DS-FACE 人脸识别引擎接口文档

Restful Web Service API



**北京深感科技有限公司**

目录

[DS-FACE 人脸识别引擎接口文档 1](#_Toc519245669)

[Restful Web Service API 1](#_Toc519245670)

[简介 5](#_Toc519245671)

[核心概念 5](#_Toc519245672)

[通用响应信息 5](#_Toc519245673)

[通用错误响应信息 6](#_Toc519245674)

[人脸检测接口 6](#_Toc519245675)

[概述 6](#_Toc519245676)

[图片要求 6](#_Toc519245677)

[调用URL 7](#_Toc519245678)

[调用方法 7](#_Toc519245679)

[请求参数 7](#_Toc519245680)

[返回值说明 8](#_Toc519245681)

[当前API特有的错误信息 8](#_Toc519245682)

[示例 8](#_Toc519245683)

[请求示例 8](#_Toc519245684)

[响应示例 9](#_Toc519245685)

[相关资源 9](#_Toc519245686)

[人脸图片注册接口 10](#_Toc519245687)

[概述 10](#_Toc519245688)

[调用URL 10](#_Toc519245689)

[调用方法 10](#_Toc519245690)

[请求参数 10](#_Toc519245691)

[返回值说明 11](#_Toc519245692)

[当前API特有的错误信息 11](#_Toc519245693)

[示例 11](#_Toc519245694)

[请求示例 11](#_Toc519245695)

[请求成功响应示例 12](#_Toc519245696)

[相关资源 12](#_Toc519245697)

[人脸图片查询接口 12](#_Toc519245698)

[概述 12](#_Toc519245699)

[调用URL 13](#_Toc519245700)

[调用方法 13](#_Toc519245701)

[请求参数 13](#_Toc519245702)

[返回值说明 13](#_Toc519245703)

[当前API特有的错误信息 13](#_Toc519245704)

[示例 14](#_Toc519245705)

[请求示例 14](#_Toc519245706)

[请求成功响应示例 14](#_Toc519245707)

[相关资源 14](#_Toc519245708)

[人脸图片搜索接口 15](#_Toc519245709)

[概述 15](#_Toc519245710)

[调用URL 15](#_Toc519245711)

[调用方法 15](#_Toc519245712)

[请求参数 15](#_Toc519245713)

[返回值说明 16](#_Toc519245714)

[当前API特有的错误信息 16](#_Toc519245715)

[示例 17](#_Toc519245716)

[请求示例 17](#_Toc519245717)

[请求成功响应示例 17](#_Toc519245718)

[相关资源 17](#_Toc519245719)

[静态人脸图片比对接口 18](#_Toc519245720)

[概述 18](#_Toc519245721)

[调用URL 18](#_Toc519245722)

[调用方法 18](#_Toc519245723)

[请求参数 18](#_Toc519245724)

[返回值说明 18](#_Toc519245725)

[当前API特有的错误信息 19](#_Toc519245726)

[示例 19](#_Toc519245727)

[请求示例 19](#_Toc519245728)

[请求成功响应示例 19](#_Toc519245729)

[相关资源 19](#_Toc519245730)

[人脸库图片删除接口 19](#_Toc519245731)

[概述 19](#_Toc519245732)

[调用URL 20](#_Toc519245733)

[调用方法 20](#_Toc519245734)

[请求参数 20](#_Toc519245735)

[返回值说明 20](#_Toc519245736)

[当前API特有的错误信息 20](#_Toc519245737)

[示例 21](#_Toc519245738)

[请求示例 21](#_Toc519245739)

[请求成功响应示例 21](#_Toc519245740)

[相关资源 21](#_Toc519245741)

[创建人脸库接口 21](#_Toc519245742)

[概述 21](#_Toc519245743)

[调用URL 22](#_Toc519245744)

[调用方法 22](#_Toc519245745)

[请求参数 22](#_Toc519245746)

[返回值说明 22](#_Toc519245747)

[当前API特有的错误信息 22](#_Toc519245748)

[示例 22](#_Toc519245749)

[请求示例 22](#_Toc519245750)

[请求成功响应示例 23](#_Toc519245751)

[相关资源 23](#_Toc519245752)

[获取人脸库信息接口 23](#_Toc519245753)

[概述 23](#_Toc519245754)

[调用URL 23](#_Toc519245755)

[调用方法 24](#_Toc519245756)

[请求参数 24](#_Toc519245757)

[返回值说明 24](#_Toc519245758)

[当前API特有的错误信息 25](#_Toc519245759)

[示例 25](#_Toc519245760)

[请求示例 25](#_Toc519245761)

[请求成功响应示例 25](#_Toc519245762)

[相关资源 26](#_Toc519245763)

[获取人脸库属性接口 26](#_Toc519245764)

[概述 26](#_Toc519245765)

[调用URL 26](#_Toc519245766)

[调用方法 26](#_Toc519245767)

[请求参数 27](#_Toc519245768)

[返回值说明 27](#_Toc519245769)

[当前API特有的错误信息 27](#_Toc519245770)

[示例 27](#_Toc519245771)

[请求示例 27](#_Toc519245772)

[请求成功响应示例 27](#_Toc519245773)

[相关资源 28](#_Toc519245774)

[获取人脸库列表接口 28](#_Toc519245775)

[概述 28](#_Toc519245776)

[调用URL 28](#_Toc519245777)

[调用方法 28](#_Toc519245778)

[请求参数 28](#_Toc519245779)

[返回值说明 29](#_Toc519245780)

[当前API特有的错误信息 29](#_Toc519245781)

[示例 29](#_Toc519245782)

[请求示例 29](#_Toc519245783)

[请求成功响应示例 29](#_Toc519245784)

[相关资源 30](#_Toc519245785)

[修改人脸库基本属性接口 30](#_Toc519245786)

[概述 30](#_Toc519245787)

[调用URL 30](#_Toc519245788)

[调用方法 31](#_Toc519245789)

[请求参数 31](#_Toc519245790)

[返回值说明 31](#_Toc519245791)

[当前API特有的错误信息 31](#_Toc519245792)

[示例 31](#_Toc519245793)

[请求示例 31](#_Toc519245794)

[请求成功响应示例 32](#_Toc519245795)

[相关资源 32](#_Toc519245796)

[删除人脸库接口 32](#_Toc519245797)

[概述 32](#_Toc519245798)

[调用URL 32](#_Toc519245799)

[调用方法 32](#_Toc519245800)

[请求参数 32](#_Toc519245801)

[返回值说明 33](#_Toc519245802)

[当前API特有的错误信息 33](#_Toc519245803)

[示例 33](#_Toc519245804)

[请求示例 33](#_Toc519245805)

[请求成功响应示例 33](#_Toc519245806)

[相关资源 34](#_Toc519245807)

# 简介

此文档描述了DeepSense人脸识别引擎所提供的web service接口及详细的使用说明。DeepSense人脸识别引擎提供人脸检测服务，人脸特征提取服务，人脸图片的入库、查询和删除服务，以及人脸图片的比对服务等。系统采用基于消息队列技术的分布式架构，提供标准的HTTP API, 以Restful Web Service接口的形式对外提供服务。

# 核心概念

**人脸**

系统中使用FaceID的概念来标识一张人脸。

**人脸集合**

系统中使用FaceSet来表示一组人脸（FaceID）的集合。在一个FaceSet中FaceID不重复。FaceSetID是一个FaceSet的唯一标识。

# 通用响应信息

所有的API成功调用后会返回一个JSON格式的字符串。通用的返回值定义如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 是否必须 | 字段 | 类型 | 说明 |
| 必须 | id | String | 返回message的id。 |
| 可选 | header | 自定义类型 |  |
| 必须（二选一） | body | 自定义类型 | 详见各个API说明。 |
| error | 自定义类型 | 详见通用错误响应信息。 |

当API调用成功时, HTTP 状态代码为200，API返回的结果包含在body类型中，没有error类型信息。

当API调用成功，但API执行过程中出现任何可控制异常。API返回HTTP状态码仍为200， 返回结果中只包含error类型的信息，没有body类型信息。例如：传递的参数类型正确，但范围超出API的允许值，或者传入的图片类型不是API支持的格式等情况。

当API调用失败时，常见的返回值类型如下：

|  |  |  |
| --- | --- | --- |
| HTTP  状态代码 | 错误信息 | 说明 |
| 400 | ERROR\_BAD\_PARAM | 传入参数错误。某个参数解析出错（比如必须是数字，但是输入的是非数字字符串; 或者长度过长，等等.） |
| 404 | API\_NOT\_FOUND | 所调用的API不存在。 |
| 500 | ERROR\_DETECT\_FACE\_FAILED | 服务器内部错误，当此类错误发生时请再次请求，如果持续出现此类错误，请联系技术支持。 |
| ERROR\_NO\_RESPONSE |

# 通用错误响应信息

当Restful API调用不成功时，统一使用error字段来返回错误。error字段的JSON格式定义如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 是否必须 | 字段 | 类型 | 说明 |
| 必须 | code | String | 返回自定义的error code，详见各个API说明。 |
| 可选 | literal | String | 错误简短描述。 |
| 可选 | detail | String | 错误详细信息。 |

# 人脸检测接口

## 概述

此接口用于检测输入的图像中是否包含人脸。

输入：图像数据及相关信息。

输出：检测人脸结果。

## 图片要求

图片格式：JPG(JPEG)，PNG, **更多格式需要测试确定**

图片像素尺寸：最小 48\*48 像素，最大 4096\*4096 像素

图片文件大小：2 MB

最小人脸像素尺寸： 系统能够检测到的人脸框为一个正方形，正方形边长的最小值不低于 40 像素。

## 调用URL

/ds-server/face/v2/detect

## 调用方法

POST

## 请求参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | | 类型 | 描述 | 是否必须 |
| image | imageData | 字符串。 | Base64 编码的二进制图片数据。 | Yes |
| imageWidth | 整型 | 图像像素宽度。 | No |
| imageHeight | 整型 | 图像像素高度。 | No |
| numComponents | 整型 | 每个像素字节数。 | No |
| imageFormat | 整型 | 图像格式。  默认为0 (JPG) | Yes |
| detectableFaceSizeLevel | | 整型 | 可检测的人脸大小级别。从1级开始，共5级。级别越小可检测的人脸越小。默认是3级。 | No |
| detectableConfidenceLevel | | 整型 | 检测人脸的可信度级别。从1级开始，共5级。级别越小，可信度越低，检测出“假脸”的可能性越高，一般用于最大限度的保留可能有人脸的检测情况。级别越大，可信度越高，但是遗漏“真脸”的可能性越大。默认级别为3级。 | No |

## 返回值说明

API执行成功时，返回的json类型数据描述如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| hasFace | Boolean | 图像中是否检测到人脸。 | Yes |
| count | 整型 | 返回检测到的人脸个数。 | No |
| faceInfo | 自定义faceInfo类型数组 | 返回检测到的人脸详细信息。 | No |

其中faceInfo类型定义如下:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| xtl | 整型 | 检测到的人脸框的左上角x坐标 | Yes |
| ytl | 整型 | 检测到的人脸框的左上角y坐标 | Yes |
| xbr | 整型 | 检测到的人脸框的右下角x坐标 | Yes |
| ybr | 整型 | 检测到的人脸框的右下角y坐标 | Yes |
| size | 整型 | 检测到的人脸框的大小 | No |
| confidence | 整型 | 检测到人脸的置信度 | Yes |
| quality | 整型 | 检测到的人脸的质量 | No |

## 当前API特有的错误信息

|  |  |  |
| --- | --- | --- |
| HTTP Status Code | Error Code | Details |
| 400 | ERROR\_BAD\_PARAM | no request body or input image data is invalid |
| ERROR\_INCOMPATIBLE\_IMAGE | Image does not meet the requirements |
| 500 | ERROR\_NO\_RESPONSE | Maybe internal error, try again or contact technical support. |
| 200 | ERROR\_DETECT\_FACE\_FAILED | failed to do detect face on input image. You can find detailed error message in general error message. |

## 示例

### 请求示例

|  |
| --- |
| POST http://192.168.1.69:8080/ds-server/face/v2/detect HTTP/1.1  Accept-Encoding: gzip,deflate  Content-Type: application/json  Content-Length: 39964  Host: 192.168.1.69:8080  Proxy-Connection: Keep-Alive  User-Agent: Apache-HttpClient/4.1.1 (java 1.5)  {  "image":{  "imageData":"/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAA........QAHwAAAQUBAFFJ7/",  "imageFormat":0  },  "detectableFaceSizeLevel":1,  "detectableConfidenceLevel":1  }  } |

### 响应示例

|  |
| --- |
| HTTP/1.1 200 OK  Transfer-Encoding: chunked  Content-Type: application/json;charset=UTF-8  Date: Wed, 10 Jan 2018 08:08:20 GMT  Keep-Alive: timeout=38  Server: Apache-Coyote/1.1  {  "header": {},  "body": {"voDetectFaceResult": {  "hasFace": true,  "count": 1,  "faceInfo": [ {  "xtl": 67,  "ytl": 126,  "xbr": 231,  "ybr": 290,  "size": 0,  "confidence": 819,  "quality": 0  }]  }}  } |

## 相关资源

None

# 人脸图片注册接口

## 概述

此接口用于把输入的人脸图片添加到人脸库中。

输入：图像数据及相关信息。

## 调用URL

/ds-server/face/v2/enroll

## 调用方法

POST

## 请求参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | | 类型 | 描述 | 是否必须 |
| image | imageData | 字符串。 | Base64 编码的二进制图片数据。 | Yes |
| imageWidth | 整型 | 图像像素宽度。 | No |
| imageHeight | 整型 | 图像像素高度。 | No |
| numComponents | 整型 | 每个像素字节数。 | No |
| imageFormat | 整型 | 图像格式。  默认为0 (JPG) | Yes |
| appImgId | | 字符串 | 应用程序定义的图片唯一ID。此ID由应用程序传入且必须保证唯一。最长256个字符。 | No |
| faceSetId | | 字符串 | FaceSet ID代表了系统中唯一的一个FaceSet。应用程序指定图片要注册到那个FaceSet中。 | （2选1）  不知道,注册到系统唯一一个默认的FaceSet中。 |
| appFaceSetId | | 字符串 | 应用程序为对应的人脸库设置的ID。 |
| notifySS | | Boolean | 是否将新加入到faceSet的图片特征加载到缓存 | No |

## 返回值说明

API执行成功时且人脸图片成功加入到数据库时，返回的json类型数据描述如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| faceId | 字符串 | 引擎为成功加入到人脸库的每一个人脸信息返回的唯一ID。32个字符的字符串。 | Yes |
| faceSetId | 字符串 | 成功注册时返回对应的人脸库的ID。 | Yes |
| appFaceSetId | 整型 | 成功注册时返回应用程序为对应的人脸库设置的ID。 | Yes |

API执行成功时但人脸图片没有成功到数据库时，返回的通用的错误响应信息。详细错误类型可以参见error字段。

## 当前API特有的错误信息

|  |  |  |
| --- | --- | --- |
| HTTP Status Code | Error Code | Details |
| 400 | ERROR\_BAD\_PARAM | no request body or input image data is invalid. |
| ERROR\_NOT\_SUPPORTED\_FORMAT | not supported image format |
| ERROR\_INCOMPATIBLE\_IMAGE | Image does not meet the requirements |
| 200 | ERROR\_EXTRACT\_FAILED | failed to detect face on input image. |
| 500 | ERROR\_EXTRACT\_FAILED | Maybe internal error, try again or contact technical support. |

## 示例

### 请求示例

|  |
| --- |
| POST http://192.168.1.69:8080/ds-server/face/v2/enroll HTTP/1.1  Accept-Encoding: gzip,deflate  Content-Type: application/json  Content-Length: 39901  Host: 192.168.1.69:8080  Proxy-Connection: Keep-Alive  User-Agent: Apache-HttpClient/4.1.1 (java 1.5)  {  "image":{  "imageData":"/9j/4AAQSkZJRgABAQAAAQABAAD/2........QAAAAQUBAFFJ7/15E9D/2Q==",  "imageFormat":0  }  } |

### 请求成功响应示例

|  |
| --- |
| HTTP/1.1 200 OK  Transfer-Encoding: chunked  Content-Type: application/json;charset=UTF-8  Date: Wed, 10 Jan 2018 08:22:19 GMT  Keep-Alive: timeout=38  Server: Apache-Coyote/1.1  {  "header": {},  "body": {"voFaceEnrollResult": {  "faceId": "46b61f04475640b5b5ffccbeb5e454c6",  "faceSetId": "\_default\_face\_set"  }}  } |

## 相关资源

None

# 人脸图片查询接口

## 概述

此接口实现了用输入的人脸图片ID查到对应图片的详细信息

输入：人脸图片ID和人脸库ID

## 调用URL

/ds-server/face/v2/getFaceInfo

## 调用方法

GET

## 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| faceId | 字符串 | 引擎为成功加入到人脸库的每一个人脸图片返回的唯一ID。32个字符的字符串。 | 必须（2选1）  如果都指定，则使用faceId。 |
| appImgId | 字符串 | 如果输入了应用程序定义的图片ID，则返回该ID，否则返回结果不包含该项。 |
| faceSetId | 字符串 | 人脸库ID | 必须（2选1）  如果都指定，则使用faceSetId。 |
| appFaceSetId | 字符串 | 应用程序定义的人脸库Id |

## 返回值说明

API执行成功时，返回的json类型数据描述如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| faceId | 字符串 | 引擎为成功加入到人脸库的每一个人脸图片返回的唯一ID。32个字符的字符串。 | Yes |
| appImgId | 字符串 | 如果输入了应用程序定义的图片 | No |
| faceImage | 字符串 | 图片的Base64码 | No |
| faceFeature | 字符串 | 图片特征的Base64码 | Yes |

## 当前API特有的错误信息

|  |  |  |
| --- | --- | --- |
| HTTP Status Code | Error Code | Details |
| 200 | ERROR\_FIND\_FACESET | faceSet not find by faceSetId |
| 200 | ERROR\_FIND\_FACESET | faceSet not find by appFaceSetId |
| 200 | ERROR\_FIND\_FACE | face not find by faceId |
| 200 | ERROR\_FIND\_FACE | face not find by appImgId |

## 示例

### 请求示例

|  |
| --- |
| GET http://192.168.1.78:8080/ds-server/face/v2/getFaceInfo?faceId=7860196acba24c8c8a1cf221ae9205a2&faceSetId=\_default\_face\_set HTTP/1.1  Accept-Encoding: gzip,deflate  Host: 192.168.1.78:8080  Proxy-Connection: Keep-Alive  User-Agent: Apache-HttpClient/4.1.1 (java 1.5) |

### 请求成功响应示例

|  |
| --- |
| HTTP/1.1 200 OK  Transfer-Encoding: chunked  Content-Type: application/json;charset=UTF-8  Date: Fri, 13 Jul 2018 02:20:31 GMT  Keep-Alive: timeout=38  Server: Apache-Coyote/1.1  {  "header": {},  "body": {"voFace": {  "faceId": "7860196acba24c8c8a1cf221ae9205a2",  "faceFeature": "AAAEYiMBIwUBkAHgAAAAAAAA.... vYbADL3Scmy+W",  }}  } |

## 相关资源

None

# 人脸图片搜索接口

## 概述

此接口实现了用输入的人脸图片和选择的人脸库中已经注册的所有人脸图片做比对，按照输入的阈值和要求返回的最多记录个数，返回比对结果。

输入：图像数据及相关信息。

## 调用URL

/ds-server/face/v2/search

## 调用方法

POST

## 请求参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | | 类型 | 描述 | 是否必须 |
| image | imageData | 字符串 | Base64 编码的二进制图片数据。 | Yes |
| imageWidth | 整型 | 图像像素宽度。 | No |
| imageHeight | 整型 | 图像像素高度。 | No |
| numComponents | 整型 | 每个像素字节数。 | No |
| imageFormat | 整型 | 图像格式。  默认为0 (JPG) | Yes |
| faceSetId | | 字符串 | 要搜索的人脸库在系统中的唯一ID | Yes |
| appFaceSetId | | 字符串 | 应用程序为要搜索的人脸库设置的唯一ID | No |
| topN | | 整型 | 返回比分最高的前多少条比对记录。默认是1。 | No |
| threshold | | 浮点型  （float） | 相似度阈值（0-100之间）。如果设置，只有超过此阈值的比中记录才会返回，可以不设。如果同时设置了topN和threshold，则会返回阈值超过threshold的最多topN条记录。默认不设。 | No |

## 返回值说明

API执行成功时，返回的json类型数据描述如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| searchResult | 数组 | 搜索结果对象数组。  注：如果指定了阈值，但没有超过设置阈值的比中记录，则此数组为空。 | Yes |

searchResult数组中单个元素的结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | | 类型 | 描述 | 是否必须 |
| searchResultInEngine | faceId | 字符串 | 代表人脸库中一张人脸图片的唯一ID。32个字符的字符串。 | Yes |
| appImgId | 字符串 | 应用程序定义的代表人脸库中一张人脸图片的唯一ID。如果应用程序在注册人脸图片时没有指定这个ID，则返回结果不包含该项。 | No |
| score | 浮点型 | 输入人脸图片与faceId所代表的人脸图片的相似度。（1～100） | Yes |
| faceSetId | 字符串 | 人脸库ID | No |
| appFaceSetId | 字符串 | 应用程序指定的人脸库ID | No |

API执行失败时，返回的通用的错误响应信息。详细错误类型可以参见error字段。

## 当前API特有的错误信息

|  |  |  |
| --- | --- | --- |
| HTTP Status Code | Error Code | Details |
| 400 | ERROR\_BAD\_PARAM | no image data |
| ERROR\_NOT\_SUPPORTED\_FORMAT | not supported image format |
| ERROR\_INCOMPATIBLE\_IMAGE | Image does not meet the requirements |
| 200 | ERROR\_EXTRACT\_FAILED | failed to detect face on input image. |

## 示例

### 请求示例

|  |
| --- |
| POST http://192.168.1.69:8080/ds-server/face/v2/search HTTP/1.1  Accept-Encoding: gzip,deflate  Content-Type: application/json  Content-Length: 39948  Host: 192.168.1.69:8080  Proxy-Connection: Keep-Alive  User-Agent: Apache-HttpClient/4.1.1 (java 1.5)  {  "image":{  "imageData":"/9j/4AAQSkZJRgABAQAAAQABAA........QAAAAQUBAFFJ7/15E9D/2Q==",  "imageFormat":0},  "faceSetId":"\_default\_face\_set",  "topN":1  } |

### 请求成功响应示例

|  |
| --- |
| HTTP/1.1 200 OK  Transfer-Encoding: chunked  Content-Type: application/json;charset=UTF-8  Date: Wed, 10 Jan 2018 08:35:29 GMT  Keep-Alive: timeout=38  Server: Apache-Coyote/1.1  {  "header": {},  "body": {"voFaceSearchResponse": {"searchResult": [{"searchResultInEngine": {  "faceSetId": "\_default\_face\_set",  "faceId": "89393fffc49a4c81bd3519abcaa29c73",  "score": 99.99977  }}]}}  } |

## 相关资源

None

# 静态人脸图片比对接口

## 概述

此接口实现了对输入的两张人脸图片做比对，返回两张人脸图片的相似度。

输入：两张人脸图像数据及相关信息。

## 调用URL

/ds-server/face/v2/verify

## 调用方法

POST

## 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| ImageData1 | 字符串 | Base64 编码的二进制图片数据。 | Yes |
| ImageWidth1 | 整型 | 图像像素宽度。 | No |
| ImageHeight1 | 整型 | 图像像素高度。 | No |
| numComponents1 | 整型 | 每个像素字节数。 | No |
| imageFormat1 | 整型 | 图像格式。  默认为0 (JPG) | Yes |
| ImageData2 | 字符串 | Base64 编码的二进制图片数据。 | Yes |
| ImageWidth2 | 整型 | 图像像素宽度。 | No |
| ImageHeight2 | 整型 | 图像像素高度。 | No |
| numComponents2 | 整型 | 每个像素字节数。 | No |
| imageFormat2 | 整型 | 图像格式。  默认为0 (JPG) | Yes |

## 返回值说明

API执行成功时，返回的json类型数据描述如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| confidence | 浮点型 | 输入的两张人脸图片的相似度。（1～100） | Yes |

API执行失败时，返回的通用的错误响应信息。详细错误类型可以参见error字段。

## 当前API特有的错误信息

|  |  |
| --- | --- |
| Error Code | Details |
| ERROR\_BAD\_PARAM | no image data |
| ERROR\_NOT\_SUPPORTED\_FORMAT | not supported image format |

## 示例

### 请求示例

|  |
| --- |
| POST http://localhost:8080/ds-server/restful/face/detect HTTP/1.1  Accept-Encoding: gzip,deflate  Content-Type: application/json  Content-Length: 64603  Host: localhost:8080  Connection: Keep-Alive  User-Agent: Apache-HttpClient/4.1.1 (java 1.5)  {  "imageFormat":0,  "detectableFaceSizeLevel":1,  "detectableConfidenceLevel":1,  "imageData":"/9j/4AAQSkZJRgAhEBAxEB/8........QAHwAAAQUBAFFJ7/15E9D/2Q=="  } |

### 请求成功响应示例

|  |
| --- |
| HTTP/1.1 200 OK  ...  TBD |

## 相关资源

None

# 人脸库图片删除接口

## 概述

此接口实现了根据输入的图片ID信息从人脸库中删除相关的人脸记录。

输入：人脸图片在数据库的唯一ID信息。可以是在添加人脸照片到人脸库时指定的appImgId或添加成功后引擎返回的faceId。

## 调用URL

/ds-server/face/v2/delete

## 调用方法

DELETE

## 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| faceId | 字符串 | 引擎为成功加入到人脸库的每一个人脸图片返回的唯一ID。32个字符的字符串。 | 必须（2选1）  如果都指定，则使用faceId。 |
| appImgId | 字符串 | 如果输入了应用程序定义的图片ID，则返回该ID，否则返回结果不包含该项。 |
| faceSetId | 字符串 | 人脸库ID | 必须（2选1）  如果都指定，则使用faceSetId。 |
| appFaceSetId | 字符串 | 应用程序定义的人脸库Id |
| notifySS | boolean | 是否将faceSet中的图片从缓存中清除 | No |

## 返回值说明

API执行成功时，返回的json类型数据描述如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| faceCount | 整型 | 返回删除指定的人脸图片后，当前人脸库总共包含的记录数。 | No |

API执行失败时，返回的通用的错误响应信息。详细错误类型可以参见error字段。

## 当前API特有的错误信息

|  |  |  |
| --- | --- | --- |
| HTTP Status Code | Error Code | Details |
| 200 | ERROR\_FIND\_FACE | face can not find by appFaceSetId |
| 200 | ERROR\_FIND\_FACE | face can not find by faceSetId |

## 示例

### 请求示例

|  |
| --- |
| DELETE http://192.168.1.214:8080/ds-server/face/v2/delete?faceId=18f10a4404e8446fb67a4afd02c1ebfa&faceSetId=\_default\_face\_set&notifySS=true HTTP/1.1  Accept-Encoding: gzip,deflate  Content-Type: application/json  Content-Length: 116  Host: 192.168.1.214:8080  Proxy-Connection: Keep-Alive  User-Agent: Apache-HttpClient/4.1.1 (java 1.5) |

### 请求成功响应示例

|  |
| --- |
| HTTP/1.1 200  Transfer-Encoding: chunked  Content-Type: application/json;charset=UTF-8  Date: Tue, 10 Jul 2018 01:35:52 GMT  Keep-Alive: timeout=38  {  "header": {},  "body": {}  } |

## 相关资源

None

# 创建人脸库接口

## 概述

此接口实现了创建人脸库。

输入：人脸库的基本属性信息。

## 调用URL

/ds-server/face/v2/createFaceSet

## 调用方法

POST

## 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| appFaceSetId | 字符串 | 应用程序定义的人脸库Id | No |
| tag | 字符串 | 自定义标签，用来对人脸库分组 | No |
| description | 字符串 | 人脸库的描述信息 | No |
| userData | 字符串 | 自定义用户信息 | No |

## 返回值说明

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| faceSetId | 字符串 | 人脸库Id（系统默认生成UUID） | Yes |
| appFaceSetId | 字符串 | 应用程序定义的人脸库Id | No |
| tag | 字符串 | 自定义标签，用来对人脸库分组 | No |
| description | 字符串 | 人脸库的描述信息 | No |
| userData | 字符串 | 自定义用户信息 | No |

## 当前API特有的错误信息

|  |  |  |
| --- | --- | --- |
| HTTP Status Code | Error Code | Details |
| 200 | DS\_EX\_ALREADY\_EXIST | Same application-faceset id existed |

## 示例

### 请求示例

|  |
| --- |
| POST http://192.168.1.214:8080/ds-server/face/v2/createFaceSet HTTP/1.1  Accept-Encoding: gzip,deflate  Content-Type: application/json  Content-Length: 94  Host: 192.168.1.214:8080  Proxy-Connection: Keep-Alive  User-Agent: Apache-HttpClient/4.1.1 (java 1.5)  {  "appFaceSetId":"11111",  "tag":"test1",  "userData":"xxxxx1",  "description":"xxxx1"  } |

### 请求成功响应示例

|  |
| --- |
| HTTP/1.1 200  Transfer-Encoding: chunked  Content-Type: application/json;charset=UTF-8  Date: Mon, 09 Jul 2018 12:41:50 GMT  Keep-Alive: timeout=38  {  "header":{},  "body":{  "voCreateFaceSetResponse":{"faceSetId":"a94621dd058a403ebb3cb161af65a3a8"}  }  } |

## 相关资源

None

# 获取人脸库信息接口

## 概述

此接口用于获取人脸库的详细信息。

## 调用URL

/ds-server/face/v2/getFaceSetDetailInfo

## 调用方法

GET

## 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| start | 整型 | 人脸库中人脸记录的起始序号。范围从1到人脸库中人脸记录的总个数。配合返回值中的next参数一起使用来枚举人脸库中的所有记录。  没有指定的情况下，此API返回从最小序号开始的最多100条人脸记录信息。并通过next返回值，返回第101条记录的序号。 | 可选 |
| count | 整型 | 返回的条目数 | NO |
| faceSetId | 字符串 | 人脸库Id | 必须（2选1）  也可都指定 |
| appFaceSetId | 字符串 | 应用程序定义的人脸库Id |

## 返回值说明

API执行成功时，返回的json类型数据描述如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| faceCount | 整型 | 当前人脸库总共包含的记录数。 | No |
| FaceIds | 数组 | 从指定序号（没有指定，从最小序号）开始的最多100条人脸记录的图片ID值数组。 | Yes |
| appImgIds | 数组 | 从指定序号（没有指定，从最小序号）开始的最多100条人脸记录的应用程序设定的图片ID值数组。如果应用程序在注册人脸图片时，没有指定appImgId，则不返回此项。 | No |
| next | 整型 | 用于进行下一次请求。返回值表示排在此次返回的所有人脸记录之后的下一个人脸记录的序号。  如果返回此字段，则说明未返回人脸库中的所有记录。可以将此字段的返回值，在下一次调用时传入 start 字段中，获取接下来的 人脸记录。  如果没有返回该字段，则说明已经返回人脸库下的所有人脸记录。 | No |

API执行失败时，返回的通用的错误响应信息。详细错误类型可以参见error字段。

## 当前API特有的错误信息

|  |  |  |
| --- | --- | --- |
| HTTP Status Code | Error Code | Details |
| 400 | ERROR\_BAD\_PARAM | count can not more than 100 or can not less than 1 |
| 200 | ERROR\_FIND\_FACE | face can not find by appFaceSetId |
| 200 | ERROR\_FIND\_FACE | face can not find by faceSetId |

## 示例

### 请求示例

|  |
| --- |
| GET http://192.168.1.214:8080/ds-server/face/v2/getFaceSetDetailInfo?faceSetId=\_default\_face\_set&start=100&count=20 HTTP/1.1  Accept-Encoding: gzip,deflate  Host: 192.168.1.214:8080  Proxy-Connection: Keep-Alive  User-Agent: Apache-HttpClient/4.1.1 (java 1.5) |

### 请求成功响应示例

|  |
| --- |
| HTTP/1.1 200  Transfer-Encoding: chunked  Content-Type: application/json;charset=UTF-8  Date: Tue, 10 Jul 2018 05:57:33 GMT  Keep-Alive: timeout=38  {  "header": {},  "body": {"voFaceSetDetailInfo": {  "faceCount": 10000001,  "faceIds": [  "18f10a4404e8446fb67a4afd02c1ebfa",  "e50efba1b7cf45709e3fcff27fbda86a",  "c127e24d43024429bb03022090e83c37",  "272ad2f30e3849928dd9531c7d849cd2",  "713b1f7f992544c3a6ec81bef9cb8956",  "a3619bdb7aa54a14a3e58a85d53123f2",  "b09a40221e6a4530a95e3041a5132ad2",  "0c97522aaf6d4f75892749dcdde6ad4d",  "8aaa7ca35e1f4778a644336ee577dcaa",  "2ce8912489874e5c96eaf4b0431c0b9c"  ],  "appImgIds": [  null,  null,  null,  null,  null,  null,  null,  null,  null,  null  ],  "next": 110  }}  } |

## 相关资源

None

# 获取人脸库属性接口

## 概述

此接口用于获得指定人脸库属性。

输入：人脸库ID或应用程序定义的人脸库ID

## 调用URL

/ds-server/face/v2/getFaceSetById

## 调用方法

GET

## 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| faceSetId | 字符串 | 人脸库Id | 2选1 |
| appFaceSetId | 字符串 | 应用程序定义的人脸库Id |

## 返回值说明

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| faceSetId | 字符串 | 人脸库Id | 必须（2选1） |
| appFaceSetId | 字符串 | 应用程序定义的人脸库Id |
| tag | 字符串 | 自定义标签，用来对人脸库分组 | No |
| description | 字符串 | 人脸库的描述信息 | No |
| userData | 字符串 | 自定义用户信息 | No |

## 当前API特有的错误信息

|  |  |  |
| --- | --- | --- |
| HTTP Status Code | Error Code | Details |
| 200 | ERROR\_FIND\_FACESET | face can not find by faceSetId |
| 200 | ERROR\_FIND\_FACESET | face can not find by appFaceSetId |

## 示例

### 请求示例

|  |
| --- |
| GET http://192.168.1.214:8080/ds-server/face/v2/getFaceSetById?appFaceSetId=11111 HTTP/1.1  Accept-Encoding: gzip,deflate  Host: 192.168.1.214:8080  Proxy-Connection: Keep-Alive  User-Agent: Apache-HttpClient/4.1.1 (java 1.5) |

### 请求成功响应示例

|  |
| --- |
| HTTP/1.1 200  Transfer-Encoding: chunked  Content-Type: application/json;charset=UTF-8  Date: Tue, 10 Jul 2018 02:45:07 GMT  Keep-Alive: timeout=38  {  "header": {},  "body": {"voFaceSetInfo": {  "faceSetId": "a94621dd058a403ebb3cb161af65a3a8",  "appFaceSetId": "11111",  "tag": "test1",  "userData": "xxxxx1",  "description": "xxxx1"  }}  } |

## 相关资源

None

# 获取人脸库列表接口

## 概述

此接口用于获得人脸库集合。

输入：人脸库的信息起始序号和请求数目

## 调用URL

/ds-server/face/v2/getFaceSetList

## 调用方法

GET

## 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| start | 整型 | 开始索引默认从第一个记录开始 | No |
| count | 整型 | 返回的条目数 | NO |

## 返回值说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | | 类型 | 描述 | 是否必须 |
| voFaceSetInfos | faceSetId | 字符串 | 人脸库Id | 必须（2选1） |
| appFaceSetId | 字符串 | 应用程序定义的人脸库Id |
| tag | 字符串 | 自定义标签，用来对人脸库分组 | No |
| userData | 字符串 | 自定义用户信息 | No |

## 当前API特有的错误信息

|  |  |  |
| --- | --- | --- |
| HTTP Status Code | Error Code | Details |
| 400 | ERROR\_BAD\_PARAM | faceSetCount can not more than 100 or can not less than 1 |

## 示例

### 请求示例

|  |
| --- |
| GET http://192.168.1.78:8080/ds-server/face/v2/getFaceSetList HTTP/1.1  Accept-Encoding: gzip,deflate  Host: 192.168.1.78:8080  Proxy-Connection: Keep-Alive  User-Agent: Apache-HttpClient/4.1.1 (java 1.5) |

### 请求成功响应示例

|  |
| --- |
| HTTP/1.1 200 OK  Transfer-Encoding: chunked  Content-Type: application/json;charset=UTF-8  Date: Tue, 10 Jul 2018 06:07:04 GMT  Keep-Alive: timeout=38  Server: Apache-Coyote/1.1  {  "header": {"count": 6},  "body": {"voFaceSetInfos": [  {  "faceSetId": "1bd13eaf9b284d75bb2729adb8f3a21a",  "appFaceSetId": "11111",  "tag": "test1"  },  {"faceSetId": "\_default\_face\_set"},  {  "faceSetId": "a13aa01c656e40b0b2404b3b5fef9ada",  "appFaceSetId": "55555",  "tag": "test1"  },  {  "faceSetId": "a6dff0d1102f4d50bd9378dd6ebc9347",  "appFaceSetId": "44444",  "tag": "test1"  },  {  "faceSetId": "410ff8149aa64d129784e4a8aad3661e",  "appFaceSetId": "33333",  "tag": "test1"  },  {  "faceSetId": "fc2c372c86a94977a525c24b4e7122a2",  "appFaceSetId": "22222",  "tag": "test1"  }  ]}  } |

## 相关资源

None

# 修改人脸库基本属性接口

## 概述

此接口用于修改人脸库基本属性值。

输入：人脸库的基本属性信息。

## 调用URL

/ds-server/face/v2/modifyFaceSetInfo

## 调用方法

POST

## 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| faceSetId | 字符串 | 人脸库Id | 必须（2选1）  也可都指定 |
| appFaceSetId | 字符串 | 应用程序定义的人脸库Id |
| tag | 字符串 | 自定义标签，用来对人脸库分组 | No |
| userData | 字符串 | 自定义用户信息 | No |
| description | 字符串 | 自定义描述信息 | No |

## 返回值说明

API执行成功，返回通用成功消息。

## 当前API特有的错误信息

|  |  |  |
| --- | --- | --- |
| HTTP Status Code | Error Code | Details |
| 200 | ERROR\_FIND\_FACESET | face can not find by faceSetId |
| 200 | ERROR\_FIND\_FACESET | face can not find by appFaceSetId |

## 示例

### 请求示例

|  |
| --- |
| POST http://192.168.1.214:8080/ds-server/face/v2/modifyFaceSetInfo HTTP/1.1  Accept-Encoding: gzip,deflate  Content-Type: application/json  Content-Length: 117  Host: 192.168.1.214:8080  Proxy-Connection: Keep-Alive  User-Agent: Apache-HttpClient/4.1.1 (java 1.5)  {    "appFaceSetId": "11111",  "tag": "test2",  "userData": "aaa1",  "description": "xxxx1"  } |

### 请求成功响应示例

|  |
| --- |
| HTTP/1.1 200  Transfer-Encoding: chunked  Content-Type: application/json;charset=UTF-8  Date: Tue, 10 Jul 2018 01:35:52 GMT  Keep-Alive: timeout=38  {  "header": {},  "body": {}  } |

## 相关资源

None

# 删除人脸库接口

## 概述

此接口用于删除人脸库。

输入：人脸库的唯一ID或应用程序定义的人脸库ID。

## 调用URL

/ds-server/face/v2/deleteFaceSet

## 调用方法

DELETE

## 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 描述 | 是否必须 |
| faceSetId | 字符串 | 人脸库Id | 2选1 |
| appFaceSetId | 字符串 | 应用程序定义的人脸库Id |

## 返回值说明

API执行成功，返回通用成功消息。

## 当前API特有的错误信息

|  |  |  |
| --- | --- | --- |
| HTTP Status Code | Error Code | Details |
| 200 | ERROR\_FIND\_FACESET | faceSet can not find by faceSetId |
| 200 | ERROR\_FIND\_FACESET | faceSet can not find by faceSetId |

## 示例

### 请求示例

|  |
| --- |
| DELETE http://192.168.1.214:8080/ds-server/face/v2/deleteFaceSet?appFaceSetId=11111 HTTP/1.1  Accept-Encoding: gzip,deflate  Content-Type: application/json  Content-Length: 0  Host: 192.168.1.214:8080  Proxy-Connection: Keep-Alive  User-Agent: Apache-HttpClient/4.1.1 (java 1.5) |

### 请求成功响应示例

|  |
| --- |
| HTTP/1.1 200  Transfer-Encoding: chunked  Content-Type: application/json;charset=UTF-8  Date: Tue, 10 Jul 2018 01:35:52 GMT  Keep-Alive: timeout=38  {  "header": {},  "body": {}  } |

## 相关资源

None