

Astro 轻应用

API 参考

文档版本 01
发布日期 2024-04-25



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 使用前必读.....	1
1.1 概述.....	1
1.2 调用说明.....	1
1.3 终端节点.....	1
1.4 约束限制.....	2
1.5 基本概念.....	2
2 API 概览.....	4
3 如何调用 API.....	8
3.1 构造请求.....	8
3.2 认证鉴权.....	12
3.3 返回结果.....	25
4 快速入门.....	28
4.1 创建对象并定义字段.....	28
4.2 获取对象列表.....	30
5 API.....	33
5.1 数据模块.....	33
5.1.1 按记录 ID 查询数据.....	33
5.1.2 按记录 ID 删除数据.....	35
5.1.3 按记录 ID 更新数据.....	36
5.1.4 按记录 ID 列表查询数据.....	37
5.1.5 新增对象数据.....	39
5.1.6 按记录 ID 列表删除数据.....	41
5.1.7 按条件更新数据.....	42
5.1.8 按条件查询对象数据.....	44
5.1.9 按条件删除对象数据.....	49
5.1.10 按条件查询对象记录数.....	51
5.1.11 批量插入或修改对象数据.....	53
5.2 元数据.....	55
5.2.1 操作对象.....	55
5.2.2 操作对象实例.....	56
5.2.3 Object 对象的操作.....	57
5.2.3.1 创建对象.....	57

5.2.3.2 查询所有 Object 对象.....	59
5.2.3.3 按照 object_id 查询 Object.....	61
5.2.3.4 根据 object_id 为 Object 添加字段.....	63
5.2.3.5 根据 object_id 新增 trigger.....	66
5.2.3.6 根据 object_id 新增校验 validator.....	68
5.2.3.7 根据 object_id 新增 layout.....	69
5.2.4 Flow 对象的操作.....	71
5.2.4.1 创建 Flow 对象.....	71
5.2.4.2 查询 Flow 对象.....	73
5.2.4.3 按照 flow_id 查询/查询 flow_id 的入参出参信息.....	75
5.2.4.4 根据 flow_id 修改 Flow 对象.....	77
5.2.4.5 按照 flow_id 删除 flow.....	79
5.2.5 BPM 对象的操作.....	80
5.2.5.1 创建 BPM 对象.....	81
5.2.5.2 按条件查询 BPM 对象.....	87
5.2.5.3 按 ID 查询 BPM 对象.....	88
5.2.5.4 根据 ID 修改 BPM 对象.....	94
5.2.5.5 按照 ID 删除 BPM.....	100
5.2.6 Picklist 对象的操作.....	101
5.2.6.1 创建 Picklist 对象.....	101
5.2.6.2 查询选项列表值集.....	103
5.2.6.3 按 ID 增加选项列表值集的值.....	105
5.2.6.4 根据 ID 查询选项列表值集或引用该选项列表值集的对象字段列表.....	106
5.2.6.5 根据 ID 修改选项列表值集的基本信息.....	108
5.2.6.6 根据 ID 删除选项列表值集或选项列表值集的值.....	110
5.2.7 Application 对象的操作.....	111
5.2.7.1 列出所有应用.....	111
5.2.7.2 根据 ID 查询应用.....	113
5.3 服务编排.....	116
5.3.1 编排 Flow.....	116
5.3.2 查看服务编排运行实例.....	118
5.4 脚本.....	120
5.4.1 运行脚本.....	120
5.4.2 调试脚本.....	121
5.4.3 获取脚本签名信息.....	123
5.4.4 获取脚本内置库信息.....	126
5.4.5 获取 Flow 公式信息.....	127
5.5 BPM.....	128
5.5.1 启动 BPM 实例.....	128
5.5.2 查询 BPM 实例.....	130
5.5.3 挂起/恢复/终止 BPM 实例.....	133
5.5.4 删除 BPM 实例.....	135

5.5.5 配置或修改 BPM 实例中变量的值.....	136
5.6 导入导出.....	137
5.6.1 导出数据.....	137
5.6.2 导入数据.....	141
5.6.3 导出元数据.....	142
5.6.4 导入元数据.....	144
5.6.5 使用模板导入数据.....	145
5.7 连接器.....	146
5.7.1 上传文件.....	146
5.7.2 下载文件.....	150
5.7.3 删除文件.....	151
5.7.4 删除目录.....	152
5.7.5 列举目录下文件.....	154
5.7.6 分段上传.....	156
5.7.7 获取 endpoint, 即文件存储地址的前缀.....	162
5.7.8 分享 Token.....	164
6 脚本中预置的 API.....	167
7 标准页面中预置的 API.....	169
8 高级页面和 Astro 大屏应用中预置的 API.....	189
9 权限管理.....	196
10 附录.....	198
10.1 状态码.....	198
10.2 错误码.....	200
A 修订记录.....	203

1 使用前必读

1.1 概述

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，从而实现所见即所得的快速应用开发和构建。

您可以使用本文提供的API，在AstroZero上对数据、元数据、访问控制、脚本、服务编排等进行相关操作，如运行服务编排、终止（删除）服务编排实例等，支持的全部操作可参见[API概览](#)。

在调用Astro轻应用API前，请确保已充分了解Astro轻应用相关概念，详细信息请参见[产品介绍](#)。

1.2 调用说明

AstroZero提供了REST（Representational State Transfer）风格API，支持您通过HTTPS请求调用，调用方法请参见[如何调用API](#)。

1.3 终端节点

终端节点即调用API的请求地址，AstroZero当前仅支持在**华北-北京四**的环境使用，AstroZero服务的终端节点开发态默认域名为“appcube.cn-north-4.huaweicloud.com”。

说明

本文档所有示例和描述中，使用的AstroZero服务开发环境域名，均以“appcube.cn-north-4.huaweicloud.com”为例。实际使用过程中，请以业务的实际情况为准。

1.4 约束限制

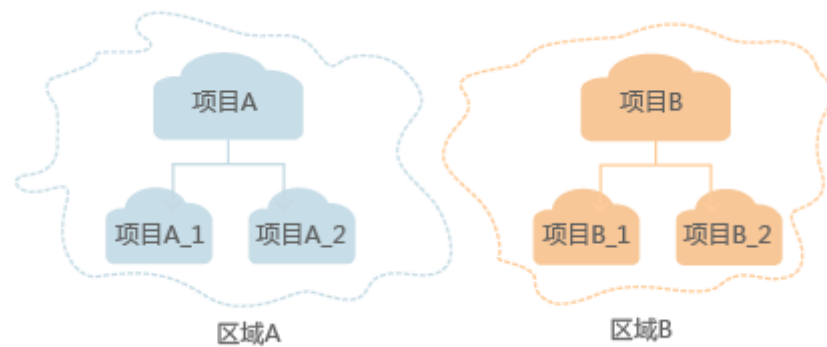
使用Restful API接口调用脚本时，脚本执行时长超过3分钟，会报超时错误。

1.5 基本概念

使用AstroZero API涉及的常用概念。

- 账号
用户注册账号时，账号对其所拥有的资源及云服务具有完全的访问权限，可以重置用户密码、分配用户权限等。由于账号是付费主体，为了确保账号安全，建议您不要直接使用账号进行日常管理工作，而是创建用户并使用他们进行日常工作。
- 用户
由账号在IAM中创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。在[我的凭证](#)下，您可以查看账号ID和用户ID。通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。
在AstroZero中，用户访问AstroZero来开发应用、管理配置应用和业务用户。
- 业务用户
业务用户是指访问在AstroZero中开发的一个业务应用的用户账号。例如，设备维修管理业务应用由某用户A开发，使用该业务应用的客服人员、派单员及维修人员都是业务用户，用户A是用户，不是业务用户。
- 区域
区域是指云资源所在的物理位置，同一区域内可用区间内网互通，不同区域间内网不互通。通过在不同地区创建云资源，可以将应用程序设计的更接近特定客户的要求，或满足不同地区的法律或其他要求。
您可以从[地区和终端节点](#)中，查询服务所在的区域。
- 可用区（AZ，Availability Zone）
一个AZ是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。
- 项目
区域默认对应一个项目，这个项目由系统预置，用来隔离物理区域间的资源（计算资源、存储资源和网络资源），以默认项目为单位进行授权，用户可以访问您账号中该区域的所有资源。如果您希望进行更加精细的权限控制，可以在区域默认的项目中创建子项目，并在子项目中创建资源，然后以子项目为单位进行授权，使得用户仅能访问特定子项目中资源，使得资源的权限控制更加精确。

图 1-1 项目隔离模型



在[我的凭证](#)下，您可以查看项目ID。

- 企业项目

企业项目是项目的升级版，针对企业不同项目间资源的分组和管理，是逻辑隔离。企业项目中可以包含多个区域的资源，且项目中的资源可以迁入迁出。

关于企业项目ID的获取及企业项目特性的详细信息，请参见《[企业管理服务用户指南](#)》。

2 API 概览

数据模块

接口	说明
按记录ID查询数据	按记录ID查询数据。
按记录ID删除数据	按记录ID删除数据。
按记录ID更新数据	按记录ID更新数据。
按记录ID列表查询数据	按记录ID列表查询数据。
新增对象数据	新增对象数据。
按记录ID列表删除数据	按记录ID列表删除数据。
按条件更新数据	按条件更新数据。
按条件查询对象数据	按条件查询对象数据。
按条件删除对象数据	按条件删除对象数据。
按条件查询对象记录数	按条件查询对象记录数。
批量插入或修改对象数据	批量插入或修改对象数据，若数据包含id进行更新，若数据不含id进行插入。

元数据

接口	说明
创建对象	添加一个Object对象。
查询所有Object对象	查询所有Object对象。
按照object_id查询Object	按照ID查询Object。
根据object_id为Object添加字段	为object添加自定义字段。

接口	说明
根据object_id新增trigger	根据object_id新增trigger。
根据object_id新增校验validator	根据object_id新增校验validator。
根据object_id新增layout	按object_id新增layout。
创建Flow对象	添加一个flow对象。
查询Flow对象	获取匹配查询条件的flow列表。
按照flow_id查询/查询flow_id的入参出参信息	按flow_id查询Flow/查询flow_id的入参出参信息。
根据flow_id修改Flow对象	根据id修改flow对象。
按照flow_id删除flow	按flow_id删除Flow。
创建BPM对象	创建一个BPM对象。
按条件查询BPM对象	获取匹配查询条件的BPM列表。
按ID查询BPM对象	按bpm_id查询BPM对象。
根据ID修改BPM对象	根据id修改BPM对象。
按照ID删除BPM	按bpm_id删除BPM。
创建Picklist对象	创建一个Picklist对象。
查询选项列表值集	查询选项列表值集。
按ID增加选项列表值集的值	按picklist_id增加选项列表值集的值。
根据ID查询选项列表值集或引用该选项列表值集的对象字段列表	根据ID查询选项列表值集或引用该选项列表值集的对象字段列表。
根据ID修改选项列表值集的基本信息	根据id修改Picklist对象。
根据ID删除选项列表值集或选项列表值集的值	根据ID删除选项列表值集或选项列表值集的值。

服务编排

接口	说明
编排Flow	编排某个服务编排。
查看服务编排运行实例	查询服务编排信息。

脚本

接口	说明
运行脚本	运行脚本。
调试脚本	调试脚本。
获取脚本签名信息	获取脚本签名信息。
获取脚本内置库信息	获取脚本内置库的声明。
获取Flow公式信息	获取服务编排的公式。

BPM

接口	说明
启动BPM实例	创建并启动一个BPM实例。
查询BPM实例	获取当前账号所有的BPM实例信息。
挂起/恢复/终止BPM实例	挂起/恢复/终止一个BPM实例。BPM实例被挂起后，不会在用户的待办任务列表中出现，流程SLA定时器停止，也不会进行任何跟催。恢复后，可继续进行操作。终止操作则强行结束一个BPM实例，可在历史任务中查看。
删除BPM实例	删除一个BPM实例。彻底删除一个BPM实例，仅当管理员清除过期数据时可以删除，其他情况不建议删除BPM实例数据。
配置或修改BPM实例中变量的值	设置或修改BPM实例中变量的值，一次可以修改一个、多个或者所有变量的值。不存在的变量或者只读变量会被忽略。

导入导出

接口	说明
导出数据	导出数据表中的具体数据。AstroZero中创建的对象，其内部的数据是支持导出的，该接口支持将对象中的数据按照设定的周期频率、筛选条件进行导出。
导入数据	导入数据。导入的数据源为zip包，包中的数据文件应为txt或csv格式，每个文件第一行为数据所属对象信息，第二行为字段名，第三行开始每一行为一条数据记录。
导出元数据	导出数据表的元数据，即将数据表的各个字段信息导出，但不包含具体的数据记录。

接口	说明
导入元数据	导入数据表的元数据。若已有数据表的元数据，可使用该功能快速根据元数据创建对象表。
使用模板导入数据	按照所使用的模板格式，导入数据。使用该功能前，需要创建好可使用的数据导入模板，在模板中定义数据的映射规则，调用接口时需指定模板名称，在该模板中，需要指定对应的操作脚本。

连接器

接口	说明
上传文件	将本地文件上传到存储中。使用该API上传文件，默认只会上传到连接器下配置的第一个桶中。
下载文件	从存储中下载文件。
删除文件	从存储中删除无用的文件。
删除目录	从存储中删除某目录。如果目录下不为空，删除目录的同时也会删除该目录下文件。
列举目录下文件	列举存储中某目录下文件。
分段上传	对于较大文件上传，可以切分成段上传。用户可以在如下的应用场景内（但不仅限于此），使用分段上传的模式： <ul style="list-style-type: none"> 上传超过100MB大小的文件。 网络条件较差，和OBS等存储服务端之间的链接经常断开。 上传前无法确定将要上传文件的大小。
获取endpoint，即文件存储地址的前缀	endpoint为文件存储地址的前缀，该前缀加上上传对象的路径，是实际访问对象的全路径。 上传文件 中，返回的是上传文件在桶上的路径。通过调用本节接口，返回endpoint，组装起来才是文件完整的访问地址。
分享Token	通过调用 获取分享Token ，可以获取临时Token，用户只要使用该临时Token即可访问文件，不用再使用“access-token”进行鉴权。

3 如何调用 API

3.1 构造请求

本节介绍REST API请求的组成，并以使用OAuth 2.0协议的客户端鉴权模式获取用户access_token为例说明如何调用API，该API获取access_token，access_token可以用于调用其他API时鉴权。

请求 URI

请求URI由如下部分组成：

{URI-scheme} :// {Endpoint} / {resource-path} ? {query-string}

尽管请求URI包含在请求消息头中，但大多数语言或框架都要求您从请求消息中单独传递它，所以在此单独强调。

表 3-1 URI 中的参数说明

参数	说明
URI-scheme	表示用于传输请求的协议，当前所有API均采用HTTPS协议。
Endpoint	指定承载REST服务端点的AstroZero服务器域名或IP。 例如，AstroZero开发环境服务器在“华北-北京四”区域的Endpoint为“appcube.cn-north-4.huaweicloud.com”。
resource-path	资源路径，即API访问路径。从具体API的URI模块获取，例如“获取用户Token”API的resource-path为“/baas/auth/v1.0/oauth2/token”。
query-string	查询参数，是可选部分，并不是每个API都有查询参数。查询参数前面需要带一个“？”，形式为“参数名=参数取值”，例如“limit=10”，表示查询不超过10条数据。多个查询参数之间使用“&”隔开。

例如，使用OAuth 2.0协议的客户端鉴权模式，获取用户access_token，“/baas/auth/v1.0/oauth2/token”为资源路径，假设AstroZero服务开发态域名为“appcube.cn-north-4.huaweicloud.com”，URI拼接起来如下所示。

```
https://appcube.cn-north-4.huaweicloud.com/baas/auth/v1.0/oauth2/token
```

请求方法

HTTP请求方法（也称为操作或动词），它告诉服务你正在请求什么类型的操作。

表 3-2 HTTP 方法

方法	说明
GET	请求服务器返回指定资源，如获取对象列表。
PUT	请求服务器更新指定资源，如按记录ID更新数据。
POST	请求服务器新增资源或执行特殊操作，如新增对象数据。
DELETE	请求服务器删除指定资源，如按记录ID列表删除对象数据。
HEAD	请求服务器资源头部。
PATCH	请求服务器更新资源的部分内容。当资源不存在的时候，PATCH可能会去创建一个新的资源。

在使用OAuth 2.0的客户端鉴权模式，获取用户access_token的URI部分，您可以看到其请求方法为“POST”，则其请求为：

```
POST https://appcube.cn-north-4.huaweicloud.com/baas/auth/v1.0/oauth2/token
```

请求消息头

附加请求头字段，如指定的URI和HTTP方法所要求的字段。例如定义消息体类型的请求头“Content-Type”，请求鉴权信息等。

详细的公共请求消息头字段请参见[表3-3](#)。

表 3-3 公共请求消息头

消息头名称	描述	是否必选
Content-Type	HTTP协议中设定的一个参数，用于标识返回的内容用什么格式去解析。 <ul style="list-style-type: none"> 配置为“application/json”。表示浏览器将返回内容解析为json对象。 配置为“application/x-www-form-urlencoded”，表示urlencode格式。 	是
tenant-id	账号的标识，不能重复，用于定义哪个账号接收请求。	否

消息头名称	描述	是否必选
access-token	access-token是调用AstroZero接口API要用到的访问令牌，在调用API的时候将access-token加到请求消息头，从而通过身份认证，获得操作API的权限。 当AstroZero的用户和业务用户通过账号密码登录AstroZero后，后台会生成一个访问令牌，在响应消息中携带，并存在浏览器中。下次再调用请求后，会带上这个访问令牌，表示这个用户已经被鉴权过。当第三方系统要访问AstroZero的业务接口时，需要提前使用OAuth协议进行接入认证，获取客户端鉴权ID（Client ID）和客户端鉴权密钥（Client Secret），再使用客户端鉴权ID和客户端鉴权密钥调用接口“/baas/auth/v1.0/oauth2/token”获取access_token，从而获取AstroZero业务接口的访问令牌。	否 使用access-token认证时该字段必选。

对于使用OAuth 2.0的客户端鉴权模式，获取用户access_token接口，由于不需要access-token认证，所以只添加“Content-Type”为“application/x-www-form-urlencoded”即可，添加消息头后的请求如下所示。

```
POST https://appcube.cn-north-4.huaweicloud.com/baas/auth/v1.0/oauth2/token
Content-Type: application/x-www-form-urlencoded
```

请求消息体（可选）

请求消息体通常以结构化格式（如JSON或XML）发出，与请求消息头中Content-type对应，传递除请求消息头之外的内容。若请求消息体中参数支持中文，则中文字符必须为UTF-8编码。

每个接口的请求消息体内容不同，也并不是每个接口都需要有请求消息体（或者说消息体为空），GET、DELETE操作类型的接口就不需要消息体，消息体具体内容需要根据具体接口而定。

对于使用OAuth 2.0的客户端鉴权模式，获取的用户access_token接口，您可以从[表 3-4](#)看到所需的请求参数及参数说明。

表 3-4 请求参数说明

参数名称	类型	必选（M）/可选（O）	参数位置	参数含义
grant_type	String	M	Body	授权模式，OAuth2.0中的grant_type字段的取值。 例如：client_credentials（即客户端模式）

参数名称	类型	必选 (M) / 可选 (O)	参数位置	参数含义
client_id	String	M	Body	<p>客户端ID, 获取方法如下:</p> <ol style="list-style-type: none"> 1. 登录承载REST服务端点的AstroZero环境。 2. 在经典版开发环境中, 单击主菜单中的“管理”, 进入管理页面。 3. 选择“系统管理 > OAuth”, 单击“新建”。 4. 输入OAuth名称, 设置授权类型为“客户端模式”, 选择一个用户 (当鉴权成功后, 将获得到和此用户相同的权限。注意不要具有选择匿名用户权限 “Anonymous_User_Profile” 的用户, 因为该权限不能访问 API)。并单击“保存”。 5. 在OAuth管理列表页, 单击具体OAuth所在行, 下载密钥文件到本地, 从中获取客户端鉴权ID “client_id” 取值。
client_secret	String	M	Body	<p>客户端密钥。 参考client_id, 获取客户端鉴权密钥 “client_secret” 取值。</p>
redirect_url	String	O	Body	重定向URL。
locale	String	O	Body	<p>语言。 例如: en_US</p>

将消息体加入后的请求如下所示, 加粗的斜体字段需要根据实际值填写。

```
POST https://appcube.cn-north-4.huaweicloud.com/baas/auth/v1.0/oauth2/token
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&client_id=*****&client_secret=*****
```


发起请求

到这里为止这个请求需要的内容就具备齐全了，您可以使用curl、Postman或直接编写代码等方式发送请求调用API。

对于使用OAuth 2.0的客户端鉴权模式，获取的用户access_token接口。返回的响应消息中，“access_token”就是需要获取的用户access_token。有了access_token之后，就可以使用access_token调用AstroZero的其他API。

3.2 认证鉴权

AstroZero提供的所有接口都需要通过认证之后才可以访问。认证方式主要分以下两类：

- 在请求消息头上，设置“access-token”
- 在请求消息头上，设置“Authorization”

两类方式都是通过不同的认证接口，来获取到Token。Token在计算机系统中代表令牌（临时）的意思，拥有Token代表拥有某种权限。Token认证是在调用API的时候将Token加到请求消息头，从而通过身份认证，获得操作API的权限。

平台用户分以下几类，不同类别的用户访问接口的权限会有所不同：

- 平台用户 (User)：一般对应运行环境、开发环境用户，用于管理开发应用。
使用的认证机制为：“账号密码登录认证”、“单点登录认证”和“OAuth 2.0鉴权登录认证”。
- 业务用户 (Portal User)：业务用户是访问AstroZero提供的业务服务的用户。
使用的认证机制为：“账号密码登录认证”、“单点登录认证”和“OAuth 2.0鉴权登录认证”。
- 匿名用户 (Guest)：对于该类用户，平台接口都需要进行校验，未登录前使用Guest用户身份来访问平台接口，一般会赋予最小的权限。
使用的认证机制为：“OAuth 2.0鉴权登录认证”。

获取Token的认证接口，如表3-5所示。

表 3-5 认证接口说明

接口	说明	生成Token的用户	是否可以 直接访问 接口
账号密码登录	通过账号密码直接访问登录接口，成功校验则返回“access-token”，写到Cookies上。	平台用户、业务用户	是

接口	说明	生成Token的用户	是否可以 直接访问 接口
单点登录（“CAS”协议）	支持CAS单点登录方式，通过CAS server登录后携带ticket重定向到AstroZero。AstroZero后台拿ticket到CAS Server校验ticket合法性，得到ticket对应的账号（可选）、用户信息，然后校验该用户是否存在，在指定了账号的情况下，如果该用户不存在，则创建对应用户；没有指定租户的情况下该用户不存在则直接报错，并返回到CAS Server。 成功验证用户身份后，则生成Session写到cookie上，重定向到指定页面。	平台用户、业务用户	是
使用OAuth 2.0的客户端（即client_credentials）鉴权模式	根据鉴权ID和鉴权密钥，调用获取用户Token的接口，来获取access_token。匿名用户一般通过该方式来获取Token。 具体操作，请参见 通过“客户端模式”获取的access-toke... 。	平台用户、匿名用户	是
使用OAuth 2.0的授权码（即authorization_code）鉴权模式	根据鉴权ID、鉴权密钥、授权码调用获取用户Token接口来获取access_token。后续需要通过access_token获取授权用户信息。 具体操作，请参见 通过“授权码模式”获取的access-toke... 。	平台用户、匿名用户	是
供华为WeLink单点登录	AstroZero开发的轻应用或者We码支持WeLink登录，在WeLink打开轻应用或者We码可以实现免登访问，详情请参见 发布企业WeLink小程序和发布WeLink轻应用 。	业务用户	否
业务用户登录（服务编排中PORTALUSERLOGIN公式）	指定业务用户名，使用服务编排的“PORTALUSERLOGIN”公式来登录。	业务用户	否（只能在服务编排中访问接口）

接口	说明	生成Token的用户	是否可以 直接访问 接口
业务用户通过 RefreshToken 获取新的Token	<p>可以通过配置内置系统参数 “bingo.service.refreshtoken.enable” 为 “是” 的方式来选择开启 RefreshToken 特性。开启后，业务用户通过服务编排的 “PORTALUSERLOGIN” 公式登录 AstroZero 时，会在 Cookies 中多返回一个 “refresh-token” 参数。可再通过发起请求 “POST https://AstroZero域名/baas/auth/v1.0/refresh-token” 来获取一个新的Token（调该接口时需要在请求消息头上设置参数 “Content-Type” 值为 “application/json”，请求消息体中设置参数 “grant_type” 值为 “refresh_token”，“refresh_token” 值为 Cookies 中返回的 “refresh-token” 参数值）。原有的Token不管有没有失效，都会被置成失效状态。一般情况下，“refresh-token” 的有效期限会比较长，在用户的权限配置中可通过配置 “刷新凭证时长” 来控制其有效期。</p> <p>具体操作，请参见业务用户通过 RefreshToken 获取新的Token。</p>	业务用户	是
JWT (JSON Web Tokens) 鉴权方式	<p>根据其他方式生成的 “access-token”，可以通过 jwt 接口换成一个 JWT 形式的 token。一般用于 AstroZero 内部服务之前的相互访问。内部服务会在 etcd 上配置一个 “SecretKey”，内部网元读取 “SecretKey” 来解 JWT 得到用户信息。</p>	平台用户	是
供CAS Server使用的账号密码登录方式	<p>AstroZero 自身的CAS Server没有独立的持久化，使用的Authcenter模块的用户体系，由Authcenter提供一个接口来做用户鉴权。</p> <pre>sequenceDiagram participant CAS Server participant Authcenter</pre> <p>CAS Server ->> Authcenter: 通过账号密码访问调用 Authcenter 认证接口 Authcenter -->> CAS Server: 返回认证结果，成功则包含用户账号信息 CAS Server ->> CAS Server: 生成 TCG CAS Server ->> Authcenter: 携带ticket 重定向到 Authcenter</p>	平台用户	否

OAuth 鉴权

AstroZero平台提供了两种授权模式进行OAuth鉴权，分别是“客户端模式”和“授权码模式”，两种授权模式适用的鉴权范围有所不同。

- 通过“客户端模式”获取的access-token，可用于调用API接口时进行鉴权，使用时在请求消息头上设置“access-token”。
 - 参考[如何登录经典版环境配置](#)中操作，登录经典版环境配置。
 - 在左侧导航栏中，选择“系统管理 > OAuth”菜单，新建OAuth管理。

图 3-1 新建客户端模式 OAuth 管理



⚠ 注意


- 第三方通过OAuth认证接入系统后，将以选择的用户身份操作数据，所以需要确保选择的用户具有调用API的相关权限。
 - 选择的用户的权限不能为匿名用户权限“Anonymous_User_Profile”，因为Guest用户没有访问API的权限，因此无法配置OAuth。
- 在OAuth管理列表中，单击该鉴权所在行的 ，下载密钥文件。

表 3-6 密钥文件参数

参数	参数说明
username	当“授权类型”配置为“客户端模式”时，密钥文件会包含该参数，为创建OAuth时选择的用户名。
client_id	鉴权的客户端ID。
client_secret	鉴权密钥。

- 根据鉴权客户端ID和鉴权密钥，通过Post请求调用接口“https://AstroZero域名/baas/auth/v1.0/oauth2/token”，获取access_token。

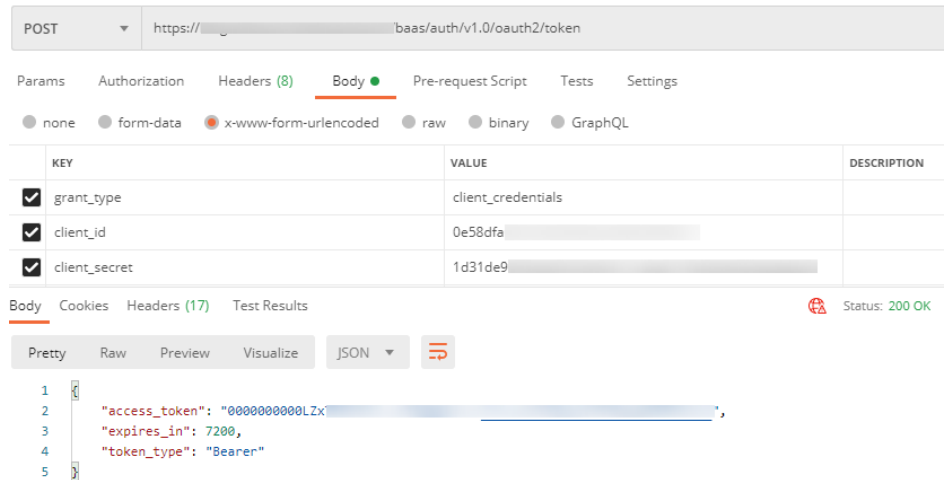
表 3-7 请求参数说明

参数名	类型	必选 (M) / 可选 (O)	参数位置	参数含义	值
grant_type	String	M	Body	授权模式，OAuth2.0中的grant_type字段的取值。	client_credentials (即客户端模式)
client_id	String	M	Body	鉴权的客户端ID，可在OAuth查看。	从上一步骤获取。
client_secret	String	M	Body	鉴权密钥，可在OAuth查看。 说明 URL查询参数中，请勿配置client_secret，建议将其配置到请求体中。	从上一步骤获取。
redirect_url	String	O	Body	重定向URL。	http://www.example.com
locale	String	O	Body	语言。	en_US

表 3-8 返回参数说明

参数	参数说明
access_token	认证后可信任的Token凭证。
expires_in	当前access_token的有效期，单位：秒。
token_type	access_token类型。

图 3-2 接口调用结果



说明

当接口调用出错时，会返回错误码“400”及错误信息说明，错误响应的Body体格式如下所示。

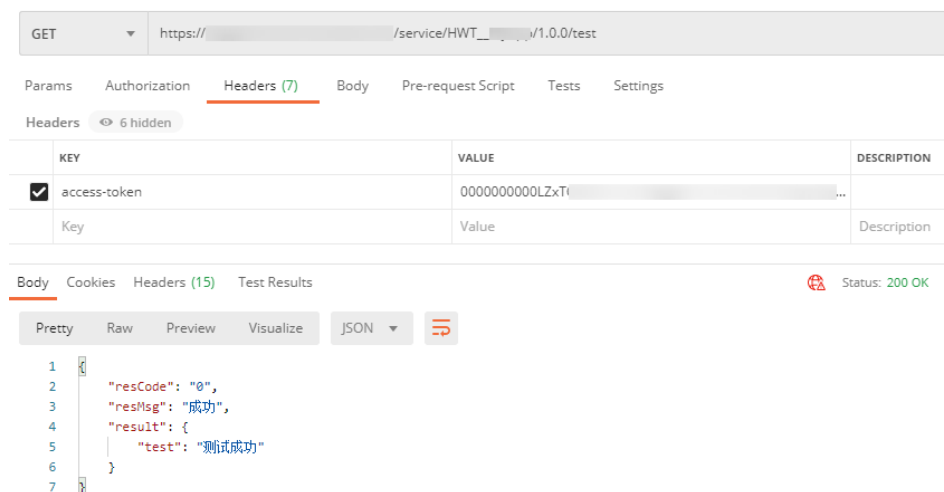
```

{
  "error": "unauthorized_client",
  "error_description": "The client is not authorized to request a token using this method."
}
```

其中，error表示错误提示，error_description表示错误描述信息。

- e. 将Headers中的access-token参数，设置为上一步得到的access_token，成功调用自定义的API接口。

图 3-3 成功调通 API 接口



说明

返回消息体中包含三种参数，“resCode”和“resMsg”参数代表此次请求的状态码及请求信息，您可以通过自定义错误码信息来设置请求错误时的返回信息，具体参考[错误码](#)。“result”参数为调用的接口返回的数据，与查看API信息时接口“返回消息”内容一致。

- 通过“授权码模式”获取的access-token，可用于获取用户信息时进行鉴权，使用时在请求消息头上设置“Authorization”。
 - 参考[如何登录经典版环境配置](#)中操作，登录经典版环境配置。
 - 在左侧导航栏中，选择“系统管理 > OAuth”菜单，新建OAuth管理。

图 3-4 新建授权码模式 OAuth 管理

说明

重定向地址将作为第三方系统同意授权后的跳转页面，并在页面地址上携带授权码信息。


- 在OAuth管理列表中，单击该鉴权所在行的 ，下载密钥文件。

表 3-9 密钥文件参数

参数	参数说明
redirect_uri	当“授权类型”配置为“授权码模式”时，密钥文件会包含该参数，为上一步设置的重定向地址。
client_id	鉴权的客户端ID。
client_secret	鉴权密钥。

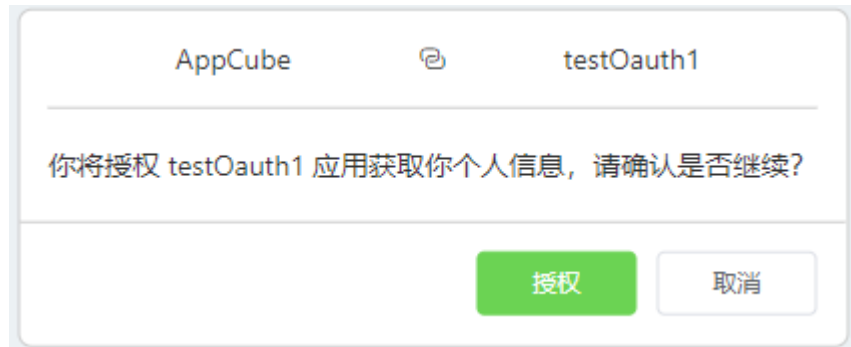
- 根据鉴权ID和重定向地址，在浏览器上访问接口“https://AstroZero域名/baas/auth/v1.0/oauth2/authorize”，获取授权码信息。

表 3-10 请求参数说明

参数名	类型	必选 (M) / 可选 (O)	参数位置	参数含义	值
response_type	String	M	Params	授权接口返回类型，即返回授权码	code
client_id	String	M	Params	客户端鉴权ID，可在OAuth查看。	从上一步骤获取。

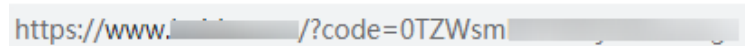
参数名	类型	必选 (M) / 可选 (O)	参数位置	参数含义	值
redirect_url	String	M	Params	重定向地址。	从上一步骤获取。

图 3-5 页面效果



- e. 单击“授权”按钮，跳转到重定向地址，并携带授权码信息。

图 3-6 授权码信息



- f. 根据授权码code，根据鉴权ID、鉴权密钥通过Post请求调用接口“https://AstroZero域名/baas/auth/v1.0/oauth2/token”，获取access_token。

表 3-11 请求参数说明

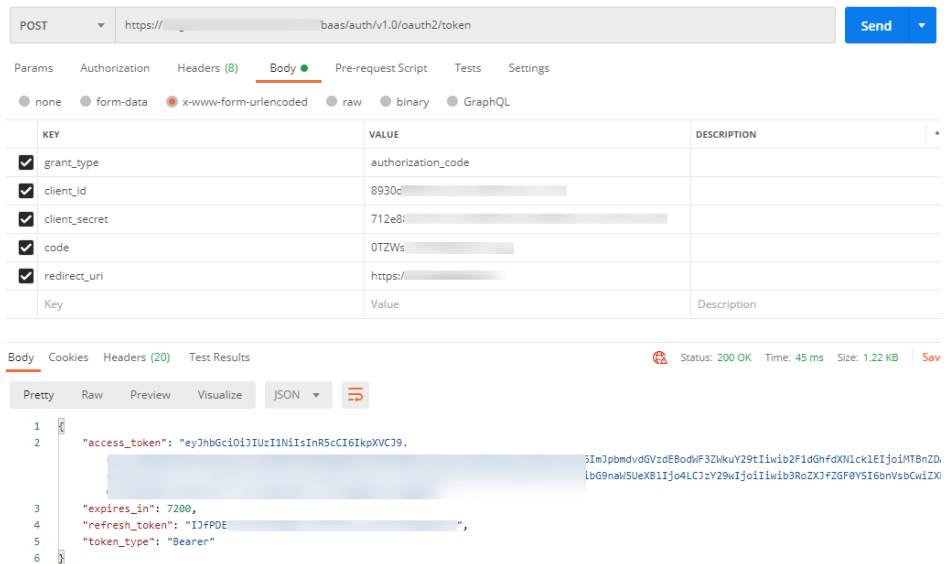
参数名	类型	必选 (M) / 可选 (O)	参数位置	参数含义	值
grant_type	String	M	Body	授权模式，OAuth2.0中的grant_type字段的取值。	authorization_code (即授权码模式)
client_id	String	M	Body	客户端鉴权ID，可在OAuth查看。	从c中获取。
client_secret	String	M	Body	客户端鉴权密钥，可在OAuth查看。	从c中获取。
redirect_url	String	M	Body	重定向地址。	从c中获取。

参数名	类型	必选 (M) / 可选 (O)	参数位置	参数含义	值
code	String	M	Body	授权码。	从e中获取。
locale	String	O	Body	语言。	en_US

表 3-12 返回参数说明

参数	参数说明
access_token	认证后可信任的Token凭证。
expires_in	当前access_token的有效期。
refresh_token	用于刷新access_token。
token_type	access_token类型。

图 3-7 接口调用结果



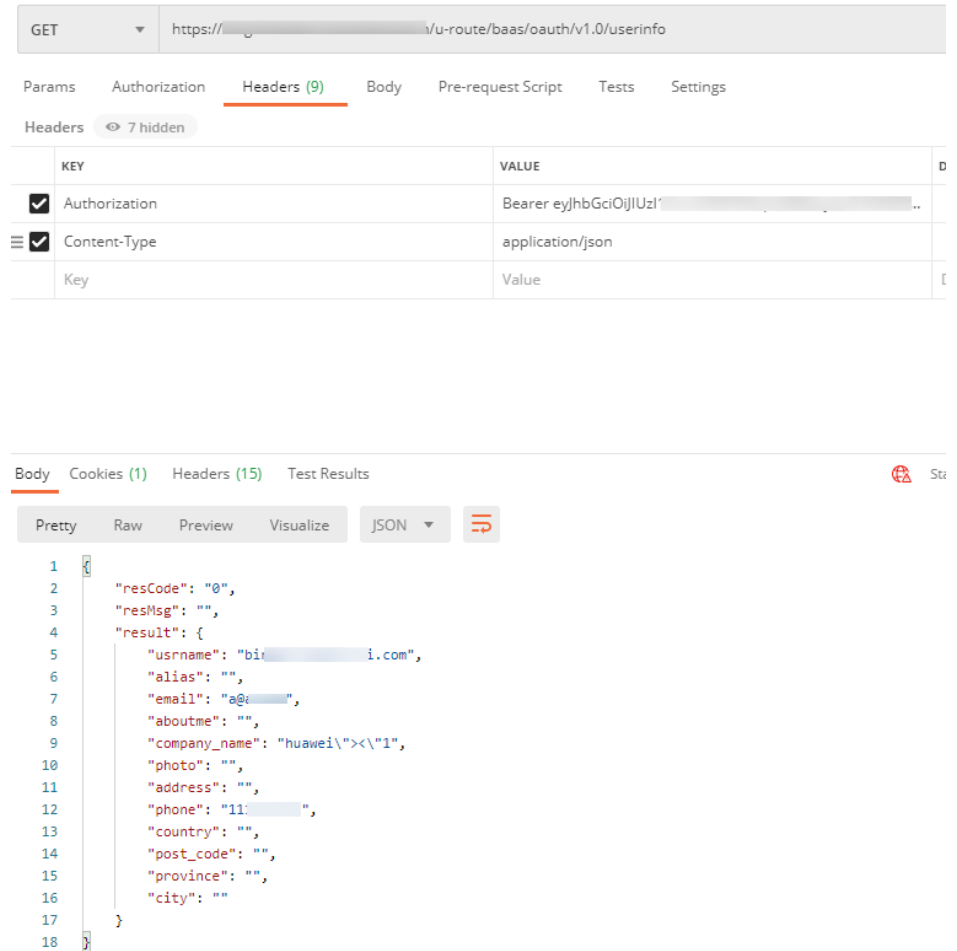
注意

使用授权码获取access-token后即失效，重新获取access-token需要重新生成授权码。

- g. 将得到的token_type与access_token进行拼接，以{token_type}{access_token}的形式设置为Headers中的Authorization参数，通过Get请求

调用接口 “https://AstroZero域名/u-route/baas/oauth/v1.0/userinfo”，获取用户信息。

图 3-8 获取用户信息



业务用户通过 RefreshToken 获取新的 Token

步骤1 开启RefreshToken特性。

1. 参考[如何登录经典版环境配置](#)中操作，登录经典版环境配置。
2. 在左侧导航栏中，选择“系统管理 > 系统参数”。
3. 在“内置系统参数”页签，查找“bingo.service.refreshtoken.enable”，单击搜索结果中的该参数名，在参数详情页单击“值”后的编辑按钮，设置参数值为“是”，单击“保存”。

图 3-9 查找参数



图 3-10 修改值为是



步骤2 业务用户（例如“test_cs”）通过服务编排的“PORTALUSERLOGIN”公式登录 AstroZero。


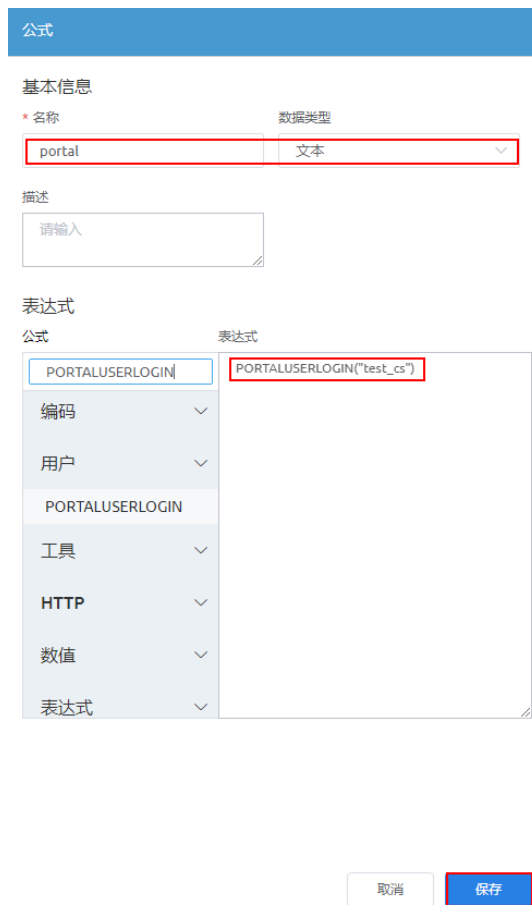
1. 在“我的应用”中，单击某个应用，进入应用开发平台。
2. 将鼠标放在左侧某个文件夹，单击界面上出现的“+”，在弹出菜单中选择“服务编排”。
3. 设置标签为“testPortalUserLogin”，单击“名称”的输入框，系统会自动填充。然后单击“添加”。
4. 在服务编排编辑器页面右侧，选择 ，在全局上下文页面单击“公式”后的加号。
5. 在公式页面设置“名称”为“portal”，“数据类型”设置为“文本”，表达式为“PORTALUSERLOGIN("test_cs")”，单击“保存”，创建公式变量“portal”，表达式中的“test_cs”为要登录系统的业务用户名。

图 3-11 新建公式变量





6. 在服务编排编辑器页面右侧，选择.
7. 在全局上下文页面，单击“变量”后的加号，默认创建名为“variable0”的文本类型变量。
8. 在服务编排编辑器页面右侧，选择，按照下图从全局上下文页面中拖拽参数到出参区域，设置服务编排的出参。

图 3-12 设置服务编排的出参

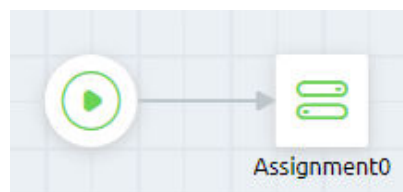




- 在左侧拖拽“逻辑”下“赋值”图标至画布中，按照下图设置赋值图元。

图 3-13 设置赋值图元



- 连接所有元素。

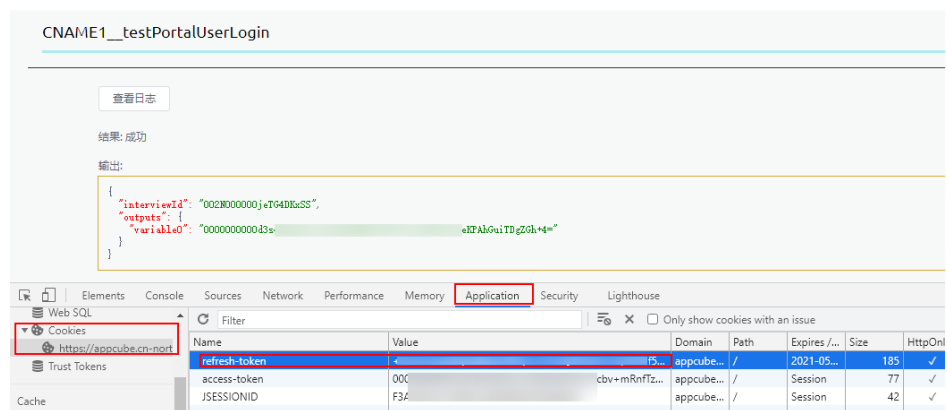


- 在页面上方，单击 ，保存服务编排。
- 保存成功后，单击 ，不用设置输入参数直接单击“运行”，运行服务编排。输出如下，表示业务用户已登录AstroZero。

```
{
  "interviewId": "002N000000jeTG4DKxSS",
  "outputs": {
    "variable0": "XXX"
  }
}
```

其中，“variable0”取值为业务用户的access-token值。

- 在输出的页面，按“F12”或者“Ctrl + Shift + I”，可开启调试工具。
- 在Cookies中，获取“refresh-token”值。



步骤3 使用Postman发送请求，获取新Token。

1. 在本地PC上下载并安装Postman工具，该工具仅用于测试使用。
2. 使用Postman进行POST请求，URL配置为“https://AstroZero域名/baas/auth/v1.0/refreshToken”，在请求消息头上设置参数“Content-Type”值为“application/json”，请求消息体中设置参数“grant_type”值为“refresh_token”，“refresh_token”值为步骤2.14获取的“refresh-token”参数值。输出的result值即为新的Token，原有的Token不管有没有失效，都会被置成失效状态。

图 3-14 Postman 发送请求

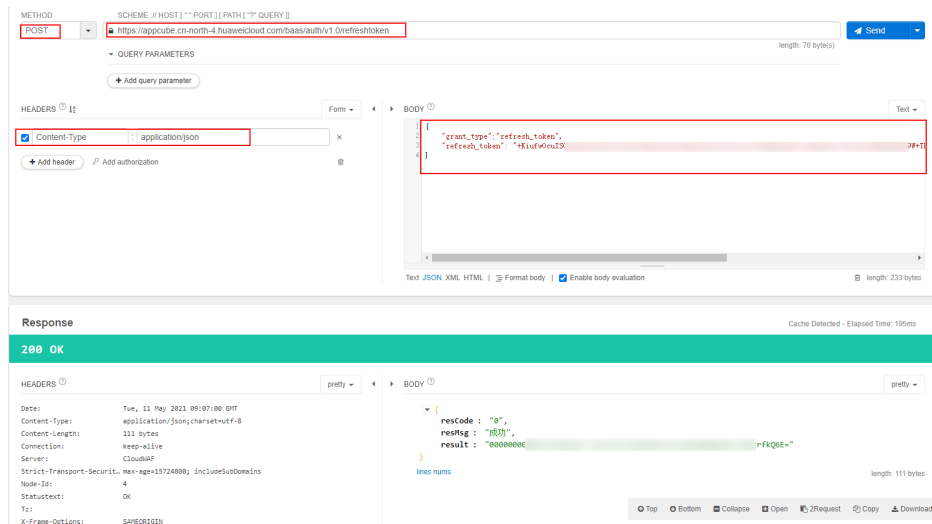


表 3-13 请求消息头

消息头名称	描述	是否必选
Content-Type	HTTP协议中设定的一个参数，用于标识返回的内容用什么格式去解析，此处必须配置为“application/json”。表示浏览器将返回内容解析为json对象。	是

表 3-14 消息体说明

参数名	描述	是否必选
grant_type	授权类型，配置为“refresh_token”。	是
refresh_token	配置为步骤2.14获取的“refresh-token”参数值。	是

----结束

3.3 返回结果

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。

状态码

状态码是一组从2xx（成功）到4xx或5xx（错误）的数字代码，状态码表示了请求响应的状态，完整的状态码列表请参见[状态码](#)。

对于使用OAuth 2.0的客户端鉴权模式，获取的用户access_token接口，如果调用后返回状态码为“200”，则表示请求成功。

响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”。

对于使用OAuth 2.0的client credentials鉴权模式，获取的用户Token接口，返回如下所示的消息头。

图 3-15 响应消息头

KEY	VALUE	DESCRIPTION
grant_type	client_credentials	
client_id	a3f1...b79fa8da0a	
client_secret	a7ad...8a9af3a288b8f8513ce6844c8b8ac	
Key	Value	Description

KEY	VALUE
Date	Tue, 26 May 2020 07:28:47 GMT
Content-Type	application/json;charset=UTF-8
Content-Length	125
Connection	keep-alive
Server	CloudWAF
Set-Cookie	HWWAFSEID=865a273d2496ff026; path=/ HWWAFSESTIME=1590478127210; path=/
Cache-Control	no-cache, no-store, max-age=0, must-revalidate
Expires	Fri, 01 Jan 1990 00:00:00 GMT
Pragma	no-cache
Strict-Transport-Security	max-age=15724800; includeSubDomains;
bingo-env	dev
X-Frame-Options	SAMEORIGIN
X-Content-Type-Options	nosniff
X-XSS-Protection	1
X-Via	1.1 P5bjwBGP2ut45:1 (Cdn Cache Server V2.0), 1.1 P5gdzjdx10uq29:8 (Cdn Cache Server V2.0)
X-WS-Request-Id	5eccc52f_P5gdzjdx10nh28_10040-25339

响应消息体（可选）

响应消息体通常以结构化格式（如JSON或XML）返回，与响应消息头中Content-type对应，传递除响应消息头之外的内容。

对于使用OAuth 2.0的客户端鉴权模式，获取的用户access_token接口，返回如下消息体。其中，“access_token”是需要获取的用户Token，取值以变量“XXX”表示。有了Token之后，您可以使用access_token认证调用其他API。

```
{
  "access_token": "XXX",
  "expires_in": 600,
  "token_type": "Bearer"
}
```

当接口调用出错时，会返回错误码“400”及错误信息说明，错误响应的Body体格式如下所示。

```
{
  "error": "unauthorized_client",
}
```

```
"error_description": "The client is not authorized to request a token using this method."  
}
```

其中，`error`表示错误提示，`error_description`表示错误描述信息。

4 快速入门

4.1 创建对象并定义字段

场景描述

创建一个车辆类型对象VehicleType，并定义字段，用于保存车辆类型、类型编号信息。

下面介绍如何调用创建对象并定义字段API，API的调用方法请参见[如何调用API](#)。

创建对象

步骤1 根据请求URI和请求方法，明确请求如下：

```
POST https://appcube.cn-north-4.huaweicloud.com/u-route/baas/metadata/v1.0/object
```

步骤2 设置请求消息头。

表 4-1 请求消息头

消息头名称	描述	是否必选
Content-Type	HTTP协议中设定的一个参数，用于标识返回的内容用什么格式去解析，此处必须配置为“application/json”。表示浏览器将返回内容解析为json对象。	是
access-token	当用户登录系统鉴权成功后，后台会生成一个token，在响应消息中携带，并存在浏览器中。下次再调用请求后，会带上这个token，表示这个用户已经被鉴权过。	是

步骤3 构造请求消息体。

```
{  
  "label": "VehicleType",  
  "name": "VehicleType_CST",  
  "description": "",  
  "fields": [  
    {  
      "label": "Name",
```

```

    "name": "name",
    "dataType": "Name"
  },
],
"view": {
  "ObjectDetailLayout": {
    "name": "_layout_VehicleType_CST_detail",
    "fields": [
      "name"
    ]
  },
  "ObjectRecordsLayout": {
    "name": "_layout_VehicleType_CST_records",
    "fields": [
      "name"
    ]
  }
}
}
}

```

返回如下响应，表示成功创建对象，记录“result”字段取值“cust000000aHYAsrPh4K”，为对象ID。

```

{
  "resCode": "0",
  "resMsg": "成功",
  "result": "cust000000aHYAsrPh4K"
}

```

---结束

自定义字段

步骤1 根据请求URI和请求方法，明确请求如下，其中“cust000000aHYAsrPh4K”为对象ID：

```

POST https://appcube.cn-north-4.huaweicloud.com/u-route/baas/metadata/v1.0/object/cust000000aHYAsrPh4K/fields

```

步骤2 设置请求消息头。

表 4-2 请求消息头

消息头名称	描述	是否必选
Content-Type	HTTP协议中设定的一个参数，用于标识返回的内容用什么格式去解析，此处必须配置为“application/json”。表示浏览器将返回内容解析为json对象。	是
access-token	当用户登录系统鉴权成功后，后台会生成一个token，在响应消息中携带，并存在浏览器中。下次再调用请求后，会带上这个token，表示这个用户已经被鉴权过。	是

步骤3 构造请求消息体，创建“VehicleType”、“VehicleNumber”字段。

```

{
  "fields": [
    {
      "label": "VehicleType",
      "name": "VehicleType_CST",
      "dataType": "text",
      "dataLength": 255,

```

```
    "caseSensitive": false,
    "isRequired": false,
    "isUnique": false,
    "isSearchable": false
  },
  {
    "label": "VehicleNumber",
    "name": "VehicleNumber__CST",
    "dataType": "text",
    "dataLength": 255,
    "caseSensitive": true,
    "isRequired": false,
    "isUnique": true,
    "isSearchable": false
  }
]
```

返回如下响应，表示成功创建字段。

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": [
    "cust000000aHaE6S3mim",
    "cust000000aHaE6TlNVA"
  ]
}
```

----结束

4.2 获取对象列表

场景描述

如果用户想要查看该账号下的所有对象列表，可以使用查询Object列表接口查看。

下面介绍如何调用查询Object列表API，API的调用方法请参见[如何调用API](#)。

调用接口

步骤1 根据请求URI和请求方法，明确请求如下。

```
GET https://appcube.cn-north-4.huaweicloud.com/u-route/baas/metadata/v1.0/object?flag=S&type=cst,std
```

URI中参数说明如下：

- flag=S：查询对象结果不带任何其他关联信息（默认）。
- flag=D：查询没有关联Tab的对象。
- type=cst：查询用户自定义对象（如果没有type则为该值，默认值）。
- type=std：查询标准对象。
- type=stp：查询setup对象。
- type=cst,std：查询自定义对象和标准对象，其他组合类似，中间用逗号隔开。

对象列表以对象名按照字母A-Z进行排序。

步骤2 设置请求消息头。

表 4-3 请求消息头

消息头名称	描述	是否必选
Content-Type	HTTP协议中设定的一个参数，用于标识返回的内容用什么格式去解析，此处必须配置为“application/json”。表示浏览器将返回内容解析为json对象。	是
access-token	当用户登录系统鉴权成功后，后台会生成一个token，在响应消息中携带，并存在浏览器中。下次再调用请求后，会带上这个token，表示这个用户已经被鉴权过。	是

该接口不用构造请求消息体。返回如下响应，表示成功查询对象列表。

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": [
    {
      "id": "stad00000000000000GH",
      "keyPrefix": "10gh",
      "name": "AttachFile",
      "category": 2,
      "label": "Attach File",
      "pluralLabel": "Attach Files",
      "version": 55834574848,
      "contextSetup": 0,
      "trackFieldHistory": false,
      "allowReports": false,
      "allowActivities": false,
      "allowChatter": false,
      "allowStreamingApiAccess": false,
      "allowBulkApiAccess": false,
      "allowSearch": false,
      "allowSharing": false,
      "sharingMode": 0,
      "deploymentStatus": 0,
      "properties": "",
      "status": 0,
      "createdBy": "",
      "createdDate": "",
      "lastModifiedBy": "",
      "lastModifiedDate": "",
      "description": "",
      "objectType": 0,
      "protectMode": 0,
      "installedPackage": "",
      "createdBy.name": "",
      "lastModifiedBy.name": "",
      "childObjects": null,
      "parentObjects": [
        {
          "parentObject": "stad00000000000000GD",
          "cascadeDelete": false
        },
        {
          "parentObject": "stad00000000000000GG",
          "cascadeDelete": false
        },
        {
          "parentObject": "stup000000000000002I",
          "cascadeDelete": false
        }
      ]
    }
  ]
}
```

```
    },  
    {  
      "parentObject": "stup000000000000003L",  
      "cascadeDelete": false  
    }  
  ]  
}  
}
```

----结束

5 API

5.1 数据模块

5.1.1 按记录 ID 查询数据

功能介绍

按记录ID查询数据，该接口为同步接口（同步接口是指调用后，会等待函数执行结束后，才返回结果）。

📖 说明

在此之前需要先创建对象以及相关数据记录。

URI

GET *AstroZero*域名/u-route/baas/data/v1.0/bobject/{*objectName*}/{*recordId*?
fields={*f1*,*f2*}

表 5-1 路径参数

参数	是否必选	描述
objectName	是	数据对象的名称。
recordId	是	记录ID，可通过调用 按条件查询对象数据 接口查看（响应消息中的“id”）。
fields	否	对象显示的字段，多个字段则用逗号分隔。如果该字段不为空，则仅返回显示的字段和必要信息（如数据记录所有者信息和id）；为空则返回整个记录数据。

请求参数

无

响应参数

表 5-2 响应参数说明

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Array	如果请求成功，数组中只有一个对象，对象中包含记录中的数据。

请求示例

查询一个记录，例如获取“TestProject_testForObject_CST”对象中，记录ID为“cfw0000000aKin6AQgfQ”的数据。

```
GET https://AstroZero域名/u-route/baas/data/v1.0/bobject/TestProject_testForObject_CST/cfw0000000aKin6AQgfQ
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": [
    {
      "TestProject_address_CST": null,
      "createdBy": "10gd000000SDzgxNSckK",
      "createdBy__objectType": "User",
      "createdBy.name": "test0xx",
      "createdDate": "2020-05-28 14:34:40",
      "currencyIsoCode": "",
      "id": "cfw0000000aKin6AQgfQ",
      "installedPackage": null,
      "lastModifiedBy": "10gd000000SDzgxNSckK",
      "lastModifiedBy__objectType": "User",
      "lastModifiedBy.name": "test0xx",
      "lastModifiedDate": "2020-05-28 14:34:40",
      "name": "testd",
      "owner": "10gd000000SDzgxNSckK",
      "owner__objectType": "User",
      "owner.name": "test0xx"
    }
  ]
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.1.2 按记录 ID 删除数据

功能介绍

按记录ID删除数据，该接口为同步接口（同步接口是指调用后，会等待函数执行结束后，才返回结果）。

📖 说明

在此之前需要先创建对象以及待删除的数据记录。

URI

DELETE *AstroZero*域名/u-route/baas/data/v1.0/bobject/{*ObjectName*}/{*recordId*}

表 5-3 路径参数

参数	是否必选	描述
objectName	是	数据对象的名称。
recordId	是	记录ID，可通过调用 按条件查询对象数据 接口查看（响应消息中的“id”）。

请求参数

无

响应参数

表 5-4 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Number	删除记录的个数，响应成功时携带。

请求示例

删除一个记录，例如删除“TestProject_testForObject_CST”对象，记录ID为“cfw0000000aKin6AQgfQ”的数据。

```
DELETE https://AstroZero域名/u-route/baas/data/v1.0/bobject/TestProject_testForObject_CST/cfw0000000aKin6AQgfQ
```


响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": 1
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.1.3 按记录 ID 更新数据

功能介绍

按记录ID更新数据，该接口为同步接口（同步接口是指调用后，会等待函数执行结束后，才返回结果）。

说明

在此之前需要先创建对象以及待更新的数据记录。

URI

PUT *AstroZero*域名/u-route/baas/data/v1.0/bobject/{*objectName*}/{*recordId*}

表 5-5 路径参数

参数	是否必选	描述
objectName	是	数据对象的名称。
recordId	是	记录ID，可通过调用 按条件查询对象数据 接口查看（响应消息中的“id”）。

请求参数

表 5-6 请求参数

参数	是否必选	参数类型	描述
setValue	是	Object	更改的属性和新值组成的对象。

响应参数

表 5-7 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Number	更新记录的个数，成功时携带。

请求示例

更新一个记录，例如更新“TestProject__testForObject__CST”对象，记录ID为“cfw0000000aKin6AQgfQ”的数据。

```
PUT https://AstroZero域名/u-route/baas/data/v1.0/boject/TestProject__testForObject__CST/cfw0000000aKin6AQgfQ
```

```
{
  "setValue": {
    "name": "newName"
  }
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": 1
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.1.4 按记录 ID 列表查询数据

功能介绍

按记录ID列表查询数据，该接口为同步接口（同步接口是指调用后，会等待函数执行结束后，才返回结果）。

说明

在此之前需要先创建对象以及待查询的数据记录。

URI

GET *AstroZero*域名/u-route/baas/data/v1.0/bobject/{*objectName*?
fields={*f1,f2*&records={*id1,id2*}

表 5-8 路径参数

参数	是否必选	描述
objectName	是	数据对象的名称。
records	是	记录ID列表，至少有一个记录ID。 其中， <i>{id1,id2}</i> 可通过调用 按条件查询对象数据 接口查看（响应消息中的“id”）。
fields	否	对象显示的字段，多个字段用逗号分隔。如果该字段不为空，则仅返回显示的字段和必要信息（如owner信息和id）；为空则返回整个记录字段。

请求参数

无

响应参数

表 5-9 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Array	查询的记录数据所形成的数组。

请求示例

查询两个记录，例如查询“TestProject_testForObject_CST”对象中，记录ID为“cfw0000000aKin6AQgfQ”、“cfw0000000aKlMSFn7vk”的数据。

```
GET https://AstroZero域名/u-route/baas/data/v1.0/bobject/TestProject_testForObject_CST?records=cfw0000000aKin6AQgfQ,cfw0000000aKlMSFn7vk
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": [
```

```
{
  "TestProject__address__CST": null,
  "createdBy": "10gd000000SDzgxNSckK",
  "createdBy.__objectType": "User",
  "createdBy.name": "test",
  "createdDate": "2020-05-28 14:34:40",
  "currencyIsoCode": "",
  "id": "cfw0000000aKin6AQgfQ",
  "installedPackage": null,
  "lastModifiedBy": "10gd000000SDzgxNSckK",
  "lastModifiedBy.__objectType": "User",
  "lastModifiedBy.name": "test",
  "lastModifiedDate": "2020-05-28 15:03:15",
  "name": "newName",
  "owner": "10gd000000SDzgxNSckK",
  "owner.__objectType": "User",
  "owner.name": "test"
},
{
  "TestProject__address__CST": null,
  "createdBy": "10gd000000SDzgxNSckK",
  "createdBy.__objectType": "User",
  "createdBy.name": "test",
  "createdDate": "2020-05-28 15:10:38",
  "currencyIsoCode": "",
  "id": "cfw0000000aKlMSFn7vk",
  "installedPackage": null,
  "lastModifiedBy": "10gd000000SDzgxNSckK",
  "lastModifiedBy.__objectType": "User",
  "lastModifiedBy.name": "test",
  "lastModifiedDate": "2020-05-28 15:10:38",
  "name": "test2",
  "owner": "10gd000000SDzgxNSckK",
  "owner.__objectType": "User",
  "owner.name": "test"
}
]
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.1.5 新增对象数据

功能介绍

新增对象数据，该接口为同步接口（同步接口是指调用后，会等待函数执行结束后，才返回结果）。

说明

新增对象数据前，需要先创建对象。

URI

POST AstroZero域名/u-route/baas/data/v1.0/bobject/{objectName}

表 5-10 路径参数

参数	是否必选	描述
objectName	是	数据对象的名称。

请求参数

表 5-11 请求参数

参数	是否必选	参数类型	描述
records	是	Array	记录的属性和值组成的对象数组。

响应参数

表 5-12 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Array	为记录ID组成的对象数组。

请求示例

新增两个记录，例如新增“TestProject__testForObject__CST”对象的两个记录。

POST https://AstroZero域名/u-route/baas/data/v1.0/bobject/TestProject__testForObject__CST

```
{
  "records": [
    {
      "name": "name1"
    },
    {
      "name": "name2"
    }
  ]
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": [
    {
```

```
    "id": "cfw0000000aKmGskAsDo"  
  },  
  {  
    "id": "cfw0000000aKmGskAsDp"  
  }  
]  
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.1.6 按记录 ID 列表删除数据

功能介绍

按记录ID列表删除数据，该接口为同步接口（同步接口是指调用后，会等待函数执行结束后，才返回结果）。

说明

在此之前需要先创建对象以及待删除的数据记录。

URI

DELETE *AstroZero*域名/u-route/baas/data/v1.0/bobject/{*objectName*}

表 5-13 路径参数

参数	是否必选	描述
objectName	是	数据对象的名称。

请求参数

表 5-14 请求参数

参数	是否必选	参数类型	描述
records	是	Array	记录ID列表（最大支持100个记录ID）。

响应参数

表 5-15 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Number	删除记录的个数，成功时携带

请求示例

删除两个记录，例如删除“TestProject_testForObject_CST”对象，记录ID为“cfw0000000aKmGskAsDo”和“cfw0000000aKmGskAsDp”的数据。

```
DELETE https://AstroZero域名/u-route/baas/data/v1.0/bobject/TestProject_testForObject_CST?records=cfw0000000aKmGskAsDo,cfw0000000aKmGskAsDp
```

响应示例

```
{  "resCode": "0",  "resMsg": "成功",  "result": 2}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.1.7 按条件更新数据

功能介绍

按条件更新数据，该接口为同步接口（同步接口是指调用后，会等待函数执行结束后，才返回结果）。

📖 说明

在此之前需要先创建对象以及待更新的数据记录。

URI

```
PUT AstroZero域名/u-route/baas/data/v1.0/bobject/{objectName}
```

表 5-16 路径参数

参数	是否必选	描述
objectName	是	数据对象的名称。

请求参数

表 5-17 请求参数

参数	是否必选	参数类型	描述
setValue	是	Object	修改对象的字段和值。
condition	是	Object	更新记录的条件。

响应参数

表 5-18 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Number	更新记录的个数，成功时携带

请求示例

更新“TestProject_testForObject_CST”对象中，满足name字段值为“newName”的记录，更新name字段值为“name3”。

PUT https://AstroZero域名/u-route/baas/data/v1.0/bobject/TestProject_testForObject_CST

```
{
  "setValue": {
    "name": "name3"
  },
  "condition": {
    "conjunction": "AND",
    "conditions": [
      {
        "field": "name",
        "operator": "eq",
        "value": "newName"
      }
    ]
  }
}
```


响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": 1
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.1.8 按条件查询对象数据

功能介绍

按条件查询对象数据。

URI

POST *AstroZero*域名/u-route/baas/data/v1.0/query/{*objectName*?
base=*X*&fields=*f1,f2*

表 5-19 路径参数

参数	是否必选	描述
objectName	是	数据对象的名称。
base	否	查询参数，用于区分objectName传入数据样式。支持如下选项： <ul style="list-style-type: none">• I：对应对象的ID。• N：传入的对象名称。
fields	否	查询参数，对象显示的字段。

请求参数

表 5-20 请求参数

参数	是否必选	参数类型	描述
condition	否	Object	<p>表示查询的条件，包含以下元素，若为空则表示查询所有对象数据。</p> <ul style="list-style-type: none"> • conjunction: 表示需要满足的条件个数。 <ul style="list-style-type: none"> - AND: 表示条件都要满足。 - OR: 表示至少要满足一个条件。 • conditions: 筛选条件的数组。如果参数为空，代表查询所有对象数据，若不为空，其中的每一个元素对象都包含如下三个字段。 <ul style="list-style-type: none"> - field: 类型string，字段名称。 - operator: 类型string，操作符。例如，“eq”表示等于、“ne”表示不等于、“gt”表示大于、“lt”表示小于、“ge”表示大于等于、“le”表示小于等于。 - value: 类型string，对比值。 • relation: 数组中包含parents（向上查询）和children（向下查询）两个元素，两者分别对应相应的数组。其中，parents包含condition数组和relatedField（关联字段名称）。children包含condition数组和relationName（关联字段名称）。 condition数组包含： <ul style="list-style-type: none"> - conjunction: 取值为AND或OR。 - conditions: 同上conditions参数。 - relation: 同上relation参数。

参数	是否必选	参数类型	描述
relation	否	Object	<p>关联查询，包含parents（向上查询）和children（向下查询）两个元素，两者分别对应相应的数组。</p> <p>其中，parents包含condition数组和relatedField（关联字段名称）。children包含condition数组和relationName（关联字段名称）。</p> <p>condition数组包含：</p> <ul style="list-style-type: none"> • conjunction：取值为AND或OR。 • conditions：同上conditions参数。 • relation：同上relation参数。 <pre> { //向上查询 parents: [//第一级父对象 { relatedField: "CarHolder__CST", //当前对象的字段(Lookup/MasterDetail类型),据此确定父对象 options: { fields: ["id", "name", "Age__CST"] //父对象的待查字段 }, parents: [//(可选)第二级父对象 { relatedField: "cust__CST", //父对象的字段(Lookup/MasterDetail类型),据此确定祖父对象 options: { fields: ["id", "name", "Email__CST"] //祖父对象的待查字段 } }] }], //向下查询 childs: [//第一级子对象 { relationName: "Contacts", //子对象关系名(子对象定义Lookup/MasterDetail时指定),据此确定子对象 options: { fields: ["id", "name", "Age__CST", "cust__CST"] //子对象的待查字段 }, childs: [//(可选)第二级子对象 { relationName: "ContactCars", //孙子对象关系名 options: { fields: ["id", "name", "Price__CST", "CarHolder__CST"] //孙子对象待查字段 } }] }] } </pre>

参数	是否必选	参数类型	描述
options	否	Object	<p>附加选项，对查询返回的结果进行如下操作。</p> <ul style="list-style-type: none"> orderBy: 按指定字段的值排序，取值为asc（升序）和desc（降序） <pre>options: { orderBy: [{ field: "name", order: db.Order.desc }], }</pre> groupby: 按照指定字段进行分组 <pre>options: { groupby: [{ field: "name", }], }</pre> distinct: 去重 <pre>options: { distinct: true, }</pre> 限制查询返回的记录数: 通过skip值指定从第多少条开始，limit指定总共返回多少条记录 <pre>{ options: { skip: 100, limit: 100 } }</pre>

响应参数

表 5-21 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Array	查询返回的数据。
count	number	对象中满足条件的总记录数。

请求示例

查询 “TestProject_testForObject1_CST” 对象中，name值为 “zhang” 的记录数，且限制只返回3条，总记录数为5条，返回的记录按 “TestProject_length_CST” 字段降序排列，查询参数base为N表示按对象名称进行查询。

POST https://AstroZero域名/u-route/baas/data/v1.0/query/TestProject_testForObject1_CST?base=N

```
{
  "condition": {
    "conjunction": "AND",
    "conditions": [
      {
        "field": "name",
        "operator": "eq",
        "value": "zhang"
      }
    ]
  },
  "options": {
    "orderby": [
      {
        "field": "TestProject_length_CST",
        "order": "desc"
      }
    ],
    "limit": 3
  }
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": [
    {
      "TestProject_age_CST": null,
      "TestProject_length_CST": "888",
      "TestProject_name_CST": null,
      "createdBy": "10gd000000SDzgxNSckK",
      "createdBy_objectType": "User",
      "createdBy.name": "test0xx",
      "createdDate": "2020-05-29 12:50:59",
      "currencyIsoCode": "",
      "id": "cglE000000aMGHkvmhmb",
      "installedPackage": null,
      "lastModifiedBy": "10gd000000SDzgxNSckK",
      "lastModifiedBy_objectType": "User",
      "lastModifiedBy.name": "test0xx",
      "lastModifiedDate": "2020-05-29 12:50:59",
      "name": "zhang",
      "owner": "10gd000000SDzgxNSckK",
      "owner_objectType": "User",
      "owner.name": "test0xx"
    },
    {
      "TestProject_age_CST": null,
      "TestProject_length_CST": "456",
      "TestProject_name_CST": null,
      "createdBy": "10gd000000SDzgxNSckK",
      "createdBy_objectType": "User",
      "createdBy.name": "test0xx",
      "createdDate": "2020-05-29 12:50:59",
      "currencyIsoCode": "",
      "id": "cglE000000aMGHkvmhma",
      "installedPackage": null,
      "lastModifiedBy": "10gd000000SDzgxNSckK",
      "lastModifiedBy_objectType": "User",
      "lastModifiedBy.name": "test0xx",
      "lastModifiedDate": "2020-05-29 12:50:59",
      "name": "zhang",
      "owner": "10gd000000SDzgxNSckK",
      "owner_objectType": "User",
      "owner.name": "test0xx"
    }
  ]
}
```

```
"lastModifiedDate": "2020-05-29 12:50:59",
"name": "zhang",
"owner": "10gd000000SDzgXNSckK",
"owner__objectType": "User",
"owner.name": "test0xx"
},
{
  "TestProject__age__CST": null,
  "TestProject__length__CST": "17",
  "TestProject__name__CST": null,
  "createdBy": "10gd000000SDzgXNSckK",
  "createdBy__objectType": "User",
  "createdBy.name": "test0xx",
  "createdDate": "2020-05-29 12:43:11",
  "currencyIsoCode": "",
  "id": "cglE000000aMFjB9ipAu",
  "installedPackage": null,
  "lastModifiedBy": "10gd000000SDzgXNSckK",
  "lastModifiedBy__objectType": "User",
  "lastModifiedBy.name": "test0xx",
  "lastModifiedDate": "2020-05-29 12:43:11",
  "name": "zhang",
  "owner": "10gd000000SDzgXNSckK",
  "owner__objectType": "User",
  "owner.name": "test0xx"
}
],
"count": 5
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.1.9 按条件删除对象数据

功能介绍

按条件删除对象数据。

URI

POST *AstroZero*域名/u-route/baas/data/v1.0/delete/{objectName}

表 5-22 参数说明

参数	是否必选	描述
objectName	是	数据对象的名称。

请求参数

表 5-23 请求参数

参数	是否必选	参数类型	描述
condition	是	Object	<p>表示删除的条件，其中包含以下三个元素，为空则表示删除所有对象数据。</p> <ul style="list-style-type: none"> • conjunction: 表示需要满足的条件个数。 <ul style="list-style-type: none"> - AND: 表示条件都要满足。 - OR: 表示至少要满足一个条件。 • conditions: 筛选条件的数组。如果参数为空，代表查询所有对象数据，若不为空，其中的每一个元素对象都包含如下三个字段。 <ul style="list-style-type: none"> - field: 类型string，字段名称。 - operator: 类型string，操作符。 - value: 类型string，对比值。 • relation: 数组中包含parents（向上查询）和children（向下查询）两个元素，两者分别对应相应的数组。其中，parents包含condition数组和relatedField（关联字段名称）。children包含condition数组和relationName（关联字段名称）。 <p>condition数组包含：</p> <ul style="list-style-type: none"> - conjunction: 取值为AND或OR。 - conditions: 同上conditions参数。 - relation: 同上relation参数。

响应参数

表 5-24 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Number	删除记录数。

请求示例

删除 “TestProject_testForObject1_CST” 对象中，name为 “test” 的记录数。
POST https://AstroZero域名/u-route/baas/data/v1.0/delete/TestProject_testForObject1_CST

```
{
  "condition": {
    "conjunction": "AND",
    "conditions": [
      {
        "field": "NAME",
        "operator": "eq",
        "value": "test"
      }
    ]
  }
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": 0
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.1.10 按条件查询对象记录数

功能介绍

按条件查询对象记录数。

URI

POST AstroZero域名/u-route/baas/data/v1.0/count/{objectName}

表 5-25 路径参数

参数	是否必选	描述
objectName	是	数据对象的名称。
base	否	查询参数，区分传入数据样式，支持选项如下： <ul style="list-style-type: none">L：传入的数据ID。N：传入对象名称。

请求参数

表 5-26 请求参数

参数	是否必选	参数类型	描述
condition	是	Object	<p>表示查询的条件，其中包含以下三个元素，若为空则表示查询所有对象数据。</p> <ul style="list-style-type: none"> • conjunction: 表示需要满足的条件个数。 <ul style="list-style-type: none"> - AND: 表示条件都要满足。 - OR: 表示至少要满足一个条件。 • conditions: 筛选条件的数组。如果参数为空，代表查询所有对象数据，若不为空，其中的每一个元素对象都包含三个字段： <ul style="list-style-type: none"> - field: 类型string，字段名称。 - operator: 类型string，操作符。例如，操作符“eq”表示等于，“ne”表示不等于，“gt”表示大于，“lt”表示小于，“ge”表示大于等于，“le”表示小于等于。 - value: 类型string，对比值。 • relation: 数组中包含parents（向上查询）和children（向下查询）两个元素，两者分别对应相应的数组。 其中，parents包含condition数组和relatedField（关联字段名称）。children包含condition数组和relationName（关联字段名称）。 condition数组包含： <ul style="list-style-type: none"> - conjunction: 取值为AND或OR。 - conditions: 同上conditions参数。 - relation: 同上relation参数。

响应参数

表 5-27 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Number	返回的记录数。

请求示例

查询 “TestProject_testForObject1_CST” 对象中，name为 “lisi” 的记录数。

POST `https://AstroZero域名/u-route/baas/data/v1.0/count/TestProject_testForObject1_CST`

```
{
  "condition": {
    "conjunction": "AND",
    "conditions": [
      {
        "field": "NAME",
        "operator": "eq",
        "value": "lisi"
      }
    ]
  }
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": 8
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.1.11 批量插入或修改对象数据

功能介绍

批量插入或修改对象数据，若数据包含id进行更新，若数据不含id进行插入。

URI

POST `AstroZero域名/u-route/baas/data/v1.0/upsert/{objectName}`

表 5-28 路径参数

参数	是否必选	描述
objectName	是	数据对象的名称。

请求参数

表 5-29 请求参数

参数	是否必选	参数类型	描述
records	是	Array	需要插入或者更新的数据组成的数组，若数据中包含id则视为更新，否则视为插入新数据。

响应参数

表 5-30 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Array	插入或者更新成功，返回相应记录id组成的数组。

请求示例

向对象“TestProject__testForObject1__CST”中插入三条数据，body体中含有要插入的records参数。

POST https://AstroZero域名/u-route/baas/data/v1.0/upsert/TestProject__testForObject1__CST

```
{
  "records":[
    {
      "name":"zhang",
      "TestProject__length__CST":"17"
    },
    {
      "name":"li",
      "TestProject__length__CST":"30"
    },
    {
      "name":"wang",
      "TestProject__length__CST":"20"
    }
  ]
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": [
    {
      "id": "cglE000000aMFjB9ipAu"
    }
  ],
}
```

```
{
  "id": "cglE000000aMFjB9ipAv"
},
{
  "id": "cglE000000aMFjB9ipAw"
}
]
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2 元数据

5.2.1 操作对象

功能介绍

该接口是一个统一的接口，会根据对象名称的不同进行创建、更新、删除，查询操作。

URI

*POST、GET、PUT、DELETE AstroZero域名/u-route/baas/metadata/v1.0/
{object_name}*

表 5-31 路径参数

参数	是否必选	参数类型	描述
object_name	是	String	对象的名称。

请求参数

请求体参数会根据object_name的不同而有所不同，下面会对具体某类别的object如何操作做详细的说明。

响应参数

响应参数介绍，请参见如下具体某类别的object。

请求示例

具体请求示例，请参见如下具体某类别的object。

响应示例

具体响应示例，请参见如下具体某类别的object。

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.2 操作对象实例

功能介绍

该接口是一个统一的接口，会根据对象名称和对象实例的ID对该对象的实例进行的不同操作，例如对该对象创建记录、更新记录、删除记录、查询记录等。

URL

*POST、GET、PUT、DELETE AstroZero域名/u-route/baas/metadata/v1.0/
{object_name}/{object_id}*

表 5-32 路径参数

参数	是否必选	参数类型	描述
object_name	是	String	对象的名称。
object_id	是	String	所填对象对应的实例ID，可通过调用 查询所有Object对象 接口查看（响应消息中的“id”）。

请求参数

请求体参数会根据object_name即具体对象的不同而有所不同，下面会对具体某类别的object如何操作进行详细说明。

响应参数

响应参数介绍，请参见如下具体某类别的object。

请求示例

具体请求示例，请参见如下具体某类别的object。

响应示例

具体响应示例，请参见如下具体某类别的object。

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.3 Object 对象的操作

5.2.3.1 创建对象

功能介绍

添加一个对象（Object）。

URI

POST AstroZero域名/u-route/baas/metadata/v1.0/Object

请求参数

表 5-33 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	创建表对象时的名字，一旦创建不可修改。
label	否	String	对象的别名，可以修改。
description	否	String	对象的描述信息。
fields	是	Array of field object	对象的字段列表。
view	是	Object of layout object	是一个对象，其中包含 ObjectDetailLayout对象和 ObjectRecordsLayout对象。

表 5-34 fields 字段参数

参数	是否必选	参数类型	描述
label	是	String	字段标签。
name	是	String	字段名。
dataType	是	String	字段类型。

响应参数

表 5-35 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	string	成功会返回Object的一个ID值。

请求示例

创建Object对象实例。

```
POST https://AstroZero域名/u-route/baas/metadata/v1.0/Object
```

```
{
  "label": "VehicleType",
  "name": "VehicleType_CST",
  "description": "",
  "fields": [
    {
      "label": "Name",
      "name": "name",
      "dataType": "Name"
    }
  ],
  "view": {
    "ObjectDetailLayout": {
      "name": "_layout_VehicleType_CST_detail",
      "fields": [
        "name"
      ]
    },
    "ObjectRecordsLayout": {
      "name": "_layout_VehicleType_CST_records",
      "fields": [
        "name"
      ]
    }
  }
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": "cust000000aKtFZpERvc"
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.3.2 查询所有 Object 对象

功能介绍

查询所有Object对象。

URI

GET AstroZero域名/u-route/baas/metadata/v1.0/Object

请求参数

无

响应参数

表 5-36 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Array	成功返回会返回该账号下所有定义的Object对象，以一个列表展示，每个值是一个对象。

请求示例

查询所有Object实例。

```
GET https://AstroZero域名/u-route/baas/metadata/v1.0/Object
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": [
    {
      "id": "cust000000aKtFZpERvc",
      "keyPrefix": "c02W",
      "name": "SmartCampus__OperatorOrganization__CST",
      "category": 3,
      "label": "OperatorOrganization",
      "pluralLabel": "OperatorOrganizations",
      "version": 7,
      "contextSetup": 0,
      "trackFieldHistory": false,
    }
  ]
}
```



```
"allowReports": false,
"allowActivities": false,
"allowChatter": false,
"allowStreamingApiAccess": false,
"allowBulkApiAccess": false,
"allowSearch": false,
"allowSharing": false,
"sharingMode": 0,
"deploymentStatus": 0,
"properties": "",
"status": 0,
"createdBy": "10gd000000SDyqaQW6uu",
"createdDate": "2020-03-12 15:14:56",
"lastModifiedBy": "10gd000000SDyqaQW6uu",
"lastModifiedDate": "2020-03-12 15:15:00",
"description": "",
"objectType": 0,
"protectMode": 1,
"installedPackage": "003L000000YGxJbahwps",
"createdBy.name": "test0xx",
"lastModifiedBy.name": "test0xx",
"childObjects": null,
"parentObjects": [
  {
    "parentObject": "cust000000PeSg3hYOOG",
    "cascadeDelete": true
  },
  {
    "parentObject": "stad00000000000000GD",
    "cascadeDelete": false
  },
  {
    "parentObject": "stad00000000000000GG",
    "cascadeDelete": false
  },
  {
    "parentObject": "stup000000000000002I",
    "cascadeDelete": false
  },
  {
    "parentObject": "stup000000000000003L",
    "cascadeDelete": false
  }
]
},
{
  "id": "cust000000aKoa6I4E6K",
  "keyPrefix": "cE6K",
  "name": "test_qw_CST",
  "category": 3,
  "label": "qw",
  "pluralLabel": "qws",
  "version": 3,
  "contextSetup": 0,
  "trackFieldHistory": false,
  "allowReports": false,
  "allowActivities": false,
  "allowChatter": false,
  "allowStreamingApiAccess": false,
  "allowBulkApiAccess": false,
  "allowSearch": false,
  "allowSharing": false,
  "sharingMode": 0,
  "deploymentStatus": 0,
  "properties": "",
  "status": 0,
  "createdBy": "10gd000000SDyqaQW6uu",
  "createdDate": "2020-05-28 15:55:42",
  "lastModifiedBy": "10gd000000SDyqaQW6uu",
```

```

"lastModifiedDate": "2020-05-28 17:47:03",
"description": "",
"objectType": 0,
"protectMode": 0,
"installedPackage": "",
"createdBy.name": "test0xx",
"lastModifiedBy.name": "test0xx",
"childObjects": null,
"parentObjects": [
  {
    "parentObject": "stad00000000000000GD",
    "cascadeDelete": false
  },
  {
    "parentObject": "stad00000000000000GG",
    "cascadeDelete": false
  },
  {
    "parentObject": "stup000000000000002I",
    "cascadeDelete": false
  },
  {
    "parentObject": "stup000000000000003L",
    "cascadeDelete": false
  }
]
}
}
}

```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.3.3 按照 object_id 查询 Object

功能介绍

按照ID查询Object。

URI

*GET AstroZero*域名/u-route/baas/metadata/v1.0/Object/{object_id}

表 5-37 路径参数

参数	是否必选	参数类型	描述
object_id	是	String	Object的ID值，可通过调用 查询所有Object对象 接口查看（响应消息中的“id”）。

请求参数

无

响应参数

表 5-38 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	成功返回Object对象各字段的详细信息，包含表信息，被依赖和依赖的表信息。

请求示例

按照ID查询Object。

GET <https://AstroZero域名/u-route/baas/metadata/v1.0/Object/cust000000aKtFZpERvc>

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "tableInfo": {
      "id": "cust000000aKtFZpERvc",
      "keyPrefix": "cRvc",
      "name": "test_dfg_CST",
      "category": 3,
      "label": "dfg",
      "pluralLabel": "dfgs",
      "version": 0,
      "contextSetup": 0,
      "trackFieldHistory": false,
      "allowReports": false,
      "allowActivities": false,
      "allowChatter": false,
      "allowStreamingApiAccess": false,
      "allowBulkApiAccess": false,
      "allowSearch": false,
      "allowSharing": false,
      "sharingMode": 0,
      "deploymentStatus": 0,
      "properties": "",
      "status": 0,
      "createdBy": "10gd000000SDyqaQW6uu",
      "createdDate": "2020-05-28 17:01:02",
      "lastModifiedBy": "10gd000000SDyqaQW6uu",
      "lastModifiedDate": "2020-05-28 17:01:02",
      "description": "",
      "objectType": 0,
      "protectMode": 0,
      "installedPackage": "",
      "createdBy.name": "test0xx",
      "lastModifiedBy.name": "test0xx"
    }
  }
}
```

```
},
"childObjects": null,
"parentObjects": [
  {
    "parentObject": "stad00000000000000GD",
    "cascadeDelete": false
  },
  {
    "parentObject": "stad00000000000000GG",
    "cascadeDelete": false
  },
  {
    "parentObject": "stup000000000000002I",
    "cascadeDelete": false
  },
  {
    "parentObject": "stup000000000000003L",
    "cascadeDelete": false
  }
]
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.3.4 根据 object_id 为 Object 添加字段

功能介绍

为object添加自定义字段。

URI

POST *AstroZero*域名/u-route/baas/metadata/v1.0/Object/{object_id}/Field

表 5-39 路径参数

参数	是否必选	参数类型	描述
Object_id	是	String	Object的ID，可通过调用 查询所有Object对象 接口查看（响应消息中的“id”）。

请求参数

表 5-40 请求 body 参数

参数	是否必选	参数类型	描述
name	是	String	字段名称。
label	是	String	字段标签。
inlineHelpText	否	String	字段的帮助说明。
dataLength	是	Int	字段的长度，取值为1~255。
dataType	是	String	字段的类型。
description	否	String	字段的描述。
defaultValue	否	String	默认值。
caseSensitive	否	Boolean	是否大小写敏感。
isRequired	否	Boolean	字段是否必填。
isUnique	否	Boolean	字段中数据是否不重复。
profiles	否	Object	权限分配。
layouts	否	String	对象字段布局页面名。
view	否	Object	每项为一个string数组，名为包含该字段的Layout，内容为该layout包含的字段。
isSearchable	否	Boolean	字段是否可搜索。

响应参数

表 5-41 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	string	成功返回会返回Object的一个ID值。

请求示例

根据object_id为Object添加字段。

POST <https://AstroZero.域名/u-route/baas/metadata/v1.0/Object/cust000000aKoa6I4E6K/Field>

```
{
  "label": "test",
  "name": "CNAME1__test__CST",
  "inlineHelpText": "dsf",
  "description": "fsdf",
  "dataType": "text",
  "dataLength": 255,
  "defaultValue": "vfdsg",
  "caseSensitive": true,
  "isRequired": true,
  "isUnique": true,
  "isSearchable": true,
  "profiles": [
    {
      "profile": "000T0000000000000002",
      "readable": true,
      "editable": true
    },
    {
      "profile": "000T0000000000000003",
      "readable": true,
      "editable": true
    },
    {
      "profile": "000T0000000000000004",
      "readable": true,
      "editable": true
    },
    {
      "profile": "000T0000000000000006",
      "readable": true,
      "editable": true
    },
    {
      "profile": "000T000000Tiq6fUQhjk",
      "readable": true,
      "editable": true
    },
    {
      "profile": "000T000000Tiq6UCs2y",
      "readable": true,
      "editable": true
    },
    {
      "profile": "000T000000W1a8iTFglU",
      "readable": true,
      "editable": true
    }
  ],
  "layouts": [
    "000D000000aKoa6OJOxk"
  ],
  "view": {
    "000D000000aKoa6OJOxk": {
      "fields": [
        "name",
        "test_df_CST",
        "test_sdadsd_CST",
        "test_sdd_CST"
      ]
    }
  }
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": "cust000000iY8i3Jvyng"
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.3.5 根据 object_id 新增 trigger

功能介绍

为Object对象添加对象触发器。

URI

POST AstroZero域名/u-route/baas/metadata/v1.0/Object/{object_id}/Trigger

表 5-42 路径参数

参数	是否必选	参数类型	描述
Object_id	是	String	Object的ID，可通过调用 查询所有Object对象 接口查看（响应消息中的“id”）。

请求参数

表 5-43 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	Trigger名称。
label	否	String	Trigger标签。
description	否	String	trigger的描述信息。
script	是	String	实现Trigger的脚本名。
jscode	否	string	Trigger下的javascript代码。
active	否	String	Trigger是否激活。

参数	是否必选	参数类型	描述
event	否	String	触发时机，可选择BeforeInsert, AfterInsert, BeforeUpdate, AfterUpdate, BeforeDelete或AfterDelete。

响应参数

表 5-44 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	string	成功返回会返回Object的一个ID值。

请求示例

根据object_id新增trigger。

```
POST https://AstroZero域名/u-route/baas/metadata/v1.0/Object/cust000000aKtFZpERvc/Trigger
{
  "name": "Trigger_01",
  "label": "Trigger 01",
  "description": null,
  "script": "string",
  "jscode": "Hello, world;\r\n",
  "active": true,
  "event": "BeforeInsert"
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "Success",
  "result": "001i000000SlrEi3E0a8"
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.3.6 根据 object_id 新增校验 validator

功能介绍

根据object_id新增对象校验器。在向对象中录入记录数据时，为了保证对象数据的有效性，可以定义一些校验规则，对字段值以及各字段之间的逻辑关系进行校验。例如，name字段不能为空，MinPrice字段不能大于maxPrice字段等。

URI

POST AstroZero域名/u-route/baas/metadata/v1.0/Object/{object_id}/Validator

表 5-45 路径参数

参数	是否必选	参数类型	描述
Object_id	是	String	Object的ID，可通过调用 查询所有Object对象 接口查看（响应消息中的“id”）。

请求参数

表 5-46 请求 Body 体参数

参数	是否必选	参数类型	描述
name	是	String	字段校验器名。
label	否	String	字段校验器标签。
description	否	String	字段校验器描述。
active	是	String	设置激活状态，如果存在该参数的情况下，会忽略其他参数，即激活/冻结时，不能同时修改其他属性。
errorExpression	否	String	校验器判断的规则表达式。
errorMessage	否	String	错误信息。
errorLocation	否	String	错误信息出现的位置。

响应参数

表 5-47 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	string	成功返回会返回Object的一个ID值。

请求示例

根据object_id新增对象校验器。

```
POST https://AstroZero域名/u-route/baas/metadata/v1.0/Object/cust000000aKtFZpERvc/Validator
{
  "active": true,
  "name": "age",
  "label": "age",
  "errorExpression": "age_CST>10",
  "errorMessage": "The age is greater than 10.",
  "errorLocation": "topOfPage"
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "Success",
  "result": "001j000000SNRBr3JnOq"
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.3.7 根据 object_id 新增 layout

功能介绍

按object_id新增layout。可以通过layout可视化各字段值，在新建字段时，把字段添加到具体某个layout中可以可视化这个字段。

URI

```
POST AstroZero域名/u-route/baas/metadata/v1.0/Object/{object_id}/Layout
```

表 5-48 路径参数

参数	是否必选	参数类型	描述
Object_id	是	String	Object的ID, 可通过调用 查询所有Object对象 接口查看（响应消息中的“id”）。

请求参数

表 5-49 请求 Body 体参数

参数	是否必选	参数类型	描述
name	是	String	创建表对象时的名称，一旦创建不可修改。
viewName	否	String	对象的别名，可以修改。
existingPageLayout	是	String	现有页面布局的ID，获取方法如下： 1. 进入已创建的应用。 2. 在模块树中，单击已创建的对象。 3. 在“布局”页签，即可获取页面布局的ID。

响应参数

响应参数如[表5-50](#)所示。

表 5-50 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	string	成功返回，会返回一个ID字符串。

请求示例

按object_id新增layout。

```
POST https://AstroZero域名/u-route/baas/metadata/v1.0/Object/cust000000aKoa6I4E6K/Layout
{
  "name": "dsaf",
  "existingPageLayout": "000D000000aKoa6OJOxl",
  "viewName": "dsaf"
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": "000D000000aL60EyXgUS"
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.4 Flow 对象的操作

5.2.4.1 创建 Flow 对象

功能介绍

添加一个服务编排（Flow）对象。

URI

POST AstroZero域名/u-route/baas/metadata/v1.0/Flow

请求参数

表 5-51 请求参数

参数	是否必选	参数类型	描述
name	是	String	服务编排的名称。
label	是	String	服务编排的标签。
processType	是	String	服务编排的类型，有两种可选“Event Trigger”和“Autolaunched Flow”。
startElement	否	String	服务编排的起始元素。
interviewLabel	否	String	服务编排实例的标签。
localUrl	是	String	本地url。

参数	是否必选	参数类型	描述
event	否	String	服务编排涉及的事件。
variables	否	Array	如果服务编排类型是“Event Trigger”，则可以填写绑定具体事件。

响应参数

表 5-52 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	成功返回，会返回服务编排的id。

请求示例

创建服务编排对象实例，服务编排对象的标签为“das”，名称为“t_das”。

POST `https://AstroZero域名/u-route/baas/metadata/v1.0/Flow`

```
{
  "label": "das",
  "name": "t_das",
  "processType": "Autolaunched Flow",
  "localUrl": "/run/t_das",
  "startElement": "",
  "event": "",
  "eventCriteria": "",
  "interviewLabel": "das {!$Flow.CurrentDateTime}"
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "errors": [
      "Flow(t_das) 的startElement为必选属性, 必须输入"
    ],
    "id": "000y000000aM8XixTN0S",
    "warnings": []
  }
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.4.2 查询 Flow 对象

功能介绍

获取匹配查询条件的服务编排列表。

URI

*GET AstroZero*域名/u-route/baas/metadata/v1.0/Flow

请求参数

表 5-53 请求参数

参数	是否必选	参数类型	描述
like	否	String	带上“like=true”，则表示name为模糊匹配，否则为精确匹配。 默认为精确匹配。
name	否	String	服务编排的名称。
processType	否	String	服务编排的类型，有两种可选“Event Trigger”和“Autolaunched Flow”。
active	否	String	服务编排是否已启用。
baseline	否	String	是否是系统预置的服务编排。
skip	否	String	带上skip，表示从多少条后开始查找“skip>=0”。
limit	否	String	带上limit，表示每次最大返回记录数“limit>=1”。

响应参数

表 5-54 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	array	成功会返回匹配条件的flow列表。

请求示例

按条件查询服务编排的实例。

```
GET https://AstroZero域名/u-route/baas/metadata/v1.0/Flow?name=testName &processType=Autolaunched  
Flow&active=true
```

响应示例

```
{  
  "resCode": "0",  
  "resMsg": "成功",  
  "result": [  
    {  
      "bolD": null,  
      "catalogue": "Out",  
      "createdBy": "testUserId",  
      "createdBy.__objectType": "User",  
      "createdBy.name": "test0xx",  
      "createdDate": "2019-08-04 16:21:58",  
      "custom": true,  
      "description": "Updated Description",  
      "id": "000y000000aM8XixTN0S",  
      "installedPackage": null,  
      "lastModifiedBy": "testUserId",  
      "lastModifiedBy.__objectType": "User",  
      "lastModifiedBy.name": "test0xx",  
      "lastModifiedDate": "2019-08-04 16:22:58",  
      "name": "testName",  
      "namespace": "testSpace",  
      "owner": "testUserId",  
      "owner.__objectType": "User",  
      "owner.name": "test0xx",  
      "protectMode": 0,  
      "structMembers": [  
        {  
          "createdBy": "testUserId",  
          "createdBy.__objectType": "User",  
          "createdBy.name": "test0xx",  
          "createdDate": "2019-08-04 16:21:58",  
          "currencyFactor": 1,  
          "custom": true,  
          "dataType": "text",  
          "defaultValue": null,  
          "defaultValueType": null,  
          "description": null,  
          "id": "000y000000bM2XixVA0P",  
          "ignoreNull": true,  
          "installedPackage": null,  
          "isCollection": true,  
          "isNested": false,  
          "lastModifiedBy": "testUserId",  
          "lastModifiedBy.__objectType": "User",  
          "lastModifiedBy.name": "test0xx",  
          "lastModifiedDate": "2019-08-04 16:21:58",  
          "name": "testName",  
          "namespace": null,  
          "owner": "testUserId",  
          "owner.__objectType": "User",  
          "owner.name": "test0xx",  
          "parent": "testId",  
          "parent.__objectType": "ExtStruct",  
          "parent.name": "TestName",  
          "picklistValues": null,  
          "protectMode": 0,  
          "required": true,  
          "structType": null  
        }  
      ]  
    }  
  ]  
}
```

```
]
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.4.3 按照 flow_id 查询/查询 flow_id 的入参出参信息

功能介绍

按flow_id查询服务编排，或查询flow_id的入参出参信息。

URI

GET AstroZero域名/u-route/baas/metadata/v1.0/Flow/{flow_id}

表 5-55 路径参数

参数	是否必选	参数类型	描述
flow_id	是	String	服务编排实例的id, 可通过调用 查询Flow对象 接口查看（响应消息中的“id”）。

请求参数

表 5-56 请求参数

参数	是否必选	参数类型	描述
flag	否	String	分以下四种情况： <ul style="list-style-type: none"> 不带flag，按Flow_id查询服务编排。 带flag= check，按Flow_id检查服务编排信息。 带flag=Describe，按flow_id查询单个服务编排的入参出参信息。 带flag= dep，按flow_id查询单个服务编排所依赖的对象。

响应参数

表 5-57 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	成功会返回服务编排的信息。

请求示例

按flow_id查询服务编排/查询flow_id的入参出参信息。

GET <https://AstroZero域名/u-route/baas/metadata/v1.0/Flow/000y000000aM8XixTN0S>

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "active": false,
    "catalogue": "",
    "compilerVersion": 0,
    "createdBy": "testUserId",
    "createdBy.__objectType": "User",
    "createdBy.name": "test0xx",
    "createdDate": "2019-08-03 18:19:13",
    "custom": true,
    "description": "",
    "event": "",
    "eventCriteria": "",
    "id": "testId",
    "installedPackage": null,
    "interviewLabel": "",
    "label": "A Brand New Label",
    "lastModifiedBy": "testUserId",
    "lastModifiedBy.__objectType": "User",
    "lastModifiedBy.name": "test0xx",
    "lastModifiedDate": "2019-08-04 17:57:29",
    "localUrl": "",
    "lockUser": "",
    "name": "testName",
    "owner": "testUserId",
    "owner.__objectType": "User",
    "owner.name": "test0xx",
    "private": false,
    "processType": "",
    "protectMode": 0,
    "reserved": 0,
    "startElement": "",
    "status": "Error",
    "usingSLA": null,
    "version": "1.0.1"
  }
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.4.4 根据 flow_id 修改 Flow 对象

功能介绍

根据id修改服务编排对象。

URI

PUT *AstroZero*域名/u-route/baas/metadata/v1.0/Flow/{flow_id}

表 5-58 路径参数

参数	是否必选	参数类型	描述
Flow_id	是	String	服务编排实例的ID，可通过调用 查询Flow对象 接口查看（响应消息中的“id”）。

请求参数

表 5-59 请求参数

参数	是否必选	参数类型	描述
label	是	String	服务编排的标签。
processType	是	String	服务编排的类型。 <ul style="list-style-type: none"> Autolaunched Flow: 自启动服务编排，在接口调用后会立即执行服务编排模型定义的逻辑。 Event Trigger: 事件触发的服务编排，则会在事件触发时才会开始执行服务编排模型定义的逻辑。当选择该类型时，您需要配置具体的事件以及条件规则。
startElement	否	String	服务编排的起始元素。
interviewLabel	否	String	服务编排实例的标签。
localUrl	否	String	本地url。

参数	是否必选	参数类型	描述
event	否	String	服务编排涉及的事件。
variables	否	Array	如果服务编排类型是Event Trigger, 则可以填写绑定具体事件。
formulas	否	Array of objects	服务编排中定义的公式。
choices	否	Array of objects	服务编排中的选择判断。
structs	否	Array of objects	服务编排中定义绑定的结构体。
constants	否	Array of objects	常量。
structVariables	否	Array of objects	结构体变量。
objectVariables	否	Array of objects	对象变量。
screens	否	Array of objects	服务编排输入的显示界面。
localUrl	否	String	服务编排接口本地URL。
compilerVersion	否	String	最终版本。
catalogue	否	String	服务编排的分类。
private	否	String	是否私有。私有的服务编排不能通过Rest API或者自定义API（公共接口）进行调用。 <ul style="list-style-type: none"> • true • false

响应参数

响应参数如[表5-60](#)所示。

表 5-60 响应参数

参数	参数类型	描述
resCode	String	返回码, 如果返回“0”代表请求成功, 其他错误码说明请参考 错误码 。

参数	参数类型	描述
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	返回服务编排的错误和警告信息。

请求示例

修改服务编排对象实例。

```
PUT https://AstroZero域名/u-route/baas/metadata/v1.0/Flow/000y000000aM8XixTN0S
{
  "label": "updated label",
  "processType": "Autolaunched Flow"
}
```

响应示例

```
{
  "resCode": "XXX",
  "resMsg": "失败",
  "result": {
    "errors": [
      "string"
    ],
    "warnings": [
      "string"
    ]
  }
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.4.5 按照 flow_id 删除 flow

功能介绍

按flow_id删除服务编排。

URI

DELETE *AstroZero*域名/u-route/baas/metadata/v1.0/Flow/{flow_id}

表 5-61 路径参数

参数	是否必选	参数类型	描述
flow_id	是	String	服务编排实例的 ID，可通过调用 查询Flow对象 接口查看（响应消息中的“id”）。

请求参数

无

响应参数

表 5-62 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	string	成功后，会返回数字1。

请求示例

删除服务编排实例。

DELETE <https://AstroZero域名/u-route/baas/metadata/v1.0/Flow/000y000000aM8XixTN0S>

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": "1"
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.5 BPM 对象的操作

5.2.5.1 创建 BPM 对象

功能介绍

创建一个流程编排（BPM）对象。

URI

POST AstroZero域名/u-route/baas/bp/v2.0/metadata/definitions

请求参数

表 5-63 请求参数

参数	是否必选	参数类型	描述
name	是	String	BPM的名称。
label	是	String	BPM的标签。
description	否	String	BPM的描述文档。
version	否	String	版本号。
instanceLabel	否	String	实例标签，可以内嵌系统变量。
laneSet	否	object	泳道信息。
config	否	object	配置信息。
startEvents	否	Array of object	开始事件定义。
dataMappers	否	Array of object	数据映射定义。
userTasks	否	Array of object	用户任务定义。
serviceTasks	否	Array of object	自动任务定义。
businessRuleTasks	否	Array of object	决策表任务定义。
callActivities	否	Array of object	活动调用定义。
subProcesses	否	Array of object	内嵌子流程定义。
throwEvents	否	Array of object	抛出事件定义。
catchEvents	否	Array of object	捕捉事件定义。
exclusiveGateways	否	Array of object	排他网关定义。
parallelGateways	否	Array of object	并行网关定义。
eventGateways	否	Array of object	事件网关定义。

参数	是否必选	参数类型	描述
usingSLA	否	Array of object	BPM引用的SLA。
slas	否	Array of object	SLA定义。
constants	否	Array of object	常量定义。
variables	否	Array of object	变量定义。

响应参数

表 5-64 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	成功会返回BPM的一个ID值，失败会返回错误和警告信息。

请求示例

创建BPM实例。

POST <https://AstroZero域名/u-route/baas/bp/v2.0/metadata/definitions>

```
{
  "config": {
    "cancelable": false,
    "noCancelAfterElem": ""
  },
  "dataMappers": [
    {
      "assignmentItems": [
        {
          "assignTo": "dateList",
          "name": "",
          "operator": "add",
          "value": "2019-03-04",
          "valueType": ""
        },
        {
          "assignTo": "dateList",
          "name": "",
          "operator": "add",
          "value": "2020-01-02",
          "valueType": ""
        }
      ]
    },
    {
      "connector": "userTask",
      "connectorPoint": "{\r\n\"fromPort\": \"L1\", \"toPort\": \"L1\", \"points\": [\r\n\"619 125\", \"643 125\", \"662.5 125\", \"662.5 100\", \"682 100\", \"706 100\"]}",
      "description": ""
    }
  ]
}
```

```

        "label": "Data Mapper",
        "locationX": 575,
        "locationY": 125,
        "name": "dataMapper",
        "connectors": []
    }
],
"description": "",
"label": "SimpleBP",
"laneSet": {
    "label": "",
    "lanes": [
        {
            "elements": [
                "userTask",
                "userTask1",
                "dataMapper",
                "start"
            ],
            "height": 201.35220031738282,
            "label": "lane",
            "name": "lane",
            "role": "LoanRequest.Applicant",
            "width": 1080
        }
    ],
    "name": ""
},
"name": "SimpleBP",
"slas": [
    {
        "active": true,
        "criteria": "",
        "deadlineDays": 0,
        "deadlineHours": 0,
        "deadlineIncreaseUrgencyBy": 2,
        "deadlineMinutes": 0,
        "deadlineOnlyBusinessDays": false,
        "goalActions": [
            {
                "action": "",
                "criteria": "{!IsMatched} && {!Owner}",
                "name": "sla-act",
                "type": "Notify Manager"
            }
        ],
        "goalDays": 0,
        "goalHours": 0,
        "goalIncreaseUrgencyBy": 1,
        "goalMinutes": 0,
        "goalOnlyBusinessDays": false,
        "label": "userTask1_sla",
        "name": "userTask1_sla",
        "startWhen": "",
        "variable": "",
        "actions": []
    },
    {
        "active": true,
        "criteria": "",
        "deadlineDays": 0,
        "deadlineHours": 0,
        "deadlineIncreaseUrgencyBy": 2,
        "deadlineMinutes": 0,
        "deadlineOnlyBusinessDays": false,
        "goalActions": [
            {
                "action": "NotifyBPManager",
                "criteria": "",
            }
        ]
    }
]

```



```

        "name": "sla-act1",
        "type": "Notify Manager"
    }
],
"goalDays": 1,
"goalHours": 0,
"goalIncreaseUrgencyBy": 1,
"goalMinutes": 0,
"goalOnlyBusinessDays": false,
"label": "userTask_sla",
"name": "userTask_sla",
"startWhen": "",
"variable": "",
"actions": []
}
],
"startEvents": [
    {
        "connector": "dataMapper",
        "connectorPoint": "{\"fromPort\":\"R1\",\"toPort\":\"L0\",\"points\":[\"151.5 100\",\"175.5 100\",\"341.25 100\",\"341.25 110.5\",\"507 110.5\",\"531 110.5\"]}",
        "label": "Start",
        "locationX": 125,
        "locationY": 100,
        "name": "start",
        "type": "Start",
        "startType": "",
        "renderType": "",
        "render": "",
        "document": "",
        "description": ""
    }
],
"startStage": null,
"startType": "",
"status": "Warning",
"status.name": "Warning",
"userTasks": [
    {
        "approvalType": "Single",
        "connector": "userTask1",
        "connectorPoint": "{\"fromPort\":\"R1\",\"toPort\":\"L1\",\"points\":[\"794 100\",\"818 100\",\"875 100\",\"875 100\",\"932 100\",\"956 100\"]}",
        "description": "wuli desc",
        "groupAsSingleParticipant": false,
        "inputAssignments": [
            {
                "input": "$Task.Title",
                "name": "",
                "value": "{$GlobalConstant.EmptyString}",
                "valueType": ""
            }
        ],
        "label": "User Task",
        "locationX": 750,
        "locationY": 100,
        "name": "userTask",
        "participants": [
            {
                "type": "user",
                "value": "ecommerce6"
            }
        ],
        "preActions": [
            {
                "action": "{$GlobalConstant.EmptyString}",
                "criteria": "",
                "name": "action",
                "type": "Set Status"
            }
        ]
    }
]

```

```
    }
  ],
  "render": "SimpleBP$form1$form",
  "renderType": "Form",
  "routeType": "Name and Expressions",
  "router": {
    "defaultRouteTo": "",
    "defaultRouteType": "",
    "name": ""
  },
  "title": "wu li tu cao",
  "usingSLA": "userTask_sla",
  "routeTo": "",
  "faultConnector": "",
  "boundaryEvents": [],
  "outputAssignments": [],
  "document": "",
  "priority": 3
},
{
  "approvalType": "Single",
  "groupAsSingleParticipant": false,
  "label": "User Task",
  "locationX": 1000,
  "locationY": 100,
  "name": "userTask1",
  "priority": 3,
  "render": "SimpleBP$ut1$form",
  "renderType": "Form",
  "routeType": "Last Assignee In Lane",
  "router": {
    "defaultRouteTo": "",
    "defaultRouteType": "",
    "name": ""
  },
  "title": "T2",
  "routeTo": "",
  "connector": "",
  "faultConnector": "",
  "boundaryEvents": [],
  "inputAssignments": [],
  "outputAssignments": [],
  "document": "",
  "description": ""
}
],
"variables": [
  {
    "currencyFactor": 0,
    "dataType": "checkBox",
    "defaultValue": "${GlobalConstant.True}",
    "defaultValueType": "",
    "description": "",
    "extNS": "",
    "extType": "",
    "isCollection": false,
    "isInput": false,
    "isNested": false,
    "isOutput": false,
    "name": "IsMatched"
  },
  {
    "currencyFactor": 0,
    "dataType": "text",
    "defaultValue": "Hello",
    "defaultValueType": "",
    "description": "",
    "extNS": "",
    "extType": ""
  }
]
```

```
    "isCollection": false,
    "isInput": false,
    "isNested": false,
    "isOutput": false,
    "name": "Owner"
  },
  {
    "currencyFactor": 0,
    "dataType": "date",
    "defaultValue": "",
    "defaultValueType": "",
    "description": "",
    "extNS": "",
    "extType": "",
    "isCollection": false,
    "isInput": false,
    "isNested": false,
    "isOutput": false,
    "name": "dateVar"
  },
  {
    "currencyFactor": 0,
    "dataType": "date",
    "defaultValue": "",
    "defaultValueType": "",
    "description": "",
    "extNS": "",
    "extType": "",
    "isCollection": true,
    "isInput": false,
    "isNested": false,
    "isOutput": false,
    "name": "dateList"
  },
  {
    "currencyFactor": 0,
    "dataType": "extStruct",
    "defaultValue": "",
    "defaultValueType": "",
    "description": "",
    "extNS": "SimpleBP",
    "extType": "Form1Form",
    "isCollection": false,
    "isInput": false,
    "isNested": false,
    "isOutput": false,
    "name": "Form1Form"
  }
],
"version": "1.0.1",
"webSite": null,
"constants": [],
"serviceTasks": [],
"businessRuleTasks": [],
"callActivities": [],
"subProcesses": [],
"endEvents": [],
"sendEvents": [],
"throwEvents": [],
"catchEvents": [],
"exclusiveGateways": [],
"parallelGateways": [],
"eventGateways": [],
"usingSLA": ""
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "errors": [],
    "id": "000y000000aM8XixTN0S",
    "warnings": []
  }
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.5.2 按条件查询 BPM 对象

功能介绍

获取匹配查询条件的BPM列表。

URI

GET AstroZero域名/u-route/baas/bp/v2.0/metadata/definitions

请求参数

表 5-65 请求参数

参数	是否必选	参数类型	描述
like	否	String	带上“like=true”，表示name为模糊匹配，否则为精确匹配。 默认为精确匹配。
name	否	String	BPM的名称。
active	否	String	BPM是否被激活。
skip	否	String	带上skip，从多少条后开始查找 “skip>=0”。
limit	否	String	带上limit，每次最大返回记录数 “limit>=1”。

响应参数

表 5-66 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	array	成功会返回匹配条件的BPM列表。

请求示例

按条件查询BPM的实例。

GET <https://AstroZero域名/u-route/baas/bp/v2.0/metadata/definitions?name=BP&active=false>

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": [
    {
      "active": false,
      "config": "{\"cancelable\":false,\"noCancelAfterElem\":\"\",\"noCancelAfterElemLabel\":\"\"}",
      "description": "",
      "id": "001L000000Peuw1Q318a",
      "label": "SimpleBP",
      "name": "SimpleBP",
      "status": "Warning",
      "version": "1.0.1"
    }
  ]
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.5.3 按 ID 查询 BPM 对象

功能介绍

按bpm_id查询BPM对象。

URI

GET AstroZero域名/u-route/baas/bp/v2.0/metadata/definitions/{bpm_id}

表 5-67 路径参数

参数	是否必选	参数类型	描述
bpm_id	是	String	BPM的实例ID，可通过调用 按条件查询BPM对象接口 查看（响应消息中的“id”）。

请求参数

无

响应消息

表 5-68 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	成功会返回BPM的信息。

请求示例

查询BPM对象实例。

GET <https://AstroZero域名/u-route/baas/bp/v2.0/metadata/definitions/001L000000Peuw1Q318a>

响应示例

```
{
  "id": "001L000000Peuw1Q318a",
  "config": {
    "cancelable": false,
    "noCancelAfterElem": ""
  },
  "dataMappers": [
    {
      "assignmentItems": [
        {
          "assignTo": "dateList",
          "name": "",
          "operator": "add",
          "value": "2019-03-04",
          "valueType": ""
        },
        {
          "assignTo": "dateList",
          "name": "",
          "operator": "add",
          "value": "2020-01-02",
          "valueType": ""
        }
      ]
    }
  ]
}
```

```
    }
  ],
  "connector": "userTask",
  "connectorPoint": "{\n  \"fromPort\": \"R1\", \"toPort\": \"L1\", \"points\": [\n    \"619 125\", \"643 125\", \"662.5 125\", \"662.5 100\", \"682 100\", \"706 100\"]\n}",
  "description": "",
  "label": "Data Mapper",
  "locationX": 575,
  "locationY": 125,
  "name": "dataMapper",
  "connectors": []
}
],
"description": "",
"label": "SimpleBP",
"laneSet": {
  "label": "",
  "lanes": [
    {
      "elements": [
        "userTask",
        "userTask1",
        "dataMapper",
        "start"
      ],
      "height": 201.35220031738282,
      "label": "lane",
      "name": "lane",
      "role": "LoanRequest.Applicant",
      "width": 1080
    }
  ],
  "name": ""
},
"name": "SimpleBP",
"slas": [
  {
    "active": true,
    "criteria": "",
    "deadlineDays": 0,
    "deadlineHours": 0,
    "deadlineIncreaseUrgencyBy": 2,
    "deadlineMinutes": 0,
    "deadlineOnlyBusinessDays": false,
    "goalActions": [
      {
        "action": "",
        "criteria": "{!IsMatched} && {!Owner}",
        "name": "sla-act",
        "type": "Notify Manager"
      }
    ],
    "goalDays": 0,
    "goalHours": 0,
    "goalIncreaseUrgencyBy": 1,
    "goalMinutes": 0,
    "goalOnlyBusinessDays": false,
    "label": "userTask1_sla",
    "name": "userTask1_sla",
    "startWhen": "",
    "variable": "",
    "actions": []
  },
  {
    "active": true,
    "criteria": "",
    "deadlineDays": 0,
    "deadlineHours": 0,
    "deadlineIncreaseUrgencyBy": 2,
```

```
"deadlineMinutes": 0,
"deadlineOnlyBusinessDays": false,
"goalActions": [
  {
    "action": "NotifyBPManager",
    "criteria": "",
    "name": "sla-act1",
    "type": "Notify Manager"
  }
],
"goalDays": 1,
"goalHours": 0,
"goalIncreaseUrgencyBy": 1,
"goalMinutes": 0,
"goalOnlyBusinessDays": false,
"label": "userTask_sla",
"name": "userTask_sla",
"startWhen": "",
"variable": "",
"actions": []
},
"startEvents": [
  {
    "connector": "dataMapper",
    "connectorPoint": "{\"fromPort\":\"R1\",\"toPort\":\"L0\",\"points\":[\"151.5 100\",\"175.5 100\",\"341.25 100\",\"341.25 110.5\",\"507 110.5\",\"531 110.5\"]}",
    "label": "Start",
    "locationX": 125,
    "locationY": 100,
    "name": "start",
    "type": "Start",
    "startType": "",
    "renderType": "",
    "render": "",
    "document": "",
    "description": ""
  }
],
"startStage": null,
"startType": "",
"status": "Warning",
"status.name": "Warning",
"userTasks": [
  {
    "approvalType": "Single",
    "connector": "userTask1",
    "connectorPoint": "{\"fromPort\":\"R1\",\"toPort\":\"L1\",\"points\":[\"794 100\",\"818 100\",\"875 100\",\"875 100\",\"932 100\",\"956 100\"]}",
    "description": "wuli desc",
    "groupAsSingleParticipant": false,
    "inputAssignments": [
      {
        "input": "$Task.Title",
        "name": "",
        "value": "${!$GlobalConstant.EmptyString}",
        "valueType": ""
      }
    ]
  },
  {
    "label": "User Task",
    "locationX": 750,
    "locationY": 100,
    "name": "userTask",
    "participants": [
      {
        "type": "user",
        "value": "ecommerce6"
      }
    ]
  }
],
```



```

"preActions": [
  {
    "action": "{!$GlobalConstant.EmptyString}",
    "criteria": "",
    "name": "action",
    "type": "Set Status"
  }
],
"render": "SimpleBP$form1$form",
"renderType": "Form",
"routeType": "Name and Expressions",
"router": {
  "defaultRouteTo": "",
  "defaultRouteType": "",
  "name": ""
},
"title": "wu li tu cao",
"usingSLA": "userTask_sla",
"routeTo": "",
"faultConnector": "",
"boundaryEvents": [],
"outputAssignments": [],
"document": "",
"priority": 3
},
{
  "approvalType": "Single",
  "groupAsSingleParticipant": false,
  "label": "User Task",
  "locationX": 1000,
  "locationY": 100,
  "name": "userTask1",
  "priority": 3,
  "render": "SimpleBP$ut1$form",
  "renderType": "Form",
  "routeType": "Last Assignee In Lane",
  "router": {
    "defaultRouteTo": "",
    "defaultRouteType": "",
    "name": ""
  },
  "title": "T2",
  "routeTo": "",
  "connector": "",
  "faultConnector": "",
  "boundaryEvents": [],
  "inputAssignments": [],
  "outputAssignments": [],
  "document": "",
  "description": ""
}
],
"variables": [
  {
    "currencyFactor": 0,
    "dataType": "checkBox",
    "defaultValue": "{!$GlobalConstant.True}",
    "defaultValueType": "",
    "description": "",
    "extNS": "",
    "extType": "",
    "isCollection": false,
    "isInput": false,
    "isNested": false,
    "isOutput": false,
    "name": "IsMatched"
  },
  {
    "currencyFactor": 0,

```

```
    "dataType": "text",
    "defaultValue": "Hello",
    "defaultValueType": "",
    "description": "",
    "extNS": "",
    "extType": "",
    "isCollection": false,
    "isInput": false,
    "isNested": false,
    "isOutput": false,
    "name": "Owner"
  },
  {
    "currencyFactor": 0,
    "dataType": "date",
    "defaultValue": "",
    "defaultValueType": "",
    "description": "",
    "extNS": "",
    "extType": "",
    "isCollection": false,
    "isInput": false,
    "isNested": false,
    "isOutput": false,
    "name": "dateVar"
  },
  {
    "currencyFactor": 0,
    "dataType": "date",
    "defaultValue": "",
    "defaultValueType": "",
    "description": "",
    "extNS": "",
    "extType": "",
    "isCollection": true,
    "isInput": false,
    "isNested": false,
    "isOutput": false,
    "name": "dateList"
  },
  {
    "currencyFactor": 0,
    "dataType": "extStruct",
    "defaultValue": "",
    "defaultValueType": "",
    "description": "",
    "extNS": "SimpleBP",
    "extType": "Form1Form",
    "isCollection": false,
    "isInput": false,
    "isNested": false,
    "isOutput": false,
    "name": "Form1Form"
  }
],
"version": "1.0.1",
"usingSLA": ""
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.5.4 根据 ID 修改 BPM 对象

功能介绍

根据id修改BPM对象。

URI

PUT AstroZero域名/u-route/baas/bp/v2.0/metadata/definitions/{bpm_id}

表 5-69 路径参数

参数	是否必选	参数类型	描述
bpm_id	是	String	BPM实例的ID，可通过调用 按条件查询BPM对象接口查看 （响应消息中的“id”）。

请求参数

表 5-70 请求参数

参数	是否必选	参数类型	描述
name	是	String	BPM的名称。
label	是	String	BPM的标签。
description	否	String	BPM的描述文档。
version	否	String	版本号
instanceLabel	否	String	实例标签，可以内嵌系统变量。
laneSet	否	object	泳道信息。
config	否	object	配置信息。
startEvents	否	Array of object	开始事件定义。
dataMappers	否	Array of object	数据映射定义。
userTasks	否	Array of object	用户任务定义。
serviceTasks	否	Array of object	自动任务定义。
businessRuleTasks	否	Array of object	决策表任务定义。
callActivities	否	Array of object	活动调用定义。
subProcesses	否	Array of object	内嵌子流程定义。

参数	是否必选	参数类型	描述
throwEvents	否	Array of object	抛出事件定义。
catchEvents	否	Array of object	捕捉事件定义。
exclusiveGateways	否	Array of object	排他网关定义。
parallelGateways	否	Array of object	并行网关定义。
eventGateways	否	Array of object	事件网关定义。
usingSLA	否	Array of object	BPM引用的SLA。
slas	否	Array of object	SLA定义。
constants	否	Array of object	常量定义。
variables	否	Array of object	变量定义。

响应参数

表 5-71 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	成功会返回BPM的信息。

请求示例

修改BPM对象实例。

PUT <https://AstroZero域名/u-route/baas/bp/v2.0/metadata/definitions/001L000000Peuw1Q318a>

```
{
  "config": {
    "cancelable": false,
    "noCancelAfterElem": ""
  },
  "dataMappers": [
    {
      "assignmentItems": [
        {
          "assignTo": "dateList",
          "name": "",
          "operator": "add",
          "value": "2019-03-04",

```

```

        "valueType": ""
    },
    {
        "assignTo": "dateList",
        "name": "",
        "operator": "add",
        "value": "2020-01-02",
        "valueType": ""
    }
],
"connector": "userTask",
"connectorPoint": "{\"fromPort\":\"R1\",\"toPort\":\"L1\",\"points\":[[\"619 125\",\"643 125\",\"662.5 125\",\"662.5 100\",\"682 100\",\"706 100\"]]}",
"description": "",
"label": "Data Mapper",
"locationX": 575,
"locationY": 125,
"name": "dataMapper",
"connectors": []
}
],
"description": "",
"label": "SimpleBP",
"laneSet": {
    "label": "",
    "lanes": [
        {
            "elements": [
                "userTask",
                "userTask1",
                "dataMapper",
                "start"
            ],
            "height": 201.35220031738282,
            "label": "lane",
            "name": "lane",
            "role": "LoanRequest.Applicant",
            "width": 1080
        }
    ],
    "name": ""
},
"name": "SimpleBP",
"slas": [
    {
        "active": true,
        "criteria": "",
        "deadlineDays": 0,
        "deadlineHours": 0,
        "deadlineIncreaseUrgencyBy": 2,
        "deadlineMinutes": 0,
        "deadlineOnlyBusinessDays": false,
        "goalActions": [
            {
                "action": "",
                "criteria": "{!IsMatched} && {!Owner}",
                "name": "sla-act",
                "type": "Notify Manager"
            }
        ]
    },
    {
        "goalDays": 0,
        "goalHours": 0,
        "goalIncreaseUrgencyBy": 1,
        "goalMinutes": 0,
        "goalOnlyBusinessDays": false,
        "label": "userTask1_sla",
        "name": "userTask1_sla",
        "startWhen": "",
        "variable": ""
    }
]

```

```
"actions": []
},
{
  "active": true,
  "criteria": "",
  "deadlineDays": 0,
  "deadlineHours": 0,
  "deadlineIncreaseUrgencyBy": 2,
  "deadlineMinutes": 0,
  "deadlineOnlyBusinessDays": false,
  "goalActions": [
    {
      "action": "NotifyBPMManager",
      "criteria": "",
      "name": "sla-act1",
      "type": "Notify Manager"
    }
  ],
  "goalDays": 1,
  "goalHours": 0,
  "goalIncreaseUrgencyBy": 1,
  "goalMinutes": 0,
  "goalOnlyBusinessDays": false,
  "label": "userTask_sla",
  "name": "userTask_sla",
  "startWhen": "",
  "variable": "",
  "actions": []
}
],
"startEvents": [
  {
    "connector": "dataMapper",
    "connectorPoint": "{\"fromPort\":\"R1\",\"toPort\":\"L0\",\"points\":[\"151.5 100\",\"175.5 100\",\"341.25 100\",\"341.25 110.5\",\"507 110.5\",\"531 110.5\"]}",
    "label": "Start",
    "locationX": 125,
    "locationY": 100,
    "name": "start",
    "type": "Start",
    "startType": "",
    "renderType": "",
    "render": "",
    "document": "",
    "description": ""
  }
],
"startStage": null,
"startType": "",
"status": "Warning",
"status.name": "Warning",
"userTasks": [
  {
    "approvalType": "Single",
    "connector": "userTask1",
    "connectorPoint": "{\"fromPort\":\"R1\",\"toPort\":\"L1\",\"points\":[\"794 100\",\"818 100\",\"875 100\",\"875 100\",\"932 100\",\"956 100\"]}",
    "description": "wuli desc",
    "groupAsSingleParticipant": false,
    "inputAssignments": [
      {
        "input": "$Task.Title",
        "name": "",
        "value": "{!$GlobalConstant.EmptyString}",
        "valueType": ""
      }
    ]
  }
],
"label": "User Task",
"locationX": 750,
```

```
"locationY": 100,
"name": "userTask",
"participants": [
  {
    "type": "user",
    "value": "ecommerce6"
  }
],
"preActions": [
  {
    "action": "{!$GlobalConstant.EmptyString}",
    "criteria": "",
    "name": "action",
    "type": "Set Status"
  }
],
"render": "SimpleBP$form1$form",
"renderType": "Form",
"routeType": "Name and Expressions",
"router": {
  "defaultRouteTo": "",
  "defaultRouteType": "",
  "name": ""
},
"title": "wu li tu cao",
"usingSLA": "userTask_sla",
"routeTo": "",
"faultConnector": "",
"boundaryEvents": [],
"outputAssignments": [],
"document": "",
"priority": 3
},
{
  "approvalType": "Single",
  "groupAsSingleParticipant": false,
  "label": "User Task",
  "locationX": 1000,
  "locationY": 100,
  "name": "userTask1",
  "priority": 3,
  "render": "SimpleBP$ut1$form",
  "renderType": "Form",
  "routeType": "Last Assignee In Lane",
  "router": {
    "defaultRouteTo": "",
    "defaultRouteType": "",
    "name": ""
  },
  "title": "T2",
  "routeTo": "",
  "connector": "",
  "faultConnector": "",
  "boundaryEvents": [],
  "inputAssignments": [],
  "outputAssignments": [],
  "document": "",
  "description": ""
}
],
"variables": [
  {
    "currencyFactor": 0,
    "dataType": "checkBox",
    "defaultValue": "{!$GlobalConstant.True}",
    "defaultValueType": "",
    "description": "",
    "extNS": "",
    "extType": ""
  }
]
```

```
    "isCollection": false,
    "isInput": false,
    "isNested": false,
    "isOutput": false,
    "name": "IsMatched"
  },
  {
    "currencyFactor": 0,
    "dataType": "text",
    "defaultValue": "Hello",
    "defaultValueType": "",
    "description": "",
    "extNS": "",
    "extType": "",
    "isCollection": false,
    "isInput": false,
    "isNested": false,
    "isOutput": false,
    "name": "Owner"
  },
  {
    "currencyFactor": 0,
    "dataType": "date",
    "defaultValue": "",
    "defaultValueType": "",
    "description": "",
    "extNS": "",
    "extType": "",
    "isCollection": false,
    "isInput": false,
    "isNested": false,
    "isOutput": false,
    "name": "dateVar"
  },
  {
    "currencyFactor": 0,
    "dataType": "date",
    "defaultValue": "",
    "defaultValueType": "",
    "description": "",
    "extNS": "",
    "extType": "",
    "isCollection": true,
    "isInput": false,
    "isNested": false,
    "isOutput": false,
    "name": "dateList"
  },
  {
    "currencyFactor": 0,
    "dataType": "extStruct",
    "defaultValue": "",
    "defaultValueType": "",
    "description": "",
    "extNS": "SimpleBP",
    "extType": "Form1Form",
    "isCollection": false,
    "isInput": false,
    "isNested": false,
    "isOutput": false,
    "name": "Form1Form"
  }
],
"version": "1.0.1",
"usingSLA": ""
}
```


响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "errors": [
      "string"
    ],
    "warnings": [
      "string"
    ]
  }
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.5.5 按照 ID 删除 BPM

功能介绍

按bpm_id删除BPM。

URI

DELETE *AstroZero*域名/u-route/baas/bp/v2.0/metadata/definitions/{bpm_id}

表 5-72 路径参数

参数	是否必选	参数类型	描述
bpm_id	是	String	BPM实例的ID，可通过调用 按条件查询BPM对象 接口查看（响应消息中的“id”）。

请求参数

无

响应参数

表 5-73 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	string	成功后，会返回数字1。

请求示例

删除BPM实例。

```
DELETE https://AstroZero域名/u-route/baas/bp/v2.0/metadata/definitions/001L000000Peuw1Q318a
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": "1"
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.6 Picklist 对象的操作

5.2.6.1 创建 Picklist 对象

功能介绍

创建一个Picklist对象。

URI

POST AstroZero域名/u-route/baas/metadata/v1.0/Picklist

请求参数

表 5-74 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	选项列表值集名称。 注意 若环境中设置了命名空间，请在名称中加上前缀“命名空间_”。
masterLabel	是	String	选项列表值集的标签。
description	否	String	选项列表值集的描述。
useFirstAsDefault	否	String	是否使用第一个值作为默认值。
picklistValues	否	Array	添加一个或多个选项列表值，取值中禁止出现分号“;”。

响应参数

表 5-75 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	string	成功会返回一个状态码。

请求示例

创建Picklist对象实例。

POST `https://AstroZero域名/u-route/baas/metadata/v1.0/Picklist`

```
{
  "name": "CNAME1__testName",
  "picklistValues": [
    {
      "label": "A",
      "value": "A"
    },
    {
      "label": "B",
      "value": "B"
    },
    {
      "label": "C",
      "value": "C"
    }
  ]
}
```

```
],
  "masterLabel": "Label for pick list ABC"
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": 0
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.6.2 查询选项列表值集

功能介绍

查询选项列表值集。

URI

GET AstroZero域名/u-route/baas/metadata/v1.0/Picklist

请求参数

表 5-76 请求参数

参数	是否必选	参数类型	描述
flag	否	String	查询选项列表值集列表的两种显示方式。 <ul style="list-style-type: none"> flag=A: 表示查询所有选项列表值集, 包含全局选项列表值集和字段级选项列表值集。 不携带flag, 可根据条件进行查询。
project	否	String	Picklist所属的项目名称。
type	否	String	Picklist所属的项目类型。
name	否	String	Picklist的名称。
isGlobal	否	String	是否为全局Picklist。 <ul style="list-style-type: none"> true false, 默认为false

响应参数

表 5-77 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Array	成功会返回匹配条件的选项列表值集。

请求示例

按条件查询Picklist的实例。

GET <https://AstroZero域名/u-route/baas/metadata/v1.0/Picklist?name=testName&isGlobal=false>

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": [
    {
      "controlField": null,
      "createdBy": "testUserId",
      "createdBy.__objectType": "User",
      "createdBy.name": "test0xx",
      "createdDate": "2019-08-05 20:38:24",
      "custom": true,
      "description": "",
      "id": "testListId",
      "installedPackage": null,
      "isGlobal": true,
      "isRestricted": false,
      "isSorted": false,
      "lastModifiedBy": "testUserId",
      "lastModifiedBy.__objectType": "User",
      "lastModifiedBy.name": "test0xx",
      "lastModifiedDate": "2019-08-05 20:38:24",
      "masterLabel": "testLabel",
      "name": "testName",
      "owner": "testUserId",
      "owner.__objectType": "User",
      "owner.name": "test0xx",
      "protectMode": 0
    }
  ]
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.6.3 按 ID 增加选项列表值集的值

功能介绍

按picklist_id增加选项列表值集的值。

URI

POST AstroZero域名/u-route/baas/metadata/v1.0/Picklist/{Picklist_id}

表 5-78 路径参数

参数	是否必选	参数类型	描述
Picklist_id	是	String	Picklist实例的ID, 可通过调用 查询选项列表值集 接口查看 (响应消息中的“id”)。

请求参数

表 5-79 请求参数

参数	是否必选	参数类型	描述
picklistValues	是	Array	添加一个或多个选项列表值, 取值中禁止出现分号“;”, 且不得包含待操作的选项列表值集中的任何一个值。

响应参数

表 5-80 响应参数

参数	参数类型	描述
resCode	String	返回码, 如果返回“0”代表请求成功, 其他错误码说明请参考 错误码 。
resMsg	String	返回消息, 如果成功状态, 通常会返回“Success”, 其他情况会返回具体的错误信息。
result	String	成功后, 会返回数字0。

请求示例

根据ID增加选项列表值集的值。

```
POST https://AstroZero域名/u-route/baas/metadata/v1.0/Picklist/000y000000aM92DCu6hU
{
```

```
"picklistValues": [  
  {  
    "value": "F",  
    "label": "ff"  
  }  
]
```

响应示例

```
{  
  "resCode": "0",  
  "resMsg": "成功",  
  "result": 0  
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.6.4 根据 ID 查询选项列表值集或引用该选项列表值集的对象字段列表

功能介绍

根据ID查询选项列表值集，或引用该选项列表值集的对象字段列表。

URI

GET AstroZero域名/u-route/baas/metadata/v1.0/Picklist/{Picklist_id}

表 5-81 路径参数

参数	是否必选	参数类型	描述
Picklist_id	是	String	Picklist实例的ID，可通过调用 查询选项列表值集 接口查看（响应消息中的“id”）。

请求参数

表 5-82 请求参数

参数	是否必选	参数类型	描述
flag	否	String	查询选项列表值集或引用该选项列表值集的对象字段列表。 <ul style="list-style-type: none"> flag=R: 表示查询引用该选项列表值集的对象字段列表。 不带flag参数, 表示只查询该选项列表值集。

响应参数

表 5-83 响应参数

参数	参数类型	描述
resCode	String	返回码, 如果返回“0”代表请求成功, 其他错误码说明请参考 错误码 。
resMsg	String	返回消息, 如果成功状态, 通常会返回“Success”, 其他情况会返回具体的错误信息。
result	Object	成功会返回选项列表值集或引用该选项列表值集的对象字段列表。

请求示例

查询Picklist对象实例。

GET <https://AstroZero域名/u-route/baas/metadata/v1.0/Picklist/000y000000aM92DCu6hU>

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": [
    {
      "id": "testListId",
      "name": "testName",
      "isSorted": false,
      "isRestricted": false,
      "controlField": "",
      "masterLabel": "testLabel",
      "description": "",
      "isGlobal": true,
      "isBaseline": false,
      "picklistValues": [
        {
          "id": "testValueId",
          "value": "A",

```



```

        "isDefault": false,
        "isActive": true,
        "label": "A",
        "sequenceNo": 0,
        "color": "",
        "controlFieldValue": "",
        "description": "",
        "isBaseline": false,
        "createdBy": "testUserId",
        "createdBy.name": "test0xx",
        "createdDate": "2019-08-05 20:38:24",
        "lastModifiedBy": "testUserId",
        "lastModifiedBy.name": "test0xx",
        "lastModifiedDate": "2019-08-05 20:38:24",
        "owner": "testUserId",
        "owner.name": "test0xx",
        "custom": true
    }
  ],
  "createdBy": "testUserId",
  "createdBy.name": "test0xx",
  "createdDate": "2019-08-05 20:38:24",
  "lastModifiedBy": "testUserId",
  "lastModifiedBy.name": "test0xx",
  "lastModifiedDate": "2019-08-05 20:38:24",
  "owner": "testUserId",
  "owner.name": "test0xx",
  "custom": true
}
]
}

```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.6.5 根据 ID 修改选项列表值集的基本信息

功能介绍

根据id修改Picklist对象。

URI

PUT AstroZero域名/u-route/baas/metadata/v1.0/Picklist/{Picklist_id}

表 5-84 路径参数

参数	是否必选	参数类型	描述
Picklist_id	是	String	Picklist实例的ID，可通过调用 查询选项列表值集 接口查看（响应消息中的“id”）。

请求参数

表 5-85 请求 Body 参数

参数	是否必选	参数类型	描述
masterLabel	是	String	选项列表值集标签。
description	否	String	选项列表值集描述。

响应参数

表 5-86 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	String	成功后，返回数字1。

请求示例

修改Picklist对象实例。

```
PUT https://AstroZero域名/u-route/baas/metadata/v1.0/Picklist/000y000000aM92DCu6hU
{
  "masterLabel": "string",
  "description": "string"
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": "1"
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.6.6 根据 ID 删除选项列表值集或选项列表值集的值

功能介绍

根据ID删除选项列表值集或选项列表值集的值。

- 若需要删除整个选项列表值集，需要确保该选项列表值集未被对象引用，此时不带value参数。
- 若需要删除某个选项列表值集中的值，需要确保待删除的值不是最后一个已激活的值，此时需要带value参数。

URI

DELETE *AstroZero*域名/u-route/baas/metadata/v1.0/Picklist/{Picklist_id}

表 5-87 路径参数

参数	是否必选	参数类型	描述
Picklist_id	是	String	Picklist实例的ID，可通过调用 查询选项列表值集 接口查看（响应消息中的“id”）。

请求参数

表 5-88 请求参数

参数	是否必选	参数类型	描述
value	否	String	仅在需要删除具体选项时使用，选项列表值集列表中待删除值的ID。

响应参数

表 5-89 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	string	成功后，会返回数字1。

请求示例

删除Picklist实例。

DELETE `https://AstroZero域名/u-route/baas/metadata/v1.0/Picklist/000y000000aM92DCu6hU`

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": 1
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.7 Application 对象的操作

5.2.7.1 列出所有应用

功能介绍

查询当前账号下所有应用。

URI

GET `AstroZero域名/u-route/baas/metadata/v1.0/Application`

请求消息

无

响应参数

表 5-90 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Array	成功返回会返回该账号下所有应用，以一个列表展示，每个值是一个应用。

表 5-91 其中 result 内每个对象的主要属性

参数	参数类型	描述
id	String	应用ID。
label	String	应用的展示名称。
name	String	应用在系统内的唯一标识。
description	String	应用的描述信息。
type	String	应用类型。
logo	String	应用图标，可填图片地址（其前缀是受限的）。
properties	String	应用属性。

请求示例

查询当前账号下所有的应用。

GET <https://AstroZero域名/u-route/baas/metadata/v1.0/Application>

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": [{
    "active": true,
    "belongTo": null,
    "category": null,
    "coreVersion": null,
    "createdBy": "10gd000000bZHUAq7Pea",
    "createdBy__objectType": "User",
    "createdBy.name": "test",
    "createdDate": "2019-02-10 22:14:41",
    "custom": true,
    "defaultTabId": "001g0000000000000000",
    "defaultTabId__objectType": "Tab",
    "defaultTabId.name": "Home",
    "description": "应用1",
    "id": "000f000000Nhliwso4p6",
    "installedPackage": "003L000000mR6pf2uFqi",
    "installedPackage__objectType": "PackageInstall",
    "installedPackage.name": "FacilityManagement",
    "introductionPageId": null,
    "isIncremental": false,
    "label": "Facility Management",
    "lastModifiedBy": "10gd000000bZHUAq7Pea",
    "lastModifiedBy__objectType": "User",
    "lastModifiedBy.name": "test",
    "lastModifiedDate": "2020-01-10 15:21:18",
    "logo": null,
    "name": "SmartCampus_FacilityManagement",
    "owner": "10gd000000bZHUAq7Pea",
    "owner__objectType": "User",
    "owner.name": "test",
    "properties": "{\"version\":\"0.0.0\",\"packageId\":\"testPackageId\"}",
    "protectMode": 1,
    "runtimeProperties": null,
    "supportBO20": false,
  ]
}
```

```
"theme": null,
"type": "Custom Application",
"type.name": "Custom Application",
"visibility": null,
"visibleScope": null
},
{
  "active": true,
  "belongTo": null,
  "category": "",
  "coreVersion": "1.3.3",
  "createdBy": "10gd000000bZHUAq7Pea",
  "createdBy.__objectType": "User",
  "createdBy.name": "test",
  "createdDate": "2020-09-28 20:30:35",
  "custom": true,
  "defaultTabId": "001g0000000000000000",
  "defaultTabId.__objectType": "Tab",
  "defaultTabId.name": "Home",
  "description": "演示流程场景可配置案例。",
  "id": "000f000000ddHP8z87N2",
  "installedPackage": "003L000000llRwGwwuDg",
  "installedPackage.__objectType": "PackageInstall",
  "installedPackage.name": "$caseDemo_a",
  "introductionPageID": null,
  "isIncremental": false,
  "label": "故障单流程",
  "lastModifiedBy": "10gd000000bZHUAq7Pea",
  "lastModifiedBy.__objectType": "User",
  "lastModifiedBy.name": "test",
  "lastModifiedDate": "2021-08-02 10:20:56",
  "logo": "/img/03.cd9e2a5.jpg",
  "name": "caseDemo",
  "owner": "10gd000000bZHUAq7Pea",
  "owner.__objectType": "User",
  "owner.name": "test",
  "properties": "{\"version\":\"0.0.0\",\"packageId\":\"003G000000ddHP8QV3VA\"}",
  "protectMode": 0,
  "runtimeProperties": "{\"version\":\"180828\",\"packageId\":\"003G000000ddHP8QV3VA\"}",
  "supportBO20": false,
  "theme": null,
  "type": "Custom Application",
  "type.name": "Custom Application",
  "visibility": null,
  "visibleScope": null
}
]
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.2.7.2 根据 ID 查询应用

功能介绍

根据应用ID查询应用详细信息。

URI

GET AstroZero域名/u-route/baas/metadata/v1.0/Application/{app_id}

表 5-92 参数说明

参数	是否必选	参数类型	描述
app_id	是	String	应用的ID值，可通过调用 列出所有应用 接口查看（响应消息中的“id”）。

请求参数

无

响应参数

表 5-93 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	成功返回应用的详细信息，具体返回的信息可参考 表5-94 。

表 5-94 result 内每个对象的主要属性

参数	参数类型	描述
id	String	应用ID。
label	String	应用的展示名称。
name	String	应用在系统内的唯一标识。
description	String	应用的描述信息。
type	String	应用类型。
logo	String	应用图标，可填图片地址（其前缀是受限的）。
properties	String	应用属性。

请求示例

按照应用ID “000f000000g3Yhaz5ipE” 查询该应用详细信息。

```
GET https://AstroZero域名/u-route/baas/metadata/v1.0/Application/000f000000g3Yhaz5ipE
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "active": true,
    "belongTo": null,
    "category": "",
    "coreVersion": "1.3.3",
    "createdBy": "10gd000000bZHUAq7Pea",
    "createdBy.__objectType": "User",
    "createdBy.name": "test",
    "createdDate": "2020-12-28 09:33:29",
    "custom": true,
    "defaultTabId": "001g0000000000000000",
    "defaultTabId.__objectType": "Tab",
    "defaultTabId.name": "Home",
    "description": "",
    "id": "000f000000g3Yhaz5ipE",
    "installedPackage": "003L000000mrA59c0v6e",
    "installedPackage.__objectType": "PackageInstall",
    "installedPackage.name": "$AstroZero_phoneSales_a",
    "introductionPageId": null,
    "isIncremental": false,
    "label": "手机销量统计",
    "lastModifiedBy": "10gd000000bZHUAq7Pea",
    "lastModifiedBy.__objectType": "User",
    "lastModifiedBy.name": "test",
    "lastModifiedDate": "2021-01-15 15:22:55",
    "logo": "",
    "name": "AstroZero_phoneSales",
    "owner": "10gd000000bZHUAq7Pea",
    "owner.__objectType": "User",
    "owner.name": "test",
    "properties": "{\"version\":\"180828\",\"skava\":{\"id\":\"001S000000g3YhZ1bP9c\"},\"websiteld\": \"517297\", \"name\": \"AstroZero_phoneSales_000000000Z16qkNc1Uv\"}, \"packageId\": \"003G000000g3Yhapix6\"}",
    "protectMode": 0,
    "runtimeProperties": null,
    "supportBO20": false,
    "theme": null,
    "type": "Custom Application",
    "type.name": "Custom Application",
    "visibility": null,
    "visibleScope": null
  }
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.3 服务编排

5.3.1 编排 Flow

功能介绍

编排某个服务编排。除简单运行外还可以对服务编排进行分步执行，支持启动服务编排、运行服务编排、执行下一步、执行上一步、正常完成服务编排、终止（删除）服务编排运行实例等操作，传入参数包括需要执行的操作名称和需要执行的服务编排名称或服务编排ID。

URI

POST *AstroZero*域名/u-route/baas/v1.0/flow/{action}/{name_or_id}

表 5-95 参数说明

参数	是否必选	描述
action	是	需要对服务编排执行的操作，各个操作取值及其含义如下： <ul style="list-style-type: none">• start：启动服务编排，系统标准启动操作。• run：开始执行服务编排，自定义启动操作。• next：执行服务编排的下一步。• back：执行服务编排的上一步。• finish：结束运行的服务编排，即正常完成服务编排。• terminate：终止运行的服务编排实例和相关资源。• del：删除服务编排实例和相关资源。
name_or_id	是	当action取值为start或run时，需为服务编排名称。其他操作需为服务编排的运行实例ID，可通过调用 查看服务编排运行实例 接口查看（响应消息中的“id”）。

请求参数

请求体中的参数均为用户自定义的，为该服务编排中定义的输入参数，具体参数视服务编排而定。

响应参数

表 5-96 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	在成功请求时，会有内容返回。内部包含interviewId和outputs两个字段，result字段的说明请参见 表5-97 。

表 5-97 result 字段说明

参数	参数类型	描述
interviewId	String	服务编排的运行实例ID。流程启动成功后，可以从返回结果中获取到运行实例ID。
outputs	Any	结果，服务编排的执行输出，由服务编排输出决定，可为任意取值。

请求示例

开始运行服务编排，例如调用start命令执行了服务编排“testFlowXyz”，传入参数为自定义的参数。

```
POST https://AstroZero域名/u-route/baas/v1.0/flow/start/testFlowXyz
{
  "paramA": "valueA",
  "paramB": {
    "subParamBA": [
      "A",
      "B"
    ],
    "subParamBB": "BB"
  }
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": "1234567890",
  "output": {
    "outA": "123",
    "outB": "456"
  }
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.3.2 查看服务编排运行实例

功能介绍

查询服务编排运行实例的信息，即根据输入的参数筛选出符合条件的服务编排实例的具体信息并返回。

URI

GET *AstroZero*域名/u-route/baas/v1.0/flow/{action}

表 5-98 参数说明

参数	是否必选	描述
action	是	需要查询的服务编排状态或者根据id查询服务编排信息，其中各个状态取值及其含义如下： <ul style="list-style-type: none">activeFlow：正在执行的服务编排。suspendedFlow：延时的服务编排。terminatedFlow：被终止的服务编排。completedFlow：正常完成的服务编排。instanceId：根据id查询服务编排信息，请求时该参数需替换为具体的服务编排运行实例ID。
owner	否	所属账号ID。
creator	否	创建者ID。
fields	否	字符串数组，为需要查询的字段，不传则默认。
orderField	否	用于排序的字段，需要在fields中。
orderBy	否	排序方式，0代表Asc升序，1代表Desc降序。
limit	否	数据条数限制。
offset	否	数据起始位置。

请求参数

无

响应参数

表 5-99 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	在成功请求时会有内容返回。内部包含请求中传入的fields中的字段或默认字段。

请求示例

查询实例ID为“002N000000jrOCerKZBQ”的服务编排实例。
GET <https://AstroZero域名/u-route/baas/v1.0/flow/002N000000jrOCerKZBQ>

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "createdBy": "10gd000000QwWjXspAOW",
    "createdBy.__objectType": "User",
    "createdBy.name": "bingo",
    "createdDate": "2021-05-19 11:41:30",
    "curElement": "isMasterAccountByIdentityId",
    "description": "",
    "endTime": "",
    "id": "002N000000jrOCerKZBQ",
    "label": "isMasterAccount 2021-05-19 11:41:30",
    "lastHandler": null,
    "lastModifiedBy": "10gd000000QwWjXspAOW",
    "lastModifiedBy.__objectType": "User",
    "lastModifiedBy.name": "bingo",
    "lastModifiedDate": "2021-05-19 11:41:30",
    "name": "CP_DS__isMasterAccountForOperator",
    "owner": "10gd000000QwWjXspAOW",
    "owner.__objectType": "User",
    "owner.name": "bingo",
    "rootID": "",
    "runMode": 2,
    "startTime": "2021-05-19 11:41:30",
    "state": "Broken",
    "suspended": false,
    "type": "Flow",
    "type.name": "Flow",
    "version": "1.0.1"
  }
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.4 脚本

5.4.1 运行脚本

功能介绍

运行脚本，该接口是同步接口（同步接口是指调用后，会等待函数执行结束后，才返回结果）。满足输入参数条件后执行该脚本，不会在控制台打印出日志，但会输出运行结果。

在脚本保存成功和输入参数正确的情况下，可以运行该脚本，但该接口目前已很少使用（可以用custom API运行脚本）。

URI

POST *AstroZero*域名/u-route/baas/script/v1.0/run/{scriptName}{?version=1.0.1}

表 5-100 参数说明

参数	是否必选	描述
scriptName	是	脚本的名称。
version	否	脚本的版本号，表示运行具体版本号对应的脚本。可选参数，不配置表示运行最新版本的脚本。

请求参数

请求体中的参数均为用户自定义的，为该脚本中定义的输入参数，具体参数视脚本而定。

响应参数

表 5-101 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	在成功请求时，会有内容返回。

请求示例

运行脚本，假设脚本“命名空间__editEquipment”有输入参数equipment、name、命名空间__equipmentSN__CST。

```
POST https://AstroZero域名/u-route/baas/script/v1.0/run/命名空间__editEquipment?version=1.0.1
{
  "equipment": {
    "name": "某某园区A栋1单元16号",
    "命名空间__equipmentSN__CST": "3217890020"
  }
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "value": {
      "equipmentId": "c9uK000000iYJt41oj8S"
    },
    "log": ""
  }
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.4.2 调试脚本

功能介绍

调试脚本，该接口是同步接口（同步接口是指调用后，会等待函数执行结束后，才返回结果）。满足输入参数条件后执行该脚本，会在控制台打印出debug日志和运行结果。

在脚本保存成功和输入参数正确的情况下，才可以成功执行该脚本。

URI

POST AstroZero域名/u-route/baas/script/v1.0/debug/{scriptName}{?version=1.0.1}

表 5-102 路径参数

参数	是否必选	描述
scriptName	是	脚本名称，请确保该脚本已启用。
version	否	脚本的版本号信息，表示运行具体版本号对应的脚本。 可选参数，不配置表示运行最新版本的脚本。

请求参数

请求参数请填写具体该脚本的入参。

响应参数

表 5-103 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	result object	在成功请求时，会有内容返回。

表 5-104 响应参数 result

参数	参数类型	描述
value1	任意类型	执行脚本的返回值。
log	String	debug日志。

请求示例

调试脚本，假设脚本“命名空间__editEquipment”有输入参数equipment、name、命名空间__equipmentSN__CST。

```
POST https://AstroZero域名/u-route/baas/script/v1.0/debug/命名空间__editEquipment?version=1.0.1
{
  "equipment": {
    "name": "某某园区A栋1单元14号",
    "命名空间__equipmentSN__CST": "32178900014"
  }
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "value": {
      "equipmentId": "c9uK000000iYJXUUnQcC"
    },
    "log": "0331 15:20:11.125[debug]vm[694]>>> Build #AstroZero Core 1.3.3.5 on amd64\nBuilt on 2021-03-03 14:02:59 \nCommit #73a5610\n0331 15:20:11.125[debug]vm[694]>>> node: 4\n0331 15:20:11.126[debug]vm[694]>>> script: 命名空间__editEquipment 1.0.1 EditEquipment.editEquipment\n0331 15:20:11.126[debug]vm[694]>>> locale: zh_CN\n0331 15:20:11.126[debug]vm[694]>>> timezone: (GMT+08:00) China Standard Time (Asia/Shanghai)\n"
```

```
}  
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.4.3 获取脚本签名信息

功能介绍

获取脚本签名信息，该接口是同步接口（同步接口是指调用后，会等待函数执行结束后，才返回结果）。获取脚本的名称、输入参数、输出参数、声明的类名、方法名等相关信息。

只有用户保存或另存脚本时会调用该接口，当脚本中定义的输入参数或输出参数的类型错误时，该接口会执行报错。

URI

GET *AstroZero*域名/u-route/baas/script/v1.0/signature/{scriptName}?
version=1.0.1}

表 5-105 路径参数

参数	是否必选	描述
scriptName	是	脚本的名称。
version	否	取值为脚本的版本号信息，表示获取具体版本号对应的脚本签名信息。 可选参数，不填写表示最新版本的脚本。

请求参数

无

响应参数

表 5-106 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。

参数	参数类型	描述
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	result object	在成功请求时，会有内容返回。

表 5-107 result 参数说明

参数	参数类型	描述
description	String	脚本的描述信息。
script	String	脚本名称。
methodName	String	方法名称，在脚本使用@action.method注解后才会有方法名。
methodObject	String	方法所属的类。
input	String	输入类型，一般为Input。
output	String	输出类型，一般为Output。
label	String	脚本标签。
inputs	Array of inputs objects	输入类型中的变量信息描述。
outputs	Array of inputs objects	输出类型中的变量信息描述。
types	Object	脚本中使用@action.param注解的类名。

表 5-108 inputs 和 outputs 参数说明

参数	参数类型	描述
description	String	脚本的描述信息。
name	String	脚本名称。
label	String	方法名称，在脚本使用@action.method注解后才会有方法名。
message	String	运行脚本参数不合法输入时，返回的错误信息。
isCollection	String	变量类型是否是集合。
pattern	String	输入变量值需要满足的正则表达式。
valueType	String	变量的类型。

参数	参数类型	描述
required	String	变量是否必填。

请求示例

获取脚本“命名空间__editEquipment”签名信息。

GET https://AstroZero域名/u-route/baas/script/v1.0/signature/命名空间__editEquipment

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "description": "",
    "input": "ActionInput",
    "inputs": [
      {
        "description": "",
        "isCollection": false,
        "label": "equipment",
        "lengthInCharacter": false,...
      }
    ],
    "label": "",
    "methodName": "editEquipment",
    "methodObject": "EditEquipment",
    "output": "ActionOutput",
    "outputs": [
      {
        "description": "",
        "isCollection": false,
        "label": "",
        "lengthInCharacter": false,...
      }
    ],
    "script": "命名空间__editEquipment",
    "types": {
      "ActionInput": [
        {
          "description": "",
          "isCollection": false,
          "label": "equipment",
          "lengthInCharacter": false,...
        }
      ],
      "ActionOutput": [
        {
          "description": "",
          "isCollection": false,
          "label": "",
          "lengthInCharacter": false,...
        }
      ]
    }
  }
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.4.4 获取脚本内置库信息

功能介绍

获取脚本内置库的声明，该接口为同步接口（同步接口是指调用后，会等待函数执行结束后，才返回结果）。获取AstroZero平台可以调用的typeScript库，其中包括脚本函数以及函数相关信息，并且可以获取ES5语法的标准库。

AstroZero平台可以调用的脚本函数必须已经映射到go函数中，才可以成为平台可以调用的typeScript库一部分。

URI

GET *AstroZero*域名/u-route/baas/script/v1.0/declaration

请求参数

无

响应参数

表 5-109 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
Result	result object	在成功请求时会有内容返回。

表 5-110 result 参数说明

参数	参数类型	描述
bingo.d.ts	String	可以调用的脚本库。
lib.es5.d.ts	String	ES5语法的标准库。

请求示例

获取内置库的声明。

```
GET https://AstroZero域名/u-route/baas/script/v1.0/declaration
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "bingo.d.ts": "string",
    "lib.es5.d.ts": "string"
  }
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.4.5 获取 Flow 公式信息

功能介绍

获取Flow的公式，该接口是同步接口（同步接口是指调用后，会等待函数执行结束后，才返回结果）。这些公式可以使用Flow对变量或表达式进行转换，然后赋值给其他变量。

URI

GET *AstroZero*域名/u-route/baas/formula/v1.0/functions

请求参数

无

响应参数

表 5-111 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	result object	在成功请求时会有内容返回。

表 5-112 result 参数说明

参数	参数类型	描述
name	String	服务编排的公式名称。
category	String	服务编排公式的类别。
format	String	服务编排公式的调用格式。
description	String	服务编排公式的描述信息。

请求示例

获取服务编排，可以调用的公式。

GET `https://AstroZero域名/u-route/baas/formula/v1.0/functions`

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "name": "string",
    "category": "string",
    "format": "string",
    "description": "string"
  }
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.5 BPM

5.5.1 启动 BPM 实例

功能介绍

创建并启动一个BPM实例。

📖 说明

启动BMP实例前，需要先通过界面编排定义好BPM的元数据信息。

URI

POST `AstroZero域名/u-route/baas/bp/v2.0/runtime/instances`

请求参数

表 5-113 请求参数

参数	是否必选	参数类型	描述
name	是	String	BPM的名称，在界面处创建BPM时配置的名称。
version	否	String	BPM版本号，为可选参数，因为一个BPM可以存在多个版本。如果不带版本号，表示运行当前启用的版本。
variables	否	Object	BPM元数据中定义的变量，包括自定义变量和系统变量，key-value形式，key为变量名字，value为变量的值。如果需要在启动BPM前设置BPM的变量，可以填写该参数。如需启动时设置变量，可以在variables中设置系统变量\$Flow.BusinessKey。如需设置BPM title，可以设置系统变量\$Flow.Title。

响应参数

表 5-114 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	在成功请求时会有内容，返回启动BPM后，BPM实例的运行信息。

请求示例

创建并启动一个BPM实例，例如创建并启动了一个BPM，BPM名称为“createCloudOK_NEW”，版本号为“1.0.1”，带有4个启动参数。

POST <https://AstroZero域名/u-route/baas/bp/v2.0/runtime/instances>

```
{
  "name": "createCloudOK_NEW",
  "version": "1.0.1",
  "variables": {
    "name": "sun",
    "memorySize": "15",
    "storageSize": "15",
    "expiryDate": "2018-12-12"
  }
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "Success",
  "result": {
    "name": "bpName1",
    "version": "1.0.1",
    "id": "002N000000Sy2NphoRbU",
    "processDefID": "001L000000S8Yi3xG9Gi",
    "interviewLabel": "BP Label 1",
    "nextElem": "subProcess",
    "isTest": true,
    "state": "Dispatched",
    "bpStatus": "New",
    "urgency": 0,
    "taskOutcome": "",
    "businessKey": "002N000000Sy2NphoRbU",
    "startTime": "2019-08-27 10:53:04",
    "endTime": "",
    "Ext1": "",
    "Ext2": "",
    "Ext3": "",
    "Ext4": "",
    "Ext5": "",
    "Ext6": "",
    "Ext7": 0,
    "Ext8": 0,
    "suspended": false,
    "isClosed": false
  }
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.5.2 查询 BPM 实例

功能介绍

获取当前账号下BPM实例信息。

URI

GET *AstroZero*域名/u-route/baas/bp/v2.0/runtime/instances?states=X

表 5-115 路径参数

参数	是否必选	描述
states	是	BPM实例状态，取值如下： <ul style="list-style-type: none"> • New • Running • Waiting for user interaction • Waiting for page event • Completed OK • Paused • Broken • Suspended • User Terminated • Terminated • Dispatched • CompletedOkWait

请求参数

表 5-116 请求参数

参数	是否必选	参数类型	描述
fields	否	String	指明要查询的字段，多个字段名字之间以逗号分隔，可选参数。如果不指明字段，则返回缺省的字段。
orderField	否	String	用来排序的字段名字，可选参数。
orderType	否	String	排序类型，取值如下： <ul style="list-style-type: none"> • asc：升序 • desc：降序

响应参数

表 5-117 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。

参数	参数类型	描述
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	在成功请求时会有内容，返回查询到的BPM实例信息列表。

请求示例

查询当前BPM实例列表，例如示例中查询到了一个匹配的记录项。

```
GET https://AstroZero域名/u-route/baas/bp/v2.0/runtime/instances?states=Completed  
OK&orderField=title&orderType=asc
```

响应示例

```
{  
  "resCode": "0",  
  "resMsg": "Success",  
  "result": [  
    {  
      "bpStage": "Stage 1",  
      "bpStatus": "begin",  
      "bpStep": "Transfer 1",  
      "businessKey": "BK001",  
      "createdBy": "10gd0000003T2E10UTfk",  
      "createdBy._objectType": "User",  
      "createdBy.name": "ecommerce",  
      "createdDate": "2018-08-13 16:21:17",  
      "curElement": "",  
      "description": "Transfer 1",  
      "endTime": "2018-08-17 14:40:09",  
      "id": "002N000000lpNGFfNvm4",  
      "label": "TestFSM2 2018-08-13 16:21:17",  
      "lastHandler": null,  
      "lastModifiedBy": "10gd0000003T2E10UTfk",  
      "lastModifiedBy._objectType": "User",  
      "lastModifiedBy.name": "ecommerce",  
      "lastModifiedDate": "2018-08-17 14:40:09",  
      "name": "TestFSM2",  
      "owner": "10gd0000003T2E10UTfk",  
      "owner._objectType": "User",  
      "owner.name": "ecommerce",  
      "startTime": "2018-08-13 16:21:17",  
      "state": "Completed OK",  
      "suspended": false,  
      "title": "Hello, this is a title",  
      "type": "BP",  
      "urgency": 0,  
      "version": "0.0.1"  
    }  
  ]  
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.5.3 挂起/恢复/终止 BPM 实例

功能介绍

挂起/恢复/终止一个BPM实例。BPM实例被挂起后，不会在用户的待办任务列表中出现，流程SLA定时器停止，也不会进行任何跟催。恢复后，可继续进行操作。终止操作则强行结束一个BPM实例，可在历史任务中查看。

URI

PUT *AstroZero*域名/u-route/baas/bp/v2.0/runtime/instances/{instance_id}

表 5-118 路径参数

参数	是否必选	描述
instance_id	是	对应一个具体BPM实例的标识，可通过调用 查询BPM实例 接口查看（响应消息中的“id”）。

请求参数

表 5-119 请求参数

参数	是否必选	参数类型	描述
action	是	String	代表对BPM实例操作的动作名。 <ul style="list-style-type: none"> • suspend：挂起。 • resume：恢复。 • terminate：终止。 • restart：重启。 • cancel：撤销。 • rollBackPreviousTask：打回任务，即管理员或当前任务处理人将流程回退到上一步任务。 • revokePreviousTask：撤回任务，即任务提交人撤回已提交审批的任务，任务重新回到待审批。

响应参数

表 5-120 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	在成功请求时会有内容，返回启动BPM之后，BPM实例的运行信息。

请求示例

对一个已启动运行的BPM实例进行暂停操作，被操作的BPM实例ID是“002N000000Q3QW4IMAOO”。

PUT <https://AstroZero域名/u-route/baas/bp/v2.0/runtime/instances/002N000000Q3QW4IMAOO>

```
{
  "action": "suspend"
}
```

响应示例

可以看到返回的报文中，BPM实例的suspended字段被置为了true。

```
{
  "resCode": "0",
  "resMsg": "Success",
  "result": {
    "name": "SimpleBP",
    "version": "1.0.1",
    "id": "002N000000Q3QW4IMAOO",
    "processDefID": "001L000000Peuw1Q318a",
    "interviewLabel": "SimpleBP 2019-05-10 15:14:09",
    "nextElem": "",
    "isTest": true,
    "state": "New",
    "bpStatus": "New",
    "urgency": 0,
    "taskOutcome": "",
    "businessKey": "002N000000Q3QW4IMAOO",
    "startTime": "2019-05-10 15:14:09",
    "endTime": "",
    "Ext1": "",
    "Ext2": "",
    "Ext3": "",
    "Ext4": "",
    "Ext5": "",
    "Ext6": "",
    "Ext7": 0,
    "Ext8": 0,
    "suspended": true,
    "isClosed": false
  }
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.5.4 删除 BPM 实例

功能介绍

删除一个BPM实例。彻底删除一个BPM实例，仅当管理员清除过期数据时可以删除，其他情况不建议删除BPM实例数据。

URI

DELETE *AstroZero*域名/u-route/baas/bp/v2.0/runtime/instances/{instance_id}

表 5-121 路径参数

参数	是否必选	描述
instance_id	是	对应一个具体BPM实例的标识，可通过调用 查询BPM实例 接口查看（响应消息中的“id”）。

请求参数

无

响应参数

表 5-122 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。

请求示例

对一个已存在的BPM实例进行删除操作，被操作的BPM实例ID是“002N000000Q3QW4IMA00”。

DELETE *https://AstroZero*域名/u-route/baas/bp/v2.0/runtime/instances/002N000000Q3QW4IMA00

响应示例

```
{
  "resCode": "0",
  "resMsg": "success"
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.5.5 配置或修改 BPM 实例中变量的值

功能介绍

设置或修改BPM实例中变量的值，一次可以修改一个、多个或者所有变量的值。不存在的变量或者只读变量，会被忽略。入参以key-value集合形式表示，key表示变量名字，value表示变量的值。

URI

PUT *AstroZero*域名/u-route/baas/bp/v2.0/runtime/instances/{*instance_id*}/variables

表 5-123 路径参数

参数	是否必选	描述
instance_id	是	对应一个具体BPM实例的标识，可通过调用 查询BPM实例 接口查看（响应消息中的“id”）。

请求参数

设置或修改变量的值。入参以key-value集合形式表示，key表示变量名字，value表示变量的值。

无固定的参数，所有参数都有开发者自行指定，结构体报文可见下文示例。

响应参数

表 5-124 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。

请求示例

对一个BPM实例修改或设置，设置其中两个开发者自定义的参数值，将变量cpuNumber的值改为1，custVar1的值改为value1。被操作的BPM实例ID是“002N000000Q3QW4IMAOO”。

PUT *https://AstroZero域名/u-route/baas/bp/v2.0/runtime/instances/002N000000Q3QW4IMAOO/variables*

```
{
  "cpuNumber":1,
  "custVar1":"value1"
}
```

响应示例

```
{
  "resCode":"0",
  "resMsg":"Success",
  "result":null
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.6 导入导出

5.6.1 导出数据

功能介绍

导出数据表中的具体数据。平台中创建的对象中，其内部的数据是支持导出的，该接口支持将对象中的数据按照设定的周期频率、筛选条件进行导出。根据设定的执行时间，会自动生成定时任务，定期或单次执行导出任务，数据导出完成后可调用文件下载接口下载文件，导出完成后您可根据得到的定时任务id查看该任务的执行情况。

URI

POST *AstroZero域名/u-route/baas/bulk/v1.0/export/data?flag=template&file_type=csv*

表 5-125 路径参数

参数	是否必选	描述
file_type	是	导出文件的格式，支持csv格式和txt格式。 <ul style="list-style-type: none">csv：数据以csv格式导出。txt：数据以txt格式导出。

请求参数

表 5-126 请求参数

参数	是否必选	参数类型	描述
cond	否	cond objects	导出筛选条件，为一个结构体，包含conditions和conjunction两个字段。
encode-type	是	String	数据导出文件的编码格式，支持utf-8和gbk两种类型。
firstScheduleDate	否	String	首次导出时间，格式为“yyyy-MM-dd HH:mm:ss” <ul style="list-style-type: none"> • scheduleMode=2时，必选。 • scheduleMode=1时，与isImmediate互斥。
objects	否	Array of Strings	需要导出的对象，为一字符串数组，数组中每个字符串为一个对象名，可同时导出多个对象数据，也可通过templateName参数指定模板选择导出的数据。
scheduleMode	是	Number	导出模式。 <ul style="list-style-type: none"> • 1：仅执行一次。 • 2：按设定的频率执行多次。
scheduleInterval	否	Number	“scheduleMode=2”时必选，导出任务执行间隔。
intervalUnit	否	Number	“scheduleMode=2”时必选，导出任务执行间隔对应的单位。 <ul style="list-style-type: none"> • Hour：小时 • Day：天 • Month：月
isImmediate	否	Boolean	是否立即导出，“scheduleMode=1”时可选，“scheduleMode=1”时与firstScheduleDate互斥。
templateName	否	String	模板名称，数据导出使用的模板名称，也可通过objects参数直接指定希望导出的对象。

表 5-127 cond 参数说明

参数	参数类型	描述
conditions	Array of condition objects	条件数组，数组中每项作为一个条件。

参数	参数类型	描述
conjunction	String	conditions数组中各个条件的组合方式。 <ul style="list-style-type: none"> • AND: 与 • OR: 或

表 5-128 条件参数 condition 说明

参数	是否必选	描述
field	是	筛选条件字段。 <ul style="list-style-type: none"> • createDate: 创建时间 • createBy: 创建者id • lastModifiedDate: 最后编辑时间 • lastModifiedBy: 最后编辑者id
operator	是	筛选运算操作。 <ul style="list-style-type: none"> • gt: 大于 • ge: 大于等于 • eq: 等于 • ne: 不等于 • lt: 小于 • le: 小于等于 • timeFunc: 时间函数 <p>需要指出的是，只有时间类型的筛选字段支持所有操作，其他类型的字段仅支持eq/ne（等于或不等于）。</p>
value	是	进行比较的具体值，当operator为timeFunc时，支持以下函数： <ul style="list-style-type: none"> • today: 今天 • yesterday: 昨天 • last week: 上周 • last month: 上月 • last year: 去年 • this week: 本周 • this month: 本月 • this year: 今年 <p>若比较的字段为时间类型，则此处使用的时间字符串应为“yyyy-MM-dd HH:mm:ss”格式。</p>

响应参数

表 5-129 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	String	创建成功的记录ID，成功时返回。

请求示例

- 示例1

如下请求创建了一个仅执行一次的数据导出任务，执行时间为2020-05-31 00:00:00，导出的文件格式为csv，编码格式为utf-8，需要导出的对象通过名为“myTemplate”的模板指定，未指定任何导出筛选条件。

```
POST https://AstroZero域名/u-route/baas/bulk/v1.0/export/data?flag=template&file_type=csv
{
  "scheduleMode": 1,
  "firstScheduleDate": "2020-05-31 00:00:00",
  "encodeType": "utf-8",
  "templateName": "myTemplate"
}
```

- 示例2

如下请求创建了周期性执行的数据导出任务，首次执行时间为2020-06-01 00:00:00，往后每隔一个月执行一次，导出的文件格式为txt，编码格式为GBK，需要导出的对象以数组的形式列在objects中，同时设定筛选条件，仅导出创建者id（createBy）为“1234567890xyz”的数据。

```
POST https://AstroZero域名/u-route/baas/bulk/v1.0/export/data?flag=template&file_type=txt
{
  "scheduleMode": 2,
  "firstScheduleDate": "2020-06-01 00:00:00",
  "scheduleInterval": 1,
  "intervalUnit": "Month",
  "templateName": "template01",
  "encodeType": "gbk",
  "objects": [
    "Flow",
    "Page"
  ],
  "cond": {
    "conjunction": "AND",
    "conditions": [
      {
        "field": "createdBy",
        "operator": "eq",
        "value": "1234567890xyz"
      }
    ]
  }
}
```

响应示例

- 响应示例1

```
{  "resCode": "0",  "resMsg": "成功",  "result": "abcdefghijklmnopqrst"}
```

- 响应示例2

```
{  "resCode": "0",  "resMsg": "成功",  "result": "1234567890abcdefg"}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.6.2 导入数据

功能介绍

将本地包中的数据，批量导入到AstroZero相应环境中。导入的数据源为zip包，包中的数据文件应为txt或csv格式，每个文件第一行为数据所属对象信息，第二行为字段名，第三行开始每一行为一条数据记录。导入过程可能耗时较长，本接口不会立即返回结果。导入完成后，可根据得到的定时任务id，查看该任务的执行情况。

URI

POST *AstroZero*域名/u-route/baas/bulk/v1.0/import/data?
filename={filename}&flag={hasid}&comment=X

表 5-130 路径参数

参数	是否必选	描述
filename	是	上传的文件名称，此处文件名对应的文件数据上传时应转为二进制流的形式传输。
flag	是	当前支持的flag。 <ul style="list-style-type: none">hasid：导入的源数据中，已存在ID值。noid：导入的源数据中，没有ID值。
comment	是	是否对zip文件进行完整性校验。 <ul style="list-style-type: none">true：校验文件的完整性。false：不校验文件的完整性。

请求参数

无

响应参数

本接口无响应参数，HTTP状态码为200时说明数据已成功提交，请据此判断请求结果。

请求示例

该请求实现将本地“abcdefg.zip”中的数据批量导入到AstroZero相应环境（开发环境或运行环境，根据接口中域名决定）中，且告知服务器导入的数据中不包含ID值。
POST `https://AstroZero域名/u-route/baas/bulk/v1.0/import/data?filename=abcdefg.zip&flag=noid&comment=true`

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": "002S000000sncZ4nOeS8"
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.6.3 导出元数据

功能介绍

导出数据表的元数据，即将数据表的各个字段信息导出，但不包含具体的数据记录。导出完成后，可根据得到的定时任务id，查看该任务的执行情况。

URI

POST `AstroZero域名/u-route/baas/bulk/v1.0/export/schema?flag={template}`

表 5-131 路径参数

参数	是否必选	描述
flag	否	当前支持的flag为template。当“flag=template”时，需要通过templateName同时指定需要使用的模板名称。

请求参数

表 5-132 请求参数

参数	是否必选	参数类型	描述
encodeType	是	String	元数据导出文件的编码格式。 <ul style="list-style-type: none">• utf-8: utf-8编码• gbk: GBK编码
firstScheduleDate	否	String	首次执行时间，格式为“yyyy-MM-dd HH:mm:ss”。 <ul style="list-style-type: none">• scheduleMode=2时，必选。• scheduleMode=1时，与isImmediate互斥。
objects	否	Array of Strings	一个字符串数组，每一项为一个希望导出元数据的对象名称。
scheduleMode	是	Number	导出模式。 <ul style="list-style-type: none">• 1: 仅执行一次• 2: 按设定频率执行
intervalUnit	否	String	执行间隔单位，“scheduleMode=2”时必选，可选值及其意义如下： <ul style="list-style-type: none">• Hour: 小时• Day: 天• Month: 月
scheduleInterval	否	Number	执行间隔，为一数值，“scheduleMode=2”时必选。
isImmediate	否	Boolean	是否立即导出，“scheduleMode=1”时可选，“scheduleMode=1”时与firstScheduleDate互斥。
templateName	否	String	需要使用的模板名称，与查询参数中的flag配合使用，“flag=template”必选。

响应参数

表 5-133 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。

参数	参数类型	描述
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	String	创建成功的记录ID，成功时返回。

请求示例

该请求导出对象表Lead和PortalUser中的元数据。

POST `https://AstroZero域名/u-route/baas/bulk/v1.0/export/schema`

```
{
  "scheduleMode":1,
  "taskStatus":"Active",
  "firstScheduleDate":"2020-06-02 00:00:00",
  "encodeType":"gbk",
  "objects":["Lead","PortalUser"]
}
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": "1234567890abcdefg"
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.6.4 导入元数据

功能介绍

导入数据表的元数据。若已有数据表的元数据，则可使用该功能快速根据元数据创建对象表。

URI

POST `AstroZero域名/u-route/baas/bulk/v1.0/import/schema?filename=X`

表 5-134 路径参数

参数	是否必选	描述
filename	是	上传的文件名称，需要为zip包，包中可包含多个元数据文件。

请求参数

上传的文件应转换为流传入，无其他参数。

响应参数

表 5-135 响应参数

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	String	创建成功的记录ID，成功时返回。

请求示例

请求导入importtemplate.zip包中的元数据，并得到id为1234567890abcdefg的定时任务。

```
POST https://AstroZero域名/u-route/baas/bulk/v1.0/import/schema?filename=importtemplate.zip
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": "1234567890abcdefg"
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.6.5 使用模板导入数据

功能介绍

按照所使用的模板格式导入数据。使用该功能前，需要创建好可使用的数据导入模板。在模板中定义数据的映射规则，调用接口时需指定模板名称，在该模板中，需要指定对应的操作脚本。因数据导入可能耗时较长，该接口不会立即返回导入结果。导入完成后，可根据得到的定时任务id，查看该任务的执行情况。

URI

POST *AstroZero*域名/u-route/baas/bulk/v1.0/template/xlsx?
template={template}&filename={filename}

表 5-136 路径参数

参数	是否必选	描述
template	是	使用的数据导入模板名称。
filename	是	上传的文件名称，支持xls和xlsx格式的文件。

请求参数

无

响应参数

该接口无响应参数，HTTP状态码为200时即表示文件上传成功，具体的数据导入任务将在后台进行。

请求示例

该请求使用名为myTemplate的数据导入模板，导入了本地文件“myData.xlsx”。

```
POST https://AstroZero域名/ u-route/baas/bulk/v1.0/template/xlsx?  
template=myTemplate&filename=myData.xlsx
```

响应示例

```
{  
  "resCode": "0",  
  "resMsg": "成功",  
  "result": "002S000000sncZ4nOeS8"  
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.7 连接器

5.7.1 上传文件

功能介绍

将本地文件上传到存储中。使用该API上传文件，默认只会上传到连接器下配置的第一个桶中。待上传的文件可以是文本文件、图片、视频等任何类型。

URI

POST *AstroZero*域名/u-route/baas/sys/v1.1/connectors/{connector_type}/{connector_name}/putobject?object=X

表 5-137 路径参数

参数	是否必选	参数类型	描述
connector_type	是	String	连接器的类型。 <ul style="list-style-type: none"> obs: 与OBS对接的连接器。 minio: 与MINIO对接的连接器。 objectstorageproxy: 对象存储代理类型。
connector_name	是	String	连接器实例的名称, 即创建连接器时命名的连接器实例名称。
object	否	String	待上传文件所在桶里的全路径, 包含文件名、文件类型、上传到桶里的路径。不携带路径时, 表示上传到桶的根目录下。 例如: 取值为“a/b/c/1.txt”, 表示将文件“1.txt”上传到桶的“a/b/c”路径下。如果桶中不存在该路径, 会创建该路径。

请求参数

请求中的body类型, 按照“multipart/form-data”格式。

表 5-138 请求参数

参数	是否必选	参数类型	描述
dir	否	String	上传文件在桶上的目录, 如果为空则表示桶上的根目录。
acl	否	String	上传文件的访问控制权限。 <ul style="list-style-type: none"> private: 私有 public-read: 公共读 public-read-write: 公共读写 default: 和桶配置的权限一致 不填的话, 和桶配置的权限一致。
object	是	String	要上传的文件数据, 包含文件名和文件内容。

响应参数

表 5-139 响应参数说明

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	如果请求成功，返回上传文件在桶上的路径。

请求示例

已创建好如[图5-1](#)所示的OBS类型连接器，连接器实例名称为“命名空间__TestOBS”，向该OBS桶的“a/b/c”路径下上传“1.txt”文件。调用API上传文件，只能上传到连接器配置的第一个桶中。

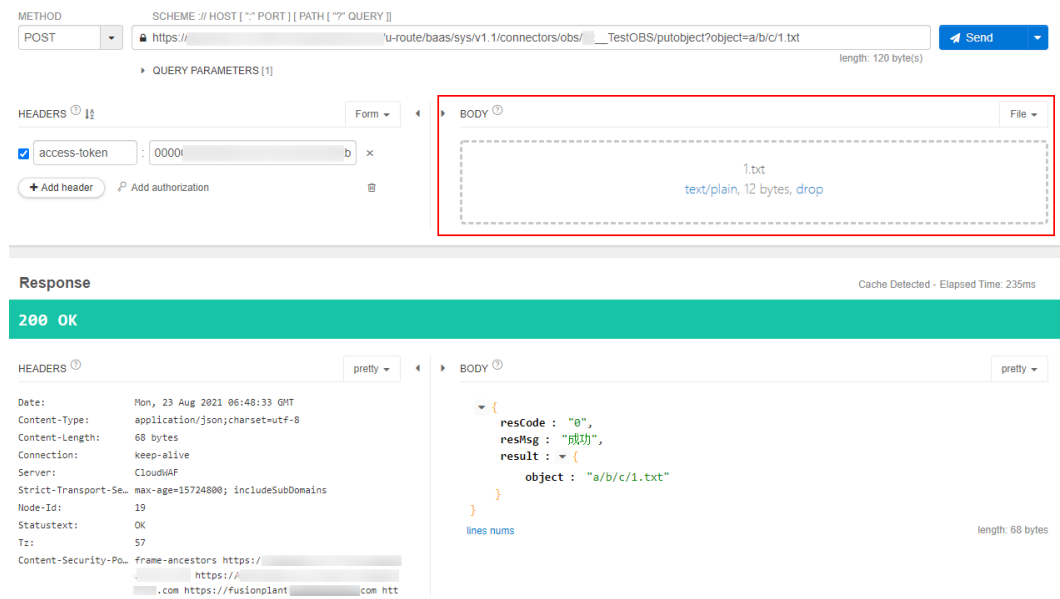
图 5-1 OBS 连接器实例



POST `https://AstroZero域名/u-route/baas/sys/v1.1/connectors/obs/命名空间__TestOBS/putobject?object=a/b/c/1.txt`

请求参数中设置“multipart/form-data”格式的object，即要上传的文件数据。

图 5-2 设置请求参数



上述操作执行完成后，在OBS桶中可查看到“a/b/c”路径下成功上传了“1.txt”文件。

图 5-3 上传结果



响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "object": "a/b/c/1.txt"
  }
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.7.2 下载文件

功能介绍

从存储中下载文件。

URI

GET *AstroZero*域名/u-route/baas/sys/v1.1/connectors/{connector_type}/
{connector_name}/viewobject?object=X&x-image-process=Y

表 5-140 路径参数

参数	是否必选	参数类型	描述
connector_type	是	String	连接器的类型。 <ul style="list-style-type: none">obs: 与OBS对接的连接器minio: 与MINIO对接的连接器objectstorageproxy: 对象存储代理类型
connector_name	是	String	连接器实例的名称, 即创建连接器时命名的连接器实例名称。
object	是	String	文件所在桶里的全路径, 包含文件名。
x-image-process	否	String	在OBS中定义, 表示图片处理服务, 更多介绍参见《 图片处理特性指南 》。 示例: <ul style="list-style-type: none">命令方式: x-image-process=image/commands样式方式: x-image-process=style/stylename

请求参数

表 5-141 请求消息头参数说明

消息头名称	是否必选	参数类型	描述
Range	否	String	查看文件内容的范围, 不填写表示返回全部文件内容。 示例: 取值“bytes=0-”, 表示返回全部字节文件内容。

响应参数

只返回文件内容。

请求示例

已创建好OBS类型连接器，连接器实例名称为“命名空间_TestOBS”，OBS桶中“a/b/c”路径下存在“1.txt”文件，执行如下请求，下载“1.txt”文件内容。

```
GET https://AstroZero域名/u-route/baas/sys/v1.1/connectors/obs/命名空间_TestOBS/viewobject?object=a/b/c/1.txt
```

响应示例

返回如下文件内容。

```
test
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.7.3 删除文件

功能介绍

从存储中删除无用的文件。

URI

```
DELETE AstroZero域名/u-route/baas/sys/v1.1/connectors/{connector_type}/{connector_name}/delobject?object=X
```

表 5-142 路径参数

参数	是否必选	参数类型	描述
connector_type	是	String	连接器的类型。 <ul style="list-style-type: none"> obs: 与OBS对接的连接器。 minio: 与MINIO对接的连接器。 objectstorageproxy: 对象存储代理类型。
connector_name	是	String	连接器实例的名称，即创建连接器时命名的连接器实例名称。
object	是	String	文件所在桶里的全路径，包含文件名。

请求参数

无

响应参数

表 5-143 响应参数说明

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。

请求示例

已创建好OBS类型连接器，连接器实例名称为“命名空间_TestOBS”，OBS桶中“a/b/c”路径下存在“1.txt”文件，执行如下请求，删除该文件。

```
DELETE https://AstroZero域名/u-route/baas/sys/v1.1/connectors/obs/命名空间_TestOBS/delobject?object=a/b/c/1.txt
```

响应示例

```
{  "resCode": "0",  "resMsg": "成功"}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.7.4 删除目录

功能介绍

从存储中删除某目录。如果目录下不为空，删除目录的同时也会删除该目录下文件。

URI

```
DELETE AstroZero域名/u-route/baas/sys/v1.1/connectors/{connector_type}/  
{connector_name}/deldir?dir=path
```

表 5-144 路径参数

参数	是否必选	参数类型	描述
connector_type	是	String	连接器的类型。 <ul style="list-style-type: none"> obs: 与OBS对接的连接器。 minio: 与MINIO对接的连接器。 objectstorageproxy: 对象存储代理类型。
connector_name	是	String	连接器实例的名称，即创建连接器时命名的连接器实例名称。
dir	是	String	要删除的目录，如果目录为空或者“/”，则代表删除根目录。

请求参数

无

响应参数

表 5-145 响应参数说明

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。

请求示例

已创建好OBS类型连接器，连接器实例名称为“命名空间_TestOBS”，OBS桶中存在“a/b”目录，执行如下请求，删除该目录。

```
DELETE https://AstroZero域名/u-route/baas/sys/v1.1/connectors/obs/命名空间_TestOBS/deldir?dir=a/b
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功"
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.7.5 列举目录下文件

功能介绍

列举存储中某目录下文件。

URI

GET *AstroZero*域名/u-route/baas/sys/v1.1/connectors/{connector_type}/{connector_name}/listdir?dir=path&max-keys=A&marker=B

表 5-146 路径参数

参数	是否必选	参数类型	描述
connector_type	是	String	连接器的类型。 <ul style="list-style-type: none"> obs: 与OBS对接的连接器。 minio: 与MINIO对接的连接器。 objectstorageproxy: 对象存储代理类型。
connector_name	是	String	连接器实例的名称，即创建连接器时命名的连接器实例名称。
dir	是	String	要列举的目录，如果目录为空或者“/”，则代表列举根目录。
max-keys	否	Number	最大访问文件个数，最大支持1000个，缺省值为“1000”。
marker	否	String	起始访问位置文件，注意起始位置文件并不会包含到返回结果中，且起始访问文件必须是全路径的，即包含目录。如果不指定，则从头开始。 示例：marker=a/b/c/1.txt

请求参数

无

响应参数

表 5-147 响应参数说明

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Array of objects	返回目录的文件列表，列表中元素字段含义如下： <ul style="list-style-type: none">• name: 文件名或目录路径• isDir: 是否为目录• size: 文件大小

请求示例

已创建好OBS类型连接器，连接器实例名称为“命名空间_TestOBS”，执行如下请求，列举OBS桶中“a/b/c”路径下文件，起始文件为“1.txt”。

```
GET https://AstroZero域名/u-route/baas/sys/v1.1/connectors/obs/命名空间_TestOBS/listdir?dir=a/b/c&max-keys=1000&marker=a/b/c/1.txt
```

响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": [
    {
      "name": "test.zip",
      "isDir": false,
      "size": 833154
    },
    {
      "name": "图片_20201220181511.png",
      "isDir": false,
      "size": 78530
    },
    {
      "name": "月度总结.txt",
      "isDir": false,
      "size": 824
    }
  ]
}
```

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.7.6 分段上传

对于较大文件上传，可以切分成段上传。用户可以在如下的应用场景内（但不仅限于此），使用分段上传的模式：

- 上传超过100MB大小的文件。
- 网络条件较差，和OBS等存储服务端之间的链接经常断开。
- 上传前无法确定将要上传文件的大小。

分段上传分为如下3个步骤：

1. **初始化分段上传任务**，即创建分段上传任务。
2. **上传分段**。
3. **完成上传合并段或终止（即取消）分段上传任务**。

完成上传合并段，文件会存储到桶中。终止（即取消）分段上传任务，之前所有上传的分段内容将会被丢弃。

初始化分段上传任务

- 功能介绍

使用分段上传方式传输数据前，必须先创建一个初始化分段上传任务。该操作会返回一个该任务的全局唯一标识（uploadId），用于标识本次分段上传任务。可以根据这个唯一标识来发起后续相关的操作，如上传分段任务、合并段完成上传、终止（即取消）分段上传任务。

- URI

POST *AstroZero*域名/u-route/baas/sys/v1.1/connectors/{connector_type}/{connector_name}/createmultipartupload?object=X

表 5-148 路径参数

参数	是否必选	参数类型	描述
connector_type	是	String	连接器的类型。 <ul style="list-style-type: none">• obs：与OBS对接的连接器。• minio：与MINIO对接的连接器。• objectstorageproxy：对象存储代理类型。
connector_name	是	String	连接器实例的名称，即创建连接器时命名的连接器实例名称。
object	是	String	上传文件到桶里的全路径，包含文件名、文件类型、上传到桶里的路径。不携带路径时，表示上传到桶的根目录下。 示例：取值为“a/b/c/test.mp4”，表示将文件“test.mp4”上传到桶的“a/b/c”路径下。如果桶中不存在该路径，会创建该路径。

- 请求参数

表 5-149 请求消息头参数说明

消息头名称	是否必选	参数类型	描述
Content-Type	是	String	上传文件的媒体类型，必须与文件名一致，比如对于“test.mp4”来说，“Content-Type”值为“video/mp4”。

- 响应参数

表 5-150 响应参数说明

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	返回创建的分段任务对象。对象中包含“uploadId”字段，表示该任务的全局唯一标识，用于标识本次分段上传任务。可以根据这个唯一标识，来发起后续相关的操作，如上传分段任务、合并段完成上传、终止（即取消）分段上传任务。

- 请求示例

已创建好OBS类型连接器，连接器实例名称为“命名空间_TestOBS”，需要向该OBS桶的“a/b/c”路径下上传大文件。调用如下请求，初始化分段上传任务。

```
POST https://AstroZero域名/u-route/baas/sys/v1.1/connectors/obs/命名空间_TestOBS/createmultipartupload?object=a/b/c/test.mp4
```

- 响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "uploadId": "0000017B761F112966C45D8FF0F75F13"
  }
}
```

请记录“uploadId”的值，用于标识本次分段上传任务，后续相关的操作需要使用到该标识。

上传分段

- 功能介绍

[初始化分段上传任务](#)之后，可以根据指定的对象文件名和“uploadId”来分段上传数据。每一个上传的段都有一个标识它的号码，即分段号（partNumber，范围是1~10000）。对于同一个分段上传任务（即uploadId相同），分段号“partNumber”不但唯一标识这一段数据，也标识了这段数据在整个对象文件内

的相对位置。如果使用同一个分段号上传了新的数据，那么存储服务端上已有的这个段号的数据将被覆盖。**除了最后一个分段以外**，其他分段大小必须**大于**5MB，最大值对于OBS、MINIO本身来说没有限制，通过AstroZero平台该接口调用时，所有分段最大值不超过100MB。每个分段不需要按顺序上传，甚至可以在不同进程、不同机器上上传，存储服务端会按照分段号排序组成最终对象。

- URI

POST *AstroZero*域名/u-route/baas/sys/v1.1/connectors/{connector_type}/{connector_name}/uploadpart?object=X&uploadId=Y&partNumber=Z

表 5-151 路径参数

参数	是否必选	参数类型	描述
connector_type	是	String	连接器的类型。 <ul style="list-style-type: none"> • obs: 与OBS对接的连接器。 • minio: 与MINIO对接的连接器。 • objectstorageproxy: 对象存储代理类型。
connector_name	是	String	连接器实例的名称，即创建连接器时设置的连接器实例名称。
object	是	String	上传文件到桶里的全路径，包含文件名、文件类型、上传到桶里的路径。不携带路径时，表示上传到桶的根目录下。 示例：取值为“a/b/c/test.mp4”，表示将文件“test.mp4”上传到桶的“a/b/c”路径下。如果桶中不存在该路径，会创建该路径。
uploadId	是	String	整个分段上传任务的标识，从 初始化分段上传任务 的返回结果中获取该值。
partNumber	是	Number	分段号，取值范围为1~10000。 对于同一个分段上传任务（即uploadId相同），分段号“partNumber”不但唯一标识这一段数据，也标识了这段数据在整个对象文件内的相对位置。

- 请求参数

请求中的body类型，按照“application/octet-stream”格式上传文件分段内容。

- 响应参数

表 5-152 响应参数说明

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。

参数	参数类型	描述
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	返回分段任务对象。对象中包含如下字段： <ul style="list-style-type: none"> ETag：文件分段的标识，用于后续完成上传时使用。 partNumber：分段号，用于后续完成上传时使用。

- 请求示例

已创建好OBS类型连接器，连接器实例名称为“命名空间_TestOBS”，需要向该OBS桶的“a/b/c”路径下上传大文件。已有初始化分段上传任务，标识“uploadId”为“0000017B761F112966C45D8FF0F75F13”，调用如下请求，创建分段上传任务，上传test.mp4，分段号为“1”。

POST `https://AstroZero域名/u-route/baas/sys/v1.1/connectors/obs/命名空间_TestOBS/uploadpart?object=a/b/c/test.mp4&uploadId=0000017B761F112966C45D8FF0F75F13&partNumber=1`

在请求Body中设置“application/octet-stream”格式的object，即要上传的文件数据。

- 响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "ETag": "\"d41d8cd98f00b204e9800998ecf8427e\"",
    "partNumber": 1
  }
}
```

请记录下“ETag”和“partNumber”的值，后续完成上传时需要使用。

完成上传合并段

- 功能介绍

所有分段上传完成后，需要调用完成上传接口，在存储服务端（如OBS）合并段生成最终对象文件。在执行完成上传操作时，需要提供所有有效的分段列表（包括分段号和分段ETag值）。AstroZero收到提交的分段列表后，会逐一验证每个段的有效性。当所有段验证通过后，将把这些分段组合成最终的对象。合并的分段任务可以是不连续的分段号。

完成上传任务后，“uploadId”就失效了，不能再用于终止（即取消）分段上传任务。

- URI

POST `AstroZero域名/u-route/baas/sys/v1.1/connectors/{connector_type}/{connector_name}/completeupload?object=X&uploadId=Y`

表 5-153 路径参数

参数	是否必选	参数类型	描述
connector_type	是	String	连接器的类型。 <ul style="list-style-type: none"> obs: 与OBS对接的连接器。 minio: 与MINIO对接的连接器。 objectstorageproxy: 对象存储代理类型。
connector_name	是	String	连接器实例的名称, 即创建连接器时命名的连接器实例名称。
object	是	String	上传文件到桶里的全路径, 包含文件名、文件类型、上传到桶里的路径。不携带路径时, 表示上传到桶的根目录下。 示例: 取值为“a/b/c/test.mp4”, 表示将文件“test.mp4”上传到桶的“a/b/c”路径下。如果桶中不存在该路径, 会创建该路径。
uploadId	是	String	整个分段上传任务的标识, 从 初始化分段上传任务 的返回结果中获取该值。

- 请求参数
数组类型, 每个元素中包含如[表5-154](#)所示字段。

表 5-154 请求参数说明

参数	是否必选	参数类型	描述
ETag	是	String	文件分段的标识。从 上传分段 的返回结果中获取该值。
partNumber	是	String	分段号。 对于同一个分段上传任务 (即uploadId相同), 分段号“partNumber”不但唯一标识这一段数据, 也标识了这段数据在整个对象文件内的相对位置, 从 上传分段 的返回结果中获取该值。

- 响应参数

表 5-155 响应参数说明

参数	参数类型	描述
resCode	String	返回码, 如果返回“0”代表请求成功, 其他错误码说明请参考 错误码 。

参数	参数类型	描述
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。

- 请求示例

已创建好OBS类型连接器，连接器实例名称为“命名空间_TestOBS”，需要向该OBS桶的“a/b/c”路径下上传大文件。已有初始化分段上传任务和两个分段上传任务，标识“uploadId”为“0000017B761F112966C45D8FF0F75F13”，调用如下请求，完成分段上传任务，合并所有分段生成最终对象文件。

POST `https://AstroZero域名/u-route/baas/sys/v1.1/connectors/obs/命名空间_TestOBS/completeupload?object=a/b/c/test.mp4&uploadId=0000017B761F112966C45D8FF0F75F13`

```
[[
  {
    "ETag": "\"d41d8cd98f00b204e9800998ecf8427e\"",
    "partNumber": 1
  }, {
    "ETag": "\"7dcad0c8a7398333ff25418bad511581\"",
    "partNumber": 2
  }
]]
```

- 响应示例

```
{
  "resCode": "0",
  "resMsg": "成功"
}
```

终止（即取消）分段上传任务

- 功能描述

分段上传任务可以被取消，当一个分段上传任务被取消后，就不能再使用其“uploadId”做任何操作，已经上传段也会被删除。

采用分段上传方式上传文件过程中或上传文件失败后会在桶内产生段，这些段会占用存储空间，可以通过取消该分段上传任务来清理掉不需要的段，节约存储空间。

- URI

POST `AstroZero域名/u-route/baas/sys/v1.1/connectors/{connector_type}/{connector_name}/abortupload?object=X&uploadId=Y`

表 5-156 路径参数

参数	是否必选	参数类型	描述
connector_type	是	String	连接器的类型。 <ul style="list-style-type: none"> obs：与OBS对接的连接器。 minio：与MINIO对接的连接器。 objectstorageproxy：对象存储代理类型。
connector_name	是	String	连接器实例的名称，即创建连接器时，设置的连接器实例名称。

参数	是否必选	参数类型	描述
object	是	String	上传文件到桶里的全路径，包含文件名、文件类型、上传到桶里的路径。不携带路径时，表示上传到桶的根目录下。 示例：取值为“a/b/c/test.mp4”，表示将文件“test.mp4”上传到桶的“a/b/c”路径下。如果桶中不存在该路径，会创建该路径。
uploadId	是	String	整个分段上传任务的标识，从 初始化分段上传任务 的返回结果中获取该值。

- 请求参数
无
- 响应参数

表 5-157 响应参数说明

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。

- 请求示例
已创建好OBS类型连接器，连接器实例名称为“命名空间_TestOBS”，需要向该OBS桶的“a/b/c”路径下上传大文件。已有初始化分段上传任务和分段上传任务，标识“uploadId”为“0000017B76500F0662CABA7128913936”，调用如下请求，取消该分段上传任务。

```
POST https://AstroZero域名/u-route/baas/sys/v1.1/connectors/obs/命名空间_TestOBS/abortupload?object=a/b/c/test3.mp4&uploadId=0000017B76500F0662CABA7128913936
```

- 响应示例

```
{
  "resCode": "0",
  "resMsg": "成功"
}
```

5.7.7 获取 endpoint，即文件存储地址的前缀

endpoint为文件存储地址的前缀，该前缀加上传对象的路径，是实际访问对象的全路径。[上传文件](#)返回的是上传文件在桶上的路径。通过调用本节接口，返回endpoint，组装起来才是文件完整的访问地址。

URI

```
GET AstroZero域名/u-route/baas/sys/v1.1/connectors/{connector_type}/{connector_name}/getendpoint
```

表 5-158 路径参数

参数	是否必选	参数类型	描述
connector_type	是	String	连接器的类型。 <ul style="list-style-type: none">obs: 与OBS对接的连接器。minio: 与MINIO对接的连接器。objectstorageproxy: 对象存储代理类型。
connector_name	是	String	连接器实例的名称, 即创建连接器时, 设置的连接器实例名称。

请求参数

无

响应参数

表 5-159 响应参数说明

参数	参数类型	描述
resCode	String	返回码, 如果返回“0”代表请求成功, 其他错误码说明请参考 错误码 。
resMsg	String	返回消息, 如果成功状态, 通常会返回“Success”, 其他情况会返回具体的错误信息。
result	Object	如果请求成功, 返回“endpoint”, 即文件存储地址的前缀。“endpoint”和 上传文件 返回的上传文件, 在桶上的路径组装起来才是文件完整的访问地址。

请求示例

已创建好OBS类型连接器, 连接器实例名称为“命名空间_TestOBS”, 执行如下请求, 获取endpoint, 即文件存储地址的前缀。

```
GET https://AstroZero域名/u-route/baas/sys/v1.1/connectors/obs/命名空间_TestOBS/getendpoint
```

响应示例

```
{  "resCode": "0",  "resMsg": "成功",  "result": {    "endpoint": "https://test.obs.***.com/"  }}
```

调用[上传文件](#)接口在桶“a/b/c”路径下成功上传了“1.txt”文件后, 该文件的完整访问地址为“https://test.obs.***.com/a/b/c/1.txt”。

状态码

状态码请参见[状态码](#)。

错误码

错误码请参见[错误码](#)。

5.7.8 分享 Token

通过调用[获取分享Token](#)，可以获取临时Token，用户只要使用该临时Token即可访问文件，不用再使用“access-token”进行鉴权。

分享Token分为如下2个步骤：

1. [获取分享Token](#)，获取临时Token。
2. [查看分享对象文件](#)，使用临时Token访问文件。

获取分享 Token

- 功能介绍
用于获取临时Token，使用该Token可访问文件，不用再使用“access-token”进行鉴权。获取的Token是临时性的，存在有效期。

- URI

POST *AstroZero*域名/u-route/baas/sys/v1.1/connectors/{connector_type}/{connector_name}/sharetoken

表 5-160 路径参数

参数	是否必选	参数类型	描述
connector_type	是	String	连接器的类型。 <ul style="list-style-type: none"> • obs: 与OBS对接的连接器。 • minio: 与MINIO对接的连接器。 • objectstorageproxy: 对象存储代理类型。
connector_name	是	String	连接器实例的名称，即创建连接器时，设置的连接器实例名称。

- 请求参数

表 5-161 请求 body 参数

参数	是否必选	参数类型	描述
shareType	是	Integer	分享类型： <ul style="list-style-type: none"> • 0: 代表按文件分享。 • 1: 代表按目录分享。 • 2: 用字符数组表示多个文件路径。

参数	是否必选	参数类型	描述
key	否	String	分享的对象、目录。 如果是目录的话，必须以“/”结尾，表示是个目录。
keys	否	String	多个文件路径。
timeoutInMinutes	是	Integer	分享的有效期，单位为分钟。 取值范围：1~1440。即最短1分钟，最长24小时。

- 响应参数

表 5-162 响应参数说明

参数	参数类型	描述
resCode	String	返回码，如果返回“0”代表请求成功，其他错误码说明请参考 错误码 。
resMsg	String	返回消息，如果成功状态，通常会返回“Success”，其他情况会返回具体的错误信息。
result	Object	返回结果对象。对象中包含“shareToken”字段，表示临时Token。

- 请求示例

已创建好OBS类型连接器，连接器实例名称为“命名空间_TestOBS”。调用如下请求，获取某些文件的临时访问Token。

POST `https://AstroZero域名/u-route/baas/sys/v1.1/connectors/obs/命名空间_TestOBS/sharetoken`

```
{
  "shareType": 0,
  "key": "a/b/c/1.txt",
  "keys": [
    "a/b/c/2.txt",
    "ROMAExchange.zip",
    "some/dir/c.jpg"
  ],
  "timeoutInMinutes": 60
}
```

 **注意**

使用者拿到URL后即可访问，所以要避免分享敏感文件。

- 响应示例

```
{
  "resCode": "0",
  "resMsg": "成功",
  "result": {
    "shareToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzIjotOeij6NEo"
  }
}
```

请记录下“shareToken”的值，后续访问文件使用。

查看分享对象文件

- 功能介绍
使用[获取分享Token](#)，返回的临时Token查看文件。
- URI
GET *AstroZero*域名/u-route/baas/sys/v1.1/objectstorage/viewshareobject?
object=X&shareToken=Y

表 5-163 路径参数

参数	是否必选	参数类型	描述
object	是	String	文件所在桶里的全路径，包含文件名。
shareToken	是	String	获取分享Token 中，返回的临时Token。

- 请求参数
无
- 响应参数
只返回文件内容。
- 请求示例
已创建好OBS类型连接器，连接器实例名称为“命名空间_TestOBS”。调用如下请求，使用临时访问Token查看桶中的“a/b/c/1.txt”文件。
GET *https://AstroZero*域名/u-route/baas/sys/v1.1/objectstorage/viewshareobject?object=a/b/c/1.txt&shareToken=*eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzaGFuZGVyZVR5cGUjAsInNoYXJIS2V5IjoiYS9iL2MvMS50eHQiLCJzaGFuZGVyZUtleXMiOlsiYS9iL2MvMi50eHQiLCJST01BRXhjaGFuZ2Uuemlwi5pyl5bq5oC757uTlnR4dCjJb25uZWN0b3JlUeXBljoib2JzliwiY29ubmVjdG9yljoiZ3pfX1Rlc3RPQIMiLCJzZXNzaW9uIjpw7lRlbnFudEEljoiMDAwMDAwMDAwMGJaSFUzVjMxUXUiLCJvc2VySUQiOiIxMGdkMDAwMDAwYlplIVUFXN1BlYSIsIkxvY2FsZSI6eyJMYW5nljoiemhfQ04ifX0slmV4cCI6MTYyOTc5MTA3Ny45Nzc5OH0.HQIMtULKsjTOgbtoSwOdqwX4W8FYgXTGTAtOejj6NEo*
- 响应示例
test

6 脚本中预置的 API

AstroZero平台的脚本引擎采用TypeScript语言。脚本执行时，TypeScript语言会被翻译成JavaScript语言，由JavaScript引擎执行。

在JavaScript es5的官方标准库外，AstroZero还扩展了10+预置标准库（即预置API），帮助您更高效地开发脚本，标准库说明如表6-1所示。有关预置标准库的详细说明，请参考[脚本标准库](#)。

库的使用方法举例：

```
import * as http from 'http';
```

表 6-1 标准库说明

标准库名称	说明
sys	提供系统层面的操作，例如获取系统参数值、提供alarm脚本库在脚本里埋点上报告警。
context	脚本执行的上下文，例如获取当前租户ID、获取当前登录用户ID等。
action	定义脚本的输入输出方法，以及调用脚本的方法。
buffer	操作二进制缓存区，例如将缓冲区内容转换为字符串。
setup	对系统对象Standard Object的增、删、改、查。
db	对自定义对象Customer Object的增、删、改、查。
sql	执行sql语句，目前只支持执行select查询语句。
bo	对Business Object的增、删、改、查。
meta	操作Object的元数据，例如获取字段名称、获取字段长度。
http	http或者https操作。
iconv	将字符串转为GBK、GB18030、GB2132或ISO8859_1编码格式，或者解码字符串。
crypto	加密操作，包含对哈希、HMAC、加密、解密、签名及验证的封装。

标准库名称	说明
decimal	将从http请求或从数据库中查询出来的number类型值，转换成Decimal类型，并能进行加、减、乘、除等数学运算。
excel	操作Excel文件，例如生成Excel文件。
text	操作文本，例如将文本转换为二进制。
xml	操作XML文件，例如读取XML文件内容。
uuid	生成全局唯一标识。

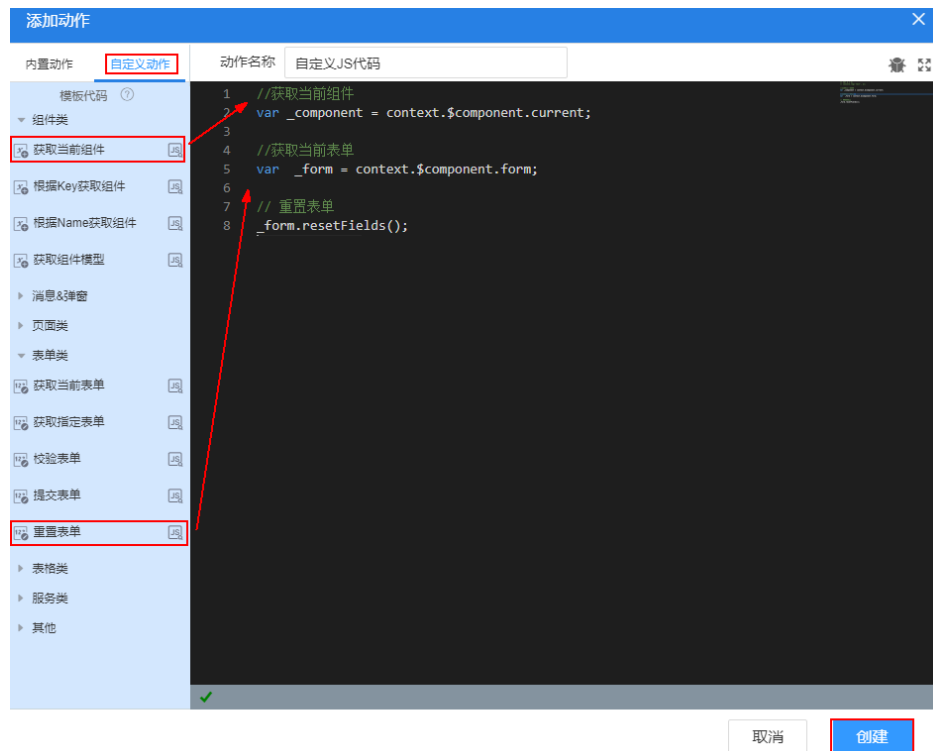
7 标准页面中预置的 API

在开发标准页面时，可以通过编写JS代码，来实现页面组件与后台接口之间的交互。如图7-1所示，在编写JS代码时，可直接使用系统预置的API。

系统预置的API代码结构样例如下，提供事件上下文context，用于封装常用的对象及API。

```
context
├── $app // 当前app信息
├── $user // 当前用户信息
├── $page // 页面相关操作
├── $component // 组件相关操作
├── $model // 数据模型相关操作
├── $message // 提示消息
├── $dialog // 对话框
├── $bp // BPM相关操作
├── $bpm // BPM相关操作，推荐使用该API，原子级操作
└── $params // 获取组件事件的入参
```

图 7-1 使用内置的模块代码



下面按照分类介绍标准页面中预置API。

组件类

- 获取当前组件
context.\$component.current
- 根据KEY获取组件
API: context.\$component.get(_compKey: number)
入参: 组件KEY, 类型为number。
出参: Component。
示例如下:

```
let _compKey = 1546395654403000
let _component = context.$component.get(_compKey)
```
- 根据名称获取组件
API: context.\$component.get(_compName: string)
示例如下:

```
let _compName = 'input_1'
let _component = context.$component.get(_compName)
```
- 获取组件绑定的模型
API: _component.getDataModel()
_component为当前操作的组件。
示例如下:

```
let _component = context.$component.current
let _model = _component.getDataModel()
```

消息及弹框

📖 说明

消息及弹框的默认显示时间为5秒，可以通过传入第二个参数duration，来自定义显示时间。
例如，自定义提示成功消息时间为8秒，执行语句如下：

```
API: context.$message.success(message: string, duration: 8)
```

- 提示成功消息

API: context.\$message.success(message: string)

示例：

```
// 成功消息  
context.$message.success('Successfully')
```

- 提示错误消息

API: context.\$message.error(message: string)

示例：

```
// 失败消息  
context.$message.error('Failed')
```

- 普通消息弹出框

API: context.\$dialog.info(option: object)

入参: option, 对话框参数, 结构如下：

```
{  
  title: string, // 标题  
  content: string, // 内容  
  okText: string, // 确认按钮名称, 可选  
  onOk: function // 确认回调函数, 可选  
}
```

示例：

```
// 普通弹出消息  
context.$dialog.info({ title: 'Info', content: 'content', okText: 'ok', onOk: () => {alert('click ok')}})
```

- 确认消息弹出框

API: context.\$dialog.confirm(option: object)

入参: option, 对话框参数, 结构如下：

```
{  
  title: string, // 标题  
  content: string, // 内容  
  okText: string, // 确认按钮名称, 可选  
  cancelText: string, // 取消按钮名称, 可选  
  onOk: function, // 确认回调函数, 可选  
  onCancel: function // 取消回调函数, 可选  
}
```

示例：

```
// 确认弹出消息  
context.$dialog.confirm({ title: 'Confirm', content: 'content', okText: 'ok', cancelText: 'cancel', onOk: () => {alert('click ok')}, onCancel: () => {alert('click cancel')}})
```

- 成功消息弹出框

API: context.\$dialog.success(option: object)

入参: option, 对话框参数, 结构如下：

```
{  
  title: string, // 标题  
  content: string, // 内容  
  okText: string, // 确认按钮名称, 可选  
  onOk: function // 确认回调函数, 可选  
}
```


示例:

```
// 成功弹出消息  
context.$dialog.success({ title: 'Info', content: 'content', okText: 'ok', onOk: () => {alert('click ok')}})
```

- 错误消息弹出框

API: context.\$dialog.error(option: object)

入参: option, 对话框参数, 结构如下:

```
{  
  title: string, // 标题  
  content: string, // 内容  
  okText: string, // 确认按钮名称, 可选  
  onOk: function // 确认回调函数, 可选  
}
```

示例:

```
// 错误弹出消息  
context.$dialog.error({ title: 'Error', content: 'content', okText: 'ok', onOk: () => {alert('click ok')}})
```

- 弹出页面

API: context.\$dialog.popup({ title: 'Error', page: 'date111', width: 60, height: 500, okText: 'ok', loading: true, params: {}, onCancel: () => {alert('click cancel')}, onOk: (data) => { data.visible = false; alert('click ok')}})

入参: option, 对话框参数, 结构如下:

```
{  
  title: string, // 标题  
  page: string // 弹出页面名称  
  width: integer // 页面宽度, 小于100为百分比, 大于100为px  
  height: integer // 页面高度  
  okText: string, // 确认按钮名称, 可选  
  params: object // 页面参数  
  onCancel: function // 取消时回调函数, 可选  
  onOk: function // 确认回调函数, 可选  
  maskClosable: boolean // 是否允许点击遮罩层关闭, 默认true  
  footerHide: boolean // 是否隐藏按钮(确定、取消)区域, 默认false  
  loading: boolean // 点击确定按钮时, 确定按钮是否显示 loading 状态, 开启则需在onOk中手动设置 visible来关闭对话框  
  draggable: boolean // 是否可以拖动, 默认false  
  showCancel: boolean // 是否展示取消按钮, 默认true  
  okFirst: boolean // 是否确定按钮在取消按钮前面, 默认false  
}
```

📖 说明

loading:false用于确定按钮是否显示loading状态, 默认为false即不显示loading状态, 如果设置为true显示loading, onOk里需要使用visible来关闭对话框。

footerHide:false用于控制弹出页面上的按钮是否显示, 默认为false表示不显示, 设置为true可显示按钮。

- 子页面中关闭当前页面

API: Page.\$popup.get(Page.\$root.popupKey).close();

页面类

- 获取当前页面

API: context.\$page.current

- 当前页面绑定的数据模型

API: context.\$page.getModel()

- 页面URL参数

API: context.\$page.params.paramName

示例:

```
// 假设当前页面URL为: https://localhost:8080/besBaas/page#/std_1?a=1&b=2, 则获取URL参数方式如下:
```

```
let paramA = context.$page.params.a  
let paramB = context.$page.params.b
```

- 在新的导航条打开页面

API: context.\$page.open(url: string, queryString: string)

入参: url表示页面地址。queryString表示查询参数 (URL参数), 格式为“key=value”, 多个参数用“&”符号连接。

示例:

```
context.$page.open('https://localhost:8080/besBaas/page#/std_2', 'a=1&b=2')
```

- 在当前导航条打开页面

API: context.\$page.load(url: string, queryString: string)

- 刷新当前页面

API: context.\$page.reload()

- 在新的导航条打开标准页面

API: context.\$page.openStdPage(pageName: string, queryString: string)

入参: pageName表示页面名称。queryString表示查询参数 (URL参数), 格式为“key=value”, 多个参数用“&”符号连接。

示例:

```
context.$page.openStdPage('std_2', 'a=1&b=2')
```

- 在当前导航条打开标准页面

API: context.\$page.loadStdPage(pageName: string, queryString: string)

- 在新的导航条打开高级页面

API: context.\$page.openAdvPage(pageName: string, websiteName:string, queryString: string)

入参: pageName表示页面名称。websiteName表示为高级页面的站点名称, queryString表示查询参数 (URL参数), 格式为“key=value”, 多个参数用“&”符号连接。

- 在当前导航条打开高级页面

API: context.\$page.loadAdvPage (pageName: string, websiteName:string, queryString: string)

- 关闭当前页面

API: window.close()

图 7-2 关闭页面语句

```
// 成功消息
context.$message.success('保存成功');
if (!Page.params.id) {
  // 当前表单
  let _form = context.$component.form
  _form.resetFields();
}
else {
  window.close();
  //关闭页面
}
}).catch((e => {
  this.$dialog.error({
    title: '错误',
    content: e.resMsg
  });
}));
```

表单类

- 获取当前表单
API: context.\$component.form
如果当前组件为表单，则直接返回。否则，根据组件上下文向上查找最近的表单组件。
- 根据组件Key值获取表单
API: context.getFormByKey(_formKey: number)
入参: 表单组件的Key值，类型为number。
- 校验整个表单
 - API: _form.formValidate()
_form为已获取的表单组件，返回promise。
 - API: _form.formValidateUnPromise
非promise，返回true/false。
- 表单的单字段校验:
API: _form.validateField(prop, callback)
参数prop为需校验的属性，参数callback为检验完回调函数，返回错误信息。
示例:

```
_form.validateField("name",function(errorMsg){
  console.log("validate field", errorMsg)
});
```
- 表单重置
API: _form.resetFields()
_form为已获取的表单组件。
- 表单提交
API: _form.formSave()

`_form`为已获取的表单组件。API内部实现已先进行表单校验，校验成功后进行提交保存操作。

表格类

- 获取当前表格

API: `context.$component.table`

如果当前组件为表格，则直接返回。如果当前组件有关联的表格，则返回对应的表格（如表格查询条件、分页等）。否则，根据组件上下文向上查找最近的表格组件。

- 获取查询条件

API: `_table.getCondition()`

`_table`为已获取的表格组件，返回表格绑定的查询条件。

- 设置查询条件

API:

```
// 先获取条件，再进行赋值
var condition = _table.getCondition();
// 字段满足单个条件，默认操作是contains
condition.queryInfo = {
  "fieldName": "1"
}
// 字段满足单个条件，使用自定义操作
condition.queryInfo = {
  "fieldName": {
    operator: "contains",
    value: "1"
  }
}
// 字段满足多个条件
condition.queryInfo = {
  "fieldName": [{
    operator: "contains",
    value: "1"
  }],
  {
    operator: "contains",
    value: "2"
  }
}
// 分页条件
condition.pageInfo: {
  "pageSize": 10,
  "curPage": 1
}
```

其中，“`_table`”为已获取的表格组件。

入参：表格查询条件。

- 执行查询

API: `_table.doQuery()`

其中，“`_table`”为已获取的表格组件。

- 初始化行数据

API: `_table.rowInit()`

其中，“`_table`”为已获取的表格组件，初始化一条新的行数据，此方法执行后，数据并未添加到表格。

- 添加行数据

API: `_table.addRow(newRowData: object)`

其中，“_table”为已获取的表格组件，此方法将初始化好的行数据添加到表格，此方法执行后，数据仅在前端表格保存，并未持久化到后端。

入参：行数据。

- 获取表格当前选中的行数据

API: `_table.getSelectedData()`

其中，“_table”为已获取的表格组件。

- 删除表格行数据

API: `_table.doDelete(rows: array)`

其中，“_table”为已获取的表格组件。

入参：表格行数据列表。

示例：

```
// 当前表格
let _table = context.$component.table
// 获取当前选中的行
let rows = _table.getSelectedData()
// 删除选中行
_table.doDelete(rows)
```

- 获取表格修改的数据（新增、修改、删除）

API: `_table.getOperatedData()`

其中，“_table”为已获取的表格组件。

- 保存表格数据

API: `_table.doSave(operatedData: object)`

其中，“_table”为已获取的表格组件，此方法将当前表格做的修改操作进行持久化操作。

入参：修改的表格数据。

示例：

```
// 当前表格
let _table = context.$component.table
// 获取修改的数据
let operatedData = _table.getOperatedData()
// 提交表格修改
_table.doSave(operatedData)
```

对象操作

- 获取对象

API: `context.object(objectName)`

入参：objectName，对象名称。

出参：对象。

示例如下：

```
let _object = context.object("objectName")
```

- 查询对象所有数据

API:

```
_object.query().then(function (response) {
  // TODO: Your business logic
})
```

- 查询满足条件数据

例如，查询满足name = 'test' 条件的对象数据，代码如下：

```
let _condition = {conjunction: 'AND', conditions: [{ field: 'name', operator: 'eq', value: 'test' }]}
_object.query({condition:_condition}).then(function (response) {
  // TODO: Your business logic
})
```

- 统计满足条件的对象数据数量

API: `_object.count(_condition)`

其中，“_object”为对象名称。

入参: `_condition`，查询条件，可选。

出参: Promise。

示例：

```
// 统计所有数据
_object.count().then(function (response) {
  // TODO: Your business logic
})

// 统计满足条件数据
_object.count(_condition).then(function (response) {
  // TODO: Your business logic
})
```

- 查询满足条件数据，且只查询部分字段

```
let _fields = 'id,name'
_object.query({condition:_condition}, _fields).then(function (response) {
  // TODO: Your business logic
})
```

- 查询满足条件数据，且只查询部分字段，支持分页和排序

例如，根据name字段降序排序，跳过前10条记录分页，每页最多显示10条数据。示例代码如下：

```
let _fields = 'id,name'
let _options= {'skip': 10,'limit': 10,'orderby': [{'field': 'name','order': 'desc'}]}
_object.query({condition:_condition}, _fields, _options).then(function (response) {
  // TODO: Your business logic
})
```

- 通用对象数据查询

API: `_object.query({condition:_condition}, fields: string)`

入参: `_condition`，查询条件，可选，参考如下结构。

```
{
  conjunction: 'AND', // 条件连接符，AND 或 OR
  conditions: [
    {
      field: 'name', // 条件字段
      operator: 'eq', // 条件比较符
      value: 'test' // 条件值
    },
    {
      condition: { // 嵌入条件
        conjunction: 'OR',
        conditions: [
          {
            field: 'status',
            operator: 'eq',
            value: '1'
          },
          {
            field: 'status',
            operator: 'eq',
            value: '2'
          }
        ]
      }
    }
  ]
}
```

```
    }  
  }  
}  
]
```

以上示例表达内容为: `name = 'test' && (status = '1' || status = '2')`
`fields`: 查询字段 (只返回指定的字段), 多个字段以逗号分割, 可选。
出参: Promise。

- 插入对象数据, 支持批量操作

API: `_object.insert(data: Object[])`

入参: `data`, 待插入数据, 对象数组。

出参: Promise, 返回新插入数据的记录ID。

示例:

```
let data = [{ name: 'test', label: 'test' }]  
_object.insert(data).then(function (response){  
  if (response && '0' === response.resCode) {  
    let _id = response.result[0].id}})
```

- 根据记录ID更新对象数据

API: `_object.updateByID(_id: string, data: Object)`

入参: `id`表示记录ID, `data`表示对象更新内容。

出参: Promise。

示例:

```
let _id = '{{ id }}'  
// 根据记录ID更新数据  
_object.updateByID(_id, { label: 'test2' }).then(function (response) {  
  // TODO: Your business logic  
})
```

- 根据记录ID查询对象数据

API: `_object.queryByID(_id: string)`

入参: `id`表示记录ID。

出参: Promise。

示例:

```
let _id = '{{ id }}'  
// 根据记录ID查询单条数据  
_object.queryByID(_id).then(function (response) {  
  // TODO: Your business logic  
})
```

- 根据记录ID删除对象数据

API: `_object.deleteByID(_id: string)`

入参: `id`表示对象记录ID。

出参: Promise。

示例:

```
let _id = '{{ id }}'  
// 根据记录ID删除数据  
_object.deleteByID(_id).then(function (response) {  
  // TODO: Your business logic  
})
```

- 根据记录ID批量查询对象数据

API: `_object.batchQueryByIDs(_ids: string, fields: string)`

入参：ids表示记录ID，多个ID以逗号分割。fields表示查询字段，多个字段以逗号分割，可选。

出参：Promise。

示例：

```
_object.batchQueryByIds('id1,id2,id3').then(function (response) {  
  // TODO: Your business logic  
})
```

- 根据记录ID批量删除对象数据

API：_object.batchDeleteByIds(_ids: string)

入参：ids表示记录ID，多个ID以逗号分割。

出参：Promise。

示例：

```
_object.batchDeleteByIds('id1,id2,id3').then(function (response) {  
  // TODO: Your business logic  
})
```

- 批量插入或更新对象数据

API：_object.batchUpsert(data: Object[])

入参：data，待插入或更新的数据对象，包含id时表示进行更新。

出参：Promise。

示例：

```
let data = [{id: '1538033158273005', label: 'test'}, {name: 'test', label: 'test'}]  
  
_object.batchUpsert(data).then(function (response) {  
  // TODO: Your business logic  
})
```

脚本操作

- 初始化Script

API：context.script(scriptName)

入参：scriptName，表示Script名称。

出参：Script

示例：

```
// 初始化脚本  
let _script = context.script('{{ scriptName }}')
```

- 运行Script

API：_script.run()

入参：无。

出参：Promise。

示例：

```
// 初始化脚本  
let _script = context.script('{{ scriptName }}')  
  
// 运行脚本  
_script.run().then(function(response) {  
  // TODO: Your business logic  
})
```


服务编排相关

- 初始化Flow

API: `context.flow(flowName)`

入参: `flowName`表示Flow名称。

出参: Flow

示例:

```
// 初始化Flow
let _flow = context.flow("flowName")
```

- 设置Flow版本号

API: `_flow.version(version: string)`

入参: `version`表示flow版本号。

出参: Flow。

示例:

```
// 设置Flow版本号
_flow.version("0.0.1")
```

- 运行Flow

API: `flow.run(params: Object)`

入参: `params`, 输入参数, 如下示例:

```
{
  param1: 'param1',
  param2: {
    param21: 'param21',
    param22: 'param22'
  }
}
```

出参: Promise。

示例:

```
// 适用于一次调用, Flow即完成的场景
// 运行Flow
_flow.run({ param1: 'param1' }).then(function (res) {
  // TODO: Youre business logic
})
```

- 启动Flow

API: `_flow.start(params: Object)`

入参: `params`, 输入参数, 如下示例:

```
{
  param1: 'param1',
  param2: {
    param21: 'param21',
    param22: 'param22'
  }
}
```

出参: Promise。

示例:

```
// 以下适用于多步操作的业务场景
_flow.start({ param1: 'param1' }).then(function (res) {
  // TODO: Youre business logic
})
```

- 执行Flow下一步

API: `_flow.next(interviewID: string, params: Object)`

入参：interviewID为Flow运行实例ID，params为输入参数。

```
{
  param1: 'param1',
  param2: {
    param21: 'param21',
    param22: 'param22'
  }
}
```

出参：Promise。

示例：

```
// 下一步
_flow.next(`${ interviewID }`, { param1: 'param1' }).then(function (res) {
  // TODO: Youre business logic
})
```

- 返回Flow上一步

API：_flow.back(interviewID: string, params: Object)

入参：interviewID为Flow运行实例ID，params为输入参数。

```
{
  param1: 'param1',
  param2: {
    param21: 'param21',
    param22: 'param22'
  }
}
```

出参：Promise。

示例：

```
// 上一步
_flow.back(`${ interviewID }`, { param1: 'param1' }).then(function (res) {
  // TODO: Youre business logic
})
```

- 正常完成Flow

API：_flow.finish(interviewID: string, params: Object)

入参：interviewID表示Flow运行实例ID；params：输入参数，如下示例：

```
{
  param1: 'param1',
  param2: {
    param21: 'param21',
    param22: 'param22'
  }
}
```

出参：Promise。

示例：

```
// 完成Flow
_flow.finish(`${ interviewID }`, { param1: 'param1' }).then(function (res) {
  // TODO: Youre business logic
})
```

- 恢复Flow运行

API：_flow.resume(interviewID: string, params: Object)

入参：interviewID表示Flow运行实例ID；params：输入参数，如下示例：

```
{
  param1: 'param1',
  param2: {
    param21: 'param21',
    param22: 'param22'
  }
}
```

出参：Promise。

示例：

```
// 恢复Flow
_flow.resume("{{ interviewID }}", { param1: 'param1' }).then(function (res) {
  // TODO: Youre business logic
})
```

- 删除Flow实例

API：_flow.terminate(interviewID: string, params: Object)

入参：interviewID表示Flow运行实例ID；params：输入参数，如下示例：

```
{
  param1: 'param1',
  param2: {
    param21: 'param21',
    param22: 'param22'
  }
}
```

出参：Promise。

示例：

```
// 删除Flow
_flow.terminate("{{ interviewID }}", { param1: 'param1', param2: { param21: 'param21', param22:
'param22' } }).then(function (res) {
  // TODO: Youre business logic
})
```

服务请求

调用服务的API

API：context.service('{{url}}').run(_inputParams)

入参：“url”为服务接口的URL，在APP或者BO视图下单击“服务”，可查看到封装Flow、脚本或者对象操作接口的URL。“_inputParams”为输入参数。

示例：

```
var _inputParams = {};
// 调用服务
context.service('{{ url }}').run(_inputParams).then(function(response) {
  // TODO: Your business logic
});
```

BPM 相关（非原子级）

- 获取参数信息

API：context.\$bp.loadVariables()

入参：无。

出参：参数对象。

示例：

```
context.$bp.loadVariables().then(function (data) {
  context.$model.ref("loanRequestData").setData(data.result.LoanRequestData);
  context.$model.ref("statusUpdates").setData(data.result.Status);
  context.$model.ref("negotiateCount").setData(data.result.NegotiateCount);
});
```

- 设置BPM中的变量

API：context.\$bp.putVariables (variables:object, instId: string)

入参：“variables”为参数对象，必选；“instId”为实例ID，可选。

出参：执行结果。

示例：

```
context.$bp.putVariables({
  LoanRequest: "test"
});
```

- 数据提交到BPM

API: context.\$bp.submitTask (variables:object)

入参：“variables”为参数对象，必选。

出参：无。

示例：

```
var data = context.$model.ref("loanRequestData").getData();
context.$bp.submitTask({
  LoanRequest: data
});
```

BPM 相关（原子级）

📖 说明

context.\$bp和context.\$bpm的区别在于后者保证了操作的原子性。例如，调用context.\$bp.submitTask (variables:object)数据提交成功时，会有弹框提示且点击“确定”按钮会自动关闭当前页面。而调用context.\$bpm.submitTask (variables:object)提交时，只会提交任务而不会有后续的弹框提示等操作；并且当不在BPM上下文环境中提交任务时（即当BPM实例没有运行时提交任务），会有相关提示信息。

推荐使用context.\$bpm。

- 获取参数信息

API: context.\$bpm.loadVariables()

入参：无。

出参：参数对象。

示例：

```
context.$bpm.loadVariables().then(function (data) {
  context.$model.ref("loanRequestData").setData(data.result.LoanRequestData);
  context.$model.ref("statusUpdates").setData(data.result.Status);
  context.$model.ref("negotiateCount").setData(data.result.NegotiateCount);
});
```

- 设置BPM中的变量

API: context.\$bpm.putVariables (variables:object, instId: string)

入参：“variables”为参数对象，必选；“instId”为实例ID，可选。

出参：执行结果。

示例：

```
context.$bpm.putVariables({
  LoanRequest: "test"
});
```

- 数据提交到BPM

API: context.\$bpm.submitTask (variables:object)

入参：“variables”为参数对象，必选。

出参：无。

示例：

```
var data = context.$model.ref("loanRequestData").getData();
context.$bpm.submitTask({
```

```
LoanRequest: data
});
```

- 获取BPM实例中的所有变量和变量值
API: context.\$bpm.getVarsByInstanceld("instance_id")
入参: “instance_id” 为BPM实例的ID。
出参: BPM实例中的所有变量和变量值。
示例:

```
context.$bpm.getVarsByInstanceld("instance_id").then(function (resp) {
  // 回调逻辑
});
```

- 获取一个BPM实例数据
API: context.\$bpm.getBPIstance("instance_id")
入参: “instance_id” 为BPM实例的ID。
出参: BPM实例数据。
示例:

```
context.$bpm.getBPIstance("instance_id").then(function (resp) {
  // 回调逻辑
});
```

公共 API

获取csrf token

API: context.\$utils.getCSRFToken()

其他

- 启动定时任务
API: context.timerTask(_timerHandler, _timeout, _executionCondition);
入参: 分别是定时执行的业务逻辑、任务时间间隔、执行条件。
示例:

```
// 任务间隔时间, 毫秒
var _timeout = 1000;

// 定时执行业务逻辑
var _timerHandler = function() {
  // TODO: Your business logic
};

// 执行条件
var _executionCondition = function() {
  return true;
};

// 启动定时任务
context.timerTask(_timerHandler, _timeout, _executionCondition);
```

- 下载EDM上存储的文档, EDM是系统预置的EDM类型存储器。
API: context.\$edm.download()
示例:

```
// 下载文档
context.$edm.download('{{docId}}', '{{docType}}', '{{subClassName}}', '{{docVersion}}', function() {
  // 下载失败回调方法
});
```

- 预览EDM上存储的文档，EDM是系统预置的EDM类型存储器。

API: `context.$edm.preview()`

示例:

```
// 预览文档
context.$edm.preview(' {{hw_doc_id}} ', ' {{hw_doc_version}} ', ' {{docFormat}} ', function() {
  // 预览后回调方法
}, ' {{container}} ', function() {
  // 预览失败回调方法
});
```

- 拦截请求

API:

```
// 提供三个参数，url: 指定的URL; successFun: 请求成功回调方法; failFun: 请求失败回调方法。
// 根据需要填写，例：若不需要successFun可不提供。例如：{url:"", failFun:function(err){ }}
context.filter({
  url: "",
  successFun: function (response) {
    // TODO: Your business logic
  },
  failFun: function (error) {
    // TODO: Your business logic
  }
});
```

- 获取当前语言下的国际化信息。

API: `context.$t(name: string)`

入参: “name” 为多语言条目的唯一名称，必选。

出参: 当前语言的值。

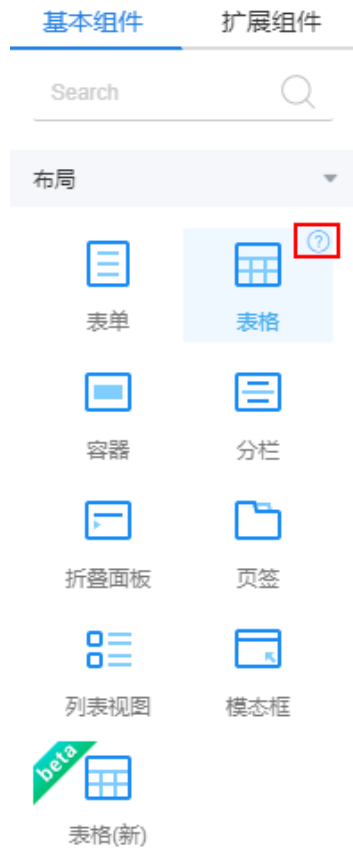
示例:

```
context.$t("app1.button.label")
```

- 获取入参

有些组件开放的事件是有入参的，在事件自定义代码中可以通过“`context.$params`”获取入参。`context.$params`值为数组，对应事件的多个入参。查看入参定义的方法是：请在标准页面设计界面左侧组件区域，将鼠标悬浮在组件上，组件右上会出现问号图标，单击该图标会出现该组件的使用说明，说明中包含事件入参定义。

图 7-3 查看组件说明




举例如下：

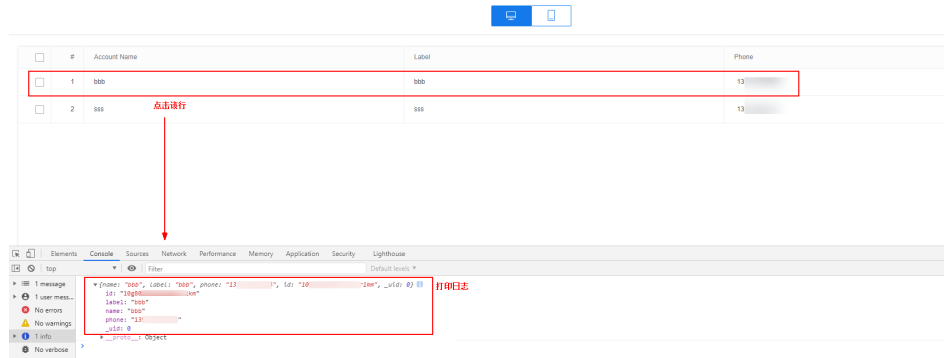
- a. 在表格的“事件”页签，单击“点击行”事件后的加号。



- b. 在“添加动作”窗口中输入如下代码：

```
var data = context.$params[0];  
console.log(data);
```
- c. 单击页面上方保存按钮，再单击页面上方预览按钮.

- d. 在弹出的预览浏览器页面按“F12”可开启调试工具。单击表格某一行触发事件。在“Console”页签查看日志相关信息。日志中会打印事件的入参，即点击的当前行数据。



模型相关

- 获取data model= model_1实例的数据。
API: `$model.ref("model_1").getData()`
- 设置data model= model_1实例的数据，入参data和model_1定义结构一致，当改变整个模型数据时使用。
API: `$model.ref("model_1").setData(data)`
- 设置data model= model_1实例的字段值，只改变已有模型数据某个字段值时使用。
API: `$model.ref("model_1").setValue(field, value)`

Tab 页相关

仅支持平台的运行态使用，可使用portal框架暴露的API来实现，平台运行态会暴露一个bingo全局变量供使用，标准页面由于在iframe内，需要使用`window.parent.bingo`访问：

- 打开tab页，返回tabId。
API: `openTab(name, url)`
- 切换tab页。
API: `switchTab(tabId)`
- 关闭指定Tab页。
API: `removeTab(tabId)`
- 当前Tab页。
API: `getCurrentTab()`
- 默认Tab页。
API: `getDefaultTab()`
- 打开指定Tab。
API: `open({type:'Tab', id: tabId})`
- 打开指定Tab。
API: `openTabByMenuName(menuName)`

- 返回上一个路由。
API: `back()`

8 高级页面和 Astro 大屏应用中预置的 API

在开发高级页面、Astro Canvas大屏或小屏页面过程中，系统为页面组件和桥接器提供了一些预置API。

组件

在开发自定义组件时，需要定义“*组件名.js*”文件，该文件为组件逻辑文件，是整个组件的渲染核心JS，在组件编辑状态和页面最终的发布运行态都会被加载执行。主要包含的预置API说明，请参见[表8-1](#)。

表 8-1 组件名.js 中预置 API 说明

分类	API及说明
组件实例预置API	<ul style="list-style-type: none"> ● init方法：主要包含组件渲染需要初始化数据的入口函数。 ● render方法：负责整个组件渲染的业务逻辑方法。 ● getConnectorProperties：获取桥接器属性的配置值。 ● getConnectorInstanceByName：通过桥接器的变量名称获取桥接器实例，其中“<i>APIConnector</i>”类型为String，表示桥接器的变量名称。 ● ConnectorIns.process(renderCb, errCb)：通过桥接器实例调用process函数，主要作用为发起调用服务的请求和处理返回结果信息。 <ul style="list-style-type: none"> - 在通过process函数发起请求前，需设置桥接器实例的requestParams属性为请求的参数，例如： ConnectorIns.requestParams = param; - 参数renderCb，类型为Function，当返回结果信息后调用该函数，用于处理返回信息。 - 参数errCb，类型为Function，调用该函数来处理错误。 ● ConnectorIns.query(param)：通过桥接器实例调用query函数，主要作用为发起调用服务的请求和处理返回结果信息，参数说明如下： <ul style="list-style-type: none"> - param：请求参数。 - 其返回结果为Promise对象。 ● getProperties：用于返回该组件的自定义属性值。 开发Widget时，可以给组件自定义属性，在开发界面可对这些属性进行配置，通过getProperties方法可返回自定义属性值。例如，var properties = thisObj.getProperties()。 ● getContainer：用于返回渲染该组件的容器dom节点。 代码示例：var elem = thisObj.getContainer ()，然后通过jquery去查询组件中某个dom节点时，可通过\$("#id", elem)的方式来获取。 ● getWidgetBasePath：用于返回该组件静态资源的根路径。 例如，某个组件的静态资源路径为“https://10.10.10.1:12900/default/0000000000NABzEjpNIH/assets/bundle/widget/172a6056501-a6f8ce1f-2ed9-4a9a-b883-251aac14e0a/v1591923270914/test0609.js”，执行thisObj.getWidgetBasePath()返回“https://10.10.10.1:12900/default/0000000000NABzEjpNIH/assets/bundle/widget/172a6056501-a6f8ce1f-2ed9-4a9a-b883-251aac14e0a/v1591923270914”。 ● getMessages：返回该组件国际化配置文件中定义的国际化的文件内容，需要配置vue和vuel18n使用。 例如，开发组件时定义了messages-en.json和messages-zh.json国际化文件。thisObj.getMessages()返回的就是这两个文件中定义的国际化的内容。 ● hideWidget：隐藏该组件的dom结构，预览时调用该方法可以隐藏Widget。

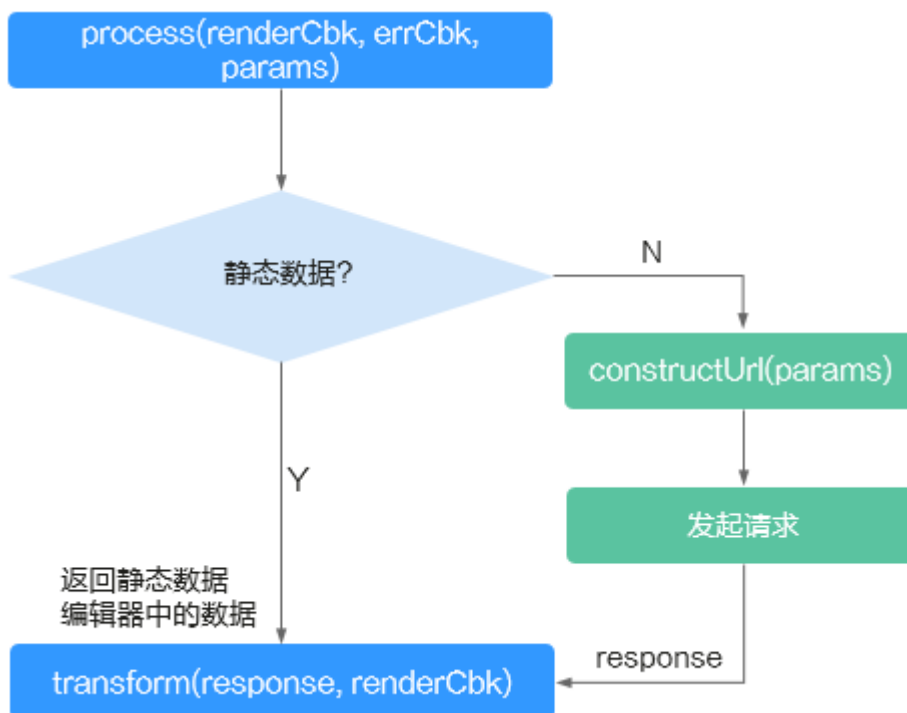
分类	API及说明
	<p>例如，执行thisObj.hideWidget()方法，隐藏该组件的dom结构。</p> <ul style="list-style-type: none"> • showWidget: 显示该组件的dom结构，预览时调用该方法可以展示Widget。 例如，执行thisObj.showWidget()方法，显示该组件的dom结构。 • SITE_ROOT_PATH: 为AstroZero预置的参数，用于获取当前站点URL地址的根路径。站点和应用是一一对应的关系，当创建应用时，系统默认会创建并分配一个站点。 假如，某个页面的URL地址为“https://10.10.10.1:12900/magno/render/cool_app_0000000000NABzEjpNIH/page1”，则使用SITE_ROOT_PATH获取的根路径为“/magno/render/cool_app_0000000000NABzEjpNIH”。 • triggerEvent: 用于触发事件，参数说明如下： <ul style="list-style-type: none"> - eventName: 指定触发的事件名称。 - {}: 为事件触发时传递的参数赋值，例如{param: value}。
Studio对象 预置API	<ul style="list-style-type: none"> • Studio.registerWidget: 用来定义组件。 • Studio.registerConnector: 用来定义桥接器。 • Studio.registerEvents: 用于注册事件，只有通过此API注册后的事件才会在组件的事件列表中展现。 <ul style="list-style-type: none"> - thisObj: 当前组件实例对象，指为该组件注册事件。 - "eventName": 事件名称，应该与触发事件API中的第一个参数保持一致。 - "Event Label": 事件标签名，在事件列表中展示的事件标题。 - {}: 定义该事件触发时传递的参数模型，例如[{"name": "param"}]。 • Studio.registerAction: 用来注册动作。只有通过此API注册后的事件，才会在组件的动作列表中展现。 <ul style="list-style-type: none"> - thisObj: 当前组件实例对象，指为该组件注册动作。 - actionName: 动作名称。 - Action Label: 动作标签名，在动作列表中展示的动作标题。 - {}: 事件触发时传入的参数。 - \$.proxy(this.receiveActionCb, this): 该动作的回调函数，在回调函数中定义该动作的执行逻辑。 - {}: 此参数在开发过程中置为空数组即可。 • Studio.registerRouter: 用来定义路由。 • Studio.inReader: 判断当前页面状态是开发态或预览态。其中，“true”表示在预览态（即运行态），“false”表示在开发态。 • StudioToolkit.getCatalogProperties(): 用于获取当前站点的元数据。

分类	API及说明
magno对象 预置API	<p>savePropertiesForWidget(props): 用于在开发态中保存当前组件的高级设置属性信息, 参数为要设置的相关属性集对象。</p> <p>参数props示例:</p> <pre>{ "prop1": "value1", "prop2": ""value2" }</pre>
HttpUtils工 具类预置API	<p>AstroZero平台内置了HttpUtils工具类, 预置API如下:</p> <ul style="list-style-type: none"> HttpUtils.getCookie: 用于获取某个cookie的值。 代码示例: HttpUtils.getCookie() HttpUtils.setCookie: 用于设置cookie的值。 代码示例: HttpUtils.setCookie("key","value") HttpUtils.getI18n: 返回一个VueI18n的实例, 配合Vue和VueI18n使用。该VueI18n实例可以在new一个Vue实例时赋值为i18n参数。 代码示例: HttpUtils.getI18n({locale: HttpUtils.getLocale(), messages: thisObj.getMessages()}) HttpUtils.getCsrfToken: 通过ajax的方式去调用平台的接口时使用。如果使用平台桥接器的方式, 平台会自动在请求header中, 添加csrf-token头。 <pre>HttpUtils.getCsrfToken(function(csrfToken) { \$.ajax({ ... headers: { CSRF-Token: csrfToken }, }) })</pre> HttpUtils.refreshToken: 用于手动刷新页面的accessToken, 通常用在执行退出逻辑时, 执行该方法刷新页面的access token。 HttpUtils.getUrlParam: 用于获取查询字符串中的参数。 例如, 某个页面的URL为“https://10.10.10.1:12900/magno/render/cool_app_0000000000NABzEjpNIH/page1?param=1”, 通过执行HttpUtils.getUrlParam("param")返回该参数的值“1”。 HttpUtils.getLocalStorage: 用于获取某个localStorage的值。 代码示例: HttpUtils.getLocalStorage("key") HttpUtils.setLocalStorage: 用于设置某个localStorage的值。 代码示例: HttpUtils.setLocalStorage("key","value")

桥接器

用户不需要自定义开发桥接器, 使用系统预置的“通用AstroZero API数据桥接器”即可调用应用的服务接口。AstroZero也支持用户自定义桥接器, 并上传到平台中, 提供给页面组件使用。在桥接器中, AstroZero对外开放的自定义部分主要执行流程如图8-1所示:

图 8-1 桥接器执行流程图




桥接器中主要的自定义函数为**createUrl**和**transform**，您可以按需求封装上述函数，以此实现不同功能的桥接器。自定义桥接器中常用的API如表8-2所示，在经典版应用设计器左侧列表中，单击，选择“高级页面 > 桥接器模板”，下载模板解压后可查看API调用示例。

表 8-2 自定义桥接器中常用的 API 列表

API名称	详细说明
init()	桥接器初始化函数，在该函数中主要实现为对setInputParams函数的调用，设置桥接器的自定义配置项。
setUrl(url)	在createUrl(params)中必须要调用该API，用于设置该桥接器请求的URL。

API名称	详细说明
setLoadMethod(type, dataType, callback, callbackName, ajaxOptions)	<p>在constructUrl(params)中调用该API，用于对请求constructUrl(params)的参数做处理。该API中参数说明如下：</p> <ul style="list-style-type: none"> • type：表示请求方式，支持POST、GET、PUT和DELETE。默认值为GET。 • dataType：预期服务器返回的数据类型。一般取值为json，表示返回JSON格式的数据。 • callback：dataType取值为“jsonp”的场景使用，表示在一个jsonp请求中重写回调函数的名字。 • callbackName：dataType取值为“jsonp”的场景使用，表示为jsonp请求指定一个回调函数名。 • ajaxOptions：AJAX请求设置，示例如下： <pre> { data: JSON.stringify(inputParamsObj.input), contentType: "text/html;charset=utf-8", beforeSend: function(xhr) { xhr.setRequestHeader("bass-method", "post"); xhr.setRequestHeader("Content-Type", "application/json"); } } </pre>
constructUrl(params)	<p>在发起请求前执行该API，该API用于构造请求的URL。其中，“params”为请求参数。</p> <p>使用说明：在该API中必须要调用setUrl(url)对URL赋值，还可调用setLoadMethod对请求参数做处理，该操作作为可选。</p>
transform(resultData, renderCbkb)	<p>处理请求响应，参数说明如下：</p> <ul style="list-style-type: none"> • resultData：请求的响应结果。 • renderCbkb：组件中定义的当请求获取响应信息后的回调函数。
getUrl()	<p>获取该桥接器请求的URL。</p>

API名称	详细说明
<p>setInputParams([{},...,{}])</p>	<p>设置桥接器的自定义配置项。</p> <ul style="list-style-type: none"> • type: 配置项的类型，常用类型包含“text”、“option”。 • name: 配置项的变量名称。 • label: 配置项的显示标题。 • value: 配置项的默认值。 • validation: 配置项的验证规则。 <p>桥接器自定义配置项代码示例：</p> <pre> this.setInputParams([{ "type": "select", "name": "requestMethod", "label": JSON.stringify({zh_CN: "请求方法", en_US: "Request Method"}), "options": [{ "label": "post", "value": "post", "selected": "true" }, { "label": "get", "value": "get" }, { "label": "put", "value": "put" }, { "label": "delete", "value": "delete" }], "validation": { "rules": {} } }, { "type": "text", "name": "prefix", "label": JSON.stringify({zh_CN: "前缀", en_US: "prefix"}), "value": "", "validation": { "rules": {} } }]); </pre>
<p>getInputParams()</p>	<p>获取桥接器的自定义配置项的设置值。</p>

9 权限管理

在AstroZero中，可以通过设置权限配置Profile，来配置用户权限。系统预置了如下几种Profile，禁止删除：

- System Administrator Profile：系统管理员，拥有AstroZero全部权限。
- Developer Profile：开发者权限，一般给用户使用，拥有此权限的用户可以在AstroZero进行开发，例如新增一个对象、为对象增加一个字段、新增一个流程等。
- Portal User Profile：业务用户的权限，一般给业务用户使用，拥有此权限的业务用户可以通过服务编排鉴权登录AstroZero。
- Anonymous User Profile：游客用户的访问权限，请根据自身业务需求，给Profile分配适当的权限。
- Standard User Profile：运行态权限，开通AstroZero服务时自带的权限，拥有这个权限可以运行系统中的流程、对系统已有对象进行记录的增删改查操作，但没有开发权限，如新增一个对象或新增一个流程等。
- NoCode Developer Profile：零代码应用开发权限，只有运行环境才会预置该权限。
- NoCode Manager Profile：零代码应用数据管理权限，只有运行环境才会预置该权限。

一个Profile可以分配给很多用户，但是每个用户有且只有一个Profile。Profile能够控制权限列表如下，具体Profile权限配置请参考[管理权限配置](#)。

- 基本权限控制
基本权限是指操作AstroZero内置的资源时，在接口上设置权限校验，当前Profile需要勾选上对应的权限项才能通过访问，否则会报权限错误，对应状态码“403”。例如，创建对象需要开发应用权限，创建用户需要管理用户权限等。
- App与Menu展示控制
控制用户在打开应用时，默认展示的应用及展示应用下的哪个菜单项。仅在前端做页面控制。
- 对象、字段权限控制
用户在访问具体对象字段数据记录时的权限控制。其中，对象权限包括：
 - 读取：读取对象记录的权限，没有此权限访问对象数据时会报无权限。
 - 创建：创建对象记录的权限。

- 编辑：修改对象记录数据的权限。
- 删除：删除对象记录的权限。
- 查看全部：能够查看该对象的所有记录，不管该记录是否属于操作者（OWNER）。
- 修改全部：能够增删改对象的所有记录，不管该记录是否属于操作者（OWNER）。
- API读：通过HTTP接口发起读取操作，不影响在服务编排、脚本等发起的操作。
- API写：通过HTTP接口发起写操作，不影响在服务编排、脚本等发起的操作。

某些特殊的对象不允许直接访问，如User对象，所有Profile只开放读取功能，增删改都要通过接口来操作。字段权限如下：

- 读：读取某个字段值的权限
- 写：修改某个字段值的权限

某些对象的某些字段不允许直接读取与修改，如User对象的密码字段等。

- 对象标准页面的展示控制
展示对象的默认布局，包括列表页，详情页，创建与编辑布局。仅供页面展示使用。
- 接口访问控制（服务编排、脚本、BPM）
在运行服务编排、脚本或BPM资源时需要检查是否拥有该资源的运行权限，如果没有则报错。例如，需要运行脚本“view_product”，则需要勾选上“view_product”的运行权限才能运行。
- 连接器访问控制
目前只向对外提供HTTP接口的连接器进行权限控制，包括S3、OBS、MINIO、ObjectStorageProxy类型连接器。
在运行这些连接器的接口时，需要检查其运行权限。
- 事件访问控制
目前只向对外提供HTTP接口的事件进行权限控制，例如发送某种类型事件的权限。
- 业务权限访问控制
是否拥有指定的业务权限。

10 附录

10.1 状态码

状态码如表10-1所示。

表 10-1 状态码

状态码	编码	错误码说明
100	Continue	继续请求。 这个临时响应用来通知客户端，它的部分请求已经被服务器接收，且仍未被拒绝。
101	Switching Protocols	切换协议。只能切换到更高级的协议。 例如，切换到HTTP的新版本协议。
200	ok	请求成功。
201	Created	创建类的请求完全成功。
202	Accepted	已经接受请求，但未处理完成。
203	Non-Authoritative Information	非授权信息，请求成功。
204	NoContent	请求完全成功，同时HTTP响应不包含响应体。 在响应OPTIONS方法的HTTP请求时返回此状态码。
205	Reset Content	重置内容，服务器处理成功。
206	Partial Content	服务器成功处理了部分GET请求。
300	Multiple Choices	多种选择。请求的资源可包括多个位置，相应可返回一个资源特征与地址的列表用于用户终端（例如浏览器）选择。

状态码	编码	错误码说明
301	Moved Permanently	永久移动，请求的资源已被永久的移动到新的URI，返回信息会包括新的URI。
302	Found	资源被临时移动。
303	See Other	查看其它地址。 使用GET和POST请求查看。
304	Not Modified	所请求的资源未修改，服务器返回此状态码时，不会返回任何资源。
305	Use Proxy	所请求的资源必须通过代理访问。
306	Unused	已经被废弃的HTTP状态码。
400	BadRequest	非法请求。 建议直接修改该请求，不要重试该请求。
401	Unauthorized	在客户端提供认证信息后，返回该状态码，表明服务端指出客户端所提供的认证信息不正确或非法。
402	Payment Required	保留请求。
403	Forbidden	请求被拒绝访问。 返回该状态码，表明请求能够到达服务端，且服务端能够理解用户请求，但是拒绝做更多事情，因为该请求被设置为拒绝访问，建议直接修改该请求，不要重试该请求。
404	NotFound	所请求的资源不存在。 建议直接修改该请求，不要重试该请求。
405	MethodNotAllowed	请求中带有该资源不支持的方法。 建议直接修改该请求，不要重试该请求。
406	Not Acceptable	服务器无法根据客户端请求的内容特性完成请求。
407	Proxy Authentication Required	请求要求代理的身份认证，与401类似，但请求者应当使用代理进行授权。
408	Request Time-out	服务器等候请求时发生超时。 客户端可以随时再次提交该请求而无需进行任何更改。
409	Conflict	服务器在完成请求时发生冲突。 返回该状态码，表明客户端尝试创建的资源已经存在，或者由于冲突请求的更新操作不能被完成。
410	Gone	客户端请求的资源已经不存在。 返回该状态码，表明请求的资源已被永久删除。

状态码	编码	错误码说明
411	Length Required	服务器无法处理客户端发送的不带Content-Length的请求信息。
412	Precondition Failed	未满足前提条件，服务器未满足请求者在请求中设置的其中一个前提条件。
413	Request Entity Too Large	由于请求的实体过大，服务器无法处理，因此拒绝请求。为防止客户端的连续请求，服务器可能会关闭连接。如果只是服务器暂时无法处理，则会包含一个Retry-After的响应信息。
414	Request-URI Too Large	请求的URI过长（URI通常为网址），服务器无法处理。
415	Unsupported Media Type	服务器无法处理请求附带的媒体格式。
416	Requested range not satisfiable	客户端请求的范围无效。
417	Expectation Failed	服务器无法满足Expect的请求头信息。
422	Unprocessable Entity	请求格式正确，但是由于含有语义错误，无法响应。
429	Too Many Requests	表明请求超出了客户端访问频率的限制或者服务端接收到多于它能处理的请求。建议客户端读取相应的Retry-After首部，然后等待该首部指出的时间后再重试。
500	InternalServerError	表明服务端能被请求访问到，但是不能理解用户的请求。
501	Not Implemented	服务器不支持请求的功能，无法完成请求。
502	Bad Gateway	充当网关或代理的服务器，从远端服务器接收到了一个无效的请求。
503	Service Unavailable	被请求的服务无效。 建议直接修改该请求，不要重试该请求。
504	Server Timeout	请求在给定的时间内无法完成。客户端仅在为请求指定超时（Timeout）参数时会得到该响应。
505	HTTP Version not supported	服务器不支持请求的HTTP协议的版本，无法完成处理。

10.2 错误码

调用接口出错后，将不会返回结果数据。调用方可根据每个接口对应的错误码来定位错误原因。当调用出错时，HTTP请求返回一个4xx或5xx的HTTP状态码。返回的消息

体中是具体的错误代码及错误信息。在调用方找不到错误原因时，可以联系华为云客服，并提供错误码，以便尽快帮您解决问题。

错误响应 Body 体格式说明

当接口调用出错时，会返回错误码及错误信息说明，错误响应的Body体格式如下所示。

```
{
  "resCode": "错误码",
  "resMsg": "错误信息"
}
```

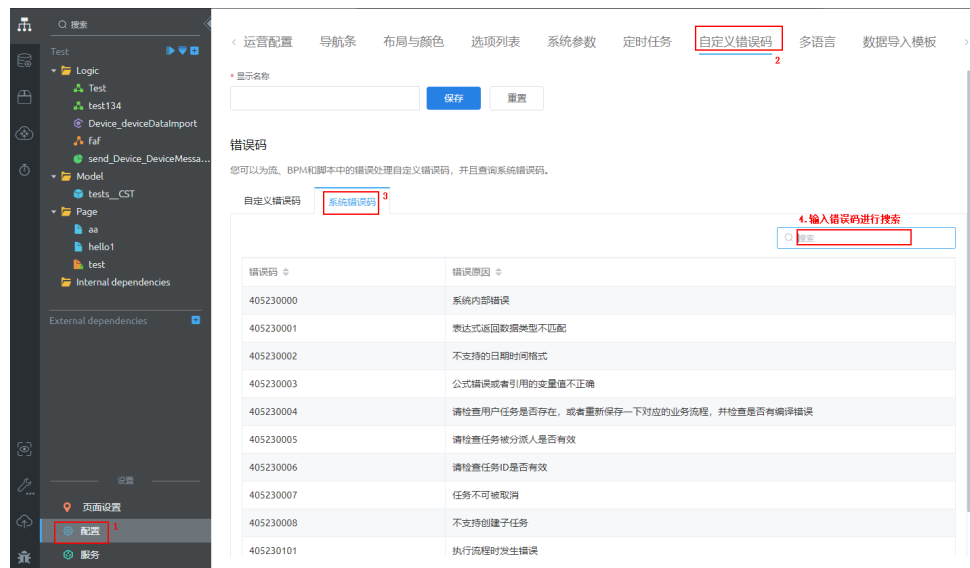
其中，resCode表示错误码，resMsg表示错误描述信息。

错误码说明

查看系统预置错误码有以下两个入口，请根据错误码查找具体的错误原因，并进行处理。例如“405230004”错误码，请根据错误码搜索，查看该错误码产生的原因，查看原因是“请检查用户任务是否存在，或者重新保存对应的业务流程，并检查是否有编译错误”，按照提示进行处理。如果不明确处理方式，请联系工程师处理。

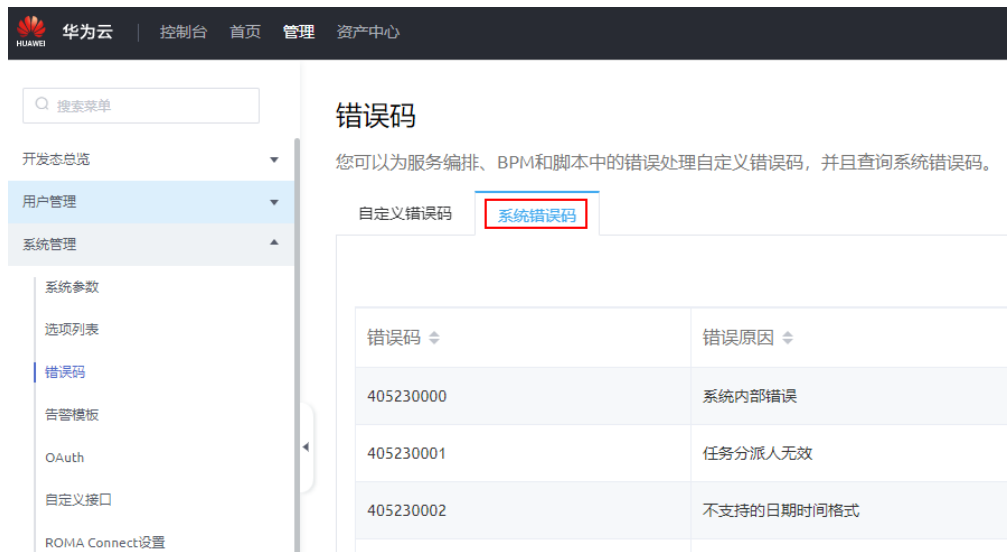
- 在**AstroZero经典版应用设计器**左侧菜单栏下方，选择“配置”。在打开的页签，选择“自定义错误码”。在“系统错误码”中，可进行查看。

图 10-1 App 内查看错误码



- 在**AstroZero经典版环境配置**中，选择“系统管理 > 错误码”，在“系统错误码”下可进行查看。

图 10-2 查看系统错误码



- 在**AstroZero新版环境配置**中，选择“维护 > 全局元素 > 错误码”，在“系统错误码”下可进行查看。

图 10-3 在新版环境配置，查看系统错误码



A 修订记录

发布日期	修订记录
2024-03-29	下线Astro Bot接口。
2023-04-17	新增API <ul style="list-style-type: none">一键更新动态实体：该接口用来更新技能下的动态实体和应用配置，并完成后续的训练和发布。查询动态实体更新进度：该接口用于查询动态实体更新进度。
2023-03-21	新增Astro Bot接口，华为云Astro智能助手（Astro Bot）为 白名单特性 ，请联系后台管理人员开通后方可使用。 <ul style="list-style-type: none">创建会话：该接口用于获取session_Id。发送消息：该接口用于发送消息进行会话。重启会话：该接口用于刷新对话流程。关闭会话：该接口用于关闭指定的会话。
2023-01-16	<ul style="list-style-type: none">脚本中预置的API：修改脚本标准库链接。标准页面中预置的API：消息及弹框中，增加修改默认显示时间的描述。
2022-11-01	<ul style="list-style-type: none">启动BPM实例：功能介绍中，增加“启动BPM实例前，需要先通过界面编排定义好BPM的元数据信息”的说明。高级页面和Astro大屏应用中预置的API：修改表8-1中，参数HttpUtils.getCsrfToken的描述。
2022-04-20	新增API <ul style="list-style-type: none">列出所有应用：新增查询所有应用的接口。根据ID查询应用：新增根据ID，查询应用的接口。
2022-02-08	<ul style="list-style-type: none">编排Flow：增加请求实例。新增对象数据：增加“新增对象数据前，需要先创建对象”的说明。

发布日期	修订记录
2021-05-17	<ul style="list-style-type: none">• 脚本中预置的API: 增加脚本中, 预置API的介绍。• 标准页面中预置的API: 新增标准页面中, 预置API的介绍。• 高级页面和Astro大屏应用中预置的API: 新增高级页面和Astro大屏应用中, 预置API的介绍。
2020-04-30	第一次正式发布