

Astro 轻应用

产品介绍

文档版本 01
发布日期 2024-12-27



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

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

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

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

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

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目录

1 什么是 Astro 轻应用.....	1
2 产品优势.....	3
3 应用场景.....	4
4 产品规格差异.....	6
5 权限管理.....	9
6 约束与限制.....	12
7 基本概念.....	18

1 什么是 Astro 轻应用

Astro轻应用（Astro Zero，简称AstroZero）是零代码和低代码应用开发平台，源于华为应用开发和数字化转型的实践，提供了云上无码化、低码化、支持多码化的应用开发模式，屏蔽了技术的复杂性，提升了企业开发的效率。同时提供应用资产的开发标准和微服务框架，助力企业不断沉淀可复制的套件，加速应用的定制，并通过开放的生态，实现套件资产的商业变现。

通过AstroZero提供的界面、逻辑、对象等可视化编排工具，以“拖、拉、拽”的方式来快速构建应用。

为什么选择 AstroZero

- **以模型驱动，快速构建业务应用**
 - 模型驱动，支持数据导入导出，对象编排。
 - 关联一键生成页面，快速构建应用。
 - 适配不同业务场景的页面标准。
 - 支持标准页面自适应布局，高级页面适配多终端。
- **脚本引擎，在线低码开发复杂业务逻辑**
 - 支持自定义开发，定制更灵活
 - 现代化的编辑界面，语法高亮显示，代码联想提示。
 - 在线IDE，智能提示，自动补全，快速编译。
 - 支持页面调测和性能查看，快速定位逻辑Bug。
- **服务编排，搭积木式应用构建，积木零件组合复用**
 - 拖拉拽式编排流程，图形化展示服务逻辑。
 - 逻辑单元封装，通过图元形式集成到服务编排中，可重新组合。
 - 运行服务编排支持断点调试，日志跟踪，页面性能查看。
- **集成开发，灵活实现系统间对接**
 - 连接器&API开放，可以快速对接第三方系统。
 - 业务逻辑单元，沉淀领域业务功能，多场景复用。
 - 定制高级页面组件，沉淀行业资产。
 - 可配置触发器，自动执行逻辑任务。

更多选择理由，请参见[产品优势](#)和[应用场景](#)。

访问方式

您可以通过以下任何一种方式访问AstroZero。

- 管理控制台
基于浏览器的可视化界面，即控制台方式访问AstroZero，详情请参见[用户指南](#)。
- REST API
使用AstroZero提供的REST API接口以编程方式访问AstroZero，详情请参见[API参考](#)。

学习路径

您可以借助[成长地图](#)，快速了解产品，由浅入深学习使用和运维AstroZero。

2 产品优势

随着人工智能、物联网、区块链等领域飞速发展，传统行业在向互联网化转型的过程中，对软件技术的要求越来越高，软件实现越来越复杂，软件构建技术难度以指数级增长。AstroZero充分考虑了开发者和企业用户的痛点，提供了高效率、高效能的开发平台。

降低应用构建门槛

全云化线上开发，No Code/Low Code开发模式，多终端适配，所见即所得的应用构建，图形化业务编排，具备极低的学习成本，让业务人员可直接构建应用，人人都是开发者。

加速业务敏捷创新

拖拽式页面构建，简单易用的数据建模，灵活可视的服务编排、流程编排、事件编排，丰富的预置资产，积木式搭建应用效率提升10+倍，极大缩短了创新周期。

沉淀行业知识，快速复制

创新的业务资产沉淀机制，针对性沉淀行业资产，可快速在多个项目间复用和灵活扩展，简化传统交付流程，快速实现批量复制。

支撑大型企业级应用构建

完善的应用打包、部署、升级能力，高效的团队协同开发能力，支持多沙箱验证，大幅减少环境部署时间和成本，支撑企业级应用构建。

3 应用场景

AstroZero覆盖了低代码、零代码和行业应用等多种开发场景，满足行业客户、合作伙伴和开发者的各种需求。

轻应用构建

低代码提供了丰富的轻应用模板，包括办公管理、人事管理、项目管理、运营推广、培训赋能等领域，开发者可基于模板快速构建应用，加速企业数字化。

使用低代码构建轻应用具备如下优势：

- 高效页面编排
丰富的UI组件库，易用的拖拽式页面构建体验。
- 极简流程配置
极简式流程编排，流程配置百变随心。
- 多屏融合
支持Web版和Mobile版本，一次开发，多屏使用。

零代码应用构建

无需编码经验，在不需要编写代码的情况下，通过拖拽组件、设置属性、配置节点等来代替传统的代码实现环节，即可让想法变为现实，搭建出自己满意的应用。

使用零代码构建应用具备如下优势：

- 丰富的场景模板
提供多场景模板，即装即用，灵活改造，让业务人员迅速做出属于自己的满意应用。
- 轻松简单的操作界面
丰富的功能组件，搭配灵活随心，权限可控，团队协作更省心。
- 多设备支持
支持PC端、移动端、微信小程序/H5、WeLink集成、便捷实用。

行业应用构建

面向园区、城市、能源、教育和交通等行业，提供端到端全流程应用构建解决方案，加速全场景行业数字化。

使用低代码构建行业应用具备如下优势：

- 丰富的行业资产沉淀
全方位资产视图，积累应用资产、BO资产、组件资产100+，应用搭建灵活高效。
- 高效构建大型企业级应用
全流程可视化开发，自动化调测，项目级协助共享，端到端安装部署，助力大型企业级应用高效构建。
- 保护现有投资
现有业务资产以Native服务方式接入复用，与现网系统快速集成，投资一次，长期受益。

4 产品规格差异

AstroZero提供了免费版、标准版、专业版和专享版四种规格套餐，各套餐的功能说明如表1所示，具体计费详情请参见[AstroZero价格详情](#)。专业版和专享版支持回退到免费版，回退后不支持访问运行环境。

- 免费版：免费使用，最多可添加10个用户（包括业务用户），可以使用的资源有限，免费版提供开发环境和开发零代码应用的运行环境，不提供低代码应用的运行环境。
- 标准版：适用于零代码应用开发场景，开发者不需要有代码开发经验，企业自用首选。
- 专业版：专业版在标准版基础上提供了低代码应用运行环境（多租户共享运行环境），且提供了更多功能和资源，适用于专业开发者。
- 专享版：使用专享版前，请确认已[提交工单](#)申请开通专享版，否则，您将无法购买专享版。专享版提供了物理隔离的运行环境，运行环境实例发放到租户虚拟私有云中。

须知

- 应用正式发布上线前，需要在测试环境上进行测试，AstroZero使用沙箱环境作为测试环境，四种规格套餐均支持购买沙箱。
- [表4-1](#)中的“不限制”，是在系统允许的范围内不进行限制。若想了解更多，请参考[AstroZeroLicense](#)。

表 4-1 套餐规格说明

功能特性		免费版	标准版	专业版	专享版
用户数	业务用户数	总数默认最多10个，不支持扩展用户。	总数默认最多30个，支持扩展用户。	总数默认最多30个，支持扩展用户。	• 500 • 2000 • 5000
	开发者账号数				
	扩展用户/新增用户	不支持	9元/用户/月	39元/用户/月	不支持

功能特性		免费版	标准版	专业版	专享版
零代码应用构建	内部分享	支持	支持	支持	支持
	公开分享	支持	支持	支持	支持
	附件上传	不支持	支持	支持	支持
低代码应用开发	企业应用开发	支持	支持	支持	支持
	资产开发	支持	支持	支持	支持
	低代码应用数 (除了BO和Native Service, 其他所有类型的定制应用数)	10	50	不限	不限
	物联接入	不支持	不支持	支持	支持
	原生服务接入	不支持	不支持	支持	支持
	发布移动应用	不支持	不支持	支持	支持
应用运行	运行零代码应用	支持	支持	支持	支持
	零代码应用数	100	不限	不限	不限
	数据量(租户级)	1万条(不支持扩容)	4000条/用户	4000条/用户(支持单独扩容)	<ul style="list-style-type: none"> 500人: 200G 2000人: 800G 5000人: 2T
	运行低代码应用	不支持	不支持	支持	支持
	低代码应用数	不支持	不支持	不限	不限
	物联接入调用次数(租户级)	不支持	不支持	2500次/用户/天	次数不限制
环境管理	租户子域名	支持	支持	支持	支持
	开发环境	支持	支持	支持	支持
	商用运行环境	不支持	共享环境(多租户共享运行环境)	共享环境(多租户共享运行环境)	独享运行环境、物理隔离
	企业独立域名	不支持	不支持	支持	支持, 可独立域名

功能特性		免费版	标准版	专业版	专享版
	调测沙箱	不支持	不支持	按需购买沙箱	按需购买沙箱
	租户级日志服务	不支持	不支持	支持	支持

5 权限管理

如果需要对AstroZero的资源，给企业中的员工设置不同的访问权限，以达到不同员工之间的权限隔离，可以使用统一身份认证服务（Identity and Access Management，简称IAM）进行精细的权限管理。该服务提供了用户身份认证、权限分配、访问控制等功能，可以帮助您安全的控制华为云资源的访问。

通过IAM，可以在华为云账号中给员工创建IAM用户，并使用策略来控制用户对华为云资源的访问范围。例如，员工中有负责软件开发的人员，希望该类员工只拥有AstroZero的使用权限，但是不希望拥有删除AstroZero等高危操作的权限，那么可以使用IAM为开发人员创建用户，通过授予仅能使用AstroZero，但是不允许删除AstroZero的权限策略，控制他们对AstroZero资源的使用范围。

如果华为云账号已能满足您的要求，不需要创建独立的IAM用户进行权限管理，可以跳过本章节，不影响您使用AstroZero服务的其它功能。

IAM是华为云提供权限管理的基础服务，无需付费即可使用，只需要为您账号中的资源进行付费。关于IAM的详细介绍，请参见《IAM产品介绍》。

AstroZero 权限

默认情况下，新建的IAM用户没有任何权限，需要将其加入用户组，并给用户组授予策略或角色，才能使得用户组中的用户获得对应的权限，这一过程称为授权。授权后，用户就可以基于被授予的权限对云服务进行操作。

AstroZero部署时通过物理区域划分，为项目级服务。授权时，“作用范围”需要选择“区域级项目”，然后在指定区域（如华北-北京4）对应的项目（cn-north-4）中设置相关权限，并且该权限仅对此项目生效。如果在“所有项目”中设置权限，则该权限在所有区域项目中都生效。访问AstroZero时，需要先切换至授权区域。

根据授权精细程度分为角色和策略。

- 角色：IAM最初提供的一种根据用户的工作职能，定义权限的粗粒度授权机制。该机制以服务为粒度，提供有限的服务相关角色用于授权。由于华为云各服务之间存在业务依赖关系，因此给用户授予角色时，可能需要一并授予依赖的其他角色，才能正确完成业务。角色并不能满足用户对精细化授权的要求，无法完全达到企业对权限最小化的安全管控要求。
- 策略：IAM最新提供的一种细粒度授权的能力，可以精确到具体服务的操作、资源以及请求条件等。基于策略的授权是一种更加灵活的授权方式，能够满足企业对权限最小化的安全管控要求。

如表5-1所示，包括了AstroZero的所有系统权限。

表 5-1 AstroZero 系统权限

系统角色/策略名称	描述	类别	依赖关系
Astro Zero Instance ManageAccess	Astro轻应用实例管理权限，拥有订购、退订、查看和升级AstroZero实例的权限。	系统策略	无
Astro Zero Instance ViewAccess	Astro轻应用实例查看权限，只拥有查看AstroZero实例的权限，不可进行退订、升级等操作。 若IAM用户登录AstroZero服务控制台，查看不到AstroZero实例，处理方式有以下两种（任选其一即可）： <ul style="list-style-type: none"> 给IAM用户所在用户组，添加“Astro Zero Instance ViewAccess”权限。 不要将IAM用户，添加到任何用户组中。 	系统策略	无
Astro Zero IAM User QueryAccess	Astro轻应用IAM用户查询权限，只有华为云账号或具有“Astro Zero IAM User QueryAccess”权限的IAM用户，才能执行创建AstroZero开发者账号的操作。	系统策略	无

表5-2列出了AstroZero常用操作与系统策略的授权关系，您可以参照该表选择合适的系统策略。

表 5-2 AstroZero 常用操作与系统策略之间的关系

操作	Astro Zero Instance ManageAccess	Astro Zero Instance ViewAccess	Astro Zero IAM User QueryAccess
查看Astro轻应用实例列表及详情	√	√	x
订购Astro轻应用实例	√	x	x
退订Astro轻应用实例	√	x	x
变更Astro轻应用实例规格	√	x	x

操作	Astro Zero Instance ManageAccess	Astro Zero Instance ViewAccess	Astro Zero IAM User QueryAccess
修改Astro轻应用实例详情	√	x	x
创建Astro轻应用开发者账号	x	x	√ 除了具备该权限外，还需要具备AstroZero本身的权限，即在权限配置文件Profile中，还需勾选“管理用户&用户权限”。
查询Astro轻应用IAM用户	x	x	√

除了IAM提供的认证和授权功能，AstroZero本身也提供了用户管理和权限配置（Profile）功能，来管理用户（包括业务用户），控制用户、业务用户等的操作权限，具体说明可参考[AstroZero用户概述](#)。

6 约束与限制

AstroZero 实例

表 6-1 AstroZero 实例规格约束与限制

限制项	约束与限制
实例版本	<p>AstroZero目前支持的实例版本有：免费版、标准版、专业版和专享版。</p> <ul style="list-style-type: none">• 免费版：仅提供开发环境和开发零代码应用的运行环境，不提供低代码应用的运行环境。• 标准版：适用于开发零代码应用，开发者不需要有任何代码开发经验。• 专业版：适用于专业开发者，提供了开发环境和多租户共享的运行环境。• 专享版：需要先提交工单申请开通专享版，才可以正常购买。专享版提供了开发环境和物理隔离的运行环境，即运行环境实例发放到租户虚拟私有云中。• 申请AstroZero免费试用或购买AstroZero商用实例时，会同步创建一个Astro workflow免费基础版实例。
用户数	<ul style="list-style-type: none">• 免费版：10个（包括业务用户），不支持扩展用户。• 标准版：默认为30个（包括业务用户），支持扩展用户。• 专业版：默认为30个（包括业务用户），支持扩展用户。• 专享版：提供了500个、2000个和5000个用户数（包括业务用户）供选择。

变更实例

表 6-2 变更实例约束与限制

限制项	约束与限制
变更AstroZero实例规格/类型	<ul style="list-style-type: none"> 建议在业务低峰期变更实例，业务高峰期变更实例可能会失败。 免费版实例不支持升级规格。 标准版可以升级到专业版。 标准版、专业版无法升级到专享版，只能单独购买。 标准版、专业版可以直接回退到免费版，回退后不会删除数据，但是不支持访问运行环境。

命名空间

表 6-3 命名空间约束与限制

限制项	约束与限制
命名空间个数	一个租户只能创建一个命名空间，且创建后不支持修改和删除。
命名空间名称	<ul style="list-style-type: none"> 长度不能超过15个字符。 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，不能包含两条连续的下划线，且不允许以下划线结尾。

应用

表 6-4 应用约束与限制

限制项	约束与限制
应用类型	在AstroZero中，支持创建标准应用、扩展应用、BO和原生服务四种类型的应用。 更多信息，请参见 使用AstroZero创建应用概述 。
应用名称	<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。 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。

脚本

表 6-5 脚本约束及限制

限制项	约束与限制
脚本命名	<ul style="list-style-type: none"> 脚本采用小驼峰命名，例如createDeviceInstance。 结构体（struct）采用大驼峰命名，例如QueryPaymentResult。 结构体内的字段采用小驼峰命名，例如customerName。 类、枚举值和接口采用大驼峰命名。 函数采用小驼峰命名。 属性或变量采用小驼峰命名。
脚本中注释	<ul style="list-style-type: none"> 无用的代码不能以注释形式存在。 能用代码说明的尽量不要添加注释，脚本注释尽可能简洁。 建议注释统一用英文。 出入参不必写注释说明。 <p>更多信息，请参见AstroZero脚本开发规则及约束限制。</p>
脚本中引用	<ul style="list-style-type: none"> 脚本中不要包含没用到的标准库或对象的引用。 只引用使用到的对象，而不用import *。 @action.method属于必须的方法注解，写在方法上面。 为方便页面直接调用脚本，仅在脚本最后统一导出需要的对象，而不是导出所有对象。如果不需要，可以不用导出。 可以把公共的对象Object定义，按分类放在一些公共的脚本里（例如pc_XXX.ts、cm_XXX.ts），其他需要的脚本直接引用即可。
脚本定义	<ul style="list-style-type: none"> 每个字段的定义，均需要定义type、label、description、required和isCollection，有默认值的非必填。 当字段为集合类型时，需要定义成“[]”。 <pre>@action.param({ type: "Attribute", label: "Attribute", description: "attributeList", required:false isCollection: true }) attributeList: Attribute[];</pre> @action.object时，需要在脚本中详细定义清楚Object，不要引用其他脚本的Object。 如果是嵌套结构体，则从下到上粒度依次变小。 <p>更多信息，请参见AstroZero脚本开发规则及约束限制。</p>

限制项	约束与限制
脚本中SQL	<ul style="list-style-type: none"> ● 不推荐用拼接SQL方法，避免注入风险。 ● 多表复杂查询建议用“sql.exec()”或“sql.execute()”方法，“execute()”方法比“exec()”多返回字段集和操作成功数。 ● 对于单表查询和增删改SQL，推荐使用Orm接口方法。 ● 避免在循环中调用方法和操作数据库，可以用“in”来查询在集中的结果。 ● 对sql进行优化时，尽量使用有索引的字段，避免使用没有索引的字段。 ● 批量操作数据库时，尽量使用已封装好的批量操作接口。 ● 匹配查询推荐用like，日期比较推荐用“<”、“>”。 ● 使用exists替代in，使用not exists替代not in。 ● 避免在索引列上使用is null和is not null，会造成索引失效。
脚本调用	<ul style="list-style-type: none"> ● 不允许在应用的脚本中，调用BO中的脚本。 例如，设备管理应用的脚本，不可调用设备BO内的任何脚本。 ● 不允许跨BO，调用脚本。 例如，人员BO中的脚本，不允许调用设备BO内的任何脚本。

服务编排

表 6-6 服务编排约束与限制

限制项	约束与限制
服务编排命名	<ul style="list-style-type: none"> ● 驼峰命名，首字母小写，不要携带下划线。 ● 动宾结构，例如，notifyOrderCompletion，错误样例OrderCompletionNotification。 ● 尽量不用缩写，除非是专有名词，例如，invokeCRM，错误样例paymentCbk。 ● 应简单明了，表示业务意义，而不是内部实现。例如createPayment，错误样例InsertAndUpdatePayData。 ● 长度不能超过64个字符，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。

限制项	约束与限制
变量命名	<ul style="list-style-type: none"> ● 输入变量和输出变量：命名遵循接口设计文档的要求。 ● 内部变量：元素命名遵循驼峰命名原则。 ● 变量为单数时，命名包含对象名称（如Level1Catalog、Level2Catalog），变量为复数时命名包含集合名称（如Level1CatalogList或者Level2CatalogArray）。 ● 数组型变量的下标命名要符合规范，当使用I、J、K等字母来命名时，需要明确变量含义，勿重复使用。
图元命名	<p>采用动词和名词短语的形式，每个单词首字母大写，单词之间有空格。名称一般不要超过五个单词，短语尽可能简明，描述该图元的主要作用，例如Valid Input、Call Device Service。</p>
连接线命名	<p>在服务编排的图元中，除了Decision连接线外，其他图元的连接线采用系统的固定命名。</p> <p>Decision连接线名称使用英文，采用单词首字母大写，一般不要超过三个单词。采用动词和名词短语的形式，尽可能简明、准确的描述该条件的判断逻辑。</p>
图元编排	<p>为了服务编排画布排版美观以及方便后续的服务编排检视，服务编排的图元编排遵循以下原则：</p> <ul style="list-style-type: none"> ● 在配置服务编排前，需明确具体步骤和子流程的分解，合理编排图元。 ● 业务逻辑采用自上而下、从左往右的页面布局方式，业务逻辑展示清晰，同一任务多个步骤，横向排列，不同任务之间竖向排列。利用横向和竖向，做到层次缩进。 ● Decision图元里的Default改为表示主流程，类似于If Error Else MainProcess这种结构，把异常处理优先标出。 ● 在同一个版面中，图元之间的间隔大小相同。 ● 尽量避免连接线相互交叉。 ● 创建或查询较复杂的父子对象结构场景（如创建订单和相关对象），应将父对象和典型子对象的创建和查询编排到子流程中，供多个流程共享。 ● 对于服务编排中常用到的业务功能（如获取Offer实例），可以考虑编排到子流程或用脚本实现，供多个流程共享。 ● 编排每个服务编排时，首先都应该设置入参校验步骤，图元类型为“Decision”，分支优先考虑异常场景。服务编排流程中，其他的“Decision”图元也要首先考虑异常场景。 ● 避免使用循环套循环。 ● 不允许在子流程中结束，所有结束出口应在最外层服务编排中体现。子流程中，应始终返回出参。

限制项	约束与限制
参数定义	<ul style="list-style-type: none"> ● 服务编排的入参和出参，需要根据设计文档做必填参数校验。 ● 出入参数需要定义清楚每个字段。如果存在结构体，结构体中也应定义清楚每个字段。除了本身是预留的定制扩展的结构体之外，不允许有空结构的结构体。 ● 服务编排中，暴露给用户由用户输入的参数，必须放到服务编排的“入参”中。服务编排中，不能存在用户能够使用但不在“入参”及“出参”的参数。
调用约束	<ul style="list-style-type: none"> ● 不允许在应用项目的服务编排中，调用BO的内部服务编排。 ● 不允许在应用项目的服务编排中，编排调用BO的脚本。 ● 不允许跨BO调用脚本和内部服务编排。 例如，人员BO的服务编排不允许调用空间BO的内部服务编排和脚本。

应用发布与部署

表 6-7 应用发布与部署限制

限制项	约束与限制
应用安装包	<ul style="list-style-type: none"> ● 资产包：资产包一般用于测试、部署或正式运行，支持在开发环境（不能安装开发者自己开发的资产包）、沙箱环境以及运行环境中安装使用。 ● 安装包：适用于初次和大的变更打包。 ● 补丁包：适用于问题修复类打包变更。 ● 源码包：主要用于同一账号内，在不同环境之间的资产迁移，以源环境的状态继续开发（类似自己代码仓库），或者用于备份账号自己的代码，以便在需要恢复时使用。

7 基本概念

使用AstroZero开发应用前，建议您先了解相关的基本概念。

开发环境

开发环境是开发者专门用于开发应用的环境，用户可根据需求自行开发配置各种逻辑模块化元素，从而创建各类应用。开发环境默认域名为“appcube.cn-north-4.huaweicloud.com”。

沙箱环境

在应用正式发布上线前，需要在测试环境进行测试，包含数据配置。AstroZero一般以沙箱（Sandbox）环境，来作为测试环境。沙箱环境默认域名为“appcubesandbox.cn-north-4.huaweicloud.com”。

使用沙箱环境前，首先需要了解什么是沙箱。沙箱是一种安全机制，为运行中的程序提供的隔离环境。通常作为一些来源不可信、具破坏力或无法判定程序意图的程序，提供实验之用。沙箱是一种按照安全策略，限制程序行为的执行环境。在沙箱中，网络访问、对真实系统的访问、对输入设备的读取，通常被禁止或是严格限制的。沙箱属于虚拟化的一种，沙箱中的所有改动对操作系统不会造成任何损失。关于沙箱的更多介绍，请参见[购买AstroZero沙箱实例](#)。

运行环境

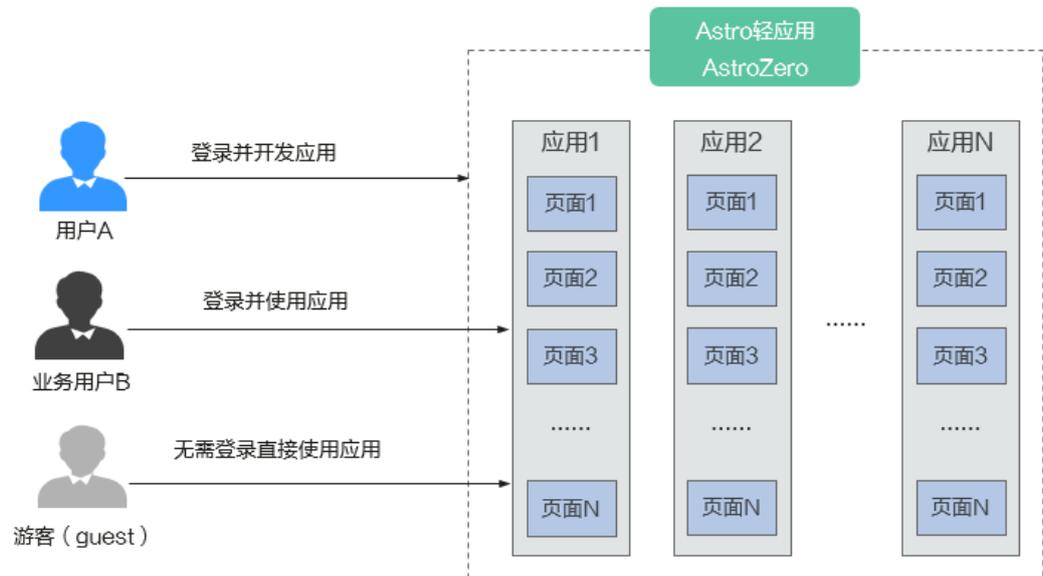
运行环境是指正式发布上线的真实环境。测试完成后，将应用发布至生产环境，即运行环境，供业务用户使用。运行环境默认域名为“appcuberun.cn-north-4.huaweicloud.com”。

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

AstroZero 中的用户

AstroZero中存在两类账号：用户（User）和业务用户（PortalUser）。用户为应用的开发者，业务用户为应用的使用者，用户开发的应用，游客（Guest）可直接使用。关于用户和业务用户的更多介绍，请参见[图解AstroZero中用户那些事](#)。

图 7-1 AstroZero 中用户行为



命名空间

由于不同租户间的数据可打包共享，为了避免数据命名的重复（例如数据对象、应用、服务编排等），租户在首次创建应用前，需要先定义一个命名空间。

命名空间唯一且每个租户只能创建一个命名空间。命名空间一旦设定不能修改，请慎重定义，建议使用公司或团队的缩写作为命名空间。命名空间创建后，后续创建的应用、数据对象、脚本、服务编排、页面等应用元素的名称前，系统会为其增加命名空间。例如，将命名空间定义为“CNAME”，后续创建的所有应用、应用元素（数据对象、脚本、页面等）都将以“CNAME_”开头。

标准页面

标准页面基于Vue技术栈，平台预置丰富基础组件，也可以支持扩展，多用于表格表单等后端管理页面的快速开发。

标准页面开发界面提供图形化、无码化在线页面开发功能，通过标准页面开发界面能够快速构建各种复杂表单表格页面、以及其他一些灵活布局页面。支持在开发界面中将组件面板上的页面组件拖拽至页面工作区域，并对组件的属性、事件进行设置，再配合事件编排完成复杂的业务功能。支持用户以积木组装的方式快速构建应用页面，提升开发效率和质量，及时响应业务需求和价值实现。更多内容请参见[标准页面](#)。

高级页面

高级页面是由一个或者多个组件拼装而成。高级页面可以使用Widget组件进行组装和配置，方便用户达到自己想要的效果。高级页面支持多种框架，支持代码开发，比较自由，多用于对页面效果要求高的场景。更多内容请参见[高级页面](#)。

工作流

工作流即业务流程管理，源自业界工作流N 2.0标准。低代码平台基于该规范实现了自己的业务流程管理系统，由于工作流本身已成为了业界一套行业规范，因此在低代码平台中称实现此类流程的引擎为工作流。更多内容请参见[工作流](#)。

对象

AstroZero中的对象（也可以称为Object）相当于传统方式开发业务系统时，数据库中创建一个表。每个对象对应一张数据库表，用于保存业务系统需要的配置数据和业务数据。更多内容请参见[对象](#)。

服务编排

服务编排是一种通过简单的拖拉拽式流程编排以及参数配置的方式来进行服务开发的能力，并支持对已开发的服务重新进行组合编排。用户能够在服务编排编辑器内以图形化编排的形式快速地进行服务的开发并扩展出更丰富的业务功能，同时能够与API接口进行绑定，以API的形式对外提供服务。更多内容请参见[服务编排](#)。

应用模板

为了降低应用开发成本，让用户通过拖拽的方式零代码或者低代码快速完成应用的搭建，AstroZero提供了丰富的应用模板。应用模板涵盖了人事管理、办公管理、问卷调查、项目管理，以及通用应用等领域，用户可以直接使用应用模板快速创建应用，基于应用模板创建应用后，用户还可快速定制和扩展应用，满足自身业务的个性化诉求。更多内容请参见[应用模板](#)。

连接器

连接器是AstroZero提供的，调用第三方服务的集成工具。AstroZero封装了不同类型的连接器对接相应的第三方服务，如OBS、MINIO、Redis和云搜索等。通过连接器，开发者无需关注具体代码实现，只需要配置一下服务地址和鉴权信息，即可在服务编排或脚本中快速集成第三方系统。更多内容请参见[连接器](#)。