

# Face++ API Python调用示例

V1.1

Updated by Mars Loo.

2015-08-24

## 人脸检测与分析

### /detection/detect

若结果的face\_id没有被加入任何faceset/person之中，则在72小时之后过期被自动清除。

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: detection\_detect\_demo1.py

#This demo show you how to use Face++ API:/detection/detect.

#本示例展示如何使用/detection/detect接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

# Here is the url to the person's face image

# 人脸部图片url

IMAGE\_URL = 'http://cn.faceplusplus.com/static/resources/python\_demo/1.jpg'

# Detect face in the picture and find out his position and attributes

# 检测出输入图片中的Face，找出图片中Face的位置及属性

face = api.detection.detect(url = IMAGE\_URL)

print\_result("sample person's face information:", face)

#### 采用img参数上传图片

原始图片大小需要小于1M，否则会返回1303错误。

face = api.detection.detect(img = facepp.File('1.jpg'))

#### 多张人脸检测

face = api.detection.detect(

img = facepp.File('2.jpg'), #2.jpg是一个有两个人脸的照片

mode = 'oneface'

)

提取一张人脸时，回显信息如下：

sample person's face information:

{'face': [{'attribute': {'age': {'range': 5, 'value': 31},

'gender': {'confidence': 99.993600000000001,

'value': 'Male'},

'race': {'confidence': 85.911799999999999,

'value': 'White'},

'smiling': {'value': 11.779500000000001}},

'face\_id': '478395d726f7149069b221beb71ada03',

'position': {'center': {'x': 46.166666999999997, 'y': 28.5},

'eye\_left': {'x': 43.588000000000001,

'y': 24.114875000000001},

'eye\_right': {'x': 49.033166999999999,

'y': 25.52075},

'height': 18.0,

'mouth\_left': {'x': 43.375833,

'y': 34.578749999999999},

'mouth\_right': {'x': 47.605666999999997,

'y': 35.151000000000003},

'nose': {'x': 46.662166999999997, 'y': 31.227},

'width': 12.0},

'tag': ''}],

'img\_height': 400,

'img\_id': '88b778125f64e40cc16be071b064a0ba',

'img\_width': 600,

'session\_id': 'dc5a04f704f948088076c4e0246505bd',

'url': None}

给mode传递参数为’normal’时，返回信息如下：

sample person's face information:

{'face': [{'attribute': {'age': {'range': 5, 'value': 31},

'gender': {'confidence': 99.993600000000001,

'value': 'Male'},

'race': {'confidence': 85.911799999999999,

'value': 'White'},

'smiling': {'value': 11.779500000000001}},

'face\_id': '7bbaadfa05b5b7f3d3b70be707bca3b4',

'position': {'center': {'x': 46.166666999999997, 'y': 28.5},

'eye\_left': {'x': 43.588000000000001,

'y': 24.114875000000001},

'eye\_right': {'x': 49.033166999999999,

'y': 25.52075},

'height': 18.0,

'mouth\_left': {'x': 43.375833,

'y': 34.578749999999999},

'mouth\_right': {'x': 47.605666999999997,

'y': 35.151000000000003},

'nose': {'x': 46.662166999999997, 'y': 31.227},

'width': 12.0},

'tag': ''},

{'attribute': {'age': {'range': 5, 'value': 28},

'gender': {'confidence': 99.999899999999997,

'value': 'Female'},

'race': {'confidence': 99.975200000000001,

'value': 'White'},

'smiling': {'value': 97.990499999999997}},

'face\_id': 'e662ac06698a75cf310b8a131ee7efe6',

'position': {'center': {'x': 55.666666999999997, 'y': 43.5},

'eye\_left': {'x': 52.986666999999997,

'y': 39.887250000000002},

'eye\_right': {'x': 57.981166999999999,

'y': 39.990749999999998},

'height': 15.5,

'mouth\_left': {'x': 53.033667000000001,

'y': 46.977499999999999},

'mouth\_right': {'x': 58.148667000000003,

'y': 46.936999999999998},

'nose': {'x': 54.736333000000002,

'y': 44.674750000000003},

'width': 10.333333},

'tag': ''}],

'img\_height': 400,

'img\_id': '456efa5e8db67d745e39f0b25934c9c0',

'img\_width': 600,

'session\_id': 'c4db19c72e8243528d810f8518f73ade',

'url': None}

#### 性别、年龄、种族、微笑程度、眼镜、脸部姿势

face = api.detection.detect(

img = facepp.File('1.jpg'),

attribute = "gender,age,race,smiling,glass,pose"

)

回显信息截取如下：

{'face': [{'attribute': {'age': {'range': 7, 'value': 33},

'gender': {'confidence': 99.939999999999998,

'value': 'Male'},

'glass': {'confidence': 99.955799999999996,

'value': 'None'},

'pose': {'pitch\_angle': {'value': -0.0047660000000000003},

'roll\_angle': {'value': -5.9759599999999997},

'yaw\_angle': {'value': -19.743576999999998}},

'race': {'confidence': 99.693899999999999,

'value': 'White'},

'smiling': {'value': 2.7136800000000001}},

#### 关于Tag字段

tag字段可以用在上传图片时加入一些其他信息，返回的人脸信息会原封不动的填写tag字段，tag信息可以通过 /info/get\_face 查询。例如：

face = api.detection.detect(

img = facepp.File('1.jpg'),

tag = 'This is 1.jpg'

)

截取的回显信息如下：

sample person's face information:

{'face': [{'attribute': {'age': {'range': 7, 'value': 33},

'gender': {'confidence': 99.939999999999998,

'value': 'Male'},

'race': {'confidence': 99.693899999999999,

'value': 'White'},

'smiling': {'value': 2.7136800000000001}},

'face\_id': '795e675d775bfb9cb36ac65719fa9544',

'position': {'center': {'x': 42.307692000000003, 'y': 23.5},

'eye\_left': {'x': 37.637557000000001,

'y': 20.637333000000002},

'eye\_right': {'x': 46.111991000000003,

'y': 19.983833000000001},

'height': 14.0,

'mouth\_left': {'x': 39.380316999999998,

'y': 27.826667},

'mouth\_right': {'x': 46.244570000000003,

'y': 27.269333},

'nose': {'x': 41.688009000000001,

'y': 24.508500000000002},

'width': 19.004525000000001},

'tag': 'This is 1.jpg'}],

#### 同步和异步

采用异步方式，Face++会立即返回一个session id，稍后可通过/info/get\_session查询结果。

face = api.detection.detect(

img = facepp.File('1.jpg'),

async = 'true')

返回信息回显如下：

sample person's face information:

{'session\_id': 'c636cbae6db94b53983597bd62de311a'}

利用info/get\_session查询：

res = api.info.get\_session(session\_id = 'c636cbae6db94b53983597bd62de311a')

print\_result("sample person's face information:", res)

回显如下：

sample person's face information:

{'create\_time': 1438125931,

'finish\_time': 1438125939,

'result': {'face': [{'attribute': {'age': {'range': 7, 'value': 33},

'gender': {'confidence': 99.939999999999998,

'value': 'Male'},

'race': {'confidence': 99.693899999999999,

'value': 'White'},

'smiling': {'value': 2.7136800000000001}},

'face\_id': 'daf8a745c989d8ab7cee15196cf910fd',

'position': {'center': {'x': 42.307692000000003,

'y': 23.5},

'eye\_left': {'x': 37.637557000000001,

'y': 20.637333000000002},

'eye\_right': {'x': 46.111991000000003,

'y': 19.983833000000001},

'height': 14.0,

'mouth\_left': {'x': 39.380316999999998,

'y': 27.826667},

'mouth\_right': {'x': 46.244570000000003,

'y': 27.269333},

'nose': {'x': 41.688009000000001,

'y': 24.508500000000002},

'width': 19.004525000000001},

'tag': ''}],

'img\_height': 1220,

'img\_id': 'd5e0f831617a4f6f41e369b6efb8f51d',

'img\_width': 900,

'url': None},

'session\_id': 'c636cbae6db94b53983597bd62de311a',

'status': 'SUCC'}

### /detection/landmark

#### 基本示例

【说明】Face++返回结果中的x、y坐标含义请参考QQ群共享FAQ。

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: detection\_landmark\_demo.py

#This demo show you how to use Face++ API:/detection/landmark.

#本示例展示如何使用/detection/landmark接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

# Here is the url to the person's face image

# 人脸部图片url

IMAGE\_URL = 'http://cn.faceplusplus.com/static/resources/python\_demo/1.jpg'

# Detect face in the picture and find out his position and attributes

# 检测出输入图片中的Face，找出图片中Face的位置及属性

face = api.detection.detect(url = IMAGE\_URL)

face\_id = face['face'][0]['face\_id']

land\_mark = api.detection.landmark(api\_key = API\_KEY, api\_secret = API\_SECRET, face\_id = face\_id)

print\_result("sample person's face information:", land\_mark)

#### 25点模式

Landmark接口默认返回83点模式的结果。

land\_mark = api.detection.landmark(api\_key = API\_KEY, api\_secret = API\_SECRET, face\_id = face\_id, type = '25p')

## 训练模型

### /train/verify

【说明】只要person的信息发生变化，就应该对person进行Train。

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: train\_verify\_demo.py

#This demo show you how to use Face++ API:/train/verify.

#本示例展示如何使用/train/verify接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

# Here is the person's face image

# 人脸部图片

IMAGE = './1.jpg'

# Detect face in the pictures and find out there position and attributes

# 检测出输入图片中的faces，找出图片中faces的位置及属性

face = api.detection.detect(img = facepp.File(IMAGE), mode = 'oneface')

# Get the face\_id.

# 获取face\_id

face\_id = face['face'][0]['face\_id']

person = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face\_id)

print\_result("Person created:", person)

res = api.train.verify(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_name = person['person\_name'])

print "session\_id:",res['session\_id']

time.sleep(5)

res = api.info.get\_session(api\_key = API\_KEY, api\_secret = API\_SECRET, session\_id = res['session\_id'])

print\_result("Train res:", res)

回显信息：

Person created:

{'added\_face': 1,

'added\_group': 0,

'person\_id': '130c8bf077cbfbc39eea51adddbd906a',

'person\_name': '0e54f9efd262410b8533a4f15ba04a5c',

'tag': ''}

session\_id: 2aa5b34deca342cfa24d09f2844205a4

Train res:

{'create\_time': 1440121542,

'finish\_time': 1440121542,

'result': {'success': True},

'session\_id': '2aa5b34deca342cfa24d09f2844205a4',

'status': 'SUCC'}

#### 通过person\_name训练person

res = api.train.verify(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_name = person['person\_name'])

### /train/search

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: train\_search\_demo.py

#This demo show you how to use Face++ API:/train/search.

#本示例展示如何使用/train/search接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

face1 = api.detection.detect(img = facepp.File('./1.jpg'))

face2 = api.detection.detect(img = facepp.File('./2.jpg'))

faceset = api.faceset.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face1['face'][0]['face\_id'] + ',' + face2['face'][0]['face\_id'])

print\_result('Faceset info:', faceset)

res = api.train.search(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_name = faceset['faceset\_name'])

print\_result('After trained:', res)

time.sleep(5)

res2 = api.info.get\_session(api\_key = API\_KEY, api\_secret = API\_SECRET,

session\_id = res['session\_id'])

print\_result('Trained result:', res2)

回显信息：

Faceset info:

{'added\_face': 2,

'faceset\_id': 'e4ab7b629b2577922e1c86106aaef09d',

'faceset\_name': '0a2b1158e1e043d8a5327c0e0abfa715',

'tag': ''}

After trained:

{'session\_id': 'b9d1ef75b0124c829501e9d3f3c0eed6'}

Trained result:

{'create\_time': 1440374502,

'finish\_time': 1440374503,

'result': {'success': True},

'session\_id': 'b9d1ef75b0124c829501e9d3f3c0eed6',

'status': 'SUCC'}

#### 通过指定faceset\_id完成search训练

res = api.train.search(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_id = faceset['faceset\_id'])

### /train/identify

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: train\_identify\_demo.py

#This demo show you how to use Face++ API:/train/identify.

#本示例展示如何使用/train/identify接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

face1 = api.detection.detect(api\_key = API\_KEY, api\_secret = API\_SECRET, img = facepp.File('./ldh.jpg'), tag = 'ldh.jpg')

face2 = api.detection.detect(api\_key = API\_KEY, api\_secret = API\_SECRET, img = facepp.File('./xtf.jpg'), tag = 'xtf.jpg')

#face3 = api.detection.detect(api\_key = API\_KEY, api\_secret = API\_SECRET, img = facepp.File('./xtf2.jpg'), tag = 'xtf2.jpg')

person1 = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET, face\_id = face1['face'][0]['face\_id'])

person2 = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET, face\_id = face2['face'][0]['face\_id'])

group = api.group.create(api\_key = API\_KEY, api\_secret = API\_SECRET, person\_id = person1['person\_id'] + ',' + person2['person\_id'])

print\_result("Group info:", group)

res = api.train.identify(api\_key = API\_KEY, api\_secret = API\_SECRET, group\_name = group['group\_name'])

time.sleep(5)

res2 = api.info.get\_session(api\_key = API\_KEY, api\_secret = API\_SECRET, session\_id = res['session\_id'])

print\_result("Train res:", res2)

回显信息：

Group info:

{'added\_person': 2,

'group\_id': '2cb9c163e0ff9af7425f86540cedd559',

'group\_name': '644e8979ba60444b87afd52634ce426e',

'tag': ''}

Train res:

{'create\_time': 1440363122,

'finish\_time': 1440363123,

'result': {'success': True},

'session\_id': '6f8ec420b6b54dc5943e62f1c35eb80c',

'status': 'SUCC'}

#### 通过指定group\_name训练identify

res = api.train.identify(api\_key = API\_KEY, api\_secret = API\_SECRET, group\_name = group['group\_name'])

## 人脸识别

### /recognition/compare

#### 基本示例

【说明】Face++接口负责计算眉毛、眼睛、鼻子、嘴巴以及整体的一个相似度，具体认为相似度是多少可以认为是同一个人，需要根据不同的业务场景自行制定，Face++不提供一个标准值。

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: recognition\_compare\_demo.py

#This demo show you how to use Face++ API:/recognition/compare.

#本示例展示如何使用/recognition/compare接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

# Here is the person's face image

# 人脸部图片

IMAGE\_ONE = './1.jpg'

IMAGE\_TWO = './2.jpg'

# Detect face in the pictures and find out there position and attributes

# 检测出输入图片中的faces，找出图片中faces的位置及属性

face1 = api.detection.detect(img = facepp.File(IMAGE\_ONE), mode = 'oneface')

face2 = api.detection.detect(img = facepp.File(IMAGE\_TWO), mode = 'oneface')

# Get the face\_ids.

# 获取face\_id

face\_id1 = face1['face'][0]['face\_id']

face\_id2 = face2['face'][0]['face\_id']

res = api.recognition.compare(api\_key = API\_KEY, api\_secret = API\_SECRET, face\_id1 = face\_id1, face\_id2 = face\_id2)

print\_result("Compare result:", res)

#### 同步和异步

采用异步方式，Face++会立即返回一个session id，稍后可通过/info/get\_session查询结果。

res = api.recognition.compare(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id1 = face\_id1, face\_id2 = face\_id2,

async = 'true')

time.sleep(5)

cmp = api.info.get\_session(session\_id = res['session\_id'])

print\_result("compare res:", cmp)

回显如下：

compare res:

{'create\_time': 1438565297,

'finish\_time': 1438565305,

'result': {'component\_similarity': {'eye': 42.200989,

'eyebrow': 70.145065000000002,

'mouth': 71.346474000000001,

'nose': 33.017764999999997},

'similarity': 43.741936000000003},

'session\_id': '8f15f2609e234929ae873abda1c214c1',

'status': 'SUCC'}

### /recognition/verify

#### 基本示例

【说明】在对一个person进行verify之前，必须先对该person进行Train

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: recognition\_verify\_demo.py

#This demo show you how to use Face++ API:/recognition/verify.

#本示例展示如何使用/recognition/verify接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

# Here is the person's face image

# 人脸部图片

IMAGE\_SOURCE = './w1.jpg'

IMAGE\_CHECK = './w2.jpg'

# Detect face in the pictures and find out there position and attributes

# 检测出输入图片中的faces，找出图片中faces的位置及属性

face1 = api.detection.detect(img = facepp.File(IMAGE\_SOURCE), mode = 'oneface')

face2 = api.detection.detect(img = facepp.File(IMAGE\_CHECK), mode = 'oneface')

# Get the face\_id.

# 获取face\_id

face\_id1 = face1['face'][0]['face\_id']

face\_id2 = face2['face'][0]['face\_id']

person = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face\_id1)

print\_result("Person created:", person)

# Start train the person

# 开始训练

res = api.train.verify(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_id = person['person\_id'])

print "session\_id:",res['session\_id']

time.sleep(5)

# Get the train result.

# 获取训练结果

res = api.info.get\_session(api\_key = API\_KEY, api\_secret = API\_SECRET, session\_id = res['session\_id'])

print\_result("Session info:", res)

# Verity if they are the same person.

# 验证是否是同一个人

res = api.recognition.verify(api\_key = API\_KEY, api\_secret = API\_SECRET, face\_id = face\_id2,

person\_id = person['person\_id'])

time.sleep(5)

# Print out the verify result.

# 打印验证结果.

print\_result("Verify res:", res)

回显信息：

Person created:

{'added\_face': 1,

'added\_group': 0,

'person\_id': '9eed0729c3568da40fb77fb1d349a662',

'person\_name': 'bda72b8f78334837bb66b3f284ebaeed',

'tag': ''}

session\_id: 2b39927d3ddb41448389ab575cab8ab4

Session info:

{'create\_time': 1440357299,

'finish\_time': 1440357310,

'result': {'success': True},

'session\_id': '2b39927d3ddb41448389ab575cab8ab4',

'status': 'SUCC'}

Verify res:

{'confidence': 79.591705000000005,

'is\_same\_person': False,

'session\_id': 'a65a9cff4d6d49f8847a6b8024491bdd'}

#### 通过person\_name验证一个人

res = api.recognition.verify(api\_key = API\_KEY, api\_secret = API\_SECRET, face\_id = face\_id2,

person\_name = person['person\_name'])

#### 同步和异步

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: recognition\_verify\_demo.py

#This demo show you how to use Face++ API:/recognition/verify.

#本示例展示如何使用/recognition/verify接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

# Here is the person's face image

# 人脸部图片

IMAGE\_SOURCE = './w1.jpg'

IMAGE\_CHECK = './w2.jpg'

# Detect face in the pictures and find out there position and attributes

# 检测出输入图片中的faces，找出图片中faces的位置及属性

face1 = api.detection.detect(img = facepp.File(IMAGE\_SOURCE), mode = 'oneface')

face2 = api.detection.detect(img = facepp.File(IMAGE\_CHECK), mode = 'oneface')

# Get the face\_id.

# 获取face\_id

face\_id1 = face1['face'][0]['face\_id']

face\_id2 = face2['face'][0]['face\_id']

person = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face\_id1)

print\_result("Person created:", person)

# Start train the person

# 开始训练

res = api.train.verify(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_id = person['person\_id'])

print "session\_id:",res['session\_id']

time.sleep(5)

# Get the train result.

# 获取训练结果

res = api.info.get\_session(api\_key = API\_KEY, api\_secret = API\_SECRET, session\_id = res['session\_id'])

print\_result("Session info:", res)

# Verity if they are the same person.

# 验证是否是同一个人

res = api.recognition.verify(api\_key = API\_KEY, api\_secret = API\_SECRET, face\_id = face\_id2,

person\_id = person['person\_id'],

async = 'true')

time.sleep(5)

# Print out the verify result.

# 打印验证结果.

res = api.info.get\_session(api\_key = API\_KEY, api\_secret = API\_SECRET, session\_id = res['session\_id'])

print\_result("Verify res:", res)

回显信息：

Person created:

{'added\_face': 1,

'added\_group': 0,

'person\_id': 'f5093eb91a013c08eddc779750aa799f',

'person\_name': 'eaa8255acfaf47449c030a0152bf5fd9',

'tag': ''}

session\_id: bead49291b454f489d7716701e419e7d

Session info:

{'create\_time': 1440122341,

'finish\_time': 1440122352,

'result': {'success': True},

'session\_id': 'bead49291b454f489d7716701e419e7d',

'status': 'SUCC'}

Verify res:

{'create\_time': 1440122357,

'finish\_time': 1440122358,

'result': {'confidence': 79.591705000000005, 'is\_same\_person': False},

'session\_id': 'a593cccf615e409dadbc820dc9cd65af',

'status': 'SUCC'}

### /recognition/identify

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: recognition\_identify\_demo.py

#This demo show you how to use Face++ API:/recognition/identify.

#本示例展示如何使用/recognition/identify接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

face1 = api.detection.detect(api\_key = API\_KEY, api\_secret = API\_SECRET, img = facepp.File('./ldh.jpg'))

face2 = api.detection.detect(api\_key = API\_KEY, api\_secret = API\_SECRET, img = facepp.File('./xtf.jpg'))

face3 = api.detection.detect(api\_key = API\_KEY, api\_secret = API\_SECRET, img = facepp.File('./xtf2.jpg'))

person1 = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face1['face'][0]['face\_id'],

tag = 'ldh')

person2 = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face2['face'][0]['face\_id'],

tag = 'xtf')

group = api.group.create(api\_key = API\_KEY, api\_secret = API\_SECRET, person\_id = person1['person\_id'] + ',' + person2['person\_id'])

res = api.train.identify(api\_key = API\_KEY, api\_secret = API\_SECRET, group\_name = group['group\_name'])

time.sleep(5)

res2 = api.recognition.identify(api\_key = API\_KEY, api\_secret = API\_SECRET, group\_id = group['group\_id'], img = facepp.File('./xtf2.jpg'))

print\_result('Identify res:', res2)

#### 通过group\_name完成identify功能

res2 = api.recognition.identify(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_name = group['group\_name'],

img = facepp.File('./xtf2.jpg'))

#### 通过url指定待identify的图片

res2 = api.recognition.identify(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_name = group['group\_name'],

url = 'http://img2.bzcm.net/20150817/86/3685299970570661930.jpg')

#### 在多脸图片中，一次只identify最大的人脸

res2 = api.recognition.identify(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_name = group['group\_name'],

url = 'http://ent.news.cn/2011-05/30/121472107\_11n.jpg',

mode = 'oneface')

#### 通过指定face\_id列表对指定的人脸进行识别

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: recognition\_identify\_demo.py

#This demo show you how to use Face++ API:/recognition/identify.

#本示例展示如何使用/recognition/identify接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

ldh = api.detection.detect(api\_key = API\_KEY, api\_secret = API\_SECRET, img = facepp.File('./ldh.jpg'))

xtf = api.detection.detect(api\_key = API\_KEY, api\_secret = API\_SECRET, img = facepp.File('./xtf.jpg'))

xtf2 = api.detection.detect(api\_key = API\_KEY, api\_secret = API\_SECRET, img = facepp.File('./xtf2.jpg'))

zyj = api.detection.detect(api\_key = API\_KEY, api\_secret = API\_SECRET, img = facepp.File('./zyj.jpg'))

zyj2 = api.detection.detect(api\_key = API\_KEY, api\_secret = API\_SECRET, img = facepp.File('./zyj2.jpg'))

pldh = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = ldh['face'][0]['face\_id'],

tag = 'ldh')

pxtf = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = xtf['face'][0]['face\_id'],

tag = 'xtf')

pzyj = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = zyj['face'][0]['face\_id'],

tag = 'zyj')

fxtf2 = xtf2['face'][0]['face\_id']

print 'xtf faceid:', fxtf2

fzyj2 = zyj2['face'][0]['face\_id']

print 'zyj faceid:', fzyj2

group = api.group.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_id = pldh['person\_id'] + ',' + pxtf['person\_id'] + ',' + pzyj['person\_id'])

res = api.train.identify(api\_key = API\_KEY, api\_secret = API\_SECRET, group\_name = group['group\_name'])

time.sleep(5)

res2 = api.recognition.identify(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_name = group['group\_name'],

key\_face\_id = fxtf2 + ',' + fzyj2)

print\_result('Identify res:', res2)

回显信息：

xtf faceid: 7d5d3fe9b86ddc16a889289d032a56a3

zyj faceid: f1d9f1d392abb5f14255f277c3d461fb

Identify res:

{'face': [{'candidate': [{'confidence': 29.766148000000001,

'person\_id': '282f20396835af51fb99e83d46103372',

'person\_name': '9c68885704c9480e90857376c7a7f9d8',

'tag': 'xtf'},

{'confidence': 9.3815609999999996,

'person\_id': '2bfd5c0e9930426ad1f5a5bc80e699f3',

'person\_name': '3670e991ae884b4f9575acf7f4ce9409',

'tag': 'zyj'},

{'confidence': 2.1163979999999998,

'person\_id': '10b23a077e46ec47159d7e1aa832c5c6',

'person\_name': '31559e4c9281499c9deff4cff7929807',

'tag': 'ldh'}],

'face\_id': '7d5d3fe9b86ddc16a889289d032a56a3',

'position': {'center': {'x': 46.217104999999997,

'y': 57.436709},

'eye\_left': {'x': 36.679605000000002,

'y': 50.319304000000002},

'eye\_right': {'x': 55.113486999999999,

'y': 49.501266000000001},

'height': 38.924050999999999,

'mouth\_left': {'x': 39.408552999999998,

'y': 70.265822999999997},

'mouth\_right': {'x': 53.704934000000002,

'y': 69.558543999999998},

'nose': {'x': 45.047696999999999,

'y': 60.138924000000003},

'width': 40.460526000000002}},

{'candidate': [{'confidence': 23.264923,

'person\_id': '2bfd5c0e9930426ad1f5a5bc80e699f3',

'person\_name': '3670e991ae884b4f9575acf7f4ce9409',

'tag': 'zyj'},

{'confidence': 16.024547999999999,

'person\_id': '10b23a077e46ec47159d7e1aa832c5c6',

'person\_name': '31559e4c9281499c9deff4cff7929807',

'tag': 'ldh'},

{'confidence': 5.1692770000000001,

'person\_id': '282f20396835af51fb99e83d46103372',

'person\_name': '9c68885704c9480e90857376c7a7f9d8',

'tag': 'xtf'}],

'face\_id': 'f1d9f1d392abb5f14255f277c3d461fb',

'position': {'center': {'x': 51.0, 'y': 27.443608999999999},

'eye\_left': {'x': 47.106999999999999,

'y': 24.342444},

'eye\_right': {'x': 53.508749999999999,

'y': 22.761053},

'height': 19.548871999999999,

'mouth\_left': {'x': 48.805500000000002,

'y': 33.306767000000001},

'mouth\_right': {'x': 53.884250000000002,

'y': 32.459173},

'nose': {'x': 50.868250000000003,

'y': 28.890301000000001},

'width': 13.0}}],

'session\_id': 'e7caf410b3054f40829d8191d56160d2'}

#### 同步和异步模式

采用异步模式调用/recognition/identify接口：

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: recognition\_identify\_demo.py

#This demo show you how to use Face++ API:/recognition/identify.

#本示例展示如何使用/recognition/identify接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

ldh = api.detection.detect(api\_key = API\_KEY, api\_secret = API\_SECRET, img = facepp.File('./ldh.jpg'))

xtf = api.detection.detect(api\_key = API\_KEY, api\_secret = API\_SECRET, img = facepp.File('./xtf.jpg'))

xtf2 = api.detection.detect(api\_key = API\_KEY, api\_secret = API\_SECRET, img = facepp.File('./xtf2.jpg'))

zyj = api.detection.detect(api\_key = API\_KEY, api\_secret = API\_SECRET, img = facepp.File('./zyj.jpg'))

zyj2 = api.detection.detect(api\_key = API\_KEY, api\_secret = API\_SECRET, img = facepp.File('./zyj2.jpg'))

pldh = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = ldh['face'][0]['face\_id'],

tag = 'ldh')

pxtf = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = xtf['face'][0]['face\_id'],

tag = 'xtf')

pzyj = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = zyj['face'][0]['face\_id'],

tag = 'zyj')

fxtf2 = xtf2['face'][0]['face\_id']

print 'xtf faceid:', fxtf2

fzyj2 = zyj2['face'][0]['face\_id']

print 'zyj faceid:', fzyj2

group = api.group.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_id = pldh['person\_id'] + ',' + pxtf['person\_id'] + ',' + pzyj['person\_id'])

res = api.train.identify(api\_key = API\_KEY, api\_secret = API\_SECRET, group\_name = group['group\_name'])

time.sleep(5)

res2 = api.recognition.identify(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_name = group['group\_name'],

key\_face\_id = fxtf2,

async = 'true')

time.sleep(5)

res3 = api.info.get\_session(api\_key = API\_KEY, api\_secret = API\_SECRET,

session\_id = res2['session\_id'])

print\_result('Identify result:', res3)

回显信息：

xtf faceid: a495b8b9784086075cec0db9802df5b0

zyj faceid: 13093d3873a6e705395f33442f6307f4

Identify result:

{'create\_time': 1440366633,

'finish\_time': 1440366644,

'result': {'face': [{'candidate': [{'confidence': 29.766148000000001,

'person\_id': 'a8f0382feba339dab1858df4320d738f',

'person\_name': '96f8e077f37e4eddafc1705144128519',

'tag': 'xtf'},

{'confidence': 9.3815609999999996,

'person\_id': '72230f0e97af17c1bf08fbb404d98542',

'person\_name': '41c2f1815d124864b4d4c14212266fc0',

'tag': 'zyj'},

{'confidence': 2.1163979999999998,

'person\_id': '2460aca21d9fa37c7b682ed8ca48f504',

'person\_name': 'cb1b55f3931b4c9e9d994e5db2d06997',

'tag': 'ldh'}],

'face\_id': 'a495b8b9784086075cec0db9802df5b0',

'position': {'center': {'x': 46.217104999999997,

'y': 57.436709},

'eye\_left': {'x': 36.679605000000002,

'y': 50.319304000000002},

'eye\_right': {'x': 55.113486999999999,

'y': 49.501266000000001},

'height': 38.924050999999999,

'mouth\_left': {'x': 39.408552999999998,

'y': 70.265822999999997},

'mouth\_right': {'x': 53.704934000000002,

'y': 69.558543999999998},

'nose': {'x': 45.047696999999999,

'y': 60.138924000000003},

'width': 40.460526000000002}}]},

'session\_id': '531ddf6d4e424d67845bc0f4bc503f72',

'status': 'SUCC'}

### /recognition/search

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: recognition\_search\_demo.py

#This demo show you how to use Face++ API:/recognition/search.

#本示例展示如何使用/recognition/search接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

face1 = api.detection.detect(img = facepp.File('./zyj.jpg'), tag = 'zyj')

face2 = api.detection.detect(img = facepp.File('./zyj2.jpg'), tag = 'zyj')

face3 = api.detection.detect(img = facepp.File('./xtf.jpg'), tag = 'xtf')

face4 = api.detection.detect(img = facepp.File('./xtf2.jpg'), tag = 'xtf')

faceset = api.faceset.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face1['face'][0]['face\_id'] + ',' + face2['face'][0]['face\_id']

+ ',' + face3['face'][0]['face\_id'])

print\_result('Faceset info:', faceset)

res = api.train.search(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_id = faceset['faceset\_id'])

print\_result('After trained:', res)

time.sleep(5)

res2 = api.recognition.search(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_id = faceset['faceset\_id'],

key\_face\_id = face4['face'][0]['face\_id'])

print\_result('Search result:', res2)

回显信息：

Faceset info:

{'added\_face': 3,

'faceset\_id': 'fb03305482bfcd6937e0989fc340f5b6',

'faceset\_name': '2f55698ef7154f99a8b5a920237d6f3c',

'tag': ''}

After trained:

{'session\_id': 'a7517e7c15bc41afa4e0c4b3db34546d'}

Search result:

{'candidate': [{'face\_id': 'c45678687985c22955a00d4100f82150',

'similarity': 81.1173,

'tag': 'xtf'},

{'face\_id': 'b250e4646c1620ec10380f4a20f5a24f',

'similarity': 44.2789,

'tag': 'zyj'},

{'face\_id': '9bbe8d5034e02620d0179f795e685c00',

'similarity': 44.195799999999998,

'tag': 'zyj'}],

'session\_id': '5764fef21b9b4f1691d35477f65655d4'}

#### 在指定了faceset\_name的某个faceset里面搜索某个脸

res2 = api.recognition.search(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_name = faceset['faceset\_name'],

key\_face\_id = face4['face'][0]['face\_id'])

#### 指定返回相似脸的数量

res2 = api.recognition.search(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_name = faceset['faceset\_name'],

key\_face\_id = face4['face'][0]['face\_id'],

count = 1)

#### 同步和异步

res2 = api.recognition.search(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_name = faceset['faceset\_name'],

key\_face\_id = face4['face'][0]['face\_id'],

count = 1,

async = 'true')

time.sleep(5)

res3 = api.info.get\_session(api\_key = API\_KEY, api\_secret = API\_SECRET,

session\_id = res2['session\_id'])

print\_result('Search result:', res3)

## Person管理

### /person/create

一个Person最多允许包含10000个Face，如果超出了，第10001个插不进去。开发版最多允许创建100个person。上线版不做限制。

#### 基本示例

【说明】一个face可以同时属于多个person

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: person\_create\_demo.py

#This demo show you how to use Face++ API:/person/create.

#本示例展示如何使用/person/create接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

person = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET)

print\_result("Person info:", person)

服务器返回信息：

Person info:

{'added\_face': 0,

'added\_group': 0,

'person\_id': 'dcc690d5df73a6822a8c9ea449ce7578',

'person\_name': 'f199a384ced347439f60df4ad6b22195',

'tag': ''}

#### 指定person\_name

person = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_name = 'Peter')

回显信息：

Person info:

{'added\_face': 0,

'added\_group': 0,

'person\_id': '1cd46b5a8da7cbfaf3c4d0344a58ca55',

'person\_name': 'Peter',

'tag': ''}

#### 新建person时指定face

person = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face\_id1 + ',' + face\_id2)

回显信息：

Person info:

{'added\_face': 2,

'added\_group': 0,

'person\_id': 'a8e05bf3dc1ef385313937f99577e535',

'person\_name': 'ebf3987e3635431188e6ee6db44bc6e3',

'tag': ''}

#### 关于tag字段

person = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

tag = 'This is a tag test')

回显信息：

Person info:

{'added\_face': 0,

'added\_group': 0,

'person\_id': '4a6df2391e282e2b56b7d4b334ca4784',

'person\_name': '22e271ef73654306bf032ef8e2934fed',

'tag': 'This is a tag test'}

#### 指定group\_id或者group\_name

【说明】关于group的使用请参考下面的“Group管理“

person = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_id = group1['group\_id'] + ',' + group2['group\_id'])

### /person/delete

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: person\_delete\_demo.py

#This demo show you how to use Face++ API:/person/delete.

#本示例展示如何使用/person/delete接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

person1 = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET)

person2 = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET)

print\_result("Create person1:", person1)

print\_result("Create person2:", person2)

res = api.person.delete(api\_key = API\_KEY, api\_secrect = API\_SECRET, person\_id = person1['person\_id'] + ',' + person2['person\_id'])

print\_result("Delete them:", res)

回显结果：

Create person1:

{'added\_face': 0,

'added\_group': 0,

'person\_id': '7af71383833a987b1ef3f7ed5878b180',

'person\_name': 'b577a7a4671248b4a559a2649133a316',

'tag': ''}

Create person2:

{'added\_face': 0,

'added\_group': 0,

'person\_id': '880cd5851845c16408650d86f733d1a5',

'person\_name': '7e44a303c5724bc6aaf1f299527e2537',

'tag': ''}

Delete them:

{'deleted': 2, 'success': True}

#### 提供person\_name以删除person

res = api.person.delete(api\_key = API\_KEY, api\_secrect = API\_SECRET, person\_name = person1['person\_name']+','+person2['person\_name'])

### /person/add\_face

【注意】一个Face可以被加入到多个Person中，网站上的描述有误，后续会修改网站。

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: person\_add\_face\_demo.py

#This demo show you how to use Face++ API:/person/add\_face.

#本示例展示如何使用/person/add\_face接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

# Here is the person's face image

# 人脸部图片

IMAGE = './1.jpg'

# Detect face in the pictures and find out there position and attributes

# 检测出输入图片中的faces，找出图片中faces的位置及属性

face = api.detection.detect(img = facepp.File(IMAGE), mode = 'oneface')

# Get the face\_ids.

# 获取face\_id

face\_id = face['face'][0]['face\_id']

person1 = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET)

person2 = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET)

print\_result("Create person1:", person1)

print\_result("Create person2:", person2)

res1 = api.person.add\_face(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_name = person1['person\_name'], face\_id = face\_id)

res2 = api.person.add\_face(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_name = person2['person\_name'], face\_id = face\_id)

print\_result("Add res1:", res1)

print\_result("Add res2:", res2)

#### 通过person\_name添加face

res = api.person.add\_face(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_name = person['person\_name'], face\_id = face\_id1 + ',' + face\_id2)

### /person/remove\_face

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: person\_remove\_face\_demo.py

#This demo show you how to use Face++ API:/person/remove\_face.

#本示例展示如何使用/person/remove\_face接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

# Here is the person's face image

# 人脸部图片

IMAGE\_ONE = './1.jpg'

IMAGE\_TWO = './2.jpg'

# Detect face in the pictures and find out there position and attributes

# 检测出输入图片中的faces，找出图片中faces的位置及属性

face1 = api.detection.detect(img = facepp.File(IMAGE\_ONE), mode = 'oneface')

face2 = api.detection.detect(img = facepp.File(IMAGE\_TWO), mode = 'oneface')

# Get the face\_ids.

# 获取face\_id

face\_id1 = face1['face'][0]['face\_id']

face\_id2 = face2['face'][0]['face\_id']

person = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET, face\_id = face\_id1 + ',' + face\_id2)

print\_result("Person created:", person)

res = api.person.remove\_face(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_name = person['person\_name'], face\_id = 'all')

print\_result("Delete res:", res)

回显信息：

Person created:

{'added\_face': 2,

'added\_group': 0,

'person\_id': '7948a79f602919ed51d5b202f07620ca',

'person\_name': 'b01120ba56104cf58c4c45c12273c8c8',

'tag': ''}

Delete res:

{'removed': 2, 'success': True}

#### 通过指定person\_name删除某个face

res = api.person.remove\_face(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_name = person['person\_name'], face\_id = face\_id1)

#### 删除某个person下的所有face

res = api.person.remove\_face(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_name = person['person\_name'], face\_id = 'all')

### /person/set\_info

#### 基本示例

【说明】name和tag是可选参数，如果调用时未指定，返回person原本的name和tag。

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: person\_set\_info\_demo.py

#This demo show you how to use Face++ API:/person/set\_info.

#

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

person = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET)

print\_result("Create person:", person)

res = api.person.set\_info(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_id = person['person\_id'],

name = 'Mars Loo',

tag = 'Mars Loo tag test')

print\_result("Set res:", res)

回显信息：

Person created:

{'added\_face': 2,

'added\_group': 0,

'person\_id': '7948a79f602919ed51d5b202f07620ca',

'person\_name': 'b01120ba56104cf58c4c45c12273c8c8',

'tag': ''}

Delete res:

{'removed': 2, 'success': True}

#### 通过指定person\_name修改person信息

res = api.person.set\_info(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_name = person['person\_ name],

name = 'Mars Loo2',

tag = 'Mars Loo tag test2')

### /person/get\_info

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: person\_get\_info\_demo.py

#This demo show you how to use Face++ API:/person/get\_info.

#本示例展示如何使用/person/ get\_info接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

# Here is the person's face image

# 人脸部图片

IMAGE\_ONE = './1.jpg'

# Detect face in the pictures and find out there position and attributes

# 检测出输入图片中的faces，找出图片中faces的位置及属性

face1 = api.detection.detect(img = facepp.File(IMAGE\_ONE), mode = 'oneface')

# Get the face\_ids.

# 获取face\_id

face\_id1 = face1['face'][0]['face\_id']

person = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET, face\_id = face\_id1)

res = api.person.get\_info(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_id = person['person\_id'])

print\_result("Person info:", res)

回显信息：

Person info:

{'face': [{'face\_id': '6af577199eef8064ab9e10720f119ea9', 'tag': ''}],

'group': [],

'person\_id': '1e92be807086505bfc9c3eabcced1b25',

'person\_name': 'e157cdeb75a544ff9a17c6663dd218dc',

'tag': ''}

#### 通过指定person\_name查询person信息

res = api.person.get\_info(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_name = person['person\_ name])

## FaceSet管理

【注意】一个Faceset最多允许包含10000个Face。开发版最多允许创建5个Faceset。上线版不做限制。

### /faceset/create

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: faceset\_create\_demo.py

#This demo show you how to use Face++ API:/faceset/create.

#本示例展示如何使用/faceset/create接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

face1 = api.detection.detect(img = facepp.File('./1.jpg'))

face2 = api.detection.detect(img = facepp.File('./2.jpg'))

faceset = api.faceset.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face1['face'][0]['face\_id'] + ',' + face2['face'][0]['face\_id'])

print\_result("Faceset created:", faceset)

回显信息：

Faceset created:

{'added\_face': 2,

'faceset\_id': '81922592a29c1c7e4f81ccfc9f266dfb',

'faceset\_name': '0894704b3600485abcbbac815657c9ca',

'tag': ''}

#### 为face\_set指定名字

faceset = api.faceset.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face1['face'][0]['face\_id'] + ',' + face2['face'][0]['face\_id'],

faceset\_name = 'Face set name test.')

#### 为face\_set指定tag字段

faceset = api.faceset.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face1['face'][0]['face\_id'] + ',' + face2['face'][0]['face\_id'],

tag = 'Face set tag test.')

### /faceset/delete

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: faceset\_delete\_demo.py

#This demo show you how to use Face++ API:/faceset/delete.

#本示例展示如何使用/faceset/delete接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

face = api.detection.detect(img = facepp.File('./1.jpg'))

faceset = api.faceset.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face['face'][0]['face\_id'],

faceset\_name = 'Face set name test3.')

print\_result("Faceset created:", faceset)

res = api.faceset.delete(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_id = faceset['faceset\_id'])

print\_result('Faceset delete:', res)

回显信息：

Faceset created:

{'added\_face': 1,

'faceset\_id': '147612bea0430e92b9706f8efe28c195',

'faceset\_name': 'Face set name test3.',

'tag': ''}

Faceset delete:

{'deleted': 1, 'success': True}

#### 通过指定faceset\_name删除faceset

res = api.faceset.delete(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_name = faceset['faceset\_name'])

### /faceset/add\_face

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: faceset\_add\_face\_demo.py

#This demo show you how to use Face++ API:/faceset/add\_face.

#本示例展示如何使用/faceset/add\_face接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

face1 = api.detection.detect(img = facepp.File('./1.jpg'))

face2 = api.detection.detect(img = facepp.File('./2.jpg'))

faceset = api.faceset.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face1['face'][0]['face\_id'],

faceset\_name = 'Face set name test6.')

print\_result("After create:", faceset)

res = api.faceset.add\_face(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_id = faceset['faceset\_id'],

face\_id = face2['face'][0]['face\_id'])

print\_result('After added:', res)

回显信息：

After create:

{'added\_face': 1,

'faceset\_id': '4fb63517a552d67e2bb8759833cff647',

'faceset\_name': 'Face set name test6.',

'tag': ''}

After added:

{'added': 1, 'success': True}

#### 通过指定faceset\_name为faceset添加多个face

res = api.faceset.add\_face(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_name = faceset['faceset\_name'],

face\_id = face1['face'][0]['face\_id'] + ',' + face2['face'][0]['face\_id'])

### /faceset/remove\_face

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: faceset\_remove\_face\_demo.py

#This demo show you how to use Face++ API:/faceset/remove\_face.

#本示例展示如何使用/faceset/remove\_face接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

face1 = api.detection.detect(img = facepp.File('./1.jpg'))

face2 = api.detection.detect(img = facepp.File('./2.jpg'))

faceset = api.faceset.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face1['face'][0]['face\_id'] + ',' + face2['face'][0]['face\_id'])

print\_result("After create:", faceset)

res = api.faceset.remove\_face(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_id = faceset['faceset\_id'],

face\_id = face1['face'][0]['face\_id'])

print\_result('After removed:', res)

#### 通过指定faceset\_name删除某个faceset

res = api.faceset.remove\_face(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_name = faceset['faceset\_name'],

face\_id = face1['face'][0]['face\_id'])

#### 删除某个faceset下面的所有face

res = api.faceset.remove\_face(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_name = faceset['faceset\_name'],

face\_id = 'all')

### /faceset/set\_info

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: faceset\_set\_info\_demo.py

#This demo show you how to use Face++ API:/faceset/set\_info.

#本示例展示如何使用/faceset/set\_info接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

faceset = api.faceset.create(api\_key = API\_KEY, api\_secret = API\_SECRET)

res = api.faceset.set\_info(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_id = faceset['faceset\_id'],

name = 'Faceset new name.')

print\_result('After setted:', res)

回显信息：

After setted:

{'faceset\_id': '2beb00c303cf24ff3652f22eee6c91de',

'faceset\_name': 'Faceset new name.',

'tag': ''}

#### 通过指定faceset\_name为faceset更改tag

res = api.faceset.set\_info(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_name = faceset['faceset\_name'],

tag = 'Faceset new tag2.')

### /faceset/get\_info

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: faceset\_get\_info\_demo.py

#This demo show you how to use Face++ API:/faceset/get\_info.

#本示例展示如何使用/faceset/get\_info接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

face1 = api.detection.detect(img = facepp.File('./1.jpg'))

face2 = api.detection.detect(img = facepp.File('./2.jpg'))

faceset = api.faceset.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face1['face'][0]['face\_id'] + ',' + face2['face'][0]['face\_id'])

res = api.faceset.get\_info(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_id = faceset['faceset\_id'])

print\_result('Faceset info:', res)

回显信息：

Faceset info:

{'face': [{'face\_id': '47a8daa4671146fdf8bfa30931d24254', 'tag': ''},

{'face\_id': 'bf836b70ba0b30e1193a78446197a622', 'tag': ''}],

'faceset\_id': 'b306790f68748d023c6d616217c01f6e',

'faceset\_name': '736f1568211e497e94e316c92ed31c3b',

'tag': ''}

#### 通过指定faceset\_name查询faceset信息

res = api.faceset.get\_info(api\_key = API\_KEY, api\_secret = API\_SECRET,

faceset\_name = faceset['faceset\_name'])

## Group管理

### /group/create

#### 基本示例

【说明】一个Group最多允许包含10000个Person。开发版最多允许创建5个group。上线版不做限制。

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: group\_create\_demo.py

#This demo show you how to use Face++ API:/group/create.

#本示例展示如何使用/group/create接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

group = api.group.create(api\_key = API\_KEY, api\_secret = API\_SECRET)

print\_result("Group created:", group)

回显信息：

Group created:

{'added\_person': 0,

'group\_id': 'e9ea216a7816b457734a0d4e4ec8e992',

'group\_name': 'db8cb8a25e574fa88842d628a47de2ba',

'tag': ''}

#### 为group指定名字

group = api.group.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_name = 'group sample')

#### 关于tag字段

group = api.group.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

tag = 'This is a tag test.')

#### 通过指定person\_name为group添加person

group = api.group.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_name = person['person\_name'])

回显信息：

Group created:

{'added\_person': 1,

'group\_id': '2ce772b8bf3c33f9bc8135edc533d49f',

'group\_name': '6914c01e38024ae0b21c908e72c50ac3',

'tag': ''}

#### 通过指定person\_id为group添加person

group = api.group.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_id = person['person\_id'])

### /group/delete

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: group\_delete\_demo.py

#This demo show you how to use Face++ API:/group/delete.

#本示例展示如何使用/group/delete接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

group = api.group.create(api\_key = API\_KEY, api\_secret = API\_SECRET)

print\_result("Group created:", group)

res = api.group.delete(api\_key = API\_KEY, api\_secret = API\_SECRET, group\_id = group['group\_id'])

print\_result("Group deleted:", res)

回显信息：

Group created:

{'added\_person': 0,

'group\_id': 'f4fbe16cc3a79162c3f97eed60fa7174',

'group\_name': '1d99cfe838e64adca7cd8c14e8db5cd0',

'tag': ''}

Group deleted:

{'deleted': 1, 'success': True}

#### 通过指定group\_name删除一个group

res = api.group.delete(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_name = group['group\_name'])

### /group/add\_person

#### 基本示例

【说明】一个Group最多允许包含10000个Person。

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: group\_add\_person\_demo.py

#This demo show you how to use Face++ API:/group/add\_person.

#本示例展示如何使用/group/add\_person接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

person = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET)

group = api.group.create(api\_key = API\_KEY, api\_secret = API\_SECRET)

print\_result("Group created:", group)

res = api.group.add\_person(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_id = group['group\_id'],

person\_id = person['person\_id'])

print\_result('Person added:', res)

回显信息：

Group created:

{'added\_person': 0,

'group\_id': 'dfa4b317592f5e0af82da450c5be08a5',

'group\_name': '03e5d3bd13104d66a987ccb182af419c',

'tag': ''}

Person added:

{'added': 1, 'success': True}

#### 通过指定group\_name为group添加perosn

res = api.group.add\_person(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_name = group['group\_name'],

person\_id = person['person\_id'])

#### 通过指定person\_name为group添加perosn

res = api.group.add\_person(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_id = group['group\_id'],

person\_name = person['person\_name'])

### /group/remove\_person

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: group\_remove\_person\_demo.py

#This demo show you how to use Face++ API:/group/remove\_person.

#本示例展示如何使用/group/remove\_person接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

person = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET)

group = api.group.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

person\_id = person['person\_id'])

print\_result("Group created:", group)

res = api.group.remove\_person(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_id = group['group\_id'],

person\_id = person['person\_id'])

print\_result('Person deleted:', res)

回显信息：

Group created:

{'added\_person': 1,

'group\_id': '852cbce87600dcd78e03a3b543b73b35',

'group\_name': '31ff324bf1e94988b7825a8b61946adf',

'tag': ''}

Person deleted:

{'removed': 1, 'success': True}

#### 通过指定group\_name从group中删除一个person

res = api.group.remove\_person(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_name = group['group\_name'],

person\_id = person['person\_id'])

#### 通过指定person\_name从group中删除一个person

res = api.group.remove\_person(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_id = group['group\_id'],

person\_name = person['person\_name'])

### /group/set\_info

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: group\_set\_info\_demo.py

#This demo show you how to use Face++ API:/group/set\_info.

#本示例展示如何使用/group/ set\_info接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

group = api.group.create(api\_key = API\_KEY, api\_secret = API\_SECRET)

print\_result("Group created:", group)

res = api.group.set\_info(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_id = group['group\_id'],

tag = 'This is a tag test')

print\_result('Group info setted:', res)

回显信息：

Group created:

{'added\_person': 0,

'group\_id': '9d1e05e1a12f66e52321e2617f578a12',

'group\_name': '82c076d8274d49d786f9d7e6a5f55bc4',

'tag': ''}

Group info setted:

{'group\_id': '9d1e05e1a12f66e52321e2617f578a12',

'group\_name': '82c076d8274d49d786f9d7e6a5f55bc4',

'tag': 'This is a tag test'}

#### 通过group\_name为group指定新的group\_name

Info:这个例子好像有点拗口。

res = api.group.set\_info(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_name = group['group\_name'],

name = 'new name')

### /group/get\_info

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: group\_get\_info\_demo.py

#This demo show you how to use Face++ API:/group/get\_info.

#本示例展示如何使用/group/get\_info接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

person1 = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET)

person2 = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET)

group = api.group.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_name = 'group sample2',

tag = 'This is a tag test.',

person\_id = person1['person\_id'] + ',' + person2['person\_id'])

info = api.group.get\_info(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_id = group['group\_id'])

print\_result("Get group info:", info)

回显信息：

Get group info:

{'group\_id': '35c1cbb786f18e5fa3ae6c142a8996ff',

'group\_name': 'group sample2',

'person': [{'person\_id': 'bdc1ae8f5f8f98deb561dc82c2313ab6',

'person\_name': 'ba9bb6ca54db4af28d030836ed909554',

'tag': ''},

{'person\_id': 'd7a4c27ce8ab609e73798458d5acabb2',

'person\_name': '6ab9bd111bd047d88e697f20b87a522e',

'tag': ''}],

'tag': 'This is a tag test.'}

#### 通过group\_name获取group相关信息

info = api.group.get\_info(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_name = group['group\_name'])

#### 查询所有未加入group的person信息

指定group\_id='none'，此时将返回所有未加入任何Group的Person，注意，是none不是None！

info = api.group.get\_info(api\_key = API\_KEY, api\_secret = API\_SECRET,

group\_id='none')

## 信息查询

### /info/get\_app

已废弃

### /info/get\_session

根据session\_id查询相关信息，使用方法见上。

### /info/get\_person\_list

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: info\_get\_person\_list.py

#This demo show you how to use Face++ API:/info/get\_person\_list.

#本示例展示如何使用/info/get\_person\_list接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

res = api.info.get\_person\_list(api\_key = API\_KEY, api\_secret = API\_SECRET)

print\_result('All persons:', res)

回显信息：

All persons:

{'person': [{'person\_id': '2577ec0e5140619480b68325746599b8',

'person\_name': '42ca16563e8448fbb2f77348aa3c2e81',

'tag': 'xtf'},

{'person\_id': '45923eefa0701e876cae89c7e2baaf95',

'person\_name': '67efce06be814819864a3874256b5c57',

'tag': 'ldh'},

{'person\_id': '3074773af6a45426c1627591fc09dc08',

'person\_name': '70c5c74a591b4e2d8d86ca13a777d3ba',

'tag': 'xtf'},

{'person\_id': '1110004b3a5e5326ebeaddfe607c5343',

'person\_name': 'fd7dff9697f54a7a802d7788749ef014',

'tag': 'xtf'}]}

### /info/get\_faceset\_list

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: info\_get\_faceset\_list.py

#This demo show you how to use Face++ API:/info/get\_faceset\_list.

#本示例展示如何使用/info/get\_faceset\_list接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

res = api.info.get\_faceset\_list(api\_key = API\_KEY, api\_secret = API\_SECRET)

print\_result('All facesets:', res)

回显信息：

All facesets:

{'faceset': [{'faceset\_id': '81922592a29c1c7e4f81ccfc9f266dfb',

'faceset\_name': '0894704b3600485abcbbac815657c9ca',

'tag': ''},

{'faceset\_id': 'e4ab7b629b2577922e1c86106aaef09d',

'faceset\_name': '0a2b1158e1e043d8a5327c0e0abfa715',

'tag': ''},

{'faceset\_id': '245bd295c89ce09ede4743bd8cd7d9ce',

'faceset\_name': '0a9ae0a0dc204e3f9361822545f60b8c',

'tag': ''},

{'faceset\_id': '2beb00c303cf24ff3652f22eee6c91de',

'faceset\_name': 'Faceset new name.',

'tag': ''},

{'faceset\_id': 'e7d7afa839d744a3aa4a21337e424c2c',

'faceset\_name': 'Faceset new name2.',

'tag': ''}]}

### /info/get\_group\_list

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: info\_get\_group\_list.py

#This demo show you how to use Face++ API:/info/get\_group\_list.

#本示例展示如何使用/info/get\_group\_list接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

res = api.info.get\_group\_list(api\_key = API\_KEY, api\_secret = API\_SECRET)

print\_result('All groups:', res)

回显信息：

All groups:

{'group': [{'group\_id': 'f45d2b07caa170d82397dfa34f4d5074',

'group\_name': '16378bc436b64bd98cc03d86a222bc82',

'tag': ''},

{'group\_id': '90064ae881737395aae88b4f7ece580a',

'group\_name': '1d717f1463e944b6a5b6e47e0121f994',

'tag': ''},

{'group\_id': 'd6a78bf6646ea28ebadc38ef573f3e86',

'group\_name': '91711a6dc2da4d6f977ce34c08853652',

'tag': ''},

{'group\_id': '4037e1775dc437be560ecece46b40bc6',

'group\_name': 'a423752f13ea4670a75dd79db175df60',

'tag': ''},

{'group\_id': '9fe0c7cb682a26ff43154170daccba6d',

'group\_name': 'eef35bdb46924781b324a32277206f0a',

'tag': ''}]}

### /info/get\_face

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: info\_get\_face.py

#This demo show you how to use Face++ API:/info/get\_face.

#本示例展示如何使用/info/get\_face接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

# Build a new face

# 建立一个face

face = api.detection.detect(api\_key = API\_SECRET, api\_secret = API\_SECRET,

img = facepp.File('./1.jpg'))

# Create a new person

# 新建一个person

person = api.person.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face['face'][0]['face\_id'])

# Create a new faceset

# 新建一个faceset

faceset = api.faceset.create(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face['face'][0]['face\_id'])

res = api.info.get\_face(api\_key = API\_KEY, api\_secret = API\_SECRET,

face\_id = face['face'][0]['face\_id'])

print\_result('Face info:', res)

回显信息：

Face info:

{'face\_info': [{'attribute': {'age': {'range': 5, 'value': 23},

'gender': {'confidence': 99.635300000000001,

'value': 'Female'},

'race': {'confidence': 99.081199999999995,

'value': 'Asian'},

'smiling': {'value': 98.647499999999994}},

'face\_id': '6e88e32afc8056fd002d0d16b9145988',

'faceset': [{'faceset\_id': 'bc5ca14551d02a7079b0b0118ba5c7ae',

'faceset\_name': 'fcecae0edd2947218a3d65aa0fa2a88b',

'tag': ''}],

'img\_id': '80ed0ab4f0001e40cbf1746284cc87b3',

'person': [{'person\_id': '210b4583d884665a3a898a5e777ef7ff',

'person\_name': '78e01fada06a4b3ebaf5922a24109158',

'tag': ''}],

'position': {'center': {'x': 49.175823999999999,

'y': 28.409091},

'eye\_left': {'x': 37.711812999999999,

'y': 19.973091},

'eye\_right': {'x': 60.694504999999999,

'y': 21.475273000000001},

'height': 35.909090999999997,

'mouth\_left': {'x': 37.332692000000002,

'y': 35.774999999999999},

'mouth\_right': {'x': 58.617581999999999,

'y': 37.878182000000002},

'nose': {'x': 46.062911999999997,

'y': 30.863636},

'width': 43.406593000000001},

'tag': '',

'url': None}]}

### /info/get\_image

#### 基本示例

#!/usr/bin/env python2

# -\*- coding: utf-8 -\*-

# $File: info\_get\_image.py

#This demo show you how to use Face++ API:/info/get\_image.

#本示例展示如何使用/info/get\_image接口

# You need to register your App first, and enter you API key/secret.

# 您需要先注册一个App，并将得到的API key和API secret写在这里。

API\_KEY = 'b579a699d9a1ba6c00da51efd68a8ac7'

API\_SECRET = '-85kMU5hqDFtLoJgXwAtteXmqwx7-vaY'

# Import system libraries and define helper functions

# 导入系统库并定义辅助函数

import time

from pprint import pformat

def print\_result(hint, result):

def encode(obj):

if type(obj) is unicode:

return obj.encode('utf-8')

if type(obj) is dict:

return dict([(encode(k), encode(v)) for (k, v) in obj.iteritems()])

if type(obj) is list:

return [encode(i) for i in obj]

return obj

print hint

result = encode(result)

print '\n'.join([' ' + i for i in pformat(result, width = 75).split('\n')])

# First import the API class from the SDK

# 首先，导入SDK中的API类

import facepp

api = facepp.API(API\_KEY, API\_SECRET)

# Build a new face

# 建立一个face

face = api.detection.detect(api\_key = API\_SECRET, api\_secret = API\_SECRET,

img = facepp.File('./1.jpg'))

res = api.info.get\_image(api\_key = API\_KEY, api\_secret = API\_SECRET,

img\_id = face['img\_id'])

print\_result('Image info:', res)

回显信息：

Image info:

{'face': [{'face\_id': 'c9fd244ecb170ba5c2a5dc43954465fb',

'position': {'center': {'x': 49.175823999999999,

'y': 28.409091},

'eye\_left': {'x': 37.711812999999999,

'y': 19.973091},

'eye\_right': {'x': 60.694504999999999,

'y': 21.475273000000001},

'height': 35.909090999999997,

'mouth\_left': {'x': 37.332692000000002,

'y': 35.774999999999999},

'mouth\_right': {'x': 58.617581999999999,

'y': 37.878182000000002},

'nose': {'x': 46.062911999999997,

'y': 30.863636},

'width': 43.406593000000001},

'tag': ''}],

'img\_id': '922363be3a267795a4cc39d2deeb76d0',

'url': None}

# 附件：源代码包

