服务编排模型。

```
$model.ref('flowDemo').run().then(function(data){
  console.log(data);
  context.$message.success('Submitted successfully. ');
}).catch(function(error){
  context.$message.error('Submission failed:' + error.resMsg);
});
```

步骤5 为标准页面添加一个表格组件。

1. 从“基本组件 > 布局”中，拖拽一个表格组件到按钮组件下方。

图 4-42 添加表格组件



2. 选中表格组件，在“属性 > 数据绑定”中，单击“值（value）绑定”后的 ⚙️。
3. 单击“新建模型”，输入模型名称（obj）、“来源”选择“对象”，单击“下一步”。

图 4-43 设置模型基本信息

新增模型

1 基本信息

2 设置

3 方法

模型名称

obj

来源

自定义

自定义前端模型，可以在模型树上快速创建自定义字段

对象

由后台对象模型映射创建，支持选择字段

服务

由后台服务的出入参数模型创建，分别生成inputParam和outputParam节点

事件

由后台事件的字段映射创建，并且支持websocket刷新模型数据

取消

下一步

4. 选择**步骤二：创建对象存储模板中内容和模板实例**中创建的对象和字段，单击“下一步”，再单击“确定”。

图 4-44 模型设置

新增模型

1 基本信息

2 设置

3 方法

选择对象

☐ 包含其他应用

文档模板对象 (__docObject__CST)

选择字段

请输入字段名称进行检索

☐	字段名称	字段标签
☐	name	名称
☒	__companyName__CST	公司名称
☒	__amount__CST	合同金额
☒	__orderNum__CST	订单数目
☒	__person__CST	合同签订人
☒	__date__CST	合同日期
☒	__contractName__CST	合同名称
☒	__otherCompanyName__CST	乙方公司名称
☒	__modedoc__CST	合同模板实例
☐	createdBy	创建人

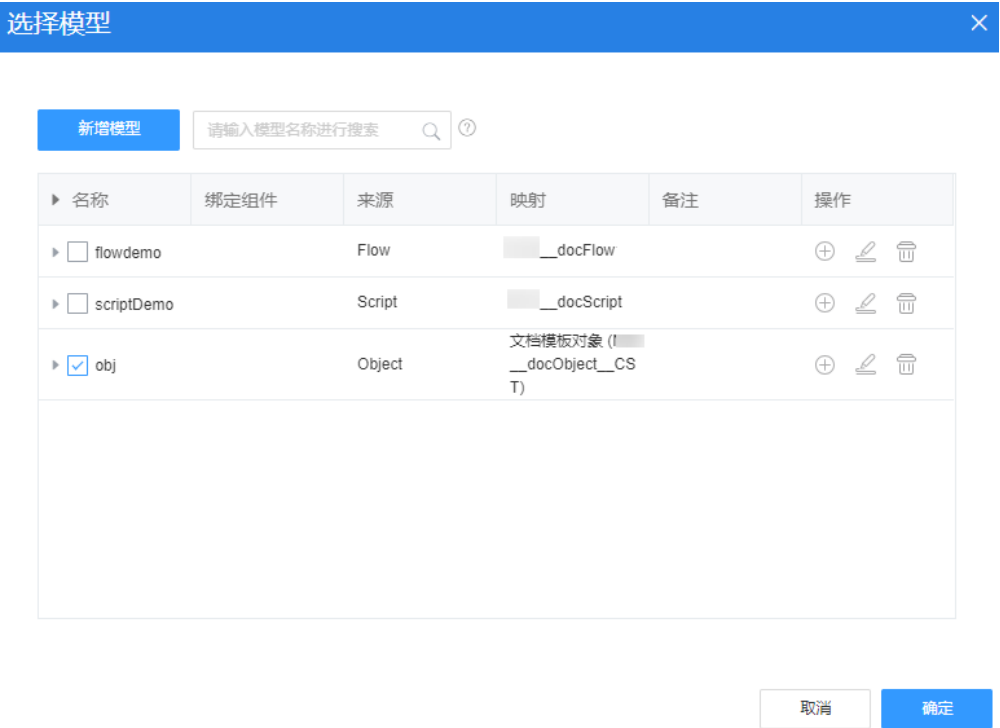
取消

上一步

下一步

5. 在选择模型中，选中创建的模型，将对象模型绑定到表格上。

图 4-45 为表格绑定模型



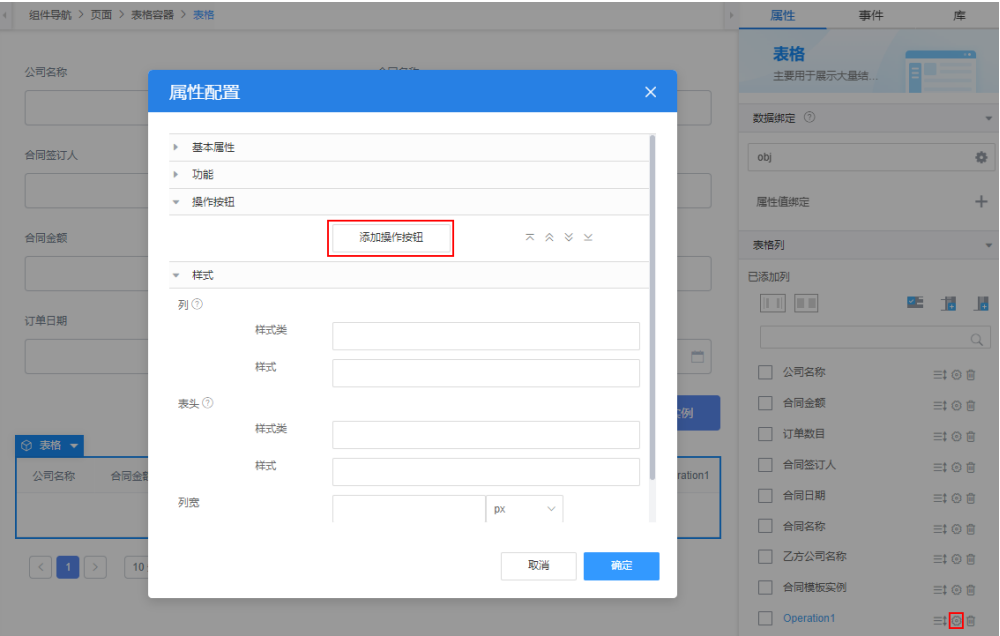
6. 选中表格组件，为表格添加一个操作列。

图 4-46 为表格添加操作列



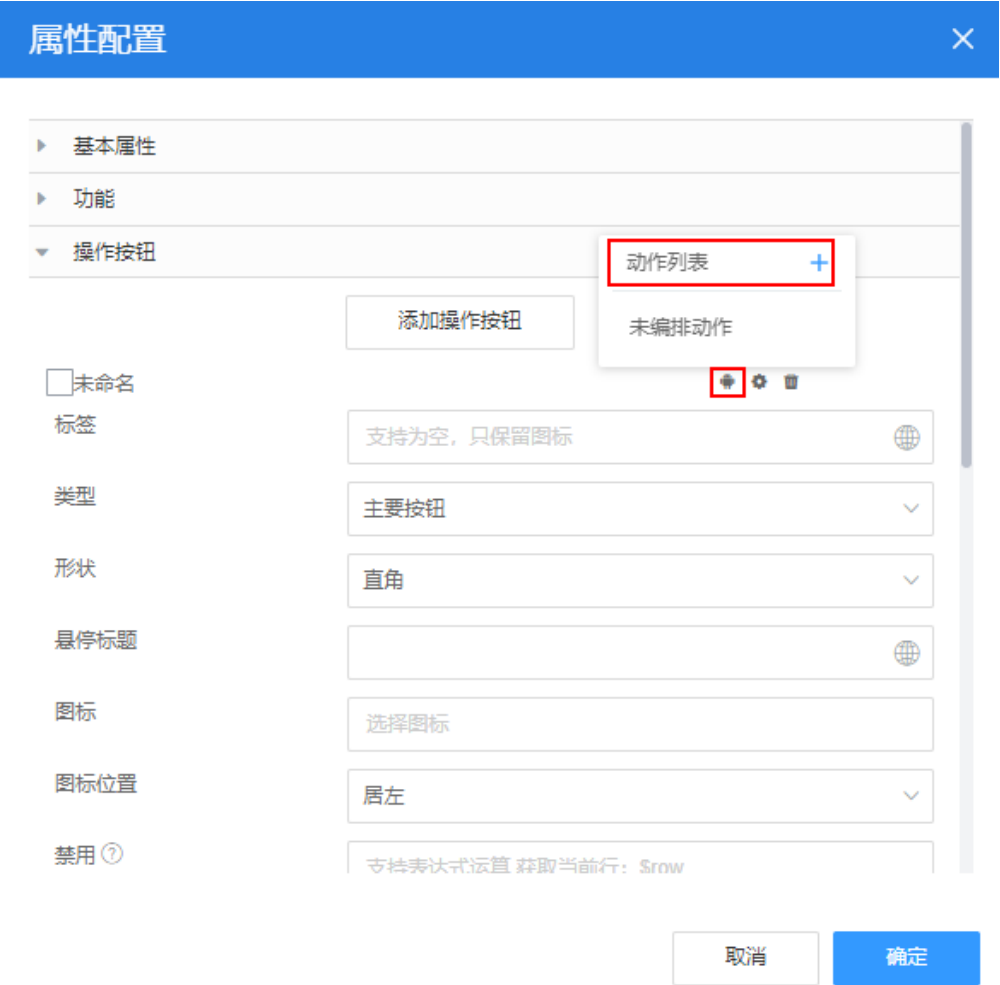
7. 单击新建操作列后的，在操作按钮中，单击“添加操作按钮”。

图 4-47 添加操作按钮



8. 单击，单击“动作列表”后的，为按钮添加事件。

图 4-48 为按钮添加动作



9. 在自定义动作中，输入如下示例代码，单击“创建”。
- 本示例代码主要实现的功能：根据合同模板实例的字段名，从OBS中下载服务编排生成的文档。其中，“命名空间_modedoc_CST”为合同模板实例的字段名，“scriptDemo”为步骤2.4中创建的脚本对象模型。

```
// base64解码函数
function base64DecodeToBinary(base64String) {
  // 解码Base64字符串
  const binaryString = atob(base64String);
  // 将解码后的字符串转换为二进制数据
  const len = binaryString.length;
  const bytes = new Uint8Array(len);
  for (let i = 0; i < len; i++) {
    bytes[i] = binaryString.charCodeAt(i);
  }
  // 将Uint8Array转换为Blob
  const blob = new Blob([bytes]);
  return blob;
}
// 获取表格中的行数据，命名空间_modedoc_CST为表格中合同模板实例的字段名
var rowData = context.$component.current.$attrs.row;
const docName = rowData.命名空间_modedoc_CST;
// 设置脚本入参
let originData = $model.ref('scriptDemo').getData();
originData.inputParam.docName = docName;
$model.ref('scriptDemo').setData(originData);
//运行脚本,从OBS中下载服务编排生成的文档
$model.ref('scriptDemo').run().then(function(data){
```

```
const decodedString = base64DecodeToBinary(data.buf);
const url = URL.createObjectURL(decodedString);
let link = document.createElement('a');
link.href = url;
// 设置下载的文档名称，命名空间__modedoc__CST为表格中合同模板实例的字段名
link.setAttribute('download', rowData.命名空间__modedoc__CST);
document.body.appendChild(link);
link.click();
URL.revokeObjectURL(link.href);
document.body.removeChild(link);
context.$message.success('DownLoad successfully:');
}).catch(function(error){
  console.log('error is', error);
  context.$message.error('DownLoad failed:' + error.resMsg);
});
```

10. 将操作按钮的“标签”设置为“下载文档”，同时将基本属性中的“列标题”设置为“下载”。

图 4-49 更改操作列标题和操作的标签



11. 返回标准页面，在属“性 > 表格区块”中，单击工具栏后的“添加”，为表格添加一个工具栏。

图 4-50 为表格添加工具栏



步骤6 标准页面设计完成后，单击页面上方的, 保存标准页面。
----结束

步骤六：验证合同文档生成和下载功能

- 步骤1** 在标准页面上方，单击, 进入预览页面。
- 步骤2** 输入合同内容，单击“提交合同参数并生成文档实例”。

图 4-51 输入合同内容



步骤3 提示“Submitted successfully”后，单击“刷新”，即可查看到提交的合同数据。

图 4-52 查看提交的合同数据

公司名称

A公司

合同金额

100000

合同日期

2024-11-07

合同类型

XXX合同

乙方公司名称

B公司

订单数量

50

提交合同参数并生成文档比例

+ 新增行

保存

删除

刷新

✓ 完成

☐	公司名称	合同金额	订单数量	合同类型	合同日期	合同名称	乙方公司名称	合同文档实例	下载
☐	A公司	100000	50	李四	2024-11-07	XXX合同	B公司	17309382923380862565154...	下载文档
☐	A公司	10	2	张三	2024-11-05	新采购合同	B公司	17308746835890138557431...	下载文档

< 1 >

10条/页

共2条

步骤4 单击数据后的“下载文档”，将合同文档下载到本地并查看文档内容。

图 4-53 合同文档被下载到本地

此电脑 > Windows (C:) > 用户 > > 下载

名称	修改日期	类型	大小
今天 (1)			
173093829233808625651548257232086e6a3-8e16-a1e9-7cde-2b7fac6f2356_NS1_contractModel.docx	2024/11/7 14:18	Microsoft Word ...	50 KB

图 4-54 查看下载到本地合同的内容

XXX合同

第一条 合同目的

本合同旨在规定 A公司 向乙方购买商品/服务的具体条款和条件。

第二条 商品/服务描述

商品/服务的详细描述，包括但不限于型号、规格、数量、单价等。

订单数目：50

第三条 价格条款

商品/服务的总额为 100000 元。

第四条 质量保证

乙方保证所提供的商品/服务符合约定的质量标准

第五条 违约责任

如一方违反合同条款，违约方应赔偿对方因此遭受的所有损失。

第六条 生效条件

本合同自双方授权代表签字盖章之日起生效。

甲方：A公司

乙方：B公司

合同签订人：李四

日期：2024-11-07

----结束

4.2 使用华为云 Astro 轻应用邮件模板发送邮件通知

应用场景

华为云Astro轻应用低代码平台提供了文档类模板创建能力，支持快速创建“Word文档”、“Excel文档”、“邮件”和“短信”四种类型的模板，以满足多样化的业务文档需求，提升文档的制作效率与规范性。

本实践以邮件模板为例进行介绍，基于该功能开发者可以预先设置好邮件格式和框架，避免每次都重复的编写。例如，商务沟通中的订单确认、会议邀请以及日常生活中的节日祝福、活动通知等，通过邮件模板只需设置收件人、主题、内容等关键信息，即可快速生成一封完整的邮件，从而节省大量的时间和精力。

方案优势

用户可以在服务编排中，通过拖、拉、拽的方式调用邮件模板节点、配置模板的输入输出参数，用于快速生成一份完整的邮件。

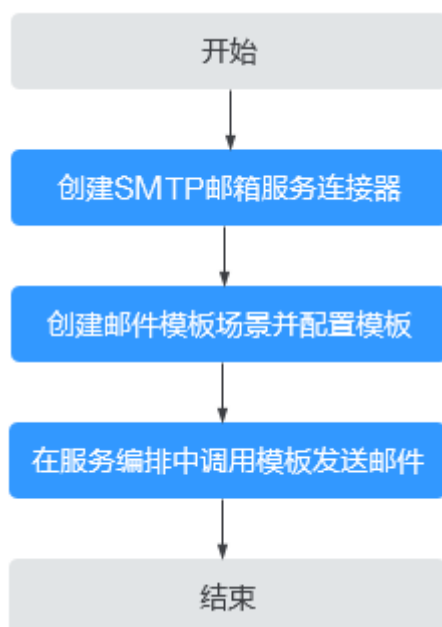
约束与限制

收件人、抄送和密送可以设置多个邮箱地址，多个邮箱之间用英文“;”隔开。收件人、抄送、密送和主题内容不能超过4096个字符。

操作流程

在华为云Astro轻应用中，通过邮件模板发送邮件的操作流程，如图4-55所示。

图 4-55 使用邮件模板发送邮件操作流程



步骤一：创建 SMTP 邮箱服务连接器

在应用中创建一个SMTP邮箱服务连接器，实现邮件发送功能。SMTP即简单邮件传输协议，是一种提供可靠且有效电子邮件传输的协议。

步骤1 创建一个低代码应用。

1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
3. 在“应用”中，单击“新建低代码应用”或单击，进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。
4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。

5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 4-56 创建一个空白应用

新建 空白应用

基本信息

添加图标

* 标签

我的第一个应用

分类

请选择

* 名称

A

描述

请输入

高级设置

取消

新建

表 4-13 新建空白应用参数说明

参数	说明	示例
标签	新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。	我的第一个应用
名称	新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下： - 长度不能超过31个字符，包括前缀命名空间的长度。 - 名称前的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。	A

- 步骤2 创建SMTP邮箱服务连接器。
1. 在应用设计器的左侧导航栏中，选择“集成”，单击“连接器 > 连接器实例”。

图 4-57 单击连接器实例



2. 在“类型 > 消息通知”中，单击“SMTP邮箱服务”，再单击“+”，进入创建SMTP邮箱服务页面。
3. 设置SMTP邮箱服务连接器的信息，单击“保存”。

图 4-58 创建 SMTP 邮箱服务连接器

创建SMTP邮箱服务

* 名称

smtp

* SMTP服务地址:端口

smtp.163.com:465

* 用户名

example@163.com

* 密码

.....

* 发件人邮箱地址

zhangsan@163.com

发件人显示名

请输入

描述

请输入

打包当前配置

☒ 打包当前配置

保存

取消

表4-14仅介绍本实践中使用到的参数，其他参数的解释请参见[对接SMTP发送邮件](#)。

表 4-14 新建 SMTP 邮箱服务参数说明

参数	说明	示例
名称	新建SMTP邮箱服务连接器的名称，创建后不支持修改。命名要求如下： - 长度不能超过64个字符，包括前缀命名空间的长度。 - 标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母，数字和下划线组成，且不能以下划线结尾。	smtp

参数	说明	示例
SMTP服务地址: 端口	邮箱开通SMTP功能时，配置的服务器地址和端口号。 常用邮箱开启SMTP后，SMTP服务器地址及端口请参见表4-15。	smtp.163.com:465
用户名	邮箱的用户名，如果未修改过，默认为邮箱地址。	example@163.com
密码	邮箱开通SMTP功能时，生成的授权密码，是一段随机字符串。	-
发件人邮箱地址	发送电子邮件的邮箱地址，通常格式为 用户名@域名 。	zhangsan@163.com

表 4-15 常用邮箱 SMTP 服务器地址及端口

邮箱	SMTP服务器地址	服务端口
163邮箱	smtp.163.com	- SSL协议时，配置为465或者994 - 其他协议时，配置为25
126邮箱	smtp.126.com	25
YEAH邮箱	smtp.yeah.net	25
QQ企业邮箱	- 国内用户： smtp.exmail.qq.com - 海外用户： hwsmtpp.exmail.qq.com	使用SSL协议，端口为465
SINA邮箱	smtp.sina.com	25

步骤3 测试邮件是否能正常发送。

1. 在连接器详情页面，单击测试中的“发送邮件”。
2. 输入收件人、主题和邮件内容，单击“发送”。

图 4-59 发送邮件

发送邮件

* 收件人

抄送

请输入

密送

请输入

* 主题

SMTP功能验证

* 邮件内容

使用SMTP发送邮件

取消

发送

收件人可收到邮件，表示测试成功，如图4-60。

图 4-60 接收邮件内容



----结束

步骤二：创建邮件模板场景并配置模板

创建一个模板场景并在场景中添加一个邮件模板。在模板场景的配置中，添加邮件模板中需要替换的参数。模板场景是一个业务场景的集合，在模板场景中支持创建多个模板，模板之间共享数据结构。

步骤1 创建邮件模板场景。

1. 在应用设计器的左侧导航栏中，选择“逻辑”，单击“更多 > 模板”。

图 4-61 单击模板



- 单击“添加模板场景”，设置模板场景的标签和名称，单击“确定”。

图 4-62 创建模板场景

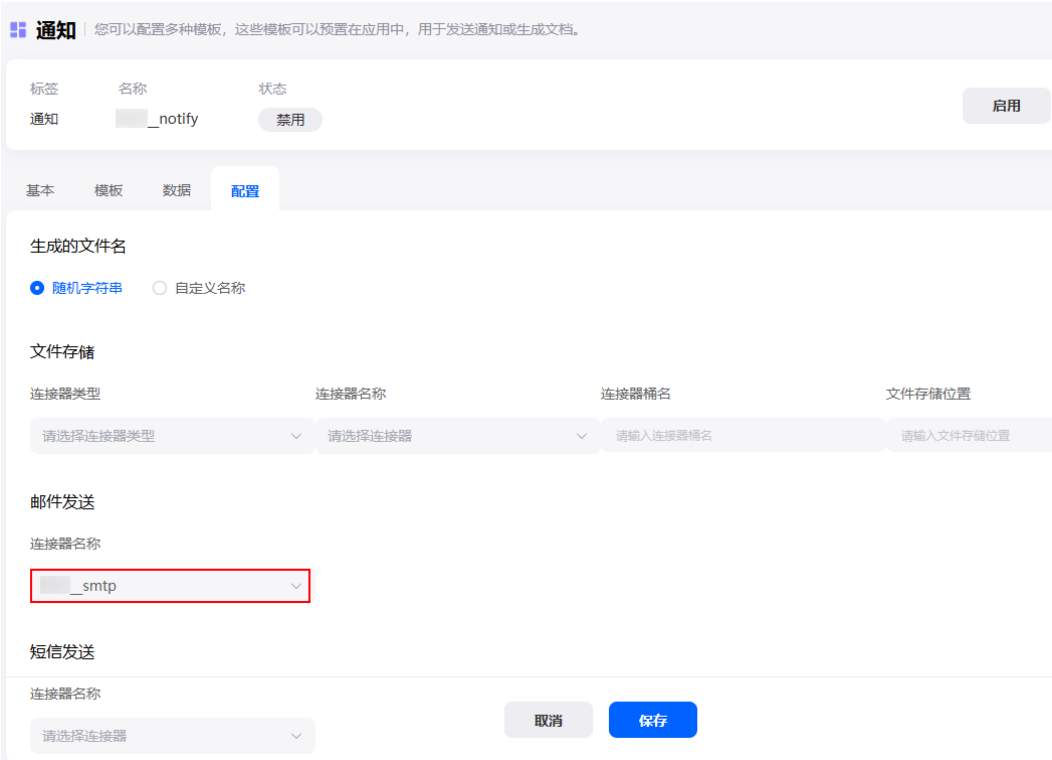


表 4-16 新建模板场景参数说明

参数	说明	示例
标签	新建模板场景的名称，创建后可修改。 取值范围：1~80个字符。	通知
名称	新建模板场景在系统中的标识，创建后不支持修改。命名要求如下： - 长度不能超过64个字符，包括前缀命名空间的长度。 - 标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母，数字和下划线组成，且不能以下划线结尾。	notify

步骤2 在“配置”页签，选择**步骤一：创建SMTP邮箱服务连接器**中创建的连接器的，单击“保存”。

图 4-63 选择连接器



步骤3 在“数据”页签，为通知模板添加表4-17中参数。
此处添加的模板参数，对应通知模板中需要替换的内容。

图 4-64 为通知模板添加公司名称参数

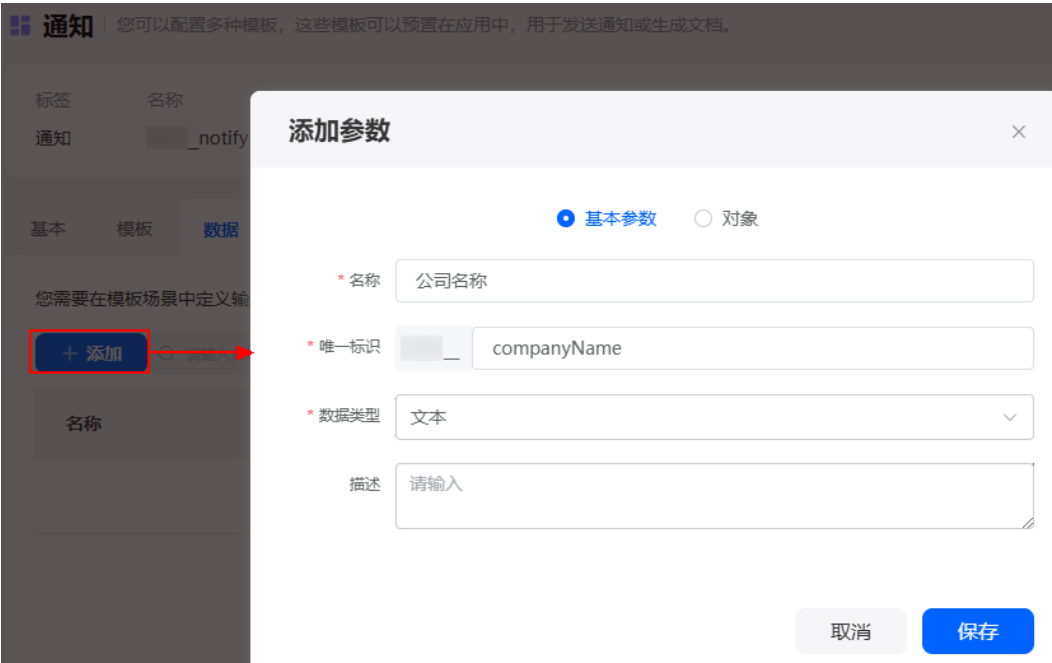


表 4-17 待添加参数

名称	唯一标识	数据类型
公司名称	companyName	文本
客户公司名称	customerCompanyName	文本
姓名	name	文本

图 4-65 查看已添加的参数



步骤4 在“模板”页签，单击“添加”，创建邮件模板。

图 4-66 新建邮件模板

新建模板

* 标签

问好模板

* 名称

emailModel

描述

模板类型

☐ Word文档

☐ Excel文档

☒ 邮件

☐ 短信

取消

确定

表 4-18 新建邮件模板参数说明

参数	说明	示例
标签	新建模板的名称，创建后可修改。 取值范围：1~80个字符。	问好模板
名称	新建模板在系统中的标识，创建后不支持修改。命名要求如下： - 长度不能超过63个字符，包括前缀命名空间的长度。 标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母，数字和下划线组成，且不能以下划线结尾。	emailModel

参数	说明	示例
模板类型	新建文档模板的类型。 - Word文档：支持用户上传Word格式的文档模板，文档的大小建议小于10M。 - Excel文档：支持用户上传Excel类型的文档模板。上传的Excel文档不能超过10个Sheet页，每个Sheet页中列不能超过200个，行不能超过1000个，单元格内字符串的长度不超过500个字符。 - 邮件：支持用户在模板编辑界面编写邮件模板。 - 短信：支持用户在模板编辑界面编写短信模板。	邮件

步骤5 编辑邮件模板。

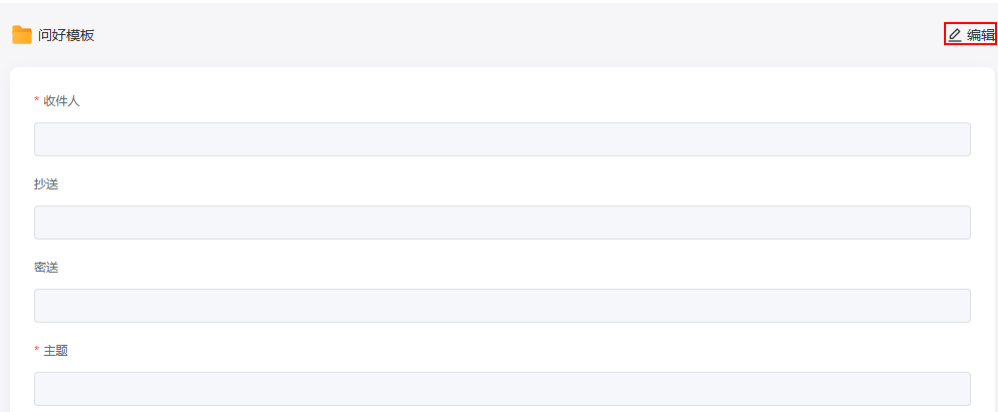
1. 在“模板”页签，单击步骤4中创建模板后的，进入邮件模板页面。

图 4-67 进入模板编辑页面



2. 单击页面右上方的“编辑”，进入模板编辑页面。

图 4-68 单击编辑



3. 编辑邮件内容，邮件中的变量为**步骤3**中添加的模板参数。

图 4-69 编辑邮件内容

问好模板

取消保存

* 收件人

.com;.com

抄送

密送

* 主题

节日问候

* 邮件内容

您好!
我是\${命名空间__name}, 来自\${命名空间__companyName}。
在这特别的日子, 我想向您及您的团队 (\${命名空间__customerCompanyName}) 致以最诚挚的祝福, 希望这一年里您的事业蒸蒸日上, 生活幸福美满。
此致
敬礼!

表 4-19 邮件内容参数说明

参数	说明	示例
收件人	收件人的邮箱地址，长度不能超过4096个字符，涉及多个地址时请使用英文“;”隔开。	***.com
主题	邮件的主题，长度不能超过4096个字符。	节日问候
邮件内容	待发送邮件的具体内容，长度不能超过4096个字符。邮件内容请尽量简洁明了、逻辑清晰，确保信息能准确传达。	您好! 我是\${命名空间__name}, 来自\${命名空间__companyName}。 在这特别的日子, 我想向您及您的团队 (\${命名空间__customerCompanyName}) 致以最诚挚的祝福, 希望这一年里您的事业蒸蒸日上, 生活幸福美满。 此致 敬礼!

4. 内容编辑完成后，单击“保存”，再单击“启用”，启用邮件模板。

步骤6 返回邮件模板场景中，单击“启用”，启用模板场景。

模板场景“状态”变为“启用”，表明模板场景启用成功。

图 4-70 查看模板创建状态



----结束

步骤三：在服务编排中调用模板发送邮件

创建一个服务编排，添加一个“调用模板”图元，实现邮件发送功能。

步骤1 在应用设计器的左侧导航栏中，选择“逻辑”，单击编排后的“+”。

步骤2 设置服务编排的标签和名称，单击“添加”。

图 4-71 新建服务编排

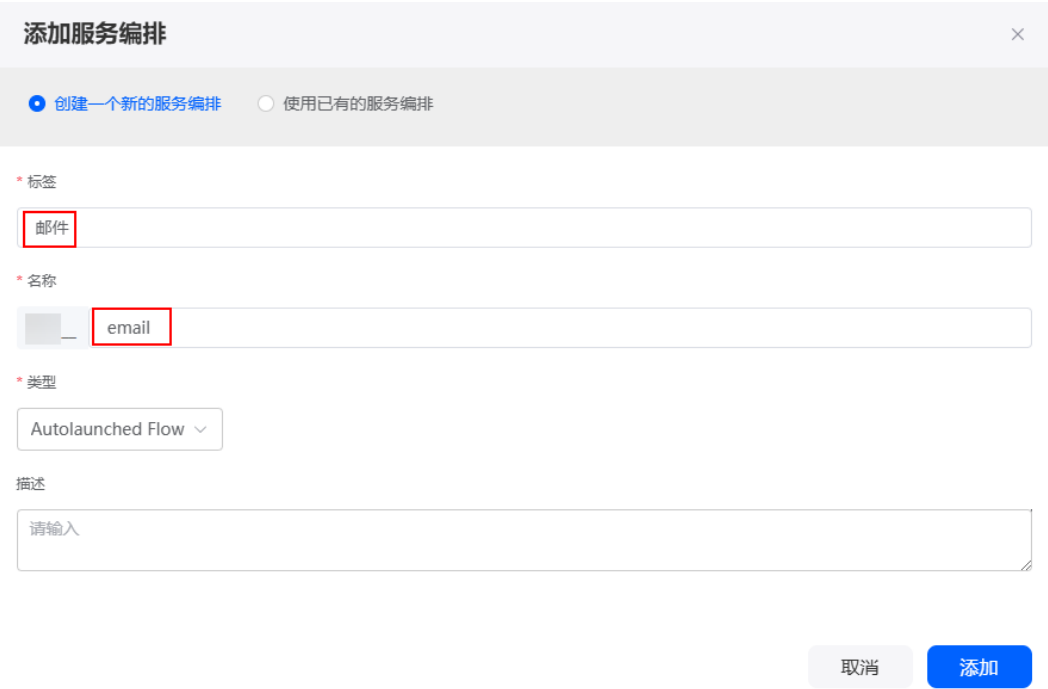


表 4-20 新建服务编排参数说明

参数	说明	示例
标签	服务编排的标签名，用于在界面展示，创建后可修改。 取值范围：1~64个字符。	邮件
名称	服务编排在系统中的唯一标识，创建后不支持修改。命名要求如下： - 长度不能超过64个字符，包括前缀命名空间的长度。 标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。 - 名称必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不能以下划线结尾。	email

步骤3 创建全局上下文变量。

1. 在服务编排设计页面，选中开始节点，单击 。
2. 在全局上下文中，单击变量后的 ，新建变量“variable0”。
3. 单击“variable0”变量后的 ，选择“设置”。
4. 将变量的“名称”设置为“companyName”，单击“保存”。

图 4-72 新建 companyName 变量

变量

* 名称

companyName

变量名是用于变量在流程中引用的唯一标识。修改变量名不会改变图元中的引用，可能导致流程不可用。

* 数据类型

文本

默认值

请输入常量或从列表中选择

描述

请输入

是否为数组

外部使用

取消

保存

5. 按照上述操作，创建表4-21中的变量。

图 4-73 查看已创建的变量

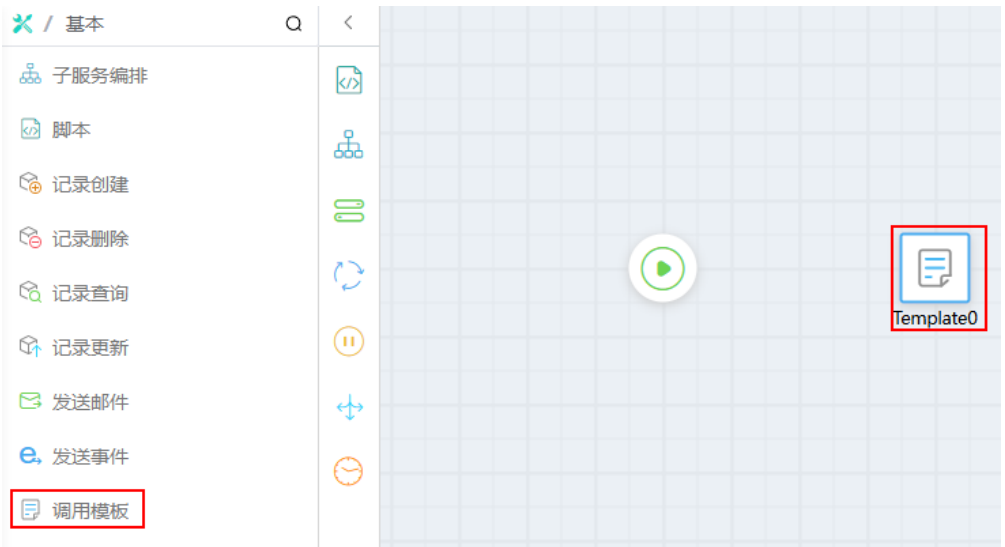


表 4-21 需要创建的全局上下文变量

名称	数据类型
companyName（已创建）	文本
customerCompanyName	文本
name	文本

- 步骤4 添加调用模板图元。
- 在“基本”中，拖拽“调用模板”图元到开始图元后。

图 4-74 拖拽调用模板图元到画布中



2. 选中调用模板图元Template0，单击，设置调用模板。

图 4-75 设置调用模板图元

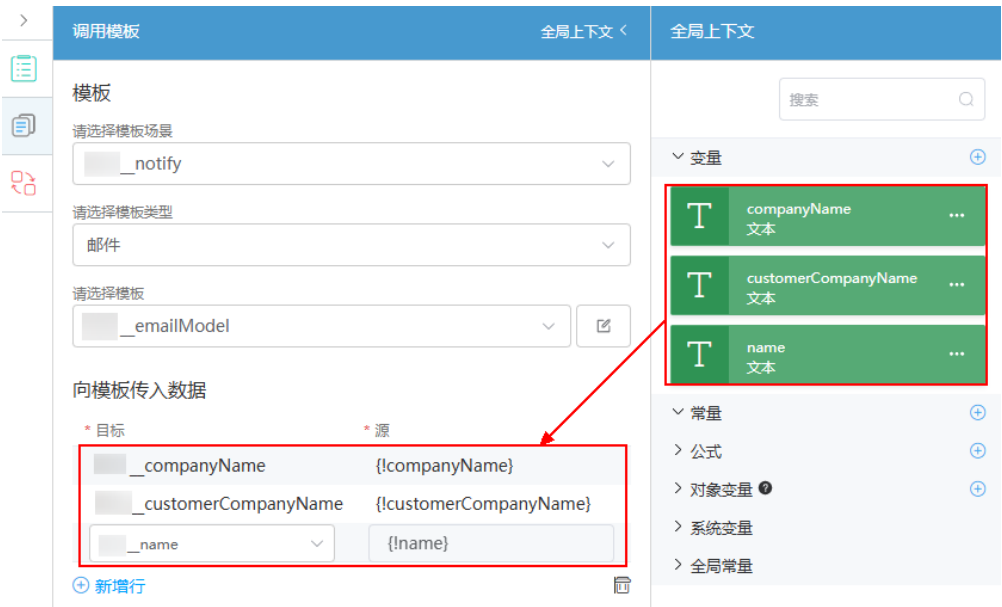


表 4-22 调用模板图元参数说明

参数	说明	示例
请选择模板场景	选择调用模板图元关联的模板场景，即步骤1中创建的。	命名空间_notify
请选择模板类型	选择待调用模板的类型。	邮件
请选择文档模板	选择模板场景中创建的邮件模板，即步骤4中创建的。	命名空间_emailModel

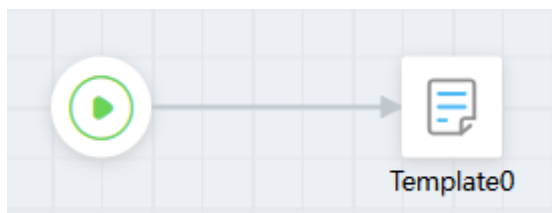
3. 在“向模板传入数据”中，单击“新增行”，将步骤3中创建的变量拖到服务编排入参中，作为服务编排的输入参数。

图 4-76 设置服务编排入参



步骤5 连接图元指定逻辑关系，即从开始图元连向调用模板图元。

图 4-77 指定图元逻辑关系



步骤6 单击页面上方的, 保存服务编排。

步骤7 单击, 进入服务编排调试页面。

在输入参数中，输入如下内容，并单击“运行”。

图 4-78 调试服务编排

The screenshot shows a web interface for debugging a service orchestration. At the top, there is a header with a grey square icon and the text '_email'. Below this, there is a section titled '请求头参数' (Request Header Parameters) with a blue button labeled '保存输入参数' (Save Input Parameters). Underneath, there is a section titled '输入参数' (Input Parameters) containing a text area with a red border. The text area contains the following JSON object:

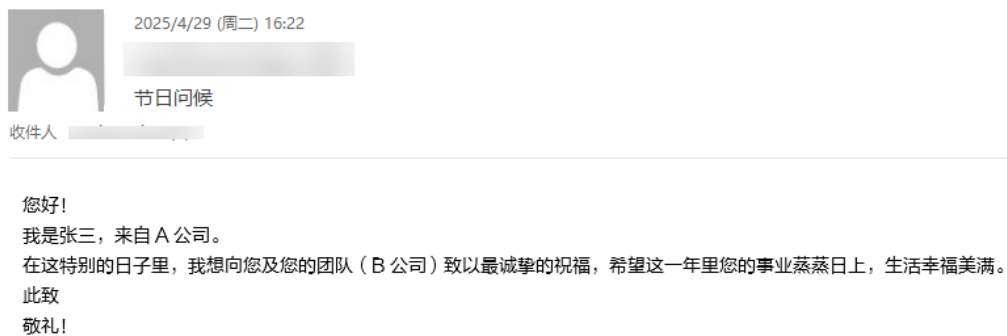
```
{
  "companyName": "A公司",
  "customerCompanyName": "B公司",
  "name": "张三"
}
```

At the bottom of the interface, there is a blue button labeled '运行' (Run) and a small black circle with a white question mark.

```
{  
  "companyName": "A公司",  
  "customerCompanyName": "B公司",  
  "name": "张三"  
}
```

运行成功后，[步骤5](#)中设置的邮件收件人，即可收到节日问候邮件，如[图4-79](#)。

图 4-79 节日问候邮件



步骤8 单击页面上方的, 启用服务编排。

----结束

5 工作流专项

5.1 通过华为云 Astro 轻应用流程模板创建出差审批电子流

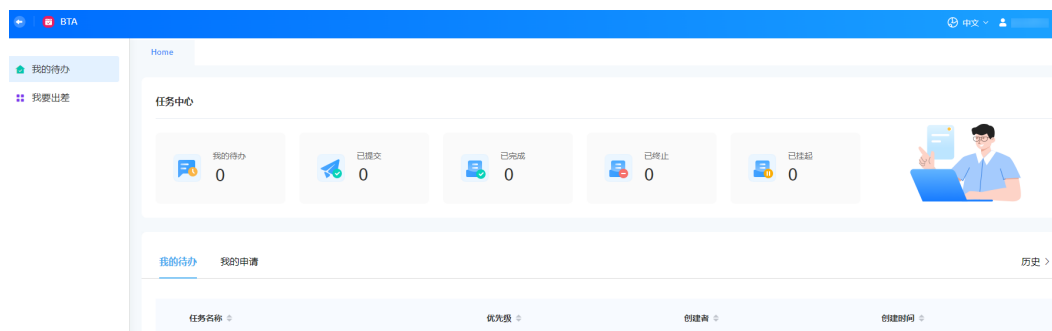
方案概述

华为云Astro轻应用低代码平台基于业界BPMN 2.0标准，实现了自己的业务流程管理系统，即工作流。华为云Astro轻应用中的工作流是一套图形化的流程编排引擎，着重于构建带有用户交互行为的业务流程，例如审批流、工单派发流程等。本实践通过创建一个出差审批应用，帮助您快速熟悉华为云Astro轻应用中的工作流。

员工出差是企业运营中最常见的业务场景，涉及到一系列的管理流程，例如出差申请、审批、行程安排、费用报销等。本实践以一个简单的员工出差场景，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，向您介绍如何使用华为云Astro轻应用中的工作流。本示例中的出差审批应用主要包括如下功能：

- 基于工作流模板创建出差电子流。
- 发送邮件。

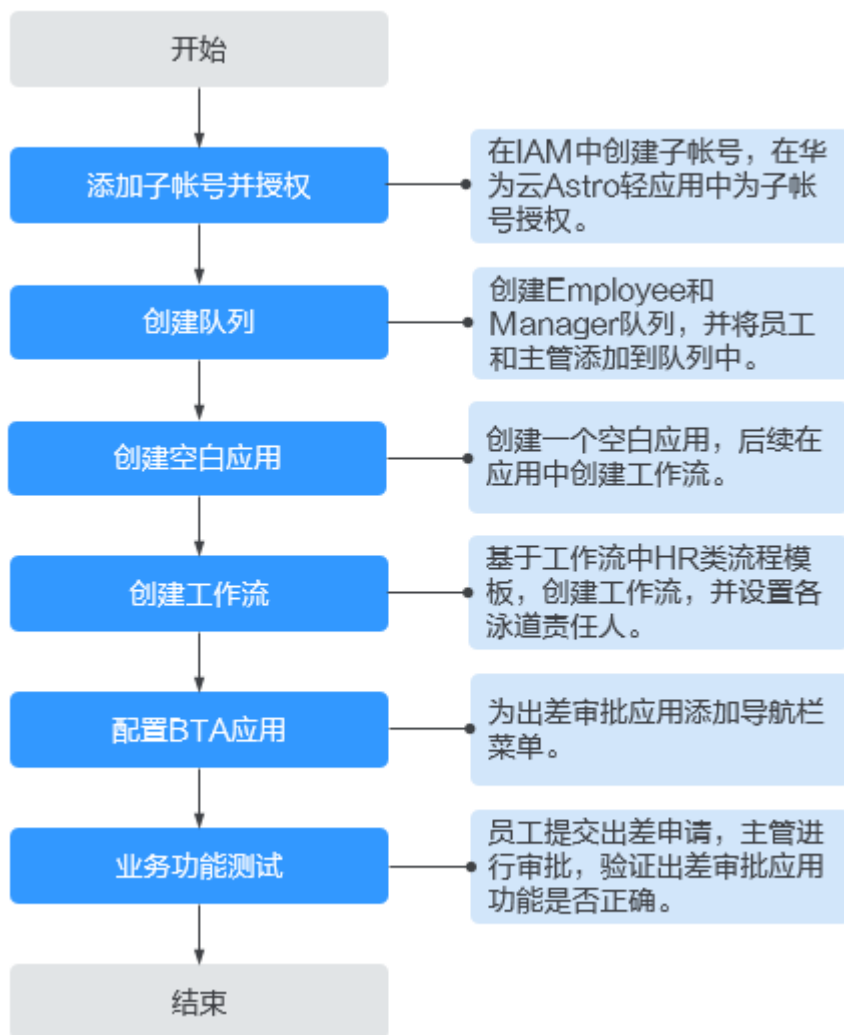
图 5-1 出差审批应用最终效果图



操作流程

在华为云Astro轻应用中开发出差审批应用的流程，如图5-2所示。

图 5-2 出差审批应用开发流程



1. **步骤一：添加业务用户并进行授权**

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、主管添加到华为云Astro轻应用中。

2. **步骤二：创建队列**

队列是华为云Astro轻应用中的一种成员集，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。本示例中需要创建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。

3. **步骤三：创建空白应用**

创建应用是在华为云Astro轻应用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。

4. **步骤四：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。

5. **步骤五：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。

6. **步骤六：业务功能调测**

验证出差审批流程是否按照预期执行，即员工提交出差申请，主管对申请进行审批，分别测试审批通过、拒绝与重填。

步骤一：添加业务用户并进行授权

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、主管添加到华为云Astro轻应用中。

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[华为云官网](#)，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 在华为云Astro轻应用中添加子账号并授权。

1. 以华为账号登录[华为云Astro轻应用服务控制台](#)。
2. 在主页中，单击“进入首页”，进入开发页面。
3. 在页面左上角，单击，选择“环境管理 > 环境配置”，进入华为云Astro轻应用环境配置。

图 5-3 进入华为云 Astro 轻应用环境配置



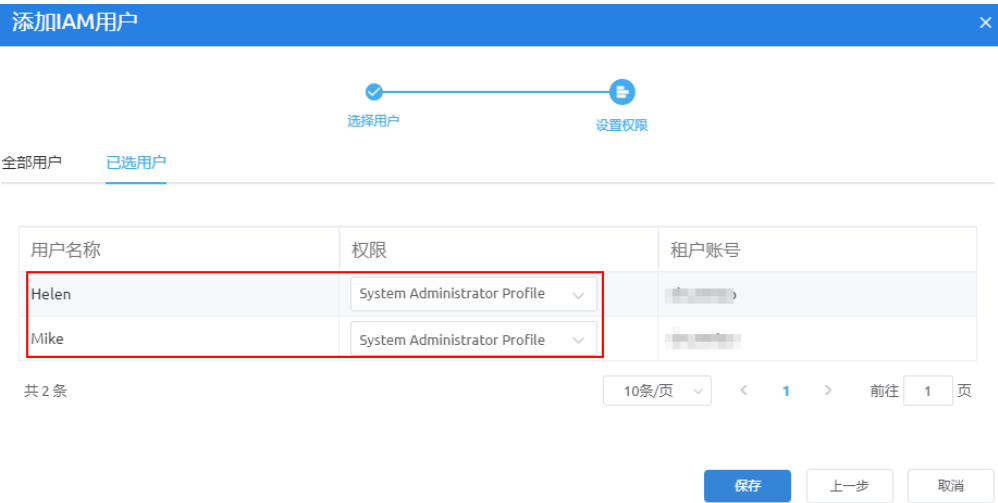
4. 在左侧导航栏中，选择“用户安全 > 用户”，单击“添加IAM用户”。
5. 在全部用户列表中，选择**步骤1**中创建的子账号，单击“下一步”。

图 5-4 选择待添加的子账号



6. 赋予主管和员工System Administrator Profile权限，单击“保存”。
- System Administrator Profile为系统管理员权限，拥有这个权限的用户拥有华为云Astro轻应用全部权限。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。

图 5-5 给子账号授权



----结束

步骤二：创建队列

队列是华为云Astro轻应用中的一种成员集，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5-1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 5-1 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 以华为账号登录华为云Astro轻应用环境配置。
- 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5-1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，本示例配置为Employee。

– 名称：新建队列的名称，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 5-6 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。
- 文档版本 01 (2025-09-04)

版权所有 © 华为云计算技术有限公司

115

图 5-7 设置标签和名称

基本信息

* 标签

Manager

* 名称

Manager

公共邮箱

请输入

工作队列管理者

请选择

☐ 给成员发送邮件

☐ 队列对象

图 5-8 为队列添加 Helen 成员及租户账号

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

用户: Helen

取消

添加

----结束

步骤 3：创建空白应用

创建应用是在华为云Astro轻应用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。

- 步骤1 以华为账号登录[华为云Astro轻应用服务控制台](#)。
- 步骤2 在主页中，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 单击新建低代码应用后的 ，进入新建空白应用页面。

图 5-9 进入创建轻应用入口



- 步骤5** 在新建低代码应用中，选择“标准应用”，单击“确定”。
- 步骤6** 设置应用标签和名称，此处均设置为BTA。

图 5-10 设置轻应用标签和名称



图5-10名称前模糊掉的内容为命名空间，在华为云Astro轻应用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7** 单击“新建”，进入BTA新版应用设计器。

图 5-11 BTA 新版应用设计器



----结束

步骤四：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。

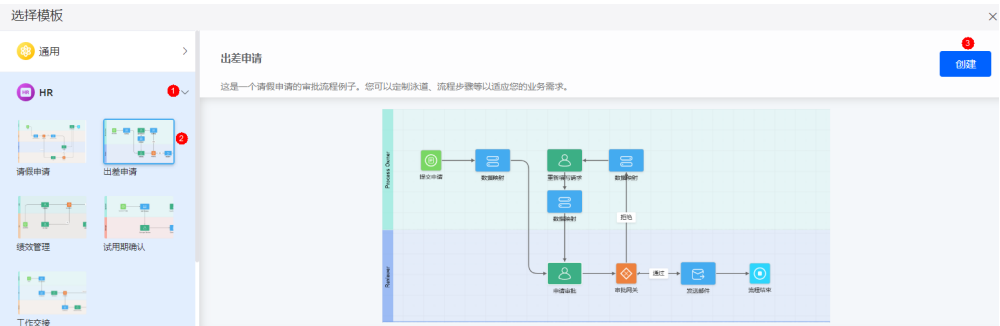
- 1. 在BTA新版应用设计器的“开始”页面中，单击“新建工作流”。
- 2. 在新建工作流页面，单击“基于模板”，设置标签、名称和描述信息。

图 5-12 添加工作流



3. 单击“选择模板”，选择“HR”中的“出差申请”模板后，单击“创建”。

图 5-13 选择出差申请模板



创建完成后，自动进入出差申请工作流编辑页面。在设置工作流前，您可以先通过表5-2了解出差申请工作流中各节点的功能。

图 5-14 出差申请工作流开发页面

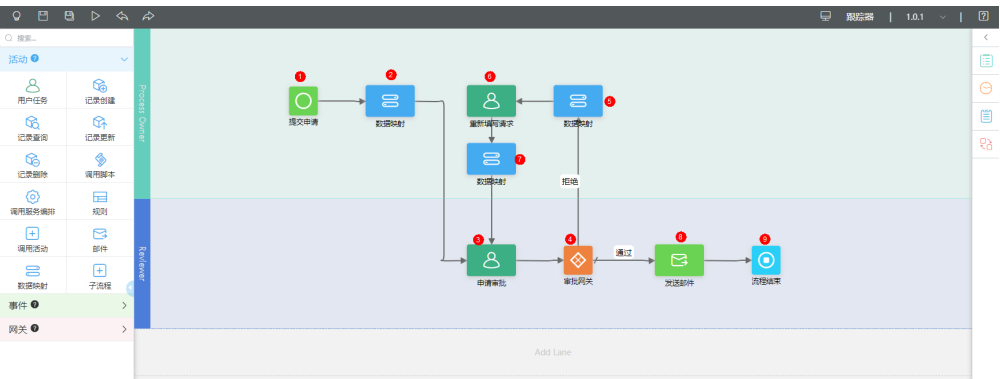


表 5-2 出差申请工作流中各节点功能介绍

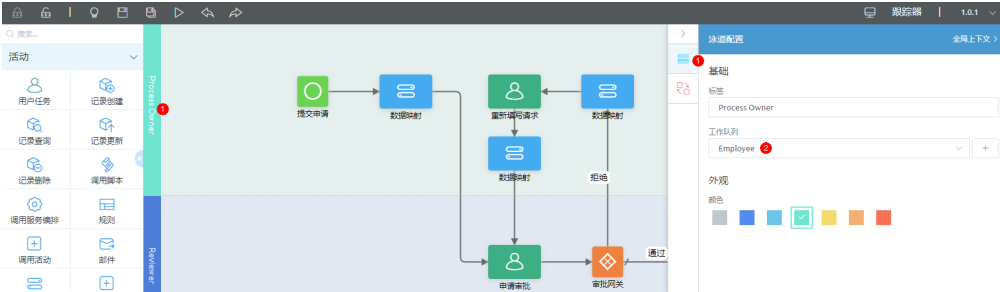
编号	节点	功能说明
1	提交申请	开始节点，调用一个标准表单 TravelRequest，供出差申请人提交出差申请。
2	数据映射	将出差申请标准表单中的请求字段映射到对象中。
3	申请审批	用户任务，将对象中的字段渲染到标准表单 Approve并且确定审批人。
4	审批网关	审批人进行通过或拒绝两种操作。
5	数据映射	申请拒绝后，将Approve标准表单中的字段映射回TravelRequest标准表单。
6	重新填写请求	刷新TravelRequest表单中的内容，重新提交出差申请。

编号	节点	功能说明
7	数据映射	将TravelRequest表单中的字段值映射到Approve中，重新发起审批申请。
8	发送邮件	审批通过，发送邮件将结果告知出差申请人。
9	流程结束	结束节点，执行到此整个工作流执行结束。

步骤2 设置工作流，配置各泳道处理人。

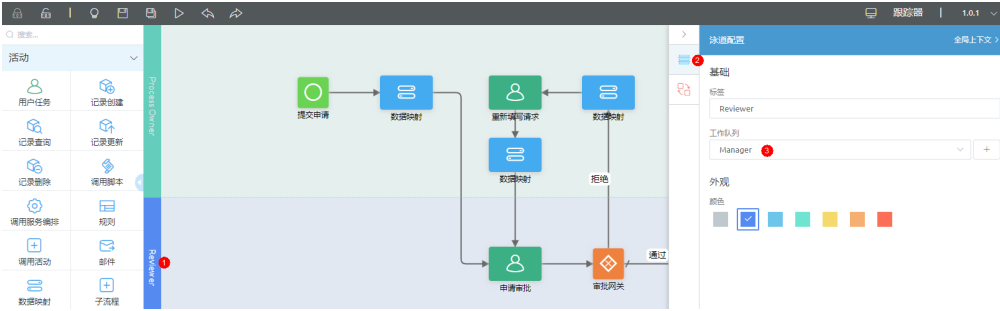
1. 在工作流开发页面，单击“Process Owner”泳道，配置工作队列为“Employee”。

图 5-15 配置员工为申请人



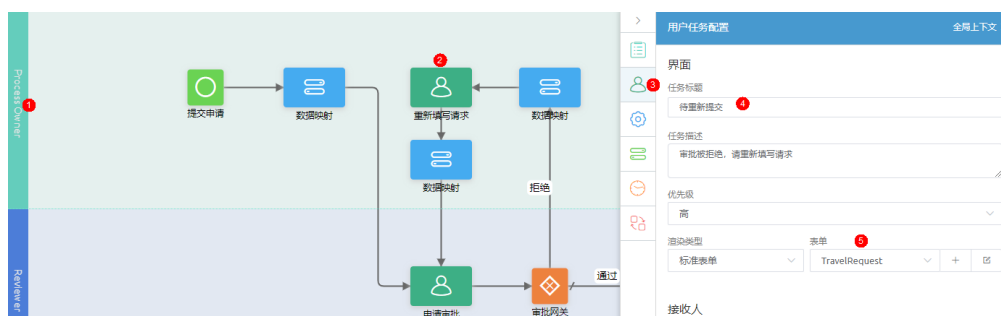
2. 单击“Reviewer”泳道，配置工作队列为“Manager”。

图 5-16 配置主管为审批人



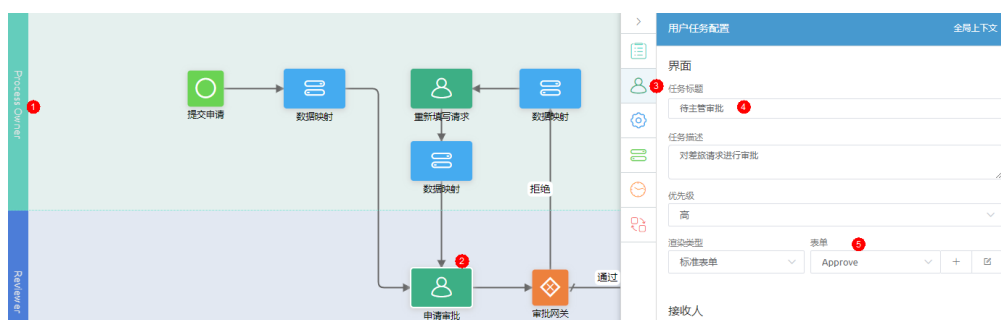
3. 单击泳道“Process Owner”上的“重新填写请求”用户任务元素，设置任务标题为“待重新提交”。
- 执行该操作的目的是，后续在“我的待办”中可以看到“待重新提交”的任务状态。

图 5-17 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 5-18 配置为待主管审批



- 单击，保存工作流。
- 单击，启用工作流。

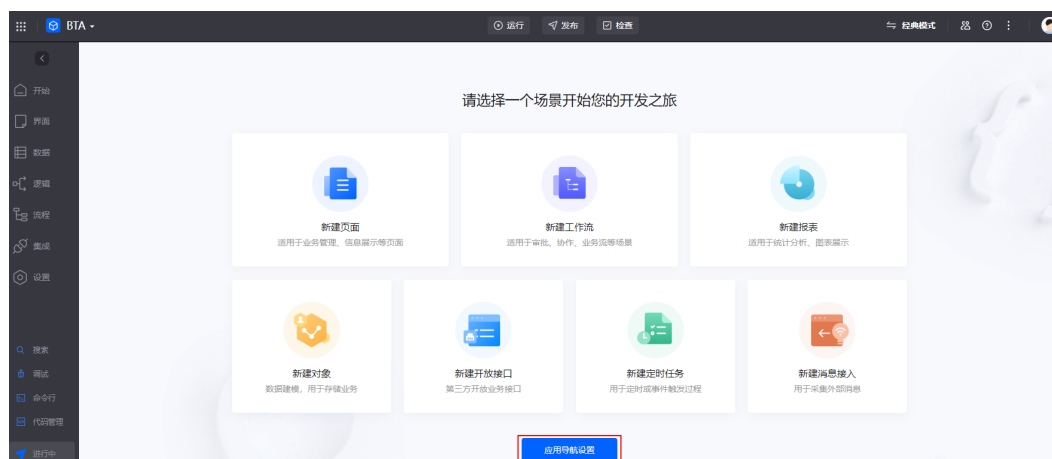
----结束

步骤五：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 5-19 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”，单击“保存”。

图 5-20 编辑页签



步骤3 在主导航设置中，单击“新建”，新建“我要出差”菜单项，单击“保存”。

图 5-21 添加我要出差页签

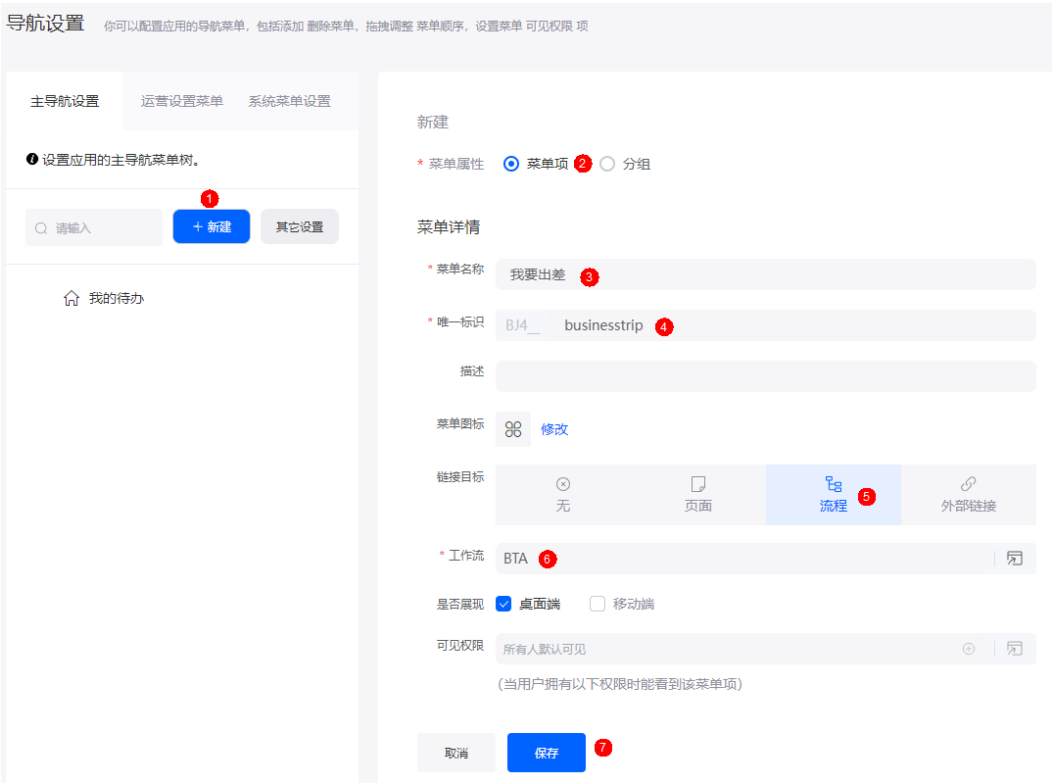
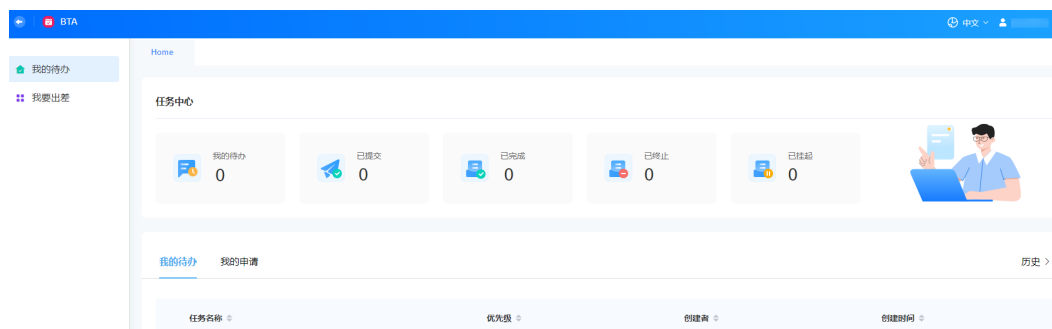


图 5-22 导航条最终效果



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 5-23 出差审批应用效果图



到此，您已完成出差审批应用的开发。

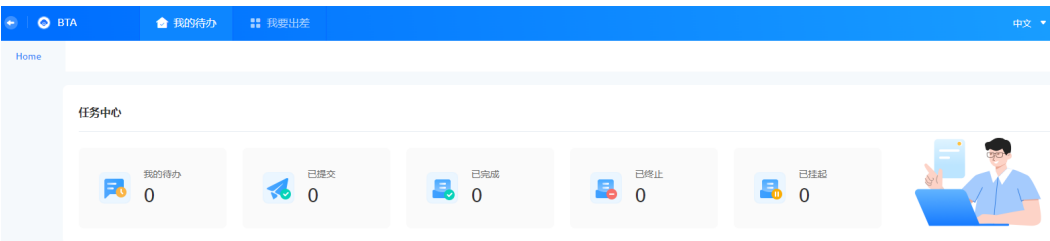
思考：如何将导航栏菜单在出差审批应用顶部显示？

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 5-24 修改菜单样式、应用图标



图 5-25 设置后效果



----结束

步骤六：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。

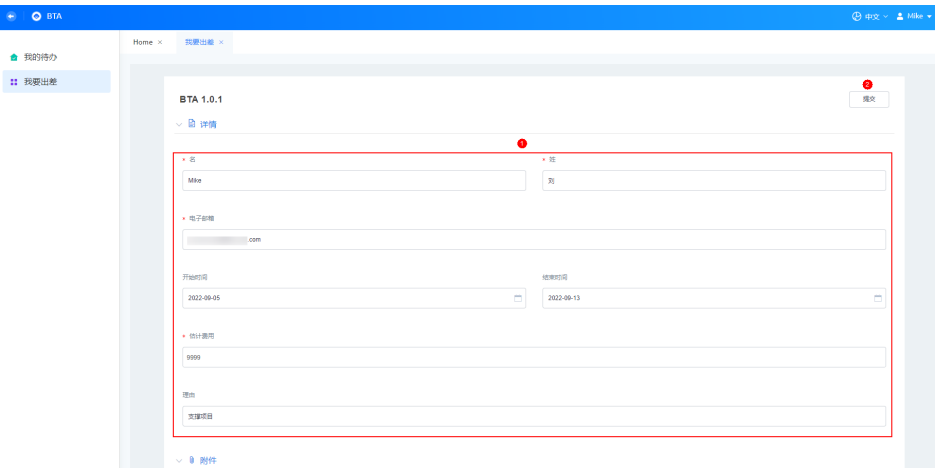
- 1. 以员工账号（本示例为Mike），登录[华为云Astro轻应用服务控制台](#)。

图 5-26 员工登录华为云 Astro 轻应用



- 2. 在主页中，单击“进入首页”，进入应用开发页面。
- 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”，进入BTA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 5-27 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。

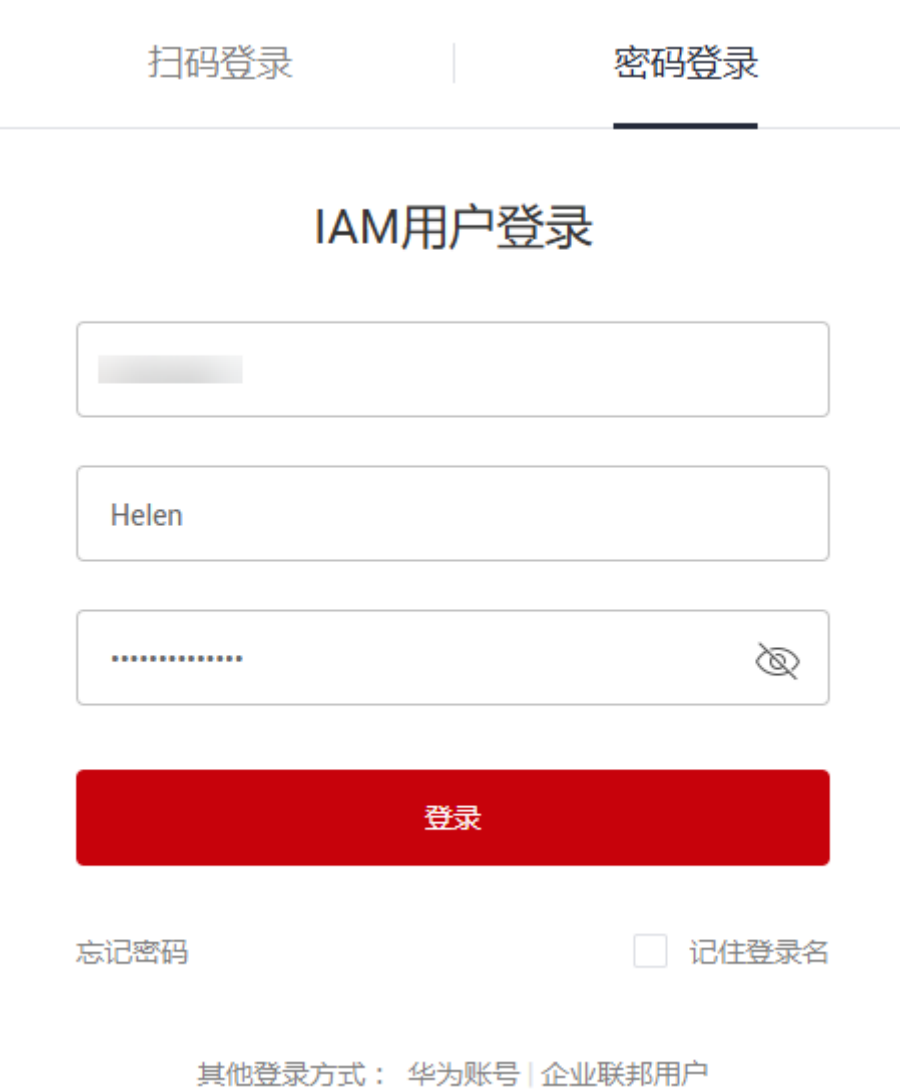
图 5-28 查看我的申请



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[华为云Astro轻应用服务控制台](#)。

图 5-29 主管登录华为云 Astro 轻应用



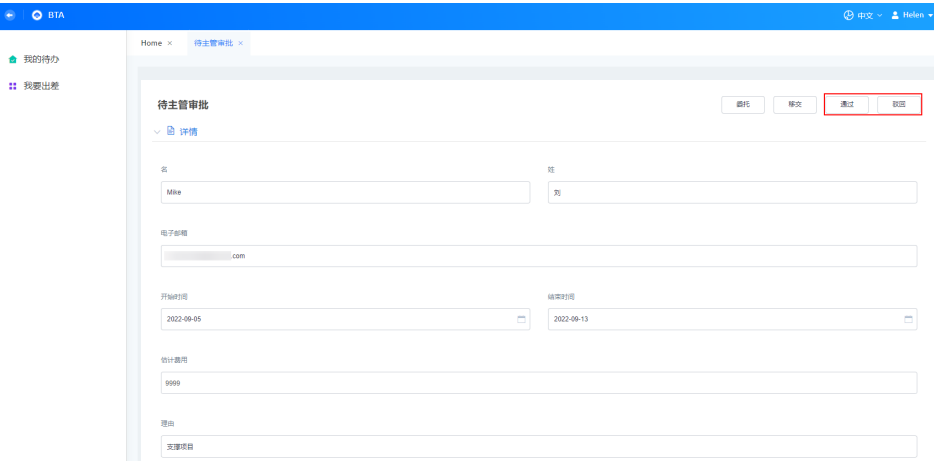
2. 在主页中，单击“进入首页”，进入应用开发。

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”，进入BTA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 5-30 我的待办

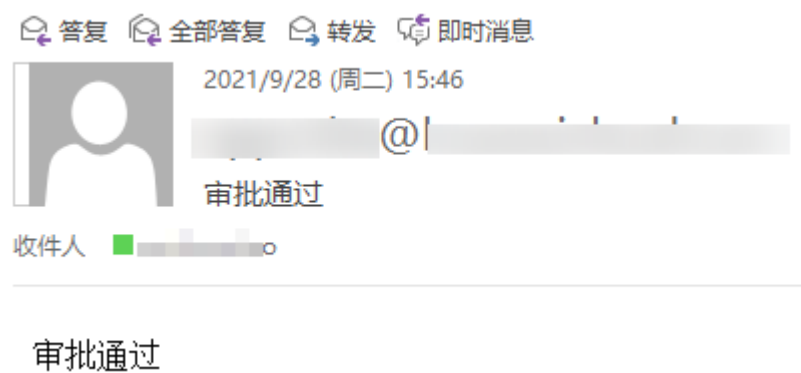


图 5-31 主管审批页面



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5-27中的员工电子邮箱中，如图5-32。

图 5-32 审批通过邮件



- 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5-33。

图 5-33 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，相信您对华为云Astro轻应用中的轻应用及工作流有了一定的了解。

----结束

5.2 通过华为云 Astro 轻应用工作流中的用户任务实现优先审批功能

应用场景

在企业运营过程中，经常会遇到一些需要快速处理或者特殊关注的任务。例如，某些紧急项目需要快速审批，以便尽快启动。当企业重要客户提出需求时，需要快速响应以满足客户需求。或者在某些特殊情况下，高层领导需要具有优先审批权。

在华为云Astro轻应用的工作流中，通过设置用户任务的优先审批功能，可轻松实现上述功能。用户任务用来表示业务流程中由用户参与完成的工作。当引擎处理到该节点时，给指定的用户或者一组用户创建待处理的任务项，等待用户的处理。

本实践将为您介绍在华为云Astro轻应用工作流中，如何通过配置用户任务的审批动作和终止动作，在会签过程中实现优先级审批功能。

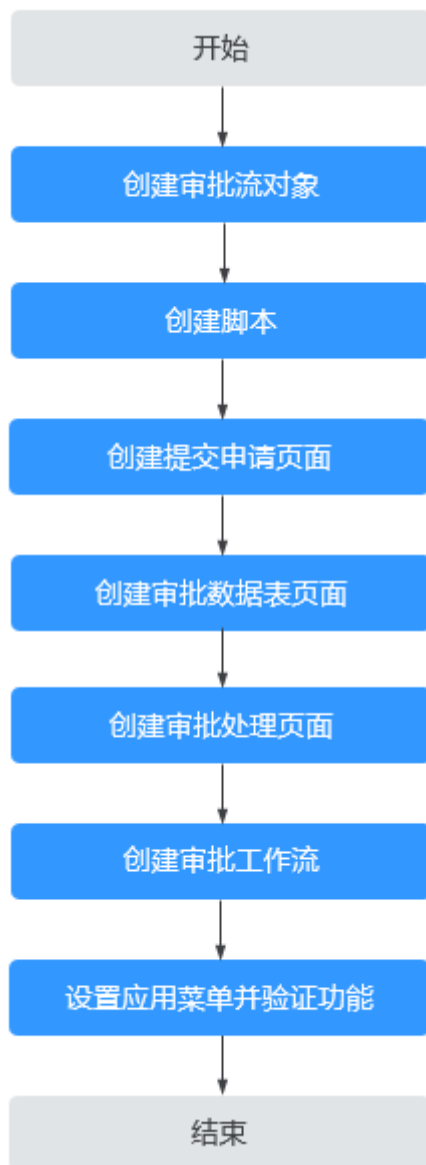
方案优势

华为云Astro轻应用工作流中内置了实现顺序审批、或签和会签等任务分派规则，可基于用户任务实现复杂的任务分派规则，应对复杂的交互流程。

操作流程

在华为云Astro轻应用的工作流中，通过配置用户任务，在会签过程中实现优先审批功能的流程，如图5-34所示。

图 5-34 优先审批功能实现流程



步骤一：创建审批流对象

创建一个审批流对象，用于存储用户提交的申请数据。

步骤1 创建一个低代码应用。

1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
3. 在“应用”中，单击“新建低代码应用”或单击，进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。

4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。
5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 5-35 创建一个空白应用

新建 空白应用

基本信息

添加图标

* 标签

我的第一个应用

* 名称

A

分类

请选择

描述

请输入

高级设置

取消

新建

表 5-3 新建空白应用参数说明

参数	说明	示例
标签	新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。	我的第一个应用
名称	新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下： - 长度不能超过31个字符，包括前缀命名空间的长度。 - 名称前的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。	A

- 步骤2 在应用设计器的左侧导航栏中，选择“数据”，单击对象中的“+”。
- 步骤3 设置对象的名称和唯一标识，单击“确定”。

图 5-36 创建对象 processDemo

创建新对象

您可以创建用来保存持久化数据的实体对象，对象里的数据存储在平台里面

从空白创建

从模板创建

您将创建单个对象，从零开始设计对象的字段，并添加测试数据

* 对象名称

processDemo

* 唯一标识

processDemo

取消

确定

表 5-4 新建 processDemo 对象参数说明

参数	说明	示例
对象名称	新建对象的名称，创建后可修改。 取值范围：1~80个字符。	processDemo
唯一标识	新建对象在系统中的标识，创建后不支持修改。命名要求如下： - 长度不能超过63个字符，包括前缀命名空间的长度。 标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。 - 必须以英文字母开头，只能由英文字母，数字和下划线组成，且不能以下划线结尾。	processDemo

- 步骤4 在已创建的对象中，单击，进入对象详情页面。
- 步骤5 在“字段”页签，单击“添加”，为对象添加apply字段。

图 5-37 添加 apply 字段

添加字段

*

显示名称

申请标题

*

唯一标识

apply

*

字段类型

文本

...

*

数据长度

64

描述

取消

确认

表 5-5 添加 apply 字段参数说明

参数	说明	示例
显示名称	新建字段的名称，创建后可修改。 取值范围：1~63个字符。	申请标题
唯一标识	新建字段在系统中的标识，创建后不支持修改。命名要求如下： - 长度不能超过63个字符，包括前缀命名空间的长度。 - 必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。	apply
字段类型	单击“...”，在弹出的页面中，根据页面提供的参数解释，选择新建字段所属的类型。	文本
数据长度	允许输入字段的长度。	64

步骤6 在“字段”页签，再次单击“添加”按钮，添加表5-6中字段。

图 5-38 processDemo 对象中包含字段



表 5-6 待添加字段

显示名称	唯一标识	字段类型
申请类型	type	文本
申请时间	deadline	日期
流程标识	bpmlId	文本
流程状态	bpmState	文本

----结束

步骤二：创建脚本

- 步骤1 创建脚本submit，用于启动工作流。
1. 在应用设计器中，选择“逻辑”，单击脚本后的“+”。

2. 新建一个空白的脚本，名称设置为“submit”，单击“添加”。

图 5-39 创建脚本 submit

3. 在脚本编辑器中，输入示例代码。

```
import * as bpm from 'bp';

@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'Object', required: true, label: 'tripData' })
  applicationData: object;
}

@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'String', required: true, label: 'interviewID' })
  interviewID: string;
}

@action.object({ type: "method" })
export class SubmitTrip {
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public run(input: ActionInput): ActionOutput {
    let out = new ActionOutput();
    out.interviewID = bpm.newClientV1().start('命名空间_bpm1', { data:
input.applicationData }).interviewId;

    return out;
  }
}
```

其中，“命名空间_bpm1”为待创建工作流的名称，即[步骤六：创建审批工作流](#)中设置的值。

4. 单击，保存脚本，保存成功后单击，激活脚本。

步骤2 创建脚本createApplication，用于提交审批流数据，并将数据写入[步骤一：创建审批流对象](#)中创建的对象。

1. 在应用设计器中，选择“逻辑”，单击脚本后的“+”。
2. 新建一个空白的脚本，名称设置为“createApplication”，单击“添加”。

3. 在脚本编辑器中，输入示例代码。

```
import * as db from 'db';

@useObject(['命名空间_processDemo__CST'])

@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'object', required: true, label: 'applicationData' })
  applicationData: object;
  @action.param({ type: 'string', required: true, label: 'interviewID' })
  interviewID: string;
}

@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'string', required: true, label: 'id' })
  id: string;
}

@action.object({ type: "method" })
export class CreateBusiTrip {
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public run(input: ActionInput): ActionOutput {
    const out = new ActionOutput();
    let engine = db.object('命名空间_processDemo__CST');
    input.applicationData['命名空间_bpmlId__CST'] = input.interviewID;
    out.id = engine.insert(input.applicationData);
    return out;
  }
}
```

其中，“命名空间_processDemo__CST”为[步骤一：创建审批流对象](#)中创建的对象，“命名空间_bpmlId__CST”为对象中添加的字段。

4. 单击，保存脚本，保存成功后单击，激活脚本。

步骤3 创建脚本queryApplication，用于根据bpmlId查询审批流数据。

1. 在应用设计器中，选择“逻辑”，单击脚本后的“+”。
2. 新建一个空白的脚本，名称设置为“queryApplication”，单击“添加”。
3. 在脚本编辑器中，输入示例代码。

```
import * as bpm from 'bp';
import * as db from 'db';
@useObject(['命名空间_processDemo__CST'])
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'String', required: true, label: 'interviewID' })
  interviewID: string;
}

@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'Object', required: true })
  detail: object;
}

@action.object({ type: "method" })
export class QueryMyApproDetail {
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public run(input: ActionInput): ActionOutput {
    let out = new ActionOutput();
    out.detail = {};
    let instanceId = input.interviewID;
    const bp = bpm.newInstanceClient();
    let appliObj = db.object('命名空间_processDemo__CST');
    let conditions = [
      {
        "field": "命名空间_bpmlId__CST",
        "operator": db.Operator.eq,
        "value": input.interviewID
      }
    ];
    let applications = appliObj.queryByCondition({
      "conjunction": "AND",

```

```

"conditions": conditions
});
if (applications.length > 0) {
  input.interviewID;
  out.detail = {
    命名空间__apply__CST: applications[0].命名空间__apply__CST,
    命名空间__deadline__CST: applications[0].命名空间__deadline__CST,
    命名空间__type__CST: applications[0].命名空间__type__CST,
    命名空间__bpmId__CST: input.interviewID
  };
}
return out;
}
}

```

其中，“命名空间__processDemo__CST”为**步骤一：创建审批流对象**中创建的对象，“命名空间__bpmId__CST”、“命名空间__apply__CST”、“命名空间__deadline__CST”和“命名空间__type__CST”为对象中添加的字段。

4. 单击，保存脚本，保存成功后单击，激活脚本。

步骤4 创建脚本approve，用于处理审批任务。

1. 在应用设计器中，选择“逻辑”，单击脚本后的“+”。
2. 新建一个空白的脚本，名称设置为“approve”，单击“添加”。
3. 在脚本编辑器中，输入示例代码。

```

import * as bpm from 'bp';
import * as context from 'context';
import * as db from 'db';
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'String', required: true, label: 'state' })
  state: string;

  @action.param({ type: 'String', required: true, label: 'interviewID' })
  interviewID: string;
}
@action.object({ type: "method" })
export class ApproveTrip {
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public run(input: ActionInput) {
    let instanceId = input.interviewID;
    let taskClient = bpm.newTaskClient();
    let res = taskClient.queryByCondition({
      condition: {
        conjunction: db.Conjunction.AND,
        conditions: [
          {
            field: "processInstID",
            operator: db.Operator.eq,
            value: instanceId
          },
          {
            field: "assigneeID",
            operator: db.Operator.eq,
            value: context.getUserId()
          }
        ]
      }
    });
    if (res.length) {
      const taskId = res[0].id;
      var count;
      let vars = taskClient.getVars(taskId, ["pass"]);
      if (vars["pass"]) {
        count = vars["pass"];
      } else {
        count = 0;
      }
    }
  }
}

```



```

    }
    if (input.state == "PASS") {
      bpm.newTaskClient().complete(taskId, "{
        pass: count + 1
      }")
    } else {
      bpm.newTaskClient().complete(taskId, "{ }")
    }
  }
}
}
}

```

4. 单击 ，保存脚本，保存成功后单击 ，激活脚本。

步骤5 创建脚本getTask，用于获取审批流中的任务id。

1. 在应用设计器中，选择“逻辑”，单击脚本后的“+”。
2. 新建一个空白的脚本，名称设置为“getTask”，单击“添加”。
3. 在脚本编辑器中，输入示例代码。

```

import * as db from 'db';
import * as bp from 'bp';
import * as context from 'context';

@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'String', required: true, label: 'interviewID' })
  interviewID: string;
}

export class ActionOutput {
  @action.param({ type: 'String', required: true, label: 'taskId' })
  taskId: string;
}

@action.object({ type: "method" })
export class ApproveTrip {
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public run(input: ActionInput): ActionOutput {
    let out = new ActionOutput();
    let instanceId = input.interviewID;

    let taskClient = bp.newTaskClient()
    let res = taskClient.queryByCondition({
      condition: {
        conjunction: db.Conjunction.AND,
        conditions: [
          {
            field: "processInstID",
            operator: db.Operator.eq,
            value: instanceId
          },
          {
            field: "assigneeID",
            operator: db.Operator.eq,
            value: context.getUserId()
          }
        ]
      }
    });
    if (res.length > 0) {
      let taskId = res[0]['id'];
      out.taskId = taskId;
      console.log(taskId)
      return out;
    }

    return out;
  }
}

```

```
}
}
```

4. 单击，保存脚本，保存成功后单击，激活脚本。

步骤6 创建脚本getUserName，用于获取用户名。

1. 在应用设计器中，选择“逻辑”，单击脚本后的“+”。
2. 新建一个空白的脚本，名称设置为“getTask”，单击“添加”。
3. 在脚本编辑器中，输入示例代码。

```
import * as objectstorage from 'objectstorage';
import * as buffer from 'buffer';
import * as excel from 'excel';
import * as context from 'context';
import * as db from 'db';//导入处理object相关的标准库

//定义入参结构
@action.object({ type: "param" })
export class ActionInput {
}
//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', required: false, label: 'string' })
  res: string;
}
//使用数据对象命名空间__processDemo__CST
@useObject(['User'])
@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();
    try {
      let s = db.object('User'); //查询User表
      let condition = {
        "conjunction": "AND",
        "conditions": [{
          "field": "id",
          "operator": "eq",
          "value": context.getUserId()
        }]
      };
      let userRecod = s.queryByCondition(condition);
      if (userRecod) {
        out.res = userRecod[0].usrName;
        return out;
      }
    } catch (error) {
      console.error(error.name, error.message);
    }
    return out;
  }
}
```

4. 单击，保存脚本，保存成功后单击，激活脚本。

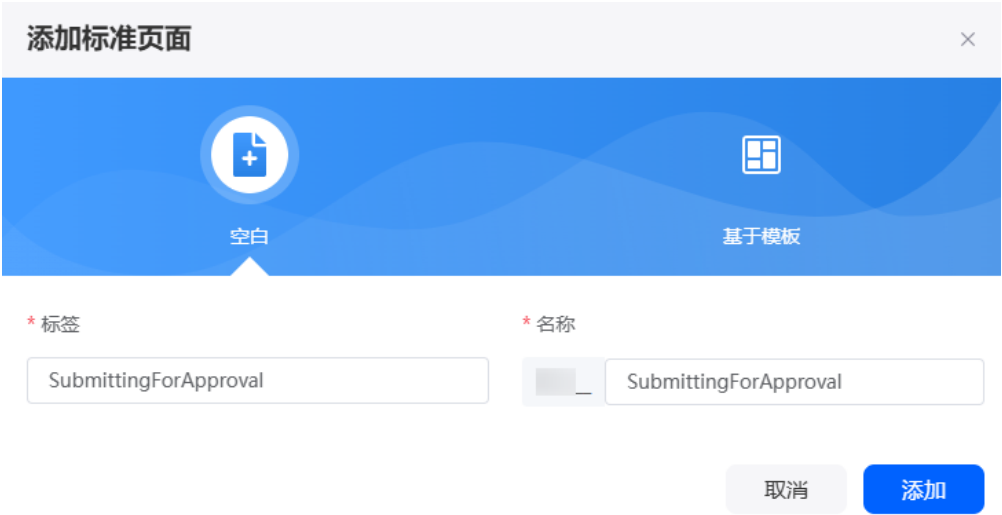
----结束

步骤三：创建提交申请页面

创建一个提交申请的标准页面，页面中包括申请标题、申请类型和申请时间三个字段，用于用户提交申请。

- 步骤1** 创建一个空白标准页面。
1. 在应用设计器中，选择“界面”，单击页面后的“+”。
 2. 输入页面的标签和名称，单击“添加”，新建一个标准页面。

图 5-40 创建提交申请标准页面



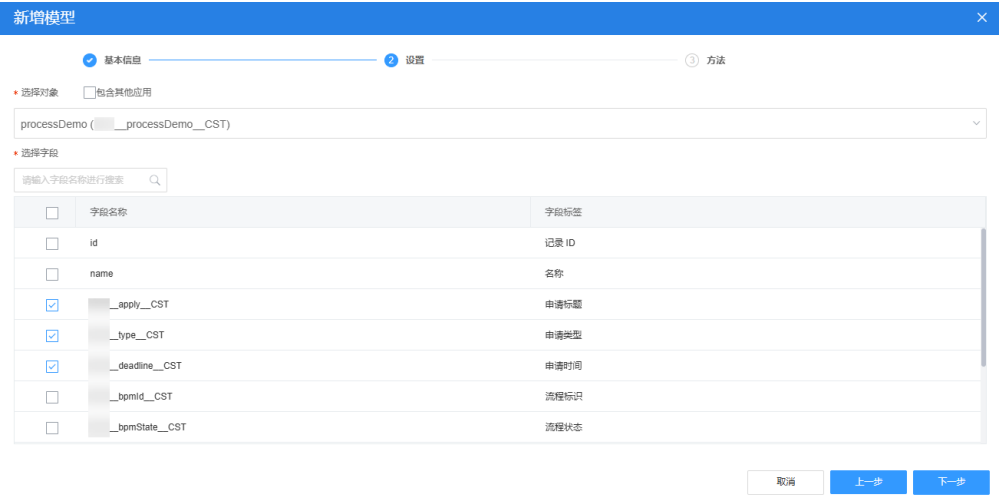
- 步骤2** 新建对象模型apply。
1. 在标准页面底部，单击“模型视图”，将页面从设计视图切换到模型视图。
 2. 单击“新增模型”，输入模型名称（如apply）、“来源”选择“对象”，单击“下一步”。

图 5-41 新建对象模型



3. 选择[步骤一：创建审批流对象](#)中创建的对象并选中apply、type和deadline字段，单击“下一步”，再单击“确定”，完成模型的创建。

图 5-42 选择对象和字段



步骤3 新建服务模型submit_bpm。

1. 在模型视图页面，单击“新增模型”，输入模型名称（如submit_bpm）、“来源”选择“服务”，单击“下一步”。

图 5-43 新建服务模型



2. 选择步骤1中创建的脚本，单击“确定”。

图 5-44 选择提交脚本

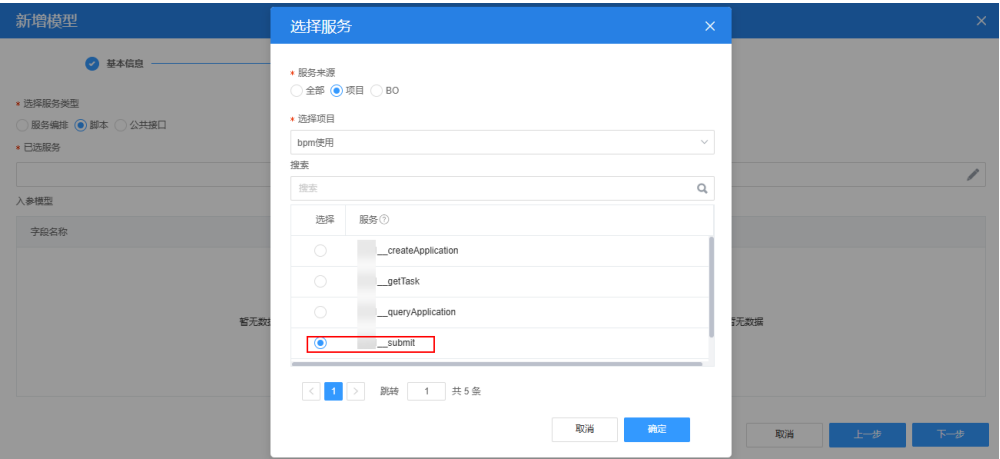
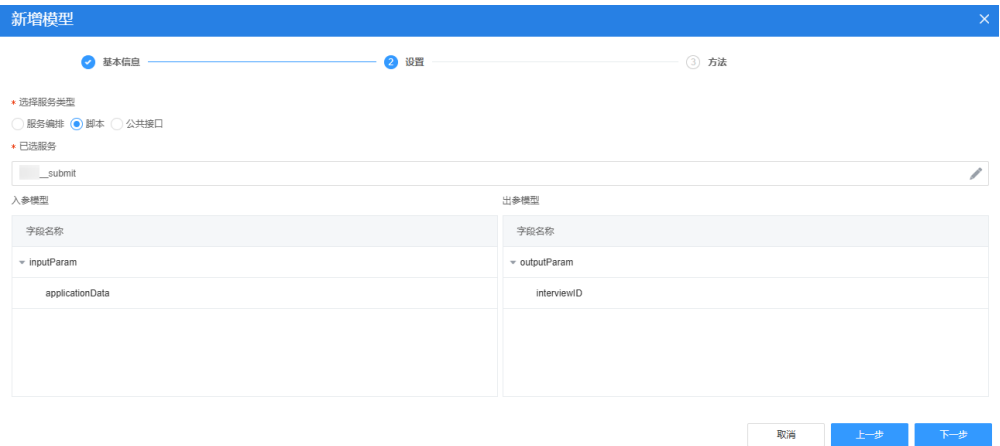
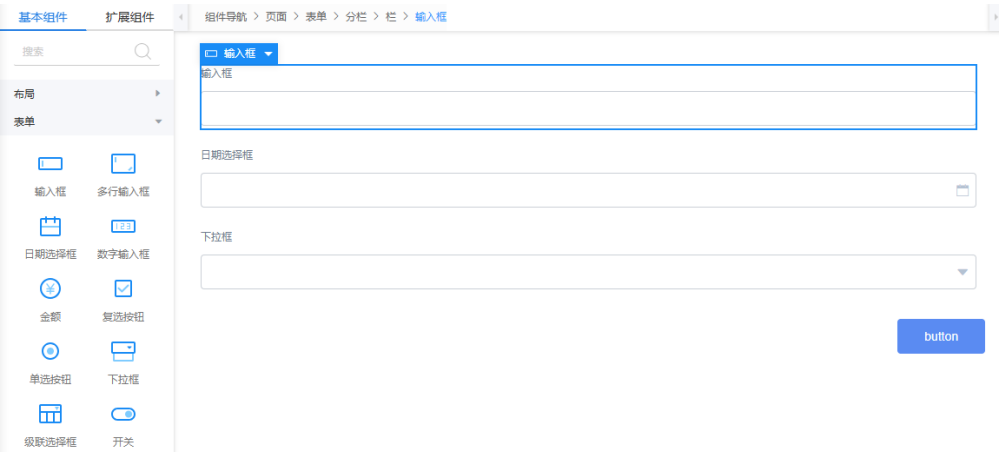


图 5-45 选择脚本后效果



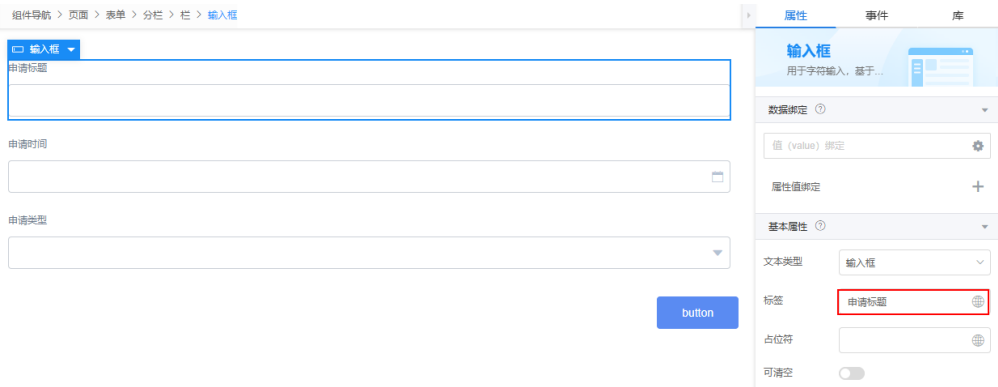
3. 单击“下一步”，再单击“确定”，完成模型的创建。
- 步骤4 返回设计视图页面，拖入输入框、日期选择框和下拉框，并绑定对应的对象模型。
1. 在标准页面底部，单击“设计视图”，将页面从模型视图切换到设计视图。
2. 在“基本组件”中，拖拽输入框、日期选择框、下拉框和按钮组件到画布中，布局图5-46。

图 5-46 提交审批标准页面布局



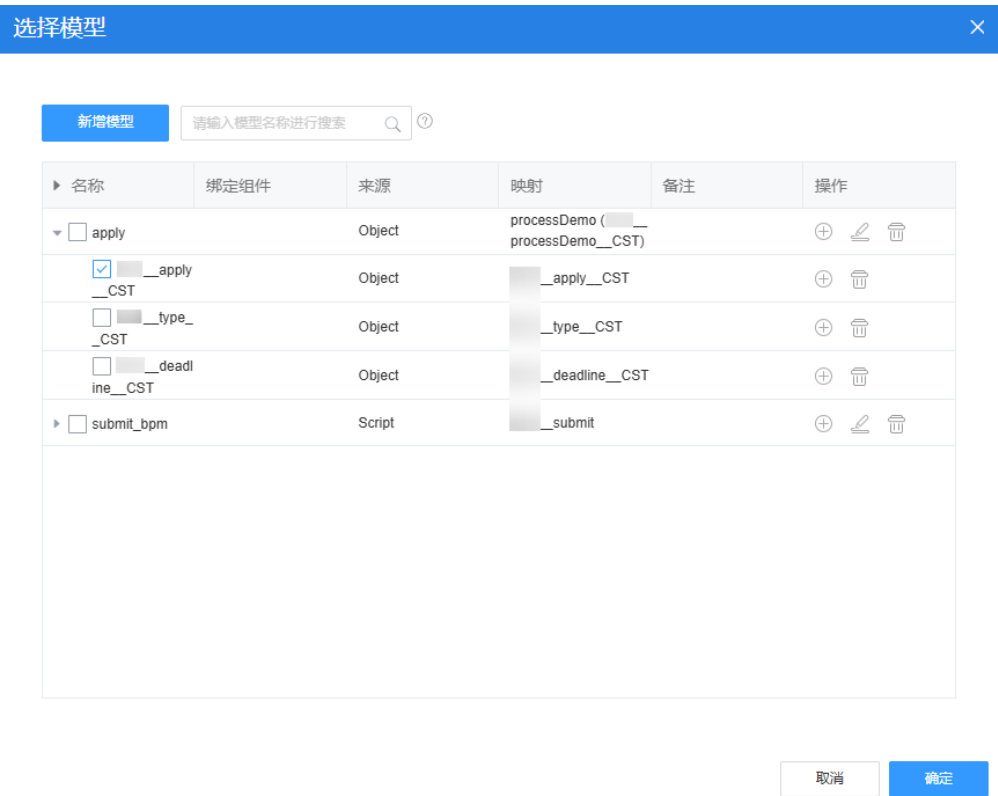
3. 选中组件，修改输入框、日期选择框、下拉框组件的标签为申请标题、申请时间和申请类型。

图 5-47 修改组件标题



4. 选中输入框组件，在“属性 > 数据绑定”中单击“值(value)绑定”后的⚙️，为组件绑定步骤2中对象模型的“命名空间__apply__CST”字段。

图 5-48 为组件绑定对象模型中的 apply 字段



5. 按照上述操作，分别为日期选择框和下拉框组件绑定步骤2中对象模型的“命名空间__deadline__CST”字段和“命名空间__type__CST”字段。

图 5-49 为组件绑定对象模型中的 deadline 字段

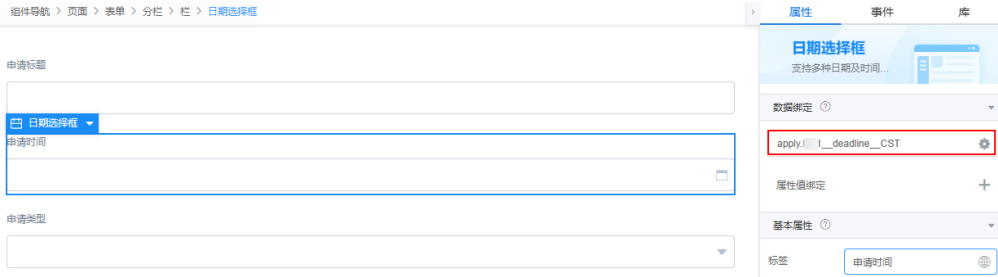
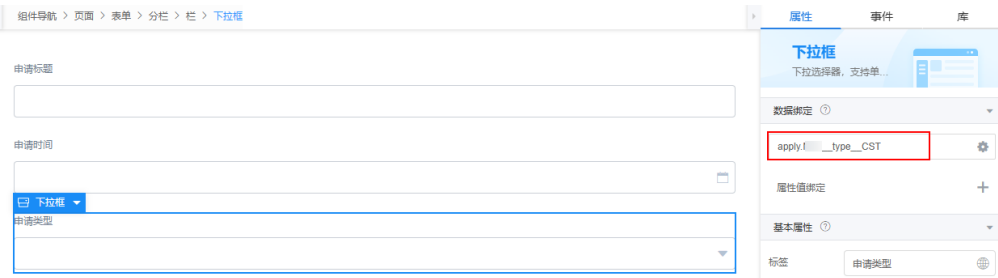


图 5-50 为组件绑定对象模型中的 type 字段



6. 选中下拉框组件，将“属性 > 基本属性 > 选项”的值设置为“出差”和“报销”。

图 5-51 设置选项值



7. 选中按钮组件，将“显示名称”修改为“提交申请”。
8. 在“事件”页签，单击“点击”后的“+”，为按钮添加如下事件。

```
var apply = $model.ref("apply").getData();
var submit = $model.ref("submit_bpm").getData();
submit.inputParam.applicationData = apply;
$model.ref("submit_bpm").run().then(function(data){
  context.$message.success('Submitted successfully.');
```

```
closeCurrentTab();
})
.catch(function (err) {
  console.log('error is', error);
  context.$message.error('Submission failed:' + error.resMsg);
});

function closeCurrentTab() {
  if (parent.bingo) {
    var removeTab = parent.bingo.removeTab;
    var getCurrentTab = parent.bingo.getCurrentTab;
    if (removeTab && getCurrentTab) {
      return removeTab(getCurrentTab());
    }
  }
  if (window.parent !== window) {
    window.parent.close();
  } else {
    window.close();
  }
}
```



```
}  
}
```

步骤5 单击页面上方的, 保存页面。

步骤6 保存成功后, 单击页面上方的, 预览效果。

----结束

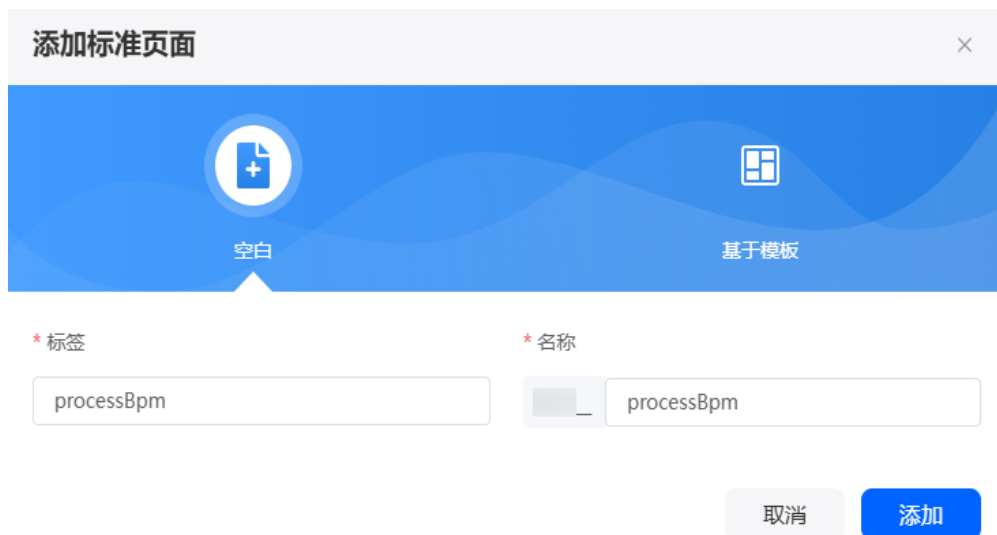
步骤四：创建审批数据表页面

创建一个标准页面, 用于呈现用户提交的审批数据, 并在操作列中增加一个审批按钮, 单击该按钮可进入审批页面。

步骤1 创建一个空白标准页面。

1. 在应用设计器中, 选择“界面”, 单击页面后的“+”。
2. 输入页面的标签和名称 (如processBpm), 单击“添加”, 新建一个标准页面。

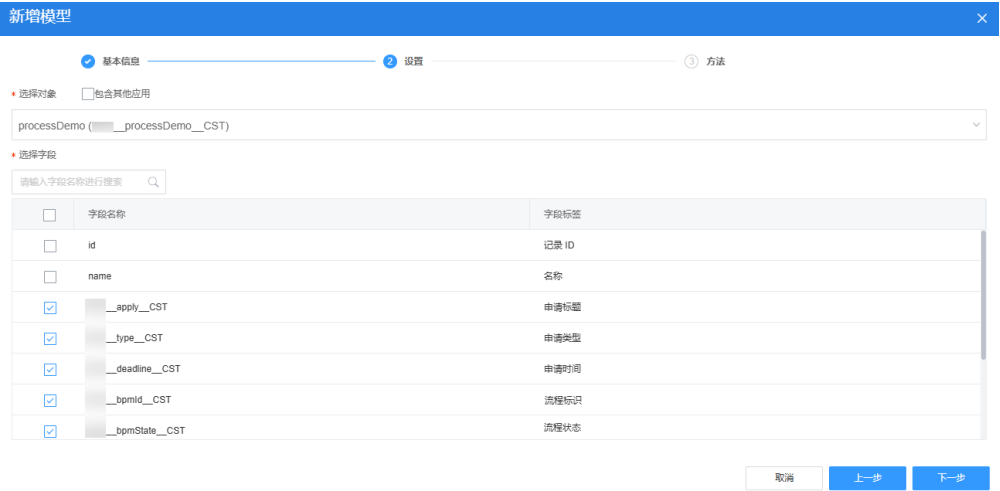
图 5-52 创建审批数据表标准页面



步骤2 新建对象模型processBpm。

1. 在标准页面底部, 单击“模型视图”, 将页面从设计视图切换到模型视图。
2. 单击“新增模型”, 输入模型名称 (如processBpm)、“来源”选择“对象”, 单击“下一步”。
3. 选择[步骤一：创建审批流对象](#)中创建的对象并选中apply、type、deadline、bpmId和bpmState字段, 单击“下一步”, 再单击“确定”, 完成模型的创建。

图 5-53 选择对象和字段



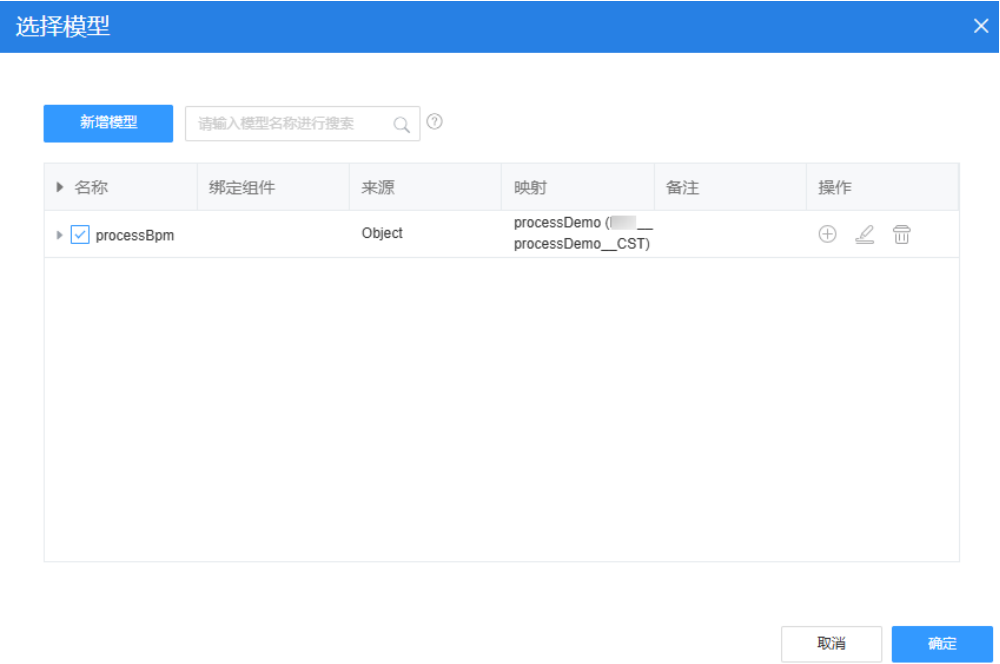
- 步骤3** 返回设计视图页面，拖入表格组件，并为组件绑定对象模型。
1. 在“基本组件”中，拖拽表格组件到画布中。

图 5-54 提交审批标准页面布局



2. 选中表格组件，在“属性 > 数据绑定”中单击“值(value)绑定”后的 ，为组件绑定 **步骤2** 中对象模型 processBpm。

图 5-55 为组件绑定对象模型 processBpm



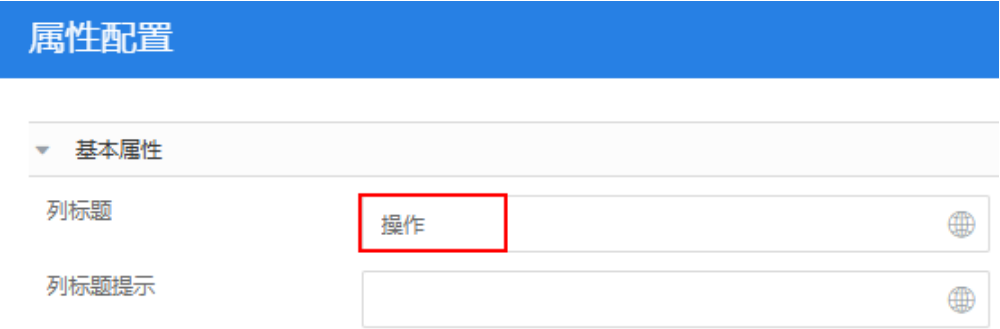
3. 选中表格组件，在“属性 > 表格列”中，单击，为表格添加操作列。

图 5-56 选择添加操作列



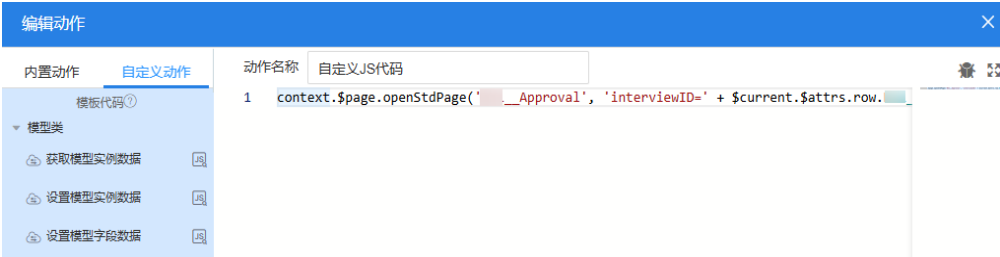
4. 单击新增操作列Operation1后的，修改列标题为“操作”。

图 5-57 修改列标题为操作



5. 在“操作按钮”中，单击“添加操作按钮”，修改“标签”为“审批”。
6. 单击审批按钮后的，再单击动作列表后的“+”，

图 5-58 为审批按钮添加自定义动作



context.\$page.openStdPage('命名空间_Approval', 'interviewID=' + \$current.\$attrs.row.命名空间_bpmId_CST);

其中，“命名空间_Approval”为步骤五：创建审批处理页面中创建的标准页面，“命名空间_bpmId_CST”为步骤一：创建审批流对象中添加的字段。

7. 在“禁用”中，输入如下代码，即审批通过后将审批的按钮置灰。

图 5-59 禁用设置



\$row.命名空间_bpmState_CST == "PASS" || \$row.命名空间_bpmState_CST == "FAIL"

其中，“命名空间_bpmState_CST”为步骤6中添加的对象字段。

步骤4 单击页面上方的，保存页面。

----结束

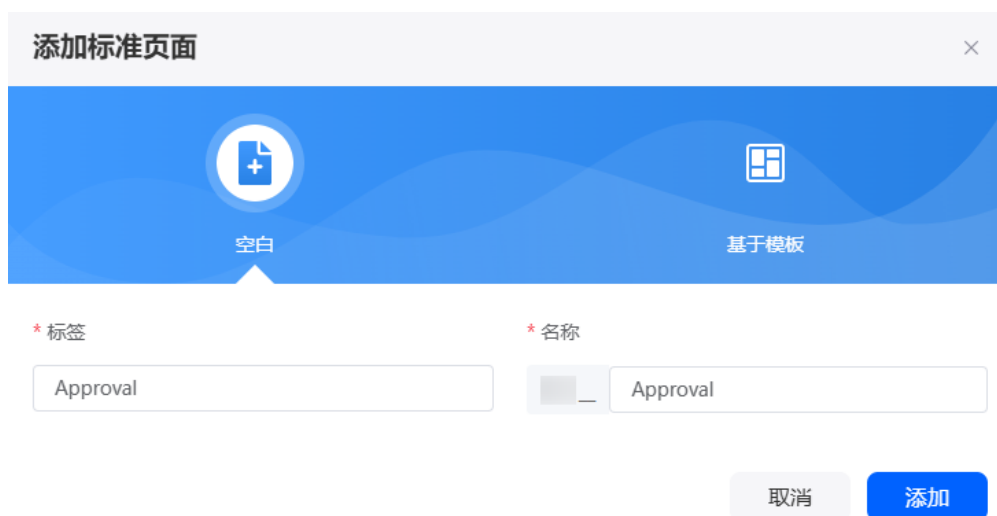
步骤五：创建审批处理页面

创建一个审批申请的标准页面，页面中包含申请标题、申请类型和申请时间三个字段，同时包括通过、不通过、经理通过和经理不通过四个按钮。

步骤1 创建一个空白标准页面。

1. 在应用设计器中，选择“界面”，单击页面后的“+”。
2. 输入页面的标签和名称（如Approval），单击“添加”，新建一个标准页面。

图 5-60 创建审批数据表标准页面



步骤2 新建脚本模型queryApplication，用于查询对应bpmlId的审批数据。

1. 在标准页面底部，单击“模型视图”，将页面从设计视图切换到模型视图。
2. 单击“新增模型”，输入模型名称（如queryApplication）、“来源”选择“服务”，单击“下一步”。
3. 选择步骤3中创建的脚本，单击“下一步”，再单击“确定”，完成模型的创建。

图 5-61 选中脚本

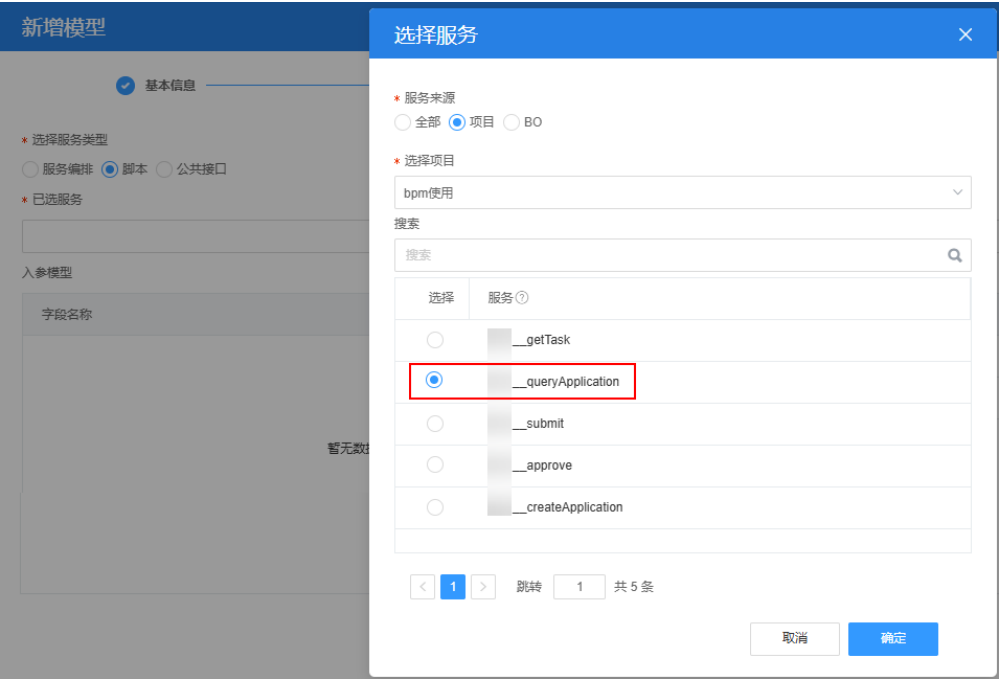
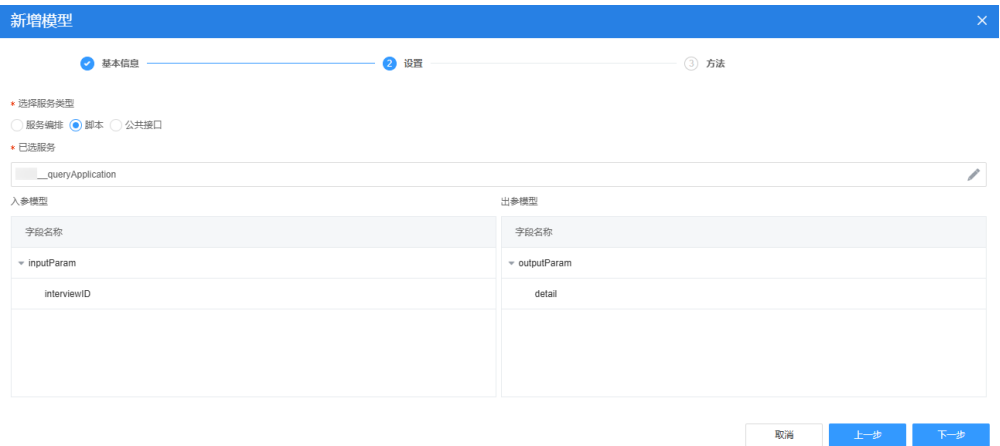


图 5-62 选中脚本后效果



- 步骤3 新建对象模型process。**
- 在模型视图页面，再次单击“新增模型”，输入模型名称（如process）、“来源”选择“对象”，单击“下一步”。
 - 选择[步骤一：创建审批流对象](#)中创建的对象并选中apply、type和deadline字段，单击“下一步”，再单击“确定”，完成模型的创建。

图 5-63 选择对象和字段

新增模型

1 基本信息

2 设置

3 方法

选择对象

☐ 包含其他应用

processDemo (processDemo__CST)

选择字段

请输入字段名称进行搜索

☐	字段名称	字段标签
☐	id	记录 ID
☐	name	名称
☒	__apply__CST	申请标题
☒	__type__CST	申请类型
☒	__deadline__CST	申请时间
☐	__bpmid__CST	流程标识
☐	__bpmState__CST	流程状态

取消

上一步

下一步

- 步骤4** 新建脚本模型approve，用于提交审批结果。
1. 在模型视图页面，再次单击“新增模型”，输入模型名称（如approve）、“来源”选择“服务”，单击“下一步”。

2. 选择步骤4中创建的脚本，单击“下一步”，再单击“确定”，完成模型的创建。

图 5-64 选择所需的脚本

新增模型

1 基本信息

2 设置

3 方法

选择服务类型

☐ 服务编排 ☒ 脚本 ☐ 公共接口

已选服务

入参模型

字段名称

暂无数

选择服务

服务来源

☐ 全部 ☒ 项目 ☐ BO

选择项目

bpm使用

搜索

搜索

选择

服务

☒ __approve

☐ __createApplication

☐ __getTask

☐ __queryApplication

< 1 >

跳转

1

共 5 条

取消

确定

图 5-65 设置后效果

新增模型

1 基本信息

2 设置

3 方法

选择服务类型

服务编排

脚本

公共接口

已选服务

__approve

入参模型

出参模型

字段名称
inputParam
state
interviewID

字段名称
outputParam

取消

上一步

下一步

- 步骤5** 新建脚本模型getTask，用于某个领导审批同意后该节点自动完成，忽略其他人的意见提交审批结果。
1. 在模型视图页面，再次单击“新增模型”，输入模型名称（如getTask）、“来源”选择“服务”，单击“下一步”。

2. 选择步骤5中创建的脚本，单击“下一步”，再单击“确定”，完成模型的创建。

图 5-66 选择所需的脚本

新增模型

选择服务

1 基本信息

2 设置

3 方法

选择服务类型

服务编排

脚本

公共接口

已选服务

入参模型

暂无数据

服务来源

全部

项目

BO

选择项目

bpm使用

搜索

搜索

选择

服务

__getTask

__queryApplication

__submit

__approve

__createApplication

< 1 >

跳转 1 共 5 条

取消

确定

图 5-67 设置后效果

新增模型

1 基本信息

2 设置

3 方法

选择服务类型

服务编排

☒ 脚本

公共接口

已选服务

__getTask

入参模型

出参模型

字段名称
inputParam
interviewID

字段名称
outputParam
taskId

- 步骤6** 新建脚本模型getUserName，用于获取用户名。
1. 在模型视图页面，再次单击“新增模型”，输入模型名称（如getUserName）、“来源”选择“服务”，单击“下一步”。

2. 选择步骤6中创建的脚本，单击“下一步”，再单击“确定”，完成模型的创建。

图 5-68 选择所需的脚本

新增模型

1 基本信息

2 设置

3 方法

选择服务类型

服务编排

☒ 脚本

公共接口

已选服务

入参模型

出参模型

字段名称
inputParam

字段名称
outputParam
res

选择服务

服务来源

全部

☒ 项目

BO

选择项目

bpm使用

搜索

搜索

选择

服务

☒ __getUserName

☐ __queryApplication

☐ __submit

☐ __approve

☐ __createApplication

☐ __getTask

< 1 >

跳转

1

共 6 条

取消

确定

图 5-69 设置后效果

新增模型

1 基本信息

2 设置

3 方法

选择服务类型

服务编排

☒ 脚本

公共接口

已选服务

__getUserName

入参模型

出参模型

字段名称
inputParam

字段名称
outputParam
res

步骤7 新增自定义模型managerFlag和approverFlag，用于控制按钮回显。

1. 在模型视图页面，单击“新增模型”，输入模型名称（如managerFlag）、“来源”选择“自定义”，单击“下一步”。
2. 单击“下一步”，再单击“确定”，完成模型的创建。
3. 按照上述操作，创建自定义模型“approverFlag”。

图 5-70 查看创建的模型

名称	绑定组件	来源	映射	备注	操作
process		Object	processDemo (processDemo_CST)		
queryApplication		Script	__queryApplication		
approve		Script	__approve		
getTask		Script	__getTask		
getUserName		Script	__getUserName		
managerFlag		-	-		
approverFlag		-	-		

步骤8 返回设计视图页面，拖入组件，并为组件绑定对象模型。

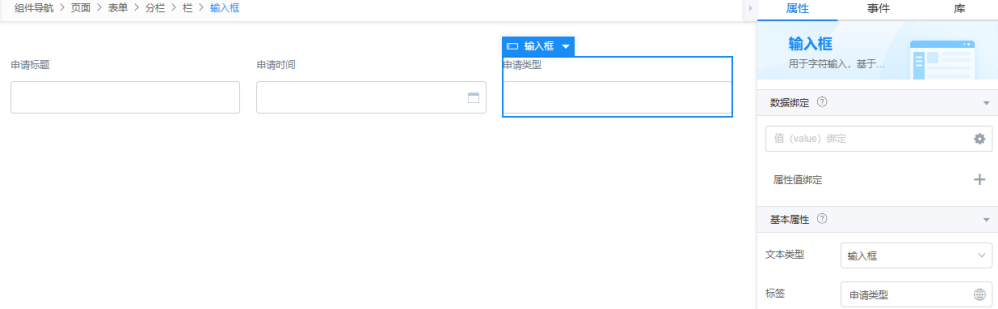
1. 在“基本组件”中，拖拽2个输入框和1个日期选择框组件到画布中。

图 5-71 审批处理标准页面布局



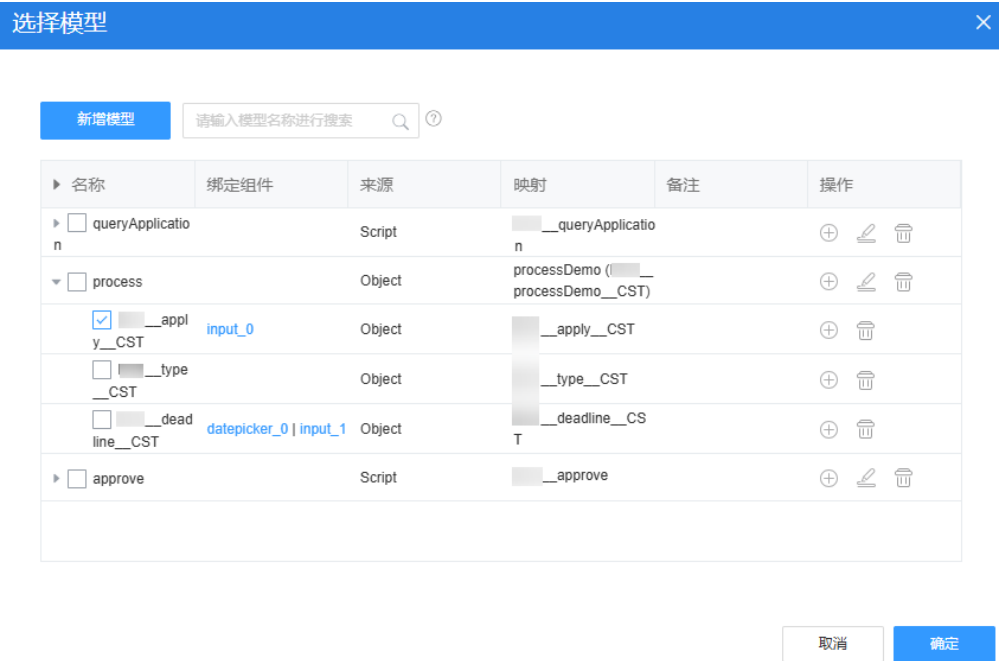
2. 选中组件，修改组件标签为申请标题、申请时间和申请类型。

图 5-72 修改组件标签



3. 选中第一个输入框组件，在“属性 > 数据绑定”中单击“值(value)绑定”后的⚙️，为组件绑定步骤3中对象模型的“命名空间__apply__CST”字段。

图 5-73 为组件绑定对象模型中的 apply 字段



4. 按照同样方法，为日期选择框组件绑定“命名空间__deadline__CST”字段，为第二个输入框绑定“命名空间__type__CST”字段。
5. 再拖入四个按钮组件到画布中，并修改按钮名称为“通过”、“驳回”、“经理通过”和“经理驳回”。

图 5-74 修改按钮名称



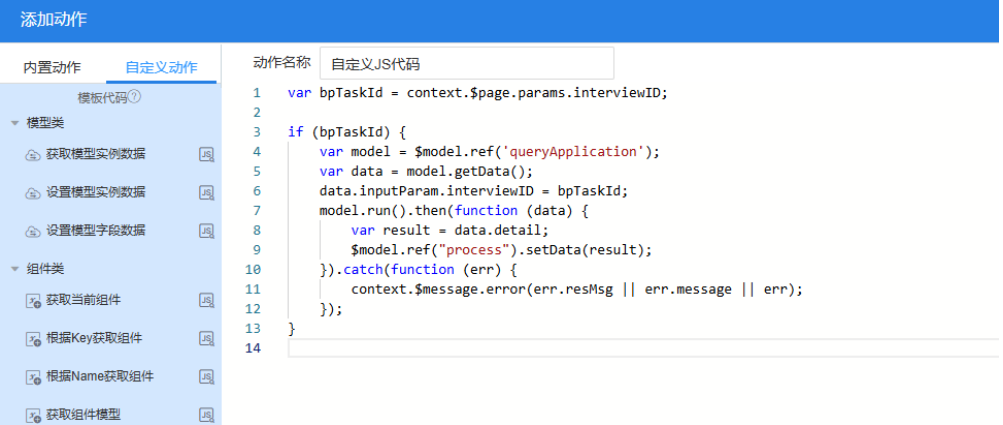
6. 选中页面组件，在“事件”中，单击加载后的“+”。

图 5-75 为页面添加事件



7. 在自定义动作中，输入如下代码。

图 5-76 输入事件代码



```
var bpTaskId = context.$page.params.interviewID;

if (bpTaskId) {
  var model = $model.ref('queryApplication');
  var data = model.getData();
  data.inputParam.interviewID = bpTaskId;
  model.run().then(function (data) {
    var result = data.detail;
    $model.ref("process").setData(result);
  }).catch(function (err) {
    context.$message.error(err.resMsg || err.message || err);
  });
}

//审批人列表
var approver = ["user_name1","user_name2"]
//经理列表
var manager = ["user_name3"]
$model.ref('getUserName').run().then(function(data){
  if (!approver.includes(data['res'])) {
    $model.ref("approverFlag").setData(true);
  }
  if (!manager.includes(data['res'])) {
    $model.ref("managerFlag").setData(true);
  } else {
    //如果是经理，把审批人的按钮也隐藏了
    $model.ref("approverFlag").setData(true);
  }
}).catch(function(error){
  console.log('error is', error);
  context.$message.error('Submission failed:' + error.resMsg);
});
```

其中，“queryApplication”为步骤2中创建的脚本模型，“user_name1”、“user_name2”为审批人名称，“user_name3”为经理审批人名称。

- 8. 选中“通过”按钮，在事件页签中，单击“点击”后的“+”，为按钮添加自定义动作。

```
var data = $model.ref('process').getData();
var approve = $model.ref('approve').getData();
approve.inputParam.interviewID = data['命名空间_bpmlId_CST'];
approve.inputParam.state = 'PASS';
$model.ref('approve').run().then(function(data){
  context.$message.success('Submitted successfully.');
```

```
var removeTab = parent.bingo.removeTab;
var getCurrentTab = parent.bingo.getCurrentTab;
if (removeTab && getCurrentTab) {
    return removeTab(getCurrentTab());
}
}
if (window.parent != window) {
    window.parent.close();
} else {
    window.close();
}
}
```

其中，“process”为步骤3中创建的对象模型，“approve”为步骤4创建脚本模型。“命名空间_bpmlid_CST”为步骤一：创建审批流对象中添加的字段。

- 再次选中“通过”按钮，在“属性”中单击“属性值绑定”后的“+”，为组件绑定步骤7中自定义的模型“approverFlag”。

图 5-77 为通过按钮绑定 approverFlag 模型



- 选中“驳回”按钮，在事件页签中，单击“点击”后的“+”，为按钮添加自定义动作。

```
var data = $model.ref('process').getData();
var approve = $model.ref('approve').getData();
approve.inputParam.interviewID = data['命名空间_bpmlid_CST'];
approve.inputParam.state = 'FAIL';
$model.ref('approve').run().then(function(data){
    context.$message.success('Submitted successfully. ');
    closeCurrentTab();
}).catch(function(error){
    console.log('error is', error);
    context.$message.error('Submission failed:' + error.resMsg);
});
function closeCurrentTab() {
    if (parent.bingo) {
        var removeTab = parent.bingo.removeTab;
        var getCurrentTab = parent.bingo.getCurrentTab;
        if (removeTab && getCurrentTab) {
            return removeTab(getCurrentTab());
        }
    }
    if (window.parent != window) {
        window.parent.close();
    } else {
        window.close();
    }
}
```

其中，“process”为步骤3中创建的对象模型，“approve”为步骤4创建的脚本模型。“命名空间_bpmlid_CST”为步骤一：创建审批流对象中添加的字段。

- 再次选中“驳回”按钮，在“属性”中单击“属性值绑定”后的“+”，为组件绑定步骤7中自定义的模型“approverFlag”。

图 5-78 为驳回按钮绑定 approverFlag 模型



12. 选中“经理通过”按钮，在事件页签中，单击“点击”后的“+”，为按钮添加自定义动作。

```
var dataTask = $model.ref('getTask').getData();
dataTask.inputParam.interviewID= context.$page.params.interviewID;
$model.ref('getTask').run().then(function(data){
  context.$message.success('Submitted successfully.');
```

```

  console.log(data);
  var taskId = data.taskId;
  var _url = `/u-route/baas/bp/v2.0/runtime/tasks/${taskId}`;
  var _inputParam = {action:'经理审批', variables:{pass:2}};
  var _method = 'PUT';
  var _header = {'Content-Type':'application/json'};
  context.service(_url).run(_inputParam,_method,_header).then(function(response){
    console.log("success");
  });
  closeCurrentTab();
}).catch(function(error){
  console.log('error is', error);
  context.$message.error('Submission failed:' + error.resMsg);
});

function closeCurrentTab() {
  if (parent.bingo) {
    var removeTab = parent.bingo.removeTab;
    var getCurrentTab = parent.bingo.getCurrentTab;
    if (removeTab && getCurrentTab) {
      return removeTab(getCurrentTab());
    }
  }
  if (window.parent !== window) {
    window.parent.close();
  } else {
    window.close();
  }
}
```

其中，“getTask”为步骤5中创建的脚本模型。

13. 再次选中“经理通过”按钮，在“属性”中单击“属性值绑定”后的“+”，为组件绑定步骤7中自定义的模型“managerFlag”。

图 5-79 为经理通过按钮选择 managerFlag 模型



14. 选中“经理驳回”按钮，在事件页签中，单击“点击”后的“+”，为按钮添加自定义动作。
- ```
var dataTask = $model.ref('getTask').getData();
dataTask.inputParam.interviewID= context.$page.params.interviewID;
$model.ref('getTask').run().then(function(data){
 context.$message.success('Submitted successfully. ');
 console.log(data);
 var taskId = data.taskId;
 var _url = `/u-route/baas/bp/v2.0/runtime/tasks/${taskId}`;
 var _inputParam = {action:'经理审批', variables:{pass:0}};
 var _method = 'PUT';
 var _header = {'Content-Type':'application/json'};
 context.service(_url).run(_inputParam,_method,_header).then(function(response){
 console.log("success");
 });
 closeCurrentTab();
}).catch(function(error){
 console.log('error is', error);
 context.$message.error('Submission failed:' + error.resMsg);
});

function closeCurrentTab() {
 if (parent.bingo) {
 var removeTab = parent.bingo.removeTab;
 var getCurrentTab = parent.bingo.getCurrentTab;
 if (removeTab && getCurrentTab) {
 return removeTab(getCurrentTab());
 }
 }
 if (window.parent != window) {
 window.parent.close();
 } else {
 window.close();
 }
}
```
- 其中，“getTask”为步骤5中创建的脚本模型。
15. 再次选中“经理驳回”按钮，在“属性”中单击“属性值绑定”后的“+”，为组件绑定步骤7中自定义的模型“managerFlag”。

图 5-80 为经理驳回按钮选择 managerFlag 模型



步骤9 单击页面上方的，保存页面。

----结束

步骤六：创建审批工作流

创建一个审批申请的工作流，通过配置用户任务的审批动作和终止动作，实现优先级审批功能。

步骤1 在应用设计器的左侧导航栏中，选择“流程”，单击工作流后的“+”。

**步骤2** 输入工作流的标签和名称（如bpm1），单击“添加”。

**步骤3** 创建变量和对象变量。

1. 单击 ，进入全局上下文页面。
2. 单击变量后的“+”，创建一个变量variable0。
3. 单击变量后的 ，选择“设置”，修改“名称”为“pass”，“数据类型”选择“数字”。

图 5-81 创建变量 pass

变量

\* 名称

pass

变量名是用于变量在流程中引用的唯一标识。修改变量名不会改变图元中的引用，可能导致流程不可用。

\* 数据类型

数字

默认值 ?

请输入常量或从列表中选择

描述

请输入

是否为数组

☐

外部使用 ?

☐

取消

保存

4. 在全局上下文中，单击对象变量后的“+”，创建一个对象变量。



图 5-82 创建对象变量 data

对象变量

对象

全局结构体

事件

\* 名称

data

变量名是用于变量在流程中引用的唯一标识。修改变量名不会改变图元中的引用，可能导致流程不可用。

\* 对象

\_\_processDemo\_\_CST

默认值

请输入常量或从列表中选择

描述

请输入

是否为数组

外部使用

取消

保存

表 5-7 新建 data 对象变量

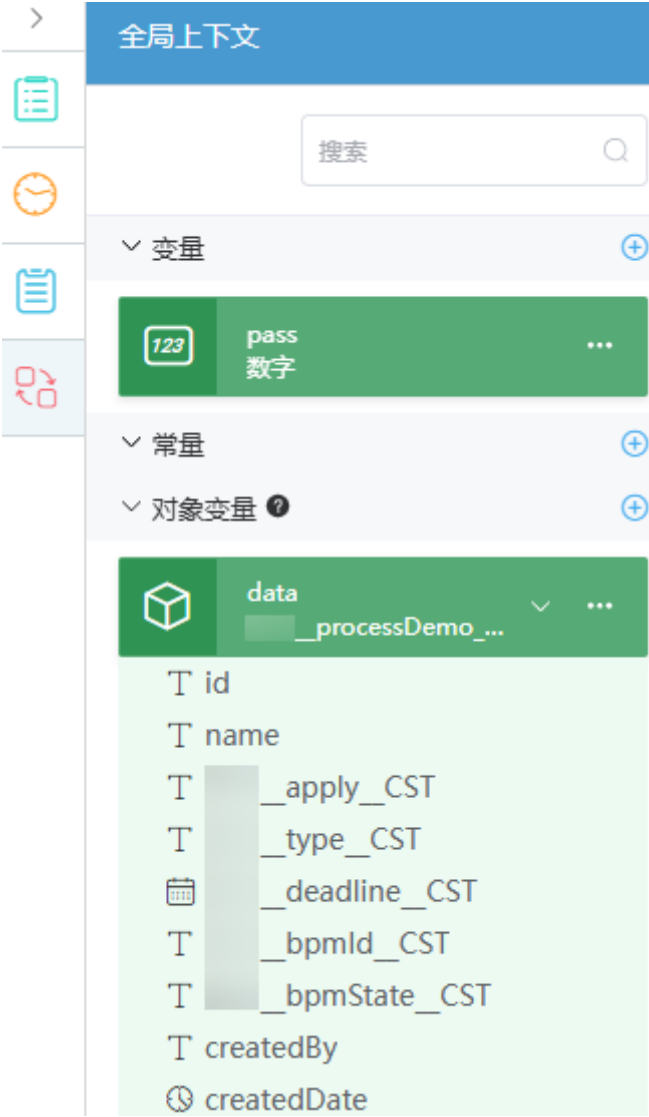
| 参数 | 说明                                                                        | 示例                     |
|----|---------------------------------------------------------------------------|------------------------|
| 名称 | 新建对象变量的名称，变量名是变量在流程中引用的唯一标识。修改变量名不会改变图元中的引用，但是可能导致流程不可用。<br>取值范围：1~80个字符。 | data                   |
| 对象 | 在下拉框中，选择具体的对象，本示例选择 <a href="#">步骤一：创建审批流对象</a> 中创建的对象。                   | 命名空间__processDems__CST |

文档版本 01 (2025-09-04)

版权所有 © 华为云计算技术有限公司

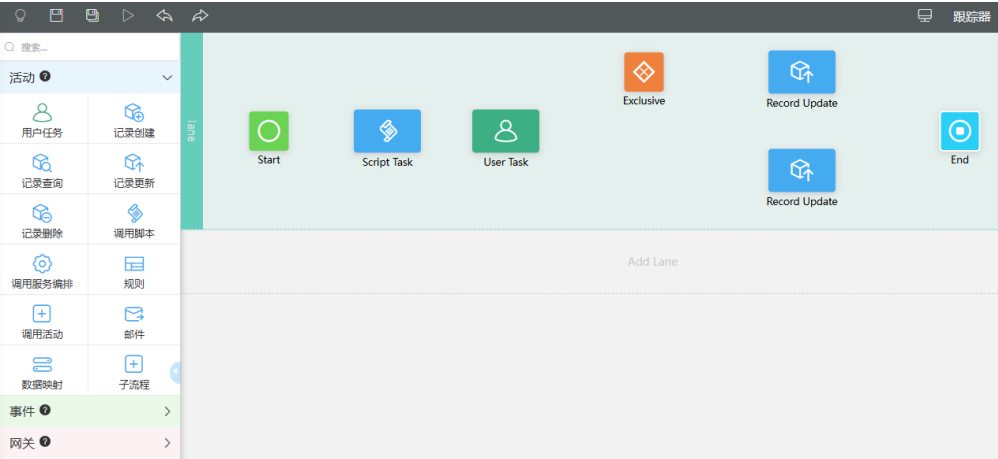
161

图 5-83 查看已创建变量



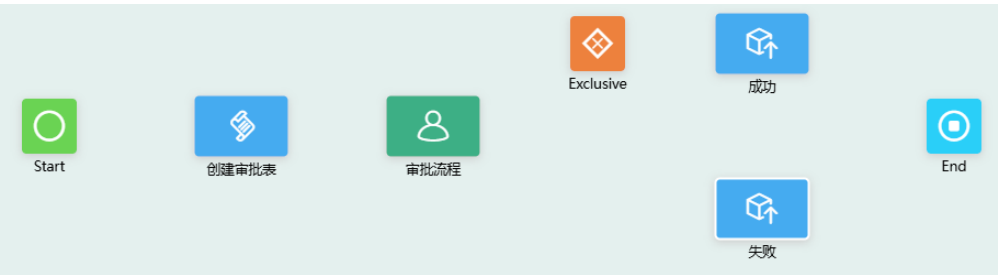
- 步骤4** 创建审批工作流。
- 按照图5-84，为工作流添加所需的图元。

图 5-84 审批 workflows 布局



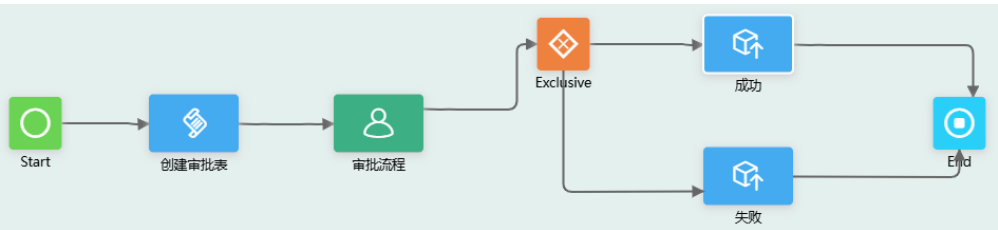
2.
- 分别选中脚本图元、用户任务图元、记录更新图元，单击 ，修改图元标签为创建审批表、审批流程、成功和失败。

图 5-85 修改图元标签



3.
- 连接图元指定图元的逻辑关系，如 [图5-86](#)。  
根据逻辑实现，需要在拖入并配置好所有图元后，将图元按照逻辑顺序有序的连接起来。在执行工作流时，系统会根据连线顺序有序执行图元任务，从而实现整个流程运转。

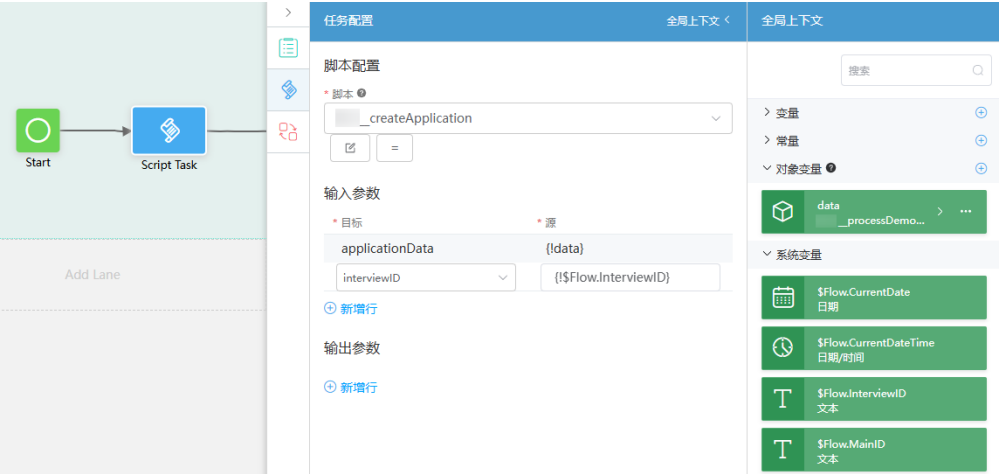
图 5-86 指定逻辑关系



步骤5 配置工作流中的图元。

1.
- 选中脚本图元，单击 ，按照 [图5-87](#) 设置脚本图元。

图 5-87 设置脚本图元



2. 选中用户任务图元，单击，按照图5-88设置用户图元。

图 5-88 设置用户图元

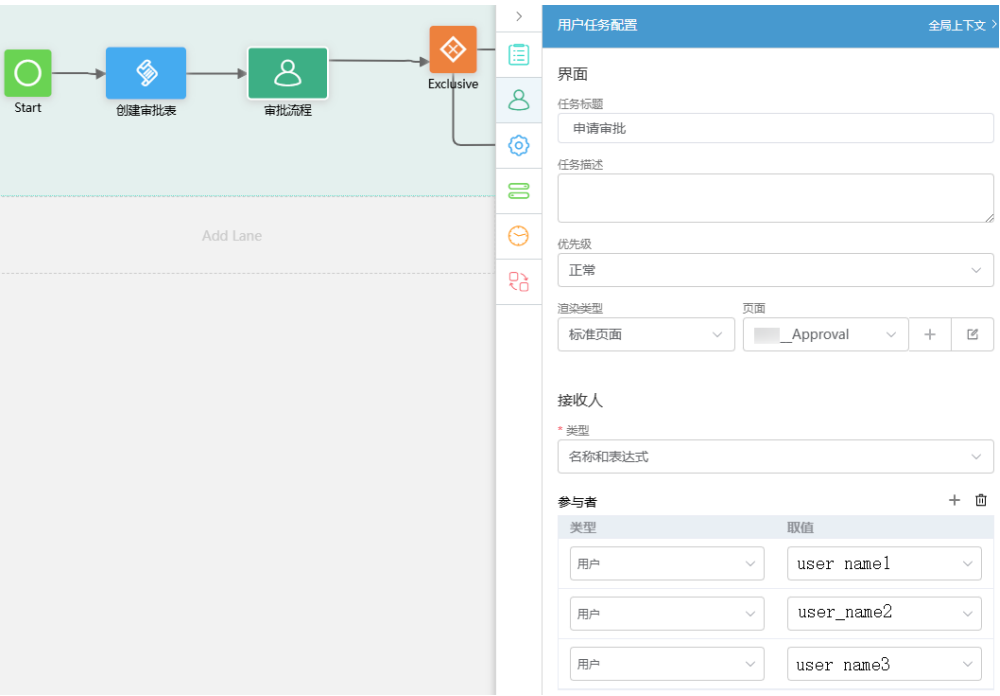


图 5-89 审批配置

审批配置

审批类型 ?

会签：每个分配人都需要审批

☐ 如果审批人中包含群组，则群组中任意一人审批即认为该群组已经审批

投票结果门槛 ?

66

默认结果 ?

审批通过

结果触发方式

☒ 当最小百分比满足时立即触发投票结果

☐ 等待所有投票完成触发投票结果

审批动作 ?

审批通过,经理审批

当存在以下动作时立即终止任务

经理审批

自动审批

☐ 流程触发人自动通过

☐ 已审批过该流程的审批人自动通过

表 5-8 用户任务参数说明

| 参数        | 说明                         | 示例                                                      |
|-----------|----------------------------|---------------------------------------------------------|
| 界面 > 任务标题 | 显示在任务界面的标题。                | 申请审批                                                    |
| 界面 > 渲染类型 | 用户处理的界面，可以是标准页面、标准表单或高级页面。 | 标准页面                                                    |
| 界面 > 页面   | 根据渲染类型，选择具体的页面。            | 选择 <a href="#">步骤五：创建审批处理页面</a> 中创建的页面“命名空间__Approval”。 |

| 参数                    | 说明                                                                                                                                                                                        | 示例                                     |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| 接收人 > 类型              | 待处理任务的用户类型。                                                                                                                                                                               | 名称和表达式                                 |
| 接收人 > 参与者 > 类型        | 当“类型”为“名称和表达式”时，该参数才会显示。参与者类型可以是用户、业务用户、组或者表达式。                                                                                                                                           | 用户                                     |
| 接收人 > 参与者 > 取值        | 选择接收该任务的用<br>户，不包括业务用户。                                                                                                                                                                   | user_name1<br>user_name2<br>user_name3 |
| 审批配置 > 审批类型           | 选择审批的类型。<br><ul style="list-style-type: none"> <li>- 或签：任一个分配人均可审批<br/>只需要接收人中的某一用户完成了审批，即可推动任务流程，走向下一个任务。</li> <li>- 会签：每个分配人都需要审批<br/>需要接收人中的用户群体满足您设置的审批条件才可推动任务流程，走向下一个任务。</li> </ul> | 会签：每个分配人都需要审批                          |
| 审批配置 > 投票结果门槛         | 该参数取值表示百分比，如果达到设置的百分比，审批结果中最高的得票结果将覆盖“\$BP.TaskOutcome”系统变量，即将执行最高得票的审批结果。                                                                                                                | 66                                     |
| 审批配置 > 默认结果           | 审批百分比（已审批的人数/总的有效审批人数）未达到“投票结果门槛”百分比时，默认的审批结果。当审批类型为“会签：每个分配人都需要审批”时，该参数才会显示。                                                                                                             | 审批通过                                   |
| 审批配置 > 审批动作           | 设定一些候选的动作，作为默认审批结果。                                                                                                                                                                       | 审批通过,经理审批                              |
| 审批配置 > 当存在如下动作时立即终止任务 | 当存在设定的某个或者某些动作时，立即终止任务。                                                                                                                                                                   | 经理审批                                   |

3. 选中排他网关和成功图元之间的连接，并做如下设置。

图 5-90 设置排他网关和成功图元之间的连线



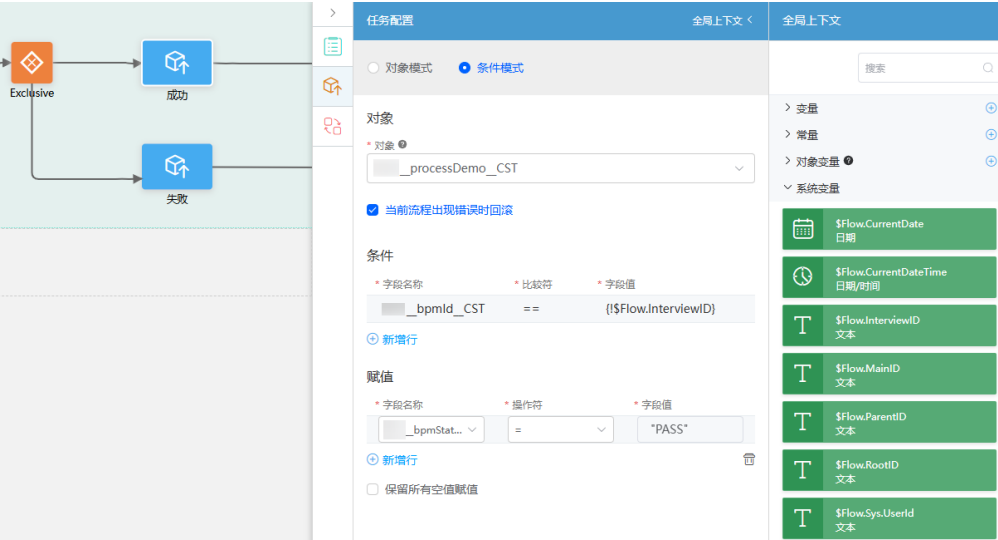
4. 选中排他网关和失败图元之间的连接，并做如下设置。

图 5-91 设置排他网关和失败图元之间的连线



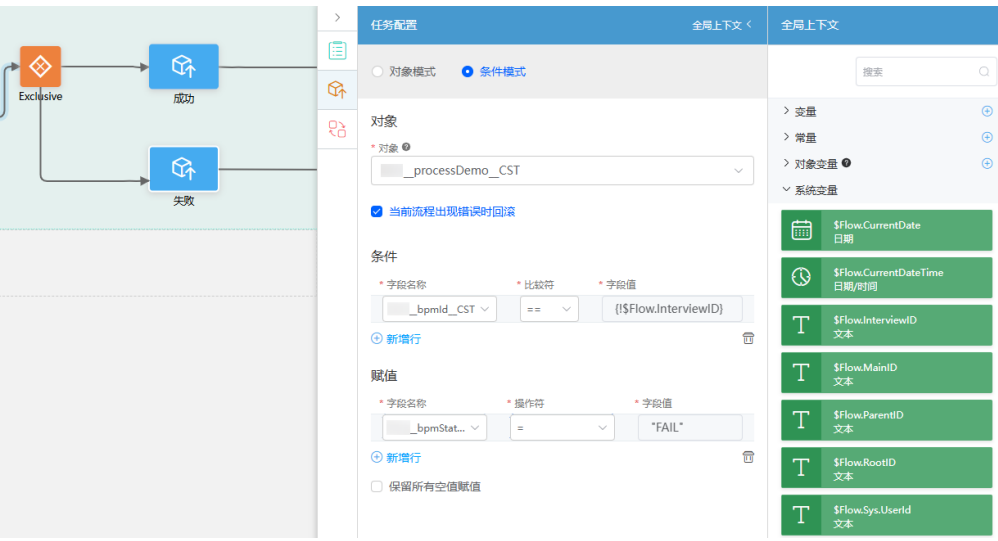
5. 选中成功图元，按照图5-92设置成功图元。

图 5-92 设置成功图元



6. 选中失败图元，按照图5-93设置失败图元。

图 5-93 设置失败图元



步骤6 单击页面上方的，保存工作流。

步骤7 保存成功后，单击，启用工作流。

----结束

步骤七：设置应用菜单并验证功能

为应用添加一个“我要出差”和“我要待办”菜单，并验证工作流是否按照预设的流程执行。

步骤1 应用导航栏设置。

1. 进入应用设计器，在“开始”中，单击“应用导航设置”。



2. 在主导航设置页签中，单击“新建”，新建一个“我要出差”菜单。

图 5-94 添加我要出差菜单

新建

\* 菜单属性

☒ 菜单项

☐ 分组

菜单详情

\* 菜单名称

我要出差

\* 唯一标识

chucha

描述

菜单图标

修改

链接目标

无

页面

流程

外部链接

\* 页面

SubmittingForApproval

...

是否展现

☒ 桌面端

☐ 移动端

可见权限

所有人默认可见

(当用户拥有以下权限时能看到该菜单项)

取消

保存

3. 按照上述操作，添加一个我的待办菜单。

图 5-95 添加我的待办菜单

新建

\* 菜单属性

☒ 菜单项

☐ 分组

菜单详情

\* 菜单名称

我的待办

\* 唯一标识

daiban

描述

菜单图标

修改

链接目标

×

无

页面

流程

外部链接

\* 页面

processBpm

...

是否展现

☒ 桌面端

☐ 移动端

可见权限

所有人默认可见

+

(当用户拥有以下权限时能看到该菜单项)

取消

保存

- 步骤2 员工提交出差申请。
1. 在应用主菜单中，单击“运行 > 立即运行”，进入应用预览页面。

2. 在“我要出差”菜单中，提交出差申请。

图 5-96 提交出差申请

Home x

我要出差 x

Home

我要出差

我的待办

Submitted successfully

申请标题

张三出差申请

申请时间

2024-12-16

申请类型

出差

提交申请

提交成功后，界面会提示“Submitted successfully”。

- 步骤3 主管审批申请。
1. 在应用主菜单中，单击“运行 > 立即运行”，进入应用预览页面。

2. 在“我的待办”菜单中，单击待审批数据后的“审批”，进入审批页面。

图 5-97 单击审批



3. 审批申请。
- 本实践中提交的申请，需要[步骤5.2](#)中配置的user\_name1、user\_name2和user\_name3三个用户审批。其中，user\_name3为经理角色，具有优先审批权，即user\_name3审批后无需user\_name1、user\_name2再审批。
- 假设，张三提交了一个出差申请，正常情况下user\_name1审批后，user\_name2再审批，最后user\_name3（经理）审批，user\_name3审批后整个流程审批结束。如果张三申请后，user\_name3（经理）直接进行了审批，则审批流程结束，无需user\_name1和user\_name2再审批，因为user\_name3具有优先审批权。

图 5-98 正常审批界面



图 5-99 经理审批界面



----结束

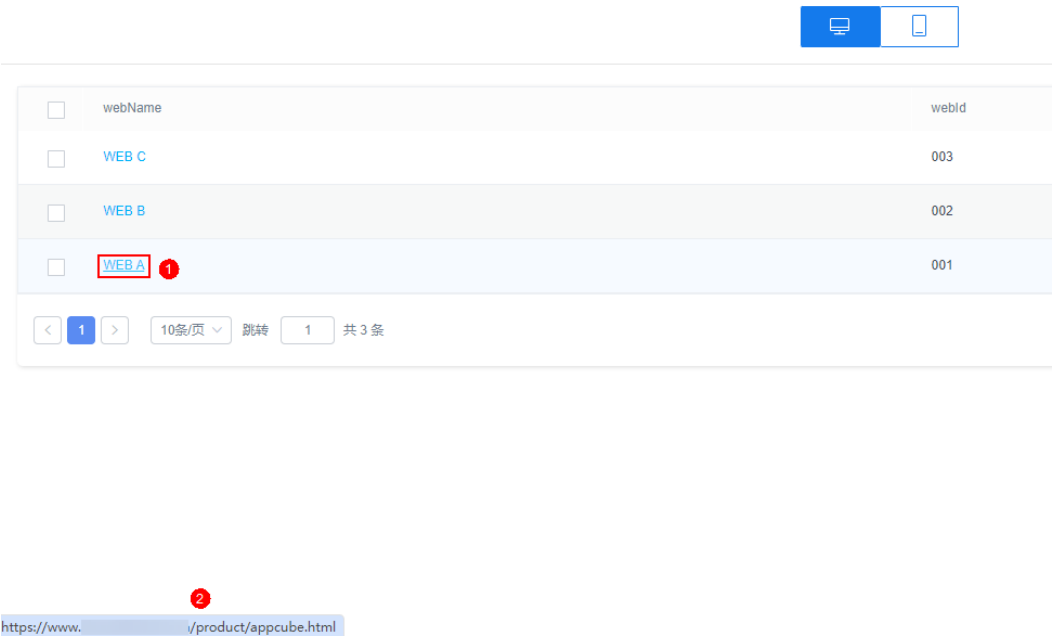
# 6 标准页面专项

## 6.1 为华为云 Astro 轻应用标准页面中表格的数据增加链接

### 期望实现效果

在标准页面中，支持为表格中的数据增加超链接，来提升用户体验和数据交互的便捷性。例如，在表格的webName列中，将鼠标移动至WEB A上在页面的左下角可查看到对应的链接地址，单击会跳转到对应的页面。

图 6-1 实现效果



### 功能实现方法

- 步骤1** 创建一个低代码应用。
1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。

2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
3. 在“应用”中，单击“新建低代码应用”或单击，进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。
4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。
5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 6-2 创建一个空白应用

新建 空白应用

基本信息



\* 标签

我的第一个应用

\* 名称

A

分类

请选择

描述

请输入

高级设置

取消

新建

表 6-1 新建空白应用参数说明

| 参数 | 说明                                                                                                                                                                                                                                                                                   | 示例      |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 标签 | 新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。                                                                                                                                                                                                                                         | 我的第一个应用 |
| 名称 | 新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下： <ul style="list-style-type: none"><li>长度不能超过31个字符，包括前缀命名空间的长度。</li><li>名称前的内容为命名空间，在华为云Astro轻应用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。</li><li>必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。</li></ul> | A       |

文档版本 01 (2025-09-04)

版权所有 © 华为云计算技术有限公司

173

步骤2 创建对象“websiteList”，并为对象添加字段。

1. 在应用设计器的左侧导航栏中，选择“数据”，单击对象中的“+”。
2. 设置对象的名称和唯一标识为“websiteList”，单击“确定”。

图 6-3 新建对象 websiteList



表 6-2 新建 websiteList 对象参数说明

| 参数   | 说明                                                                                                                                                                                                                                                | 示例          |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 对象名称 | 新建对象的名称，创建后可修改。<br>取值范围：1~80个字符。                                                                                                                                                                                                                  | websiteList |
| 唯一标识 | 新建对象在系统中的标识，创建后不支持修改。命名要求如下： <ul style="list-style-type: none"><li>长度不能超过63个字符，包括前缀命名空间的长度。<br/>标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。</li><li>必须以英文字母开头，只能由英文字母，数字和下划线组成，且不能以下划线结尾。</li></ul> | websiteList |

3. 在已创建的对象中，单击，进入对象详情页面。
4. 在“字段”页签，单击“添加”，为对象添加webName字段。

图 6-4 添加 webName 字段

添加字段

\* 显示名称

webName

\* 唯一标识

webName

\* 字段类型

文本

\* 数据长度

64

描述

取消

确认

表 6-3 添加 webName 字段参数说明

| 参数   | 说明                                                                                                                                                   | 示例      |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 显示名称 | 新建字段的名称，创建后可修改。<br>取值范围：1~63个字符。                                                                                                                     | webName |
| 唯一标识 | 新建字段在系统中的标识，创建后不支持修改。命名要求如下： <ul style="list-style-type: none"><li>长度不能超过63个字符，包括前缀命名空间的长度。</li><li>必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。</li></ul> | webName |
| 字段类型 | 单击***，在弹出的页面中，根据页面提供的参数解释，选择新建字段所属的类型。                                                                                                               | 文本      |

5. 按照上述操作，为对象添加webLink和webId字段。

表 6-4 添加 webLink 和 webId 字段

| 显示名称    | 唯一标识    | 字段类型 |
|---------|---------|------|
| webLink | webLink | 文本区  |

| 显示名称  | 唯一标识  | 字段类型 |
|-------|-------|------|
| webId | webId | 文本   |

6. 在“数据”页签，单击“添加”，为对象添加图6-5中数据。

图 6-5 为对象添加数据

字段索引验证清理数据

+ 添加 删除 导入 导出 设置出厂数据

输入名称

| <input type="checkbox"/> | 序列 | 名称 | webName | webLink                  | webId |
|--------------------------|----|----|---------|--------------------------|-------|
| <input type="checkbox"/> | 1  |    | WEB C   | https://.../blogs/354... | 003   |
| <input type="checkbox"/> | 2  |    | WEB B   | https://.../wtsne...     | 002   |
| <input type="checkbox"/> | 3  |    | WEB A   | https://.../product/...  | 001   |

步骤3 新建对象模型。

1. 在应用设计器中，选择“界面”，单击页面后的“+”。
2. 输入页面的标签和名称，单击“添加”，新建一个标准页面。
3. 在标准页面底部，单击“模型视图”，将页面从设计视图切换到模型视图。
4. 单击“新增模型”，输入模型名称（link）、“来源”选择“对象”，单击“下一步”。

图 6-6 新建模型

新增模型

基本设置

设置

方法

模型名称

link

来源

自定义

对象

属性

事件

5. 选择步骤3中创建的对象和添加的字段，单击“下一步”，再单击“确定”，完成模型的创建。

图 6-7 选择对象和字段

新增模型

基本设置

设置

方法

选择对象

选择字段

选择对象

选择字段

取消

上一步

下一步



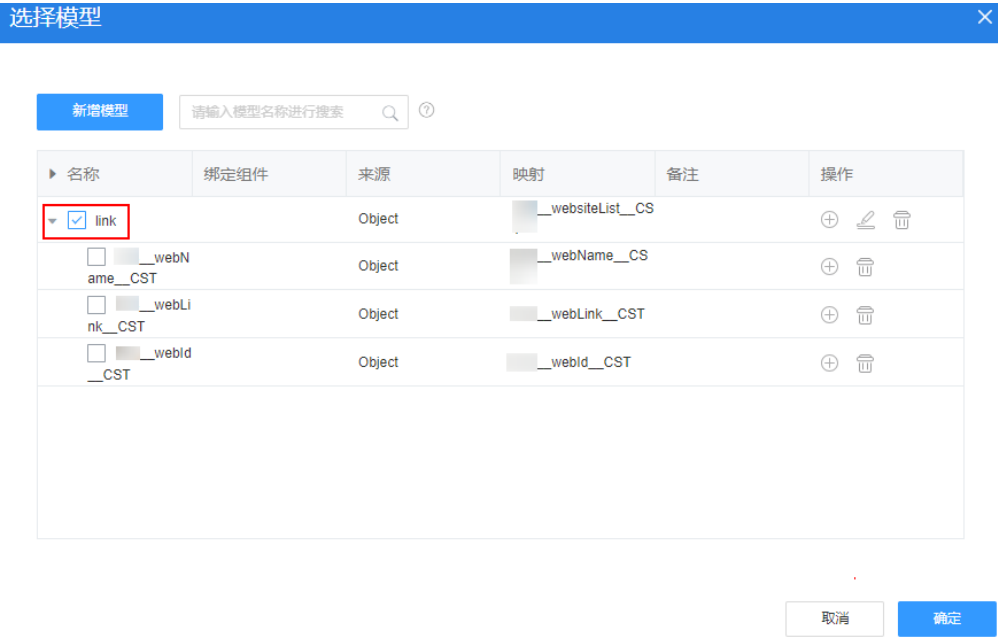
- 步骤4** 返回设计视图页面，新建表格关联模型。
1. 在标准页面中拖入一个表格组件。

图 6-8 拖入表格组件



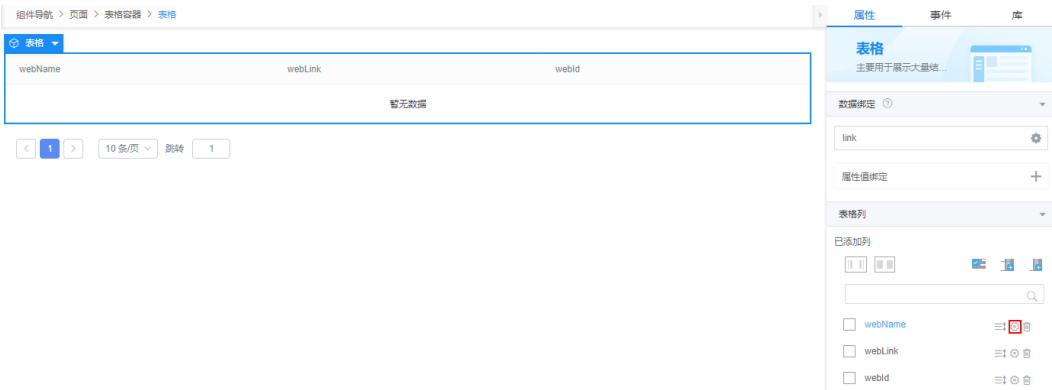
2. 选中表格组件，在“属性 > 数据绑定 > 值绑定”中，单击 。
3. 选中**步骤3**中创建的模型，单击“确定”。

图 6-9 选择模型



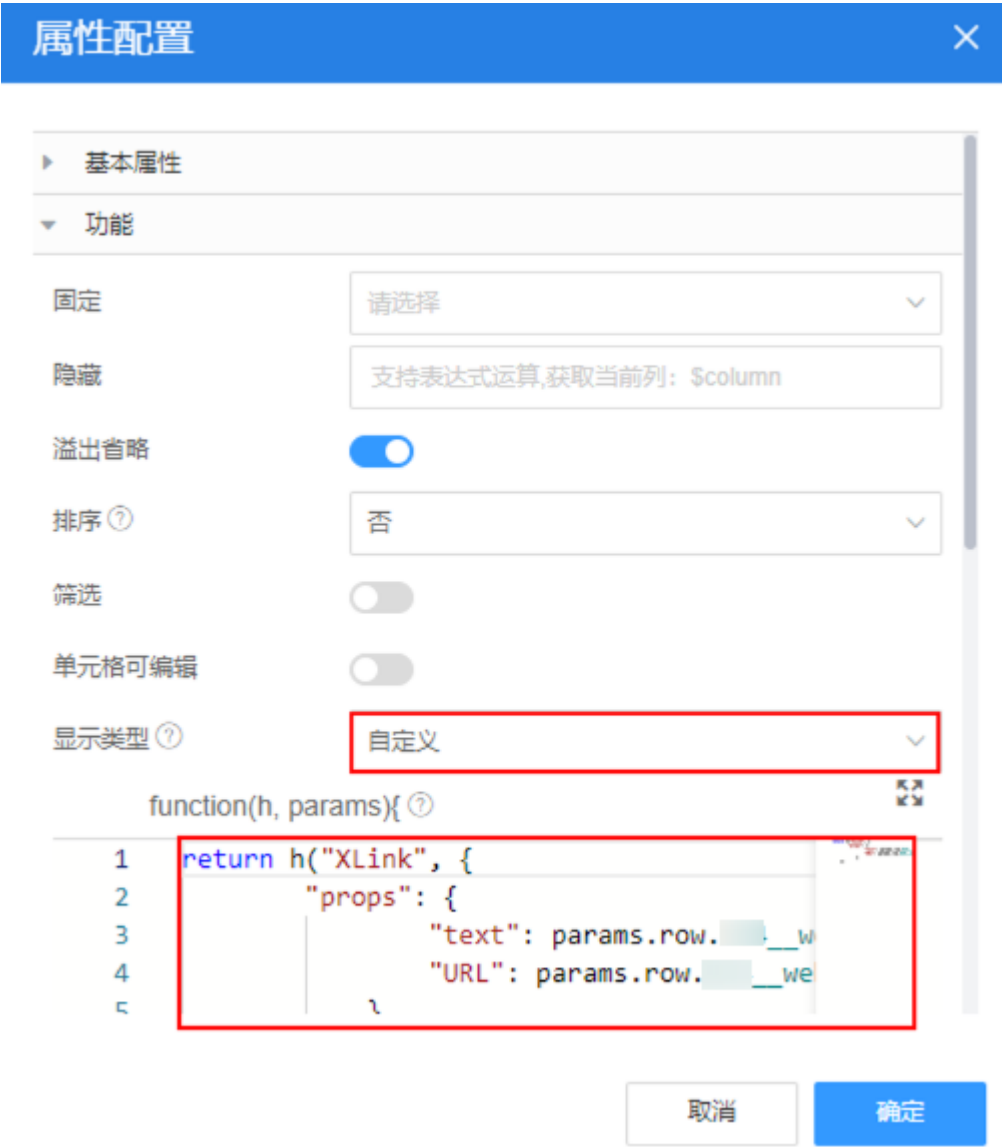
- 步骤5** 在“表格列”中，单击webName列后的 。

图 6-10 选择所需的列



步骤6 在“属性配置 > 功能”中，为webName列添加URL链接信息。

图 6-11 设置列属性



将“显示类型”设置为“自定义”，并输入如下内容：

```
return h("XLink", {
 "props": {
 "text": params.row.命名空间_webName_CST,
 "URL": params.row.命名空间_webLink_CST
 }
})
```

- 步骤7 单击页面上方的，保存页面。
- 步骤8 保存成功后，单击页面上方的，预览效果。
- 结束

## 6.2 为华为云 Astro 轻应用标准页面中的表格增加求和等计算能力

### 期望实现效果

在标准页面中，支持为表格增加求和、求积等计算能力，来提升数据的处理效率。例如，将表格中“商品花费”列的值设置为“商品数\*价格+其他成本”。

图 6-12 实现效果

| <input type="checkbox"/> | 商品名称     | 商品数目 | 商品价格 | 其他成本 | 商品花费 |
|--------------------------|----------|------|------|------|------|
| <input type="checkbox"/> | proOne   | 3    | 200  | 400  | 1000 |
| <input type="checkbox"/> | proTwo   | 2    | 150  | 200  | 500  |
| <input type="checkbox"/> | proThree | 4    | 100  | 300  | 700  |

### 功能实现方法

- 步骤1 创建一个低代码应用。
- 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
  - 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
  - 在“应用”中，单击“新建低代码应用”或单击，进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。
  - 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。
  - 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 6-13 创建一个空白应用

新建 空白应用

基本信息

添加图标

\* 标签

我的第一个应用

\* 名称

A

分类

请选择

描述

请输入

高级设置

取消

新建

表 6-5 新建空白应用参数说明

| 参数 | 说明                                                                                                                                                                                                                                                                                   | 示例      |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 标签 | 新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。                                                                                                                                                                                                                                         | 我的第一个应用 |
| 名称 | 新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下： <ul style="list-style-type: none"><li>长度不能超过31个字符，包括前缀命名空间的长度。</li><li>名称前的内容为命名空间，在华为云Astro轻应用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。</li><li>必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。</li></ul> | A       |

- 步骤2 创建对象“ProductList”，并为对象添加字段和数据。
1. 在应用设计器的左侧导航栏中，选择“数据”，单击对象中的“+”。

2. 设置对象的名称和唯一标识为“ProductList”，单击“确定”。

图 6-14 创建对象 ProductList

创建新对象

您可以创建用来保存持久化数据的实体对象，对象里的数据存储在平台里面

从空白创建

从模板创建

您将创建单个对象，从零开始设计对象的字段，并添加测试数据

\* 对象名称

ProductList

\* 唯一标识

ProductList

取消

确定

表 6-6 新建 ProductList 对象参数说明

| 参数   | 说明                                                                                                                                                                                                                                                | 示例          |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 对象名称 | 新建对象的名称，创建后可修改。<br>取值范围：1~80个字符。                                                                                                                                                                                                                  | ProductList |
| 唯一标识 | 新建对象在系统中的标识，创建后不支持修改。命名要求如下： <ul style="list-style-type: none"><li>长度不能超过63个字符，包括前缀命名空间的长度。<br/>标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。</li><li>必须以英文字母开头，只能由英文字母，数字和下划线组成，且不能以下划线结尾。</li></ul> | ProductList |

3. 在已创建的对象中，单击，进入对象详情页面。
4. 在“字段”页签，单击“添加”，为对象添加productName字段。

图 6-15 添加 productName 字段

添加字段

\*

显示名称

商品名称

\*

唯一标识

productName

\*

字段类型

文本

...

\*

数据长度

64

描述

取消

确认

表 6-7 添加 cost 字段参数说明

| 参数   | 说明                                                                                                                                                   | 示例          |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 显示名称 | 新建字段的名称，创建后可修改。<br>取值范围：1~63个字符。                                                                                                                     | 商品名称        |
| 唯一标识 | 新建字段在系统中的标识，创建后不支持修改。命名要求如下： <ul style="list-style-type: none"><li>长度不能超过63个字符，包括前缀命名空间的长度。</li><li>必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。</li></ul> | productName |
| 字段类型 | 单击“...”，在弹出的页面中，根据页面提供的参数解释，选择新建字段所属的类型。                                                                                                             | 文本          |

5. 按照上述操作，为对象添加表6-8中字段。

表 6-8 添加其他字段

| 显示名称 | 唯一标识       | 字段类型 |
|------|------------|------|
| 商品数目 | productNum | 数字   |

| 显示名称 | 唯一标识         | 字段类型 |
|------|--------------|------|
| 商品价格 | productPrice | 数字   |
| 其他成本 | cost         | 数字   |

图 6-16 查看添加的字段

字段索引验证清理数据

+ 添加

批量创建

输入名称

| 序号 | 显示名称  | 唯一标识               | 类型 | 描述    | 是否系统字段 | 是否必填 | 是否唯一 |
|----|-------|--------------------|----|-------|--------|------|------|
| 1  | 其他成本  | __cost_CST         | 数字 | 无描述信息 | —      | —    | —    |
| 2  | 商品价格  | __productPrice_CST | 数字 | 无描述信息 | —      | —    | —    |
| 3  | 商品数目  | __productNum_CST   | 数字 | 无描述信息 | —      | —    | —    |
| 4  | 商品名称  | __productName_CST  | 文本 | 无描述信息 | —      | —    | —    |
| 5  | 记录 ID | id                 | ID | 无描述信息 | ✓      | ✓    | ✓    |

6. 选择“数据”页签，单击“添加”，为对象添加图6-17中数据。

图 6-17 为对象添加数据

字段索引验证清理数据

+ 添加

删除

导入

导出

设置出厂数据

输入名称

| <input type="checkbox"/> | 序列 | 名称       | 商品名称     | 商品数目 | 商品价格 | 其他成本 |
|--------------------------|----|----------|----------|------|------|------|
| <input type="checkbox"/> | 1  | proOne   | proOne   | 3    | 200  | 400  |
| <input type="checkbox"/> | 2  | proTwo   | proTwo   | 2    | 150  | 200  |
| <input type="checkbox"/> | 3  | proThree | proThree | 4    | 100  | 300  |

步骤3 新建对象模型。

1. 在应用设计器中，选择“界面”，单击页面后的“+”。

2. 输入页面的标签和名称，单击“添加”，新建一个标准页面。

3. 在标准页面底部，单击“模型视图”，将页面从设计视图切换到模型视图。

4. 单击“新增模型”，输入模型名称（如productCost）、“来源”选择“对象”，单击“下一步”。

图 6-18 新建模型

新增模型

1 基本信息

2 设置

模型名称

productCost

来源

自定义

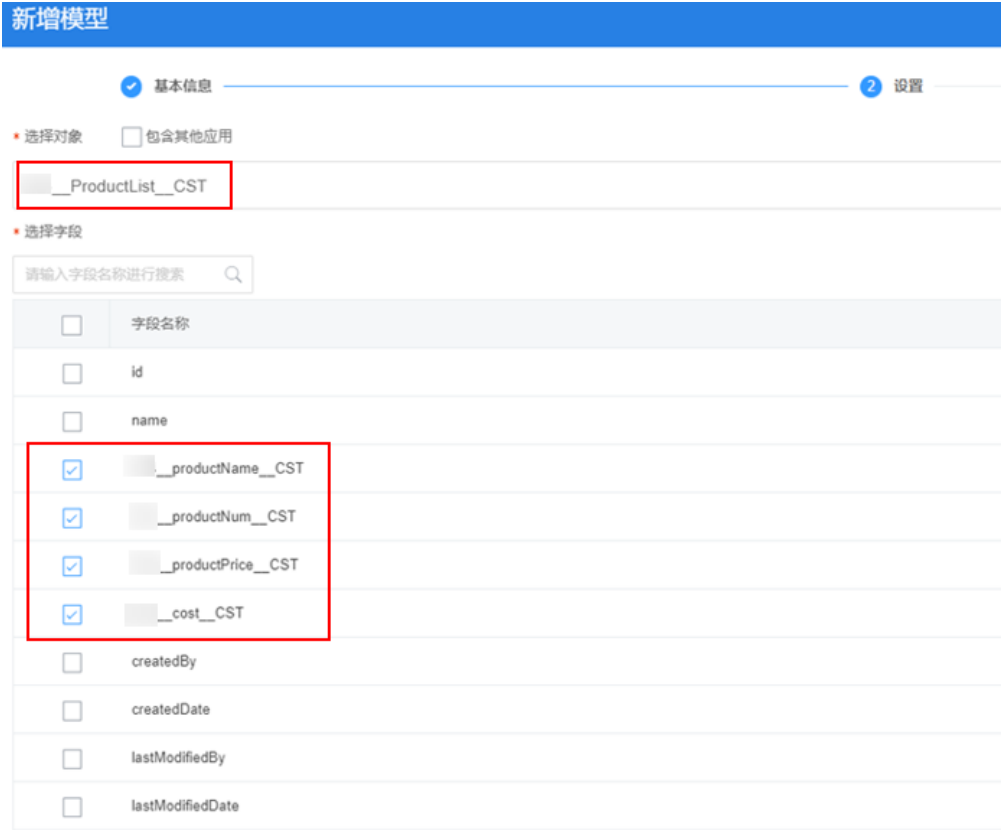
由后台前端模型，可以在模型树上快速创建自定义字段

对象

由后台对象模型映射创建，支持选择字段

5. 选择步骤3中创建的对象和添加的字段，单击“下一步”，再单击“确定”，完成模型的创建。

图 6-19 选择对象和字段



步骤4 返回设计视图页面，新建表格关联模型。

1. 在标准页面中拖入一个表格组件。

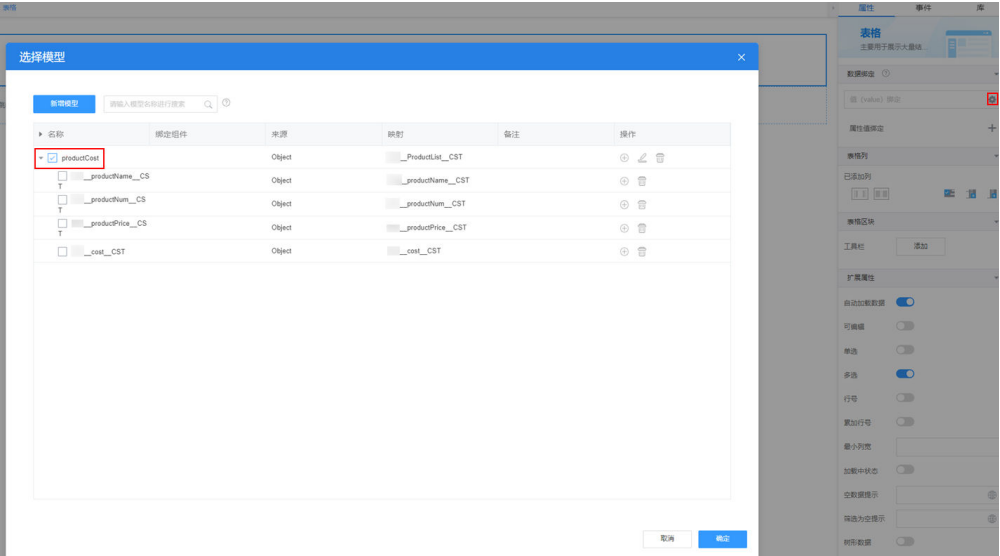
图 6-20 拖入表格组件



2. 选中表格组件，在“属性 > 数据绑定 > 值绑定”中，单击 .
3. 选中步骤3中创建的模型，单击“确定”。



图 6-21 选择模型



步骤5 选中表格，在“属性 > 表格列 > 已添加列”中，单击，添加一个空白列。

图 6-22 添加一个空白列



步骤6 单击新增空白列后的，修改“列名称”为“商品花费”，将“显示类型”设置为“自定义”并输入自定义代码。

图 6-23 修改列标题为商品花费

属性配置

▼ 基本属性

字段名

1721094248354

列标题

商品花费

列标题提示

列标题自定义渲染

▼ 功能

固定

请选择

隐藏

支持表达式运算,获取当前列: \$column

图 6-24 显示类型选择自定义并输入自定义代码

属性配置

图标位置

请选择

最大字符长度

^

v

最大宽度

^

v

px

v

必填

必填错误信息

自定义校验

显示类型 ?

自定义

function(h, params){ ?

1

2

3

4

5

6

return h("XInputNumber", {

"props": {

"value": params.row.命名空间\_\_prod

"readonly": true

}

})

取消

确定

自定义代码如下：

```
return h("XInputNumber", {
 "props": {
 "value": params.row.命名空间__productPrice__CST * params.row.命名空间__productNum__CST +
params.row.命名空间__cost__CST,
 "readonly": true
 }
})
```

**步骤7** 单击页面上方的，保存页面。

**步骤8** 保存成功后，单击页面上方的，预览效果。

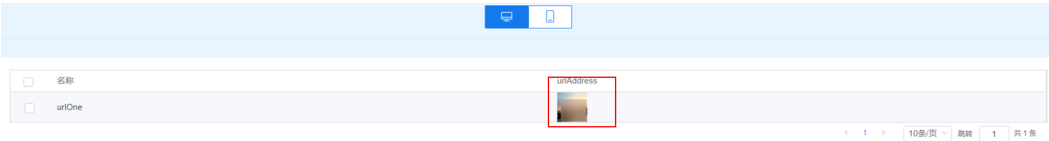
----结束

## 6.3 在华为云 Astro 轻应用标准页面的表格中显示图片

### 期望实现效果

在标准页面中，通过自定义列的显示类型，可以将图片显示在表格中。在表格中显示图片可增强信息的表达效果，信息更直观、生动和易于理解。

图 6-25 实现效果



### 功能实现方法

- 步骤1 创建一个低代码应用。
1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。

2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。

3. 在“应用”中，单击“新建低代码应用”或单击, 进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。

4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。

5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 6-26 创建一个空白应用



表 6-9 新建空白应用参数说明

| 参数 | 说明                                                                                                                                                                                                                                                                                   | 示例      |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 标签 | 新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。                                                                                                                                                                                                                                         | 我的第一个应用 |
| 名称 | 新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下： <ul style="list-style-type: none"><li>长度不能超过31个字符，包括前缀命名空间的长度。</li><li>名称前的内容为命名空间，在华为云Astro轻应用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。</li><li>必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。</li></ul> | A       |

步骤2 创建对象“urlList”，并为对象添加字段。

1.

在应用设计器的左侧导航栏中，选择“数据”，单击对象中的“+”。
2.

设置对象的名称和唯一标识为“urlList”，单击“确定”。

图 6-27 创建对象 urlList



表 6-10 新建 urlList 对象参数说明

| 参数   | 说明                               | 示例      |
|------|----------------------------------|---------|
| 对象名称 | 新建对象的名称，创建后可修改。<br>取值范围：1~80个字符。 | urlList |

| 参数   | 说明                                                                                                                                                                                                                                           | 示例      |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 唯一标识 | 新建对象在系统中的标识，创建后不支持修改。命名要求如下： <ul style="list-style-type: none"><li>长度不能超过63个字符，包括前缀命名空间的长度。标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。</li><li>必须以英文字母开头，只能由英文字母，数字和下划线组成，且不能以下划线结尾。</li></ul> | urlList |

3. 在已创建的对象中，单击，进入对象详情页面。
4. 在“字段”页签，单击“添加”，为对象添加urlAddress字段。

图 6-28 添加 urlAddress 字段

添加字段

\* 显示名称

urlAddress

\* 唯一标识

urlAddress

\* 字段类型

文本区

描述

取消

确认

表 6-11 添加 urlAddress 字段参数说明

| 参数   | 说明                               | 示例         |
|------|----------------------------------|------------|
| 显示名称 | 新建字段的名称，创建后可修改。<br>取值范围：1~63个字符。 | urlAddress |

| 参数   | 说明                                                                                                                                                   | 示例         |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 唯一标识 | 新建字段在系统中的标识，创建后不支持修改。命名要求如下： <ul style="list-style-type: none"><li>长度不能超过63个字符，包括前缀命名空间的长度。</li><li>必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。</li></ul> | urlAddress |
| 字段类型 | 单击 <b>***</b> ，在弹出的页面中，根据页面提供的参数解释，选择新建字段所属的类型。                                                                                                      | 文本区        |

5. 在“数据”页签，单击“添加”，为对象添加数据。  
其中，“urlAddress”的值请配置为图片的存放地址。

图 6-29 为对象添加数据



步骤3 新建对象模型。

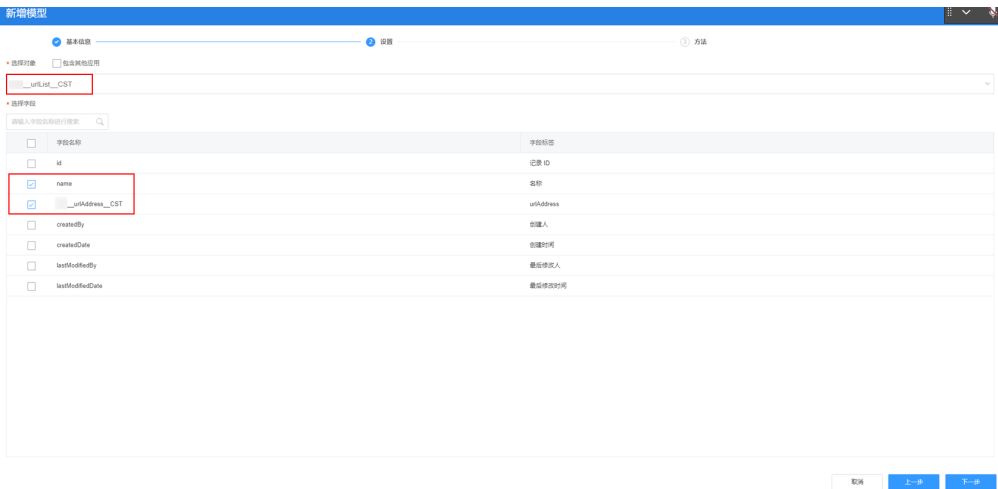
1. 在应用设计器中，选择“界面”，单击页面后的“+”，新建一个标准页面。
2. 在标准页面底部，单击“模型视图”，从设计视图切换到模型视图。
3. 单击“新增模型”，输入模型名称（如urlMod）、“来源”选择“对象”，单击“下一步”。

图 6-30 新建模型



4. 选择步骤3中创建的对象和添加的字段，单击“下一步”。

图 6-31 选择对象和字段



5. 直接单击“确定”，完成模型创建。

步骤4 返回设计视图页面，新建表格关联模型。

- 1. 在标准页面底部，单击“设计视图”，从模型视图切换到设计视图。
- 2. 在标准页面中拖入一个表格组件。

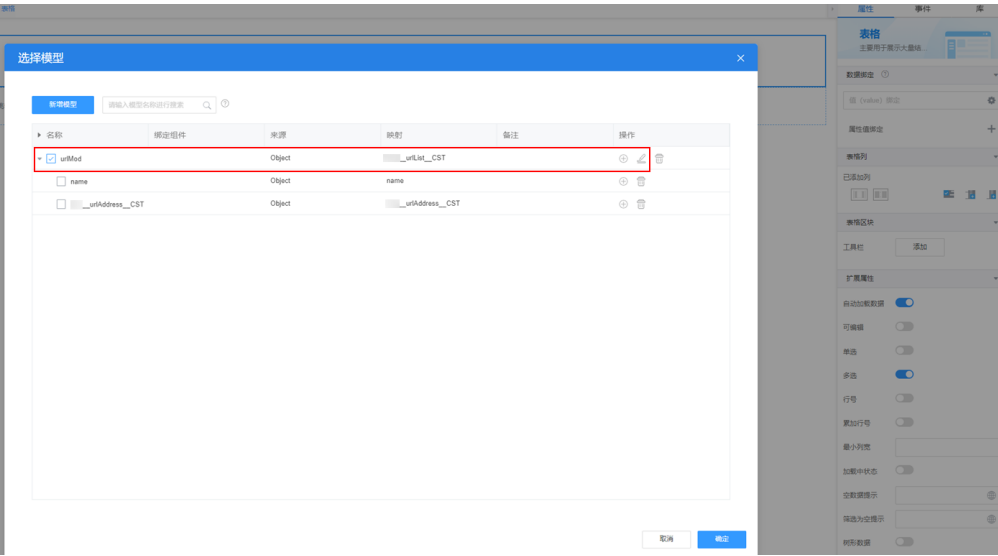
图 6-32 拖入表格组件



- 3. 选中表格组件，在“属性 > 数据绑定 > 值绑定”中，单击 .
- 4. 选中步骤3中创建的模型，单击“确定”。

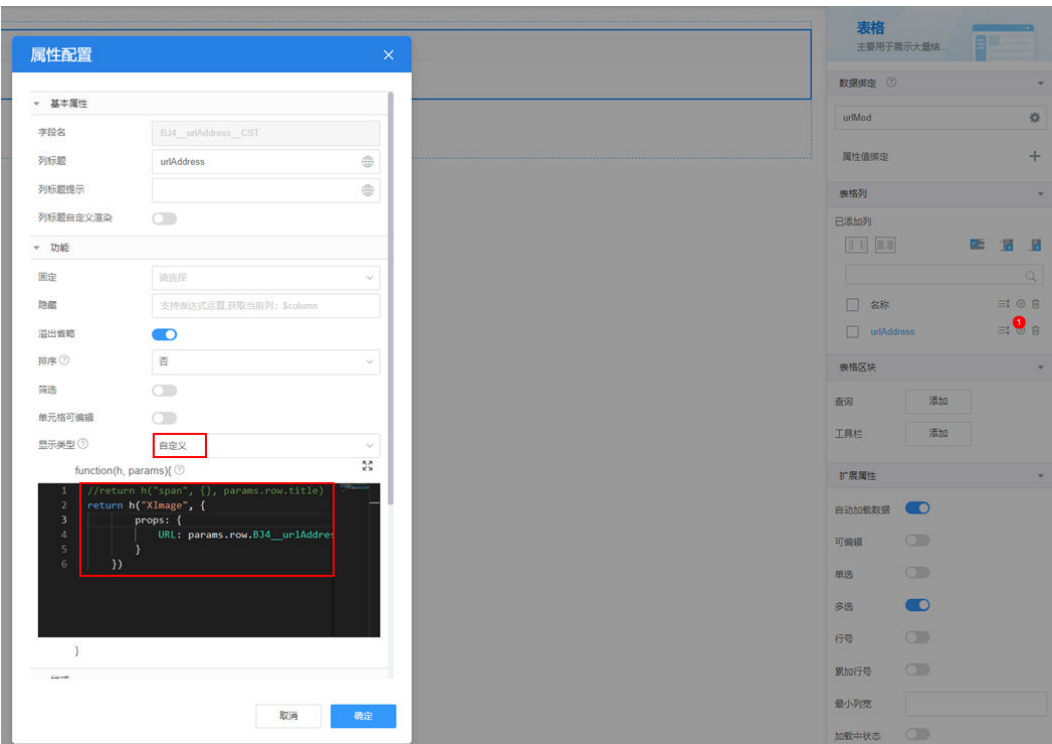


图 6-33 选择模型



- 步骤5** 选中表格，在“属性 > 表格列 > 已添加列”中，单击待显示图片列（urlAddress）后的.
- 步骤6** 设置属性，单击“确定”，返回标准页面。

图 6-34 设置属性



将“显示类型”设置为“自定义”，并输入如下内容。

```
return h("XImage",{
 "props": {
 "URL": params.row.命名空间_urlAddress_CST, width: 50, height: 50
 }
})
```

```
}
})
```

- 步骤7 单击页面上方的, 保存页面。
- 步骤8 保存成功后, 单击页面上方的, 查看页面配置效果。
- 结束

## 6.4 如何在审批页面直接复用提交的表单页面

### 期望实现效果

在标准页面中, 通过表格创建数据提交页面和通过表格创建数据审批页面是两个常用功能, 如员工出差申请、请假申请、报销单据审批及商品出入库审批等。当数据提交页面与数据审批页面内容差异不大时, 可以考虑将两个功能复用至同一页面, 避免重复开发, 提高开发效率。

图 6-35 实现效果



### 功能实现方法

- 步骤1 创建一个低代码应用。
1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作, 申请华为云Astro轻应用免费试用或购买商业实例。

2. 实例购买后, 在华为云Astro轻应用服务控制台的“主页”中, 单击“进入首页”, 进入应用开发页面。

3. 在“应用”中, 单击“新建低代码应用”或单击, 进入新建低代码应用页面。首次创建应用时, 请根据界面提示创建一个命名空间。命名空间一旦创建, 不能修改和删除, 创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。

4. 在新建低代码应用页面, 应用类型选择“标准应用”, 单击“确定”。

5. 输入应用的标签和名称, 单击“新建”, 即可进入应用设计器。

图 6-36 创建一个空白应用

新建 空白应用

基本信息



添加图标

\* 标签

我的第一个应用

\* 名称

A

分类

请选择

描述

请输入

高级设置

取消

新建

表 6-12 新建空白应用参数说明

| 参数 | 说明                                                                                                                                                                                                                                                                                   | 示例      |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 标签 | 新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。                                                                                                                                                                                                                                         | 我的第一个应用 |
| 名称 | 新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下： <ul style="list-style-type: none"><li>长度不能超过31个字符，包括前缀命名空间的长度。</li><li>名称前的内容为命名空间，在华为云Astro轻应用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。</li><li>必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。</li></ul> | A       |

- 步骤2 创建一个对象“productList”，并为对象添加字段。
1. 在应用设计器的左侧导航栏中，选择“数据”，单击对象中的“+”。

2. 设置对象的名称和唯一标识为“productList”，单击“确定”，进入对象详情页。

图 6-37 创建对象 productList

创建新对象

您可以创建用来保存持久化数据的实体对象。对象里的数据存储在平台里面

从空白创建

从模板创建

您将创建单个对象，从零开始设计对象的字段，并添加测试数据

\* 对象名称

productList

\* 唯一标识

productList

取消

确定

表 6-13 新建 productList 对象参数说明

| 参数   | 说明                                                                                                                                                                                                                                           | 示例          |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 对象名称 | 新建对象的名称，创建后可修改。<br>取值范围：1~80个字符。                                                                                                                                                                                                             | productList |
| 唯一标识 | 新建对象在系统中的标识，创建后不支持修改。命名要求如下： <ul style="list-style-type: none"><li>长度不能超过63个字符，包括前缀命名空间的长度。标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。</li><li>必须以英文字母开头，只能由英文字母，数字和下划线组成，且不能以下划线结尾。</li></ul> | productList |

3. 在已创建的对象中，单击，进入对象详情页面。
4. 在“字段”页签，单击“添加”，为对象添加productName字段。

图 6-38 添加 productName 字段

添加字段

\* 显示名称

商品名

\* 唯一标识

—

productName

\* 字段类型

文本

\* 数据长度

64

描述

取消

确认

表 6-14 添加 productName 字段参数说明

| 参数   | 说明                                                                                                                                                   | 示例          |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 显示名称 | 新建字段的名称，创建后可修改。<br>取值范围：1~63个字符。                                                                                                                     | 商品名         |
| 唯一标识 | 新建字段在系统中的标识，创建后不支持修改。命名要求如下： <ul style="list-style-type: none"><li>长度不能超过63个字符，包括前缀命名空间的长度。</li><li>必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。</li></ul> | productName |
| 字段类型 | 单击“***”，在弹出的页面中，根据页面提供的参数解释，选择新建字段所属的类型。                                                                                                             | 文本          |

5. 按照上述操作，为对象添加表6-15中字段。

表 6-15 为对象添加字段

| 显示名称 | 唯一标识        | 字段类型 | 数据长度 |
|------|-------------|------|------|
| 商品描述 | productDesc | 文本   | 255  |

| 显示名称 | 唯一标识             | 字段类型 | 数据长度 |
|------|------------------|------|------|
| 审批意见 | approvalComments | 文本   | 255  |
| 商品状态 | productStatus    | 文本   | 255  |

执行上述操作后，在对象的“字段”页签，可查看到已添加的自定义字段。

图 6-39 查看已添加的字段

名称  
productList  
修改时间  
2025-05-06 10:49:35

唯一标识  
\_productList\_CST

描述  
暂无描述

创建人

字段

索引

验证

清理

数据

+ 添加

批量创建

Q 输入名称

👁

| 序号 | 显示名称  | 唯一标识               | 类型 | 描述    | 是否系统字段 | 是否必填 | 是否唯一 |
|----|-------|--------------------|----|-------|--------|------|------|
| 1  | 商品状态  | _productStatus_... | 文本 | 无描述信息 | —      | —    | —    |
| 2  | 审批意见  | _approvalComme...  | 文本 | 无描述信息 | —      | —    | —    |
| 3  | 商品描述  | _productDesc_CST   | 文本 | 无描述信息 | —      | —    | —    |
| 4  | 商品名   | _productName_...   | 文本 | 无描述信息 | —      | —    | —    |
| 5  | 记录 ID | id                 | ID | 无描述信息 | ✓      | ✓    | ✓    |

- 步骤3 创建一个列表页，并新建对象模型。
1. 在应用设计器中，选择“界面”，单击页面后的“+”，新建一个标准页面。

图 6-40 新建列表页

添加标准页面

×

+  
空白

基于模板

\* 标签

\* 名称

列表页

List

取消

添加

2. 在标准页面底部，单击“模型视图”，从设计视图切换到模型视图。

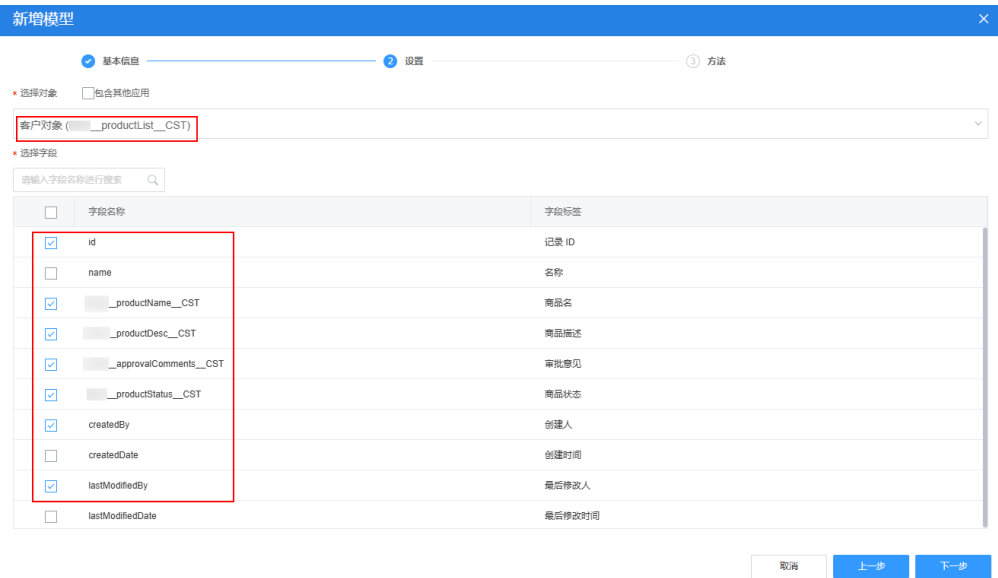
3. 单击“新增模型”，输入模型名称（如productListModel）、“来源”选择“对象”，单击“下一步”。

图 6-41 新建模型



4. 选择步骤2中创建的对象和添加的字段，单击“下一步”。

图 6-42 选择对象和字段



5. 直接单击“确定”，完成模型创建。
6. 返回设计视图页面，在标准页面底部，单击“设计视图”，从模型视图切换到设计视图。
7. 在标准页面中拖入一个表格组件。

图 6-43 拖入表格组件



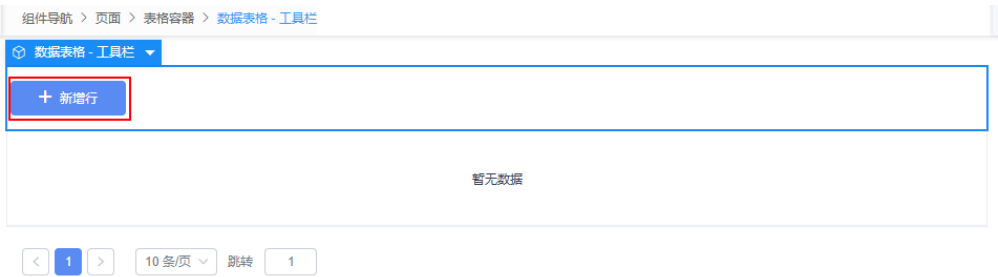
8. 选中表格组件，在“属性 > 表格区块 > 工具栏”中，单击“添加”，为表格添加一个工具栏。

图 6-44 添加工具栏



9. 删除工具栏中多余的按钮，仅保留一个“新增行”。

图 6-45 删除多余按钮



10. 选中表格组件，在“属性 > 数据绑定 > 值绑定”中，单击 。
11. 选中步骤3.3中创建的模型，单击“确定”，实现表格列表功能。



图 6-46 选择对象和字段

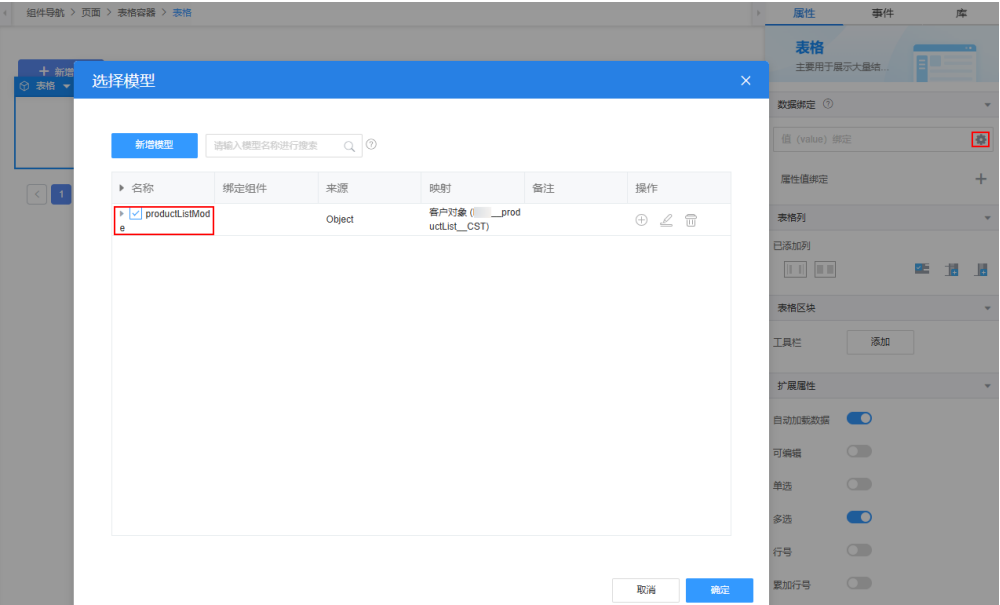


图 6-47 表格列表

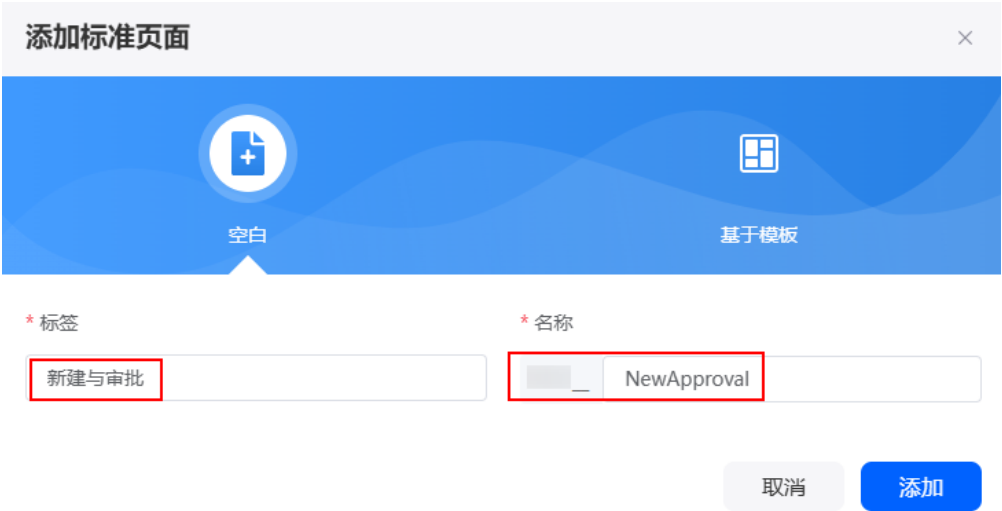


12. 单击页面上方的，保存页面。

步骤4 创建新建和审批页面，并添加模型。

1. 在应用设计器中，选择“界面”，单击页面后的“+”，新建一个标准页面。

图 6-48 创建新建和审批页面



2. 在标准页面中拖入一个表单组件，绑定数据对象并添加操作按钮。

图 6-49 拖入表单组件

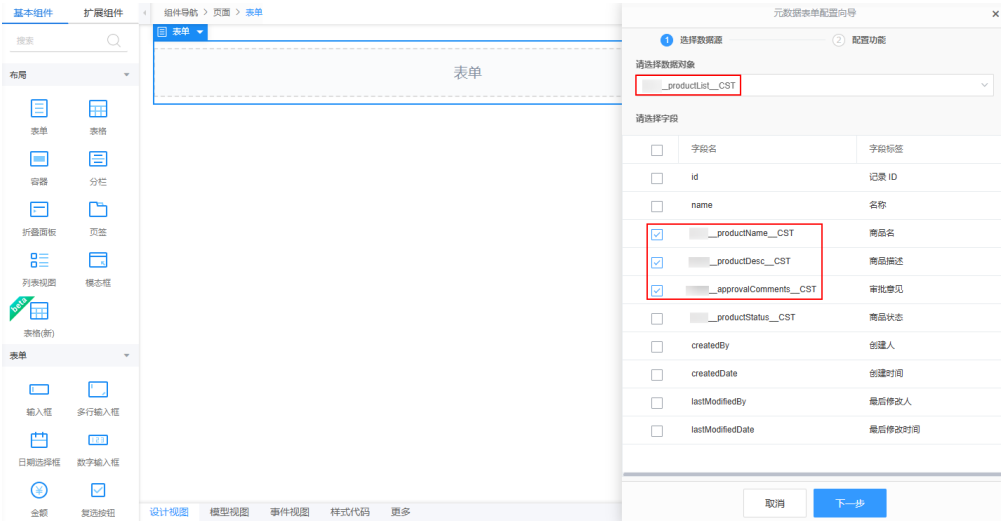
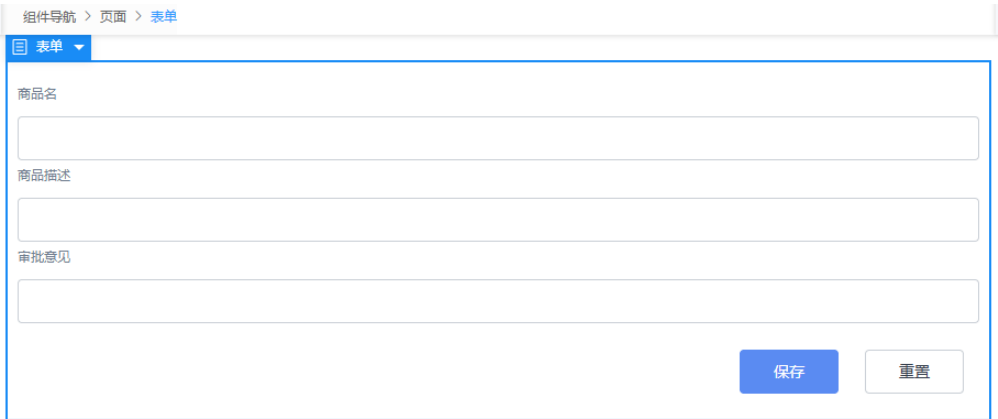


图 6-50 添加操作按钮

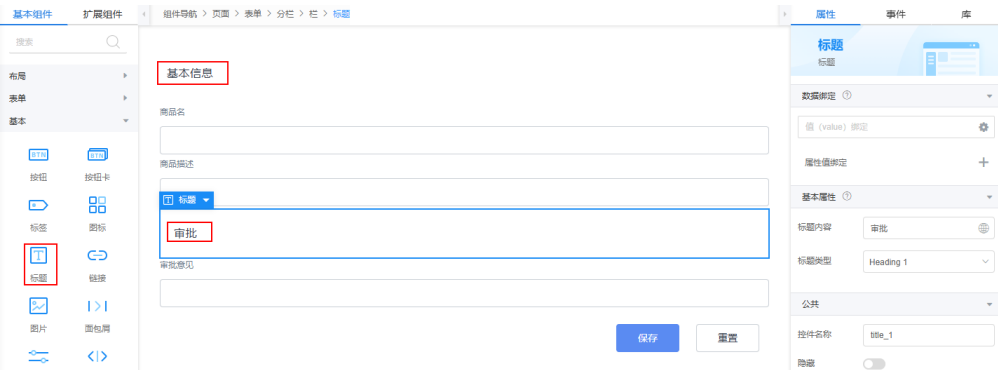


图 6-51 添加表单后效果



3. 拖拽两个“标题”组件到页面中，修改“标题内容”为“基本信息”和“审批”，布局效果如图6-52。

图 6-52 添加标题组件调整布局



4. 在标准页面底部，单击“模型视图”，从设计视图切换到模型视图。

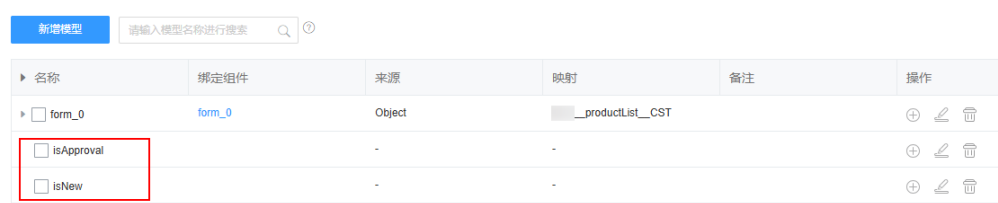
5. 单击“新增模型”，输入模型名称（如isApproval）、“来源”选择“自定义”，单击“下一步”。

图 6-53 新增 isApproval 模型



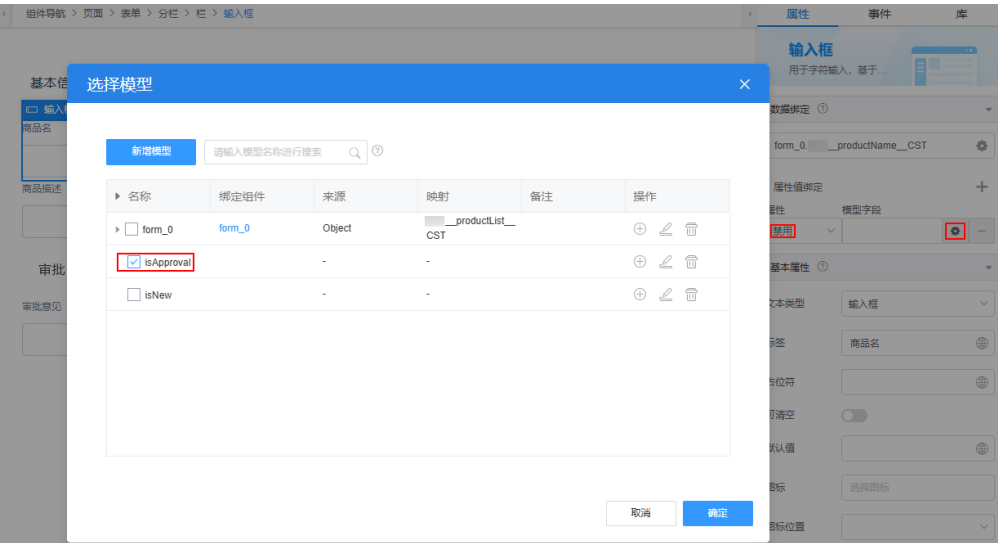
6. 保持默认，直接单击“下一步”，再单击“确定”，完成模型的创建。
7. 按照步骤4.5~步骤4.6中操作，再创建一个“isNew”模型，最终效果下图所示。

图 6-54 查看已创建的模型



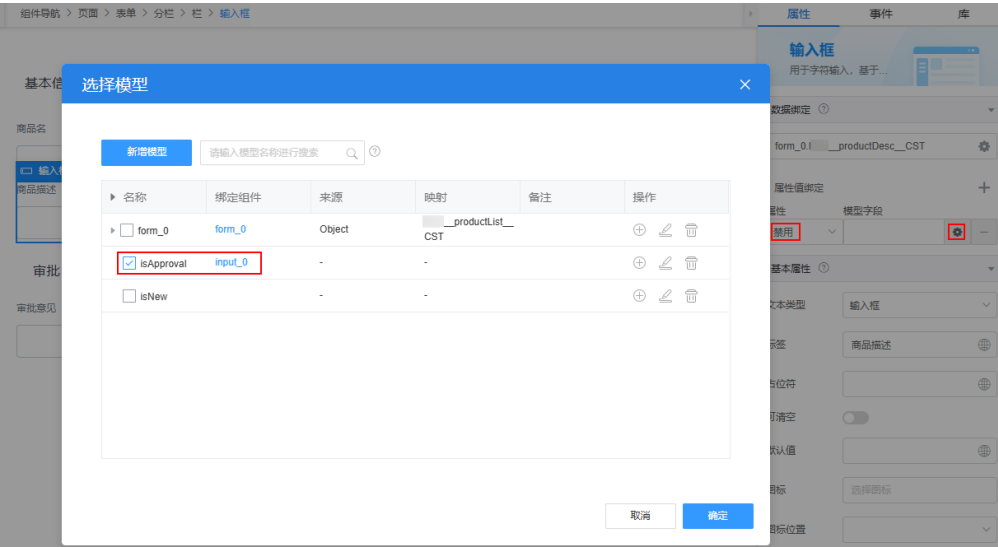
8. 在标准页面底部，单击“设计视图”返回设计视图页面，选中商品名输入框组件，在“属性 > 数据绑定 > 属性值绑定”中，单击“+”，选择“禁用”，再单击，为字段绑定isApproval模型。

图 6-55 为商品名输入框组件绑定 isApproval 模型



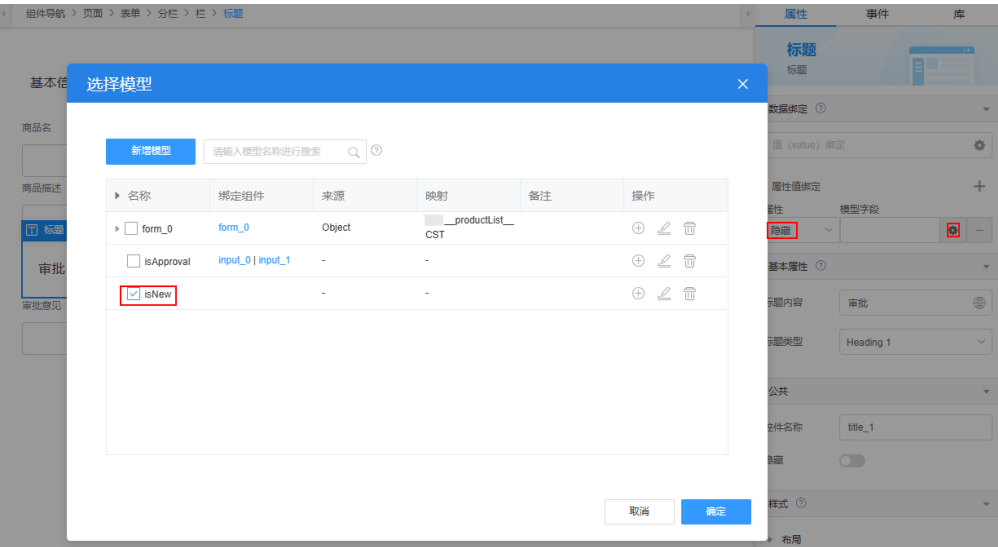
9. 选中**商品描述**输入框组件，在“属性 > 数据绑定 > 属性值绑定”中，单击“+”，选择“禁用”，再单击，为组件绑定isApproval模型。

图 6-56 为商品描述输入框绑定 isApproval 模型



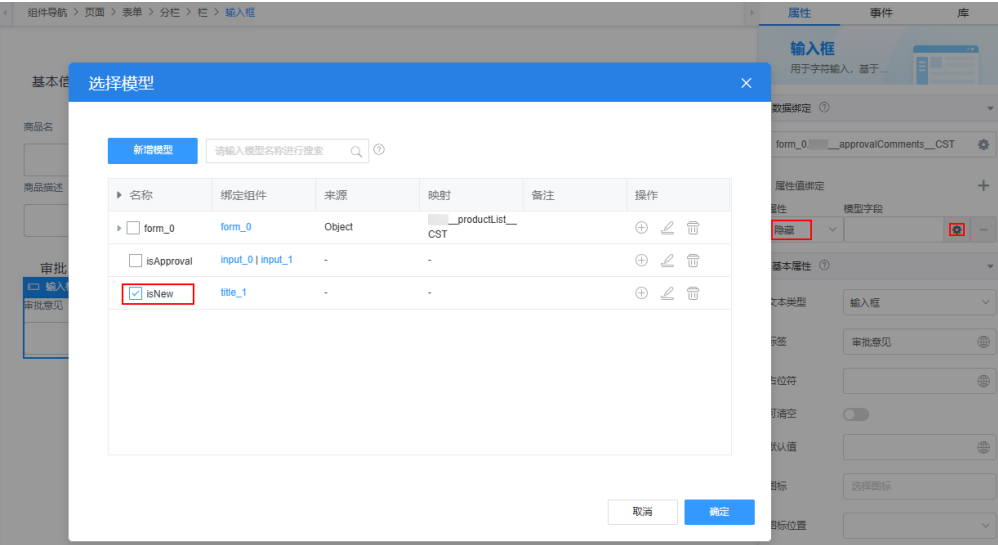
10. 选中**审批标题**组件，在“属性 > 数据绑定 > 属性值绑定”中，单击“+”，选择“隐藏”，再单击，为字段绑定isNew模型。

图 6-57 为审批标题绑定 isNew 模型



11. 同样，按照**步骤4.10**中操作，选中**审批意见**输入框组件，为组件绑定isNew模型。

图 6-58 为审批意见输入框绑定 isNew 模型



12. 选中页面组件，在“事件”页签，单击“加载”后的+，为页面添加事件。

图 6-59 为页面添加事件

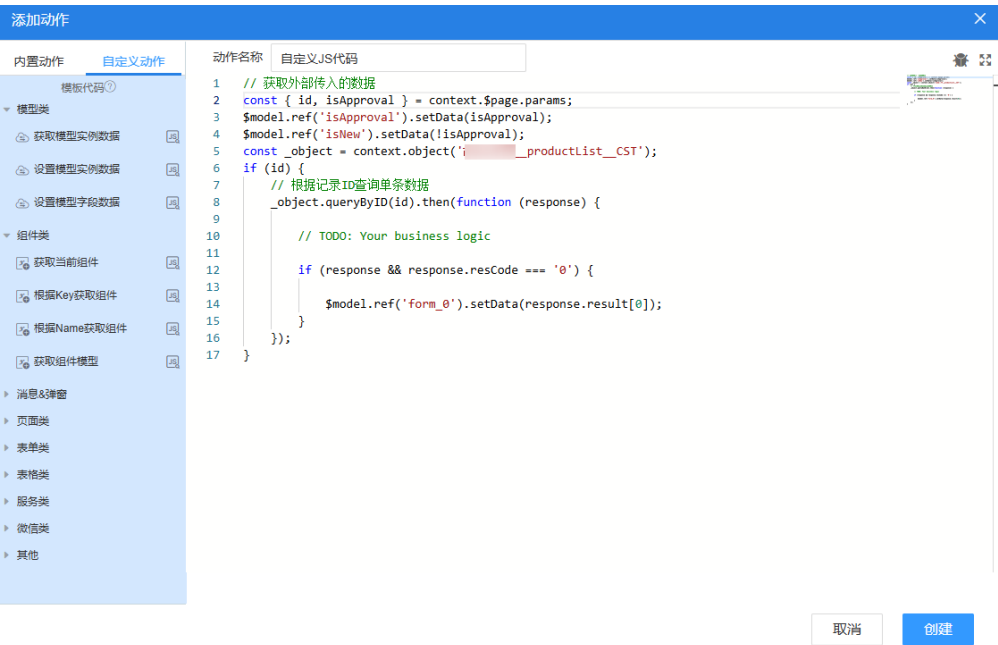


图 6-60 查看表单绑定模型的名称



在自定义动作中，输入如下示例，用于获取外部数据。其中，*isApproval*为步骤4.5中创建的模型名，*命名空间\_productList\_CST*为步骤2中创建的对象，*form\_0*为步骤4.2中表单绑定模型的名称，如图6-60。

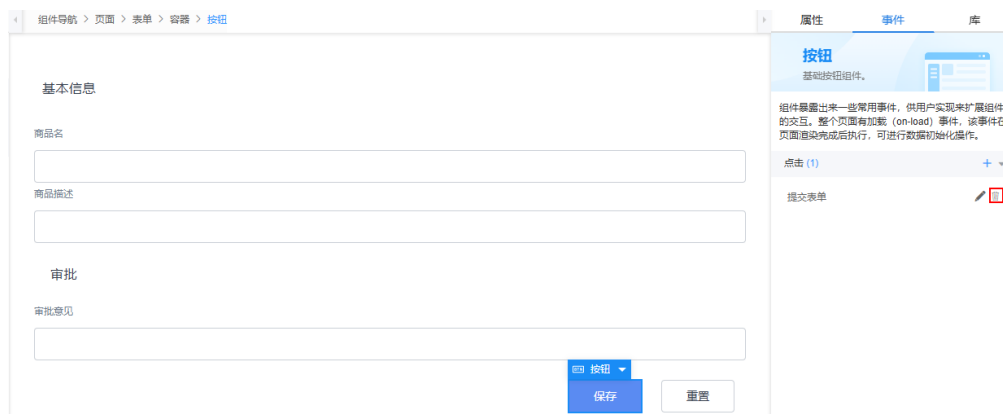
```
// 获取调用打开页面方法时传入的参数
const { id, isApproval } = context.$page.params;
$model.ref('isApproval').setData(isApproval);
$model.ref('isNew').setData(!isApproval);
const _object = context.object('命名空间_productList_CST');
if (id) {
 // 根据记录ID查询单条数据
 _object.queryById(id).then(function (response) {

 if (response && response.resCode === '0') {

 $model.ref('form_0').setData(response.result[0]);
 }
 });
}
```

- 选中“保存”按钮，在“事件”页签中，单击“提交表单”后的删除当前提交表单的方法，再单击“点击”后的“+”，为按钮添加自定义动作。

图 6-61 删除提交表单方法



在自定义动作中，输入如下示例，用于处理新建或更新的数据。其中，*isNew*为步骤4.7中创建的模型名称，*命名空间\_productList\_CST*为步骤2.2中创建对象的名称、*命名空间\_productStatus\_CST*和*命名空间\_approvalComments\_CST*为步骤2.5中添加的对象字段。

```
// 获取模型数据
const isNew = $model.ref('isNew').getData();

// 获取表单数据
const _form = context.$component.form;
const data = _form.getFormData();

// 获取Object对象
const _object = context.object('命名空间_productList_CST');

// 新建
if (isNew) {
 data.命名空间_productStatus_CST = '审批中';
}
```

```
// 插入数据，支持批量操作
_object.insert([data]).then(function (response) {
 if (response && response.resCode === '0') {
 // 成功消息
 context.$message.success('新建成功! ');
 // 关闭弹出页面，仅能在弹出页面上使用
 context.$dialog.popin();
 }
});
} else {
 const id = context.$page.params.id;
 const params = {
 命名空间__productStatus__CST: '通过',
 命名空间__approvalComments__CST: data.命名空间__approvalComments__CST
 }
 // 根据记录ID更新数据
 _object.updateByID(id, params).then(function (response) {
 if (response && response.resCode === '0') {
 // 成功消息
 context.$message.success('审批成功! ');
 // 关闭弹出页面，仅能在弹出页面上使用
 context.$dialog.popin();
 }
 });
}
```

14. 单击页面上方的, 保存页面。

步骤5 返回列表页，添加事件。

1. 选择“新增行”按钮组件，在“事件”页签，单击“新增行”后的 删除当前新增行的方法，再单击“点击”后的“+”，为按钮添加自定义动作。

图 6-62 删除预置的事件





在自定义动作中，输入如下示例，用于弹出一个标准页面。其中，*命名空间\_\_NewApproval*为步骤4中创建的标准页面的名称。

```
// 弹出标准页面
context.$dialog.popup({
 title: '新建',
 page: '命名空间__NewApproval',
 width: 600,
 height: 400,
 footerHide: true,
 showCancel: true,
 okText: 'ok',
 params: {},
 onClose: function () {
 // 需要等待数据库操作完成
 setTimeout(() => {
 // 当前表格
 const _table = context.$component.table;

 // 刷新表格
 _table.doQuery();

 }, 100);
 }
});
```

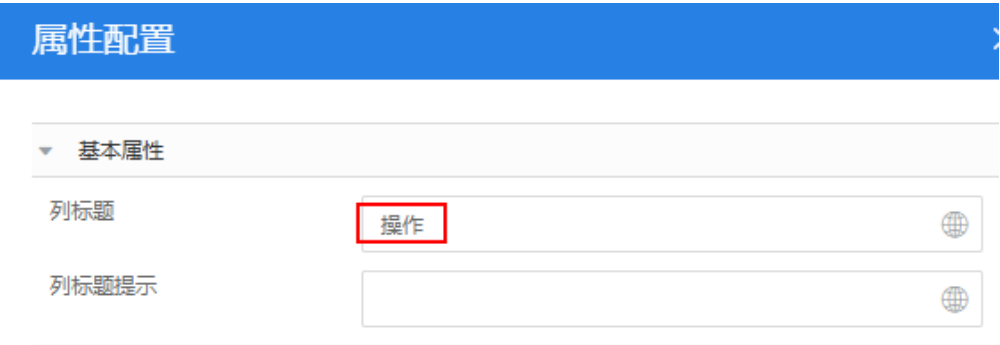
2. 选择**表格**组件，在“属性 > 表格列”中，单击，为表格添加一个操作列。

图 6-63 添加操作列



- 3. 单击新增操作列**Operation1**后的⚙️，进入属性配置页面。
- 4. 在“基本属性”中，将“列标题”修改为“操作”。

图 6-64 修改列标题



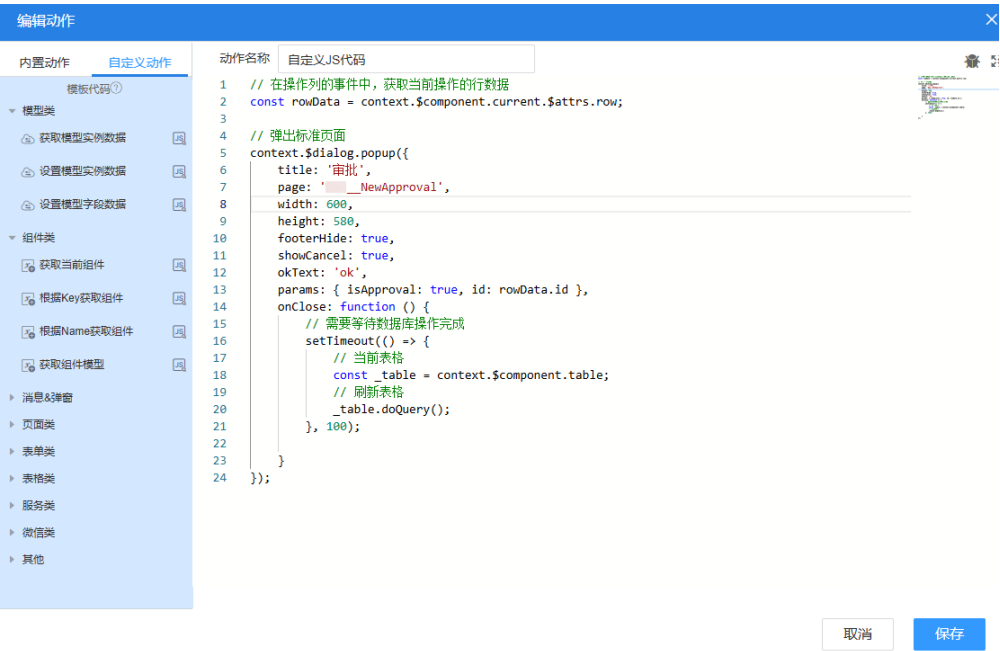
- 5. 在“操作按钮”中，单击“添加操作按钮”，新增一个操作按钮，修改按钮“标签”为“审批”。

图 6-65 新增审批操作按钮



6. 单击审批后的，再单击动作列表后的“+”，为操作列按钮添加自定义动作。

图 6-66 为操作列按钮添加自定义动作



在自定义动作中，输入如下事件，用于获取行数据。其中，命名空间 `_NewApproval` 为步骤4.1中创建的标准页面的名称。

```
// 在操作列的事件中，获取当前操作的行数据
const rowData = context.$component.current.$attrs.row;

// 弹出标准页面
context.$dialog.popup({
 title: '审批',
 page: '_命名空间__NewApproval',
 width: 600,
 height: 580,
 footerHide: true,
 showCancel: true,
 okText: 'ok',
 params: { isApproval: true, id: rowData.id },
 onClose: function () {
 // 需要等待数据库操作完成
 setTimeout(() => {
 // 当前表格
 const _table = context.$component.table;
 // 刷新表格
 _table.doQuery();
 }, 100);
 }
});
```

7. 单击页面上方的, 保存页面。

步骤6 验证效果。

- 1. 在列表页，单击页面上方的, 进入预览页面。
- 2. 单击“新增行”，输入商品名和商品描述信息，单击“保存”，添加一条数据。

图 6-67 添加一条行数据



保存成功后，在列表中可查看到已添加的数据，且“商品状态”为“审批中”。

图 6-68 查看已添加的数据



- 3. 单击新增数据后的“审批”，输入“审批意见”，单击“保存”，完成审批。

图 6-69 完成审批



保存成功后，可查看到“审批意见”为“同意”，“商品状态”变为“通过”。

图 6-70 查看审批意见



----结束

## 6.5 如何自定义华为云 Astro 轻应用标准页面组件包

### 6.5.1 自定义标准页面组件方案概述

#### 应用场景

华为云Astro轻应用提供了自定义低代码页面组件的能力，当预置的组件不能满足您的业务需求时，您可以按照本实践提供的组件规范开发符合自己业务的组件，以此来灵活地扩展平台功能，提高应用构建的效率和灵活性。

本实践以开发一个imgButton组件为例，向您介绍华为云Astro轻应用自定义组件的规范，同时阐述在实现各项功能时需修改的配置项，以及最终发布该自定义组件并在页面上使用的流程，希望在您开发自己的组件时能得到帮助。

本实践中自定义的imgButton组件，用于定制按钮样式、文本等内容。当标准页面预置的按钮组件样式、文本等内容无法满足需求时，可参照本实践进行自定义。

#### 方案优势

开发标准页面时，如果预置组件无法满足您的业务需求，通过自定义组件的方式可使组件功能更为灵活，从而增强低码页面的业务能力。

#### 约束与限制

组件的文件结构及内部配置项必须依据平台标准进行设置。

### 6.5.2 自定义标准页面组件操作流程

华为云Astro轻应用为您提供了一个自定义组件模板包workCards.zip，本章节介绍如何基于workCards.zip模板包自定义一个imgbutton组件，流程如表6-16所示。

表 6-16 自定义标准页面组件流程说明

| 序号 | 步骤                         | 说明                                                                                                                                   |
|----|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| 1  | 步骤一：获取自定义组件模板包             | 1. 获取自定义组件模板包workCards.zip。<br>2. 了解workCards.zip模板包的结构和定义规范。                                                                        |
| 2  | 步骤二：自定义标准页面组件<br>imgButton | 1. 解压workCards.zip模板包，修改解压后的文件夹名称为“imgButton”。<br>2. 根据业务需求，自定义标准页面组件。<br>3. 将imgButton文件夹中的内容进行打包，生成自定义组件包“dist\imgButton-***.zip”。 |
| 3  | 步骤三：上传自定义组件包               | 将自定义组件上传到华为云Astro轻应用。                                                                                                                |
| 4  | 步骤四：验证自定义组件是否符合预期          | 创建一个标准页面，将自定义组件拖拽到该页面中，组件能够正常显示。为自定义组件添加显示和隐藏事件，检查效果是否达到预期。                                                                          |

6.5.3 自定义标准页面组件操作步骤

步骤一：获取自定义组件模板包

为了增强标准页面组件的丰富性，华为云Astro轻应用为您提供了一个自定义组件模板workCards.zip，您可以基于该模板包自定义标准页面组件。开发标准页面组件前，请先了解自定义组件模板包的结构及定义规范，便于快速上手。

- 步骤1 单击[链接](#)，获取华为云Astro轻应用自定义组件模板workCards.zip。
- 步骤2 解压获取的workCards.zip包，了解自定义组件模板包的结构及定义规范。

图 6-71 workCards.zip 模板包结构

|                                                                                     |                   |                 |               |       |
|-------------------------------------------------------------------------------------|-------------------|-----------------|---------------|-------|
|  | build             | 2025/5/12 20:42 | 文件夹           |       |
|  | dist              | 2025/5/19 11:02 | 文件夹           |       |
|  | preview           | 2025/5/12 20:42 | 文件夹           |       |
|  | src               | 2025/5/12 20:42 | 文件夹           |       |
|  | .gitignore        | 2024/4/19 18:10 | GITIGNORE 文件  | 1 KB  |
|  | .npmrc            | 2024/4/22 9:31  | NPMRC 文件      | 1 KB  |
|  | package.json      | 2024/6/28 14:30 | JSON 文件       | 1 KB  |
|  | package-lock.json | 2024/6/28 11:12 | JSON 文件       | 31 KB |
|  | README.md         | 2024/4/19 14:03 | Markdown File | 1 KB  |
|  | vite.config.js    | 2024/6/28 14:28 | JavaScript 文件 | 2 KB  |

表 6-17 workCards.zip 模板包目录及文件说明

| 目录及文件             | 说明                                                                                     |
|-------------------|----------------------------------------------------------------------------------------|
| build             | 用于存放组件的打包脚本。                                                                           |
| dist              | 组件打包文件目录，文件打包后生成于此目录下。                                                                 |
| preview           | 组件预览文件夹。                                                                               |
| src               | 组件的主文件夹，组件的内容位于此文件夹中。<br>本实践中，自定义组件仅涉及修改“src\components\workCards”目录下的内容，详细介绍请参见表6-18。 |
| .gitignore        | Git的忽略文件。                                                                              |
| .npmrc            | 设置package.json中依赖包的安装源。                                                                |
| package.json      | 项目的依赖文件。                                                                               |
| package-lock.json | 用于锁定依赖文件的安装版本号。                                                                        |
| README.md         | 项目提示文件，用于编写对该组件的描述。                                                                    |
| vite.config.js    | Vite的配置文件。                                                                             |

图 6-72 src\components\workCards 目录结构

电脑 > DATA1 (D:) > DDD > workCards > src > components > workCards

| 名称             | 修改日期            | 类型                 | 大小   |
|----------------|-----------------|--------------------|------|
| design-time    | 2025/6/11 10:27 | 文件夹                |      |
| static         | 2025/6/11 10:27 | 文件夹                |      |
| cube.svg       | 2024/4/19 14:03 | Microsoft Edge ... | 6 KB |
| index.js       | 2024/6/26 17:23 | JavaScript 文件      | 1 KB |
| workCards.json | 2025/5/13 16:14 | JSON 文件            | 2 KB |
| workCards.svg  | 2024/6/27 9:51  | Microsoft Edge ... | 1 KB |
| workCards.vue  | 2024/6/28 15:39 | VUE 文件             | 2 KB |

表 6-18 src\components\workCards 目录结构说明

| 目录及文件                  | 说明         |
|------------------------|------------|
| design-time            | 开发态组件文件目录。 |
| static                 | 静态文件目录。    |
| cube.svg/workCards.svg | 业务中使用的图标。  |
| index.js               | 组件主入口文件。   |

| 目录及文件          | 说明                                                                                                                                                                                                                                                                                        |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| workCards.json | 组件的描述文件，用于自定义组件配置面板的规格，即校验和辅助编写自定义组件配置面板的JSON文件。workCards.json文件的编写需遵循JSON Schema规范，以便与主流的编辑器（如vscode、atom、sublime text、webstorm）集成。<br><br>workCards.json可以在插件包开发时修改，也可以在插件包编译后修改，编译前workCards.json文件存放在“src\components\workCards”目录下（如图6-73），编译后“dist\components\workCards”目录下（如表6-19）。 |
| workCards.vue  | 组件的主文件。                                                                                                                                                                                                                                                                                   |

图 6-73 编译前 workCards.json 文件所在位置

| 电脑 > DATA1 (D:) > workCards > src > components > workCards > |                 |                    |      |  |
|--------------------------------------------------------------|-----------------|--------------------|------|--|
| 名称                                                           | 修改日期            | 类型                 | 大小   |  |
| design-time                                                  | 2024/6/26 16:01 | 文件夹                |      |  |
| static                                                       | 2024/6/26 16:01 | 文件夹                |      |  |
| cube.svg                                                     | 2024/4/19 14:03 | Microsoft Edge ... | 6 KB |  |
| index.js                                                     | 2024/6/26 17:23 | JavaScript 文件      | 1 KB |  |
| workCards.json                                               | 2025/5/13 16:14 | JSON 文件            | 2 KB |  |
| workCards.svg                                                | 2024/6/27 9:51  | Microsoft Edge ... | 1 KB |  |
| workCards.vue                                                | 2024/6/28 15:39 | VUE 文件             | 2 KB |  |

图 6-74 编译后 workCards.json 文件所在位置

| 此电脑 > DATA1 (D:) > workCards > dist > components > workCards > |                 |              |  |  |
|----------------------------------------------------------------|-----------------|--------------|--|--|
| 名称                                                             | 修改日期            | 类型           |  |  |
| design-time                                                    | 2025/5/19 11:02 | 文件夹          |  |  |
| static                                                         | 2025/5/19 11:02 | 文件夹          |  |  |
| index.js                                                       | 2025/5/19 11:02 | JavaScript 文 |  |  |
| workCards.json                                                 | 2025/5/13 16:14 | JSON 文件      |  |  |



表 6-19 workCards.json 参数说明

| 参数          | 说明                                                                                                                                                                                                                                                                                                  |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name        | 组件的名称，此处配置的名称主要用于后台元数据的存储和解析等。 <ul style="list-style-type: none"><li>长度不能超过64个字符，必须以英文字母开头，只能由英文字母、数字和特殊字符（_和-）组成。建议由厂商名_组件名_版本号组成，例如aaa_img-button_1.1.1。</li><li>组件的名称会存放在界面元数据的type中，如type: "my-component1"。</li><li>zip包的名称应该和此处name的取值保持一致，例如name设置为imgbutton，zip包名必须为imgbutton.zip。</li></ul> |
| title       | 组件别名会显示在标准页面开发界面左侧“组件”页签下的扩展组件中，建议简短并能展示组件特性。<br>取值范围：1~100个字符。                                                                                                                                                                                                                                     |
| description | 组件的描述信息，即当鼠标悬停在组件上时，展示的组件描述信息。<br>取值范围：1~200个字符。                                                                                                                                                                                                                                                    |
| category    | 组件的分类，为了保持界面构建器组件分类的一致性，请使用华为云Astro轻应用提供的组件分类，具体分类如 <a href="#">表6-20</a> 所示。                                                                                                                                                                                                                       |
| icon        | 组件的图标，分为 <b>未选中时</b> 和 <b>选中时</b> 两个，必须为png格式，且每个png的大小不能超过16KB。                                                                                                                                                                                                                                    |
| 其他          | 用于定义包大小，编译后的组件包大小不能超过1MB。                                                                                                                                                                                                                                                                           |

表 6-20 华为云 Astro 轻应用组件分类

| 分类         | 说明                                                                           |
|------------|------------------------------------------------------------------------------|
| navigation | 导航组件，如菜单、工具栏或侧边栏。                                                            |
| data       | 数据组件，可以查看和编辑应用程序中的数据，例如Form、Table或List。                                      |
| common     | 通用组件，如标签、图片、标题或段落。                                                           |
| container  | 容器组件，即包含其他组件的容器。例如，栅格布局组件Row/Col、流式布局组件、Panel、Tabs、Collapse或WidgetContainer。 |
| input      | 显示和编辑实体属性，如文本框、日期选择器。                                                        |
| file       | 文件处理组件，例如文件上传下载组件、图片浏览组件或PDF预览组件。                                            |

| 分类     | 说明                        |
|--------|---------------------------|
| button | 触发动作的按钮，如保存按钮、页面跳转按钮。     |
| report | 聚合数据并以表格或图表的形式显示，如图表、透视表。 |
| widget | 业务卡片。                     |
| add-on | 扩展组件。                     |

----结束

## 步骤二：自定义标准页面组件 imgButton

本实践中，imgButton组件是采用Node.js开发的前端组件。如果没有特殊说明，本实践均基于VSCode工具，按照imgButton组件的功能点介绍自定义组件的开发过程。

**步骤1** 将**步骤一：获取自定义组件模板包**中获取的模板包进行解压，并将解压后的文件夹名称修改为“imgButton”。

**步骤2** 修改组件描述文件“workCards.json”，将“name”修改成“imgButton”，保存并退出。

```
{
 "name": "imgButton",
 "private": true,
 "version": "1.0.2",
 "type": "module",
 "scripts": {
 "dev": "vite",
 "build": "node ./build/build.mjs",
 "format": "prettier --write src/",
 "preview": "vite preview"
 },
 "dependencies": {
 "element-plus": "^2.4.2",
 "vue": "^3.3.4"
 },
 "devDependencies": {
 "@vitejs/plugin-vue": "^4.6.2",
 "jszip": "^3.10.1",
 "kolorist": "^1.8.0",
 "sass": "^1.69.5",
 "unplugin-element-plus": "^0.8.0",
 "unplugin-vue-components": "^0.25.1",
 "vite": "^4.4.5",
 "vue-style-loader": "^4.1.3"
 }
}
```

**步骤3** 修改“src”文件夹下“manifest.json”文件，将“name”和“components”的参数值修改成“imgButton”，保存并退出。

```
{
 "name": "imgButton",
 "version": "1.0.10",
 "type": "uiPackage",
 "package": {
 "components": [
 "components/imgButton/imgButton.json"
]
 },
 "images": {
```

```
"default": "./static/package-default.png",
"previews": [
 "./static/package-preview1.png",
 "./static/package-preview2.png"
],
},
"publisher": "l84182839",
"license": "MIT",
"keywords": [
 "UI Component"
],
"categories": [
 "ui:components"
],
"bugs:url": "",
"repository:url": "",
"dependencies": {}
}
```

**步骤4** 将“src\components”文件夹和文件夹下的文件名由“workCards”修改为“imgButton”。

图 6-75 修改文件夹名称



图 6-76 修改文件夹中的文件名

| 电脑 > DATA1 (D:) > D > imgButton > src > components > imgButton > |                 |                    |      |  |
|------------------------------------------------------------------|-----------------|--------------------|------|--|
| 名称                                                               | 修改日期            | 类型                 | 大小   |  |
| design-time                                                      | 2024/6/26 16:01 | 文件夹                |      |  |
| static                                                           | 2024/6/26 16:01 | 文件夹                |      |  |
| cube.svg                                                         | 2024/4/19 14:03 | Microsoft Edge ... | 6 KB |  |
| imgButton.json                                                   | 2025/5/13 16:14 | JSON 文件            | 2 KB |  |
| imgButton.svg                                                    | 2024/6/27 9:51  | Microsoft Edge ... | 1 KB |  |
| imgButton.vue                                                    | 2024/6/28 15:39 | VUE 文件             | 2 KB |  |
| index.js                                                         | 2024/6/26 17:23 | JavaScript 文件      | 1 KB |  |

**步骤5** 替换自定义组件的组件图标，此图标是自定义组件在页面构建器中显示的缩略图。

替换“src\components\imgButton\static”路径下的“default.png”和“hover.png”图标，“default.png”为默认状态的图标，“hover.png”为鼠标经过时高亮效果的图标。

**步骤6** 修改“src\components\imgButton”路径下的定义文件“imgButton.json”。

- 修改组件定义文件的基本属性。  
将“imgButton.json”文件中的“name”、“title”和“description”修改成“imgButton”，并将“properties”节点改为只包括“height”、“width”和“text”属性，具体代码如下：

```
{
 "name": "imgButton",
```

```

"title": "imgButton",
"description": "imgButton",
"category": "add-on",
"icon": {
 "default": "./static/default.png",
 "hover": "./static/hover.png"
},
"definition": {
 "props": {
 "type": "object",
 "properties": {
 "height": {
 "category": "common", //属性所属的标签，包括common（公共）、general（基础）和style（style）样式等
 "type": "integer", //该属性的数值类型，integer（数字输入框）、date（时间选择框）、string（输入框）、select（下拉框）和switch（开关）
 "title": "图片高度", //属性的标题
 "description": "设置图片高度", //属性的描述，悬浮在标题上显示
 "default": 16 //属性的默认值
 },
 "width": {
 "category": "common",
 "type": "integer",
 "title": "图片宽度",
 "description": "设置图片宽度",
 "default": 16
 },
 "text": {
 "category": "common",
 "type": "string",
 "title": "framework.label.text",
 "description": "按钮文本",
 "default": "Button"
 }
 }
 },
 "main": "./index.js",
 "mainDesignTime": "./design-time/index.js",
 "dnd": {
 "cname": "workCards",
 "dname": "workCards-design",
 "ctype": "Control",
 "displayName": "workCards",
 "selectable": true,
 "targetable": true,
 "draggable": true,
 "deletable": true,
 "wholeRow": true
 }
}

```

修改上述配置后，在华为云Astro轻应用标准页面中，显示的组件属性配置如图6-77所示。

图 6-77 组件属性显示效果

- 修改组件定义文件的事件，将事件名修改为“click”。

```
"on": {
 "type": "object",
 "properties": {
 "click": { // 代码中触发的事件名，在组件中用emit触发
 "title": "点击", //事件名称
 "description": "点击时执行的动作", //事件描述
 "type": "object"
 }
 }
},
```

- 修改组件定义文件的拖拽行为，将“cname”、“displayName”和“dname”修改成“imgButton”和“imgButtonDesign”。

```
"dnd": {
 "cname": "imgButton",
 "dname": "imgButtonDesign",
 "ctype": "Control",
 "displayName": "imgButton",
 "selectable": true,
 "targetable": true,
 "draggable": true,
 "deletable": true,
 "wholeRow": true
}
```

其中，“cname”为组件在应用引擎中显示的名称，“dname”为组件在应用开发工具中显示的名称，“displayName”为组件在页面构建器中显示的文本。

执行上述操作后，完整的“imgButton.json”文件配置示例如下：

```
{
 "name": "imgButton",
 "title": "imgButton",
 "description": "imgButton",
 "category": "add-on",
 "icon": {
 "default": "./static/default.png",
 "hover": "./static/hover.png"
 },
 "definition": {
 "props": {
 "type": "object",
 "properties": {
 "height": {
 "category": "common",
 "type": "integer",
 "title": "图片高度",
 "description": "设置图片高度",
 "default": 16
 }
 }
 }
 },
}
```

```

 "width": {
 "category": "common",
 "type": "integer",
 "title": "图片宽度",
 "description": "设置图片宽度",
 "default": 16
 },
 "text": {
 "category": "common",
 "type": "string",
 "title": "framework.label.text",
 "description": "按钮文本",
 "default": "Button"
 }
 },
 "on": {
 "type": "object",
 "properties": {
 "click": {
 "title": "点击",
 "description": "点击时执行的动作",
 "type": "object"
 }
 }
 },
 "methods": {
 }
},
"main": "./index.js",
"mainDesignTime": "./design-time/index.js",
"dnd": {
 "cname": "imgButton",
 "dname": "imgButton",
 "ctype": "Control",
 "displayName": "imgButton",
 "selectable": true,
 "targetable": true,
 "draggable": true,
 "deletable": true,
 "wholeRow": true
}
}

```

**步骤7** 修改“src/components/imgButton”路径下的视图文件“imgButton.vue”。

- 修改视图文件的布局结构。

在布局（div标记对）中添加一个“img”和一个“span”，并按示例配置，代码示例如下：

```


 {{ currentText }}


```

其中，“img”为imgButton按钮中的图片，“span”为imgButton按钮中的文本。

- 修改视图文件“imgButton.vue”的属性定义，修改“cssprefix”为“imgButton”，并添加“height”、“width”和“text”属性。

```

<script>
const cssprefix = "imgButton";

import { defineComponent } from "vue";

```

```
export default defineComponent({
 name: "imgButton",
 data() {
 return {
 currentText: this.text,
 };
 },
 props: {
 /*图片高度 */
 height: {
 type: [String, Number],
 default: "",
 },
 /*图片宽度 */
 width: {
 type: [String, Number],
 default: "",
 },
 /*标签文本内容 */
 text: {
 type: String,
 default: "Button1",
 },
 },
});
```

- 修改视图文件“imgButton.vue”中的“computed”、“watch”和“methods”。

```
computed: {
 classes() {
 return cssprefix;
 },
 comHeight() {
 return this.height || "16";
 },
 comWidth() {
 return this.width || "16";
 },
},
watch: {
 text(newVal) {
 this.currentText = newVal;
 },
},
methods: {
 onclick(e) {
 this.$emit("click", e);
 },
 setButtonText(newText) {
 console.log("setButtonText params:", newText);
 if (newText) {
 this.currentText = newText;
 }
 },
},
};
```

表 6-21 “computed”、“watch”和“methods”参数说明

| 参数          | 说明                 |
|-------------|--------------------|
| cssprefix   | 此变量用于构造样式类名，增加灵活性。 |
| currentText | 用于显示按钮文本。          |
| comHeight   | 用于获取图片的高度。         |
| comWidth    | 用于获取图片的宽度。         |

| 参数            | 说明                                                           |
|---------------|--------------------------------------------------------------|
| classes       | 用于返回样式类名，也可以直接用字符串替换，这样写只是为了增加灵活性。                           |
| onclick       | 在图片按钮被单击时，去触发其单击事件的业务逻辑。                                     |
| setButtonText | 对外显露的方法，用于修改按钮中的文本。所有定义在method中的方法，都可在事件中直接调用，调用的方式如图6-78所示。 |

图 6-78 在事件中调用 method 中的方法

```
1 var _component = context.$component.current;
2 _component.setButtonText("newText");
3
```

- 修改视图文件“imgButton.vue”中的“style”，定义图片按钮的样式类，包括图片按钮外框、图片和文本三个样式类。

```
.imgButton {
 display: inline-block;
 background-color: #8cdcdb;
 border: blue1pxsolid;
 cursor: pointer;
}
.imgButton-img {
 margin-top: 5px;
 margin-left: 5px;
}
.imgButton-title {
 padding-bottom: 10px;
 padding-right: 10px;
 font-size: 16px;
 cursor: pointer;
}
```

**步骤8** 修改“src\components\imgButton\index.js”路径下的应用开发工具和引擎的控件定义文件“index.js”，将控件定义文件中的“workCards”修改为imgButton。

```
import imgButton from './imgButton.vue'
export default imgButton
```

**步骤9** 完成自定义组件的开发后，执行如下操作，生成自定义组件包。

1. 在本地的PC机上，快捷键**Windows+R**，输入**cmd**进入命令提示符系统。
2. 进入imgButton所在的文件夹，执行如下命令，生成自定义组件包“dist\imgButton-\*\*\*.zip”。

执行yarn命令时，如果提示找不到，请先执行“npm install -g yarn”命令安装。

```
yarn install
yarn run build
```



图 6-79 生成自定义组件包

```
$ node ./build/build.mjs
clear output directory: D:\文件\imgButton\dist
Scanning components at D:\文件\imgButton\src\components
building component 'imgButton' at D:\文件\imgButton\src\components\imgButton
vite v4.5.3 building for production...
✓ 5 modules transformed.
dist/components/imgButton/index.js 4.35 kB | gzip: 3.07 kB
✓ built in 608ms
vite v4.5.3 building for production...
✓ 5 modules transformed.
dist/components/imgButton/design-time/index.js 4.36 kB | gzip: 3.07 kB
✓ built in 62ms
✓ built component 'imgButton' success
building components package manifest
✓ manifest built success
building components package
✓ generate components package success: imgButton-1.0.0.zip
```

至此，您已顺利完成组件包的自定义与打包。本实践为您提供了

----结束

步骤三：上传自定义组件包

自定义组件包开发完成后，需要将其上传到华为云Astro轻应用中。上传后，该组件即可在标准页面中使用。

- 步骤1 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
- 步骤2 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。

图 6-80 应用开发页面



- 步骤3** 单击页面左上角的，选择“环境管理 > 环境配置”，进入环境配置页面。
- 步骤4** 在环境配置页面的顶部菜单栏中，选择“维护”。

图 6-81 选择维护



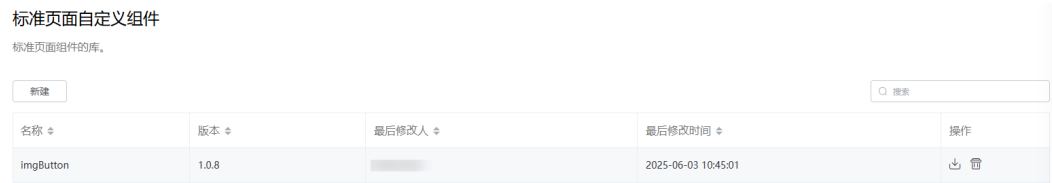
- 步骤5** 在左侧导航栏中，选择“全局元素 > 页面资产管理 > 自定义组件”。

图 6-82 选择自定义组件



- 步骤6** 在“标准页面自定义组件”页面中，单击“新建”，选择[自定义标准页面组件操作步骤](#)中的自定义组件imgButton.zip，完成组件的上传。
- 上传成功后，自动返回标准页面自定义组件页面，在组件列表中可查看到已上传的自定义组件imgButton。

图 6-83 查看已上传的自定义组件



----结束

步骤四：验证自定义组件是否符合预期

创建一个标准页面，将自定义组件拖拽到该页面中，确保组件能够正常显示。为自定义组件添加显示和隐藏事件，验证效果是否达到预期。

步骤1 创建一个低代码应用。

1. 在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
2. 在“应用”中，单击“新建低代码应用”或单击，进入新建低代码应用页面。
3. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。
4. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 6-84 创建一个空白应用

新建 空白应用

基本信息



添加图标

\* 标签

我的第一个应用

\* 名称

A

分类

请选择

描述

请输入

高级设置

取消

新建

表 6-22 新建空白应用参数说明

| 参数 | 说明                                                                                                                                                                                                                                                                                            | 示例      |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 标签 | 新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。                                                                                                                                                                                                                                                  | 我的第一个应用 |
| 名称 | 新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加 <b>命名空间</b> 。命名要求如下： <ul style="list-style-type: none"><li>长度不能超过31个字符，包括前缀命名空间的长度。</li><li>名称前的内容为命名空间，在华为云Astro轻应用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。</li><li>必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。</li></ul> | A       |

文档版本 01 (2025-09-04)

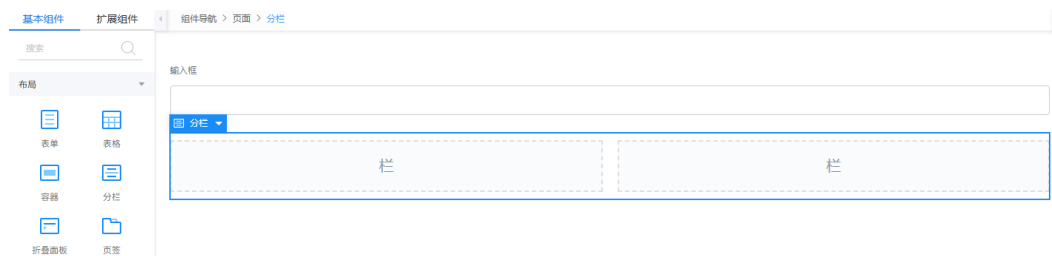
版权所有 © 华为云计算技术有限公司

227

**步骤2** 在应用设计器中，选择“界面”，单击页面后的“+”，新建一个标准页面。

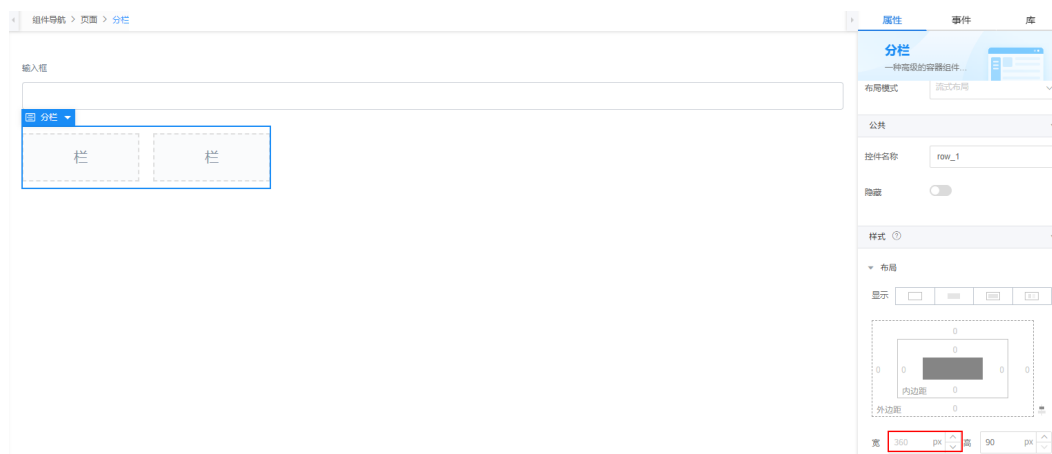
**步骤3** 从“基本组件”中，拖拽一个输入框组件和一个分栏组件到画布中，布局如图6-85。

图 6-85 拖入输入框和分栏组件



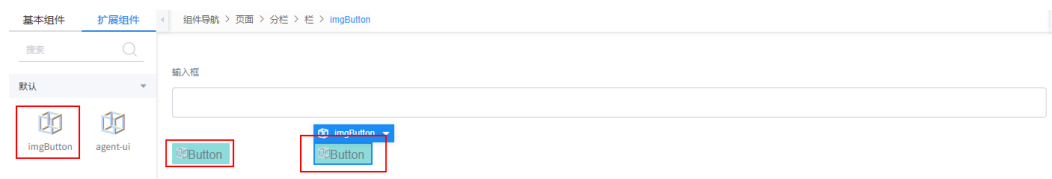
**步骤4** 选中分栏组件，在“属性 > 样式 > 布局”中，将“宽”设置为“360”px。

图 6-86 调整组件的宽度



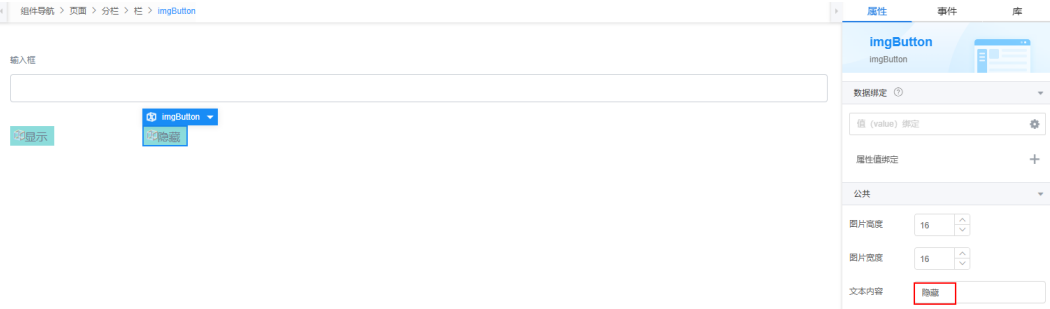
**步骤5** 在“扩展组件”中，拖拽自定义组件imgButton到分栏组件的两个栏中。

图 6-87 拖拽 imgButton 组件到分栏中



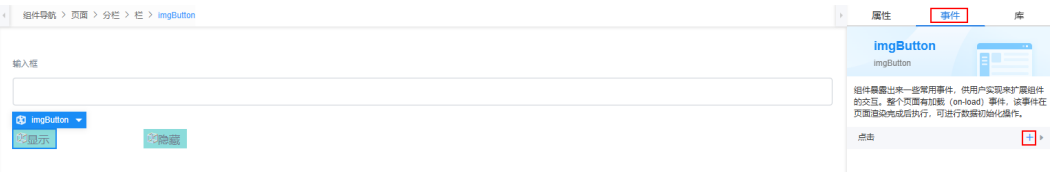
**步骤6** 选中自定义组件，在“属性 > 公共”中修改组件的“文本内容”为“显示”和“隐藏”。

图 6-88 修改组件的文本内容



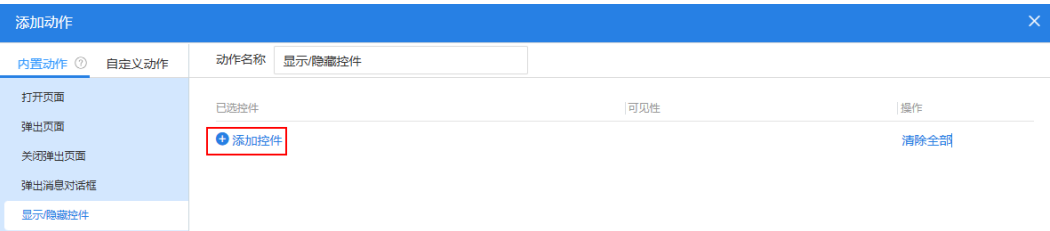
步骤7 选中显示组件，在“事件”页签中，单击“点击”后的“+”。

图 6-89 新建事件



步骤8 在“内置动作 > 显示/隐藏控件”中，单击“添加控件”。

图 6-90 单击添加控件

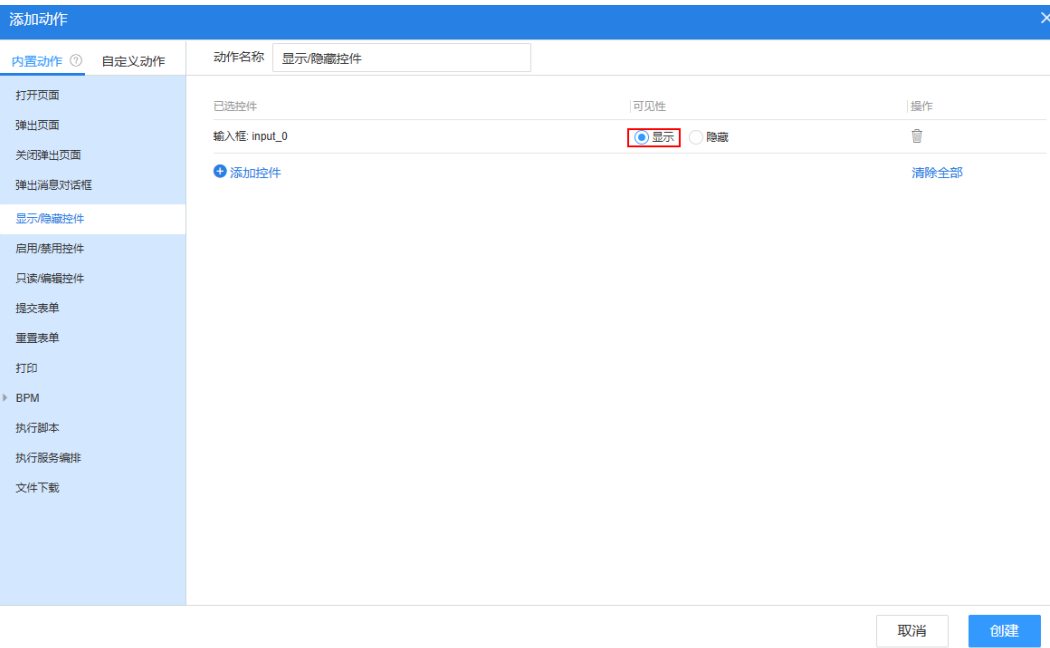


步骤9 选择“输入框”组件，将“显示”按钮的“可见性”设置“显示”，单击“创建”。

图 6-91 选中输入框组件

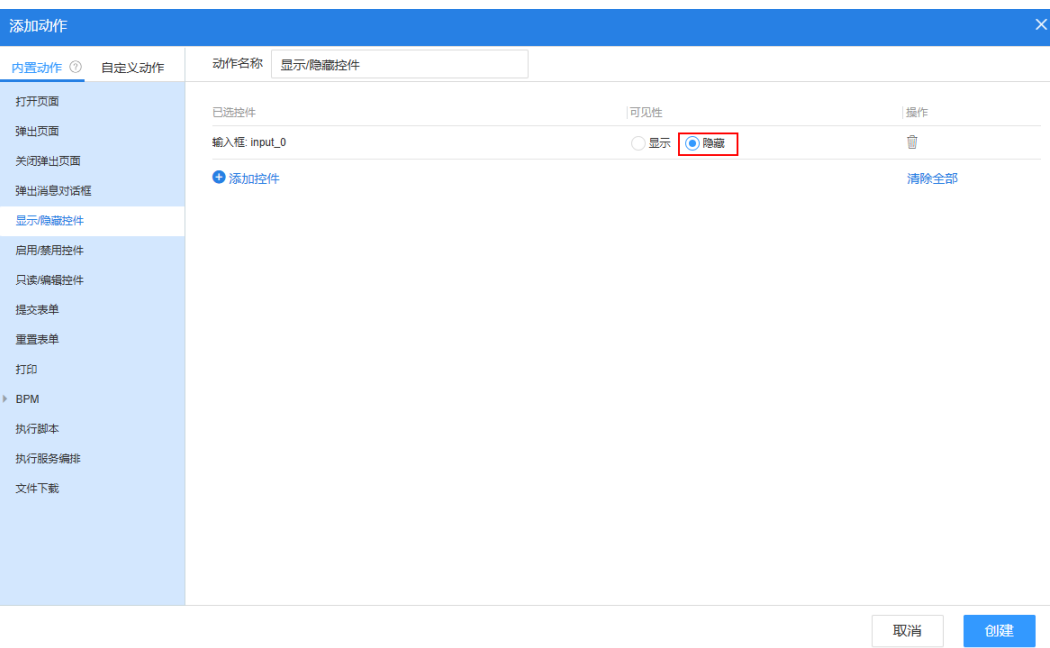


图 6-92 设置显示按钮的可见性为“显示”



步骤10 同样，按照上述操作，将隐藏按钮的“可见性”设置为“隐藏”。

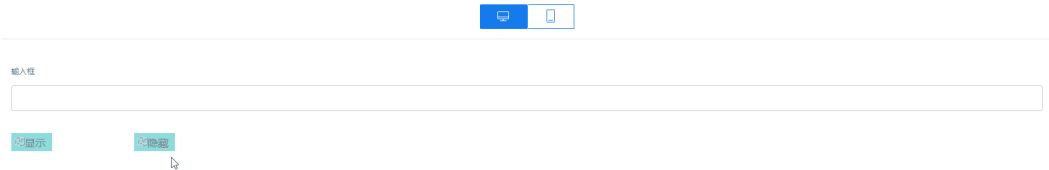
图 6-93 设置隐藏按钮的可见性为“隐藏”



步骤11 验证效果是否符合预期。

单击标准页面上方的保存设置，保存成功后单击，进入预览页面。在预览页面单击“隐藏”，输入框组件会隐藏，单击“显示”，输入框组件会再次显示。

图 6-94 预览效果



----结束

其他操作：更新自定义组件包

当开发的自定义组件功能有变动，即组件代码发生了变动时，支持对已上传的自定义组件进行更新。组件内容更新后，需要将“src/manifest.json”中的“version”版本号进行修改，假设当前版本号1.0.5，请将其修改为1.0.6，再重新执行yarn run build命令打包。打包完成后，在“dist”文件夹下会生成一个imgButton-1.0.6.zip的自定义组件包。

- 步骤1 在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
- 步骤2 单击页面左上角的, 选择“环境管理 > 环境配置”，进入环境配置页面。
- 步骤3 在环境配置页面的顶部菜单栏中，选择“维护”。
- 步骤4 在左侧导航栏中，选择“全局元素 > 页面资产管理 > 自定义组件”，进入标准页面自定义组件页面。

在自定义组件列表中，可看到当前的组件版本为“1.0.5”。

图 6-95 组件版本为 1.0.5

标准页面自定义组件

标准页面组件的库。

新建

搜索

| 名称        | 版本    | 最后修改人 | 最后修改时间              | 操作                                                                                                                                                                          |
|-----------|-------|-------|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| imgButton | 1.0.5 |       | 2025-06-03 10:45:01 |   |

- 步骤5 单击“新建”，选择“imgButton-1.0.6.zip”自定义组件包，单击“打开”。
- 组件更新后，在标准页面自定义组件列表中，可看到组件的版本已更新为“1.0.6”。进入使用自定义组件的标准页面，会发现组件内容已随之更新。

图 6-96 组件版本为 1.0.6

标准页面自定义组件

标准页面组件的库。

新建

搜索

| 名称        | 版本    | 最后修改人 | 最后修改时间              | 操作            |
|-----------|-------|-------|---------------------|---------------|
| imgButton | 1.0.6 |       | 2025-06-11 11:21:21 | <div>📄🗑</div> |

----结束



# 7 高级页面专项

## 7.1 使用华为云 Astro 轻应用自定义组件在页面中的属性

### 期望实现效果

组件预置的属性不能满足您的业务需求时，支持为组件自定义属性。例如，为组件 `widget_demo_property` 自定义 `Text Property`、`Checkbox Property` 和 `Select Property` 三个属性。组件中自定义属性在页面中的最终呈现效果如[图7-1](#)所示。

图 7-1 自定义组件属性

Text Property

text

Checkbox Property

true

Select Property

option1

### 功能实现方法

#### 步骤1 下载组件模板。

1. 在华为云Astro轻应用服务控制台的主页中，单击“进入首页”，进入应用开发页面。

2. 单击，选择“环境管理 > 环境配置”，进入环境配置。
  3. 在主菜单中，选择“维护”。
  4. 在左侧导航栏中，选择“全局元素 > 页面资产管理 > 组件模板”。
  5. 在组件列表中，单击“widgetPropertyTemplate”，进入模板详情页。
  6. 单击“下载”，设置组件的名称为“widget\_demo\_property”，单击“保存”，将模板下载到本地。
- 如果选择“下载原始模板”，下载到本地的组件包中，组件名称不会被修改。

图 7-2 保存模板



## 步骤2 自定义组件属性。

1. 在“widget\_demo\_property.editor.js”的“propertiesConfig”中，定义Widget属性，包含属性的类型、名称和在界面展示的标签名。

如下加粗代码所示，“widget\_demo\_property.editor.js”中分别定义了text、checkbox以及select类型的三个属性参数。

```
widget_demo_property = widget_demo_property.extend({
 /
 Config to define Widget Properties
 */
 propertiesConfig:[{
 config: [{
 "type": "text",
 "name": "textProperty",
 "label": "Text Property",
 "value": "text"
 },
 {
 "type": "checkbox",
 "name": "checkboxProperty",
 "label": "Checkbox Property",
 "value": "true"
 },
 {
 "type": "select",
 "name": "selectProperty",
 "label": "Select Property",
 "options": [{
 "label": "option1",
 "value": "option1",
 "selected": "true"
 },
 {
 "label": "option2",
 "value": "option2"
 }
]
 }
]
})
```

```

 }
],
},
/
Triggered when the user Creates a new widget and used to initialize the widget properties
*/
create : function(cbk)
{
 if(cbk)
 {
 this._super();
 cbk();
 }
}
});

var params = {};
Studio.registerWidget("widget_demo_property", "widget_demo_property", params);

```

其中：

- type: 属性的类型。
- name: 属性的名称。
- label: 属性在界面展示的标签名。
- value: 属性的默认取值。如果属性是select类型，则需要定义选项“options”。

2. 在“widget\_demo\_property.js”中，通过定义widgetProperties变量“var widgetProperties = thisObj.getProperties();”，调用“thisObj.getProperties”方法，来获取到这些属性。

```

var widget_demo_property = StudioWidgetWrapper.extend({
/
Triggered when initializing a widget and will have the code that invokes rendering of the widget
*/
init : function()
{
 var thisObj = this;
 thisObj._super.apply(thisObj, arguments);
 thisObj.render();
 if((typeof(Studio) != "undefined") && Studio)
 {
 /
 Register custom event or action here, and trigger the event afterwards.
 Studio.registerEvents(thisObj, "", "", EventConfig),
 Studio.registerAction(thisObj, "", "", ActionConfig, $.proxy(this.Cbk, this),);
 thisObj.triggerEvent("",)
 */
 }
},

/
Triggered from init method and is used to render the widget
*/
render : function()
{
 var thisObj = this;
 var widgetProperties = thisObj.getProperties();
 var elem = thisObj.getContainer();
 var items = thisObj.getItems();
 var connectorProperties = thisObj.getConnectorProperties();

 /
 API to get base path of your uploaded widget API file
 */
 var widgetBasePath = thisObj.getWidgetBasePath();
 if(elem)
 {

```

```
var vm = new Vue({
 el: $("#widget_demo_property", elem)[0],
 data: {
 form: {
 textProperty: widgetProperties.textProperty,
 checkboxProperty: widgetProperties.checkboxProperty,
 selectProperty: widgetProperties.selectProperty
 }
 }
});

/
API to bind global events to the item DOM, it should not be deleted if there will some events to
trigger in this widget.
*/
thisObj.sksBindItemEvent();

/
API to refresh the previously bound events when a resize or orientation change occurs.
*/
$(window).resize(function() {
 thisObj.sksRefreshEvents();
});
},
});
```

3. 在“widget\_demo\_property.ftl”中，定义渲染页面，属性配置为只读模式。

```
<div id="widget_demo_property">
 <el-form :model="form">
 <el-form-item label="Text Property">
 <el-input v-model="form.textProperty" :readonly="true"></el-input>
 </el-form-item>
 <el-form-item label="Checkbox Property">
 <el-input v-model="form.checkboxProperty" :readonly="true"></el-input>
 </el-form-item>
 <el-form-item label="Select Property">
 <el-input v-model="form.selectProperty" :readonly="true"></el-input>
 </el-form-item>
 </el-form>
</div>
```

4. 将开发好的组件代码压缩到后缀为“.zip”的压缩文件中，也可以单击[链接](#)，获取组件样例包“widget\_demo\_property.zip”。

### 步骤3 在组件库中上传组件包。

1. 在环境配置的“维护 > 全局元素 > 页面资产管理 > 组件”中，单击“提交新组件”。
2. 在提交新组件页面，设置组件基本信息，并上传压缩文件，单击“提交”。

图 7-3 上传自定义组件示例

+

上传图标

\* 名字

widgetdemoproperty

\* 上传源文件(.zip)

请选择源文件(.zip)

widget\_demo\_property.zip

组件ID

t0c

emc\_

widgetdemoproperty

分类

请选择

领域

通用

场景

高级页面

\* 发行说明

自定义组件

表 7-1 上传自定义组件参数说明

| 参数    | 说明                       | 示例                               |
|-------|--------------------------|----------------------------------|
| 名字    | 新提交组件的名称，系统会根据组件包名称自动填充。 | widgetdemoproperty               |
| 上传源文件 | 组件源文件包。                  | 选择步骤2中的 widget_demo_property.zip |
| 场景    | 组件包的应用场景，可同时选择多个。        | 高级页面                             |

| 参数   | 说明                                           | 示例    |
|------|----------------------------------------------|-------|
| 发行说明 | 组件的描述信息，按需进行设置。此处配置的信息，将会在组件详情页的“概况”页签中进行显示。 | 自定义组件 |

步骤4 关闭Vue3框架渲染组件开关。

本实践所涉及到的自定义组件是基于Vue2框架开发的，而系统是默认开启Vue3框架渲染组件的，所以您需要手动关掉Vue3框架渲染组件开关，否则拖拽组件到页面时会提示如下报错。

图 7-4 界面报错



1. 在应用设计器中，单击左侧导航栏中的“设置”。
2. 在“高级设置”中，取消勾选“页面组件的渲染框架由Vue2升级为Vue3”开关。

图 7-5 取消选中



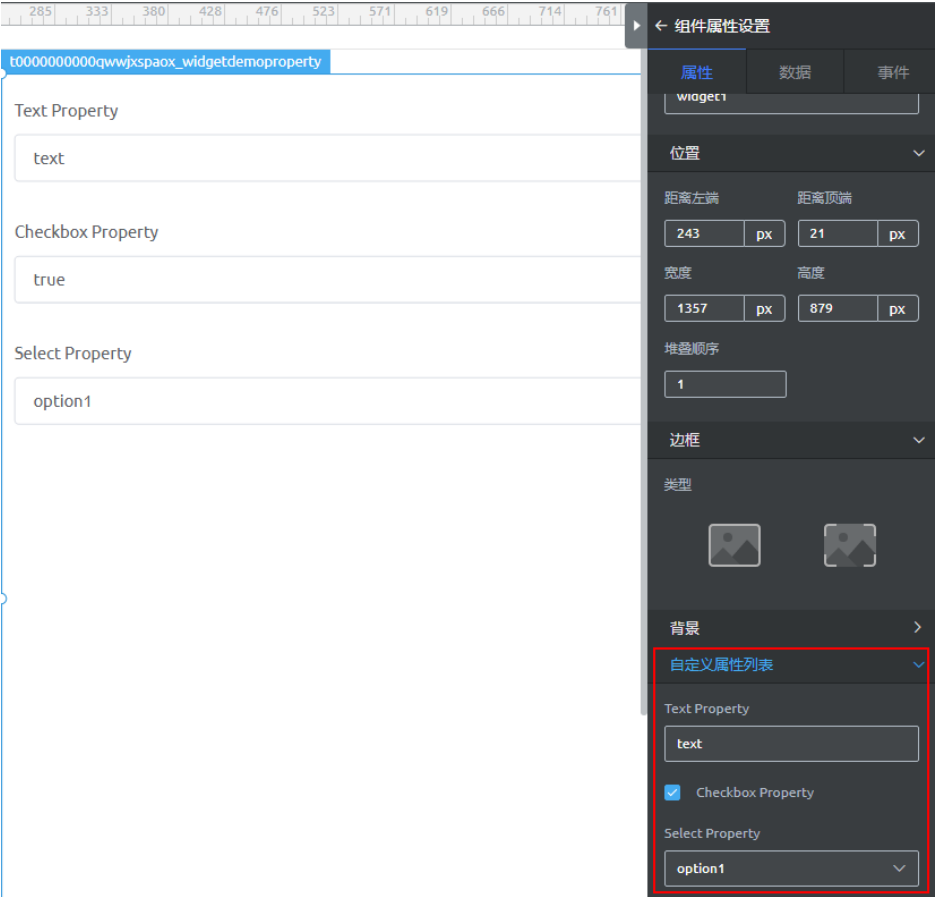
步骤5 在应用设计器中，选择“界面”，单击高级页面后的“+”，新建一个高级页面。

步骤6 在设计页面左上方单击，从“自定义”组件中拖拽步骤3中的组件至右侧空白页面。

步骤7 选中该组件，会在右侧显示该组件的属性配置面板。

步骤8 在“属性”页签下，“自定义属性列表”区域中可看到自定义的三个属性，根据需要进行修改。

图 7-6 编辑该 Widget 属性



----结束

## 7.2 使用华为云 Astro 轻应用为组件配置中英文语言属性

### 期望实现效果

为组件配置多语言属性，实现组件在不同语种环境下都可正常显示。在华为云Astro轻应用中对组件进行国际化配置，主要是修改国际化资源文件（i18n）。下面以为组件 widget\_demo\_i18n配置中文和英文两种语言属性为例，其最终实现效果如[图7-7](#)、[图7-8](#)。

图 7-7 中文环境下显示效果

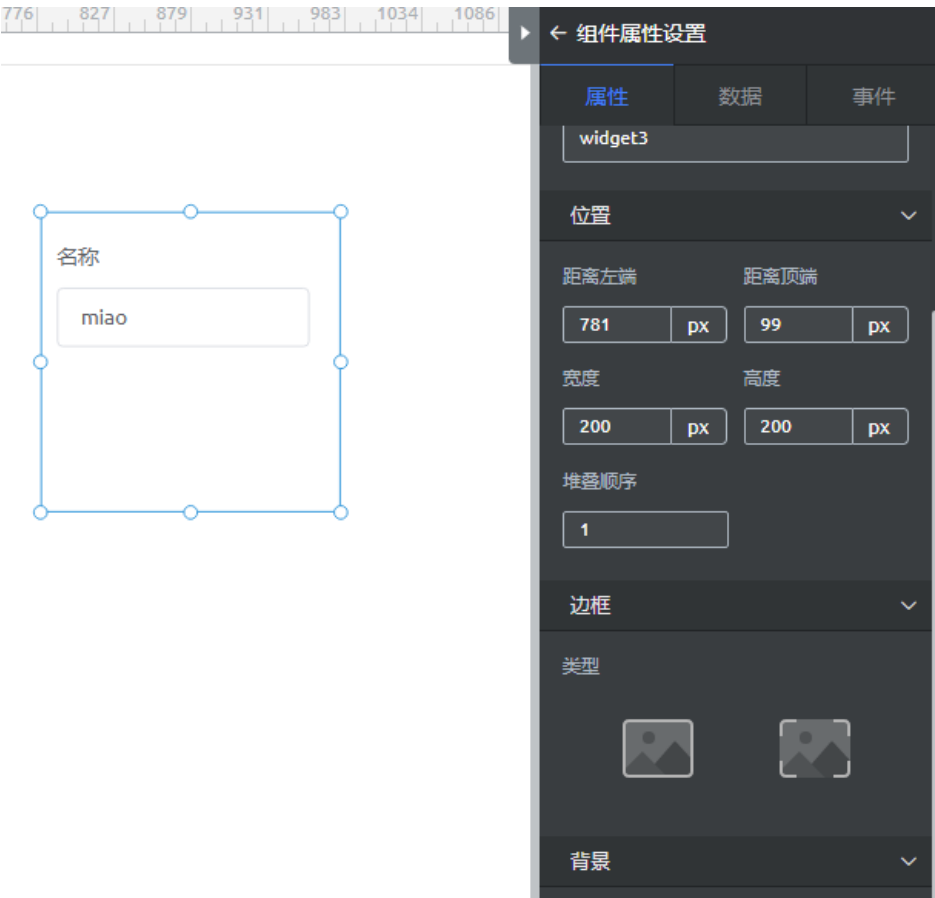
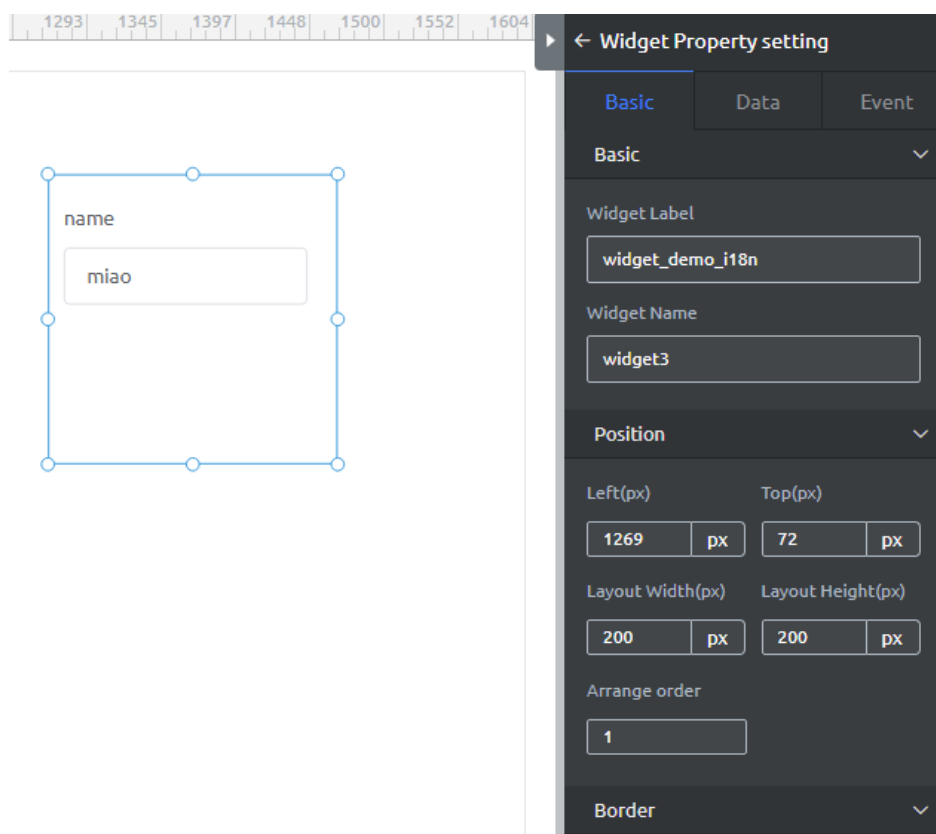




图 7-8 英文环境下显示效果

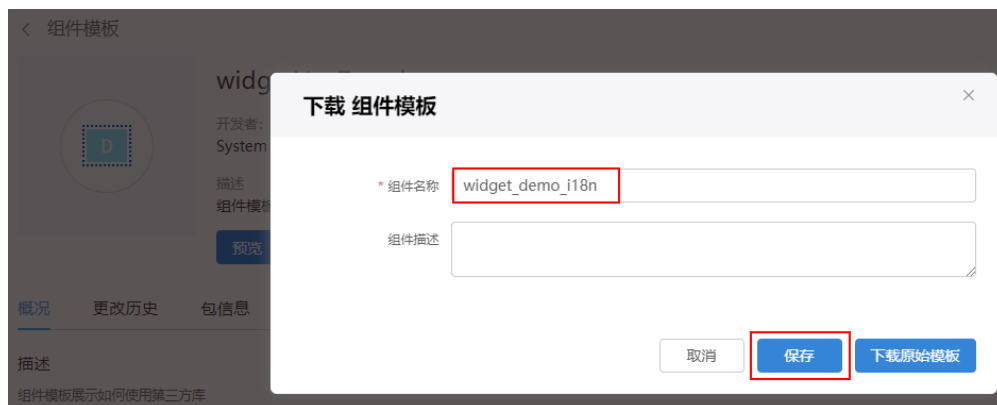


## 功能实现方法

### 步骤1 下载组件模板。

1. 在华为云Astro轻应用服务控制台的主页中，单击“进入首页”，进入应用开发页面。
2. 单击，选择“环境管理 > 环境配置”，进入环境配置。
3. 在主菜单中，选择“维护”。
4. 在左侧导航栏中，选择“全局元素 > 页面资产管理 > 组件模板”。
5. 在组件列表中，单击widgetVueTemplate，进入模板详情页。
6. 单击“下载”，设置组件的名称为“widget\_demo\_i18n”，单击“保存”，将模板下载到本地。  
如果选择“下载原始模板”，下载到本地的组件包中，组件名称不会被修改。

图 7-9 保存模板



**步骤2** 为组件设置多语言属性。

1. 增加“messages-en.json”和“messages-zh.json”国际化资源文件，“messages-zh.json”中需要使用unicode编码。

- “messages-zh.json”文件内容如下：

```
{
 "zh-CN": {
 "name": "\u540d\u79f0"
 }
}
```

- “messages-en.json”文件内容如下：

```
{
 "en-US": {
 "name": "name"
 }
}
```

2. 在“packageinfo.json”文件中，增加i18n节点，指定国际化资源文件的文件名。并增加requires节点，指定需要依赖的Vue和Vuel18n库。

其中，库文件名称和版本号，可在库详情页面获取。

```
{
 "widgetApi": [
 {
 "name": "widget_demo_i18n"
 }
],
 "widgetDescription": "widget i18n demo",
 "authorName": "test",
 "width": "",
 "height": "",
 "i18n": [
 {
 "name": "messages-en"
 },
 {
 "name": "messages-zh"
 }
],
 "requires": [
 {
 "name": "global_Vue",
 "version": "100.7"
 },
 {
 "name": "global_Vue18n",
 "version": "100.7"
 },
 {
 "name": "global_Element",

```

```
"version": "100.8"
}
]
}
```

在“requires”里增加库文件时，需要注意某些库文件之间有依赖关系，增加库文件需要有先后顺序，例如“global\_Vuel18n”是基于“global\_Vue”的，需要写在“global\_Vue”之后。

3. 在“widget\_demo\_i18n.js”的render方法中，使用平台提供的“HttpUtils.getI18n”方法返回一个“i18n”变量，并新建Vue实例传入该“i18n”变量。

```
var i18n = HttpUtils.getI18n({
 locale: HttpUtils.getLocale(),
 messages: thisObj.getMessages()
});

var vm = new Vue({
 el: $("#widget_demo_i18n", elem)[0],
 i18n: i18n,
 data: {
 form: {
 name: "miao"
 }
 }
})
```

4. 在“widget\_demo\_i18n.ftl”中，通过Vuel18n提供的“\$t”方法，使用国际化资源。

```
<div id="widget_demo_i18n">
 <el-form :model="form" :inline="true">
 <el-form-item :label="$t('name')">
 <el-input v-model="form.name"></el-input>
 </el-form-item>
 </el-form>
</div>
```

5. 将开发好的组件代码压缩到后缀为“.zip”的压缩文件中，也可以单击[下载链接](#)，获取样例包“widget\_demo\_i18n.zip”。

### 步骤3 在组件库中上传组件包。

1. 在环境配置的“维护 > 全局元素 > 页面资产管理 > 组件”中，单击“提交新组件”。
2. 在提交新组件页面，设置组件基本信息，并上传压缩文件，单击“提交”。

图 7-10 上传自定义组件示例

+

上传图标

\* 名字

widgetdemoi18n

\* 上传源文件(.zip)

请选择源文件(.zip)

widget\_demo\_i18n.zip

组件ID

t0C

henc\_

widgetdemoi18n

分类

请选择

领域

通用

场景

高级页面

×

\* 发行说明

自定义组件

表 7-2 上传自定义组件参数说明

| 参数    | 说明                       | 示例                             |
|-------|--------------------------|--------------------------------|
| 名字    | 新提交组件的名称，系统会根据组件包名称自动填充。 | widgetdemoi18n                 |
| 上传源文件 | 组件源文件包。                  | 选择步骤2.5中的 widget_demo_i18n.zip |
| 场景    | 组件包的应用场景，可同时选择多个。        | 高级页面                           |

文档版本 01 (2025-09-04)

版权所有 © 华为云计算技术有限公司

244

| 参数   | 说明                                           | 示例    |
|------|----------------------------------------------|-------|
| 发行说明 | 组件的描述信息，按需进行设置。此处配置的信息，将会在组件详情页的“概况”页签中进行显示。 | 自定义组件 |

步骤4 关闭Vue3框架渲染组件开关。

本实践所涉及到的自定义组件是基于Vue2框架开发的，而系统是默认开启Vue3框架渲染组件的，所以您需要手动关掉Vue3框架渲染组件开关，否则拖拽组件到页面时会提示如下报错。

图 7-11 界面报错



- 1. 在应用设计器中，单击左侧导航栏中的“设置”。
- 2. 在“高级设置”中，取消勾选“页面组件的渲染框架由Vue2升级为Vue3”开关。

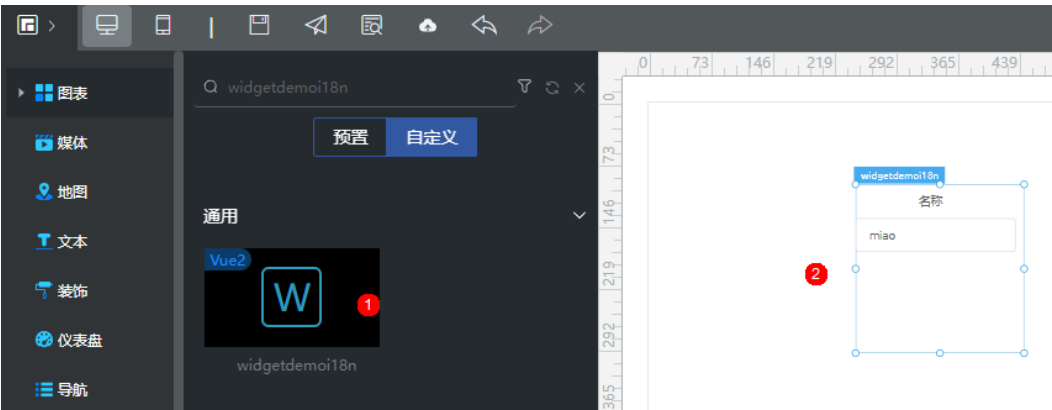
图 7-12 取消选中



步骤5 在应用设计器中，选择“界面”，单击高级页面后的“+”，新建一个高级页面。

步骤6 单击，从“全部 > 自定义”中拖拽widgetdemoi18n组件到右侧画布中。

图 7-13 拖拽组件到画布



步骤7 单击, 保存高级页面, 保存成功后单击, 发布高级页面。

步骤8 发布成功后, 单击, 预览效果。

切换环境语言, 在不同语种下, 在页面中选中Widget, 会在右侧显示该组件的属性配置面板, 查看组件属性是否符合预期。

----结束

## 7.3 使用华为云 Astro 轻应用创建高级页面适配多终端显示

### 期望实现效果

当用户开发的高级页面应用于多种设备时, 如何才能保证在不同大小的设备上, 能够呈现同样的网页? 为此, 华为云Astro轻应用提供了高级页面的电脑端和移动端两种终端视图、流式布局的响应式布局, 并为绝对布局提供“拉伸”功能辅助自适应。另外, 为确保自定义组件能够自适应不同分辨率设备, 在组件开发中用户需要遵从响应式布局设计规范。组件的响应式设计, 是高级页面适配多终端的重要前提。下面以开发一个满足响应式布局的商品列表组件为例, 向您介绍如何适配多终端。

商品列表组件可以应用于不同分辨率的手机端和电脑端, 组件中商品排布能够根据屏幕或者浏览器窗口大小自动调节。

### 功能实现方法

步骤1 下载组件模板。

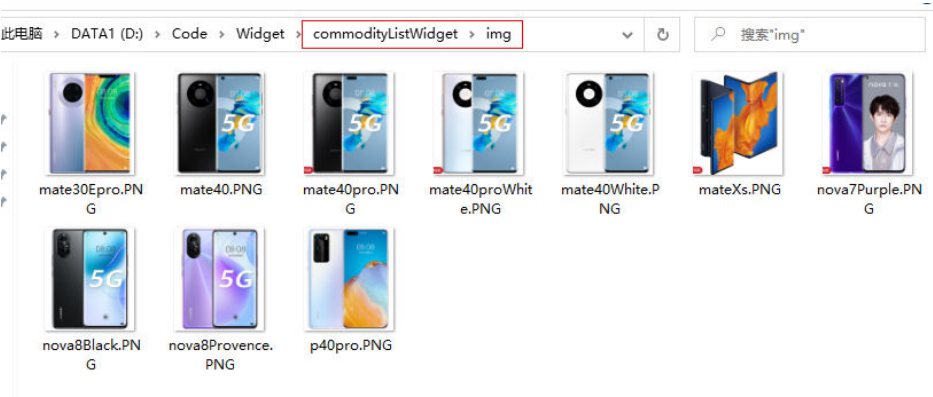
1. 在华为云Astro轻应用服务控制台的主页中, 单击“进入首页”, 进入应用开发页面。
2. 单击, 选择“环境管理 > 环境配置”, 进入环境配置。
3. 在主菜单中, 选择“维护”。
4. 在左侧导航栏中, 选择“全局元素 > 页面资产管理 > 组件模板”。
5. 在组件列表中, 单击widgetVueTemplate, 进入模板详情页。
6. 单击“下载”, 设置组件的名称为“commodityListWidget”, 单击“保存”, 将模板下载到本地。  
如果选择“下载原始模板”, 下载到本地的组件包中, 组件名称不会被修改。

图 7-14 保存模板



**步骤2** 解压下载到本地的模板包，新建“img”文件夹，存放商品图片，此处存放了10张商品图片，分辨率统一为380x440，如图7-15所示。

图 7-15 新建 img 存在商品图片



**步骤3** 在commodityListWidget.js中，新建Commodity对象并为其赋值。

其中，Commodity对象包含三个属性，分别为src（商品图片路径）、title（商品标题）和content（商品描述）。

```
var vm = new Vue({
 el: $("#widgetVueTemplate",elem)[0],
 data:{
 Commodity: [
 {
 src: widgetBasePath+"img/mate40.PNG",
 title: "HUAWEI Mate 40",
 content: "HUAWEI Mate 40 5G 全网通 8GB+128GB（亮黑色）"
 },
 {
 src: widgetBasePath+"img/mate40White.PNG",
 title: "HUAWEI Mate 40",
 content: "HUAWEI Mate 40 5G 全网通 8GB+128GB（釉白色）"
 },
 {
 src: widgetBasePath+"img/mate40pro.PNG",
 title: "HUAWEI Mate 40 Pro",
 content: "HUAWEI Mate 40 Pro 全网通 8GB+128GB（亮黑色）"
 },
 {
 src: widgetBasePath+"img/mate40proWhite.PNG",
 title: "HUAWEI Mate 40 Pro",
 content: "HUAWEI Mate 40 Pro 全网通 8GB+256GB（秘银色）"
 }
]
 }
})
```

```

 {
 src: widgetBasePath+"img/mateXs.PNG",
 title: "HUAWEI Mate Xs",
 content: "HUAWEI Mate Xs 5G 全网通 8GB+512GB (星际蓝)"
 },
 {
 src: widgetBasePath+"img/mate30Epro.PNG",
 title: "HUAWEI Mate 30E Pro",
 content: "HUAWEI Mate 30E Pro全网通 8GB+128GB (星河银)"
 },
 {
 src: widgetBasePath+"img/p40pro.PNG",
 title: "HUAWEI P40 Pro",
 content: "HUAWEI P40 Pro 5G 全网通 8GB+256GB (零度白)"
 },
 {
 src: widgetBasePath+"img/nova8Black.PNG",
 title: "HUAWEI nova 8",
 content: "HUAWEI nova 8 8GB+128GB 全网通版 (亮黑色)"
 },
 {
 src: widgetBasePath+"img/nova8Provence.PNG",
 title: "HUAWEI nova 8",
 content: "HUAWEI nova 8 8GB+128GB 全网通版 (普罗旺斯)"
 },
 {
 src: widgetBasePath+"img/nova7Purple.PNG",
 title: "HUAWEI nova 7",
 content: "HUAWEI nova 7 8GB+128GB 全网通版 (仲夏紫)"
 }
]
}
});

```

**步骤4** 在commodityListWidget.ftl中定义渲染页面，遍历Commodity对象，将多个商品的图片、标题和描述显示于组件中。

```

<div id="widgetVueTemplate" style="margin: 20px;">
 <section id="boxes">
 <div class="container">
 <div class="box" v-for="obj in Commodity">

 <h3>{{obj.title}}</h3>
 <p>{{obj.content}}</p>
 </div>
 </div>
 </section>
</div>

```

**步骤5** 在commodityListWidget.css中，定义组件的样式。

其中，采用百分比调节单个商品宽度，并设置断点，通过@media查询改变不同视区宽度下商品显示宽度的百分比。

```

/*设置组件字体、内边距、外边距和背景颜色*/
#widgetVueTemplate{
 font:15px/1.5 Arial, Helvetica, 'Microsoft YaHei';
 padding:0;
 margin:0;
 background-color: #f4f4f4;
}

/*商品列表*/
#boxes{
 margin-top:10px;
}

/*商品列表宽度为boxes宽度80%*/
.container{
 width:80%;
 margin: auto;
}

```



```

 overflow: hidden;
 }
 /*单商品样式 默认宽度为.container宽度18%，单行商品最大个数： 5*/
 #boxes .box{
 background: #ffffff;
 float:left;
 text-align: center;
 width:18%;
 padding:5px;
 margin: 7px 7px 7px 7px;
 }

 /*商品图片样式 默认宽度为.box宽度95%，为保证在屏幕宽度较大时，图片比例合理，限制其最大宽度为200px*/
 #boxes .box img{
 width: 95%;
 max-width:200px;
 margin-top: 5px;
 }

 /*商品描述样式 防止描述文字行数不同导致的商品高度不同而产生排列错位，限制其最小高度为44px*/
 #boxes .box p{
 min-height: 44px;
 }

 /*Media Queries 根据max-width（视区最大宽度）设置断点，调节单个商品的宽度以改变单行商品最大个数*/
 /*视区宽度介于1600px-1700px，单行商品最大个数： 4*/
 @media(max-width:1700px){
 #boxes .box{
 float:left;
 text-align: center;
 width: 22%;
 }
 }
 /*视区宽度介于1400px-1600px，单行商品最大个数： 3*/
 @media(max-width:1600px){
 #boxes .box{
 float:left;
 text-align: center;
 width: 30%;
 }
 }
 /*视区宽度介于800px-1400px，单行商品最大个数： 2*/
 @media(max-width:1400px){
 #boxes .box{
 float:left;
 text-align: center;
 width: 44%;
 }
 }
 /*视区宽度800px以下，单行商品最大个数： 1*/
 @media(max-width:800px){
 #boxes .box{
 float:none;
 text-align: center;
 width: 95%;
 }
 }
}

```

**步骤6** 设置完成后，重新打包（包名为commodityListWidget.zip）。

您也可以直接单击[commodityListWidget.zip](#)，下载commodityListWidget商品列表组件示例开发包。

**步骤7** 打包上传commodityListWidget商品列表组件。

1. 在环境配置页面的主菜单中，选择“维护”。
2. 在左侧导航栏中，选择“全局元素 > 页面资产管理 > 组件”。

3. 单击“提交新组件”，进入提交新组件页面。
4. 单击“请选择源文件(.zip)”，选择**步骤6**中的商品列表组件包commodityListWidget.zip，输入发行说明为“commodityListWidget”，单击“提交”。

#### 步骤8 关闭Vue3框架渲染组件开关。

本实践所涉及到的自定义组件是基于Vue2框架开发的，而系统是默认开启Vue3框架渲染组件的，所以您需要手动关掉Vue3框架渲染组件开关，否则拖拽组件到页面时会提示如下报错。

图 7-16 界面报错

❗ 当前渲染框架为VUE3，请核对插件是否适配当前渲染框架。

1. 在应用设计器中，单击左侧导航栏中的“设置”。
2. 在“高级设置”中，取消勾选“页面组件的渲染框架由Vue2升级为Vue3”开关。

图 7-17 取消选中



#### 步骤9 创建CommodityDisplay高级页面。

首次创建高级页面时，会出现视图选项，请勾选“电脑端”和“手机端”，布局类型选择“流式布局”（流式布局为自适应布局）。

图 7-18 新建 CommodityDisplay 高级页面

添加高级页面

页面模板

Q 搜索.....



空白页

\* 标签

CommodityDisplay

\* 名称

CommodityDisplay

描述

关键字

布局类型

☐ 绝对布局

☒ 流式布局

取消

添加

表 7-3 创建高级页面 CommodityDisplay 参数说明

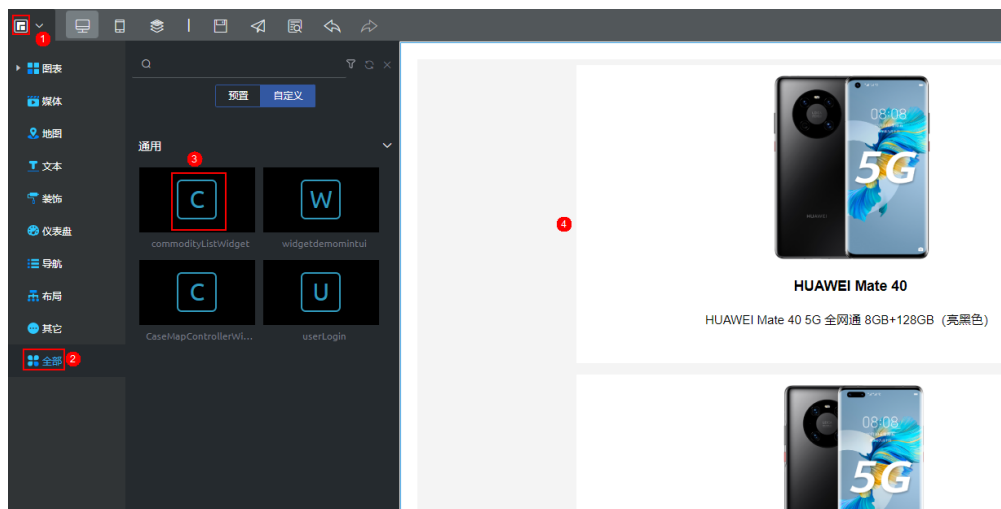
| 参数 | 说明                                                                                                                                 | 示例               |
|----|------------------------------------------------------------------------------------------------------------------------------------|------------------|
| 标签 | 新建高级页面的标签，创建后可修改。<br>取值范围：1~100个字符。                                                                                                | CommodityDisplay |
| 名称 | 新建高级页面的名称，创建后不可修改。命名要求如下： <ul style="list-style-type: none"><li>1~100个字符。</li><li>必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。</li></ul> | CommodityDisplay |

| 参数   | 说明                                                                                                                                                                                                     | 示例   |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 布局类型 | <p>选择高级页面的布局模式。</p> <ul style="list-style-type: none"> <li>绝对布局：在绝对布局中，每个组件可在页面中任意位置进行拖拽放置，组件的宽高可自定义设置。</li> <li>流式布局：在流式布局中，拖拽到页面中的组件，将根据从上到下、从左到右的顺序依次排列，组件的高度将根据组件内容大小进行自适应，宽度可按百分比进行配置。</li> </ul> | 流式布局 |

#### 步骤10 开发CommodityDisplay电脑端和移动端高级页面。

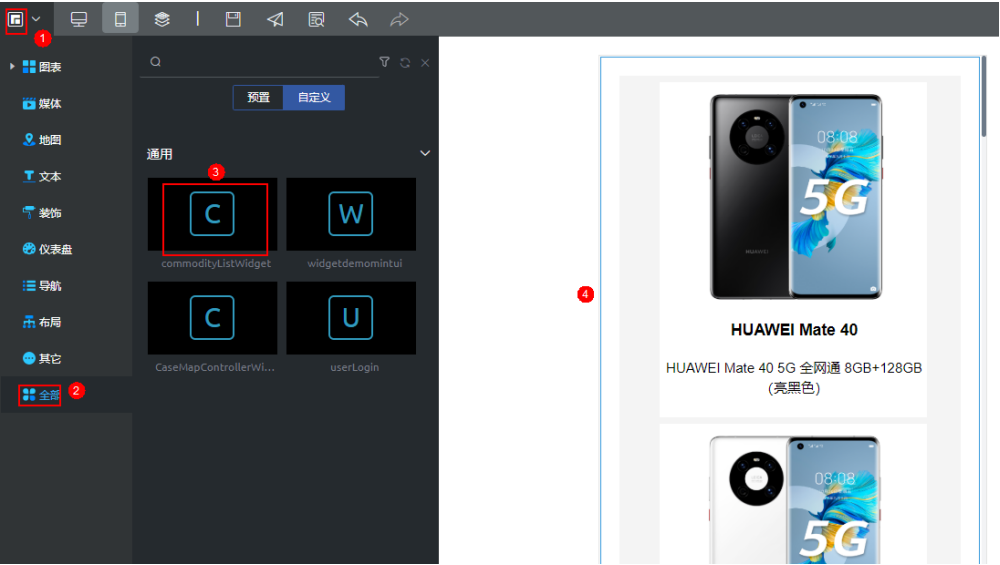
1. 在设计页面左上方，单击，从“全部 > 自定义”中拖拽commodityDetailsWidget组件到右侧画布中。

图 7-19 在电脑端拖拽 CommodityDisplay 组件



2. 在页面上方单击，切换到移动端，并拖拽commodityListWidget组件到右侧画布中。

图 7-20 在移动端拖拽 CommodityDisplay 组件



3. 单击页面上方的, 保存高级页面后, 单击, 发布高级页面。
4. 单击, 进入预览页面, 调整窗口大小, 查看效果是否符合预期。

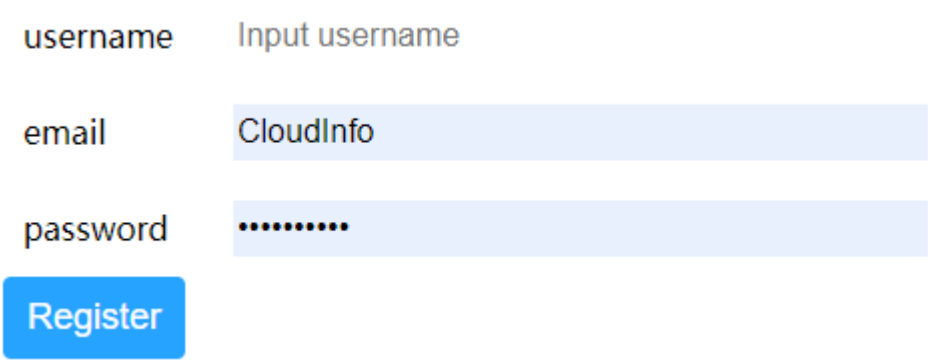
----结束

## 7.4 使用华为云 Astro 轻应用开发高级页面时如何引用第三方库

### 期望实现效果

在自定义高级页面组件开发过程中, 华为云Astro轻应用支持开发者直接引用第三方库, 在降低组件开发复杂度的同时, 丰富了组件的功能。库是支撑高级页面组件运行的第三方依赖, 如果缺少相应的库, 则高级页面组件不能正常运行。华为云Astro轻应用提供了一些系统预置库, 可在高级页面组件中直接引用或在页面设置中直接进行加载并使用。当系统预置的库无法满足需求时, 可以上传自定义库, 并加载到页面中使用。以某组件中需要使用Vue (系统预置库) 以及MintUI (自定义库) 为例, 介绍如何上传自定义库, 并在组件中使用库, 最终实现效果如图7-21所示。

图 7-21 引入库后效果

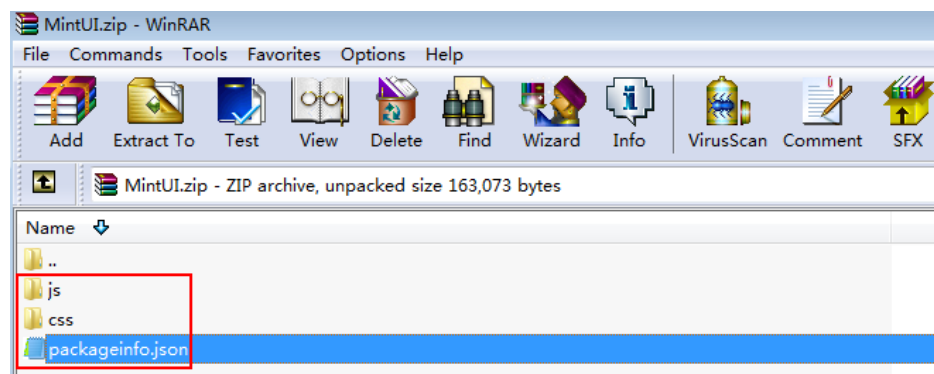


## 功能实现方法

### 步骤1 将自定义库的相关文件打成Zip包。

例如，在MintUI官网下载组件库的代码，增加一个packageinfo.json元数据描述文件，在该文件中列出Library包含的js和css文件名，并打成Zip包，如图7-22。您也可以单击[链接](#)，获取该包。

图 7-22 MintUI 库文件结构



packageinfo.json中必须要包含待引入的文件。例如，MintUI库需要引入文件“js/index.js”和“css/index.css”，请在packageinfo.json中添加这两个文件的描述。其中，“js”和“css”用于定义文件类型，“name”用于定义文件路径及名称。

```
{
 "js": [
 {
 "name": "js/index"
 }
],
 "css": [
 {
 "name": "css/index"
 }
]
}
```

### 步骤2 上传自定义库。

1. 在华为云Astro轻应用服务控制台的主页中，单击“进入首页”，进入应用开发页面。
2. 单击，选择“环境管理 > 环境配置”，进入环境配置。
3. 在主菜单中，选择“维护”。
4. 在左侧导航栏中，选择“全局元素 > 页面资产管理 > 库”。
5. 单击“提交新库”，进入提交新库页面。
6. 设置库的基本信息，上传步骤1中的zip包后，单击“提交”。

图 7-23 上传库

+

上传图标

\* 名字

MintUI

\* 上传源文件(.zip)

请选择源文件(.zip)

MintUI.zip

库ID

t00Cnemoc

MintUI

分类

请选择

库类型

请选择

\* 发行说明

自定义库

表 7-4 新建 MintUI 库参数说明

| 参数    | 说明                        | 示例       |
|-------|---------------------------|----------|
| 名字    | 新建库的名称。<br>取值范围：1~80个字符。  | MintUI   |
| 上传源文件 | 选择自定义库的压缩包。               | 选择步骤1中的库 |
| 库ID   | 上传库的ID，由字母及数字组成，且必须以字母开头。 | MintUI   |
| 发行说明  | 自定义库的描述信息，按需进行设置。         | 自定义库     |

### 步骤3 在高级页面组件中，引用库。

以自定义组件（widget\_demo\_mintui）为例，在该组件引用第三方库。

1. 在组件包的“packageinfo.json”文件中，增加requires节点，指定需要依赖库的库ID和版本号。

其中，“name”为库ID、“version”为库版本号数字部分。

例如，增加如下requires节点，库文件名称和版本号在库详情页面获取。

```
"requires": [
 {
 "name": "global_Vue",
 "version": "100.7"
 },
 {
 "name": "t0000000000fcsfrfcaks_MintUI",
 "version": "1.0.0"
 }
]
```

在“requires”里增加库文件时，需要注意某些库文件之间有依赖关系，增加库文件需要有先后顺序，例如“global\_Vue18n”是基于“global\_Vue”的，需要写在“global\_Vue”之后。

2. 在高级页面组件包的widget\_demo\_mintui.ftl中，写一个简单的表单DOM。

```
<div id="widget_demo_mintui">
 <mt-field label="username" placeholder="Input username" v-model="username"></mt-field>
 <mt-field label="email" placeholder="Input email" type="email" v-model="email"></mt-field>
 <mt-field label="password" placeholder="Input password" type="password" v-
modal="password"></mt-field>
 <mt-button type="primary" @click="submit">Register</mt-button>
</div>
```

3. 在高级页面组件包的widget\_demo\_mintui.js/render方法中，新增Vue实例。

```
Vue.use(MINT);
var vm = new Vue({
 el: $("#widget_demo_mintui", elem)[0],
 data:{
 username: "",
 email: "",
 password: ""
 },
 methods:{
 submit: function(){
 console.log(this.username + " registers");
 }
 }
})
```

4. 将修改后的内容，重新打包，您也可以单击[widget\\_demo\\_mintui.zip](#)，获取该组件包。

### 步骤4 返回环境配置，选择“维护 > 全局元素 > 页面资产管理 > 组件”，单击“提交新组件”，将自定义组件包上传至组件库。



图 7-24 上传组件

+

上传图标

\* 名字

widgetdemomintui

\* 上传源文件(.zip)

请选择源文件(.zip)

widget\_demo\_mintui.zip

组件ID

t00l

emc\_

widgetdemomintui

分类

请选择

领域

通用

场景

高级页面

×

\* 发行说明

自定义组件

表 7-5 上传自定义组件参数说明

| 参数    | 说明                       | 示例                               |
|-------|--------------------------|----------------------------------|
| 名字    | 新提交组件的名称，系统会根据组件包名称自动填充。 | widgetdemomintui                 |
| 上传源文件 | 组件源文件包。                  | 选择步骤3.4中的 widget_demo_mintui.zip |
| 场景    | 组件包的应用场景，可同时选择多个。        | 高级页面                             |

| 参数   | 说明                                           | 示例    |
|------|----------------------------------------------|-------|
| 发行说明 | 组件的描述信息，按需进行设置。此处配置的信息，将会在组件详情页的“概况”页签中进行显示。 | 自定义组件 |

步骤5 关闭Vue3框架渲染组件开关。

本实践所涉及到的自定义组件是基于Vue2框架开发的，而系统是默认开启Vue3框架渲染组件的，所以您需要手动关掉Vue3框架渲染组件开关，否则拖拽组件到页面时会提示如下报错。

图 7-25 界面报错



1. 在应用设计器中，单击左侧导航栏中的“设置”。
2. 在“高级设置”中，取消勾选“页面组件的渲染框架由Vue2升级为Vue3”开关。

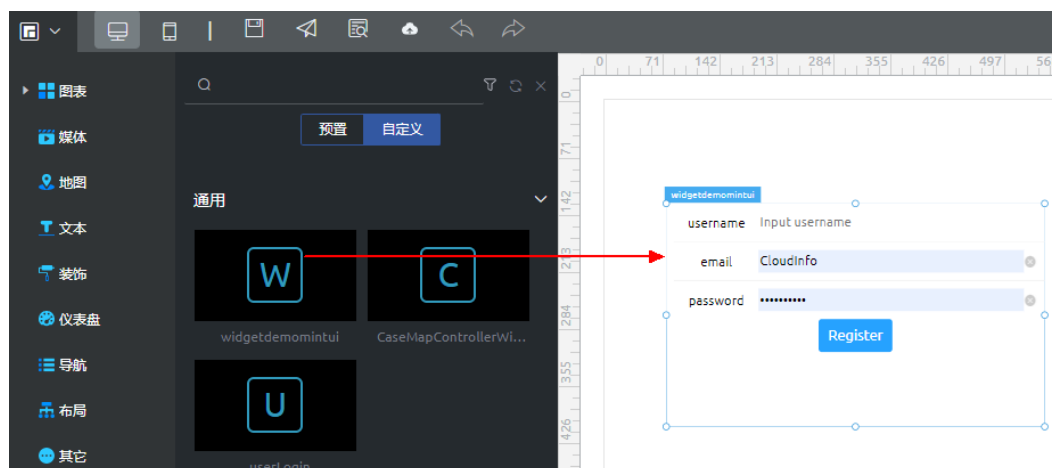
图 7-26 取消选中



步骤6 在应用设计器中，选择“界面”，单击高级页面后的“+”，新建一个高级页面。

步骤7 单击，从“全部 > 自定义”中拖拽widget\_demo\_mintui组件到右侧画布中。

图 7-27 拖拽组件到画布中



**步骤8** 单击页面上方的，保存高级页面后，单击，发布高级页面。

**步骤9** 单击，进入预览页面，查看效果是否符合预期。

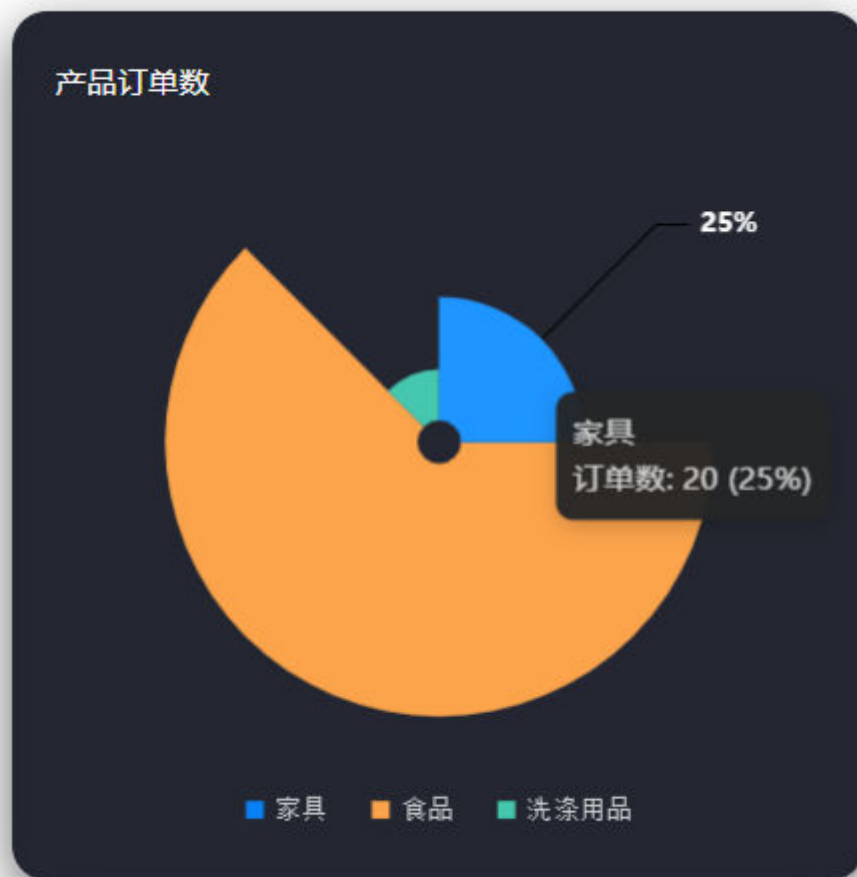
----结束

## 7.5 在华为云 Astro 轻应用高级页面中使用花瓣图展示订单数据

### 期望实现效果

高级页面中组件展示的数据除了系统预置的静态数据之外，还支持动态数据，即通过调用脚本、服务编排或对象等接口动态生成的数据。例如，将玫瑰花饼图的数据修改为订单对象中的数据。

图 7-28 实现效果



## 功能实现方法

### 步骤1 创建一个低代码应用。

1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
3. 在“应用”中，单击“新建低代码应用”或单击 ，进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。
4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。
5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 7-29 创建一个空白应用

新建 空白应用

基本信息

添加图标

\* 标签

我的第一个应用

\* 名称

A

分类

请选择

描述

请输入

高级设置

取消

新建

表 7-6 新建空白应用参数说明

| 参数 | 说明                                                                                                                                                                                                                                                                                   | 示例      |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 标签 | 新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。                                                                                                                                                                                                                                         | 我的第一个应用 |
| 名称 | 新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下： <ul style="list-style-type: none"><li>长度不能超过31个字符，包括前缀命名空间的长度。</li><li>名称前的内容为命名空间，在华为云Astro轻应用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。</li><li>必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。</li></ul> | A       |

步骤2 在应用设计器中，选择“界面”，单击高级页面后的“+”，新建一个高级页面。

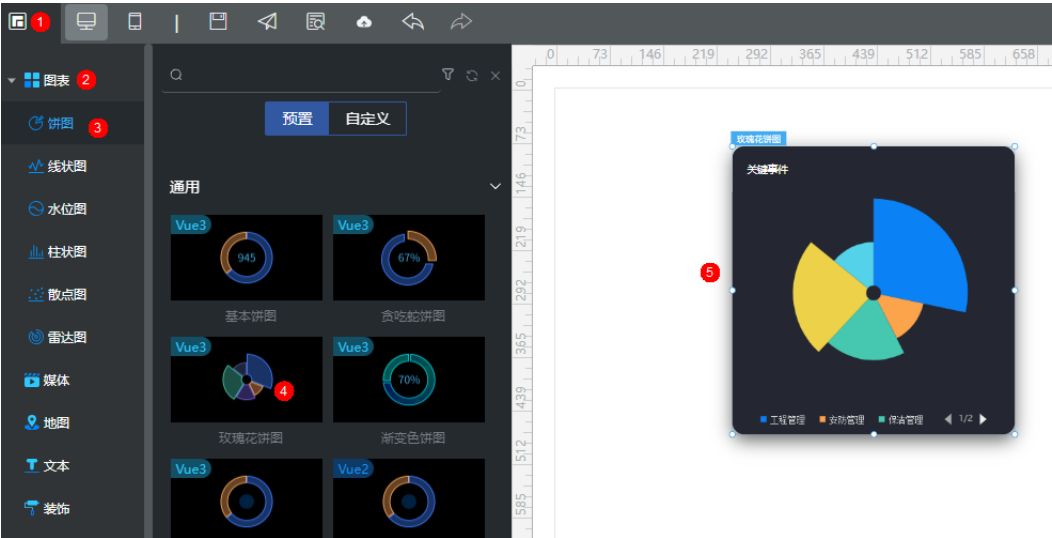
步骤3 单击，从“图表 > 饼图”中拖拽widget\_demo\_mintui组件到右侧画布中。

文档版本 01 (2025-09-04)

版权所有 © 华为云计算技术有限公司

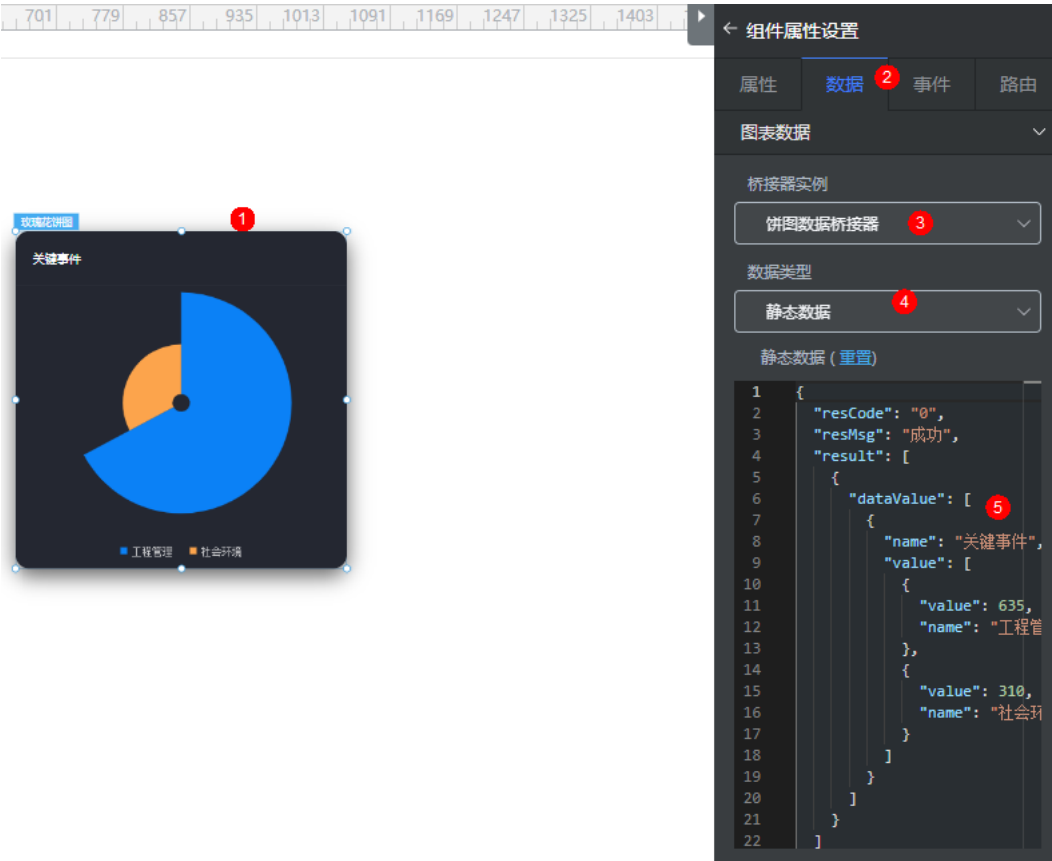
261

图 7-30 拖入一个玫瑰花饼图



步骤4 选中玫瑰花饼图，在“数据”中查看该组件对应的数据格式。

图 7-31 查看数据格式



步骤5 新建一个productList对象，并为对象添加字段和数据。

1. 在应用设计器的左侧导航栏中，选择“数据”，单击对象中的“+”。
2. 设置对象的名称和唯一标识为“productList”，单击“确定”。

图 7-32 创建对象 productList

创建新对象

您可以创建用来保存持久化数据的实体对象，对象里的数据存储在平台里面

从空白创建

从模板创建

您将创建单个对象。从零开始设计对象的字段，并添加测试数据

\* 对象名称

productList

\* 唯一标识

productList

取消

确定

3. 在已创建的对象中，单击，进入对象详情页面。
4. 在“字段”页签，单击“添加”，为对象添加productName字段。

图 7-33 添加 productName 字段

添加字段

\* 显示名称

productName

\* 唯一标识

productName

\* 字段类型

文本

\* 数据长度

64

描述

取消

确认

表 7-7 添加 productName 字段参数说明

| 参数   | 说明                               | 示例          |
|------|----------------------------------|-------------|
| 显示名称 | 新建字段的名称，创建后可修改。<br>取值范围：1~63个字符。 | productName |

| 参数   | 说明                                                                                                                                                                                                                                                | 示例          |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 唯一标识 | 新建字段在系统中的标识，创建后不支持修改。命名要求如下： <ul style="list-style-type: none"><li>- 长度不能超过63个字符，包括前缀命名空间的长度。标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。</li><li>- 必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。</li></ul> | productName |
| 字段类型 | 单击***，在弹出的页面中，根据页面提供的参数解释，选择新建字段所属的类型。                                                                                                                                                                                                            | 文本          |

5. 在“字段”页签，再次单击“添加”按钮，添加productNumber字段。

图 7-34 添加 productNumber 字段

添加字段

\* 显示名称

productNumber

\* 唯一标识

productNumber

\* 字段类型

数字

\* 数据长度

10

\* 小数位数

0

描述

取消

确认

6. 在“数据”页签，单击“添加”，为对象添加数据。



图 7-35 为对象添加数据

字段索引验证清理数据

+ 添加

删除

导入

导出

设置出厂数据

输入名称

Q

👁

| <input type="checkbox"/> | 序列 | 名称 | productName | productNumber |
|--------------------------|----|----|-------------|---------------|
| <input type="checkbox"/> | 1  | c  | 洗涤用品        | 10            |
| <input type="checkbox"/> | 2  | b  | 食品          | 50            |
| <input type="checkbox"/> | 3  | a  | 家具          | 20            |

- 步骤6 创建读取对象数据的脚本。
1.

在应用设计器的左侧导航栏中，选择“逻辑”，单击脚本中的“+”。
2.

创建一个空白脚本，名称为getDataInfo。

图 7-36 创建脚本 getDataInfo

新增脚本

×

● 创建一个新脚本

○ 使用已有脚本

\* 名称

getDataInfo

模板

空脚本

▼

描述

取消

添加

3.

在脚本编辑器中，输入如下代码。
- ```
// Here's your code.
import * as db from 'db';
@useObject(['命名空间_productList_CST'])
@action.object({ type: 'method' })
export class SearchScript {
  @action.method({ input: 'ParamsInput', output: 'ParamsOutput' })
  public run(): Object[] {
    let queryData = this.doSearchScript();
    let result: Array<Object> = [];
    result.push({
      dataValue: [{
        name: "订单数",
        value: queryData
      }]
    });
    console.log("result", result)
    return result;
  }
  private doSearchScript(): Object[] {
```

```
let sql = "select 命名空间_productName__CST as name,命名空间_productNumber__CST as value"
+ " from 命名空间_productList__CST"
let query = db.sql().exec(sql)
return query;
}
}
```

4. 单击，保存脚本，保存成功后单击，激活脚本。

步骤7 新建开放接口。

- 1. 在应用设计器中，选择“集成”。
- 2. 单击开放接口后的“+”，设置开放接口参数。

图 7-37 设置开放接口参数

新建开放接口

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

请输入

* 操作名称

showDataInfo

* 版本

1.0.0

* URL

/service/_newmod/1.0.0/showDataInfo

内容类型

application/json

分类

请输入

描述

请输入

☐ 允许匿名访问

类型

☒ 脚本 ☐ 对象

☐ 自定义响应

* 资源

_getDataInfo

* 方法

POST

保存

取消

表 7-8 新建接口参数说明

| 参数 | 说明 | 示例 |
|------|--|--------------|
| 操作名称 | 新建接口的操作名称。
命名要求如下： <ul style="list-style-type: none">取值范围：1~40个字符。必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。 | showDataInfo |
| 版本 | URL对应的版本号。 | 1.0.0 |

| 参数 | 说明 | 示例 |
|-----|---|---------------|
| URL | 新URL地址，其中“/service”是固定值，其次是“/App名称/版本号”，剩下部分请自定义。 | /showDataInfo |
| 类型 | 选择新建接口的类型。 | 脚本 |
| 资源 | 选择接口调用的脚本。 | 选择步骤6中创建的脚本 |
| 方法 | 映射后调用的方法名。 | POST |

3. 设置完成后，单击“保存”，进入接口详情页。
4. 在接口信息中，单击URL后的，复制URL。

图 7-38 复制 URL



步骤8 返回高级页面，选中“玫瑰花饼图”，在“数据”页签，设置数据桥接器。

“桥接器实例”选择“饼图数据桥接器”，“数据类型”选择“动态数据”，“URL”设置为步骤7.4中获取的URL。

图 7-39 设置组件数据



步骤9 在玫瑰花饼图组件上，单击右键选择“高级设置”，设置组件标题，字体大小等。

图 7-40 设置组件标题



步骤10 单击, 保存高级页面, 保存成功后单击, 发布高级页面。

步骤11 发布成功后, 单击, 预览效果。

----结束

7.6 在华为云 Astro 轻应用高级页面中使用轮播组件实现图片展示和 URL 跳转

期望实现效果

高级页面中的轮播组件主要用于多个图片的自动循环切换。您也可以为图片添加超链接, 即单击图片, 跳转到指定的网站。

功能实现方法

- 步骤1 创建一个低代码应用。
- 1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
 - 2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
 - 3. 在“应用”中，单击“新建低代码应用”或单击，进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。
 - 4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。
 - 5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 7-41 创建一个空白应用

新建 空白应用

基本信息



添加图标

标签

我的第一个应用

名称

A

分类

请选择

描述

请输入

高级设置

取消

新建

表 7-9 新建空白应用参数说明

| 参数 | 说明 | 示例 |
|----|--|---------|
| 标签 | 新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。 | 我的第一个应用 |

文档版本 01 (2025-09-04)

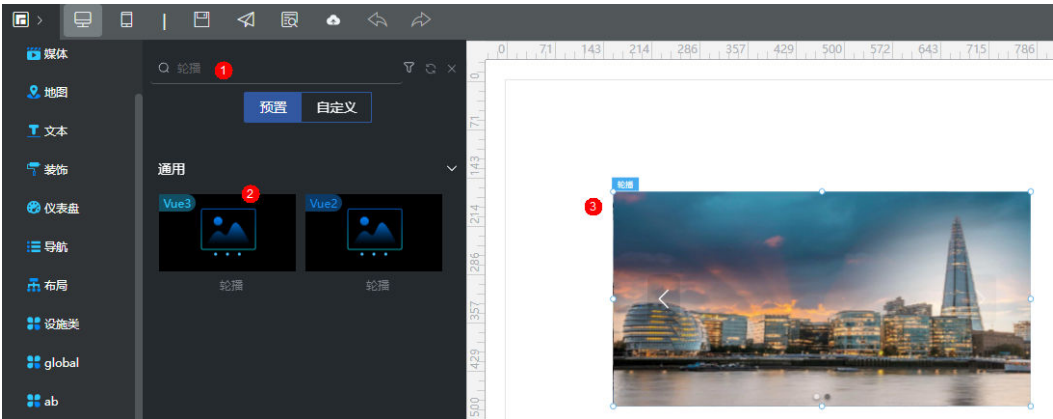
版权所有 © 华为云计算技术有限公司

270

| 参数 | 说明 | 示例 |
|----|--|----|
| 名称 | <p>新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下：</p> <ul style="list-style-type: none">长度不能超过31个字符，包括前缀命名空间的长度。名称前的内容为命名空间，在华为云Astro轻应用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。必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。 | A |

- 步骤2** 在应用设计器中，选择“界面”，单击高级页面后的“+”，新建一个高级页面。
- 步骤3** 单击，拖拽轮播组件到右侧画布中。

图 7-42 拖入轮播组件



- 步骤4** 在轮播组件上，单击右键选择“高级设置”。
- 步骤5** 单击“轮播设置”，根据个人实际需求，设置轮播方向、图片播放时间间隔等。

图 7-43 选择轮播设置



图 7-44 轮播设置



步骤6 新增轮播图片并设置URL跳转。

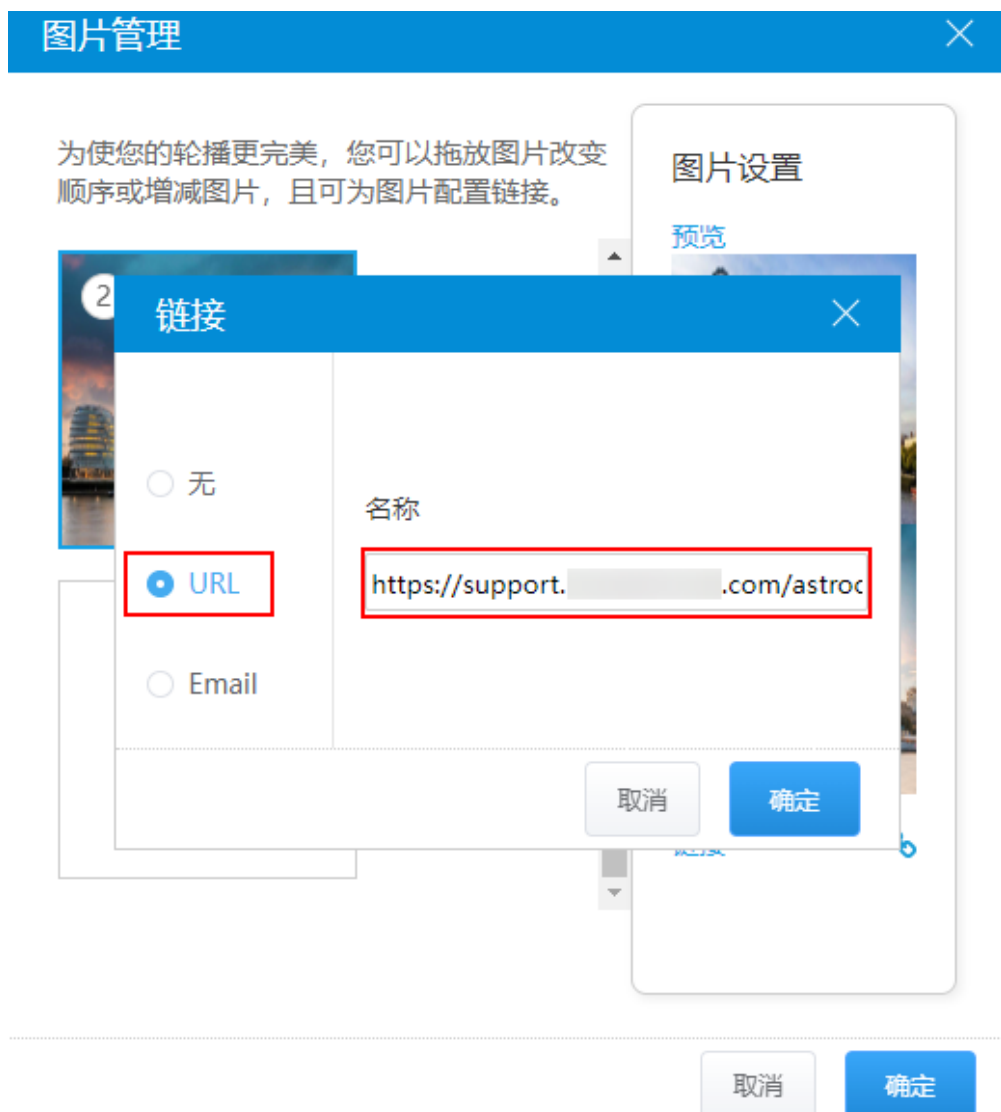
1. 再次选中轮播组件，单击右键选择“高级设置”，选择“图片管理”。

图 7-45 选择图片管理



2. 单击“添加”，选中所需的图片，单击“保存”。
3. 选中已添加的图片，在右侧图片设置中，单击链接后的 .
4. 设置跳转的URL地址，单击“确定”。

图 7-46 设置链接



步骤7 单击“确定”，返回高级页面。

步骤8 单击，保存高级页面，保存成功后单击，发布高级页面。

步骤9 发布成功后，单击，预览效果。

----结束

8 连接器专项

8.1 通过华为云 Astro 轻应用中的连接器上传并识别身份证图片

期望实现效果

华为云Astro轻应用封装了不同类型的连接器用于对接其他服务，对接后即可在应用中使用该服务。例如，通过连接器对接OCR后，可识别某用户上传到华为OBS上的身份证图片的全部信息。

图 8-1 在页面提交身份证信息

name

3

id

3

上传

Upload File

添加

图 8-2 成功识别图片

| | | | | |
|----------------------|---------------|------|---------------------|---|
| 名称 | 唯一标识 | 描述 | 创建人 | 修改时间 |
| orcry | B.M_orcry_CST | 暂无描述 | user_c00590408 | 2024-05-14 10:41:30 |
| 字段 | 索引 | 验证 | 清理 | 数据 |
| + 添加 删除 导入 导出 设置出厂数据 | | | | |
| | 序列 | 名称 | pictureName | pictureId |
| | 1 | | 3 | 3 |
| | | | picUrl | result |
| | | | https://appcubec... | ["birth":"1990-01-24","address":"湖北省承德市","number":"","name":"","sex":"女","ethnicity":"汉"] |

功能实现方法

步骤1 准备工作。

- 需要拥有一个华为账号或一个可用于访问OBS的IAM用户，即先注册华为云并实名认证、创建IAM用户、充值以及购买资源包，具体操作请参见[使用OBS前需要做的准备工作](#)。
- 获取AK（Access Key ID）、SK（Secret Access Key），即访问密钥对，具体操作请参见[获取AK/SK](#)。
- 已在华为OBS上，创建存储桶（例如“appcubecn4”），用于后续存储对象使用，具体操作请参见[如何创建桶](#)，请记录创建桶时选择的区域。

步骤2 创建一个低代码应用。

1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
3. 在“应用”中，单击“新建低代码应用”或单击，进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。
4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。
5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 8-3 创建一个空白应用

新建 空白应用

基本信息



* 标签

我的第一个应用

* 名称

A

分类

请选择

描述

请输入

高级设置

>

取消

新建

表 8-1 新建空白应用参数说明

| 参数 | 说明 | 示例 |
|----|--|---------|
| 标签 | 新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。 | 我的第一个应用 |

| 参数 | 说明 | 示例 |
|----|--|----|
| 名称 | <p>新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下：</p> <ul style="list-style-type: none">长度不能超过31个字符，包括前缀命名空间的长度。名称前的内容为命名空间，在华为云Astro轻应用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。必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。 | A |

步骤3 创建一个OBS连接器。

1. 在应用设计器中，选择“集成”，单击“连接器”下的“连接器实例”。
2. 在存储中，选择“OBS”，单击“+”，进入创建OBS页面。
3. 输入基本信息并添加桶，单击“保存”。

图 8-4 设置基本信息

创建OBS

基本信息

* 名称

uploadMod

* 鉴权方式

AK/SK

* 访问密钥

* 密钥

内容分发网络

请输入

描述

请输入

打包当前配置

☒ 打包当前配置

开启操作日志

☐ 开启操作日志

校验文件内容类型

☐ 校验文件内容类型

使用安全连接

☒ 使用安全连接

图 8-5 添加桶



表 8-2 新建 OBS 连接器参数说明

| 参数 | 说明 | 示例 |
|------|--|---------------------------|
| 名称 | 新建OBS连接器的名称。命名要求如下： <ul style="list-style-type: none">长度不能超过64个字符，包括前缀命名空间的长度。标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。 | uploadMod |
| 鉴权方式 | 根据实际情况选择鉴权方式： <ul style="list-style-type: none">选择AK/SK方式：需先获取访问密钥，并手动输入AK/SK。选择委托授权方式：需统一授权，无需再填写访问密钥。 | AK/SK |
| 访问密钥 | 配置用户的访问密钥AK。 | 步骤1中获取的Access Key ID值 |
| 密钥 | 配置与访问密钥AK结合使用的私有访问密钥SK。 | 步骤1中获取的Secret Access Key值 |

| 参数 | 说明 | 示例 |
|----|-----------------|---|
| 桶 | 设置存储桶所在的区域和桶名称。 | 区域选择“华北-北京四”，桶名称为“appcubecn4”，即步骤1中提前创建的桶 |

步骤4 创建对象orcTry，用于存放图片URL等信息。

1.

在应用设计器的左侧导航栏中，选择“数据”，单击对象中的“+”。
2.

设置对象的名称和唯一标识为“orcTry”，单击“确定”。

图 8-6 创建对象 orcTry



表 8-3 新建 orcTry 对象参数说明

| 参数 | 说明 | 示例 |
|------|---|--------|
| 对象名称 | 新建对象的名称，创建后可修改。
取值范围：1~80个字符。 | orcTry |
| 唯一标识 | 新建对象在系统中的标识，创建后不支持修改。命名要求如下： <ul style="list-style-type: none">长度不能超过63个字符，包括前缀命名空间的长度。必须以英文字母开头，只能由英文字母，数字和下划线组成，且不能以下划线结尾。 | orcTry |

3.

在已创建的对象中，单击，进入对象详情页面。
4.

在“字段”页签，单击“添加”，为对象添加pictureName字段。

图 8-7 添加 pictureName 字段

添加字段

*

显示名称

pictureName

*

唯一标识

pictureName

*

字段类型

文本

*

数据长度

64

描述

取消

确认

表 8-4 添加字段参数说明

| 参数 | 说明 | 示例 |
|------|--|-------------|
| 显示名称 | 新建字段的名称，创建后可修改。
取值范围：1~63个字符。 | pictureName |
| 唯一标识 | 新建字段在系统中的标识，创建后不支持修改。命名要求如下： <ul style="list-style-type: none">长度不能超过63个字符，包括前缀命名空间的长度。必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。 | pictureName |
| 字段类型 | 单击***，在弹出的页面中，根据页面提供的参数解释，选择新建字段所属的类型。 | 文本 |

5. 按照上述操作，为对象添加表8-5中字段。

表 8-5 添加其他字段

| 显示名称 | 唯一标识 | 字段类型 |
|-----------|-----------|------|
| pictureId | pictureId | 文本 |

| 显示名称 | 唯一标识 | 字段类型 |
|--------|--------|------|
| picUrl | picUrl | 文本区 |
| result | result | 文本区 |

步骤5 创建OCR连接器。

1. 在应用设计器中，选择“集成”，单击“连接器”下的“连接器实例”。
2. 在AI中，选择“OCR”，单击“+”，进入创建OCR页面。
3. 输入基本信息，单击“保存”。

图 8-8 设置基本信息

创建OCR

基本信息

*

名称

newOcr

*

区域

华北-北京四

*

访问密钥ID

.....

*

访问密钥

.....

打包当前配置

☑

打包当前配置

保存

取消

表 8-6 新建 OCR 连接器参数说明

| 参数 | 说明 | 示例 |
|----|---|--------|
| 名称 | 新建OCR连接器的名称。命名要求如下： <ul style="list-style-type: none">长度不能超过63个字符，包括前缀命名空间的长度。必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。 | newOcr |

| 参数 | 说明 | 示例 |
|--------|--|--|
| 区域 | OCR服务的区域。 | 华北-北京四 |
| 访问密钥ID | 访问密钥ID和访问密钥一起使用，对请求进行加密签名。 | 参考 步骤1 中操作获取，配置为“AK”的值 |
| 访问密钥 | 访问密钥ID结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。 | 参考 步骤1 中操作获取，配置为“SK”的值 |

步骤6 创建脚本，用于提交图片信息以及OCR接口识别图片内容。

1. 在应用设计器中，选择“逻辑”，单击脚本后的“+”。
2. 新建一个空白脚本，脚本名称为octTryTs。

图 8-9 新建 octTryTs 脚本

3. 在脚本编辑器中，输入如下示例代码。

```
//本脚本用于提交图片信息以及调用ocr接口识别图片内容
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 ocr from 'ocr';

//定义入参结构
@action.object({ type: "param" })
export class ActionInput {
  @action.param({ type: 'String', required: true, label: 'String' })
  proId: string;
  @action.param({ type: 'String', required: true, label: 'String' })
  proName: string;
  @action.param({ type: 'Any', required: true, label: 'Any' })
  url: any;
```

```

}
//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;
  @action.param({ type: 'String', required: true, label: 'String' })
  url: string;
}
//使用数据对象命名空间_orcTry_CST
@useObject(['命名空间_orcTry_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 cli = ocr.newClient("命名空间_newOcr"); //创建连接器实例
    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 url = "https://appcubecn4.obs.cn-north-4.myhuaweicloud.com/" + input.url[0]
      ['originalUrl']; // 拼接成完整的图片url信息
      let resp = cli.idCardWithURL(url, "front"); // 调用ocr接口，front表示识别身份证正面
      let productData = new Object();
      productData['命名空间_pictureName__CST'] = input.proName; //将入参赋值给productData变量，方便后面使用
      productData['命名空间_pictureId__CST'] = input.proId;
      productData['命名空间_picUrl__CST'] = url;
      productData['命名空间_result__CST'] = JSON.stringify(resp['result']);
      let s = db.object('BJ4_orcTry_CST'); //获取命名空间_orcTry_CST这个Object的操作实例
      let id = s.insert(productData);
      if (id) {
        out.id = id;
        out.url = url;
      } else {
        error.name = "WOERROR";
        error.message = "Unable to create pic!";
        throw error;
      }
    } catch (error) {
      console.error(error.name, error.message);
      context.setError(error.name, error.message);
    }
    return out;
  }
}
}

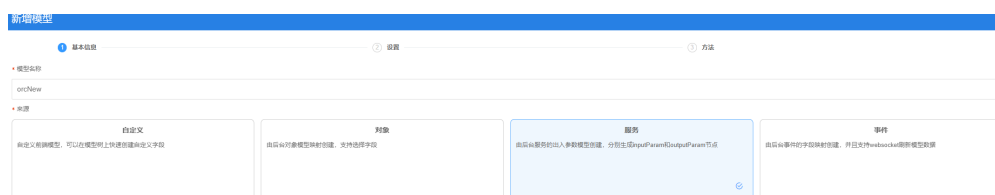
```

4. 单击，保存脚本，保存成功后单击，激活脚本。

步骤7 新建页面模型。

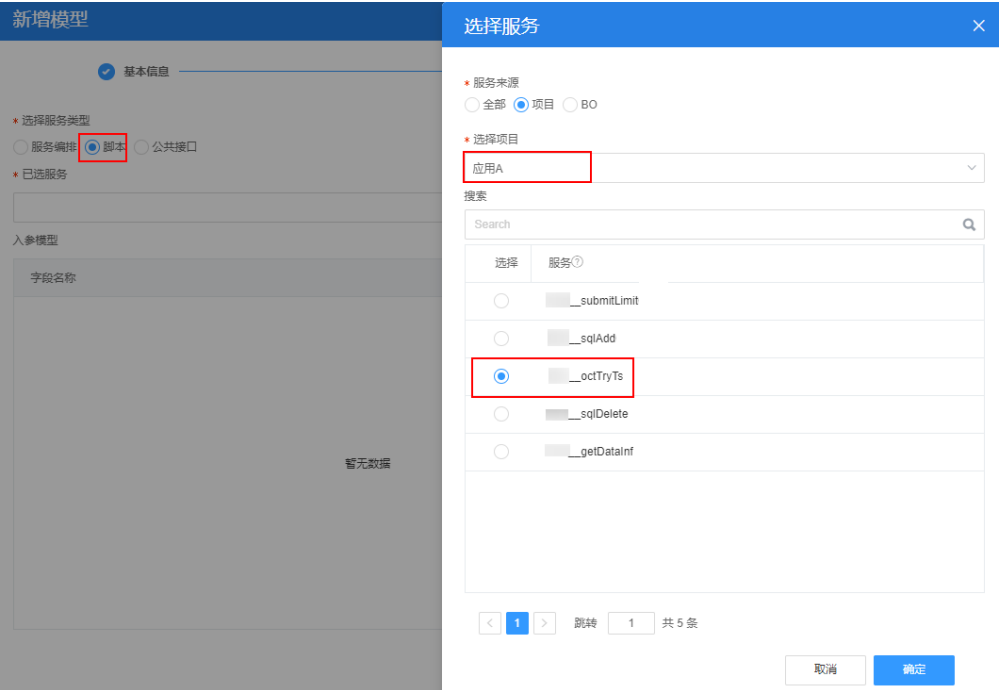
1. 在应用设计器中，选择“界面”，单击页面后的“+”，新建一个标准页面。
2. 在标准页面底部，单击“模型视图”。
3. 单击“新增模型”，输入模型名称（如orcNew）、“来源”选择“服务”，单击“下一步”。

图 8-10 新建模型



4. 选择步骤6中创建的脚本，单击“确定”后，再单击“下一步”。

图 8-11 选择脚本

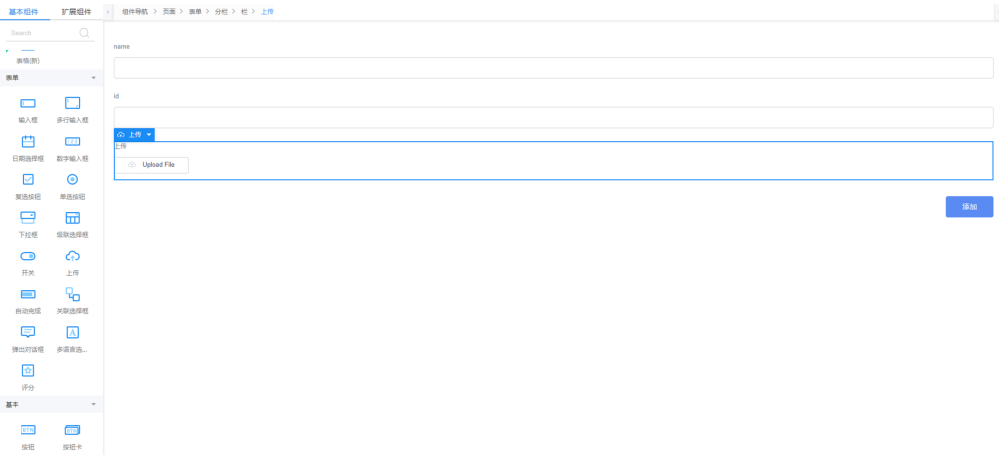


5. 直接单击“确定”，完成模型的创建。

步骤8 创建图片上传表单页面。

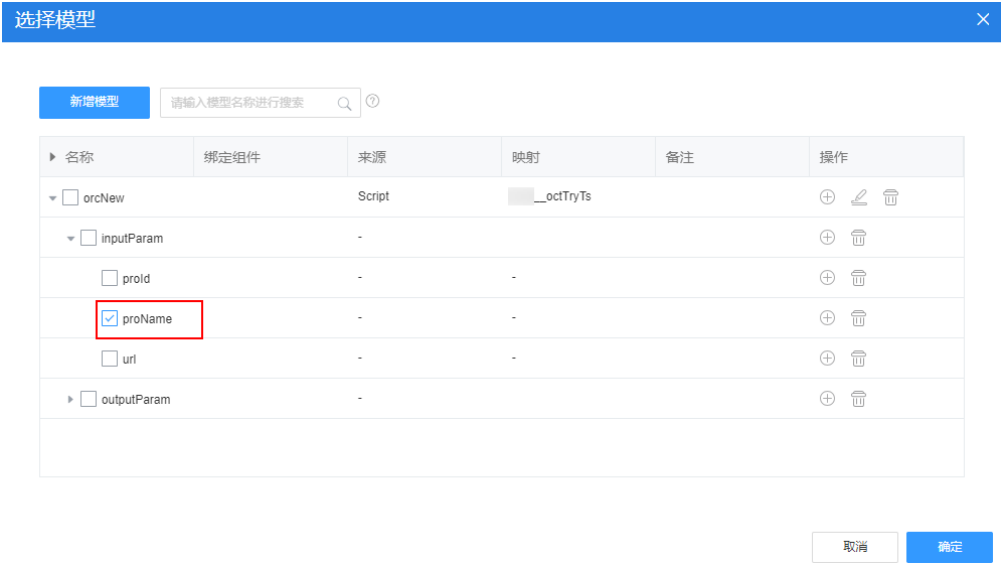
- 1. 在标准页面中，拖入2个输入框组件、1个按钮和1个上传组件。
- 2. 将输入框标签分别修改为“name”和“id”，将按钮显示名称设置为“添加”。

图 8-12 表单最终设置效果



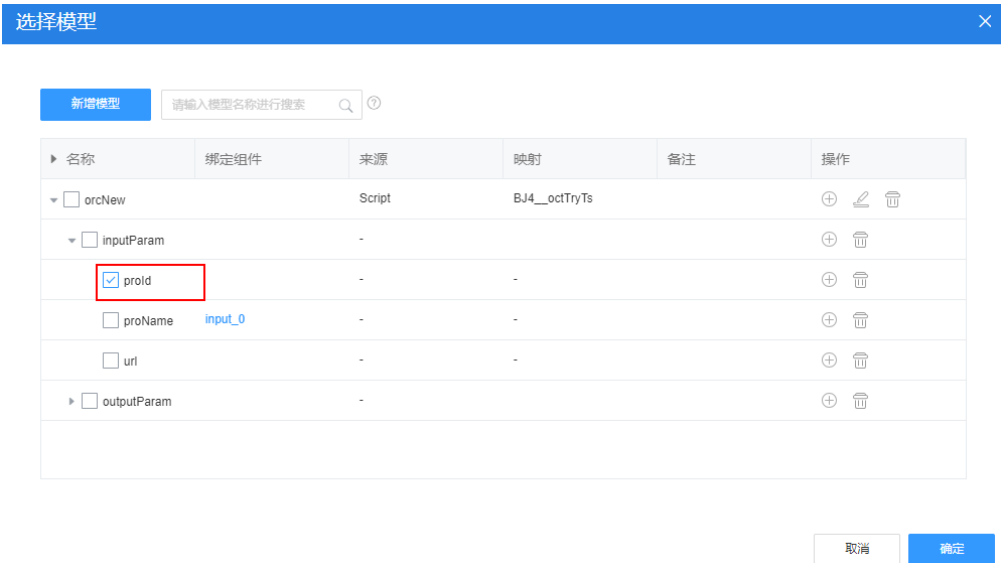
- 3. 选中name输入框，在“属性 > 数据绑定 > 值绑定”中，单击 。
- 4. 选择步骤7中创建模型中的proName字段，单击“确定”。

图 8-13 选择 proName 字段



5. 选中id输入框，在“属性 > 数据绑定 > 值绑定”中，单击 。
6. 选择步骤7中创建模型中的proId字段，单击“确定”。

图 8-14 选择 proId



7. 同样，按照上述操作，为上传组件绑定模型，并设置存储信息。

图 8-15 为上传组件绑定模型

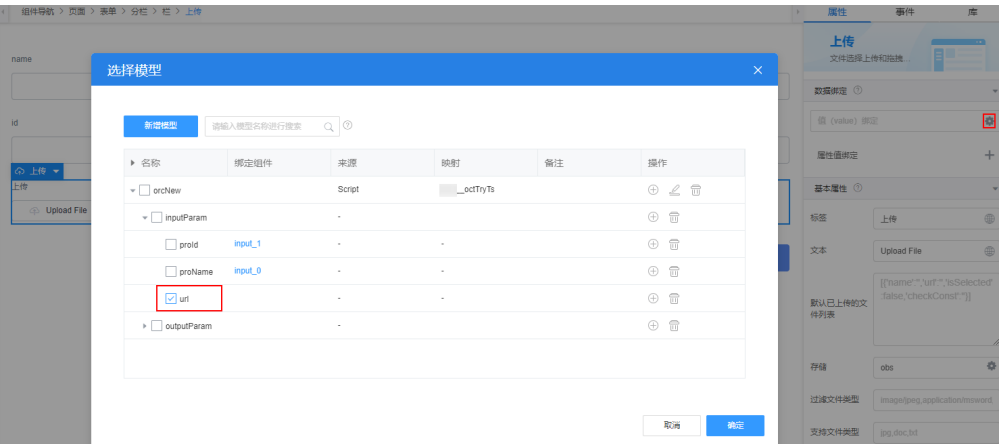
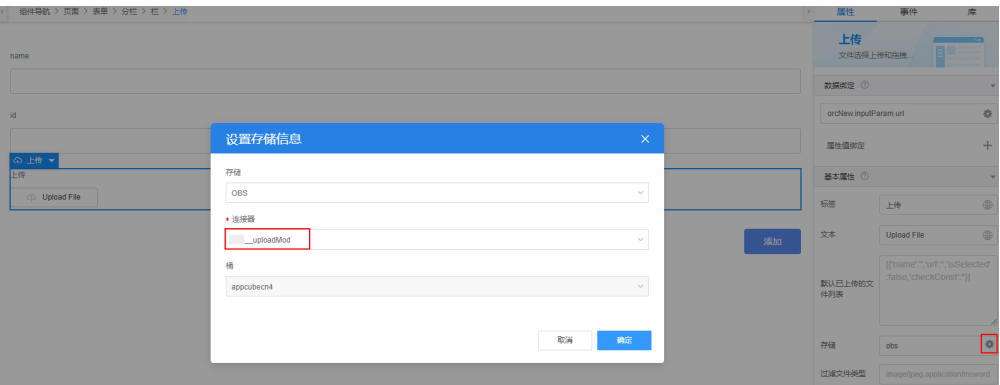
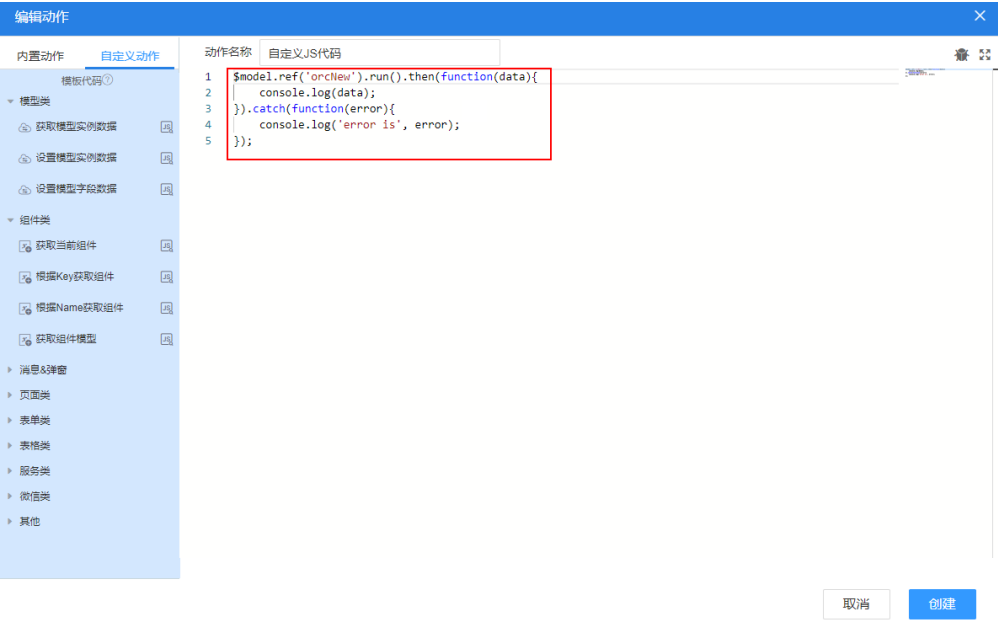


图 8-16 设置存储信息



- 步骤9** 为添加按钮，添加事件。
1. 选中添加按钮组件，选择“事件”页签。
 2. 单击“点击”后的“+”，进入添加动作页面。
 3. 在自定义动作中，输入自定义代码，单击“保存”。

图 8-17 自定义动作



其中，“orcNew”为步骤7中创建模型的名称。

```
$model.ref('orcNew').run().then(function(data){
  console.log(data);
}).catch(function(error){
  console.log('error is', error);
});
```

步骤10 返回标准页面，单击，保存页面，保存成功后单击，预览效果。

----结束

8.2 通过华为云 Astro 轻应用中的连接器实现文件上传功能

期望实现效果

华为云Astro轻应用封装了不同类型的连接器用于对接其他服务，对接后即可在应用中使用该服务。例如，通过连接器对接OBS后，可将前端页面中上传的文件存储到OBS桶中。

图 8-18 在 OBS 桶的指定路径下可查看到文件



功能实现方法

步骤1 准备工作。

- 需要拥有一个华为账号或一个可用于访问OBS的IAM用户，即先注册华为云并实名认证、创建IAM用户、充值以及购买资源包，具体操作请参见[使用OBS前需要做的准备工作](#)。
- 获取AK（Access Key ID）、SK（Secret Access Key），即访问密钥对，具体操作请参见[获取AK/SK](#)。
- 已在华为OBS上，创建存储桶（例如“bing.testonly.1”），用于后续存储对象使用，具体操作请参见[如何创建桶](#)，请记录创建桶时选择的区域。

步骤2 创建一个低代码应用。

1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
3. 在“应用”中，单击“新建低代码应用”或单击，进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。
4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。
5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 8-19 创建一个空白应用

新建 空白应用

基本信息



添加图标

* 标签

我的第一个应用

* 名称

A

分类

请选择

描述

请输入

高级设置

>

取消

新建

表 8-7 新建空白应用参数说明

| 参数 | 说明 | 示例 |
|----|--|---------|
| 标签 | 新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。 | 我的第一个应用 |

文档版本 01 (2025-09-04)

版权所有 © 华为云计算技术有限公司

289

| 参数 | 说明 | 示例 |
|----|--|----|
| 名称 | <p>新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下：</p> <ul style="list-style-type: none">长度不能超过31个字符，包括前缀命名空间的长度。名称前的内容为命名空间，在华为云Astro轻应用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。必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。 | A |

步骤3 创建一个OBS连接器。

1. 在应用设计器中，选择“集成”，单击“连接器”下的“连接器实例”。
2. 在存储中，选择“OBS”，单击“+”，进入创建OBS页面。
3. 输入基本信息并添加桶，单击“保存”。

图 8-20 设置桶基本信息

创建OBS

基本信息

* 名称

upload

* 访问密钥

* 密钥

内容分发网络

请输入

描述

请输入

打包当前配置

☒ 打包当前配置

开启操作日志

☐ 开启操作日志

校验文件内容类型

☐ 校验文件内容类型

使用安全连接

☒ 使用安全连接

图 8-21 添加桶

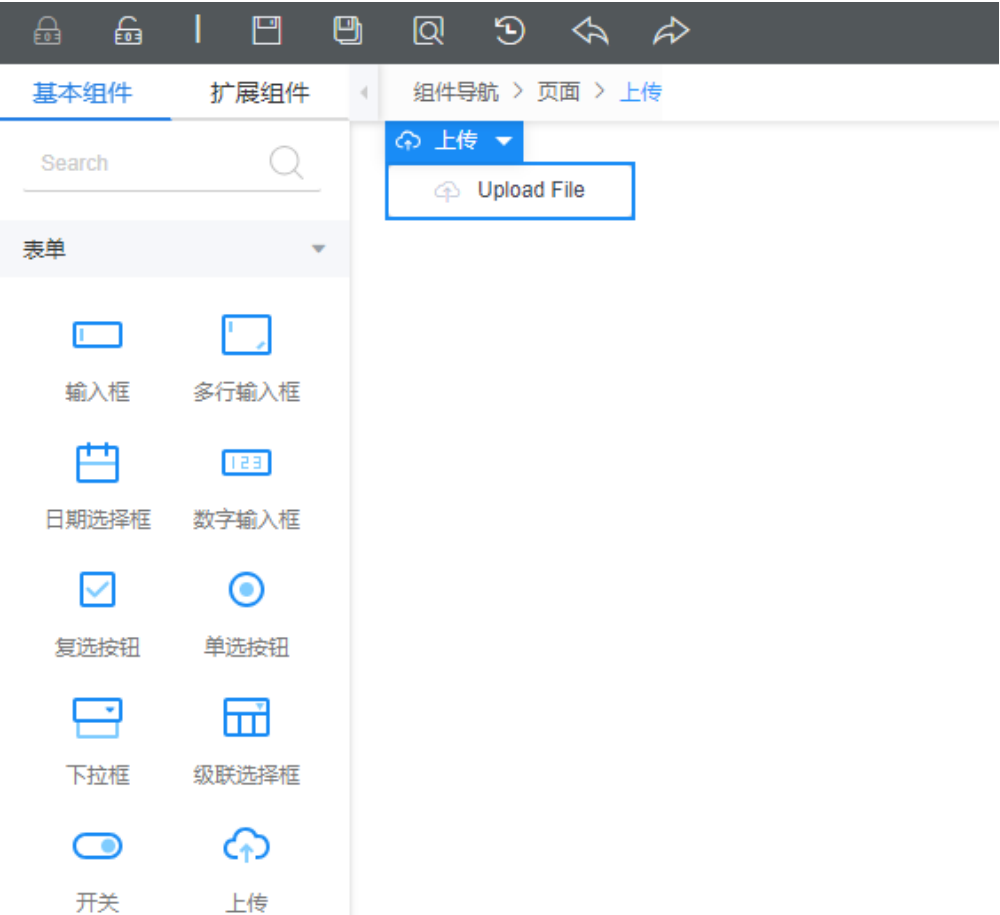


表 8-8 新建 OBS 连接器参数说明

| 参数 | 说明 | 示例 |
|------|---|--|
| 名称 | 新建OBS连接器的名称。命名要求如下： <ul style="list-style-type: none">长度不能超过64个字符，包括前缀命名空间的长度。标识前模糊掉的内容为命名空间，在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。必须以英文字母开头，只能由英文字母，数字和单下划线组成，且不能以下划线结尾。 | upload |
| 访问密钥 | 配置用户的访问密钥AK。 | 步骤1 中获取的Access Key ID值 |
| 密钥 | 配置与访问密钥AK结合使用的私有访问密钥SK。 | 步骤1 中获取的Secret Access Key值 |
| 桶 | 设置存储桶所在的区域和桶名称。 | 区域选择“华北-北京四”，桶名称为“bing.testonly.1”，即 步骤1 中提前创建的桶 |

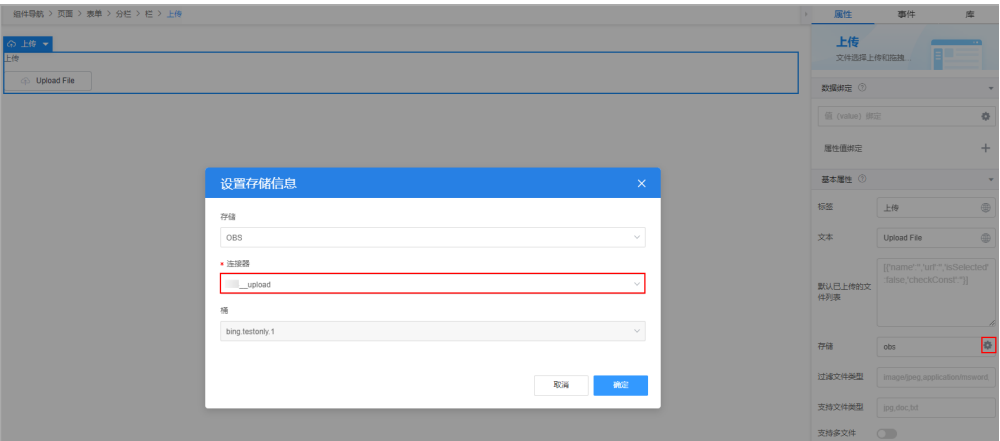
- 步骤4** 创建标准页面，用于上传文件。
- 在应用设计器中，选择“界面”，单击页面后的“+”，新建一个标准页面。
 - 在标准页面中，拖入一个“上传”组件。

图 8-22 拖入上传组件



3. 选中上传组件，存储选择“OBS”，并选择步骤3中创建好的连接器。

图 8-23 设置存储信息



4. 定义文件的上传路径。

图 8-24 设置上传路径

The screenshot shows the '数据绑定' (Data Binding) configuration page. Under the '基本属性' (Basic Properties) section, the '上传路径' (Upload Path) field is highlighted with a red rectangle and contains the text 'upLoadTest/upload'. Other visible fields include '值 (value) 绑定', '属性值绑定', '标签' (set to '上传'), '文本' (set to 'Upload File'), '默认已上传的文件列表' (containing a JSON array), '存储' (set to 'obs'), '过滤文件类型' (set to 'image/jpeg,application/msword'), '支持文件类型' (set to 'jpg,doc,txt'), '支持多文件' (checked), '同名处理' (unchecked), and '文件数量'.

步骤5 返回标准页面，单击，保存页面，保存成功后单击，预览效果。

----结束

8.3 通过华为云 Astro 轻应用连接器对接第三方数据库

应用场景

连接器是华为云Astro轻应用提供的，调用第三方服务的集成工具。通过使用华为云Astro轻应用连接器，开发者无需关注具体的代码实现，只需要配置集成服务的地址和鉴权信息，即可快速将第三方服务集成到应用中，大大提高了开发效率，降低了开发成本。

本实践将为您介绍如何通过自定义连接器将第三方数据库集成到华为云Astro轻应用中。

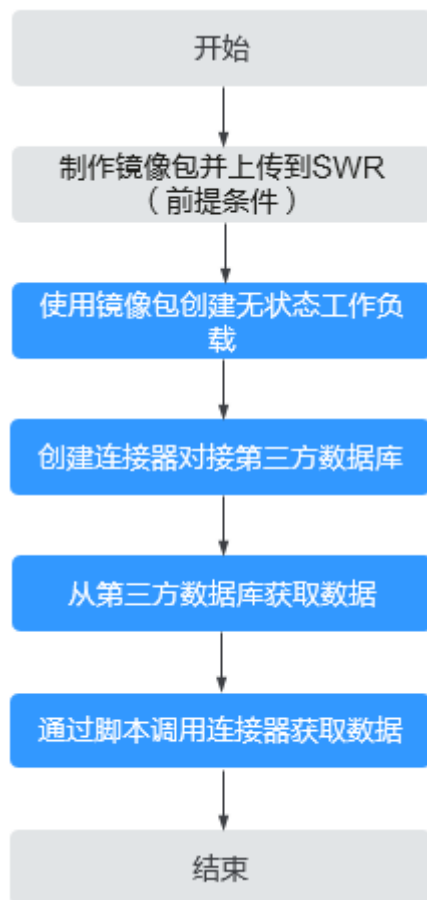
方案优势

用户可以自定义数据库的操作和鉴权接口，独立控制第三方数据库的网络。

操作流程

在华为云Astro轻应用中，通过连接器对接第三方数据库的流程，如图8-25所示。

图 8-25 通过连接器对接第三方数据库流程



前提条件

- **db-adapter**是用于操作第三方数据库的后端java demo代码。resources目录下的datasource.json文件用于配置第三方数据库的地址链接以及用户名、密码，用户可以基于自己的第三方数据库进行配置。DBExecuteController.java用于连接上数据库后，处理外部请求，返回数据库数据。

图 8-26 datasource.json

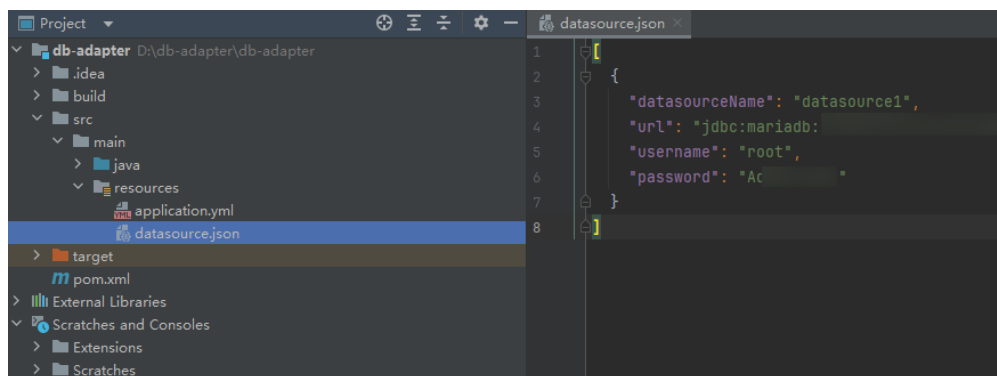
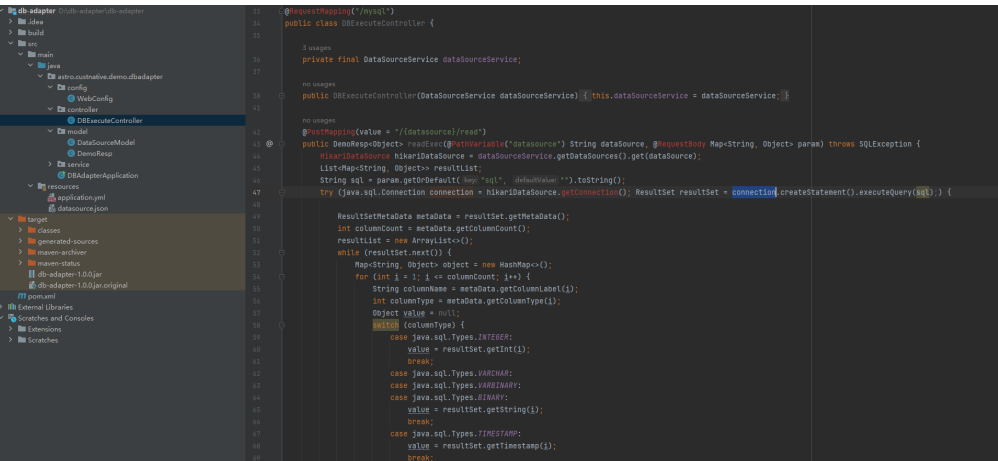


图 8-27 DBExecuteController.java



- 基于db-adapter生成jar包，把编出的jar包放在“build”目录下，构建docker镜像。已将制作好的镜像包上传到容器镜像服务SWR，如何上传镜像包请参见[页面上传镜像](#)。

使用docker save制作镜像压缩包时，请使用“{image}:{tag}”命令，不要使用“image id”，否则无法在容器镜像服务SWR页面上上传。镜像上传失败时，可参考[为什么通过页面上传镜像失败？](#)中操作处理。

图 8-28 使用 maven install，将 db-adapter 打成 jar 包

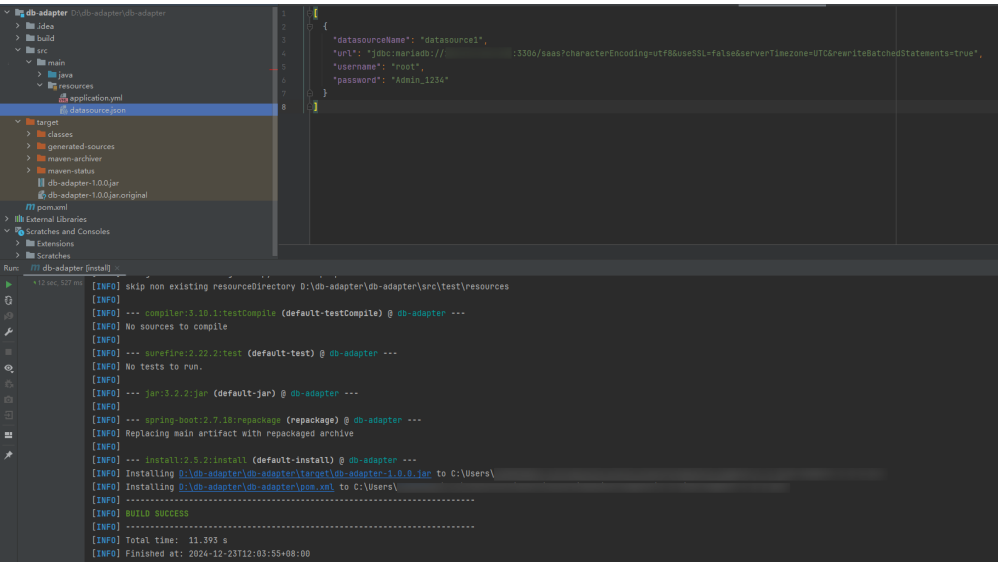
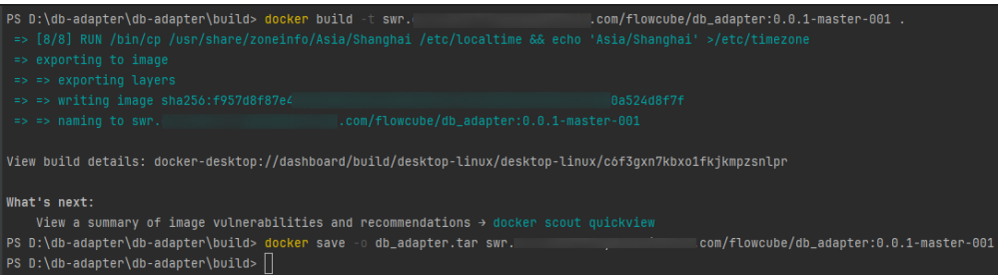


图 8-29 构建镜像



步骤一：使用镜像包创建无状态工作负载

- 步骤1 登录CCE控制台，参考[购买Standard/Turbo集群](#)中操作，购买所需的集群。
- 步骤2 集群购买成功后，参考[创建节点](#)中操作，为集群添加节点。
- 步骤3 创建一个无状态工作负载。
 - 节点添加后，在左侧导航栏中选择“Kubernetes 资源 > 工作负载 > 无状态工作负载”。
 - 单击“创建工作负载”，创建一个无状态工作负载。

图 8-30 创建无状态工作负载

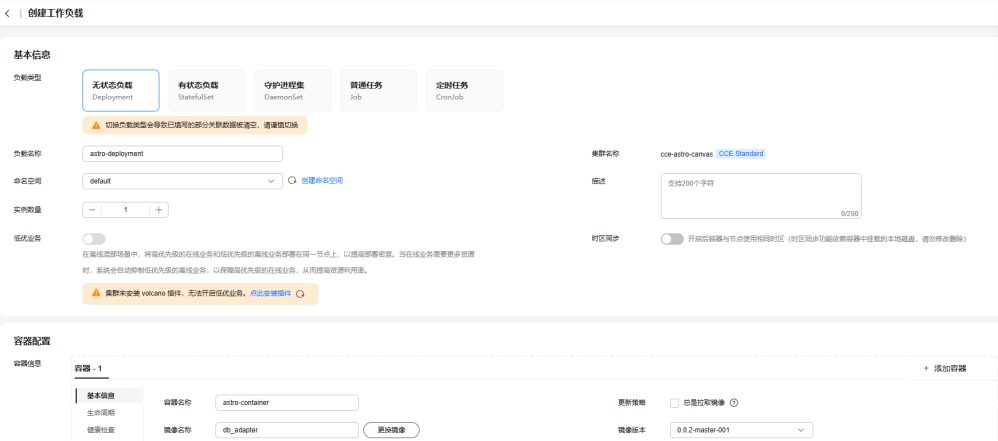


图 8-31 选择镜像



表8-9仅介绍本实践中使用到的参数，其他参数的解释请参见[创建无状态负载（Deployment）](#)。

表 8-9 无状态工作负载参数说明

| 参数 | 说明 | 示例 |
|------|--|------------------|
| 负载类型 | 选择创建工作负载的类型。 | 无状态负载 |
| 负载名称 | 新建工作负载的名称。命名要求如下： <ul style="list-style-type: none">长度不能超过63字符。可以包含小写英文字母、数字和中划线（-），并以小写英文字母开头，小写英文字母或数字结尾。 | astro-deployment |
| 实例数量 | 填写实例的数量，即工作负载Pod的数量。 | 1 |
| 容器名称 | 输入容器的名称。 | astro-container |
| 镜像名称 | 单击后方的“选择镜像”，选择容器使用的镜像，即前提条件中上传的镜像。 | db_adapter |

步骤4 为工作负载添加服务。

1. 在“无状态负载”页签，单击已创建工作负载的名称，进入负载详情页。

图 8-32 单击负载名称



2. 在“访问方式”页签中，单击“创建服务”，创建一个服务配置访问接口。

图 8-33 创建服务

创建服务

YAML创建

×

Service名称

astro-deployment-service

访问类型

集群内访问

通过集群的内部IP暴露服务，只能够在集群内部访问

节点访问

通过每个节点上的IP和静态端口（NodePort）暴露服务

负载均衡

通过ELB负载均衡对外部提供服务，高可用，超高性能，稳定安全

DNAT网关

通过NAT网关暴露集群节点访问类型服务，支持多个节点共享使用弹性IP

集群外访问推荐选择负载均衡访问类型

服务亲和

集群级别

节点级别

?

端口配置

协议

容器端口

服务端口

节点端口

操作

TCP

8086

8086

指定端口

30280

删除

+

取消

确定

表 8-10 服务参数说明

| 参数 | 说明 | 示例 |
|-------------|--|--------------------------|
| Service名称 | 输入新建服务的名称，也可以保持默认。 | astro-deployment-service |
| 访问类型 | 选择服务的访问类型。 | 节点访问 |
| 端口配置 > 协议 | 请根据业务的协议类型选择。 | TCP |
| 端口配置 > 容器端口 | 业务容器进程监听的端口，端口范围为1-65535。 | 8086 |
| 端口配置 > 服务端口 | 通过Service访问的端口，端口范围为1-65535。 | 8086 |
| 端口配置 > 节点端口 | 即NodePort，可以选择“自动生成”，也可以指定端口，默认范围为30000-32767。 | 30280 |

步骤5 测试接口的连通性。

将步骤2中添加节点的IP地址和步骤4.2中配置的节点端口拼接成完整的访问地址，访问地址由CCE节点IP端口部分、db-adapter中的context-path以及rest接口拼接而成，如http://10.10.10.1:30280/astro/native/demo/dbadapter/mysql/datasource1/read。

图 8-34 context-path

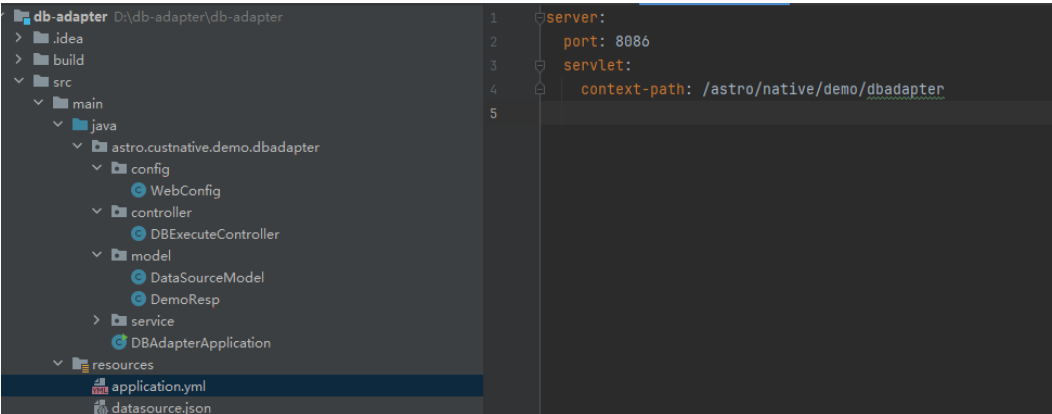
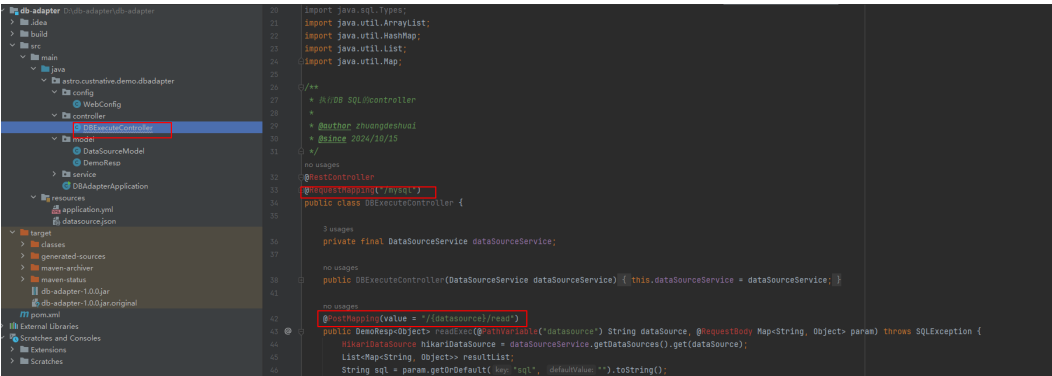
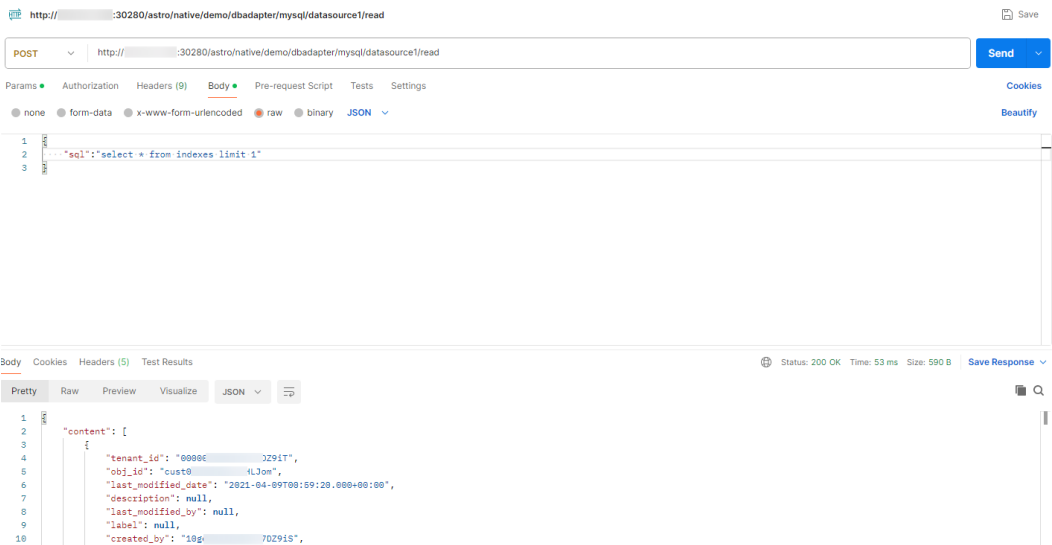


图 8-35 context-path 及 rest 接口



在Body页签中输入具体的SQL语句，单击“Send”，查看接口状态。状态为“200 OK”，表明接口调测通过。

图 8-36 测试接口的连通性



----结束

步骤二：创建连接器对接第三方数据库

- 步骤1 创建一个低代码应用。
- 1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
 - 2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
 - 3. 在“应用”中，单击“新建低代码应用”或单击，进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。
 - 4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。
 - 5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 8-37 创建一个空白应用

新建 空白应用

基本信息



添加图标

标签

我的第一个应用

名称

A

分类

请选择

描述

请输入

高级设置

取消

新建

表 8-11 新建空白应用参数说明

| 参数 | 说明 | 示例 |
|----|--|---------|
| 标签 | 新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。 | 我的第一个应用 |

文档版本 01 (2025-09-04)

版权所有 © 华为云计算技术有限公司

300

| 参数 | 说明 | 示例 |
|----|---|----|
| 名称 | <div>新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下：</div> <div><div>- 长度不能超过31个字符，包括前缀命名空间的长度。</div><div>名称前的内容为命名空间，在华为云Astro轻应用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。</div><div>- 必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。</div></div> | A |

- 步骤2 在应用设计器中，选择“集成”，单击“连接器”下的“连接器实例”。
- 步骤3 在类型中，选择“自定义连接器”，进入自定义连接器页面。
- 步骤4 单击“+”，配置连接器信息，单击“保存”。

图 8-38 创建自定义连接器

创建自定义连接器

* 标签

demoDB

* 名称

demoDB

图标



描述

请输入

保存

取消

表 8-12 自定义连接器参数说明

| 参数 | 说明 | 示例 |
|----|------------------------------------|--------|
| 标签 | 自定义连接器的标签，创建后可修改。
取值范围：1~64个字符。 | demoDB |

| 参数 | 说明 | 示例 |
|----|---|--------|
| 名称 | <p>自定义连接器的名称，名称是连接器在系统中的唯一标识，创建后不可修改。命名要求如下：</p> <ul style="list-style-type: none">长度不能超过64个字符，包括前缀命名空间的长度。标识前模糊掉的内容为命名空间，在华为云 Astro轻应用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。必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不能以下划线结尾。 | demoDB |

- 步骤5** 为自定义连接器新建认证信息。
- 在“认证信息”页签，单击“新建”，进入添加认证信息页面。
 - 设置认证信息，单击“保存”。

图 8-39 添加认证信息

添加认证信息

* 标签

noauth

* 名称

noauth6 / 59

目录

请选择

描述

☐ 使用证书

☐ 使用根证书

API主机地址

☒ 打包当前配置

可选设置

☐ 鉴权设置

* 鉴权协议OAuth

* 鉴权模式请选择

* 鉴权地址请输入

* Client ID键 请输入 值 请输入

* Client Secret键 请输入 值 请输入

消息体属性

| 键 | 属性 | 操作 |
|--------------|--------------|----|
| access_token | access_token | |
| token_type | token_type | |

取消

保存

表 8-13 添加认证信息参数说明

| 参数 | 说明 | 示例 |
|----|----------------------------------|--------|
| 标签 | 认证的标签名，用于在页面显示。
取值范围：1~64个字符。 | noauth |

| 参数 | 说明 | 示例 |
|----|--|--------|
| 名称 | 认证的名称，名称是认证在系统中的唯一标识，创建后不可修改。命名要求如下： <ul style="list-style-type: none">长度不能超过64个字符，包括前缀命名空间的长度。标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不能以下划线结尾。 | noauth |

- 步骤6** 为自定义连接器添加动作。
- 在“动作”页签中，单击“新建”。
 - 设置动作的基本信息，单击“下一步”。

图 8-40 设置基本信息

1
基本信息

2
消息头入参

3
输入参数

4
输出参数

* 标签

demoSql

* 名称

demoSql

* URL

http://30280/astro/native/demo/dbac

* 方法

POST

内容类型

application/json

返回类型

application/json

超时时间

系统默认值（20秒）

日期 格式

请选择

日期/时间 格式

请选择

禁用请求头规范化

禁用 HTML 字符转义

描述

请输入

表 8-14 动作基本信息参数说明

| 参数 | 说明 | 示例 |
|----|------------------------------------|---------|
| 标签 | 新建动作的标签名，用于在页面显示。
取值范围：1~64个字符。 | demoSql |

| 参数 | 说明 | 示例 |
|------|--|--|
| 名称 | <p>新建动作的名称，名称是动作在系统中的唯一标识。命名要求如下：</p> <ul style="list-style-type: none"> - 长度不能超过64个字符，包括前缀命名空间的长度。 <p>标识前模糊掉的内容为命名空间，在华为云Astro轻应用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。</p> <ul style="list-style-type: none"> - 必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不能以下划线结尾。 | demoSql |
| URL | 第三方提供的协议服务的URL地址，即 步骤5 中调试成功的地址。 | http://10.10.10.1:30280/astro/native/demo/dbadapter/mysql/datasource1/read |
| 方法 | 选择请求的方法。 | POST |
| 内容类型 | 输入内容的类型，该值取决于第三方支持哪种类型，是在HTTP标准协议中定义的。 | application/json |
| 返回类型 | <p>选择返回类型。</p> <p>取值为“application/json”时，表示会忽略第三方实际返回的“content-type”，使用第三方配置的“content-type”来解析第三方返回的内容。</p> <p>如果该参数取值为空，表示使用HTTP返回中的“content-type”，来解析第三方返回的内容。</p> | application/json |

步骤7 本示例无需设置消息头入参，请单击  删除第一行参数，再单击“下一步”。

步骤8 设置输入参数，单击“下一步”。

图 8-41 输入参数配置页面

新建动作

1

2

3

4

基本信息

消息头入参

输入参数

输出参数

消息体格式: 键-值

| * 标签 | * 名称 | * 参数位置 | * 数据类型 | 命名空间 | 元数据名称 | 默认值 | 描述 | 默认 | 必填 | 集合 | 操作 |
|------|------|--------|--------|------|-------|-----|-----|--------------------------|--------------------------|--------------------------|---------------------------|
| sql | sql | 消息体 | 字符串 | 请选择 | 请选择 | 请输入 | 请输入 | ☐ | ☐ | ☐ | <div>+</div> <div>-</div> |

上一步下一步取消

表 8-15 输入参数说明

| 参数 | 说明 | 示例 |
|------|-----------------------------|-----|
| 标签 | 输入参数的标签名，用于在页面显示。 | sql |
| 名称 | 新建输入参数的名称，名称是输入参数在系统中的唯一标识。 | sql |
| 参数位置 | 设置输入参数的位置。 | 消息体 |
| 数据类型 | 设置输入参数的数据类型，直接在下拉框中选择。 | 字符串 |

步骤9 本示例无需设置输出参数，请单击删除第一行参数，再单击“保存”。

步骤10 返回动作页签，单击该动作所在行的，启用该动作。

----结束

步骤三：从第三方数据库获取数据

步骤1 在自定义连接器的“动作”页签中，单击步骤6中添加的动作，进入动作列表页。

图 8-42 单击动作名称

请输入内容搜索

demoDB

demoDB

服务详情 动作 认证信息

新建 导入 导出

搜索

| 名称 | 标签 | 启用 | 修改人 | 修改日期 | 操作 |
|----------|---------|-------------------------------------|----------|---------------------|----------------------------|
| _demoSql | demoSql | ☒ | paas_... | 2024-12-10 14:11:39 | <div>ON</div> <div>-</div> |

共 1 条

10条/页

< 1 >

前往 1 页

步骤2 在页面右上方，单击“测试”，进入测试页面。

步骤3 “认证信息”选择步骤5中创建的noauth，设置输入参数，单击“测试”。

图 8-43 测试

测试

✕

* 认证信息

noauth

▼

输入参数 (请以JSON形式输入正确的字符串)

```
{"sql": "select * from indexes limit 1"}
```

参数信息

名称: sql

数据类型: String

取消

测试

```
"Server": "nginx"
}
}
```

----结束

步骤四：通过脚本调用连接器获取数据

步骤1 在应用设计器中，选择“逻辑”，单击脚本后的“+”。

步骤2 新建一个空白的脚本，名称设置为“dbscript”，单击“添加”。

图 8-44 创建脚本 dbscript

步骤3 在脚本编辑器中，输入示例代码。

示例代码中，“命名空间_demoDB”为**步骤4**中定义的连接器名称，“命名空间_noauth”为**步骤3**中添加的认证信息，“命名空间_demoSql”为**步骤6**中添加的动作。

```
import * as connector from 'connector';
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: false, label: 'String' })
  sql: string;
}
//定义出参结构
@action.object({ type: "param" })
export class ActionOutput {
  @action.param({ type: 'any' })
  res: any;
}
@action.object({ type: "method" })
```

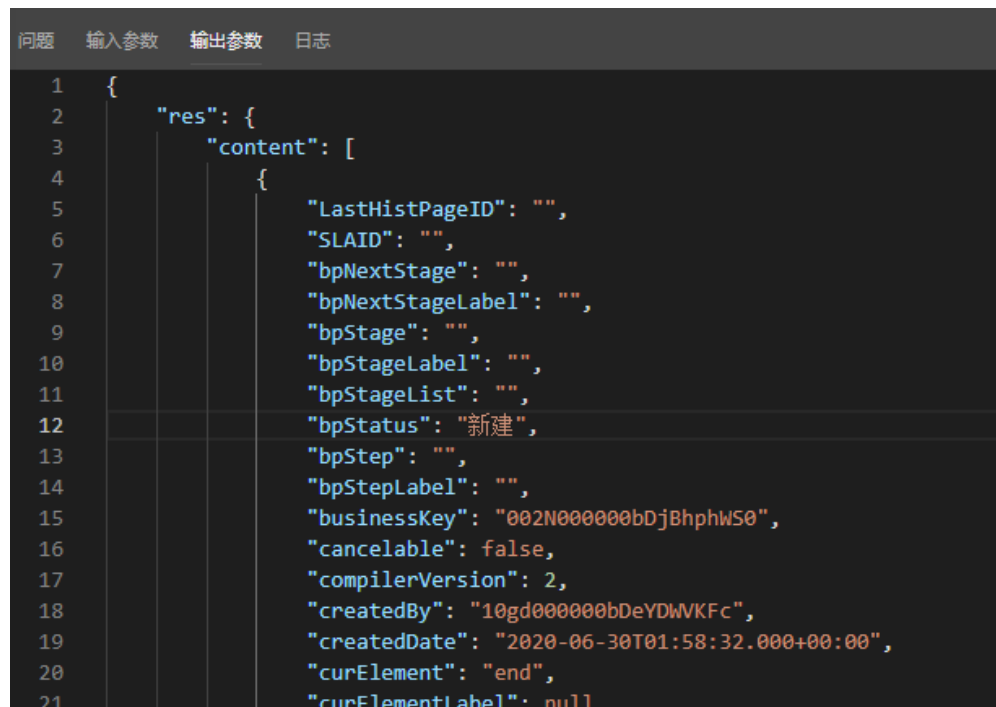
```
export class CreateWorkOrder { //定义接口类，接口的入参为ActionInput，出参为ActionOutput
  @action.method({ input: 'ActionInput', output: 'ActionOutput' })
  public createWorkOrder(input: ActionInput): ActionOutput {
    let out = new ActionOutput(); //新建出参ActionOutput类型的实例，作为返回值
    let error = new Error(); //新建错误类型的实例，用于在发生错误时保存错误信息
    try {
      // 页面上自己创建的自定义连接器的调用方法，第一个参数是创建的自定义连接器名称，第二个参数是
      认证信息名称
      let client = connector.newClient("命名空间_demoDB", "命名空间_noauth");
      //入参
      let input = { "sql": "select * from flowinstance limit 1" };
      // 第一个入参是动作名称，第二个参数是入参
      let resp = client.invoke("命名空间_demoSql", input);
      // 打印输出结果
      console.log(resp);
      out.res = resp;
    } catch (error) {
      console.error(error.name, error.message);
      context.setError(error.name, error.message);
    }
    return out;
  }
}
```

步骤4 单击页面上方的, 保存脚本。

步骤5 单击编辑器上方的, 执行脚本。

步骤6 在页面底部单击测试窗口右上角的, 在“输出参数”中可以查看到返回的数据库数据。

图 8-45 查看返回数据



----结束

9 业务用户专项

9.1 通过华为云 Astro 轻应用开发业务用户登录页

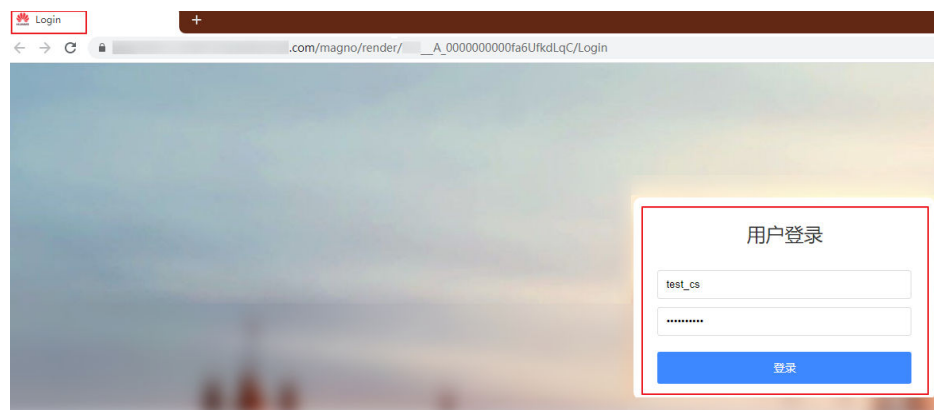
9.1.1 业务用户登录页方案概述

应用场景

使用华为云Astro轻应用开发完应用后，系统会为应用预置一个默认的登录页，业务用户可通过默认的登录页登录应用。除了使用默认的登录页，您还可以根据自身业务的实际需求，使用华为云Astro轻应用为应用自定义一个登录页。通过在登录页中输入用户名、密码和手机号码等登录信息，与系统中存储的业务用户信息进行对比，来验证业务用户的身份，并根据设置的业务用户权限，为业务用户分配相应的资源和访问权限。

例如，自定义一个图9-1中的登录页，在登录页中输入业务用户名及密码，单击“登录”，即可登录应用。其中，“登录”逻辑是通过“自定义登录”组件，调用用户登录服务编排完成的。在自定义登录页前，请先了解下方的[业务用户的登录方式](#)和[业务用户的登录机制](#)。

图 9-1 自定义登录页面



业务用户的登录方式

业务用户登录华为云Astro轻应用有两种登录方式：后台登录和前台登录。

- 业务用户在后台登录时，是使用自定义的服务编排来调用“login”脚本，查询登录账号密码，判断当前登录的账号密码是否正确，来实现业务用户后台登录功能。
- 业务用户在前台登录时，需要先在线下开发一个登录组件，上传到高级页面，并在高级页面中配置组件桥接器中的数据。最后在页面中输入登录账号密码，通过调用“用户登录服务编排”，实现业务用户页面登录功能。

业务用户的登录机制

业务用户前台登录和后台登录，在登录过程中的服务逻辑实现过程如下：

1. 通过调用“账号密码校验”脚本，查询登录账号密码，判断当前登录的账号密码是否正确。
2. 如果判断账号密码错误，直接执行“账号密码错误”。账号密码正确，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。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，验证码错误，则执行验证失败。

9.1.2 业务用户登录页后端逻辑开发实施步骤

业务用户在后台登录时，是使用自定义的服务编排，来调用“账号密码校验”脚本，查询登录账号密码，判断当前登录的账号密码是否正确来实现“业务用户”后台登录功能的。“业务用户登录”服务编排开发的大致过程为：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。

下面以“A”应用为例，介绍如何在后台实现业务用户登录。

步骤一：创建密码校验登录脚本

业务用户登录前，需要先创建一个账号密码校验脚本。

- 步骤1** 在华为云Astro轻应用服务控制台的主页中，单击“进入首页”，进入华为云Astro轻应用应用开发页面。
- 步骤2** 在“主页 > 全部应用”中，单击“A”应用后的“编辑”，进入应用设计器。
- 步骤3** 在左侧导航栏中，选择“逻辑”，单击脚本后的“+”。
- 步骤4** 选中“创建一个新脚本”，“名称”设置为“login”，单击“添加”。

图 9-2 新增脚本

新增脚本

×

● 创建一个新脚本

○ 使用已有脚本

* 名称

login

模板

空脚本

描述

取消

添加

步骤5 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";
//定义入参结构，账号的用户名、密码为必需字段。如果根据业务需要，需校验其他字段（例如验证码），则根据账号密码字段的格式进行新增即可。
@action.object({type:"param"})
export class ActionInput{
  @action.param({type:'String', required:true, label:'string'})
  username:string; //用户名
  @action.param({type:'String', required:true, label:'string'})
  password:string; //密码
  @action.param({type:'String', required:true, label:'string'})
  captcha:string; //验证码，本脚本只是为了校验账号密码，因此用不到验证码，验证码也不是必需字段。在实现业务用户后台登录的Flow中，Flow调用此脚本，需要判断验证码，所以在此脚本中添加了验证码字段。
}
//定义出参结构，结构中的字段可以根据业务需要按下方样例结构进行添加或减少。
@action.object({type:"param"})
export class ActionOutput{
  @action.param({type:'String'})
  msg:string;//登录信息
  @action.param({type:'String'})
  username:string;//用户名
  @action.param({type:'String'})
  userId:string;//用户ID
  @action.param({type:'String'})
  captcha:string;//验证码
}
//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;
        }else{
            out.msg ="账号或者密码错误! ";
        }
    }else{
        out.msg ="账号或者密码错误! ";
    }
} catch(error){
    console.error(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.Hashs.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6 单击编辑器上方的，保存脚本。

步骤7 测试脚本能否正常执行。

1. 单击编辑器上方的，执行脚本。
2. 在界面底部的输入参数中，输入如下测试数据，单击。
其中，“test_cs”、“{XXXXXXXX}”为注册的业务用户账号和密码，
“captcha”验证码非必填项为空。

在业务配置中心创建的业务用户和使用脚本创建的业务用户不能通用，故此处设置的业务用户（如test_cs）不能为在业务配置中心创建的业务用户。业务配置中心创建的业务用户，只能在默认登录页中使用。

```
{
  "username": "test_cs",
  "password": "{XXXXXXXX}",
  "captcha": ""
}
```

执行成功后，在“输出参数”中可参看结果。

```
{{
  "captcha": "",
```

```
"msg": "登录成功! ",
"userId": "10*****A",
"username": "test_cs"
}
```

步骤8 测试成功后，单击编辑器上方的，启用该脚本。

----结束

步骤二：通过服务编排开发登录页后端逻辑

步骤1 在“A”应用的设计器中，单击左侧导航栏的“逻辑”，单击编排后的，新建一个Flow目录。

图 9-3 创建目录



步骤2 将鼠标放在“Flow”上，单击界面上出现的“+”，进入添加服务编排页面。

步骤3 选中“创建一个新的服务编排”，“标签”和“名称”设置为“login”，类型设置为“Autolaunched Flow”，单击“添加”。

图 9-4 创建服务编排

添加服务编排

● 创建一个新的服务编排

使用已有的服务编排

* 标签

login

* 名称

login

* 类型

Autolaunched Flow

描述

请输入

取消

添加

步骤4 定义服务编排用到的变量。

1. 单击，展开全局上下文，再单击“变量”后的，设置参数名称为“username”。

图 9-5 新增变量



2. 重复上一步，定义表9-1中其他变量。

表 9-1 服务编排变量说明

| 名称（变量名称，唯一标识） | 数据类型 |
|---------------|------|
| username | 文本 |
| password | 文本 |

| 名称（变量名称，唯一标识） | 数据类型 |
|---------------|------|
| captcha | 文本 |
| msg | 文本 |
| userId | 文本 |
| loginMsg | 文本 |

- 单击“公式”后的 $\oplus$ ，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 9-6 添加公式变量“portalUserLogin”

- 参考上一步，创建表9-2中公式变量“verifyCode”。

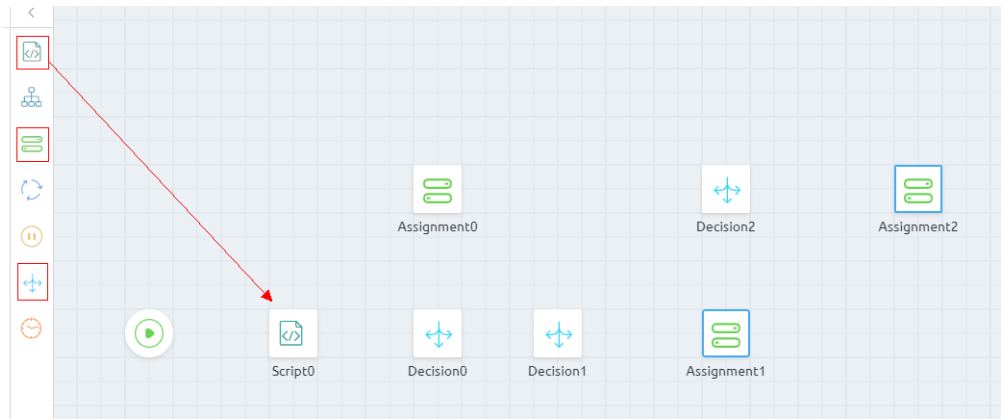
表 9-2 公式变量说明

| 名称 | 表达式 |
|-----------------|--|
| portalUserLogin | PORTALUSERLOGIN({!username}) |
| 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，图元排列如下图所示。

图 9-7 图元排列



2. 选中“Script0”图元，在右侧基本信息中，设置“标签”为“查询用户”。
3. 参考上一步，设置其他图元的“标签”属性，具体值如下表所示。

表 9-3 设置其他图元标签属性

| 名称（变量唯一标识，不需要修改） | 标签 |
|------------------|-----------|
| 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 9-8 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（命名空间_login），并配置脚本的输入输出参数。

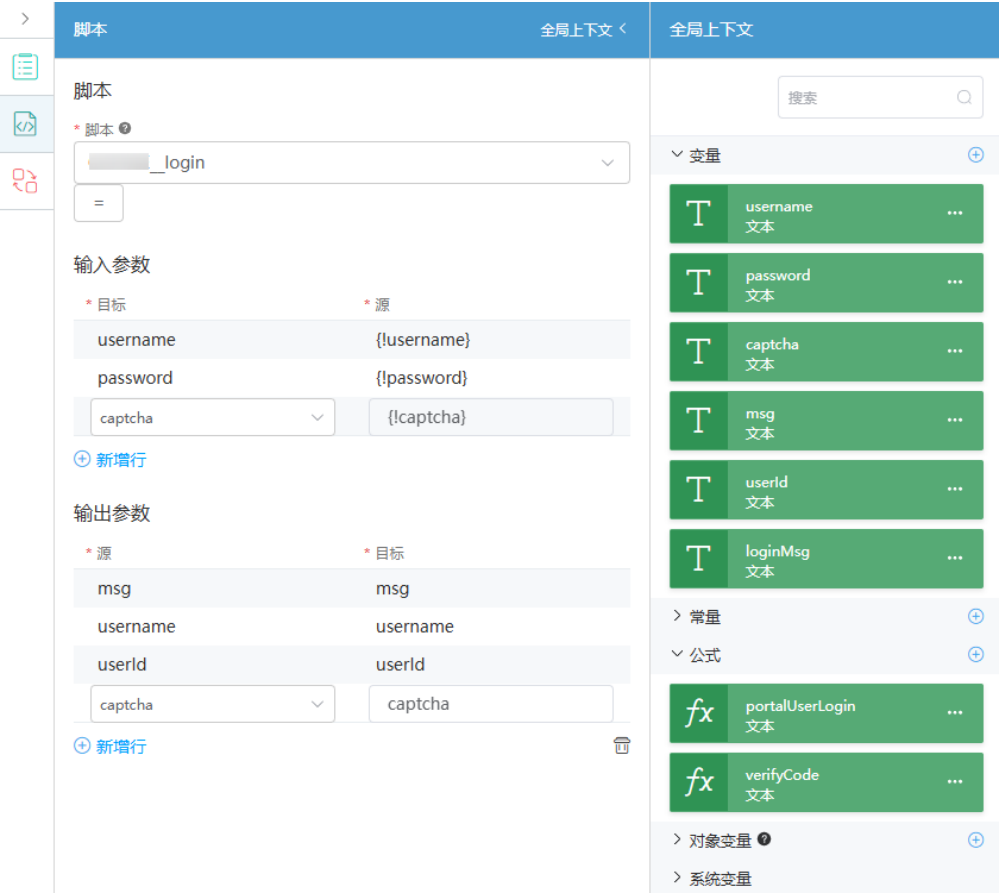
图 9-9 指定脚本



2. 单击“全局上下文”，显示变量列表，从“变量”中，拖拽“username”、“password”和“captcha”到“输入参数”下对应的“源”输入框中，在“输出参数”下，单击4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下图所示。
- 脚本图元中，输入参数、输出个数和指定脚本中需要的输入参数字段数是一致的。如果自定义脚本的输入参数有额外字段，额外的字段也需要同样操作。

- 请直接从全局上下文拖拽“变量”到对应的输入输出参数下，如果手动输入请确认输入参数与全局上下文中变量的值一致。

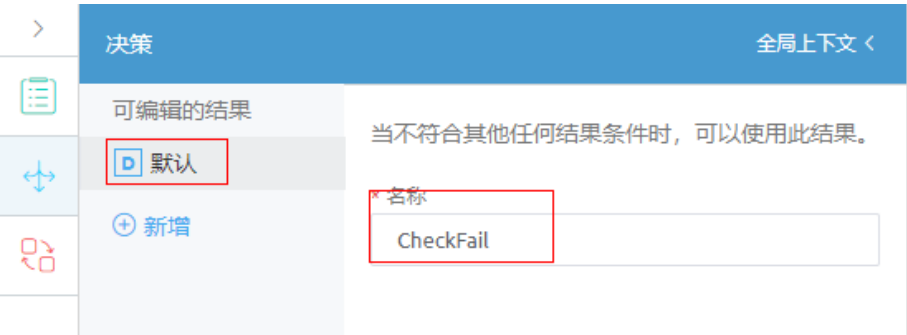
图 9-10 拖拽脚本的输入输出参数



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“CheckFail”。

图 9-11 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“CheckSuccess”，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!”。

图 9-12 修改可编辑的结果

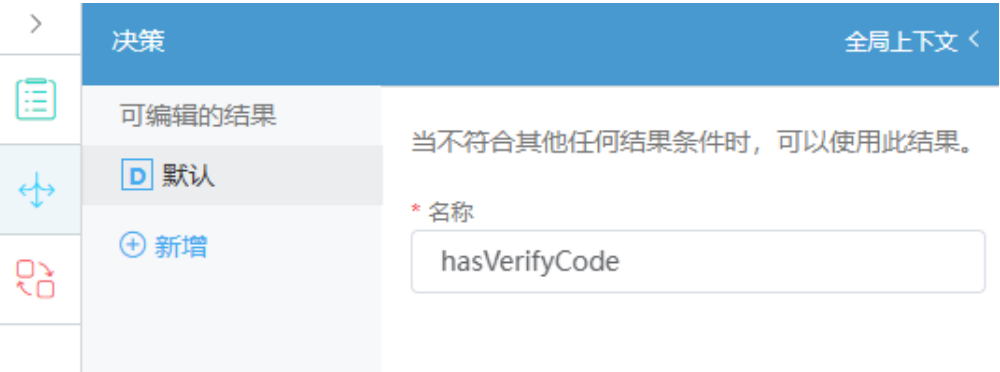


- 请直接从全局上下文拖拽变量“msg”到“资源”下，如果手动输入请确认输入参数与全局上下文中变量的值一致。
- “登录成功！”需要与“login”登录脚本中的输出参数一致。

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 9-13 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“”。

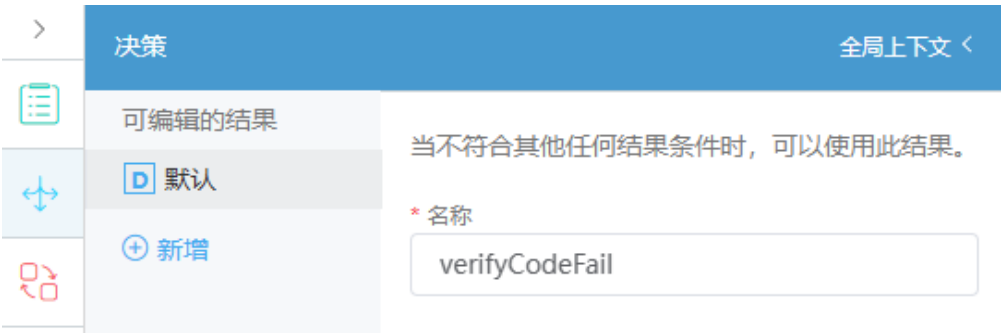
图 9-14 修改可编辑的结果



步骤9 配置“校验验证码”决策图元。

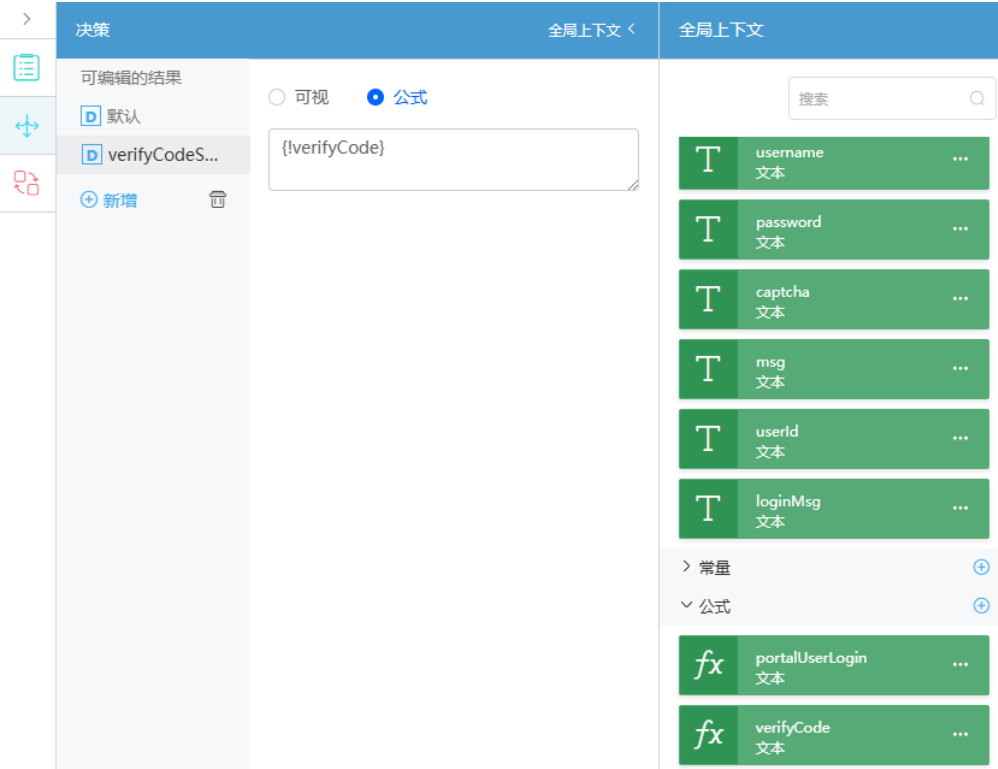
1. 选择“校验验证码”图元，在右侧单击图标，修改“默认”的“名称”为“verifyCodeFail”。

图 9-15 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 9-16 修改可编辑的结果



步骤10 配置“账号密码错误”赋值图元。

图 9-17 配置“账号密码错误”图元

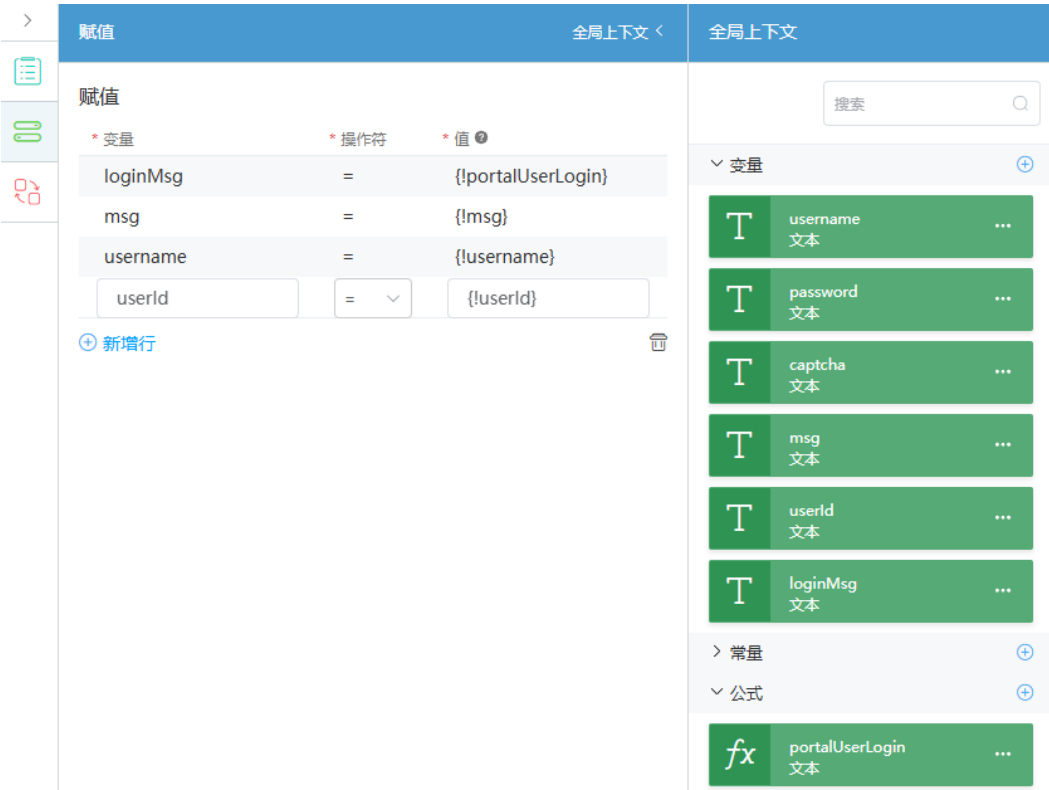


1. 选择“账号密码错误”图元，在右侧单击图标，单击“新增行”。

2. 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”。
3. 单击“新增行”，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"”。

步骤11 配置“执行登录”赋值图元。

图 9-18 拖拽“执行登录”赋值的变量及值



1. 选择“执行登录”图元，在右侧单击图标，单击4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，如下图所示。

表 9-4 变量与值对应关系

| 变量 | 操作符 | 值 |
|----------|-----|--------------------|
| loginMsg | = | {!portalUserLogin} |
| msg | = | {!msg} |
| username | = | {!username} |
| userId | = | {!userId} |

步骤12 配置“验证失败”赋值图元。

图 9-19 配置“验证失败”赋值图元



1. 选择“验证失败”图元，在右侧单击图标，单击“新增行”。
2. 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“账号或者密码错误!”、“1”。

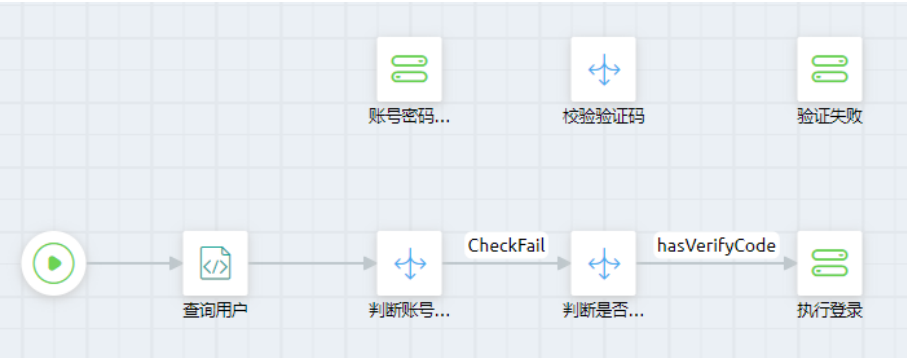
表 9-5 赋值

| 变量 | 操作符 | 值 |
|----------------|-----|--------------|
| \$Flow.ResMsg | = | "账号或者密码错误! " |
| \$Flow.ResCode | = | "1" |

步骤13 拖拽图元连线，并配置连线属性。

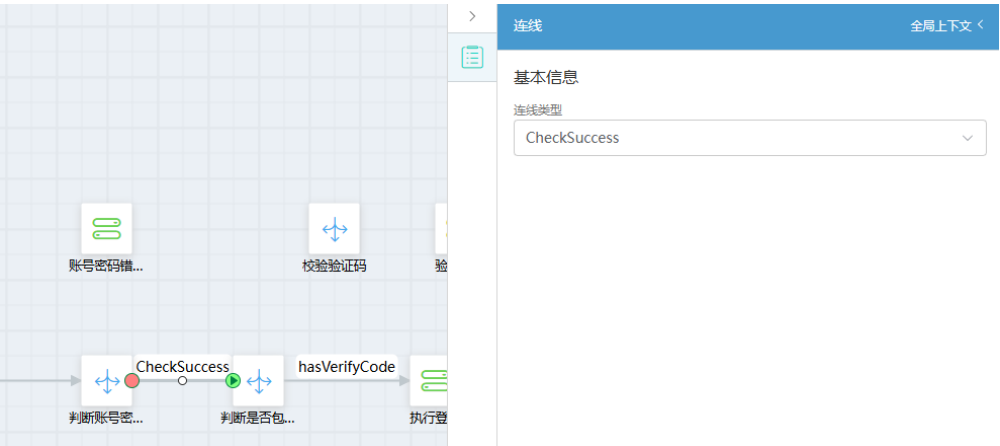
1. 在画布上，把鼠标放在起点图元上，从“+”拖动鼠标，在起点图元和“查询用户”图元间增加连线；即将当前脚本设置为服务编排的起始节点。
2. 依次在“查询用户”、“判断账号密码”、“判断是否包含验证码”、“执行登录”图元直接拖拽连线。

图 9-20 拖拽连线



3. 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，在右侧属性单击，在“连线”中修改“连线类型”为“CheckSuccess”。

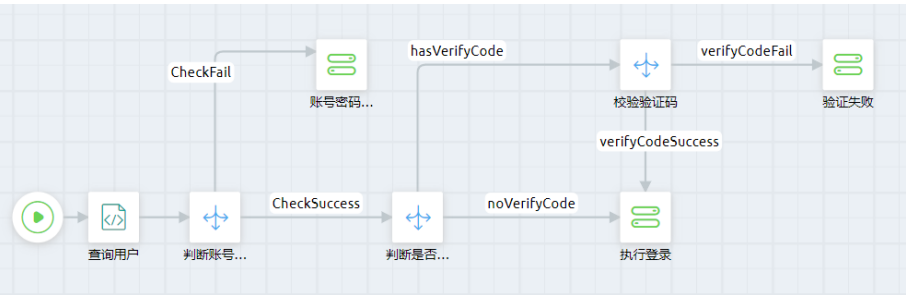
图 9-21 选中连线



4. 单击“判断是否包含验证码”与“执行登录”图元之间的连线，在右侧属性单击，在“连线”中修改“连线类型”为“noVerifyCode”。
5. 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。
6. 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。
7. 从“校验验证码”图元上，拖拽一条连线到“验证失败”图元。
8. 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，如下图所示。

图 9-22 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。

1. 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，如下图所示。

图 9-23 拖拽服务编排的输入输出参数



- 服务编排的输入参数是用来执行服务编排时输入的参数，同时也是执行账号密码校验脚本时的输入参数。所以当账号密码校验脚本时，如果有额外的输入参数字段，服务编排的输入参数也需要同步增加。
- 服务编排的输出参数是执行账号密码校验脚本时返回的参数，所以当账号密码校验脚本时，如果有额外的输出参数字段，服务编排的输出参数也需要同步增加。

2. 单击服务编排页面上方的，保存服务编排。

步骤15 测试服务编排能否正常执行。

1. 单击服务编排页面上方的，进入服务编排测试页面。

图 9-24 服务编排测试页面

__login

▶ 请求头参数

保存输入参数

输入参数

参数详情

| | |
|--------------|----------|
| 名称: username | 数据类型: 文本 |
| 名称: password | 数据类型: 文本 |

运行

- 在“Flow Run”界面中，输入测试数据，单击“运行”。

其中，“test_cs”、“***”为业务用户的账号和密码。

```
{
  "username": "test_cs",
  "password": "***",
  "captcha": ""
}
```

执行成功后，界面上会返回设备对象中的全部信息，示例如下：

图 9-25 返回值示例



返回值提示“登录成功”，完成业务用户的登录。业务用户登录成功后，返回华为云Astro轻应用，刷新页面后在页面右上角可以看到当前登录的用户已变成在服务编排中输入的业务用户。

步骤16 测试成功后，单击编辑器上方的，启用并发布服务编排。

步骤17 测试成功，使用平台开发者用户重新登录，单击服务编排编辑器上方的，启用并发布服务编排。

说明

测试成功后，此时页面的登录用户已经变成了在服务编排中输入的业务用户，直接单击服务编排编辑器上方的，会报无权限执行，因此需要切换为平台开发者用户重新登录后再启用该编排。

----结束

步骤三：新建开放接口

开放接口是用户将在应用中开发的脚本、服务编排等包装成自定义REST接口的形式，供其他系统进行调用。本示例将开发的服务编排包装发布REST接口，供[业务用户登录页前台开发实施步骤](#)中操作调用。

步骤1 在“A”应用的设计器中，单击左侧导航栏的“集成”。

步骤2 单击开放接口后的“+”，进入新建开放接口页面。

步骤3 设置接口参数，单击“保存”。

图 9-26 新建开放接口

新建开放接口

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

login

* 操作名称

login

* 版本

1.3.3

* URL

/service/{命名空间}/{应用名}/{版本}/Flow_login

内容类型

application/json

分类

请输入

描述

请输入

☐ 允许匿名访问

类型

☒ 服务编排 ☐ 脚本 ☐ 对象

☐ 自定义响应

* 资源

_login

* 方法

POST

表 9-6 开放接口参数说明

| 参数 | 说明 | 示例 |
|------|--|-------------------------------|
| 标签 | 新建API接口的标签，长度不能超过64个字符。 | login |
| 操作名称 | 新建API接口的操作名称，必须以英文字母开头，由英文字母，数字和单下划线组成，不能以单下划线结尾，且长度不能超过40个字符。 | login |
| 版本 | API接口的版本，格式为“x.y.z”。 | 1.3.3 |
| URL | API接口路径，固定以 /service/{命名空间}_{应用名}/{版本} 开头，后面接API的具体路径。此处设置的内容是新建开放接口提供给外部访问的URL。 | /service/命名空间_A/版本/Flow_login |
| 类型 | 选择资源类型，只有服务编排类型的接口能够在服务编排中被调用，其他类型接口只能通过API的形式进行调用。 | 服务编排 |

| 参数 | 说明 | 示例 |
|----|---|--|
| 资源 | 根据类型选择需要绑定的资源。 | 选择 步骤二：通过服务编排开发登录页后端逻辑 中创建的服务编排，请确保服务编排已启用，否则此处选择不到。 |
| 方法 | API接口的HTTP方法。
本示例选择“POST”，即请求服务器新增资源或执行特殊操作。 | POST |

----结束

9.1.3 业务用户登录页前台开发实施步骤

业务用户在前台登录时，需要先在线下开发一个登录组件，上传到高级页面，并在高级页面中配置组件桥接器中的数据。最后在页面中，输入登录账号密码，通过调用“用户登录服务编排”，实现“业务用户”页面登录功能。

本文以“A”应用为例，介绍如何开发登录页面，具体流程如[图9-27](#)所示。

图 9-27 应用登录页面开发流程



除了自定义登录组件外，您还可以直接使用高级页面预置的登录组件，快速实现登录功能的配置。更多关于登录组件的介绍，请参见[设置华为云Astro轻应用高级页面登录组件属性](#)。

步骤一：开发自定义组件

本示例中，提供了一个已开发好的自定义登录组件userLogin，单击[userLogin.zip](#)可直接下载使用。如果想要了解自定义登录组件的开发方法，可参考本步骤执行。

- 步骤1 在华为云Astro轻应用服务控制台的主页中，单击“进入首页”，进入应用开发页面。
- 步骤2 单击，选择“环境管理 > 环境配置”，进入环境配置。
- 步骤3 在主菜单中，选择“维护”。
- 步骤4 在左侧导航栏中，选择“全局元素 > 页面资产管理 > 组件模板”。
- 步骤5 在组件模板列表中，单击“widgetVue3Template”，进入模板详情页。
- 步骤6 单击“下载”，设置组件的名称为“userLogin”，单击“保存”，将模板下载到本地。

图 9-28 保存模板

下载 组件模板

* 组件名称

userLogin

组件描述

取消

保存

下载原始模板

步骤7 查看解压后的组件目录。

图 9-29 解压后目录

| userLogin.zip - ZIP archive, unpacked size 3,890 bytes | | | |
|--|-------|--------|---------------|
| Name | Size | Packed | Type |
| .. | | | 文件夹 |
| packageinfo.json | 271 | 141 | JSON 文件 |
| userLogin.css | 33 | 35 | 层叠样式表文档 |
| userLogin.editor.js | 1,357 | 527 | JavaScript 文件 |
| userLogin.ftl | 85 | 84 | FTL 文件 |
| userLogin.js | 2,144 | 820 | JavaScript 文件 |

- userLogin.js：存放vue业务逻辑的代码。
- userLogin.ftl：存放html代码。
- userLogin.css：存放样式代码。
- userLogin.editor.js、packageinfo.json：配置文件。

步骤8 在解压后的文件夹中，创建一个imgs文件夹，并放入一个登录页面的背景图片。

假设放入的背景图片名称为“imagebg.jpg”。

步骤9 在本地编辑器中打开文件夹，把userLogin.editor.js文件中的propertiesConfig代码改为如下代码，用于配置桥接器。

```
propertiesConfig: [  
  {  
    headerTitle: 'Parameters',  
    accordion: true,  
    accordionState: 'open',  
    config: [  
      {  
        type: 'text',  
        name: 'loginAPI',  
        label: 'login API',  
        value: '',  
      },  
    ],  
  },  
  {  
    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10 把packageinfo.json文件中加入如下**加粗内容**。

```

{
  "widgetApi": [
    {
      "name": "userLogin"
    }
  ],
  "widgetDescription": "",
  "authorName": "",
  "localFileBasePath": "",
  "requires": [
    {
      "name": "global_Vue3",
      "version": "3.3.4"
    },
    {
      "name": "global_ElementPlus",
      "version": "2.6.0"
    },
    {
      "name": "global_Vue3I18n",
      "version": "9.10.1"
    },
    {
      "name": "global_Vue3Router",
      "version": "4.3.0"
    }
  ]
}

```

步骤11 将userLogin.ftl文件中的内容，替换为如下代码。

```

<style>
  [v-cloak] {
    display: none
  }
</style>
<div v-cloak id="userLogin" class="page-login" v-cloak>
  <div :class="backgroundClass" class="bg-box">
    <!-- <div class="title">
      <span>设备管理系统</span>
    </div -->
  </div> -->

```

```

<div class="login-box">
  <div class="login-title">用户登录</div>
  <input name="username" type="text" style="display: none" />
  <input name="password" type="password" style="display: none" />
  <div class="login-form">
    <div v-show="errorDesc" class="error-line">
      <!--  -->
      <span class="error-text" v-html="errorDesc"></span>
    </div>
    <div class="login-item mg-top10">
      <!-- :placeholder="isUserName ? getTransLang('ds.commerce.storefront.web.loginname') :
getTransLang('ds.commerce.storefront.web.loginEmailOrPhone')" -->
      <input
        ref="accountInput"
        v-model="account"
        placeholder="请输入用户名"
        @keyup.enter="validateBeforeSubmit"
        autocomplete="off"
        maxlength="32"
        type="text"
      />
    </div>
    <div class="login-item mg-top10">
      <!-- :placeholder="getTransLang('ds.commerce.storefront.web.password')" -->
      <input
        v-model="password"
        @keyup.enter="validateBeforeSubmit"
        placeholder="请输入密码"
        autocomplete="off"
        maxlength="32"
        type="password"
      />
    </div>
    <div v-if="needVerify" class="login-item">
      <!-- :placeholder="getTransLang('ds.commerce.storefront.web.verificationcode')" -->
      <input
        v-model="inputImgCode"
        @keyup.enter="validateBeforeSubmit"
        placeholder="请输入验证码"
        autocomplete="off"
        maxlength="10"
        type="text"
      />
      <div class="verify-code">
        
      </div>
    </div>
    <div class="login-button" @click="validateBeforeSubmit">
      登录
    </div>
  </div>
</div>
</div>
</div>

```

步骤12 将userLogin.js文件内容，替换为如下代码。

```

userLogin = StudioWidgetWrapper.extend({
  init: function () {
    var thisObj = this
    thisObj._super.apply(thisObj, arguments)
    thisObj.render()
    thisObj.initBusi()

    if (typeof Studio !== 'undefined' && Studio) {
      Studio.registerEvents(thisObj, 'goHomepage', 'go Homepage', [])
    }
  },
  render: function () {

```

```
var thisObj = this
let tenantId;
if (Studio.inReader) {
  tenantId = STUDIO_DATA.catalogProperties["tenant-id"].value
} else {
  tenantId = magno.pageService.getCatalogProperties()["tenant-id"].value
}
HttpUtils.setCookie("tenant-id", tenantId)
HttpUtils.setCookie("locale", "zh_CN");
var elem = thisObj.getContainer()

if (elem) {
  var containerDiv = $(''.scfClientRenderedContainer', elem)
  if (containerDiv.length) {
    $(containerDiv).empty()
  } else {
    containerDiv = document.createElement('div')
    containerDiv.className = 'scfClientRenderedContainer'
    $(elem).append(containerDiv)
  }
}

thisObj.sksBindItemEvent()
$(window).resize(function () {
  thisObj.sksRefreshEvents()
})
},

initBusi: function () {
  var thisObj = this
  var widgetProperties = thisObj.getProperties()
  var BasePath = thisObj.getWidgetBasePath() // 本地路径
  var elem = thisObj.getContainer()
  const app = Vue.createApp({
    data(){
      return {
        BasePath: BasePath,
        imgCode: "",
        account: "",
        password: "",
        inputImgCode: "",
        errorDesc: "",
        needRead: false,
        backgroundClass: 'backgroundClass',
        needVerify: false,
        accountName: "",
        isMobile: false,
      }
    },
    created() {
      this.getVerifyCode()
      this.checkIsLogin()
      this.isMobile = /Android|webOS|iPhone|iPad|BlackBerry/i.test(
        navigator.userAgent
      )
    },
    watch: {
      account(newVal, oldVal) {
        if (newVal !== oldVal) {
          this.password = ""
        }
      },
    },
    methods: {
      checkIsLogin() {
        const userInfo = JSON.parse(
          window.sessionStorage.getItem('userInfo')
        )
        if (userInfo !== null && userInfo.username) {
```

```

        setTimeout(() => {
            thisObj.triggerEvent('goHomepage', {})
        }, 1000)
    },
    shouldShowImgCode() {
        connService(
            thisObj,
            `${ds_baseUrl}/identities/isNeedCaptcha`,
            {
                username: this.account,
            },
            'common.PostConnector'
        )
        .then((res) => {
            const { data = [], resp = {} } = res

            if (resp.code !== '0') {
                this.$message.error(resp.message)
            }

            if (data && data.length) {
                this.needVerify = data[0].needVerify

                if (!this.needVerify) {
                    this.confirmLogin()
                }
            }
        })
        .catch((e) => {
            this.$message.error(e.response.resMsg)
        })
    },
    // 登录按钮
    validateBeforeSubmit() {
        if (!this.password) {
            this.errorDesc = '请输入密码。'
            this.needVerify = true
            return false
        }
        if (!this.account) {
            this.errorDesc = '请输入用户名。'
            this.needVerify = true
            return false
        }

        if (!this.inputImgCode && this.needVerify) {
            this.errorDesc = '验证码错误。'
            return false
        }

        if (this.needVerify && this.inputImgCode) {
            this.confirmLogin()
            return
        }
        this.confirmLogin()
    },
    // 调登录接口
    confirmLogin() {
        var request = {
            username: this.account,
            password: this.password,
            captcha: this.inputImgCode,
        }
        //
        this.callFlowConn1(
            widgetProperties.loginAPI,

```

```

        request,
        this.callLogin,
        this.loginFail,
        'post'
    )
},

// 登录接口成功函数
callLogin(response) {
    if (response) {
        this.errorDesc = ''
        let userInfo = {
            username: response.data[0].username,
            userId: response.data[0].userId,
            profile: response.data[0].profile,
        }
        HttpUtils.setCookie('isLogged', true)
        window.sessionStorage.setItem(
            'userInfo',
            JSON.stringify(userInfo)
        )
        thisObj.triggerEvent('goHomepage', {})
    }
},

// 登录失败resMsg
loginFail(data){
    this.errorDesc = data.response.resMsg
    this.getVerifyCode()
    this.needVerify = true
},

// 获取验证码
getVerifyCode() {
    this.imgCode =
        '/u-route/baas/sys/v1.0/verificationcode?type=login&t=' +
        Date.parse(new Date())
},

// 封装flow调用后台
callFlowConn1: function (service, param, callbackFunc, callbackfail, method) {
    var thisView = this
    let mMethod
    switch (method) {
        case 'get':
            mMethod = 'common.GetConnector'
            break
        case 'post':
            mMethod = 'common.PostConnector'
            break
        case 'put':
            mMethod = 'common.PutConnector'
            break
        default:
            mMethod = 'common.FlowConnector'
            break
    }

    var connector = thisObj.getConnectorInstanceByName(mMethod)
    if (connector) {
        connector.setInputParams({
            service: service,
            needSchema: 'data',
            async: false,
        })
        connector
            .query(param) // 调用接口，以param为入参
            .done(function (response) {
                if (response.resp && response.resp.code) {
                    callbackFunc.call(thisView, response)
                }
            })
    }
}

```



```

        })
        .fail(function (response) {
            // 代表接口执行失败
            callbackfail.call(thisView, response)
        })
    },
},
},
})
app.use(ElementPlus);
thisObj.vm = app.mount("#userLogin", elem)[0]);
},
})

```

步骤13 将userLogin.css文件内容，替换为如下代码。

需要注意的是，“background”的取值需要和[步骤8](#)中创建的文件夹和图片名称保持一致。

```

.page-login {
    /* 使用webkit内核的浏览器 */
    /* Firefox版本4-18 */
    /* Firefox版本19+ */
}
.page-login * {
    box-sizing: border-box;
}
.page-login .flex {
    display: flex;
}
.page-login .login_tip {
    font-size: 12px;
    color: #167aeb;
}
.page-login .bg-box {
    position: relative;
    display: flex;
    justify-content: center;
    width: 100%;
    height: 100%;
    background: url('imgs/imagebg.jpg') no-repeat 50%;
    background-size: cover;
}
.page-login .bg-box > img {
    width: 100%;
}
@media (min-width: 767px) {
    .page-login .bg-box .login-box {
        position: absolute;
        right: 18%;
        top: 26%;
        padding: 30px 30px 20px 30px;
        width: 380px;
        background: #fff;
        border-radius: 10px;
        box-shadow: 0 8px 16px 0 rgba(0, 0, 0, 0.1);
    }
    .page-login .bg-box .login-box .login-title {
        font-size: 24px;
        color: #333;
        text-align: center;
    }
    .page-login .bg-box .login-box .login-form {
        padding-top: 20px;
    }
    .page-login .bg-box .login-box .login-form .error-line {
        display: flex;
        align-items: center;
        background: #ffebeb;
    }
}

```

```

color: #e4393c;
border: 1px solid #facc66;
padding: 3px 10px 3px 10px;
height: auto;
text-align: left;
font-size: 12px;
}
.page-login .bg-box .login-box .login-form .error-line .error-text {
padding-left: 5px;
}
.page-login .bg-box .login-box .login-form .error-line .error-text .change_tab {
color: #167aeb;
cursor: pointer;
}
.page-login .bg-box .login-box .login-form .login-item {
display: flex;
align-items: center;
margin-top: 20px;
}
.page-login .bg-box .login-box .login-form .login-item .el-input__inner {
height: 36px;
}
.page-login .bg-box .login-box .login-form .login-item > input {
flex: 1;
padding: 5px 10px;
height: 36px;
border-radius: 3px;
border: 1px solid #ddd;
}
.page-login .bg-box .login-box .login-form .login-item .verify-code {
height: 36px;
margin-left: 5px;
}
.page-login .bg-box .login-box .login-form .login-item .verify-code img {
height: 100%;
}
.page-login .bg-box .login-box .login-form .login-item .el-checkbox {
margin-right: 10px;
}
.page-login .bg-box .login-box .login-form .login-item .read-text {
font-size: 12px;
color: #333;
}
.page-login .bg-box .login-box .login-form .login-item .read-text .clickable {
color: #167aeb;
cursor: pointer;
}
.page-login .bg-box .login-box .login-form .login-item .read-text .clickable:hover {
text-decoration: underline;
}
.page-login .bg-box .login-box .login-form .mg-top10 {
margin-top: 10px;
}
.page-login .bg-box .login-box .login-button {
margin-top: 20px;
height: 40px;
line-height: 40px;
border-radius: 3px;
background: #3d88ff;
text-align: center;
color: #fff;
font-size: 14px;
cursor: pointer;
}
.page-login .bg-box .login-box .login-button:hover {
opacity: 0.8;
}
.page-login .bg-box .login-box .divide-line {
margin-top: 20px;

```

```

height: 1px;
background: #eee;
opacity: 0.8;
}
.page-login .bg-box .login-box .login-bottom {
margin-top: 20px;
justify-content: center;
align-items: center;
}
.page-login .bg-box .login-box .login-bottom .bottom-text {
font-size: 12px;
color: #999;
}
.page-login .bg-box .login-box .login-bottom .type-item {
margin-left: 10px;
width: 30px;
height: 23px;
cursor: pointer;
}
.page-login .bg-box .login-box .login-bottom .type-item img {
width: 100%;
}
}
@media (max-width: 767px) {
.page-login .bg-box .login-box {
position: absolute;
right: auto;
top: 26%;
padding: 30px 30px 20px 30px;
width: 380px;
background: #fff;
border-radius: 10px;
box-shadow: 0 8px 16px 0 rgba(0, 0, 0, 0.1);
}
.page-login .bg-box .login-box .login-title {
font-size: 24px;
color: #333;
text-align: center;
}
.page-login .bg-box .login-box .login-form {
padding-top: 20px;
}
.page-login .bg-box .login-box .login-form .error-line {
display: flex;
align-items: center;
background: #ffebeb;
color: #e4393c;
border: 1px solid #facc6;
padding: 3px 10px 3px 10px;
height: auto;
text-align: left;
font-size: 12px;
}
.page-login .bg-box .login-box .login-form .error-line .error-text {
padding-left: 5px;
}
.page-login .bg-box .login-box .login-form .error-line .error-text .change_tab {
color: #167aeb;
cursor: pointer;
}
}
.page-login .bg-box .login-box .login-form .login-item {
display: flex;
align-items: center;
margin-top: 20px;
}
.page-login .bg-box .login-box .login-form .login-item .el-input__inner {
height: 36px;
}
.page-login .bg-box .login-box .login-form .login-item > input {

```

```

flex: 1;
padding: 5px 10px;
height: 36px;
border-radius: 3px;
border: 1px solid #ddd;
}
.page-login .bg-box .login-box .login-form .login-item .verify-code {
  height: 36px;
  margin-left: 5px;
}
.page-login .bg-box .login-box .login-form .login-item .verify-code img {
  height: 100%;
}
.page-login .bg-box .login-box .login-form .login-item .el-checkbox {
  margin-right: 10px;
}
.page-login .bg-box .login-box .login-form .login-item .read-text {
  font-size: 12px;
  color: #333;
}
.page-login .bg-box .login-box .login-form .login-item .read-text .clickable {
  color: #167aeb;
  cursor: pointer;
}
.page-login .bg-box .login-box .login-form .login-item .read-text .clickable:hover {
  text-decoration: underline;
}
.page-login .bg-box .login-box .login-form .mg-top10 {
  margin-top: 10px;
}
.page-login .bg-box .login-box .login-button {
  margin-top: 20px;
  height: 40px;
  line-height: 40px;
  border-radius: 3px;
  background: #3d88ff;
  text-align: center;
  color: #fff;
  font-size: 14px;
  cursor: pointer;
}
.page-login .bg-box .login-box .login-button:hover {
  opacity: 0.8;
}
.page-login .bg-box .login-box .divide-line {
  margin-top: 20px;
  height: 1px;
  background: #eee;
  opacity: 0.8;
}
.page-login .bg-box .login-box .login-bottom {
  margin-top: 20px;
  justify-content: center;
  align-items: center;
}
.page-login .bg-box .login-box .login-bottom .bottom-text {
  font-size: 12px;
  color: #999;
}
.page-login .bg-box .login-box .login-bottom .type-item {
  margin-left: 10px;
  width: 30px;
  height: 23px;
  cursor: pointer;
}
.page-login .bg-box .login-box .login-bottom .type-item img {
  width: 100%;
}
}

```

```
.page-login ::-webkit-input-placeholder {  
  font-size: 12px;  
  color: #aaa;  
}  
.page-login :-moz-placeholder {  
  font-size: 12px;  
  color: #aaa;  
}  
.page-login ::-moz-placeholder {  
  font-size: 12px;  
  color: #aaa;  
}  
.page-login :-ms-input-placeholder {  
  font-size: 12px;  
  color: #aaa;  
}  
.page-login .logining-text {  
  margin-top: 120px;  
  text-align: center;  
  font-size: 30px;  
  color: #333;  
}
```

步骤14 将修改后的配置文件和自定义开发的文件，压缩成一个zip包，包名为userLogin.zip。

----结束

步骤二：上传自定义登录组件

自定义组件开发完成后，需要上传到华为云Astro轻应用组件库中，供高级页面使用。

步骤1 在华为云Astro轻应用环境配置页面，单击主菜单中的“维护”。

步骤2 在左侧导航栏中，选择“全局元素 > 页面资产管理 > 组件”。

步骤3 单击“提交新组件”，进入提交新组件页面。

步骤4 单击“上传”，上传自定义组件包“userLogin.zip”。

图 9-30 上传自定义组件

提交新组件

取消 提交

上传图标

* 名字
userLogin

* 上传文件(zip)
请选择上传文件(zip)
userLogin.zip

组件ID
userLogin

分类
请选择

领域
通用

场景
请选择

* 发行说明
自定义组件

表 9-7 上传组件参数说明

| 参数 | 说明 | 示例 |
|-------|--|----------------------------------|
| 名字 | 新提交组件的名称，系统会根据组件包名称自动填充。 | userLogin |
| 上传源文件 | 组件的源文件包。 | 选择 步骤一：开发自定义组件 中的自定义组件登录包 |
| 场景 | 选择组件包的应用场景，可同时勾选多项，勾选后，在相应类型页面开发中，才可使用该组件。 | 高级页面 |
| 发行说明 | 组件的描述信息，需要配置不同语种下的描述信息。
此处配置的信息，将会在组件详情页的“概况”页签中进行展示。 | 自定义组件 |

步骤5 参数设置完成后，单击“提交”，完成组件的上传。
----结束

步骤三：创建高级页面

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

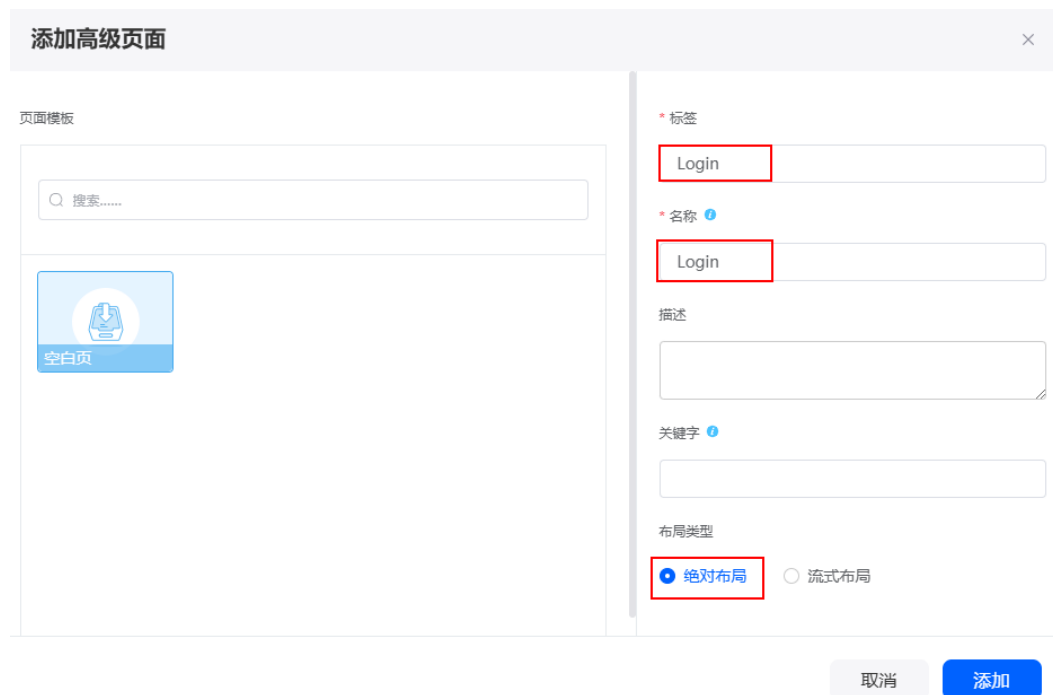
在组装业务用户登录页面时，需要配置从登录页面登录成功后跳转的页面，所以需要提前创建一个高级页面。在本示例中，假设已创建了一个Home页面。Home页面中没有实际数据，在Home页面中拖拽了几个图表组件，然后保存发布成样例页面。在实际开发过程中，可以根据业务需要，选择跳转到需要的页面，或者执行自定义的业务逻辑。

图 9-31 Home 页面



- 步骤1** 在华为云Astro轻应用服务控制台的主页中，单击“进入首页”，进入应用开发页面。
- 步骤2** 在“主页 > 全部应用”中，单击应用“A”后的编辑，进入应用“A”的设计器。
- 步骤3** 在左侧导航栏中，选择“界面”。
- 步骤4** 单击高级页面后的“+”，设置标签和名称为“Login”，并选择“绝对布局”，单击“添加”。

图 9-32 添加高级页面

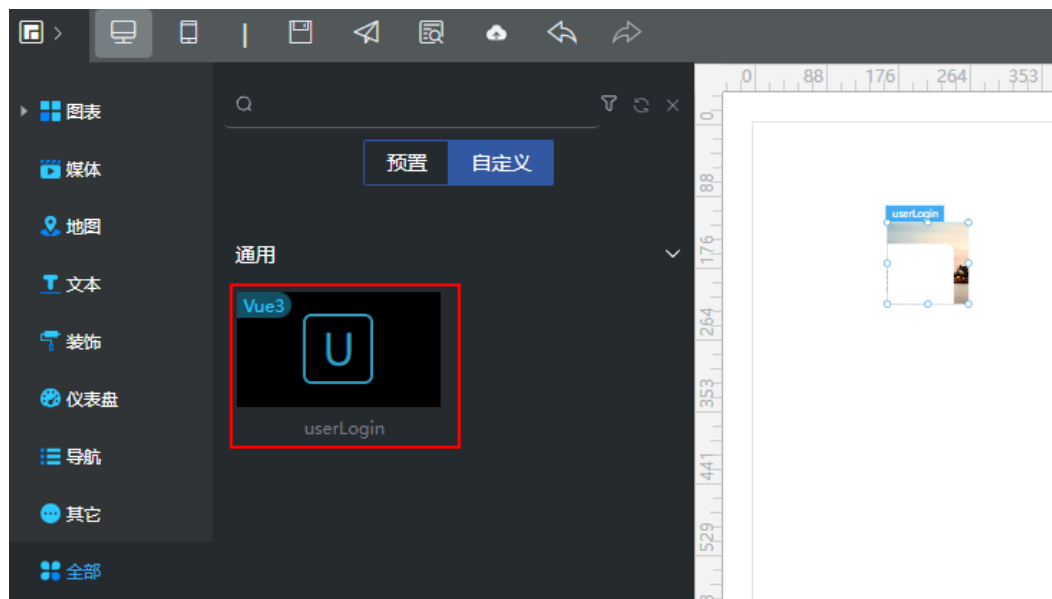


----结束

步骤四：拖拽组件开发登录页面

- 步骤1** 在**步骤三：创建高级页面**中创建的高级页面中，单击左上角的，选择“全部”，在“自定义”页签，拖拽“userLogin”组件到页面编辑区。

图 9-33 拖拽组件到编辑区



步骤2 设置自定义组件“userLogin”的位置属性。

1. 将“userLogin”组件拖拽到页面编辑区后，会在右侧显示该组件的属性配置面板。
2. 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 9-34 设置组件位置

属性

数据

事件

路由

基础

组件标题

userLogin

组件名称

widget2

位置

距离左端

0

px

距离顶端

0

px

宽度

1920

px

堆叠顺序

1

高度

1080

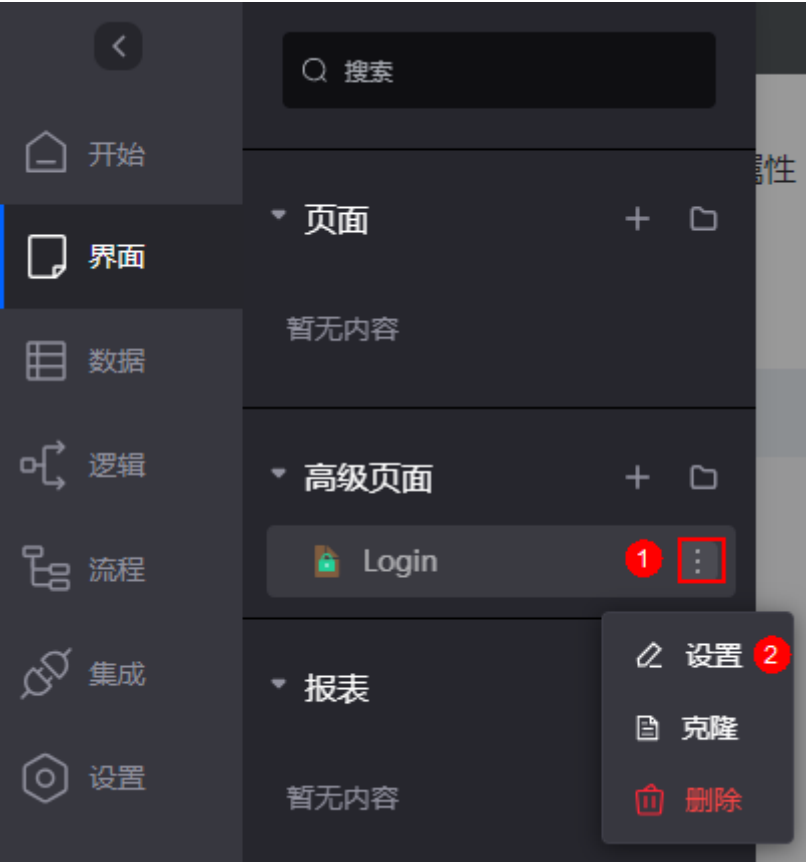
px

自适应高度

开启

3. 组件位置、样式等属性修改完成后，单击页面上方的，保存页面修改。
4. 在“界面 > 高级页面”中，将鼠标放在“Login”上，单击，选择“设置”。

图 9-35 选择设置



5. 在页面设置中，选中拉伸属性，单击“保存”。

图 9-36 设置页面拉伸

页面设置 - Login

基本设置

页面宏

水印

* 名称

Login

* 路径名称

Login

* 标题

Login

关键字

描述

拉伸

☒

缩放和拖移

☐

页面是否可定制修改

☐

浏览器缓存有效期 (天)

0

^

v

取消

保存

步骤3 设置自定义组件的桥接器。

1. 选中“userLogin”组件，在“属性 > Parameters”中，设置“login API”为“/命名空间_A/1.0.0/Flow_login”。

图 9-37 添加登录服务 URL

← 组件属性设置

属性

数据

事件

路由

基础

>

位置

>

边框

>

背景

>

Parameters

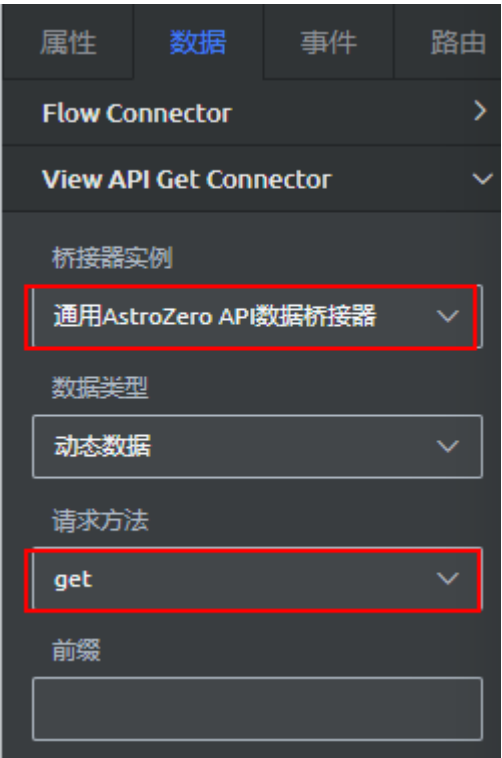
√

login API

/_A/1.0.0/Flow_login

- “login API” 为登录接口的URL后半段，“命名空间” 请根据实际情况配置，“A” 为应用名，登录接口是在[业务用户登录页后端逻辑开发实施步骤](#)中创建的。
2. 在“数据” 页签，单击“View API Get Connector”，设置“桥接器实例” 为“通用AstroZero API数据桥接器”，“数据类型” 为“动态数据”，“请求方法” 为“get”，如下图所示。

图 9-38 设置桥接器



3. 参考上一步，分别设置下图中框选的桥接器实例。

图 9-39 设置其他桥接器

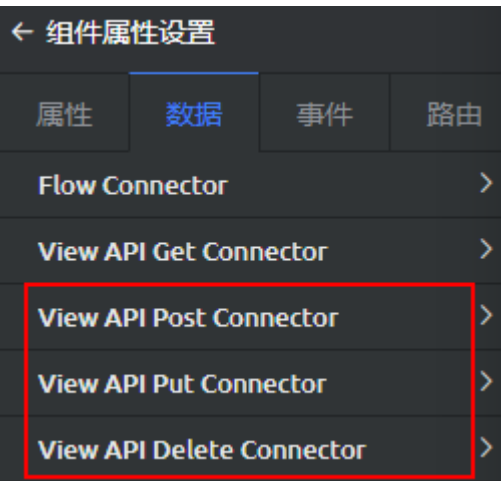


表 9-8 桥接器实例配置

| 数据名 | 桥接器实例 | 数据类型 | Request Method |
|---------------------------|----------------------|------|----------------|
| View API Post Connector | 通用AstroZero API数据桥接器 | 动态数据 | Post |
| View API Put Connector | 通用AstroZero API数据桥接器 | 动态数据 | Put |
| View API Delete Connector | 通用AstroZero API数据桥接器 | 动态数据 | Delete |

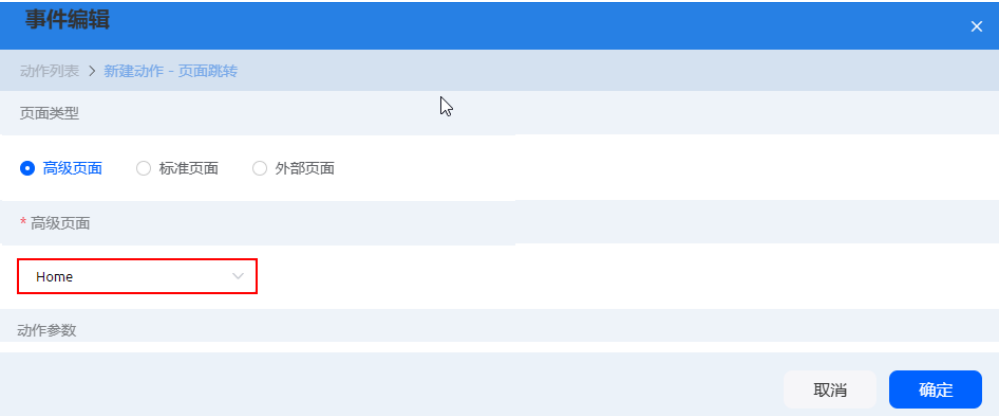
步骤4 设置自定义组件“userLogin”的“事件”，使自定义组件与其他页面关联。

图 9-40 需要配置的事件



- 1. 单击“go Homepage”后的齿轮图标，进入事件编辑页面。
- 2. 单击“新建动作”，选择“默认 > 页面跳转”，进入跳转编辑页面。
- 3. 选择需要跳转的页面，单击“确定”。
这里选择的高级页面是之前创建并发布的“Home”，在实际开发过程中，可以根据业务需要，选择跳转到需要的页面，或者执行自定义的业务逻辑。

图 9-41 编辑页面跳转

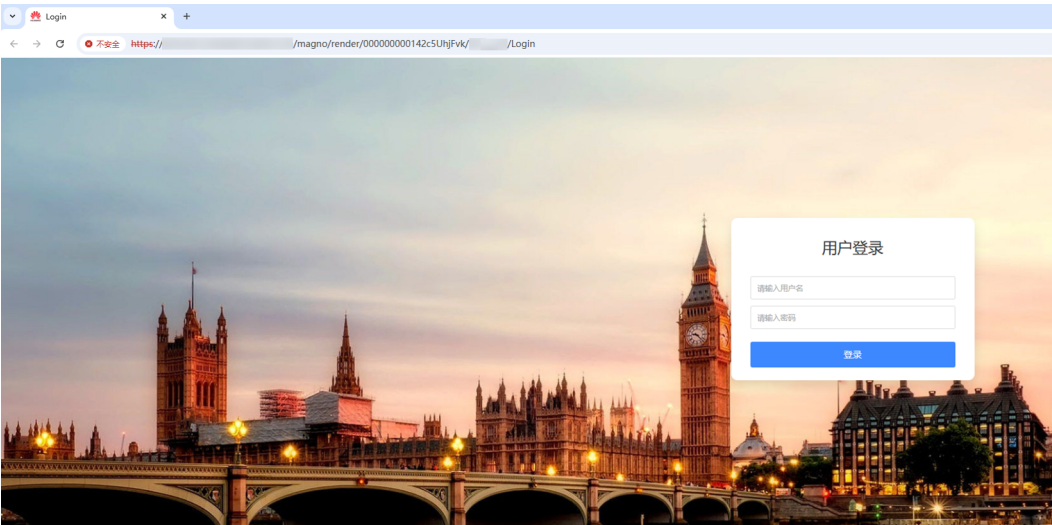


步骤5 在上述步骤完成后，单击Login页面上方的，保存页面，再单击，发布页面。

步骤6 页面发布成功后，单击，即可预览发布的登录页面。

当前登录页中，输入业务用户账号及密码，单击“登录”的登录逻辑是通过“自定义登录”组件，调用用户登录服务编排完成的。

图 9-42 预览的登录页面



步骤7 在预览的登录页面中，输入配置了权限的业务用户的账号密码，单击登录后，如果页面跳转到“Home”页面，则业务用户登录成功。

----结束

10 智能化专项

10.1 通过华为云 Astro 轻应用智能助手构建知识库

应用场景

对于查询文献资料的场景，当用户自身资料文档数量庞大，使用资料不方便、查询效率低下时，华为云Astro轻应用标准页面中的智能助手组件能够协助用户调用大模型接口获取答案。同时，通过RAG这个技术，结合知识库，为用户提供更为智能、自然的对话体验，提升信息检索的准确性和相关性。

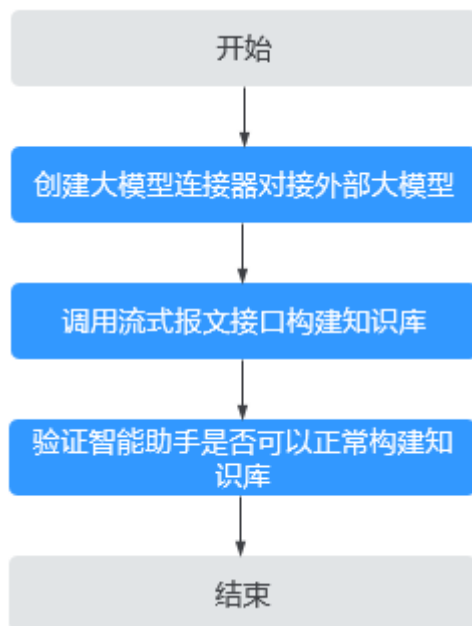
约束与限制

- 构建知识库功能受限使用，需要提交工单申请开通。
- 知识库的目录结构必须为**知识库 > 文件夹 > 文件**。用户能够创建多个知识库，在每个知识库中可以创建多个文件夹，并在文件夹中上传不同类型的文件。华为云Astro轻应用**专业版实例每个租户**最多可创建**40个**文件夹，**免费版和标准版实例**仅支持创建**1个**文件夹。上传单个文件时，单个文件的大小不能超过6MB。上传文件夹时，文件夹中文件的总大小不能超过50MB。上传文件的格式必须为TXT、DOCX、PDF、MD或CSV。

操作流程

在华为云Astro轻应用中，通过智能助手构建知识库的流程，如[图10-1](#)所示。

图 10-1 通过智能助手构建知识库操作流程



步骤一：创建大模型连接器对接外部大模型

在华为云Astro轻应用中，创建一个连接器来对接大模型接口。

步骤1 创建一个低代码应用。

1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
2. 购买实例后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
3. 在“应用”中，单击“新建低代码应用”或单击，进入新建低代码应用页面。首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。
4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。
5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 10-2 创建一个空白应用

新建 空白应用

基本信息

添加图标

* 标签

我的第一个应用

* 名称

A

分类

请选择

描述

请输入

高级设置

取消

新建

表 10-1 新建空白应用参数说明

| 参数 | 说明 | 示例 |
|----|--|---------|
| 标签 | 新建应用的标签，长度不能超过80个字符。标签是应用在系统中的唯一标识，创建后不支持修改。 | 我的第一个应用 |
| 名称 | 新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下： <ul style="list-style-type: none">长度不能超过31个字符，包括前缀命名空间的长度。名称前的内容为命名空间，在华为云Astro轻应用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。必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。 | A |

步骤2 创建大模型连接器。

1. 在应用设计器中，选择“集成 > 连接器 > 连接器实例”。

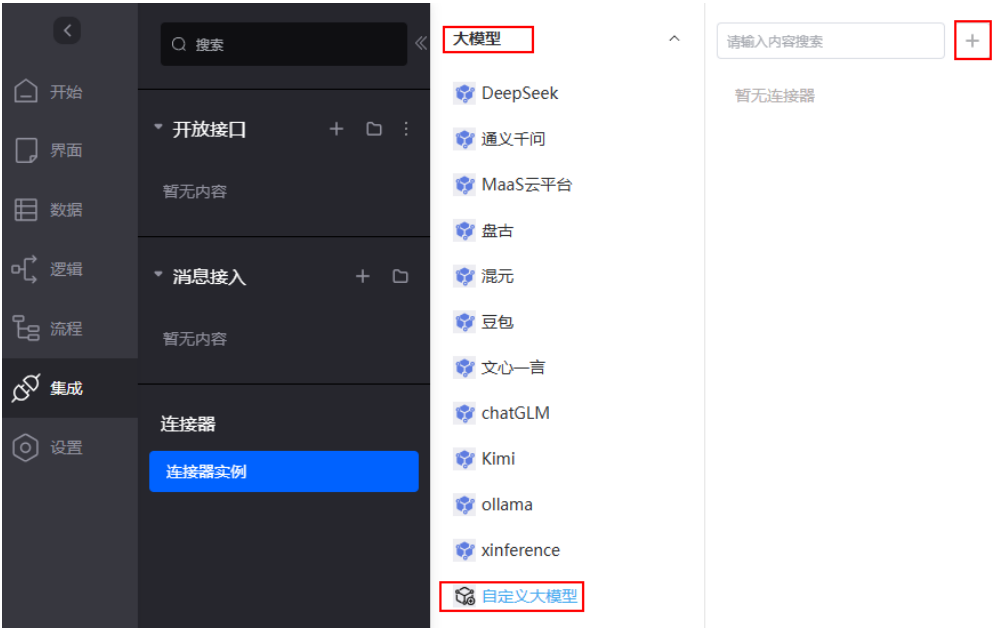
2. 选择“大模型 > 自定义大模型”，单击“+”，进入创建自定义大模型连接器页面。

文档版本 01 (2025-09-04)

版权所有 © 华为云计算技术有限公司

353

图 10-3 新建自定义大模型连接器



3. 输入大模型连接器的配置信息，单击“保存”。

图 10-4 设置大模型连接器信息

创建自定义大模型

基本信息

* 标签

DeepSeek-V3

* 名称

SelfHostDeepSeek

* 模型名称

DeepSeek-V3-0324-w8a8

* URL

/v1/chat/completion

API Key

请输入

上下文长度

请输入

最大token长度

请输入

描述

请输入

取消

保存

表 10-2 自定义大模型连接器参数说明

| 参数 | 说明 | 示例 |
|------|--|-----------------------|
| 标签 | 自定义大模型连接器的标签，创建后可修改。
取值范围：1~64个字符。 | DeepSeek-V3 |
| 名称 | 自定义连接器的名称，名称是连接器在系统中的唯一标识，创建后不可修改。命名要求如下： <ul style="list-style-type: none">长度不能超过64个字符，包括前缀命名空间的长度。名称前的内容为命名空间，在华为云Astro轻应用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。必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不能以下划线结尾。 | SelfHostDeepSeek |
| 模型名称 | 具体大模型的模型标识。在调用该连接器时，会将此字段传递给大模型API中的model字段。 | DeepSeek-V3-0324-w8a8 |
| URL | 大模型提供方提供的大模型API调用地址。 | https://***** |

步骤3 测试大模型连接器的连通性。

1.

在自定义大模型详情页面的“测试”中，单击“测试”，进入大模型连接器测试页面。
2.

在“输入内容”中，输入对话内容（例如：你好），单击“测试”，查看返回消息是否正确响应。
- 如果返回消息中有流式响应报文输出，说明大模型连接器配置成功。否则，请检查大模型连接器信息是否正确。

图 10-5 大模型连接器测试页面

测试

输入内容

你好

温度 ?

-

0.8

+

多样性 ?

-

1.0

+

重复惩罚 ?

-

0.0

+

返回消息

你好！很高兴见到你，有什么我可以帮忙的吗？

取消

测试

----结束

步骤二：调用流式报文接口构建知识库

创建一个标准页面，并拖入一个智能助手组件，通过脚本调用流式报文接口构建知识库。

- 步骤1 在应用设计器的左侧导航栏中，选择“界面”。
- 步骤2 单击页面后的“+”，创建一个空白的标准页面。

图 10-6 设置标准页面基本信息

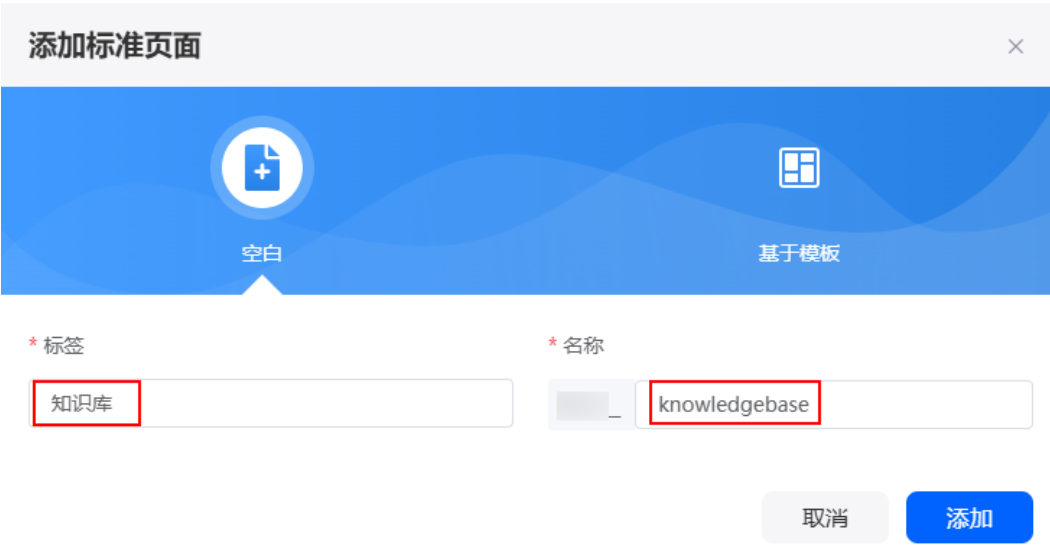


表 10-3 标准页面参数说明

| 参数 | 说明 | 示例 |
|----|---|-------------------|
| 标签 | 输入标准页面的标签名，用于在页面显示，创建后可修改。
取值范围：1~64个字符。 | 知识库 |
| 名称 | 输入标准页面的名称，名称是标准页面在系统中的唯一标识，创建后不可修改。命名要求如下： <ul style="list-style-type: none">长度不能超过64个字符，包括前缀命名空间的长度。名称前的内容为命名空间，在华为云Astro轻应用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。必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不能以下划线结尾。 | knowledg
ebase |

- 步骤3 在“基本组件 > 智能化”中，拖拽“智能助手”组件到右侧画布中。

图 10-7 添加智能助手组件



- 步骤4** 为智能助手组件添加开启知识库并绑定大数据模型。
- 选中智能助手组件，在“属性 > 高级属性”中，开启知识库。

图 10-8 开启知识库



- 开启大模型，并单击“设置默认模型”，选择**步骤一：创建大模型连接器对接外部大模型**中创建的连接器的完成模型的绑定。

图 10-9 为组件绑定模型



----结束

步骤三：验证智能助手是否可以正常构建知识库

- 步骤1

在已创建的标准页面中，单击, 进入预览页面，可查看到添加知识库按钮和绑定的大数据模型。
- 步骤2

单击“添加知识库”，进入知识库页面，再单击“知识库管理”。

团队知识库是供团队成员共同使用的知识库。创建团队知识库时，如果“可见范围”设置为“团队内可见”，且“选择团队”选择了该用户所在的部门，则仅该部门的用户能够查看与使用该知识库。

个人知识库主要由用户个人创建，也只能由用户个人使用，其余用户没有查看及使用该知识库的权限。
- 步骤3

单击“创建知识库”，设置知识库的名称和可见范围，单击“确定”，返回知识库页面，可查看到已创建的知识库。

图 10-10 新建知识库

创建知识库

* 知识库名称

常见问题知识库

* 可见范围

☐ 团队内可见

☒ 所有人可见

标签

请选择标签

描述

请输入

0/255

确定

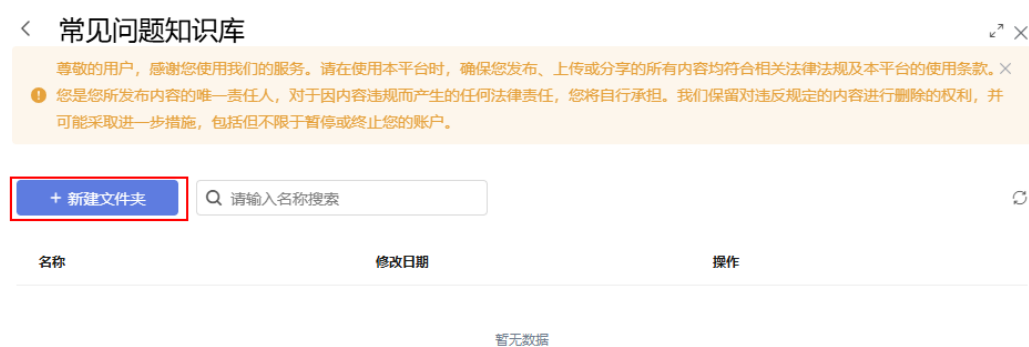
取消

表 10-4 创建知识库参数说明

| 参数 | 说明 | 示例 |
|-------|---|---------|
| 知识库名称 | 新建知识库的名称。 | 常见问题知识库 |
| 可见范围 | 选择知识库的可见范围，支持团队内可见和所有人可见。 <ul style="list-style-type: none">团队内可见：仅该部门的用户能够查看与使用该知识库。所有人可见：该租户下的所有用户都能查看并使用该知识库。 | 所有人可见 |

步骤4 在知识库列表中，单击**步骤3**中创建的知识库，进入知识库页面，单击“新建文件夹”。

图 10-11 单击新建文件夹



步骤5 设置文件夹名称，根据实际需求选择知识库Embedding模型、切分器和分段长度。

图 10-12 新建文件夹



表 10-5 新建文件夹参数说明

| 参数 | 说明 | 示例 |
|-------|-----------|--------|
| 文件夹名称 | 新建文件夹的名称。 | 高频常见问题 |

| 参数 | 说明 | 示例 |
|-------------|---|----------|
| Embedding模型 | <p>选择Embedding模型，按需进行选择。</p> <ul style="list-style-type: none"> 中文文本嵌入模型：该模型适用于中英文文本的通用型768维度的嵌入模型。 中文版文本表示模型：该模型适用于检索、分类、聚类或语义匹配等任务，目前仅支持中文文本。 向量计算模型：该模型适用于中英文文本，在文本相似度计算中表现优异，模型维度364。 华为自研代码专用模型：该模型在代码数据向量化、代码搜索任务中表现突出。 | 中文文本嵌入模型 |
| 切分器 | <p>切分器用于将文本转换为模型能够理解的单元，支持如下几种类型：</p> <ul style="list-style-type: none"> 基于段落的切分器（默认使用）：按照连续的两个换行符（\n\n）或文本段落结构进行分割。 基于行的切分器：按照换行符（\n）进行分割。 基于句子的切分器（适用于英文文本）：将文本按照英文句子进行分割。 基于字符的切分器：按照逐个字符（含空白字符）进行分割。 | 基于段落的切分器 |
| 分段预计长度 | 分片的最长长度，取值范围为[50, 1200]，默认值为1000。 | 1000 |
| 分段重叠长度 | 当前分片与上一个分片重叠的文本长度，取值范围“0”到分片的长度的字符，默认值为0。 | 0 |

| 参数 | 说明 | 示例 |
|------|--|----------|
| 子切分器 | <div>切片长度超过最大长度时应用子切分器，支持如下几种类型：</div> <ul style="list-style-type: none">• 基于字符的切分器（默认使用）：按照逐个字符（含空白字符）进行分割。• 基于句子的切分器（适用于英文文本）：将文本按照英文句子进行分割。• 基于单词的切分器（适用于英文文本）：将文本按照空白字符进行分割。 | 基于字符的切分器 |

步骤6 设置完成后，单击“确定”，返回知识库页面。

步骤7 单击已创建的文件夹（如高频常见问题），在文件夹中单击“上传文件”。

本实践以上传华为云Astro轻应用帮助中心的常见问题为例，单击[华为云Astro轻应用常见问题](#)可下载该文档。

图 10-13 单击上传文件



图 10-14 上传常见问题文件



文档上传后，文件“状态”为“入库中”。单击  按钮，刷新状态，当“状态”变为“入库成功”，说明文档可以正常使用。

步骤8 返回知识库页面，选择已创建的知识库，您可以选择一个或多个知识库。

图 10-15 选择知识库



步骤9 在智能助手中，提问华为云Astro轻应用常见问题中相关问题，可以检索到对应的文档。

----结束

10.2 通过华为云 Astro 轻应用调用大模型连接器实现智能助手聊天对话

应用场景

开发者使用华为云Astro轻应用开发标准页面时，如果需要通过智能助手组件对接外部大模型接口，可以先在大模型连接器中配置大模型对接信息，然后再在智能助手组件中选取对应的大模型连接器，即可实现智能助手对话应用。

本实践将为您介绍，如何配置大模型连接器，以及在智能助手组件中使用大模型连接器，实现智能助手聊天对话应用。

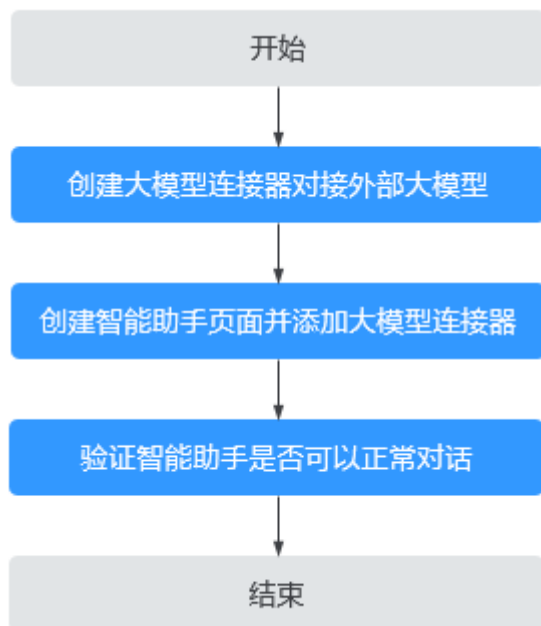
前提条件

用户需自行获取或搭建大模型资源，本实践以对接自建部署的大模型为例。

操作流程

在华为云Astro轻应用中，通过配置大模型连接器与智能助手组件实现对话应用的流程，如[图10-16](#)所示。

图 10-16 调用大模型连接器实现智能助手聊天对话操作流程



步骤一：创建大模型连接器对接外部大模型

在华为云Astro轻应用中，创建一个大型模型连接器用来对接外部大型模型接口。

步骤1 创建一个低代码应用。

1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。

3. 在“应用”中，单击“新建低代码应用”或单击 ，进入新建低代码应用页面。

首次创建应用时，请根据界面提示创建一个命名空间。命名空间一旦创建，不能修改和删除，创建前请确认好相关信息。建议使用公司或团队的缩写作为命名空间。命名空间只能由字母、数字和下划线组成，长度不能超过15个字符。并且只能以字母开头，以字母或数字结尾，同时不能包含连续的下划线。

4. 在新建低代码应用页面，应用类型选择“标准应用”，单击“确定”。
5. 输入应用的标签和名称，单击“新建”，即可进入应用设计器。

图 10-17 创建一个空白应用

新建 空白应用

基本信息

添加图标

* 标签

智能助手应用

* 名称

ChatApp

分类

请选择

描述

请输入

高级设置

取消

新建

表 10-6 新建空白应用参数说明

| 参数 | 说明 | 示例 |
|----|--|---------|
| 标签 | 新建应用的标签，长度不能超过80个字符。 | 智能助手应用 |
| 名称 | 新建应用的名称，输入标签值后单击该参数的输入框，系统会自动生成应用的名称，同时自动在名称前，添加命名空间。命名要求如下： <ul style="list-style-type: none">长度不能超过31个字符，包括前缀命名空间的长度。 名称前的内容为命名空间，在华为云Astro轻应用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。 <ul style="list-style-type: none">必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不允许以下划线结尾。 | ChatApp |

- 步骤2 创建大模型连接器。
1. 在应用设计器中，选择“集成 > 连接器 > 连接器实例”。

2. 选择“大模型 > 自定义连接器”，单击“+”，进入创建自定义大模型连接器页面。

图 10-18 新建自定义大模型连接器



3. 输入大模型连接器的配置信息，单击“保存”。

图 10-19 设置大模型连接器信息

创建自定义大模型

基本信息

* 标签

DeepSeek-V3

* 名称

SelfHostDeepSeek

* 模型名称

DeepSeek-V3-0324-w8a8

* URL

/v1/chat/completion

API Key

请输入

上下文长度

请输入

最大token长度

请输入

描述

请输入

取消

保存

表 10-7 自定义大模型连接器参数说明

| 参数 | 说明 | 示例 |
|------|--|-----------------------|
| 标签 | 自定义大模型连接器的标签，创建后可修改。
取值范围：1~64个字符。 | DeepSeek-V3 |
| 名称 | 自定义连接器的名称，名称是连接器在系统中的唯一标识，创建后不可修改。命名要求如下： <ul style="list-style-type: none">长度不能超过64个字符，包括前缀命名空间的长度。名称前的内容为命名空间，在华为云Astro轻应用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。必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不能以下划线结尾。 | SelfHostDeepSeek |
| 模型名称 | 具体大模型的模型标识。在调用该连接器时，会将此字段传递给大模型API中的model字段。 | DeepSeek-V3-0324-w8a8 |
| URL | 大模型提供方提供的大模型API调用地址。 | https://***** |

步骤3 测试大模型连接器的连通性。

1. 在自定义大模型详情页面的“测试”中，单击“测试”，进入大模型连接器测试页面。
2. 在“输入内容”中，输入对话内容（例如：你好），单击“测试”，查看返回消息是否正确响应。

如果返回消息中有流式响应报文输出，说明大模型连接器配置成功。否则，请检查大模型连接器信息是否正确。

图 10-20 大模型连接器测试页面

测试

×

输入内容

你好

温度 ?

—

0.8

+

多样性 ?

—

1.0

+

重复惩罚 ?

—

0.0

+

返回消息

你好！很高兴见到你，有什么我可以帮忙的吗？

取消

测试

步骤二：创建智能助手页面并添加大模型连接器

在智能助手模板应用中，创建一个标准页面，用于组装智能助手模板对话界面。

- 步骤1 在应用设计器的左侧导航栏中，选择“界面”。
- 步骤2 单击页面后的，设置标准页面的基本信息，单击“添加”。

图 10-21 新建标准页面

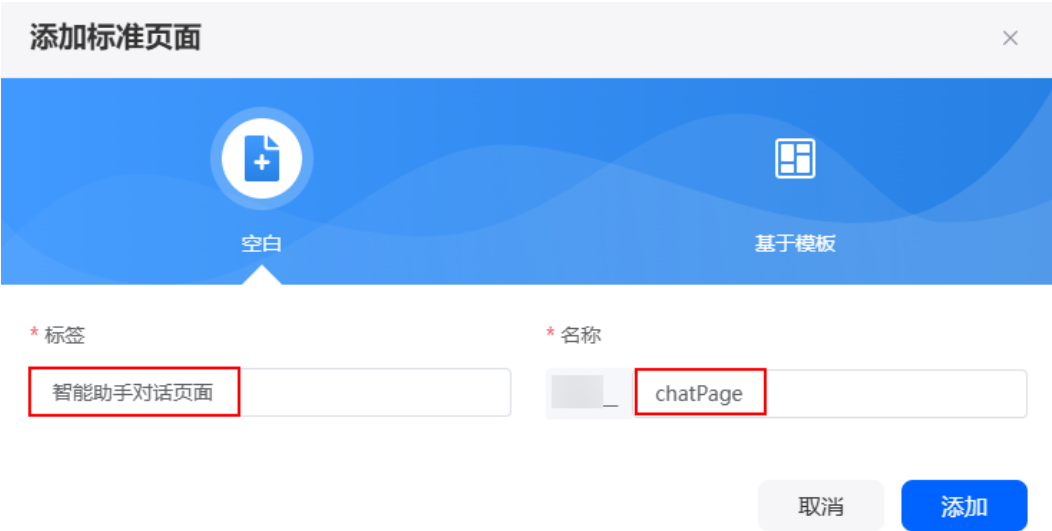
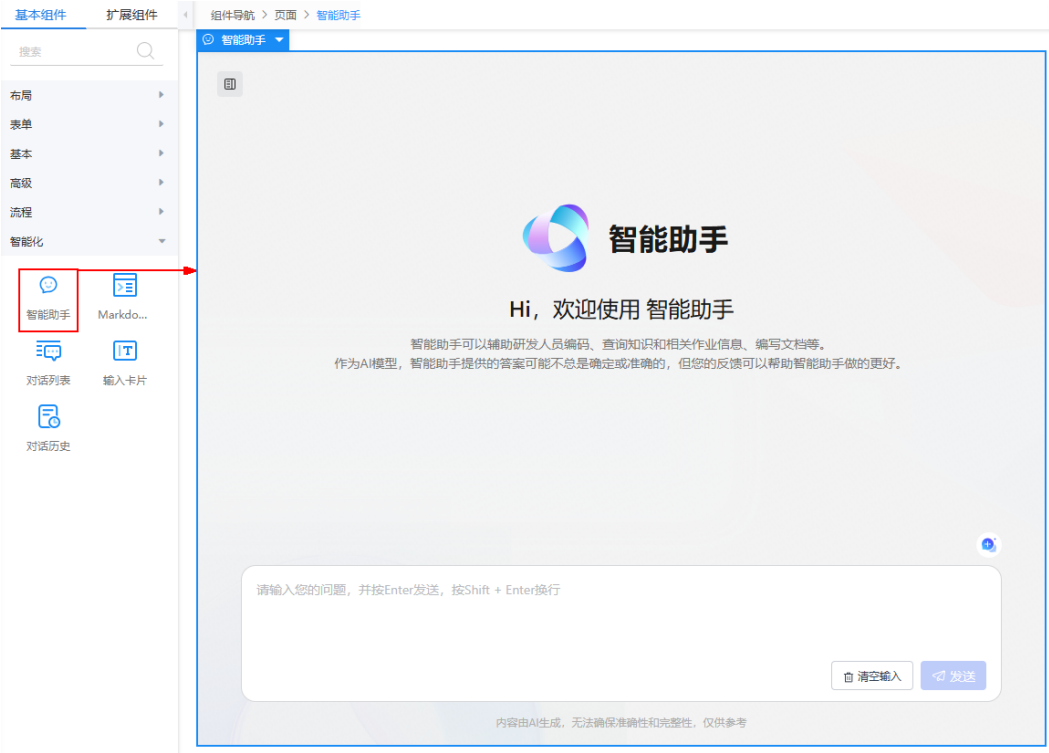


表 10-8 标准页面参数说明

| 参数 | 说明 | 示例 |
|----|--|----------|
| 标签 | 输入标准页面的标签名，用于在页面显示，创建后可修改。
取值范围：1~64个字符。 | 智能助手对话页面 |
| 名称 | 输入标准页面的名称，名称是标准页面在系统中的唯一标识，创建后不可修改。命名要求如下： <ul style="list-style-type: none">长度不能超过64个字符，包括前缀命名空间的长度。
名称前的内容为命名空间，在华为云 Astro轻应用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。必须以英文字母开头，只能由英文字母、数字或单下划线组成，且不能以下划线结尾。 | chatPage |

- 步骤3 在“基本组件 > 智能化”中，拖拽“智能助手”组件到右侧画布中。

图 10-22 拖拽智能助手组件至页面中



- 步骤4** 选中智能助手组件，在“属性 > 高级属性”中，打开“大模型”开关。
- 步骤5** 在“默认大模型”中，单击“设置默认模型”，选择**步骤一：创建大模型连接器对接外部大模型**中创建的大模型连接器。

图 10-23 智能助手高级配置



- 步骤6** 单击页面上方的，保存标准页面。

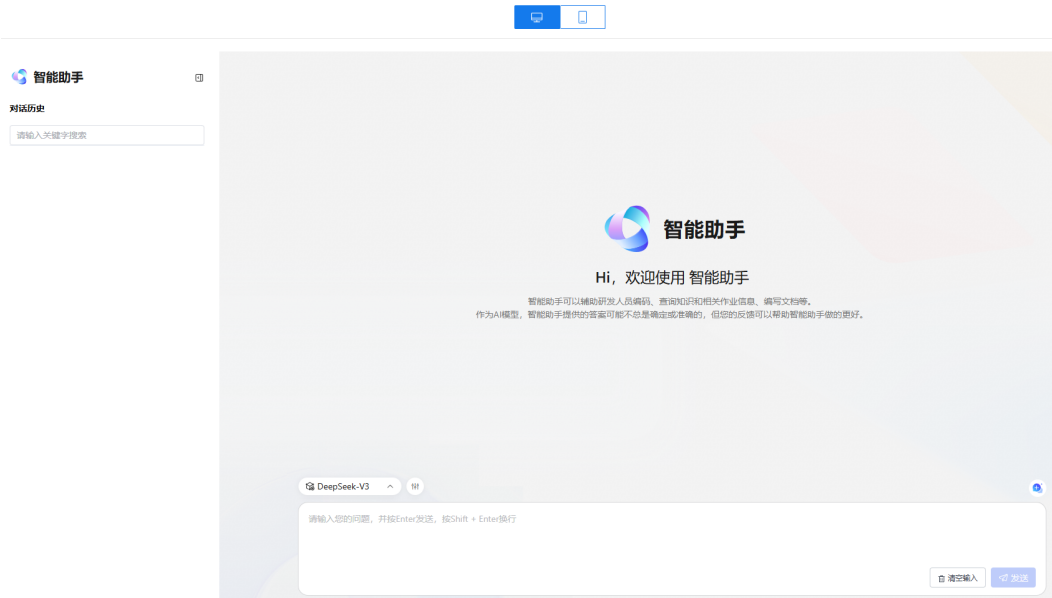
----结束

步骤三：验证智能助手对话功能

在智能助手模板应用中，预览智能助手页面，验证智能助手对话功能。

步骤1 在已保存的标准页面中，单击，进入预览页面。

图 10-24 智能助手预览页面



步骤2 在智能助手的聊天框中，输入文字，例如“你好，告诉我今天天气怎么样”，单击“发送”。

智能助手返回对应的流式结果，说明智能助手组件对接外部大模型接口成功。

----结束

11 零代码专项

11.1 如何设置选项关联

期望实现效果

在数据录入、数据查询和流程审批等类型应用中，经常会使用联动筛选、限定可选范围，确保数据逻辑一致，减少用户操作失误。例如，在点餐类的应用表单中，“饮品分类”选项有“咖啡”和“奶茶”，“具体饮品”选项有属于咖啡类的选项，如“拿铁”、“浓缩咖啡”和“焦糖玛奇朵”，也有属于奶茶类的选项，如“英式奶茶”、“港式奶茶”和“珍珠奶茶”。在华为云Astro轻应用的零代码平台通过设置“选项关联”，可以实现当“饮品分类”选择“咖啡”时，“具体饮品”只展示咖啡类的选项。

功能实现方法

步骤1 创建一个零代码应用。

1. 参考[授权用户使用华为云Astro轻应用并购买实例](#)中操作，申请华为云Astro轻应用免费试用或购买商业实例。
2. 实例购买后，在华为云Astro轻应用服务控制台的“主页”中，单击“进入首页”，进入应用开发页面。
3. 在顶部菜单栏中，单击“工作台”，进入华为云Astro轻应用零代码平台。
4. 在“全部应用”中，单击“开始创建”中的 ，选择“创建空白应用”。
5. 输入应用的标题（如点餐），单击“创建普通表单”，选择“创建空白表单”，进入点餐应用。

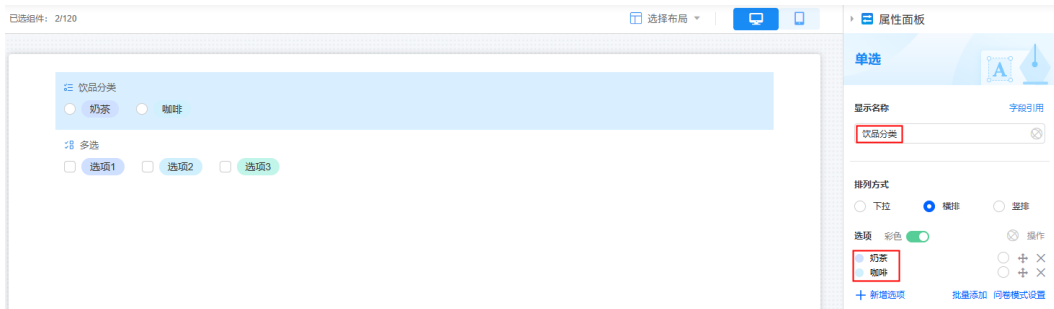
步骤2 修改表单的标题为“菜单”，删除多余的组件并从“数据组件”中拖拽一个多选按钮到画布中。

图 11-1 设计菜单表单



步骤3 选中单选按钮，修改显示名称和选项。

图 11-2 设置单选按钮



步骤4 设置多选按钮。

- 1. 选中多选按钮，修改显示名称为“具体饮品”。
- 2. 在选项中，单击“批量添加”，添加具体选项。

图 11-3 单击批量添加



3. 添加所需的选项，单击“确认”。

图 11-4 添加选项

批量添加

×

每行对应一个选项，将所有不重复项作为选项，单个选项的长度不超过85个字

拿铁

浓缩咖啡

焦糖玛奇朵

英式奶茶

港式奶茶

珍珠奶茶

取消

确认

步骤5 设置选项关联。

1. 单击“具体饮品”右侧的“问卷模式设置”。

图 11-5 单击问卷模式设置

已选组件：2/120

选择布局

属性面板

饮品分类

奶茶

咖啡

具体饮品

拿铁

浓缩咖啡

焦糖玛奇朵

英式奶茶

港式奶茶

珍珠奶茶

多选

显示名称

字段引用

具体饮品

排列方式

下拉

矩阵

选择

选项

彩色

操作

拿铁

浓缩咖啡

焦糖玛奇朵

英式奶茶

港式奶茶

珍珠奶茶

+

×

+

×

+

×

+

×

+

×

+

×

新增选项

批量添加

问卷模式设置

2. 开启“选项关联”，单击拿铁后的“添加关联”。

图 11-6 添加选项关联

问卷模式设置

×

提供问卷类表单的通用操作，如选项随机、选项互斥等能力。简单几步设置，玩转数据收集。

☐

选项展示顺序随机 开启后，该字段的选项在每次访问时出现的顺序是随机的

☐

赋分值 ? 开启后，请在下方选项内容表内设置分值或选中项

☐

选项互斥 ? 开启后，请在下方选项内容表内设置选项互斥关系

☒

选项关联 ? 开启后，请在下方选项内容表内设置选项关联

取消

确认

3. 选择关联字段的选项值，如关联“饮品分类”下的“咖啡”选项，单击“确定”。
- 当选择该选项时，当前选项“拿铁”会展示在界面上。

图 11-7 设置关联选项

选项关联

×

一个字段下的多个选项默认隐藏，不同选项的显示与否可与前面的某个字段选项选中与否相关联。

当前选项：拿铁

选择字段 1 饮品分类 ×

当“选择字段 1”选中下面选项中的任一选项

☐ 奶茶

☒ 咖啡

+ 添加条件

满足以上条件，拿铁 显示。

☐ 同步 ?

取消

确定

4. 在步骤5.3的页面中，可以通过“同步”功能，将其他选项关联到咖啡分类中。例如，将浓缩咖啡和焦糖玛奇朵关联到咖啡分类中。

图 11-8 关联其他选项

选项关联

✕

一个字段下的多个选项默认隐藏，不同选项的显示与否可与前面的某个字段选项选中与否相关联。

当前选项：拿铁

选择字段 1 饮品分类

✕

当“选择字段 1”选中下面选项中的任一选项

☐ 奶茶

☒ 咖啡

+ 添加条件

满足以上条件，拿铁 显示。

☒ 同步

?

同步全部

☐ 全选

☒ 浓缩咖啡

☒ 焦糖玛奇朵

☐ 英式奶茶

☐ 港式奶茶

浓缩咖啡

焦糖玛奇朵

取消

确定

同步成功后，返回问卷模式设置页面，此时“选项关联”已变成“修改”。单击浓缩咖啡后的“修改”，可查看到分类已修改为咖啡。

图 11-9 返回问卷模式设置

问卷模式设置

提供问卷类表单的通用操作，如选项随机、选项互斥等能力。简单几步设置，玩转数据收集。

选项展示顺序随机 开启后，该字段的选项在每次访问时出现的顺序是随机的

赋分值 开启后，请在上方选项内容表内设置分值或选中项

选项互斥 开启后，请在上方选项内容表内设置选项互斥关系

选项关联 开启后，请在上方选项内容表内设置选项关联

| 选项内容 | 设置分值 | 互斥 | 选项关联 |
|-------|------|--------------------------|------|
| 拿铁 | | ☐ | 修改 |
| 浓缩咖啡 | | ☐ | 修改 |
| 焦糖玛奇朵 | | ☐ | 修改 |
| 英氏奶茶 | | ☐ | 添加关联 |
| 港式奶茶 | | ☐ | 添加关联 |
| 珍珠奶茶 | | ☐ | 添加关联 |

取消

确认

图 11-10 查看修改后的分类

选项关联

一个字段下的多个选项默认隐藏，不同选项的显示与否可与前面的某个字段选项选中与否相关联。

当前选项：浓缩咖啡

选择字段 1 饮品分类

当“选择字段 1”选中下面选项中的任一选项

☐ 奶茶

☒ 咖啡

+ 添加条件

满足以上条件，浓缩咖啡 显示。

同步

取消

确定

5. 在问卷模式设置页面，单击港式奶茶后的“添加关联”，将分类设置为奶茶。

图 11-11 设置选项关联

选项关联

×

一个字段下的多个选项默认隐藏，不同选项的显示与否可与前面的某个字段选项选中与否相关联。

当前选项：英式奶茶

选择字段 1

饮品分类

×

当“选择字段 1”选中下面选项中的任一选项

☒ 奶茶

☐ 咖啡

+ 添加条件

满足以上条件，英式奶茶 显示。

☐ 同步 ?

取消

确定

6. 单击“同步”，将港式奶茶和珍珠奶茶添加到奶茶分类中。

图 11-12 设置港式奶茶和珍珠奶茶分类

选项关联

×

一个字段下的多个选项默认隐藏，不同选项的显示与否可与前面的某个字段选项选中与否相关联。

当前选项：英式奶茶

选择字段 1

饮品分类

×

当“选择字段 1”选中下面选项中的任一选项

☒ 奶茶

☐ 咖啡

☐ 拿铁

☐ 浓缩咖啡

☐ 焦糖玛奇朵

☒ 港式奶茶

☒ 珍珠奶茶

+ 添加条件

满足以上条件，英式奶茶 显示。

☒ 同步 ?

同步全部

港式奶茶

珍珠奶茶

取消

确定

7. 单击“确定”，返回问卷模式设置页面。
8. 在问卷模式设置页面，单击“确认”，返回表单设计页面。

步骤6 单击“保存”，并返回应用开发页面，分享表单，查看最终效果。

单击咖啡，显示图11-13，单击奶茶，显示图11-14，说明效果符合预期。

图 11-13 查看最终效果（咖啡）

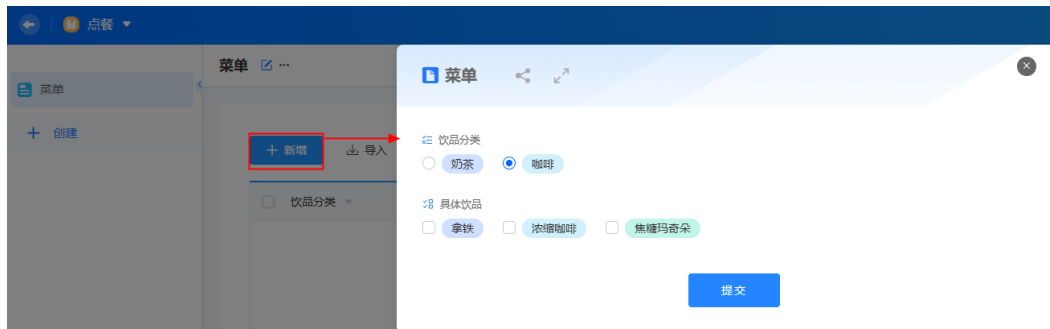


图 11-14 查看最终效果（奶茶）



----结束