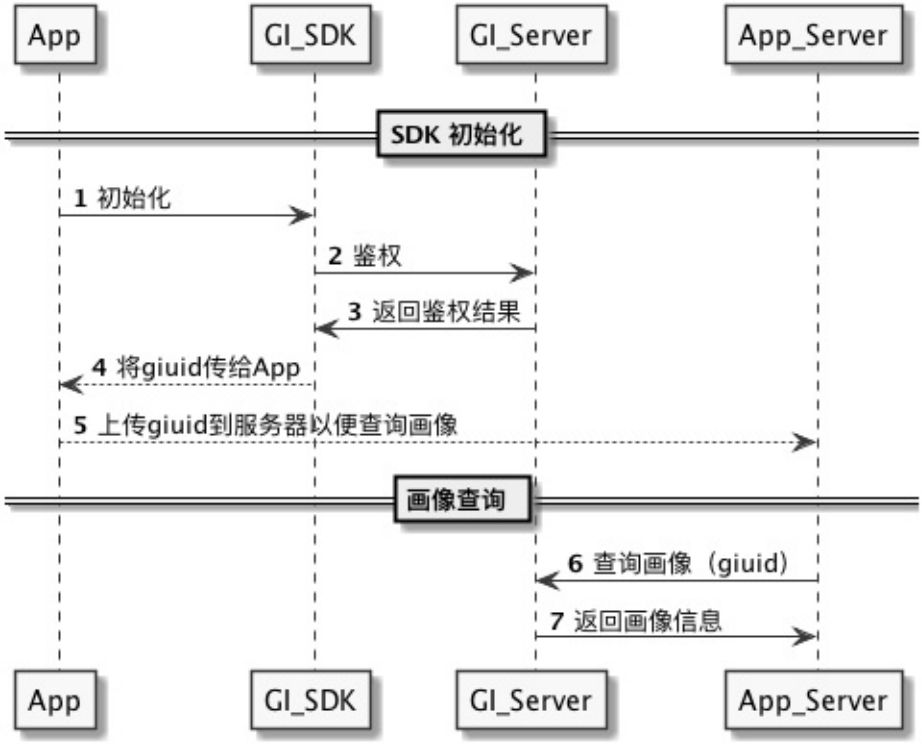
# Table of Contents

# GInsightSDK接入文档

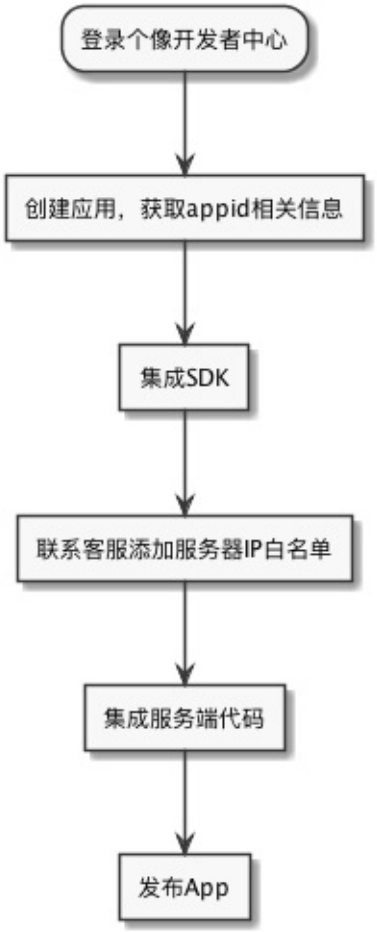
## 概述

该文档旨在指导用户如何接入GInsight SDK，个推GInsight SDK借助个推SDK提供的数据能力，为第三方应用提供了精准用户画像能力，第三方应用利用GInsight SDK提供的接口获取精准的用户画像信息。

## 技术流程




1. 终端APP在程序启动后，调用SDK提供的初始化方法
2. SDK通过GIS SERVER认证用户信息，认证成功后发送GIUID给终端APP；
3. 终端APP将获取到的GIUID上传到APP服务器；
4. APP服务器通过GIUID到GIS SERVER查询对应的用户画像。（详见《服务端集成文档》）




## 1.登录个像开发者中心并创建应用

登录个推开发者中心，进入个像产品页创建应用，获取appid。

> 为了避免个人离职造成应用交接不方便的问题，建议开发者使用企业邮箱或者公共邮箱进行注册。

## 2.集成SDK

按照各平台集成文档正确集成SDK。

Android SDK集成文档

iOS SDK集成文档

## 3.添加服务器IP白名单

画像查询接口会对访问ip进行校验，因此需要联系客服将您服务器的ip地址添加到白名单列表。

## 4.集成服务端代码

按照服务端接口文档进行集成。

# 快速接入文档

Android SDK集成文档

iOS SDK集成文档

服务端接口文档

# GInsight（Android）快速集成文档

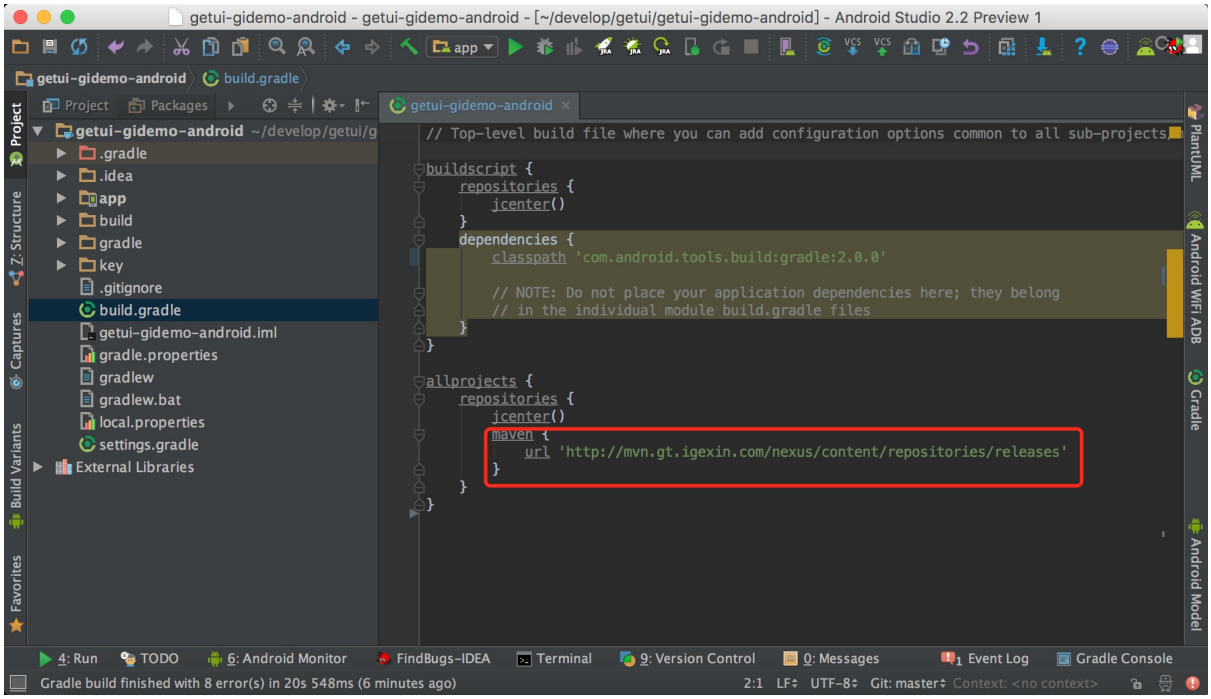
## 获取appid

登录个推开发者中心，进入个像产品页，按照步骤提示注册应用，可获得appid。

> 请确保在个像开发者中心正确配置应用包名以及签名(SHA256)

## 添加maven库地址

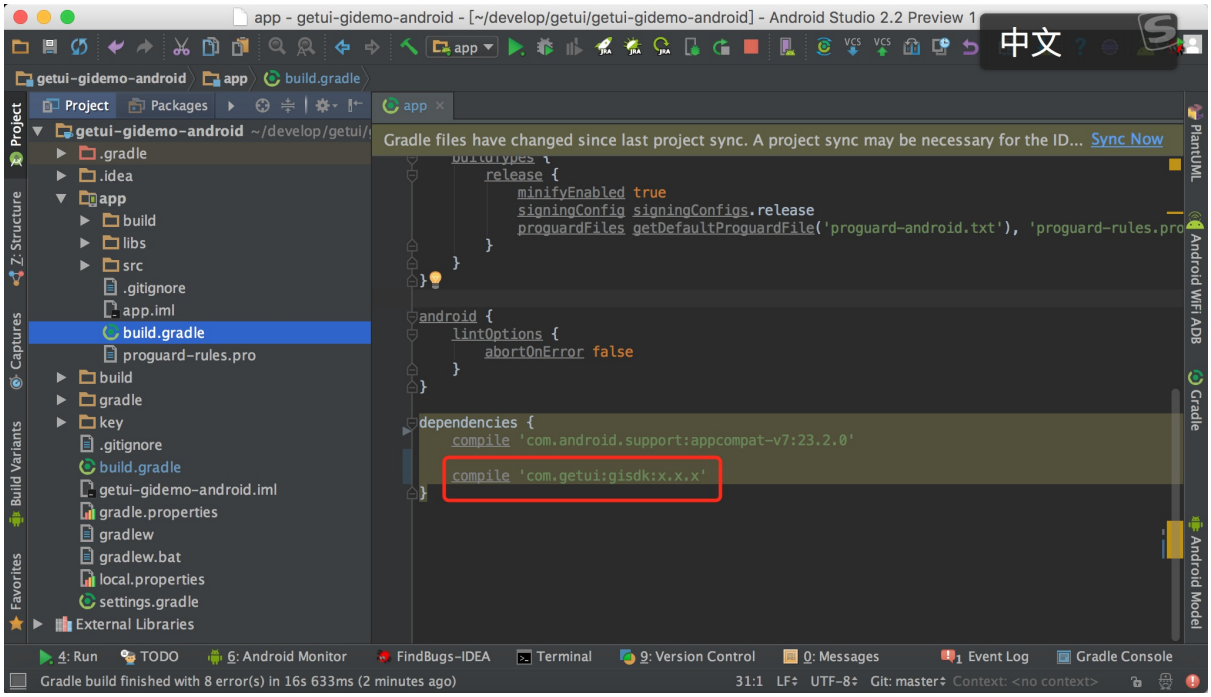
在以项目名为命名的顶层 `build.gradle` 文件中，添加个推maven库地址，如下所示：




```
// Maven URL地址
maven {
    url 'http://mvn.gt.igexin.com/nexus/content/repositories/releases'
}
```

## 配置依赖

在 `app/build.gradle` 文件中引用个像SDK依赖库，如下图所示：




```
dependencies {
    compile 'com.getui:gisdk:3.0.0'
}
```

为了获取SDK启动后的事件响应，还必须实现一个Receiver并在AndroidManifest.xml中配置，示例如下：

```
<receiver
    android:name="com.xxx.GInsightEventReceiver" android:exported="false">
    <intent-filter>
        <action android:name="com.getui.gis.action.您的AppId" /> <!—-替换成GInsight的
APPID ->
    </intent-filter>
</receiver>
```

Receiver代码示例如下：

```
package com.getui.ginsightdemo;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.util.Log;

import com.getui.gis.sdk.GInsightManager;

public class GInsightEventReceiver extends BroadcastReceiver {

    public static final String TAG = GInsightEventReceiver.class.getSimpleName();
```

```
    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getStringExtra("action");
        if (action.equalsIgnoreCase(GInsightManager.ACTION_GIUID_GENERATED)) {
            String giuid = intent.getStringExtra("giuid");
            Log.i(TAG, "giuid = " + giuid);
        }
    }
}
```

# 初始化

在您应用的启动入口（Application的onCreate中）调用SDK的初始化代码，调用方式如下：

```
GInsightManager.getInstance().init (getApplicationContext(), "您的appid");
```

# GInsightManager接口类

SDK功能接口，用于调用GInsight相关功能

## 方法详细资料

### 获取GInsightManager单例对象

```
public static GInsightManager getInstance()
```

**返回：**

```
GInsightManager单例对象
```

### 初始化

```
public void init(Context context, String appid)
```

**参数：**

```
context - application上下文

appid - 您的appid
```

### 获取SDK版本号

```
public String version()
```

**返回：**

```
SDK版本号
```

## 配置混淆

在混淆文件中加入如下配置即可：

```
-dontwarn com.getui.**
-keep class com.getui.**{*;}
```

## 添加权限声明

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.GET_TASKS" />
```

# GInsight（iOS）集成文档

## 获取appid

登陆个推开发者中心，进入个像产品页，按照步骤提示注册应用，可获得appid。

> 请确保在个像开发者中心正确配置BundleId

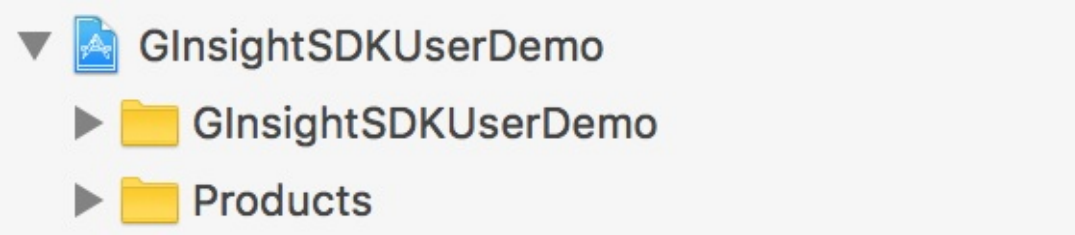
## 获取SDK

前往SDK下载中心下载对应资料包

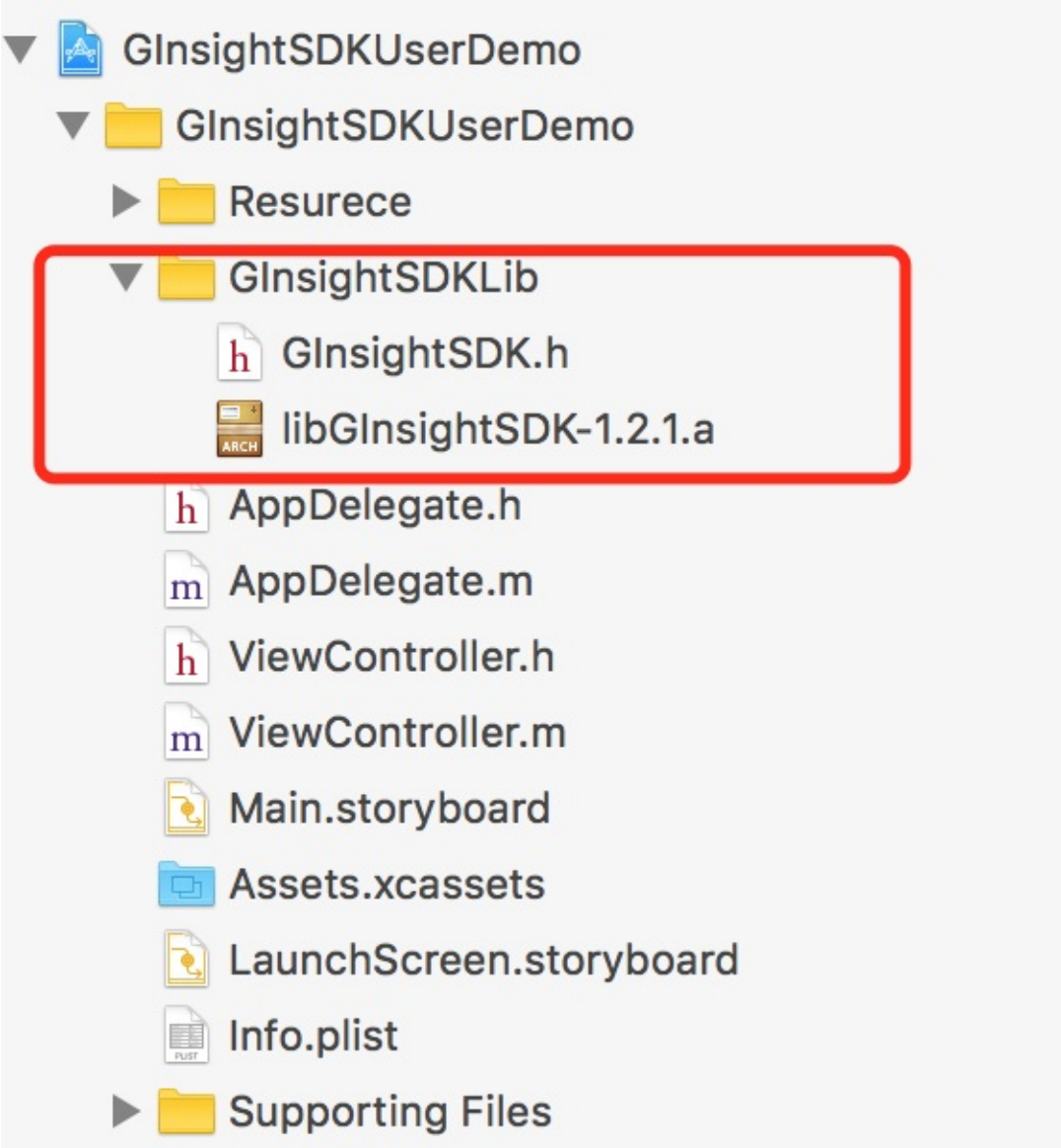
## 开始集成

### 创建项目

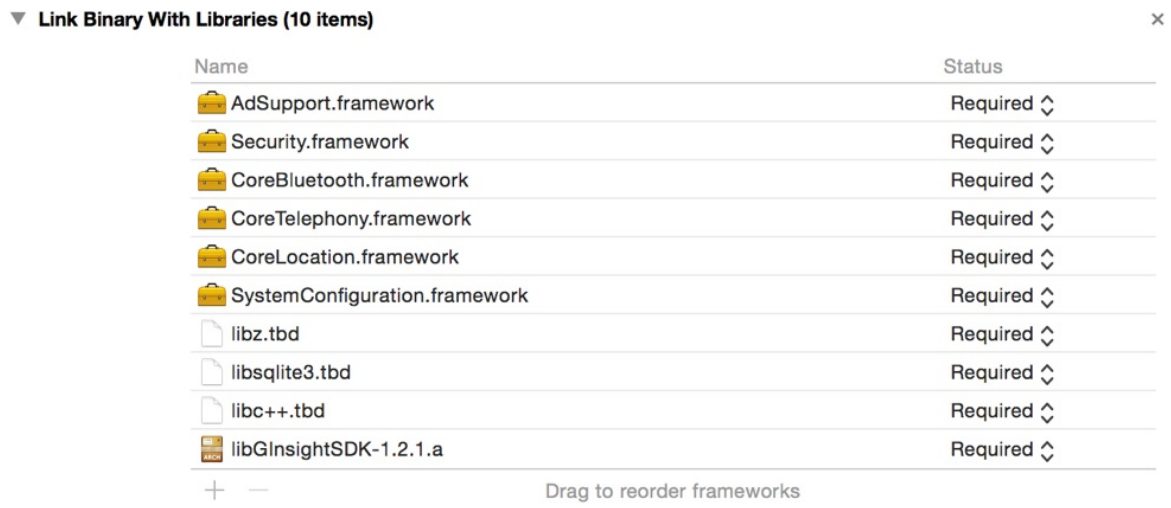
启动Xcode，创建自己的iOS项目工程：



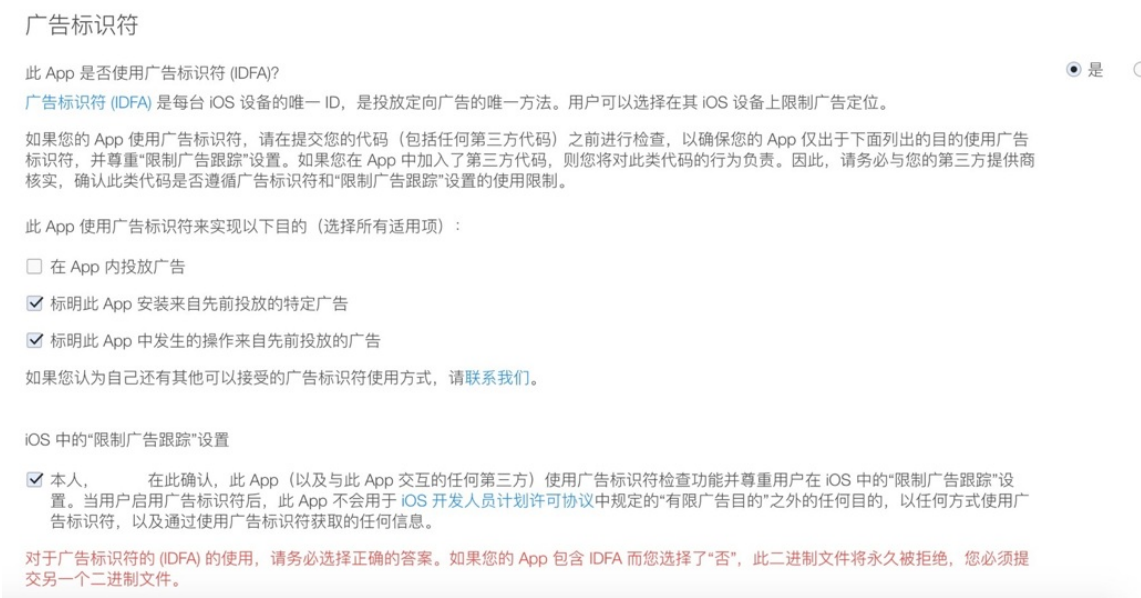

### 导入GInsight SDK和相应Framework




添加需要用到的Framework



> 注意事项：集成GInsight SDK需要使用到IDFA所以必须添加AdSupport.framework,并且为了审核通过需要勾选如图：




## 初始化

在您应用的启动入口application:didFinishLaunchingWithOptions:方法中调用GInsight SDK初始化方法

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSD
ictionary *)launchOptions {
    [GInsightSDK startSDKWithAppId:@"your appid" delegate:self];
    return YES;
}
```

同时实现两个回调；

实现获取GIUid回调委托，获取到GIUid之后上传到服务端用于查询用户画像。

```
- (void)GInsightSDKDidReceiveGiuid:(NSString *)giUid{
    //绑定成功GIUid回调
    //TODO 上传服务器用于查询画像
}
```

实现错误回调委托

```
- (void)GInsightSDKDidReceiveError:(NSError *)error{
    /* 回调错误码类型
     * 1001 APPID 不能为空
     * 1002 IDFA 获取失败
     * 1003 绑定失败
     * 1004 异常错误
     */
```

```
}
```

# SDK接口类说明

SDK功能接口，用于调用GInsight相关功能

## 方法详细资料

### 初始化

```
+ (void)startSDKWithAppId:(NSString *)appId delegate:(id<GInsightSDKDelegate>)deleg
ate;
```

**参数:**

```
appId - 您的appid

delegate - 回调委托对象
```

**示例:**

```
[GInsightSDK startSDKWithAppId:@"sBR4eTRMeD8UMdhmST40n9" delegate:self];
```

### 获取SDK版本号

```
+ (NSString *)version;
```

**返回:**

```
SDK版本号
```

**示例:**

```
NSString *version = [GInsightSDK version];
```

# GInsight服务端接口文档

> 集成服务端接口之前，请先确保已联系客服将服务器出口IP添加到个像IP白名单列表。

## 鉴权接口

### 功能说明

该接口为鉴权接口，在调用获取新画像接口之前，需先调用该接口进行鉴权，只有鉴权 成功的用户，才可以继续调用其他接口。

### URL

https://openapi-gi.getui.com/v2/{appid}/auth_sign

### HTTP请求方式

POST

### 请求参数

| 参数名称 | 参数类型 | 备注 |
| --- | --- | --- |
| sign | String | sha256(appKey+timeStamp+masterSecret) |
| timestamp | Long | 生成sign的时间戳 |

### JSON返回示例

**成功：**

```
{
    "result": true,
    "authtoken": "334a8e7abc96f15a99e9a2e796f7966f2737bee12e407ef534720fdbbccef073"
}
```

**失败：**

```
{
    "error_info": {
        "error_code": 304,
        "error_msg": "IP受限"
    }
}
```

> 鉴权生成的 authtoken 是有过期时间的，若 token 已过期，需重新进行鉴权。

# 画像查询接口

## 功能说明

查询目标用户的画像标签信息

## URL

https://openapi-gi.getui.com/v2/{appid}/query_tag

## HTTP请求方式

POST

## 请求参数

| 参数名称 | 参数类型 | 备注 |
| --- | --- | --- |
| token | String | 鉴权接口返回的authtoken |
| userIdList | List | giuid列表 |

**请求参数JSON示例：**

```
{
    "userIdList": [
        "32e1d2b480sd772a5b16929c2bb4264a",
        "41e1d2b48031772a5b16929c2bb4264b"
    ],
    "token": "334a8e7abc96f15a99e9a2e796f7966f2737bee12e407ef534720fdbbccef073"
}
```

## JSON返回示例

```
{
    "userTag": [
        {
            "error_code": 0,
            "userId": "32e1d2b480sd772a5b16929c2bb4264a",
            "tags": [
                {
                    "code": "010wyq00",
                    "weight": 3
                },
                {
```

```
                    "code": "010wyi00",
                    "weight": 1
                },
                {
                    "code": "010wyq00",
                    "weight": 3
                },
                {
                    "code": "010eq000"
                },
                {
                    "code": "010wq000",
                    "weight": 1
                },
                {
                    "code": "010wyr00",
                    "weight": 2
                },
                {
                    "code": "010iri",
                    "weight": 1
                },
                {
                    "code": "c33000000"
                }
            ]
        },
        {
            "error_msg": "用户暂无画像",
            "error_code": 105,
            "userId": "41e1d2b48031772a5b16929c2bb4264b"
        }
    ]
}
```


> userIdList 最大可同时上传 1000 个 userIdList。
>
> code：画像标签编码（标签编码对照表详见资料包）
>
> weight：画像标签权重（1：高，2：中，3：低），没有weight则表示此标签无画像权重。

## 错误码

| 错误码 | 说明 |
| --- | --- |
| 0 | 成功 |
| 102 | 参数格式错误 |
| 105 | GIUID验证失败 |
| 106 | IP 白名单鉴权失败 |

| | |
|---|---|
| 107 | IP 白名单鉴权异常 |
| 109 | 查询用户画像失败 |
| 112 | 用户暂无画像 |
| 302 | token 已过期 |
| 303 | 获取用户相关信息失败 |
| 304 | IP 受限 |
| 305 | 一分钟内获取画像频次超限 |
| 307 | 一天内获取画像总量超限 |
| 308 | APPID 不匹配 |
| 309 | APPID 和 APPKEY 不匹配 |
| 310 | 画像接口每分钟最大量超限 |
| 311 | 不是老用户，不能调用该接口 |
| 400 | 生成鉴权 token 失败 |
| 401 | 鉴权频次超限 |
| 402 | 鉴权一分钟请求总量超限 |
| 403 | 鉴权异常 |
| 404 | 验证 sign 失败 |
| 500 | 无此接口 |
| 501 | 未传 APPID |
| 502 | 未知应用 |
| 503 | 其他错误 |

# Demo示例

## 获取token

```
public class AuthSignTest {
    static String url = " https://openapi-gi.getui.com/v2/{appid}/auth_sign";

    public static void main(String[] args) throws Exception {
        URL urlObj = new URL(url);
        URLConnection con = urlObj.openConnection();
        HttpURLConnection httpURLConnection = (HttpURLConnection) con;
        // http 头部
        httpURLConnection.setRequestMethod("POST");
        httpURLConnection.setDoOutput(true);
        httpURLConnection.setRequestProperty("Content-Type", "text/html;charset=UTF
```

```
-8");
        Long timestamp = System.currentTimeMillis();

        // 在个像开发者平台获取的 app 对应的 key，可自行替换
        String appkey = "GH4u6fBaYqABoG7bgwAkn5";
        String masterSecret = "iMPZBVYZYPAfyZSnWVsAs";
        // sha256 加密，使用 org.apache.commons 包中自带的加密方法，需将加密后数据一 起上传
        String sign = DigestUtils.sha256Hex(String.format("%s%d%s", appkey, timesta
mp, masterSecret));

        JSONObject requestDataObject = new JSONObject();
        requestDataObject.put("sign", sign);
        requestDataObject.put("timestamp", timestamp);
        // 建立连接，将数据写入内存
        DataOutputStream out = new DataOutputStream(httpURLConnection.getOutputStre
am());
        out.writeBytes(requestDataObject.toString());
        out.flush();
        out.close();

        BufferedReader in = null;
        String result = "";
        // 将数据发送给服务端，并获取返回结果
        in = new BufferedReader(new InputStreamReader(httpURLConnection.getInputStr
eam()));
        String line;
        while ((line = in.readLine()) != null) {
            result += line;
        }
        System.out.println(result);
    }
}
```

## 查询用户画像

```
public class GetUserTagTest {
    static String url = " https://openapi-gi.getui.com/v2/{appid}/query_tag";

    public static void main(String[] args) throws Exception {
        URL urlObj = new URL(url);
        URLConnection con = urlObj.openConnection();
        HttpURLConnection httpURLConnection = (HttpURLConnection) con;
        // http 头部
        httpURLConnection.setRequestMethod("POST");
        httpURLConnection.setDoOutput(true);
        httpURLConnection.setRequestProperty("Content-Type", "text/html;charset=UTF
-8");
        // 需上传的参数，userIdList和token，token为鉴权成功后返回的字符串
        JSONObject requestDataObject = new JSONObject();
```

```
        List<String> userList = new ArrayList<String>();
        userList.add("b154c3fb51c213cbfb25edee3b55d127");
        userList.add("12458758785382947843623 1112");
        requestDataObject.put("token", "cb2b16456ba3943f19ed0b54e8902b53db2aa288a2e
4d56ae88aff93e0f94eef");
        requestDataObject.put("userIdList", userList);

        // 建立连接，将数据写入内存缓冲区
        DataOutputStream out = new DataOutputStream(httpURLConnection.getOutputStre
am());
        out.writeBytes(requestDataObject.toString());
        out.flush();
        out.close();

        // 发送数据给个像服务端，并获得返回结果
        BufferedReader in = null;
        String result = "";
        in = new BufferedReader(new InputStreamReader(
                httpURLConnection.getInputStream()));
        String line;
        while ((line = in.readLine()) != null) {
            result += line;
        }
        System.out.println(result);
    }
}
```

## Android SDK集成文档及Demo下载

**版本：** v3.0.0

下载

## iOS SDK集成文档及Demo下载

**版本：** v1.2.5.1

下载

# 常见问题

**Q1：个像是否收费?**

个像是收费服务，推广期有优惠政策，具体事宜请与技术支持联系（QQ:2880983150）。

**Q2：如果已经集成了个推，查询用户画像是否可以直接用CID?**

个推和个像是两个不同的产品定位，个像有自己的ID为GIUID，所以目前暂不支持用个推的CID进行查询。

**Q3：为什么只有第一次安装可以接收到giuid，重新打开app就接收不到了?**

SDK每隔12小时才会返回一次giuid，开发阶段可以通过卸载重装或者修改手机时间来重新获取giuid。

**Q4：如果没有自己的APP，是否可以获取这些画像?**

个像只能获取自己APP对应的用户的相关画像。

**Q5：画像标签的准确度如何?**

个像的标签数据是采用的个推大数据的标签，准确度有保障且个推的大数据标签是经过多家客户验证过的，可放心使用。

**Q6：如何获取SHA256签名?**

在终端命令行输入keytool -list -v -keystore 您的签名文件，如：keytool -list -v -keystore debug.keystore