API列表

API名称	认证方式	API描述	所属资源目录结构
pg_type_1583830351893	平台认证	数据申请自动生成	

API调用说明

公共请求参数

公共请求参数是指每个接口都需要使用到的请求参数.

参数名称	位置	是否必须	参数描述
X-Ca-Key	Header	是	Appkey，调用API的身份标识，可以到【API申请】功能模块查看
X-Ca-Signature	Header	是	通过签名计算规则计算的请求签名串，参照：签名计算规则
X-Ca-Timestamp	Header	否	API 调用者传递时间戳，值为当前时间的毫秒数，也就是从1970年1月1日起至今的时间转换为毫秒，时间戳有效时间为15分钟
X-Ca-Nonce	Header	否	API请求的唯一标识符，15分钟内同一X-Ca-Nonce不能重复使用，建议使用UUID，结合时间戳防重放
Content-MD5	Header	否	当请求 Body 非 Form 表单时，需要计算 Body 的 MD5 值传递给云网关进行 Body MD5 校验
X-Ca-Signature-Headers	Header	否	指定哪些Header参与签名，支持多值以","分割，默认只有X-Ca-Key参与签名，为安全需要也请将X-Ca-Timestamp、X-Ca-Nonce进行签名，例如：X-X-Ca-Signature-Headers:Ca-Timestamp,X-Ca-Nonce
Accept	Header	否	指定返回结果以哪种形式返回，可选范围为【application/json、application/xml】。默认返回为json类型，此项可以为空；如果需要返回xml，此项不可以为空，且值须为application/xml

签名计算规则

请求签名，是基于请求内容计算的数字签名，用于API识别用户身份。客户端调用API时，需要在请求中添加计算的签名（X-Ca-Signature）。

签名计算流程

准备APPkey → 构造待签名字符串stringToSign → 使用Secret计算签名

第一步.准备APPKey

Appkey，调用API的身份标识，可以到【API申请】功能模块查看

第二步.构造待签名字符串stringToSign

```
String string2Sign=
```

```
HTTPMethod + "\n" +
Accept + "\n" + //建议显示设置 Accept Header。当 Accept 为空时，部分 Http 客户端会给 Accept 设置默认值为 */*，导致签名校验失败。
Content-MD5 + "\n"
Content-Type + "\n" +
Date + "\n" +
Headers +
Url
```

其中,

HTTPMethod

为全大写, 比如POST, GET, PUT等

```
Accept、Content-MD5、Content-Type、Date 如果为空也需要添加换行符”\n”，Headers如果为空不需要添加”\n”。
```

Content-MD5

Content-MD5 是指 Body 的 MD5 值，只有当 Body 非 Form 表单时才计算 MD5，计算方式为：
String content-MD5 = Base64.encodeBase64(MD5(bodyStream.getbytes("UTF-8")));
bodyStream 为字节数组。

Headers

Headers 是指参与 Headers 签名计算的 Header 的 Key、Value 拼接的字符串，建议对 X-Ca 开头以及自定义 Header 计算签名，注意如下参数不参与 Headers 签名计算：X-Ca-Signature、X-Ca-Signature-Headers、Accept、Content-MD5、Content-Type、Date。

Headers 组织方法

先对参与 Headers 签名计算的 Header的Key 按照字典排序后使用如下方式拼接，如果某个 Header 的 Value 为空，则使用 HeaderKey + “:” + “\n”参与签名，需要保留 Key 和英文冒号。

```
String headers =
HeaderKey1 + ":" + HeaderValue1 + "\n"\+
HeaderKey2 + ":" + HeaderValue2 + "\n"\+
...
HeaderKeyN + ":" + HeaderValueN + "\n"
```

将 Headers 签名中 Header 的 Key 使用英文逗号分割放到 Request 的 Header 中，Key为：X-Ca-Signature-Headers。

Url

Url 指 Path + Query + Body 中 Form 参数，组织方法：对 Query+Form 参数按照字典对 Key 进行排序后按照如下方法拼接，如果 Query 或 Form 参数为空，则 Url = Path，不需要添加？，如果某个参数的 Value 为空只保留 Key 参与签名，等号不需要再加入签名。

```
String url =
Path +
```

```
"?" +
Key1 + "=" + Value1 +
"&" + Key2 + "=" + Value2 +
...
"&" + KeyN + "=" + ValueN
```

注意这里 Query 或 Form 参数的 Value 可能有多个，多个的时候只取第一个 Value 参与签名计算。

第三步.使用Secret计算签名

```
Mac hmacSha256 = Mac.getInstance("HmacSHA256");
byte[] keyBytes = secret.getBytes("UTF-8");
hmacSha256.init(new SecretKeySpec(keyBytes, 0, keyBytes.length, "HmacSHA256"));
String sign = new String(Base64.encodeBase64(Sha256.doFinal(stringToSign.getBytes("UTF-8")),"UTF-8"));
```

Secret 为 APP 的密钥，请在应用管理中获取。

API名称: pg_type_1583830351893

描述

数据申请自动生成

基本信息

HTTP协议：http

调用地址：/dgs/api_686979482438860800124104

方法：POST

请求入参说明

参数名称	位置	参数类型	是否必填	描述	默认值

请求Body描述(非Form表单数据)

输入参数示例:

```
{
"filter":{
"and":{
"typcategory":{
"eq":"1"
},
"typstorage":{
"eq":"1"
},
"typowner":{
"eq":"1"
},
"oid":{
"eq":"1"
},
"cmax":{
"eq":"1"
},
"typacl":{
"eq":"1"
},
```

```
"typmodin":{
"eq":"1"
},
"xmax":{
"eq":"1"
},
"typcollation":{
"eq":"1"
},
"typndims":{
"eq":"1"
},
"typtypmod":{
"eq":"1"
},
"typalign":{
"eq":"1"
},
"typreceive":{
"eq":"1"
},
"tableoid":{
"eq":"1"
},
"cmin":{
"eq":"1"
},
"typdefault":{
"eq":"1"
},
"ctid":{
"eq":"1"
},
"typmodout":{
"eq":"1"
},
"xmin":{
"eq":"1"
```

```
},
"typispreferred":{
"eq":"1"
},
"typlen":{
"eq":"1"
},
"typdefaultbin":{
"eq":"1"
},
"typrelid":{
"eq":"1"
},
"typbyval":{
"eq":"1"
},
"typarray":{
"eq":"1"
},
"typtype":{
"eq":"1"
},
"typinput":{
"eq":"1"
},
"typbasetype":{
"eq":"1"
},
"typnamespace":{
"eq":"1"
},
"typdelim":{
"eq":"1"
},
"typoutput":{
"eq":"1"
},
"typname":{
```

"eq":"1"
},
"typisdefined":{
"eq":"1"
},
"typelem":{
"eq":"1"
},
"typsend":{
"eq":"1"
},
"typnotnull":{
"eq":"1"
},
"typanalyze":{
"eq":"1"
}
}
},
"pageSize":2,
"sort":"+typarray",
"pageNum":1
}
输入参数说明:
1. 目前支持的逻辑操作符:
逻辑与:and
逻辑或:or
2. 目前支持的比较方式:
等于(=)：eq
不等于(<>):ne
大于(>):gt
小于(<):lt
大于等于(>=):gte
小于等于(<=):lte
3. 支持排序方式：
升序:+
降序:-
4. 支持分页方式:

pageSize:每页的大小

pageNum:显示页的序号，从1开始

5. 支持的查询条件有:

typarray typelem typacl typsend ctid typdefaultbin typlen typndims typbasetype typnotnull typmodout typdelim typname typnamespace typcollation typispreferred typtypmod typowner typoutput typisdefined typmodin typalign typanalyze cmax typreceive xmin typtype typcategory cmin typrelid oid tableoid typinput xmax typbyval typdefault typstorage

返回结果说明

返回参数类型

JSON(application/json)

返回结果示例


```
{
"data":{
"data":[
{
"typcategory":"1",
"typstorage":"1",
"typowner":"1",
"oid":"1",
"cmax":"1",
"typacl":"1",
"typmodin":"1",
"xmax":"1",
"typcollation":"1",
"typndims":"1",
"typtypmod":"1",
"typalign":"1",
"typreceive":"1",
"tableoid":"1",
"cmin":"1",
"typdefault":"1",
"ctid":"1",
"typmodout":"1",
"xmin":"1",
"typispreferred":"1",
```

"typlen":"1",
"typdefaultbin":"1",
"typrelid":"1",
"typbyval":"1",
"typarray":"1",
"typtype":"1",
"typinput":"1",
"typbasetype":"1",
"typnamespace":"1",
"typdelim":"1",
"typoutput":"1",
"typname":"1",
"typisdefined":"1",
"typelem":"1",
"typsend":"1",
"typnotnull":"1",
"typanalyze":"1"
},
{
"typcategory":"2",
"typstorage":"2",
"typowner":"2",
"oid":"2",
"cmax":"2",
"typacl":"2",
"typmodin":"2",
"xmax":"2",
"typcollation":"2",
"typndims":"2",
"typtypmod":"2",
"typalign":"2",
"typreceive":"2",
"tableoid":"2",
"cmin":"2",
"typdefault":"2",
"ctid":"2",
"typmodout":"2",
"xmin":"2",

```
"typispreferred":"2",
"typlen":"2",
"typdefaultbin":"2",
"typrelid":"2",
"typbyval":"2",
"typarray":"2",
"typtype":"2",
"typinput":"2",
"typbasetype":"2",
"typnamespace":"2",
"typdelim":"2",
"typoutput":"2",
"typname":"2",
"typisdefined":"2",
"typelem":"2",
"typsend":"2",
"typnotnull":"2",
"typanalyze":"2"
}
],
"pageSize":20,
"pageNum":1
},
"message":"操作成功",
"statusCode":200
}
```


返回失败示例


```
{
"message":"数据异常，请稍后重试或联系管理员",
"statusCode":603
}
```


错误码

错误码	错误信息	描述

公共异常

如何获取公共错误

所有的 API 请求只要到达了网关，网关就会返回请求结果信息。用户需要查看返回结果的头部，即 Header 部分。返回参数如示例:

```
//请求唯一ID，请求一旦进入API网关应用后，API网关就会生成请求ID并通过响应头返回给客户端，建议客户端与后端服务都记录此请求ID，可用于问题排查与跟踪
X-Ca-Request-Id: 7AD052CB-EE8B-4DFD-BBAF-EFB340E0A5AF

//API网关返回的错误消息，当请求出现错误时API网关会通过响应头将错误消息返回给客户端
X-Ca-Error-Message: Invalid Url

//当打开Debug模式后会返回Debug信息，此信息后期可能会有变更，仅用做联调阶段参考
X-Ca-Debug-Info: {"ServiceLatency":0,"TotalLatency":2}
```

在 Header 中获得 X-Ca-Error-Message 可以基本明确报错原因，而 X-Ca-Request-Id 可以用于提供给这边的支持人员，供支持人员搜索日志。

公共错误码

客户端错误码

错误代码	Http 状态码	状态语义
THROTTLED USER FLOW CONTROL	403	因用户流控被限制,调用频率过高导致被流控,可以联系 API 服务商协商放宽限制
THROTTLED APP FLOW CONTROL	403	因APP流控被限制,调用频率过高导致被流控,可以联系 API 服务商协商放宽限制
THROTTLED API FLOW CONTROL	403	因API流控被限制,调用频率过高导致被流控,可以联系API 服务商协商放宽限制.
THROTTLED GROUP FLOW CONTROL	403	因分组流控被限制,调用频率过高导致被流控,可以联系 API 服务商协商放宽限制.
EMPTY REQUEST BODY	400	body为空,请检查请求Body内容
INVALID REQUEST BODY	400	body无效,请检查请求 Body
INVALID PARAM LOCATION	400	参数位置错误,请求参数位置错误
UNSUPPORTED MULTIPART	400	不支持上传文件
APPKEY NOT EXIST	400	请求标头或请求参数上下文中不存在appKey
APPKEY INVALID	400	appKey无效
APPSECRETKEY INVALID	400	secretKey无效
UNAUTHORIZED	403	未被授权,APP未获得要调用的API的授权
API NOT EXIST	400	找不到API, 传入的API地址或者HttpMethod不正确,或已下线
REQUEST CONTEXT NEQ DEFINITION	400	API请求和定义不符

错误代码	Http 状态码	状态语义
REQUEST METHOD NOT ALLOWED	400	请求方法不允许
SIGNATURE INVALID	400	签名无效
SIGNATURE IS EMPTY	400	签名为空
INVALID CONTENT MD5	400	Content-MD5值不合法,请求 Body为空,但传入了MD5值,或MD5 值计算错误
BAD REQUEST	400	错误的请求
INVALID TIMESTAMP	400	时间戳不合法

服务器端错误码

以下为API服务端错误，如果频繁错误，可联系服务商。

错误代码	Http 状态码	错误描述	如何解决?
Internal Error	500	内部错误	建议重试,或者联系服务商
Failed To Invoke Backend Service	500	底层服务错误	API 提供者底层服务错误，建议重试，如果重试多次仍然不可用，可联系 API 服务商解决
Service Unavailable	503	服务不可用	建议重试,或者联系服务商
Async Service	504	后端服务超时	建议重试,或者联系服务商