# 认证签名

每个请求接口都需要生成一个认证签名（可复用），调用生成签名脚本（接口使用文档里附有各个语言版本的脚本代码）。

**def** sign(ak, sk, http\_method, path, params, headers, timestamp=**None**):

.....

sign\_res=sign(....)

传参说明：

ak,sk:控制台创建应用时生成的

http\_method：**"GET"**, **"POST"**,**"DELETE"**,**"PUT" (均为大写)**

Path:即为func\_api接口的名字或者说请求路径，务必**"/"** 开头

Param：为dict类型，不同的接口传参不同，详见接口文档

Headers：不写或填headers = {**"host"**: http\_host,**"Content-Type"**:**"application/json"**}

Timestamp：**None**

生成的认证字符串sign\_res放在Query或者Header里面，如果两个都存在，优先使用Query中的，即：authorization=认证字符串

签名脚本：

[https://iotdoc.horizon.ai/busiopenapi/part0\_tutorial/auth.html#part0\_auth\_1](https://iotdoc.horizon.ai/busiopenapi/part0_tutorial/auth.html" \l "part0_auth_1)

# 地平线接口测试

## #创建人脸库 create\_facesets

功能：创建用以储存用户信息的人脸库，创建成功返回得到的人脸库ID

Input: **name，description，extra**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| faceset\_id | string | 是 | 人脸库唯一id |
| name | string | 是 | 人脸库名 |
| extra | string | 否 | 人脸库额外信息 |
| description | string | 否 | 人脸库描述 |
| created\_time | string | 是 | 创建时间，ISO-8601标准格式 |
| updated\_time | string | 是 | 更新时间，ISO-8601标准格式 |

测试输入：

create\_facesets(**"smart4s"**, **"smart4s-customer"**, **"customer"**)

测试输出：

{'age': 'success', 'code': 0, 'request\_id': '474a4731-efd5-4fed-9ce2-5e0a49c3397f', 'data': {'faceset\_id': '5d5fa209ba18560008ba77d6', 'name': 'face\_ciger1', 'extra': 'no', 'description': 'faceset of cigers', 'created\_time': '2019-08-23T16:21:29+08:00', 'updated\_time': '2019-08-23T16:21:29+08:00'}}

## #查询人脸库列表 faceset\_list

功能：可以查询该签名所有的人脸库列表

Input: **None**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| faceset\_id | string | 是 | 人脸库唯一id |
| name | string | 是 | 人脸库名 |
| extra | string | 否 | 人脸库额外信息 |
| description | string | 否 | 人脸库描述 |
| created\_time | string | 是 | 创建时间，ISO-8601标准格式 |
| updated\_time | string | 是 | 更新时间，ISO-8601标准格式 |

测试输入：

faceset\_list()

测试输出：

[{'faceset\_id': '5d5fa209ba18560008ba77d6', 'name': 'face\_ciger1', 'extra': 'no', 'description': 'faceset of cigers', 'created\_time': '2019-08-23T16:21:29+08:00', 'updated\_time': '2019-08-23T16:21:29+08:00'}, {'faceset\_id': '5d5e078fba18560008ba77ba', 'name': 'autoreg\_a55dfe635d4bf67a2f1b487d\_NHcXY2YV', 'extra': '', 'description': '', 'created\_time': '2019-08-22T11:10:07+08:00', 'updated\_time': '2019-08-22T11:10:07+08:00'}, {'faceset\_id': '5d5e078eba18560008ba77b9', 'name': 'a55dfe635d4bf67a2f1b487d\_NHcXY2YV\_whitelist', 'extra': '', 'description': '', 'created\_time': '2019-08-22T11:10:06+08:00', 'updated\_time': '2019-08-22T11:10:06+08:00'}, {'faceset\_id': '5d5e078dba18560008ba77b8', 'name': 'a55dfe635d4bf67a2f1b487d\_NHcXY2YV\_blacklist', 'extra': '', 'description': '', 'created\_time': '2019-08-22T11:10:05+08:00', 'updated\_time': '2019-08-22T11:10:05+08:00'}, {'faceset\_id': '5d5e078cba18560008ba77b7', 'name': 'a55dfe635d4bf67a2f1b487d\_NHcXY2YV\_vip', 'extra': '', 'description': '', 'created\_time': '2019-08-22T11:10:04+08:00', 'updated\_time': '2019-08-22T11:10:04+08:00'}, {'faceset\_id': '5d5ce700ba18560008ba7789', 'name': 'face\_ciger', 'extra': 'update', 'description': 'just a test', 'created\_time': '2019-08-21T14:38:56+08:00', 'updated\_time': '2019-08-21T15:24:01+08:00'}]

## #删除人脸库 del\_facesets

功能：删除指定的人脸库，返回状态码

Input: **faceset\_id**

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| request\_id | string | 是 | 用于区分每一次请求的唯一的字符串。除非发生404(API\_NOT\_FOUND)错误，否则此字段必定返回 |
| code | int | 是 | 当前request返回状态码，0表示成功，其他详见错误码 |
| message | string | 是 | 当前request返回描述信息 |
| data | object | 是 | 当前request返回的数据 |

测试输入：

del\_facesets(**"5d5fa209ba18560008ba77d6"**)

测试输出：

{'message': 'success', 'code': 0, 'request\_id': '4266da3c-f6f4-428e-8fd3-433e2ef3e921', 'data': None}

## #更新人脸库update\_facesets

功能：更新人脸库的信息，如人脸库名，额外的信息，以及该人脸库的描述，返回注册的信息，以及创建的时间，更新的时间

Input: **faceset\_id**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| faceset\_id | string | 是 | 人脸库唯一id |
| name | string | 是 | 人脸库名 |
| extra | string | 否 | 人脸库额外信息 |
| description | string | 否 | 人脸库描述 |
| created\_time | string | 是 | 创建时间，ISO-8601标准格式 |
| updated\_time | string | 是 | 更新时间，ISO-8601标准格式 |

测试输入：

update\_facesets(**"5d6db70c4e7f1d0008910d5a"**,**"cig\_test\_update"**, **"cig test update faceset"**, **"test"**)

测试输出：

{'faceset\_id': '5d6db70c4e7f1d0008910d5a', 'name': 'cig\_test\_update', 'extra': 'test', 'faceset\_type': 'other', 'description': 'cig test update faceset', 'created\_time': '2019-09-03T08:42:52+08:00', 'updated\_time': '2019-09-03T08:45:19+08:00'}

## #查询人脸库 info\_faceset

功能：根据人脸库id查询该人脸库的库名，描述等信息，返回人脸库的信息

Input:**faceset\_id**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| faceset\_id | string | 是 | 人脸库唯一id |
| name | string | 是 | 人脸库名 |
| extra | string | 否 | 人脸库额外信息 |
| description | string | 否 | 人脸库描述 |
| created\_time | string | 是 | 创建时间，ISO-8601标准格式 |
| updated\_time | string | 是 | 更新时间，ISO-8601标准格式 |

测试输入：

info\_faceset(**"5d65d05cba18560008ba7859"**)

测试输出：

{'faceset\_id': '5d65d05cba18560008ba7859', 'name': 'smart4s', 'extra': 'customer', 'faceset\_type': 'other', 'description': 'smart4s-customer', 'created\_time': '2019-08-28T08:52:44+08:00', 'updated\_time': '2019-08-28T08:52:44+08:00'}

## #人脸库注册添加用户 registered\_user

功能：给指定的人脸库注册用户，需要提供注册对象的照片，及其年龄性别，返回用户的注册信息

Input:**faceset\_id,image\_type,img\_path,age,gender**

**用户照片有两种格式 ：0为线上url，1为本地照片提取的base64特征信息，及其用户属性设置如下：**

body={**"images"**: [{  
 **"image\_type"**: image\_type,  
 **"image\_base64"**: base64\_img,  
},],  
**"attributes"**: {  
 **"age"**: age,  
 **"gender"**: gender  
}}

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| face\_id | string | 是 | 用户唯一id |
| images | array | 是 | 用户人脸信息列表，见**[images说明](https://iotdoc.horizon.ai/busiopenapi/part4_api_basic/faceset_face.html" \l "images2)** |
| attributes | object | 是 | 用户属性,见**[attributes说明](https://iotdoc.horizon.ai/busiopenapi/part4_api_basic/faceset_face.html" \l "attributes2)** |
| created\_time | string | 是 | 创建时间，ISO-8601标准格式 |
| updated\_time | string | 是 | 更新时间，ISO-8601标准格式 |

测试输入：

registered\_user(**"5d5ce700ba18560008ba7789"**, 1,**"/home/hensel/Desktop/horizon/测试接口/2.jpeg"**,**"31"**, **"male"**)

测试输出：

{"message":"success","code":0,"request\_id":"21a9be8d-4314-485f-9ad9-9a997e156704","data":{"faceset\_id":"5d5ce700ba18560008ba7789","face\_id":"5d5fa90f9c81530008b17d3c","images":[{"url":"https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/Pictureserver/a55dfe635d4bf67a2f1b487d/2019823/a55dfe635d4bf67a2f1b487d\_Hib3qqePsr5G-1566550287588529460?X-Amz-Algorithm=AWS4-HMAC-SHA256\u0026X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190823%2Fcn-zhangjiakou%2Fs3%2Faws4\_request\u0026X-Amz-Date=20190823T085127Z\u0026X-Amz-Expires=604740\u0026X-Amz-SignedHeaders=host\u0026X-Amz-Signature=7b6b29b75b2d41d68ce095c77ff4fbe6e3d6322dae86bbdb119c5d20658b2a2d","expired\_time":"2019-08-30T16:40:27+08:00"}],"attributes":{"age":4,"gender":1},"created\_time":"2019-08-23T16:51:27+08:00","updated\_time":"2019-08-23T16:51:27+08:00"}}

## #获取人脸库的用户列表 user\_list

功能：查询人脸库的所有的用户信息，可以指定返回的当前页数，及每页信息条数

Input:**faceset\_id**

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| faceset\_id | string | 是 | 人脸库唯一id |
| face\_id | string | 是 | 用户唯一id |
| images | array | 是 | 用户人脸信息列表，见**[images说明](https://iotdoc.horizon.ai/busiopenapi/part4_api_basic/faceset_face.html" \l "images40)** |
| attributes | array | 是 | 用户属性,见**[attributes说明](https://iotdoc.horizon.ai/busiopenapi/part4_api_basic/faceset_face.html" \l "attributes40)** |
| created\_time | string | 是 | 创建时间，ISO-8601标准格式 |
| updated\_time | string | 是 | 更新时间，ISO-8601标准格式 |

pagination说明

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| current | int | 是 | 当前页数 |
| total | int | 是 | 总条目数（即用户总数） |
| per\_page | int | 是 | 每一页显示条数 |

images说明

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| url | string | 是 | 人脸url地址 |
| expired\_time | string | 是 | 人脸url地址过期时间 |

attributes说明

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| type | string | 是 | attribute的名称，包含系统识别的属性和用户自己添加的属性 |
| value | string | 是 | attribute的值 |
| confidence | float | 否 | 置信度 |
| is\_manual | int | 否 | 是否是手动设置的属性 |

测试输入：

user\_list(**"5d65d05cba18560008ba7859"**)

测试输出：

[{'faceset\_id': '5d65d05cba18560008ba7859', 'face\_id': '5d65d6b6ba18560008ba7885', 'images': [{'url': 'https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/Pictureserver/a55dfe635d4bf67a2f1b487d/2019828/a55dfe635d4bf67a2f1b487d\_3GN4rSANg3Mn-1566955190063549910?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T011224Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=7f92f76d59a84b258d6add617915a77805d34b7b45572f252357e4de12cbf0d5', 'expired\_time': '2019-09-10T09:01:24+08:00'}], 'attributes': [{'type': 'gender', 'value': '1', 'confidence': 0, 'is\_manual': True}, {'type': 'age', 'value': '0', 'confidence': 0, 'is\_manual': True}], 'created\_time': '2019-08-28T09:19:50+08:00', 'updated\_time': '2019-08-28T09:19:50+08:00'}, {'faceset\_id': '5d65d05cba18560008ba7859', 'face\_id': '5d666a8a9c81530008b17e4e', 'images': [{'url': 'https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/Pictureserver/a55dfe635d4bf67a2f1b487d/2019828/a55dfe635d4bf67a2f1b487d\_N5EBZMzX9HuZ-1566993034720143481?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T011224Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=083fac24919ac399c83751021cf8c69e135542773c44c451147c37db4a23a6d2', 'expired\_time': '2019-09-10T09:01:24+08:00'}], 'attributes': [{'type': 'age', 'value': '0', 'confidence': 0, 'is\_manual': True}, {'type': 'gender', 'value': '1', 'confidence': 0, 'is\_manual': True}], 'created\_time': '2019-08-28T19:50:34+08:00', 'updated\_time': '2019-08-28T19:50:34+08:00'}]

## #删除人脸库的用户 del\_user

功能：删除指定的人脸库的用户信息，返回状态码

Input:**faceset\_id,face\_id**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| request\_id | string | 是 | 用于区分每一次请求的唯一的字符串。除非发生404(API\_NOT\_FOUND)错误，否则此字段必定返回 |
| code | int | 是 | 当前request返回状态码，0表示成功，其他详见错误码 |
| message | string | 是 | 当前request返回描述信息 |
| data | object | 是 | 返回 null |

测试输入：

del\_user(**"5d5e078cba18560008ba77b7"**,**"5d5f671aba18560008ba77d2"**)

测试输出：

{'message': 'success', 'code': 0, 'request\_id': '03e00f4f-8a85-4553-a756-3d3787a928de', 'data': None}

## #更新人脸库的用户信息update\_user

功能：更新指定的人脸库里的指定用户的年龄性别以及照片信息，返回状态码

Input:**faceset\_id,face\_id**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| request\_id | string | 是 | 用于区分每一次请求的唯一的字符串。除非发生404(API\_NOT\_FOUND)错误，否则此字段必定返回 |
| code | int | 是 | 当前request返回状态码，0表示成功，其他详见错误码 |
| message | string | 是 | 当前request返回描述信息 |
| data | object | 是 | 返回 null |

测试输入：

update\_user(**"5d5ce700ba18560008ba7789"**,**"5d5dfded9c81530008b17d0f"**)

测试输出:

{'message': 'success', 'code': 0, 'request\_id': 'bb621f05-cd32-4ccf-a32e-7be243d35172', 'data': None

## #获取人脸库的用户信息user\_info

功能：查看指定人脸库的指定用户的性别年龄以及照片的信息，url为照片的下载地址

Input:**faceset\_id,face\_id**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| faceset\_id | string | 是 | 人脸库唯一id |
| face\_id | string | 是 | 用户唯一id |
| images | array | 是 | 用户人脸信息列表，见**[images说明](https://iotdoc.horizon.ai/busiopenapi/part4_api_basic/faceset_face.html" \l "image30)** |
| attributes | array | 是 | 用户属性,见**[attributes说明](https://iotdoc.horizon.ai/busiopenapi/part4_api_basic/faceset_face.html" \l "attributes30)** |
| created\_time | string | 是 | 创建时间，ISO-8601标准格式 |
| updated\_time | string | 是 | 更新时间，ISO-8601标准格式 |

images说明

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| url | string | 是 | 人脸url地址 |
| expired\_time | string | 是 | 人脸url地址过期时间，ISO-8601标准格式 |

attributes说明

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| type | string | 是 | attribute的名称，包含系统识别的属性和用户自己添加的属性 |
| value | string | 是 | attribute的值 |
| confidence | float | 否 | 置信度 |
| is\_manual | int | 否 | 是否是手动设置的属性 |

测试输入：

user\_info(**"5d5e078cba18560008ba77b7"**,**"5d5f67b39c81530008b17d35"**)

测试输出：

{'faceset\_id': '5d5e078cba18560008ba77b7', 'face\_id': '5d5f67b39c81530008b17d35', 'images': [{'url': 'https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/Pictureserver/a55dfe635d4bf67a2f1b487d/2019823/a55dfe635d4bf67a2f1b487d\_5dWKQKvsScyr-1566533555204762049?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T012255Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=be73f7209032c450bb3c5b52b86c39ac5f06b18e1c320f24ea3fc3e0e99d7403', 'expired\_time': '2019-09-10T09:11:55+08:00'}, {'url': 'https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/Pictureserver/a55dfe635d4bf67a2f1b487d/2019824/a55dfe635d4bf67a2f1b487d\_ekg6ECSSdCwo-1566612245806292821?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T012255Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=27725ac4da7901d1dbeb53c0d40abf5ce4056de98d1ed70656adc2c43a298cef', 'expired\_time': '2019-09-10T09:11:55+08:00'}], 'attributes': [{'type': 'age', 'value': '4', 'confidence': 0, 'is\_manual': True}, {'type': 'gender', 'value': '1', 'confidence': 0, 'is\_manual': True}], 'created\_time': '2019-08-23T12:12:35+08:00', 'updated\_time': '2019-08-24T10:04:05+08:00'}

## #人脸更新:向人脸库中指定用户添加人脸特征图片user\_append\_img

功能：指定人脸库的指定用户追加人脸照片信息，返回状态码

Input:**faceset\_id,face\_id**

**用户照片有两种格式 ：0为线上url，1为本地照片提取的base64特征信息**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| request\_id | string | 是 | 用于区分每一次请求的唯一的字符串。除非发生404(API\_NOT\_FOUND)错误，否则此字段必定返回 |
| code | int | 是 | 当前request返回状态码，0表示成功，其他详见错误码 |
| message | string | 是 | 当前request返回描述信息 |
| data | object | 是 | null |

测试输入：

user\_append\_img(**"5d5e078cba18560008ba77b7"**,**'5d5f67b39c81530008b17d35'**,1,**"/home/hensel/Desktop/horizon/测试接口/2.jpg"** ,**"18"**, **"male"**)

测试输出：

{'message': 'success', 'code': 0, 'request\_id': 'f43c8c54-4c80-4f01-bb1d-c276f7fd08ef', 'data': None}

## #人脸检测detect\_face

功能：检测传入的照片的人脸，image\_type为1时则为本地照片，0为线上url，可以设置一张照片中检测的人脸最大数

Input:**image\_type,img\_path**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| face\_info | object | 是 | 返回检测到的人脸信息，按面积从大到小排序, 详见[face\_info](https://iotdoc.horizon.ai/busiopenapi/part4_api_basic/face.html" \l "faceInfo) |
| distance\_threshold | object | 是 | 不同误报率下的距离阈值, 详见[distance\_threshold](https://iotdoc.horizon.ai/busiopenapi/part4_api_basic/face.html" \l "distance_thresholds),此值与1:1或1:n检索接口返回的distance值进行比较，选择一个档位作为可接受的识别阈值 |

distance\_threshold

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| 1e-3 | float | 是 | float数，千分之一误报率 |
| 1e-4 | float | 是 | float数，万分之一误报率 |
| 1e-5 | float | 是 | float数，十万分之一误报率 |
| 1e-6 | float | 是 | float数，百万分之一误报率 |

face\_info

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| face\_rect | object | 是 | 人脸框位置,详见[人脸框](https://iotdoc.horizon.ai/busiopenapi/part4_api_basic/face.html" \l "facerect) |
| keypoint | string | 否 | 关键点。5个关键点，共10个float,逗号拼接成字符串返回; 请求中带有keypoint参数返回此字段 |
| quality | float | 否 | 图片质量分数; 请求中带有quality参数返回 |
| pose | string | 否 | 姿态角度，返回值是3个float值用逗号隔开的字符串，依次对应俯仰、偏航、旋转; 请求中attribute包含pose参数返回此字段 |
| age | int | 否 | 年龄分段，具体的年龄分段表示：****0:\"0-6\", 1:\"6-12\", 2:\"12-18\", 3:\"18-28\", 4:\"28-35\", 5:\"35-45\", 6:\"45-55\", 7:\"55-\"****; 请求中attribute包含age参数返回此字段 |
| age\_score | float | 否 | 年龄分数，置信度; 请求中attribute包含age参数返回此字段 |
| gender | string | 否 | 男性：male， 女性：female; 请求中attribute包含gender参数返回此字段 |
| gender\_score | float | 否 | 性别分数，置信度; 请求中attribute包含gender参数返回此字段，越接近0是女性，接近1是男性 |
| feature | string | 否 | 人脸特征值，请求中attribute包含feature参数返回此字段 |

人脸框

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| left | int | 是 | 人脸框左上角横坐标 |
| top | int | 是 | 人脸框左上角纵坐标 |
| right | int | 是 | 人脸框右下角横坐标 |
| bottom | int | 是 | 人脸框右下角纵坐标 |

测试输入：

detect\_face(1,**"/home/hensel/Desktop/horizon/测试接口/7.png"**)

测试输出：

{'face\_info': [{'face\_rect': {'left': 97, 'top': 107, 'right': 340, 'bottom': 383}, 'keypoint': '176.208527,189.587708,1,280.507843,238.541214,0,188.699646,255.320389,1,125.497856,289.847321,1,212.815155,332.829407,1', 'quality': 0.4, 'age': 3, 'age\_score': 0.8755357, 'gender': 'male', 'gender\_score': 0.8206242}, {'face\_rect': {'left': 273, 'top': 322, 'right': 403, 'bottom': 484}, 'keypoint': '310.770844,395.073730,1,371.292755,398.230225,1,343.269806,426.333191,1,319.287384,462.514465,1,363.809418,462.467377,1', 'quality': 0.713845, 'age': 5, 'age\_score': 0.9847288, 'gender': 'male', 'gender\_score': 0.9999279}], 'distance\_threshold': {'1e-3': 1.5037, '1e-4': 1.3884, '1e-5': 1.2786, '1e-6': 1.162}}

## #人脸抠图extract\_face:

功能：与人脸检测相似，返回的结果多一个抠图框的bbox坐标，抠图框出来的人脸相比于人脸框能看到整个脑袋

Input:**img\_path**

Response:

faceExtract

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| image\_base64 | string | 提取的人脸的base64值 |
| face\_rect | object | 人脸框位置,详见[人脸框](https://iotdoc.horizon.ai/busiopenapi/part4_api_basic/face.html" \l "facerect2) |
| crop\_rect | object | 抠图框位置,详见[抠图框](https://iotdoc.horizon.ai/busiopenapi/part4_api_basic/face.html" \l "faceret23) |

人脸框

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| left | int | 是 | 人脸框左上角横坐标 |
| top | int | 是 | 人脸框左上角纵坐标坐标 |
| right | int | 是 | 人脸框右下角横坐标 |
| bottom | int | 是 | 人脸框右下角纵坐标 |

抠图框

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| left | int | 是 | 人脸框左上角横坐标 |
| top | int | 是 | 人脸框左上角纵坐标坐标 |
| right | int | 是 | 人脸框右下角横坐标 |
| bottom | int | 是 | 人脸框右下角纵坐标 |

测试输入：

extract\_face(1,**"/home/hensel/Desktop/horizon/测试接口/7.png"**)

测试输出：

[{'image\_base64': '/9........qT7eZ//2Q==', 'face\_rect': {'left': 97, 'top': 107, 'right': 340, 'bottom': 383}, 'crop\_rect': {'left': 0, 'top': 24, 'right': 439, 'bottom': 463}}]

## #人脸比对 1：1 match\_face:

功能：两个人脸照片的比对，返回相似度，特征距离

Input:**img1\_path,img2\_path**

**(这里使用"/home/hensel/Desktop/1.jpeg"**,**"/home/hensel/Desktop/2.jpeg")**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| similarity | float | 是 | (0,1]的float数，越接近1表示相似度越大 |
| distance | float | 是 | 两个图片的特征距离 |
| distance\_threshold | object | 是 | 不同误报率下的距离阈值, 详见[distance\_threshold](https://iotdoc.horizon.ai/busiopenapi/part4_api_basic/face.html" \l "distance_thresholds)，此值与1:1或1:n检索接口返回的distance值进行比较，选择一个档位作为可接受的识别阈值 |

distance\_threshold

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| 1e-3 | float | 是 | float数，千分之一误报率 |
| 1e-4 | float | 是 | float数，万分之一误报率 |
| 1e-5 | float | 是 | float数，十万分之一误报率 |
| 1e-6 | float | 是 | float数，百万分之一误报率 |

测试输入：

match\_face(**"/home/hensel/Desktop/1.jpeg"**,**"/home/hensel/Desktop/2.jpeg"**)

body ={  
**"image\_type"**:1,  
**"image\_base64\_1"**:base64\_img\_1,  
**"face\_rect\_1"**:**"107,45,183,136"**,  
**"image\_base64\_2"**:base64\_img\_2,  
**"face\_rect\_2"**:**"145,81,312,289"**}

测试输出：

{'message': 'success', 'code': 0, 'request\_id': '5a36d921-13d3-474a-87af-df7678cb8970', 'data': {'similarity': 0.7629701, 'distance': 1.0878844, 'distance\_threshold': {'1e-3': 1.5037, '1e-4': 1.3884, '1e-5': 1.2786, '1e-6': 1.162}}}

## #人脸比对 1：n 可以多个人脸库比对search\_face:

功能：一张人脸照片与指定的一个或多个人脸库里的用户比对

Input:**img\_path,faceset\_ids（**人脸库id列表**）**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| items | array | 是 | 搜索结果对象数组，根据相似度从高到低排列，具体单个结果对象，详见[匹配对象](https://iotdoc.horizon.ai/busiopenapi/part4_api_basic/face.html" \l "result2) |
| distance\_threshold | object | 是 | 不同误报率下的距离阈值, 详见[distance\_threshold](https://iotdoc.horizon.ai/busiopenapi/part4_api_basic/face.html" \l "distance_thresholds)，此值与1:1或1:n检索接口返回的distance值进行比较，选择一个档位作为可接受的识别阈值 |

匹配对象

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| id | string | 是 | 人脸的用户id |
| images | array | 是 | 当前被匹配的人脸用户的图片info，具体的图片信息，详见[ImageInfo](https://iotdoc.horizon.ai/busiopenapi/part4_api_basic/face.html" \l "imageinfo) |
| similarity | float | 是 | (0,1]的float数，越接近1表示相似度越大 |
| distance | float | 是 | 相似距离 |
| faceset\_id | string | 是 | 该人脸所在的人脸库id |
| created\_time | string | 是 | 创建时间 |
| updated\_time | string | 是 | 更新时间 |
| active\_time | string | 是 | 活跃时间，每次检索被命中的id更新该时间 |
| attributes | array | 是 | 该人脸的属性列表，包括系统识别的属性和用户手动添加的属性，详见[属性对象](https://iotdoc.horizon.ai/busiopenapi/part4_api_basic/face.html" \l "attribute) |

属性对象

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| type | string | 是 | attribute的名称，包含系统识别的属性和用户自己添加的属性 |
| value | string | 是 | attribute的值 |
| confidence | float | 否 | 置信度 |
| is\_manual | int | 否 | 是否是手动设置的属性 |
| mtime | long | 否 | 属性修改时间 |

ImageInfo

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| url | string | 是 | 图片url |
| expired\_time | string | 是 | 图片过期时间 |

distance\_threshold

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| 1e-3 | float | 是 | float数，千分之一误报率 |
| 1e-4 | float | 是 | float数，万分之一误报率 |
| 1e-5 | float | 是 | float数，十万分之一误报率 |
| 1e-6 | float | 是 | float数，百万分之一误报率 |

测试输入：

search\_face(**"/home/hensel/Desktop/7.png"**,[**"5d5ce700ba18560008ba7789"**,**"5d5e078cba18560008ba77b7"**])

测试输出：

{'distance\_thresholds': {'1e-3': 1.5037, '1e-4': 1.3884, '1e-5': 1.2786, '1e-6': 1.162}, 'items': [{'id': '5d5dff36ba18560008ba77ae', 'faceset\_id': '5d5ce700ba18560008ba7789', 'attributes': [{'type': 'age\_group', 'value': '4', 'confidence': 0, 'is\_manual': True}, {'type': 'gender', 'value': '1', 'confidence': 0, 'is\_manual': True}], 'distance': 0.5784283876419067, 'similar': 0.8865013122558594, 'images': [{'url': 'https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/Pictureserver/a55dfe635d4bf67a2f1b487d/2019822/a55dfe635d4bf67a2f1b487d\_9Hfe6to65awM-1566441270073067693?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190824%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190824T024616Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=121748d4bbf47cc1ac8d2a6d64fb388dd922605eb96432a56d906b604c0e5335', 'expired\_time': '2019-08-31T10:35:16+08:00'}], 'created\_time': '2019-08-22T10:34:30+08:00', 'updated\_time': '2019-08-22T10:34:30+08:00', 'active\_time': '2019-08-22T10:34:30+08:00'}, {'id': '5d5dfea99c81530008b17d11', 'faceset\_id': '5d5ce700ba18560008ba7789', 'attributes': [{'type': 'age\_group', 'value': '4', 'confidence': 0, 'is\_manual': True}, {'type': 'gender', 'value': '1', 'confidence': 0, 'is\_manual': True}], 'distance': 0.5784283876419067, 'similar': 0.8865013122558594, 'images': [{'url': 'https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/Pictureserver/a55dfe635d4bf67a2f1b487d/2019822/a55dfe635d4bf67a2f1b487d\_QR5TnVqVYCdd-1566441129090945689?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190824%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190824T024616Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=517715af72b052b254f3a81695da33c7c7fc386b14d43dd8ff00aeb75995728b', 'expired\_time': '2019-08-31T10:35:16+08:00'}], 'created\_time': '2019-08-22T10:32:09+08:00', 'updated\_time': '2019-08-22T10:32:09+08:00', 'active\_time': '2019-08-22T10:32:09+08:00'}, {'id': '5d5dfded9c81530008b17d0f', 'faceset\_id': '5d5ce700ba18560008ba7789', 'attributes': [{'type': 'age\_group', 'value': '3', 'confidence': 0, 'is\_manual': True}, {'type': 'gender', 'value': '1', 'confidence': 0, 'is\_manual': True}], 'distance': 1.5362746715545654, 'similar': 0.5559953451156616, 'images': [{'url': 'https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/Pictureserver/a55dfe635d4bf67a2f1b487d/2019823/a55dfe635d4bf67a2f1b487d\_TABdzdfiSyVD-1566552198135303367?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190824%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190824T024616Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=6c496b066c118a23bf539bc889f6fb9dda33cb76e768942e3317b5e1ffb88c32', 'expired\_time': '2019-08-31T10:35:16+08:00'}], 'created\_time': '2019-08-22T10:29:01+08:00', 'updated\_time': '2019-08-23T17:23:18+08:00', 'active\_time': '2019-08-22T10:29:01+08:00'}, {'id': '5d5dfe759c81530008b17d10', 'faceset\_id': '5d5ce700ba18560008ba7789', 'attributes': [{'type': 'age\_group', 'value': '4', 'confidence': 0, 'is\_manual': True}, {'type': 'gender', 'value': '1', 'confidence': 0, 'is\_manual': True}], 'distance': 1.6957714557647705, 'similar': 0.4639780521392822, 'images': [{'url': 'https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/Pictureserver/a55dfe635d4bf67a2f1b487d/2019822/a55dfe635d4bf67a2f1b487d\_imyGlNH1LDYU-1566441077041754089?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190824%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190824T024616Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=2ddaf02052056693c93b966887584196503ffa1a2d25d6639ba877188795666e', 'expired\_time': '2019-08-31T10:35:16+08:00'}], 'created\_time': '2019-08-22T10:31:17+08:00', 'updated\_time': '2019-08-22T10:31:17+08:00', 'active\_time': '2019-08-22T10:31:17+08:00'}]}

## **#启用客流工具（默认自动创建 会员 白名单 黑名单）user\_face**

功能：启用该空间id的客流工具，设备只要开机且连接外网状态，默认8:00-23:00记录客流信息。返回状态码

Input:**space\_id**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| request\_id | string | 是 | 用于区分每一次请求的唯一的字符串。除非发生404（API\_NOT\_FOUND ) 或403 （AUTHORIZATION\_ERROR）错误，此字段必定返回。 |
| code | int | 是 | 当前request返回状态码，0表示成功，其他详见错误码 |
| message | string | 是 | 当前request返回描述信息 |
| data | object | 是 | 当前request返回的数据 |

测试输入：

user\_face(**"a55dfe635d4bf67a2f1b487d\_NHcXY2YV"**)

测试输出：

{'message': 'success', 'code': 0, 'request\_id': '7017a1df-83bc-4591-9cca-2ac828b6ad4f', 'data': None}

## #关联人脸库到设备空间 faceset\_space

功能：将人脸库指定要用的设备空间，这样设备空间下的ipc上来的数据就会去检索该人脸库，确定是否属于人脸库中的人员，返回状态码

Input:**space\_ids,faceset\_ids**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| request\_id | string | 是 | 用于区分每一次请求的唯一的字符串。除非发生404（API\_NOT\_FOUND ) 或403 （AUTHORIZATION\_ERROR）错误，此字段必定返回。 |
| code | int | 是 | 当前request返回状态码，0表示成功，其他详见错误码 |
| message | string | 是 | 当前request返回描述信息 |
| data | object | 是 | 当前request返回的数据 |

测试输入：

faceset\_space([**"a55dfe635d4bf67a2f1b487d\_NHcXY2YV"**],[**"5d5ce700ba18560008ba7789"**])

测试输出：

{'message': 'success', 'code': 0, 'request\_id': '472964ef-ae45-4428-946e-13f0972b0ed2', 'data': None}

## #总客流量查询 count\_person

功能：指定设备序列以及起止时间，可选指定返回的数据的每页数据条数，以及当前页，数据为倒序输出，即第一页为最新数据，默认返回第一页，每页返回20条数据。该数据为疑似人脸框穿过出入店线的记录，未做去重处理。

Input:**device\_sn,start\_time,end\_time**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| start\_time | string | 是 | 开始时间，ISO-8601时间 |
| end\_time | string | 是 | 结束时间，ISO-8601时间 |
| space\_id | string | 是 | 设备空间唯一ID |
| device\_sn | string | 是 | IPC设备的SN码 |
| total\_in | int | 是 | 总进店客流 |
| total\_out | int | 是 | 总出店客流 |
| cycle | string | 否 | 统计周期，值为hour或day |
| records | object array | 是 | 详细的统计结果 |

records中object说明

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| age | array | 是 | 年龄维度进出店数量相关统计，参见**[age说明](https://iotdoc.horizon.ai/busiopenapi/part5_api_analysis_tools/statistics.html" \l "age_object)** |
| gender | array | 是 | 性别维度进出店数量相关统计,参见**[gender说明](https://iotdoc.horizon.ai/busiopenapi/part5_api_analysis_tools/statistics.html" \l "gender_object)** |
| faceset | array | 是 | 不同人脸库进出店数量相关统计,参见**[faceset说明](https://iotdoc.horizon.ai/busiopenapi/part5_api_analysis_tools/statistics.html" \l "faceset_object)** |

如果cycle为hour，只能查询24小时之内的数据，records中数据按照小时从小到大排序，最多24项

如果cycle为day，只能查询31天之内的数据，records中数据按照天从小到大排序，最多31项

age中的object类型说明：

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| age\_group | int | 是 | 年龄段 |
| in\_number | int | 是 | 进店统计 |
| out\_number | int | 是 | 出店统计 |

年龄段划分

-1：未识别年龄客流统计

0：0-6岁的客流统计

1：7-12岁的客流统计

2：13-18岁的客流统计

3：19-28岁的客流统计

4：29-35岁的客流统计

5：36-45岁的客流统计

6：46-55岁的客流统计

7：56+岁的客流统计

gender中的object说明：

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| gender\_group | int | 是 | 性别 |
| in\_number | int | 是 | 进店统计 |
| out\_number | int | 是 | 出店统计 |

性别划分(gender\_group) -1：未知性别(unknown) 0 ：女性(female) 1 ：男性(male)

faceset数组中object说明：

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| faceset\_id | string | 是 | 该人脸库的唯一ID |
| in\_number | int | 是 | 该人脸库的进店客流量 |
| out\_number | int | 是 | 该人脸库的出店客流 |

测试输入：

count\_person(**"0691A001100112638L"**,**"2019-08-22T13:02:16+08:00"**,**"2019-08-22T14:00:16+08:00"**)

测试输出：

{'message': 'success', 'code': 0, 'request\_id': 'a2a152b4-8a8e-4190-860f-ca17c07b15b8', 'data': {'start\_time': '2019-08-22T13:02:16+08:00', 'end\_time': '2019-08-22T14:00:16+08:00', 'device\_sn': '0691A001100112638L', 'total\_in': 70, 'total\_out': 58, 'records': [{'age': [{'age\_group': -1, 'in\_number': 64, 'out\_number': 43}, {'age\_group': 0, 'in\_number': 0, 'out\_number': 0}, {'age\_group': 1, 'in\_number': 0, 'out\_number': 0}, {'age\_group': 2, 'in\_number': 0, 'out\_number': 0}, {'age\_group': 3, 'in\_number': 2, 'out\_number': 7}, {'age\_group': 4, 'in\_number': 3, 'out\_number': 5}, {'age\_group': 5, 'in\_number': 1, 'out\_number': 1}, {'age\_group': 6, 'in\_number': 0, 'out\_number': 2}, {'age\_group': 7, 'in\_number': 0, 'out\_number': 0}], 'gender': [{'gender\_group': -1, 'in\_number': 64, 'out\_number': 43}, {'gender\_group': 0, 'in\_number': 0, 'out\_number': 7}, {'gender\_group': 1, 'in\_number': 6, 'out\_number': 8}], 'faceset': [{'faceset\_id': 'unknown', 'in\_number': 66, 'out\_number': 51}, {'faceset\_id': '5d5e078fba18560008ba77ba', 'in\_number': 4, 'out\_number': 7}]}], 'space\_id': 'a55dfe635d4bf67a2f1b487d\_NHcXY2YV'}}

## #去重客流量查询person\_count\_distinct

功能：指定设备序列以及起止时间，可选指定返回的数据的每页数据条数，以及当前页，数据为倒序输出，即第一页为最新数据，默认返回第一页，每页返回20条数据。根据指定的时间和周期，查询客流量，查询粒度为设备，统计维度包括设备、人脸库、性别、年龄、进店、出店, 统计不同维度的进出店客流量人次，去除重复客流数据。可以设置是否过滤员工(为true时候不统计人脸库类型为whitelist的人员记录，false时统计人脸库类型为whitelist的人员记录)

Input:**space\_id,device\_sns,start\_time,end\_time**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| start\_time | string | 是 | 查询的开始时间 |
| end\_time | string | 是 | 查询的结束时间 |
| space\_id | string | 否 | 查询的设备空间id |
| device\_sns | string | 否 | 查询的设备sn列表，设备sn以逗号隔开，比如sn1,sn2,sn3 |
| cycle | string | 是 | 去重周期 |
| total\_in | number | 否 | 查询时间段内的去重后总进店人数 |
| total\_out | number | 是 | 查询时间段内的去重后总出店人数 |
| records | object | 是 | 详细去重统计结果 |

records中object说明

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| gender | object | 否 | 性别维度进出店数量相关统计,参见**[gender说明](https://iotdoc.horizon.ai/busiopenapi/part5_api_analysis_tools/statistics.html" \l "gender_object)** |
| age | object | 否 | 年龄维度进出店数量相关统计，参见**[age说明](https://iotdoc.horizon.ai/busiopenapi/part5_api_analysis_tools/statistics.html" \l "age_object)** |
| faceset | objects | 是 | 不同人脸库进出店数量相关统计,参见**[faceset说明](https://iotdoc.horizon.ai/busiopenapi/part5_api_analysis_tools/statistics.html" \l "faceset_object)** |
| cycle\_begin\_time | string | 是 | ISO-8601格式去重记录的统计开始时间 |

如果cycle为hour，只能查询24小时之内的数据，records中数据按照小时从小到大排序，最多24项

如果cycle为day，只能查询31天之内的数据，records中数据按照天从小到大排序，最多31项

测试输入：

person\_count\_distinct(**"a55dfe635d4bf67a2f1b487d\_NHcXY2YV"**, [], **"2019-09-02T14:42:16+08:00"**,**"2019-09-02T14:50:16+08:00"**)

测试输出：

{'start\_time': '2019-09-02T14:42:16+08:00', 'end\_time': '2019-09-02T14:50:16+08:00', 'device\_sns': '', 'total\_in': 0, 'total\_out': 1, 'cycle': '', 'space\_id': 'a55dfe635d4bf67a2f1b487d\_NHcXY2YV', 'records': [{'cycle\_begin\_time': '2019-09-02T14:42:16+08:00', 'age': [{'age': -1, 'in\_number': 0, 'out\_number': 0}, {'age': 0, 'in\_number': 0, 'out\_number': 0}, {'age': 1, 'in\_number': 0, 'out\_number': 0}, {'age': 2, 'in\_number': 0, 'out\_number': 0}, {'age': 3, 'in\_number': 0, 'out\_number': 1}, {'age': 4, 'in\_number': 0, 'out\_number': 0}, {'age': 5, 'in\_number': 0, 'out\_number': 0}, {'age': 6, 'in\_number': 0, 'out\_number': 0}, {'age': 7, 'in\_number': 0, 'out\_number': 0}], 'gender': [{'gender\_group': -1, 'in\_number': 0, 'out\_number': 0}, {'gender\_group': 0, 'in\_number': 0, 'out\_number': 0}, {'gender\_group': 1, 'in\_number': 0, 'out\_number': 1}], 'faceset': [{'faceset\_id': '5d5e078fba18560008ba77ba', 'in\_number': 0, 'out\_number': 1}]}]}

## #查询到店记录 visitors

功能：指定查询的设备系列号以及起止时间，识别到人脸后才有记录数据，返回的结果为识别到的人脸所属的人脸库及其id号等用户信息，如果为新客即会在人脸库默认库里面创建一个用户id，该默认库的用户face\_id格式均为autoreg\_设备序列号\_.....，返回的track\_id是到店记录的唯一id，在查询的时间段内的同一个人，出现的不同时间里的track\_id不一样，match\_url为用户的首次入库的url。

Input:**device\_sn,start\_time,end\_time**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| start\_time | string | 是 | 开始时间(ISO时间) |
| end\_time | string | 是 | 结束时间(ISO时间) |
| records | object array | 是 | 具体的到店记录 |

pagination说明

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| current | int | 否 | 默认第一页开始 |
| per\_page | int | 否 | 当前页的大小，默认20 |
| total | int | 否 | 到店记录的总数 |

records数组中的object字段说明：

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| space\_id | string | 是 | 设备空间ID |
| device\_sn | string | 是 | IPC设备的SN码(长度限制20以内) |
| face\_id | 否 | string | 访客ID，未识别到的结果以noreg\_为前缀 |
| faceset\_id | 否 | string | 人脸库唯一ID，未命中人脸库返回空字符串 |
| gender | 是 | int | [访客性别](https://iotdoc.horizon.ai/busiopenapi/part5_api_analysis_tools/statistics.html" \l "gender_object) |
| age | 是 | int | [年龄段划分](https://iotdoc.horizon.ai/busiopenapi/part5_api_analysis_tools/statistics.html" \l "age_object) |
| match\_url | 否 | object | 访客对应人脸库中的头像url，未注册访客无此参数 |
| capture\_url | 是 | object | 访客本次抓拍图url |
| track\_id | 是 | string | 人员在本次抓拍记录的追踪标识 |
| capture\_time | 是 | string | 到访时间(ISO时间) |

match\_url和capture\_url说明

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| url | string | 是 | 图片url |
| expired\_at | string | 是 | 图片过期时间，ISO-8601格式 |

测试输入：

visitors(**"0691A001100112638L"**,**"2019-08-23T10:00:16+08:00"**,**"2019-08-23T13:47:16+08:00"**)

测试输出：

{'start\_time': '2019-08-23T16:00:16+08:00', 'end\_time': '2019-08-23T16:05:16+08:00', 'records': [{'face\_id': '5d5f67b39c81530008b17d35', 'faceset\_id': '5d5e078cba18560008ba77b7', 'gender': 1, 'age': 3, 'match\_url': {'url': 'https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/Pictureserver/a55dfe635d4bf67a2f1b487d/2019823/a55dfe635d4bf67a2f1b487d\_5dWKQKvsScyr-1566533555204762049?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190824%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190824T025717Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=b0b5d6e0dc519501590549a2c7baf9048a1c46211557fd8ff4e4efe8e385b9ea', 'expired\_at': '2019-08-31T10:46:17+08:00'}, 'capture\_url': {'url': 'https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/0691A001100112638L\_20190823080230\_00000000025649752994\_000011197.jpg?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190824%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190824T025717Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=b7af5a9cc79382f692a4f2ffab87c5aca6fd9bb7f8be68904a3a63da9d34bb62', 'expired\_at': '2019-08-31T10:46:17+08:00'}, 'track\_id': '1566533686', 'capture\_time': '2019-08-23T16:02:25+08:00', 'device\_sn': '0691A001100112638L', 'space\_id': 'a55dfe635d4bf67a2f1b487d\_NHcXY2YV'}]}

## #到店记录订阅subscription

功能：指定设备空间或设备订阅，订阅后可以通过建立websocket长连接监听[到店记录的推送](https://iotdoc.horizon.ai/busiopenapi/part5_api_analysis_tools/statistics.html" \l "part5_6)，返回的subscription\_id取消订阅时候需要提供，指定的client\_id用户推送接口需要提供

Input:**topic\_name,topic\_id,client\_id**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| subscription\_id | string | 是 | 订阅的id |

测试输入：

subscription(**"device"**,**"0691A001100112638L"**,**"cig123"**)

测试输出：

{'message': 'success', 'code': 0, 'request\_id': '8e102833-3fd5-46fc-817d-6dd029447f77', 'data': {'subscription\_id': '8cef6b9f-c95c-11e9-907c-0a58c0a80e02'}}

## #到店记录的实时推送ws\_sample

功能：通过websocket实时推送到店记录，输入的client\_id订阅时候自定义的，返回的数据为启动服务后的所以能识别出人脸的数据。

Input:**client\_id**

Response:

| **名称** | **必选** | **类型** | **说明** |
| --- | --- | --- | --- |
| age | 是 | int | [访客年龄段](https://iotdoc.horizon.ai/busiopenapi/part5_api_analysis_tools/statistics.html" \l "age_object) |
| manual\_age | 否 | int | 静态库用户注册时候输入的年龄,没有的话返回-1 |
| app\_id | 是 | string | 应用ID |
| capture\_url | 是 | object | 访客本次抓拍图url |
| device\_sn | 是 | string | IPC设备的SN码 |
| event\_time | 是 | string | 到访时间 |
| face\_id | 是 | string | 访客ID，未命中到任何人，未识别到的结果以noreg\_为前缀 |
| faceset\_id | 是 | string | 访客所属人脸库ID，没有命中人脸库，返回空字符串 |
| gender | 是 | int | 访客性别，1是男性，0是女性，-1未知 |
| in\_out\_type | 是 | int | 0表示进店，1表示出店 |
| match\_url | 否 | object | 访客对应人脸库中的头像url |
| space\_id | 是 | string | 设备空间唯一ID |
| track\_id | 是 | string | 人员在本次抓拍事件的追踪标识 |

测试输入：

ws\_sample(**"cig123"**)

测试输出：

{"code":200,"msg":"WebSocket connected","cause":""}

{

"age" : 4,

"app\_id" : "a55dfe635d4bf67a2f1b487d",

"capture\_url" :

{

"expire" : "2019-09-10T06:23:54Z",

"url" : "https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/0691A001100112638L\_20190903063146\_00000000008844810742\_000001774.jpg?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T063454Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=8c5646b006e98fad5f3d506a87ff097cffef9f39a66486d7080cad044a8bcc30"

},

"device\_sn" : "0691A001100112638L",

"event\_time" : 1567492831121,

"face\_id" : "5d6e079a4e01730008a8b398",

"faceset\_id" : "5d5e078cba18560008ba77b7",

"faceset\_type" : "vip",

"gender" : 1,

"in\_out\_type" : 0,

"manual\_age" : 32,

"match\_url" :

{

"expire" : "2019-09-10T06:23:54Z",

"url" : "https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/Pictureserver/a55dfe635d4bf67a2f1b487d/201993/a55dfe635d4bf67a2f1b487d\_3dSuvDFEY5ye-1567491994128486938?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T063454Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=731b33573c777db8837797a0567d174caa08250e35673a96b59c305cd9ca40c4"

},

"msg\_type" : 2,

"score" : 0.91190749999999998,

"space\_id" : "a55dfe635d4bf67a2f1b487d\_NHcXY2YV",

"track\_id" : 1567486404

}

{

"age" : 4,

"app\_id" : "a55dfe635d4bf67a2f1b487d",

"capture\_url" :

{

"expire" : "2019-09-10T06:23:56Z",

"url" : "https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/0691A001100112638L\_20190903063146\_00000000008844810742\_000001774.jpg?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T063456Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=8ab82590ef4113ead1b540059b18bed6f8f223b7808d09d8fc48617bc55ece87"

},

"device\_sn" : "0691A001100112638L",

"event\_time" : 1567492833482,

"face\_id" : "5d6e079a4e01730008a8b398",

"faceset\_id" : "5d5e078cba18560008ba77b7",

"faceset\_type" : "vip",

"gender" : 1,

"in\_out\_type" : 1,

"manual\_age" : 32,

"match\_url" :

{

"expire" : "2019-09-10T06:23:56Z",

"url" : "https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/Pictureserver/a55dfe635d4bf67a2f1b487d/201993/a55dfe635d4bf67a2f1b487d\_3dSuvDFEY5ye-1567491994128486938?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T063456Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=97680c95ff816a2fa358995cadd78a09bc33b84d220c2495761755a2c9821236"

},

"msg\_type" : 2,

"score" : 0.91190749999999998,

"space\_id" : "a55dfe635d4bf67a2f1b487d\_NHcXY2YV",

"track\_id" : 1567486404

}

{

"age" : 4,

"app\_id" : "a55dfe635d4bf67a2f1b487d",

"capture\_url" :

{

"expire" : "2019-09-10T06:26:30Z",

"url" : "https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/0691A001100112638L\_20190903064258\_00000000009514946763\_000002236.jpg?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T063730Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=a26409e30ae63f6efbd16a3ae636027cfc2feb6f9e1f577d8be5f1611cf82c2a"

},

"device\_sn" : "0691A001100112638L",

"event\_time" : 1567492984639,

"face\_id" : "autoreg\_0691A001100112638L\_1567486702",

"faceset\_id" : "5d5e078fba18560008ba77ba",

"faceset\_type" : "autoreg",

"gender" : 1,

"in\_out\_type" : 0,

"manual\_age" : -1,

"match\_url" :

{

"expire" : "2019-09-10T06:26:30Z",

"url" : "https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/0691A001100112638L\_20190903063143\_00000000008841728437\_000002022.jpg?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T063730Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=4cc53db30261c9c45a29dd1cad400afb33f092728f4ef1a871cffb1fb8e3706a"

},

"msg\_type" : 2,

"score" : 0.93085260000000003,

"space\_id" : "a55dfe635d4bf67a2f1b487d\_NHcXY2YV",

"track\_id" : 1567487650

}

{

"age" : 4,

"app\_id" : "a55dfe635d4bf67a2f1b487d",

"capture\_url" :

{

"expire" : "2019-09-10T06:26:31Z",

"url" : "https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/0691A001100112638L\_20190903064258\_00000000009514946763\_000002236.jpg?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T063731Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=e497d7c4cded08ef1e1bbcf6444f737c2079122bd9bd6e8d33c05532ef91a1ee"

},

"device\_sn" : "0691A001100112638L",

"event\_time" : 1567492988243,

"face\_id" : "autoreg\_0691A001100112638L\_1567486702",

"faceset\_id" : "5d5e078fba18560008ba77ba",

"faceset\_type" : "autoreg",

"gender" : 1,

"in\_out\_type" : 1,

"manual\_age" : -1,

"match\_url" :

{

"expire" : "2019-09-10T06:26:31Z",

"url" : "https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/0691A001100112638L\_20190903063143\_00000000008841728437\_000002022.jpg?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T063731Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=2e6d2b1dc87a447aa4d08c0c018817869ea00fb4a7214c17e1a1062c12af918e"

},

"msg\_type" : 2,

"score" : 0.93085260000000003,

"space\_id" : "a55dfe635d4bf67a2f1b487d\_NHcXY2YV",

"track\_id" : 1567487650

}

{

"age" : 4,

"app\_id" : "a55dfe635d4bf67a2f1b487d",

"capture\_url" :

{

"expire" : "2019-09-10T06:29:05Z",

"url" : "https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/0691A001100112638L\_20190903064335\_00000000009550653186\_000002240.jpg?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T064005Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=c45c58fcf0d2dc2b908c971e2f29025198653ca802d0cb73f1c99860fdf3e3c7"

},

"device\_sn" : "0691A001100112638L",

"event\_time" : 1567493141545,

"face\_id" : "autoreg\_0691A001100112638L\_1567486702",

"faceset\_id" : "5d5e078fba18560008ba77ba",

"faceset\_type" : "autoreg",

"gender" : 1,

"in\_out\_type" : 0,

"manual\_age" : -1,

"match\_url" :

{

"expire" : "2019-09-10T06:29:05Z",

"url" : "https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/0691A001100112638L\_20190903063143\_00000000008841728437\_000002022.jpg?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T064005Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=46db07c1e5970eef8510fffbf38fe98636c0890debc02fb0f362dcda457dd56b"

},

"msg\_type" : 2,

"score" : 0.93210936,

"space\_id" : "a55dfe635d4bf67a2f1b487d\_NHcXY2YV",

"track\_id" : 1567487661

}

{

"age" : 4,

"app\_id" : "a55dfe635d4bf67a2f1b487d",

"capture\_url" :

{

"expire" : "2019-09-10T07:30:07Z",

"url" : "https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/0691A001100112638L\_20190903074639\_00000000013325367181\_000003129.jpg?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T074107Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=1e50faeffb5a7b19e48fdf984ff5b919e40d8d5706397665e44bfd2349933e0f"

},

"device\_sn" : "0691A001100112638L",

"event\_time" : 1567496790342,

"face\_id" : "noreg\_0691A001100112638L\_1567488499",

"faceset\_id" : "",

"faceset\_type" : "",

"gender" : 1,

"in\_out\_type" : 1,

"manual\_age" : -1,

"match\_url" :

{

"expire" : "",

"url" : ""

},

"msg\_type" : 2,

"score" : -1.0,

"space\_id" : "a55dfe635d4bf67a2f1b487d\_NHcXY2YV",

"track\_id" : 1567488499

}

{

"age" : 4,

"app\_id" : "a55dfe635d4bf67a2f1b487d",

"capture\_url" :

{

"expire" : "2019-09-10T07:31:36Z",

"url" : "https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/0691A001100112638L\_20190903074812\_00000000013426722200\_000003144.jpg?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T074236Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=7f9bd1dffb5669c25899e7c7a521e923dbc80f172e7ce46332fad19993a6c392"

},

"device\_sn" : "0691A001100112638L",

"event\_time" : 1567496890698,

"face\_id" : "autoreg\_0691A001100112638L\_1566447550",

"faceset\_id" : "5d5e078fba18560008ba77ba",

"faceset\_type" : "autoreg",

"gender" : 1,

"in\_out\_type" : 0,

"manual\_age" : -1,

"match\_url" :

{

"expire" : "2019-09-10T07:31:36Z",

"url" : "https://horizon-ib-test.oss-cn-zhangjiakou.aliyuncs.com/0691A001100112638L\_20190828012229\_00000000086654323835\_000030365.jpg?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=LTAIiUHHchQK9Lfy%2F20190903%2Fcn-zhangjiakou%2Fs3%2Faws4\_request&X-Amz-Date=20190903T074236Z&X-Amz-Expires=604740&X-Amz-SignedHeaders=host&X-Amz-Signature=632976017abbd2425ffa0dfffc7dda921ab8ba285f14f904269821775bdd99a8"

},

"msg\_type" : 2,

"score" : 0.9344865,

"space\_id" : "a55dfe635d4bf67a2f1b487d\_NHcXY2YV",

"track\_id" : 1567488514

}

## #取消到店记录的实时推送 del\_subscription\_id

功能：取消[到店记录订阅](https://iotdoc.horizon.ai/busiopenapi/part5_api_analysis_tools/statistics.html" \l "part5_4)发起的订阅，输入的subscription\_id为订阅时候返回的id

Input:**subscription\_id**

Response:

| **名称** | **类型** | **是否必选** | **说明** |
| --- | --- | --- | --- |
| request\_id | string | 用于区分每一次请求的唯一的字符串 |  |
| code | int | 当前request返回状态码，0表示成功，其他详见错误码 |  |
| message | string | 当前request返回描述信息 |  |
| data | object | null |  |

测试输入：

del\_subscription\_id(**"47b2420b-c942-11e9-907c-0a58c0a80e02"**)

测试输出：

{'message': 'success', 'code': 0, 'request\_id': 'bc1de8af-ed74-4492-bb07-f8908101324c', 'data': None}