**app项目接口说明**

项目后端使用python的flask框架实现。

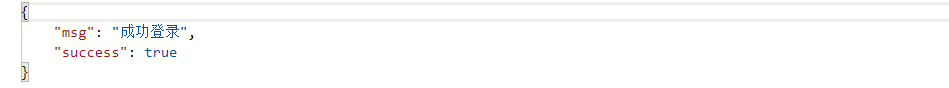
1. /login（已实现）

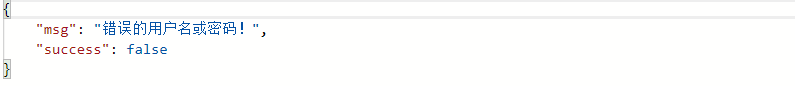
登录功能

参数：

|  |  |
| --- | --- |
| 参数名 | 含义 |
| username | 用户名 |
| password | 密码 |

返回示例：





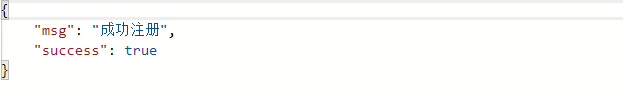
1. /regist（已实现）

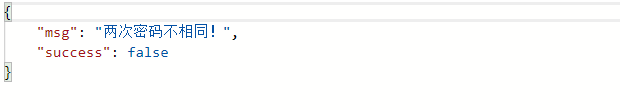
注册功能

参数：

|  |  |
| --- | --- |
| 参数名 | 含义 |
| username | 用户名 |
| Password1 | 第一次输入密码 |
| Password2 | 重复密码 |

返回示例：





1. /face\_upload（已实现）

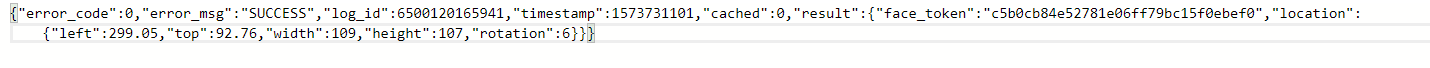
人脸上传，人脸库注册功能

参数：

|  |  |
| --- | --- |
| 参数名 | 含义 |
| image | 图片 |
| Image\_type | 图片类型（BASE64，URL等） |
| group\_id | 用户组id（若没有该用户组则新建该用户组） |
| user\_id | 用户id（若有该用户则直接将图片上传进去，没有则新建该用户并将图片上传进去） |
| action\_type | APPEND: 当user\_id在库中已经存在时，对此user\_id重复注册时，新注册的图片默认会追加到该user\_id下  REPLACE : 当对此user\_id重复注册时,则会用新图替换库中该user\_id下所有图片 |

返回示例：





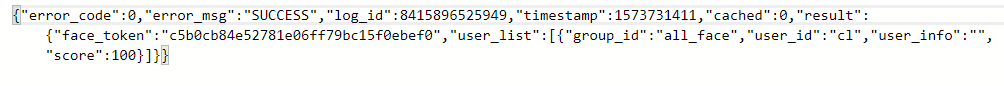
1. /face\_search（已实现）

人脸搜索，人脸对比并返回对比结果

参数：

|  |  |
| --- | --- |
| 参数名 | 含义 |
| image | 图片 |
| Image\_type | 图片类型（BASE64，URL等） |
| group\_id\_list | 用户组id列表表示从哪些用户组中进行搜索 |

返回示例：



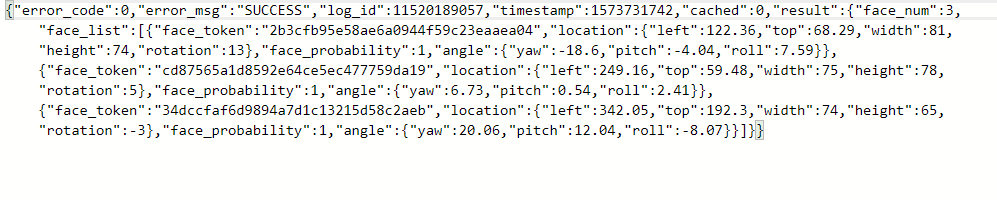
1. /face\_detect（已实现）

检测出图片中所有的人脸

参数：

|  |  |
| --- | --- |
| 参数名 | 含义 |
| image | 图片 |
| Image\_type | 图片类型（BASE64，URL等） |
| max\_face\_num | 最多检测的人脸数量（最大为10）设为n，会返回合照张前n张最大的人脸。 |

返回示例：



1. /logout（已实现）

注销，用户登出

参数：

|  |  |
| --- | --- |
| 参数名 | 含义 |
| 无 |  |

1. /muface\_search（已实现）

实现M：N的多人脸搜索

参数：

|  |  |
| --- | --- |
| 参数名 | 含义 |
| image | 图片 |
| Image\_type | 图片类型（BASE64，URL等） |
| max\_face\_num | 最多检测的人脸数量（最大为10）设为n，会返回合照张前n张最大的人脸。 |
| group\_id\_list | 用户组id列表表示从哪些用户组中进行搜索 |

返回示例：

