

# SKEye-Linux-SDK说明文档V1.0.0

## 1.概述

SKEyeSDK是一套基于SKEye人工智能开放平台的图像服务，该文档将详细说明如何在Linux平台上集成该SDK，并使SKEye人工智能开放平台图像服务。

## 2.系统需求

运行平台：基于Linux平台，该SDK提供简单清晰的 API 接口。

## 3.开发环境

Linux可自行选择，在开发工具里新建工程，将.so文件拷贝至对应目录，具体使用可参考Demo程序。

## 4.SDK包含

- SDK初始化
- 识别接口调用

## 5.Demo说明

Demo工程是Linux工程，展示如何调用SKSDK API接口，网络请求返回值数据会在Demo的界面上会有所展现，详细返回数据会以log形式打印。

## 6.关键参数说明

- api\_key: 当前APP的APP KEY
- api\_secret: 当前APP的APP SECRET
- service\_name: 需要使用的服务组别（请直接使用SDK内置常量进行传参，以避免传错影响接口调用）

## 7.接口说明

### 7.1 SDK初始化

```
void SKEyeSDK_Init(char *api_key, char *api_secret);
```

- 参数：
  - api\_key: 当前APP的APP KEY

- `api_secret`: 当前APP的APP SECRET

- 返回值:

- 无

示例代码:

```
char Api_Key[] = "942f4dea3b45def10552360de80ddasa";
char Api_Secret[] = "282e42c53058b0b08251260cad0746c2";
SKEyeSDK_Init(Api_Key, Api_Secret);
```

## 7.2 调用图像标签识别接口

(1) 识别一个本地图像/URL的标签信息

### ①直接调用

```
//路径
char *SKEyeSDK_ImagePath(char *PATH, char *service_name);
//url
char *SKEyeSDK_ImagePath(char *Image_Url, char *service_name);
```

- 参数:

- `service_name`: 需要使用的服务组别
- `PATH`: 图的绝对路径
- `Image_Url`: URL

- 返回值:

- `char *`类型的json串

示例代码:

```
char PATH[1024] = "object3.jpg";
char service_name[] = "objects";
char Image_Url[] = "http://pic.58pic.com/58pic/12/92/83/39j58PIChF6.jpg";
char *JsonData;
//路径
JsonData=SKEyeSDK_ImagePath(PATH, service_name);
//url
JsonData=SKEyeSDK_ImagePath(Image_Url, service_name);
```

### ② 接口回调

`//Image_Url`、`PATH`等参数同上(直接调用)

```
typedef void (*CallbackJson)(char *JsonData);
//URL回调
void SKEyeSDK_ImagePath(char *Image_Url, char *service_name,
CallbackJson _CallbackJson);
//路径回调
void SKEyeSDK_ImagePath(char *PATH, char *service_name,
CallbackJson _CallbackJson);
```

- 参数:

- service\_name: 需要使用的服务组别
- Image\_Url: URL
- PATH: 图的绝对路径
- CallbackJson: 回调接口

- 返回值:

- char \*类型的json串

示例代码:

```
//Image_Url、PATH等参数同上(直接调用)
//回调函数
void _JsonData(char *JsonData)
{
    printf("%s\n", JsonData);
}
SKEyeSDK_Init(Api_Key, Api_Secret);
//URL回调
SKEyeSDK_ImagePath(Image_Url, service_name, _JsonData);
//路径回调
SKEyeSDK_ImagePath(PATH, service_name, _JsonData);
```

(2) 识别一个BGR、倒立、无行对齐位数据格式图像的标签信息

① 直接调用

```
char *SKEyeSDK_Image(unsigned char *data, unsigned int imageWidth,
unsigned int imageHeight, char *service_name);
```

- 参数:

- data: 图像的unsigned char \*类型图像数据
- imageWidth: 图像宽度
- imageHeight: 图像高度
- service\_name: 需要使用的服务组别

- 返回值:

- char \*类型的json串

- 示例代码:

```

unsigned char *data;
int With,Height;
//ReadBmp函数读取bmp图片,data是BGR、倒立、无行对齐 unsigned char *型数据
if (!ReadBmp("4.bmp", data,With,Height))
{
    printf("Open is error\n");
    return 0;
}
char Api_Key[] = "942f4dea3b45def10552360de80ddasa";
char Api_Secret[] = "282e42c53058b0b08251260cad0746c2";
SEyeSDK_Init(Api_Key, Api_Secret);
char *JsonData = SEyeSDK_Image(data, With, Height, service_name);
  
```

## ②接口回调

```

typedef void (*CallBackJson)(char *JsonData);
void SEyeSDK_Image(unsigned char *data, unsigned int imageWidth,
unsigned int imageHeight, char *service_name, CallBackJson _CallBackJson);
  
```

- 参数：
  - data: 图像的unsigned char \*类型图像数据
  - imageWidth: 图像宽度
  - imageHeight: 图像高度
  - service\_name: 需要使用的服务名称
  - \_CallBackJson: 回调接口
- 返回值：
  - char \*类型的json串
- 示例代码：

```

//相关参数同上(BGR 直接调用)
//回调函数
void _JsonData(char *JsonData)
{
    printf("%s\n", JsonData);
}
unsigned char *data;
int With,Height;
//ReadBmp函数读取bmp图片,data是BGR、倒立、无行对齐 unsigned char *型数据
if (!ReadBmp("4.bmp", data,With,Height))
{
    printf("Open is error\n");
    return;
}
char Api_Key[] = "942f4dea3b45def10552360de80ddasa";
char Api_Secret[] = "282e42c53058b0b08251260cad0746c2";
SEyeSDK_Init(Api_Key, Api_Secret);
  
```

```
//回调
```

```
SKEyeSDK_Image(data, With, Height, service_name, _JsonData);
```

## 8.补充

如果遇到问题，请按以下步骤尝试解决：

- 仔细阅读Demo接口调用源码
- 查看SKEye人工智能开放平台API文档
- 联系我们的工作人员
  - 客服电话：010-62538800
  - 技术支持邮箱：support@interjoy.com.cn
  - 开发平台用户QQ群:617518775