

# 数美互联网欺诈用户识别使用手册

北京数美时代科技有限公司提供  
( 版权所有 , 翻版必究 )

## 目录

数美欺诈用户识别使用手册 .....	1
目录 .....	2
1. 接入前准备 .....	4
1.1 申请数美账号 .....	4
1.2 接收数美账号信息 .....	4
2. 接入说明 .....	5
2.1 整体接入流程 .....	5
2.2 嵌入设备指纹 SDK .....	6
2.2.1 Android SDK 接入说明 .....	6
2.2.2 IOS SDK 接入说明 .....	9
2.2.3 WEB SDK 使用说明 .....	12
2.3 用户注册 .....	13
2.3.1 客户端程序 .....	13
2.3.2 客户服务端程序 .....	13
2.3.3 数美机器注册识别服务 API .....	13
2.4 用户登录 .....	17
2.4.1 客户端程序 .....	17
2.4.2 客户服务端程序 .....	18
2.4.3 数美机器登录识别服务 API .....	18
2.5 营销活动反作弊 .....	22
2.5.1 客户端程序 .....	22
2.5.2 客户服务端程序 .....	22
2.5.3 数美营销活动反作弊服务 API .....	22
3. 接入 demo .....	27
4. 数美管理后台 .....	27
4.1 登录管理界面 .....	27
4.2 体验功能 .....	28
4.2.1 趋势分析（注册&登录） .....	28
4.2.2 历史记录（注册&登录） .....	28
4.2.3 设备信息查看（设备指纹） .....	28
4.2.4 更多 .....	29
5. FAQ .....	29
5.1 数美 SDK .....	29
5.1.1 JS SDK 支持的最低浏览器版本？ .....	29
5.1.2 JS SDK 为什么会访问 fp.ws flash 的文件？ .....	29
5.1.3 数美 JS SDK 能否可以绑定到客户的域下？ .....	29
5.1.4 JS SDK 通过异步方式加载进来，网络比较慢的情况下，是否会出现无法生成设备唯一 ID 的情况？ .....	30
5.2 接口调用 .....	30

- 5.2.1 调用接口返回参数错误（1902） ..... 30
- 5.2.2 调用接口返回无权限操作（9101） ..... 30
- 5.2.3 调用接口超时问题 ..... 30
- 5.3 更多问题 ..... 31
  - 5.3.1 电邮 ..... 31
  - 5.4.2 运营电话 ..... 31

# 1. 接入前准备

## 1.1 申请数美账号

向数美服务邮箱 ( api@ishumei.com ) 发送如下内容：

```
公司名称：xxxx
接口人邮箱：xxx@xxx
接口人手机：1xxxxxxx
```

其中：

接口人邮箱会作为数美账号标识，用于登陆数美后台管理界面。

## 1.2 接收数美账号信息

1个工作日内，数美工作人员会审核创建相应数美账号，随后接口人邮箱会收到如下邮件：

名称	具体值	说明
accessKey	xxxxxxxxx	数美 API 服务的认证码，调用数美 API 时需要传入
organization	xxxxxxxxxxx	数美分配的企业唯一标识码，调用 SDK 时需要传入
数美管理后台账号	xxxx	用于登陆数美管理后台
数美管理后台密码	xxxxx	用于登陆数美管理后台
数美管理后台地址	https://www.fengkongcloud.com	用于登陆数美管理后台

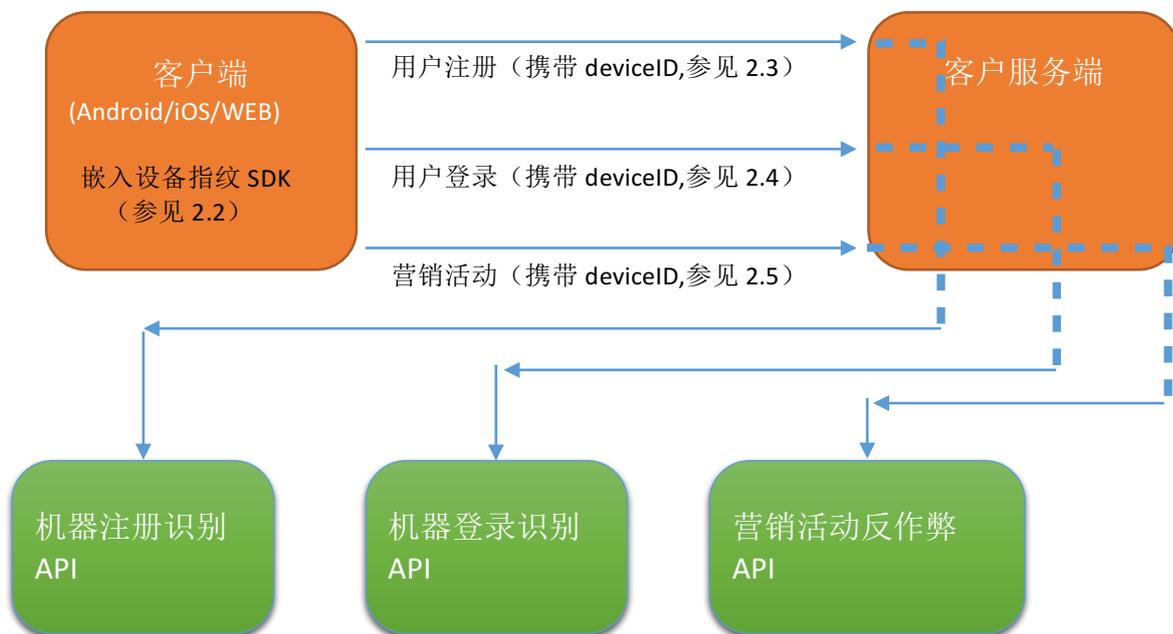
其中：

- accessKey 在后续 2.3、2.4、2.5 节使用
- organization 在后续 2.2 节-“嵌入设备指纹 SDK”会使用
- 数美管理后台功能界面展示参见第 4 章

## 2. 接入说明

### 2.1 整体接入流程

欺诈用户识别接入分为嵌入设备指纹 SDK 和 API 服务接口两部分，整体接入流



程如下所示：

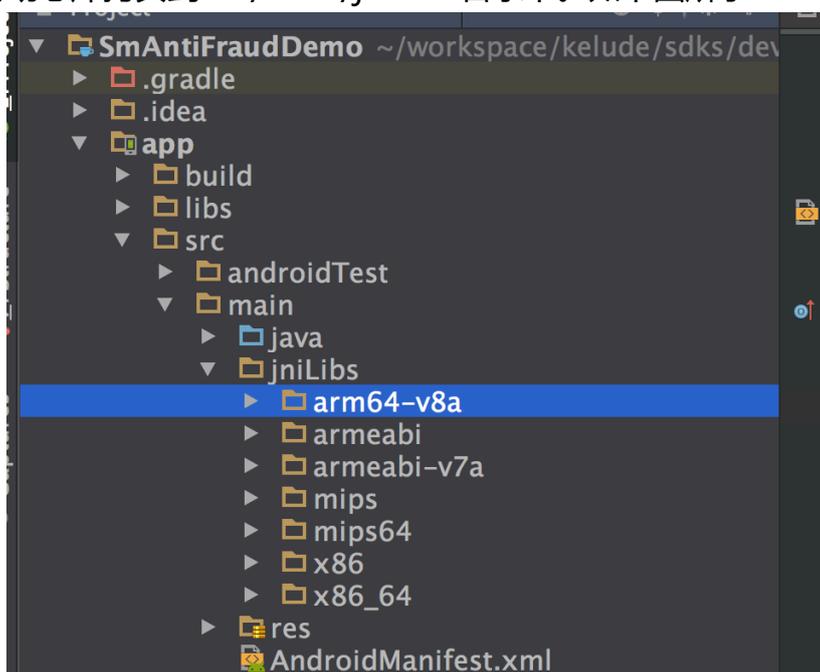
具体的接入步骤如下。

## 2.2 嵌入设备指纹 SDK

### 2.2.1 Android SDK 接入说明

#### 1. 导入 SDK 使用动态库

- 在 Module 的 src/main 下新建 jniLibs 目录(如果有此目录则忽略)。
- 将动态库拷贝到 src/main/jniLibs 目录下。如下图所示:



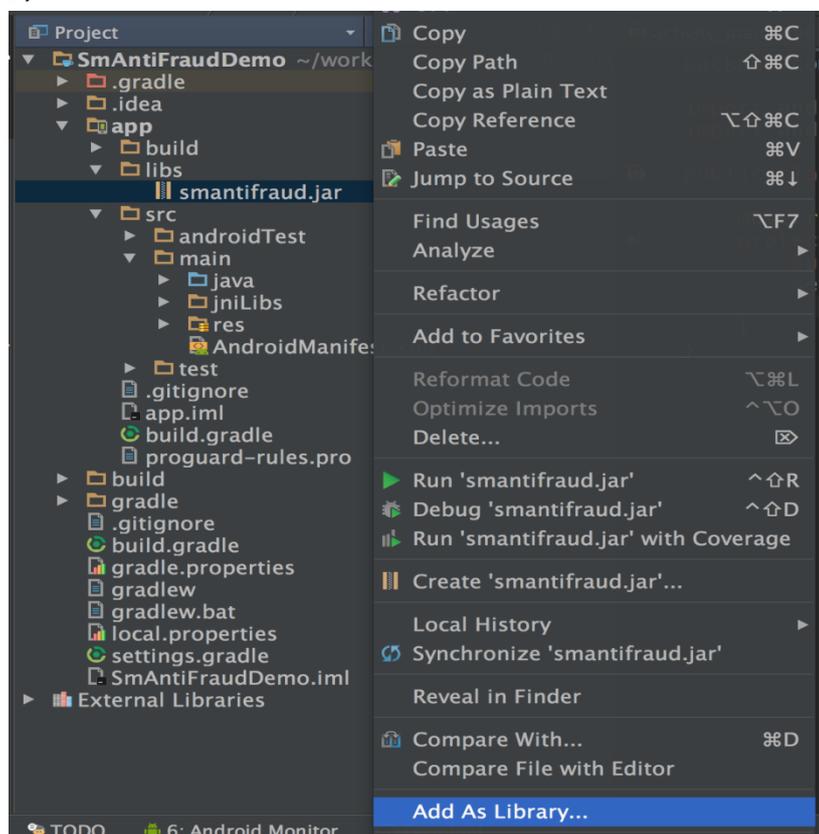
**注意**：build.gradle 中 android 域中如果设置了 sourceSets 的 jniLibs 路径，如下：

```
sourceSets {
    main {
        jniLibs.srcDirs = ['libs']
    }
}
```

请将 so 库拷贝到已设定的目录下。

## 2. 导入 jar 包

- 将 smantifraud.jar 拷贝到 Module 的 src/libs 目录下，并且将 smantifraud.jar 增加 lib 库中。到如下图所示(在 smantifraud.jar 上右击，选择 Add As Library.. 即可)：



## 3. 添加权限

- 在 AndroidManifest.xml 中添加如下权限：

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_SETTINGS"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.BLUETOOTH" />
```

#### 4. 程序启动时，初始化 SDK。代码如下所示：

```
public class MyApplication extends Application {
    private static final String TAG = "sdk demo";

    /**
     * 取得当前进程名
     * @param context
     * @return
     */
    String getCurProcessName(Context context) {
        int pid = android.os.Process.myPid();
        ActivityManager mActivityManager = (ActivityManager) context
            .getSystemService(Context.ACTIVITY_SERVICE);
        for (ActivityManager.RunningAppProcessInfo appProcess : mActivityManager
            .getRunningAppProcesses()) {
            if (appProcess.pid == pid) {
                return appProcess.processName;
            }
        }
        return null;
    }

    @Override
    public void onCreate() {
        super.onCreate();

        // 如果 AndroidManifest.xml 中没有指定主进程名字，主进程名默认与 packagename 相同
        if (getCurProcessName(this).equals(this.getPackageName())) {
            SmAntiFraud.SmOption option = new SmAntiFraud.SmOption();
            String DEBUG_ORG = "ORGXXXXXX";// organization 代码 不要传 AccessKey
            option.setOrganization(DEBUG_ORG);
            option.setChannel("appstorexxx");//渠道代码

            // 可选的方式，deviceId 拉取成功的事件监听，异步方式
            SmAntiFraud.registerServerIdCallback(
                new SmAntiFraud.IServerSmidCallback() {
                    @Override
                    public void onReceive(String serverId, int code) {
                        Log.i(TAG, "deviceId is " + serverId);
                    }
                }
            );

            SmAntiFraud.create(this, option);
            // 注意！！获取 deviceId，这个接口在真正的注册或登录事件产生的地方调用。
            // create 后马上调用返回的是本地的 deviceId，
            // 本地 deviceId 和服务器同步需要一点时间。
            String deviceId = SmAntiFraud.getDeviceId();
        }
    }
}
```

#### 注意：

SmAntiFraud.getDeviceId(); 接口在真正需要 DeviceId 时再进行调用。  
不要再 create 后立即调用。也不要缓存调用的结果，getDeviceId 在 sdk 内部会  
做缓存和更新处理。（ create 后立即调用会返回本地生成的没有和 server 同步的 ID ）

5. 生命周期控制：

注意接入的时候保证 SmAntiFraud.create 只在主进程调用一次。所以需要在调用前判断当前进程是否是主进程而不是其他 如 :push 或 :ipc 等后台服务进程。防止重复调用。

6. java 代码混淆：

当项目接入 sdk 后，项目可能要求混淆自身代码，同时可能会混淆第三方 jar 中的 class，这里需要注意的是不要混淆 sdk 的 jni 接口类，防止找不到函数，proguard-rules.pro 混淆规则文件中添加如下规则：

```
-keep class com.ishumei.dfp.SMSDK { *; }
```

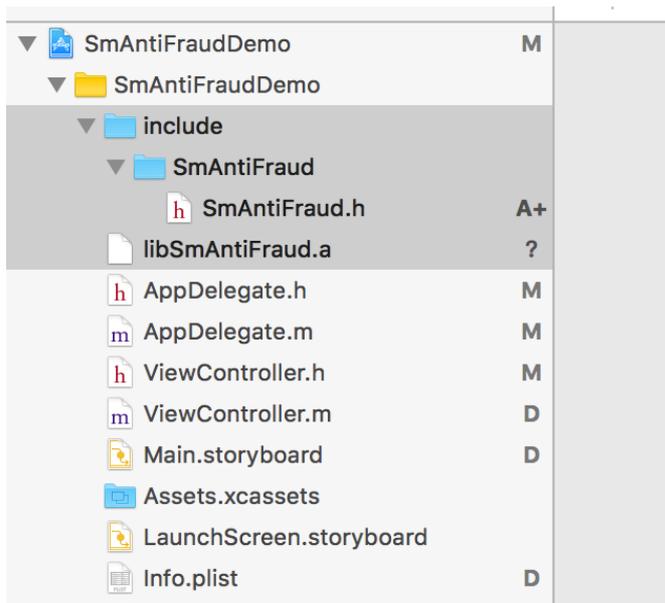
7. 检查接入是否成功

在数美管理后台可查看 SDK 上传的数据，通过检查上传数据，可确认接入是否成功。查看方法参见 4.2.3。

## 2.2.2 IOS SDK 接入说明

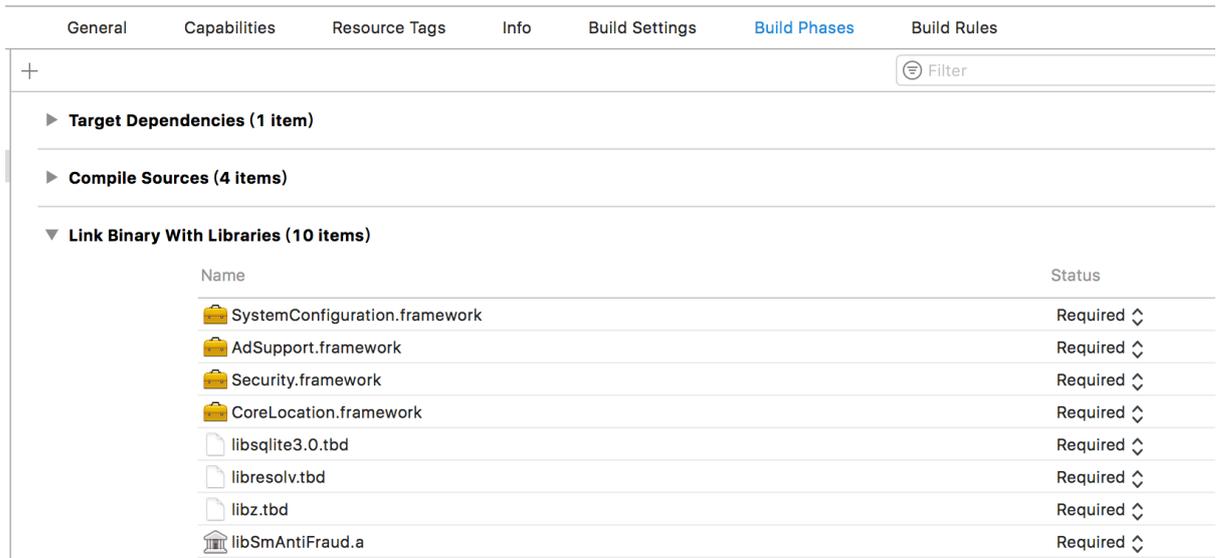
1. 导入静态库

在工程下导入 include 文件夹和 libsmantifraud.a 静态库，如下图所示：

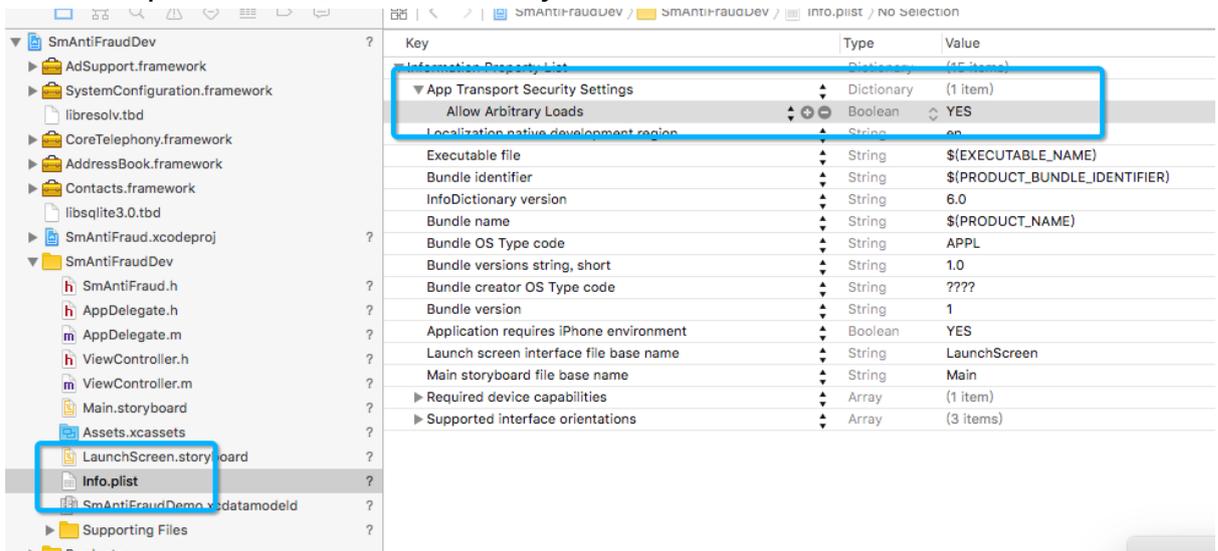


## 2. 添加 SDK 依赖

在 target 下导入如下依赖库：SystemConfiguration.framework、Security.framework、AdSupport.framework、libresolv.tbd、libz.tbd、libsqlite3.0.tbd、CoreMotion.framework、libsmantifraud.a 静态库。如下图所示：



## 3. 修改 Info.plist，新增[Allow Arbitrary Loads]配置。如下图所示：



#### 4. 程序启动时，初始化 SDK，代码如下所示：

```
SmOption *option = [[SmOption alloc] init];
[option setOrganization:@"xxxxxxxxxxxxx"];//传入 organization，不要传入 accessKey.
[option setChannel:@"xxxxxxxxxxxxx"];// 传入渠道标识

// 可选的异步处理，设置回调监听 拉取服务端 deviceId 的事件，
// 本地 deviceId 与服务器端 deviceId 同步是需要一点时间的
[option setCallback:^(NSString* serverId) {
    NSLog(@"server id:%@, %d", serverId, n);
}];

[[SmAntiFraud sharedInstance] create:option];
// 不要缓存 deviceId，在真正注册或登录事件发生时调用下面接口获得 deviceId
// 因为本地 deviceId 与服务器端 deviceId 同步是需要一点时间的
NSString* deviceId = [[SmAntiFraud sharedInstance] getDeviceId];
```

#### 注意：

getDeviceId()接口在真正需要 DeviceId 时再进行调用。

不要再 create 后立即调用。也不要缓存调用的结果，getDeviceId 在 sdk 内部会做缓存和更新处理。（create 后立即调用会返回本地生成的没有和 server 同步的 ID）

#### 5. 检查接入是否成功

在数美管理后台可查看 SDK 上传的数据，通过检查上传数据，可确认接入是否成功。查看方法参见 4.2.3。

## 2.2.3 WEB SDK 使用说明

为避免浪费双方资源，只需要在注册、登录、其他使用数美服务的页面嵌入

SDK 即可。其它页面不用嵌入。

1. 初始化，在页面底部引入如下所示代码：

```
(function() {  
  window._smConf = {  
    organization: 'xxxxxx', //传入 organization，不要传入 accessKey.  
    staticHost: 'static.fengkongcloud.com'  
  };  
  
  var url = (function () {  
    var originHost = "static2.fengkongcloud.com";  
    var isHttps = 'https:' === document.location.protocol;  
    var protocol = isHttps ? 'https://' : 'http://';  
    var fpJsPath = '/fpv2.js';  
    var url = protocol + _smConf.staticHost + fpJsPath;  
    var ua = navigator.userAgent.toLowerCase();  
    var isWinXP = /windows\s(?:nt\s5.1)(?:xp)/.test(ua);  
    var isLowIE = /msie\s[678]\.0/.test(ua);  
  
    if(isHttps && isWinXP && isLowIE) {  
      url = protocol + originHost + fpJsPath;  
    }  
  
    return url;  
  })();  
  var sm = document.createElement("script");  
  var s = document.getElementsByTagName("script")[0];  
  sm.src = url;  
  s.parentNode.insertBefore(sm, s);  
})();
```

## 2. 检查接入是否成功

在数美管理后台可查看 SDK 上传的数据，通过检查上传数据，可确认接入是否成功。查看方法参见 4.2.3。

## 2.3 用户注册

### 2.3.1 客户端程序

#### 1. 首先，获取 deviceId

##### > Android

```
String deviceId = SmAntiFraud.getDeviceId();
```

##### > IOS

```
NSString *deviceId = [[SmAntiFraud sharedInstance] getDeviceId];
```

##### > Web

```
var deviceId = SMSdk.getDeviceId();
```

#### 2. 然后，调用客户服务端注册接口

在原有注册参数（用户名、手机号等）的基础上，加入 deviceId 参数。

### 2.3.2 客户服务端程序

执行注册动作前，按照 2.3.3 的说明，调用数美机器注册识别服务 API；

### 2.3.3 数美机器注册识别服务 API

#### 请求 URL

<http://antirobot-api.fengkongcloud.com/v2/saas/register>

#### 字符编码

请求及返回结果都使用 UTF-8 字符集进行编码

#### 请求方法

POST

**建议超时时长：**

1s

**请求参数**

放在 HTTP Body 中，采用 Json 格式，具体参数如下：

字段	类型	说明	是否必须存在
accessKey	string	用于权限认证，由数美提供。获取方式参见 1.2 节。	是
appId	string	应用 id，用于区分相同公司的不同应用。内容自定义，支持 2-32 位字符。例如：“com.xxx.xxxx”	是
data	json_object	注册信息	是

其中，data 的内容如下：

字段	类型	说明	是否必须存在
registerTime	int	注册发生时间（毫秒时间戳）	是
tokenId	string	注册用户账号 id，可用于后续账号行为的比对，以检验机器注册识别效果。因此，建议客户先为新注册用户生成账号 id，再把该账号 id 传递给数美的机器注册识别服务。	是
signupPlatform	string	第三方登录提供商，可选值：qq,weibo,weixin,alipay,taobao,phone,facebook, twitter 等	是
ip	string	客户注册时的 ip	是

deviceId	string	数美设备指纹标识，由于用户行为分析。相比 mac、imei 等，数美设备指纹更难被篡改，当恶意用户篡改 mac、imei 等设备信息时，使用 deviceId 能够发现和识别此类恶意行为，同时，可用于比对数美设备黑名单，因此建议传递。	是
accountName	string	注册账号的用户名	是
phone	string	注册账号的手机号	两者不能同时为空
phoneMd5	string	注册账号手机号的 md5 值，md5 前需要格式成 11 位格式，例如： 18510971881	
channel	string	渠道信息	否
inviteTokenId	string	该注册用户的邀请者账号 id	否

### 返回参数

放在 HTTP Body 中，采用 Json 格式，具体参数如下：

字段	类型	说明	是否必须存在	可选值
code	int	返回码	是	
message	string	详细描述	是	
score	int	风险分数，[ 0,1000 ] 整数，分数越高风险越大（code 为 1100 时存在）	否	[0,1000]

riskLevel	string	风险级别 ( code 为 1100 时存在 )	否	PASS、REJECT、REVIEW，其中 PASS 为系统认定没有问题的，建议直接放行，REJECT 为系统认定有问题的，建议直接拦截，REVIEW 为系统不确定的，建议人工审核
detail	json_object	风险详情 ( code 为 1100 时存在 )	否	

其中，detail 的格式如下：

字段	类型	说明	是否必须存在
description	string	风险描述	是
model	string	模型编号	是

code 和 message 的列表如下：

code	message
1100	成功
1902	参数不合法
1903	服务失败
9100	余额不足
9101	无权限操作

示例：

### 请求示例

```
curl -d '{"accessKey":"xxxxxxxxxxxxxxxx", "appId":"xxxxxxxxxxxxxxxx",  
"data":{"registerTime":1465810229852, "accountId":"136386957", "ip":"111.85.4  
3.52", "phone":"13800138000", "tokenId":"xxxxx"}}' 'http://antirobot-  
api.fengkongcloud.com/v2/saas/register'
```

### 返回示例

```
{  
  "code":1100,  
  "detail":{  
    "description":"正常",  
    "model":"M1000",  
    "relatedItems":[]  
  },  
  "message":"成功",  
  "requestId":"585f346fd836f98403d72239f612b13a",  
  "riskLevel":"PASS",  
  "score":0  
}
```

## 2.4 用户登录

### 2.4.1 客户端程序

1. 首先，获取 deviceId

> Android

```
String deviceId = SmAntiFraud.getDeviceId();
```

> IOS

```
NSString *deviceId = [[SmAntiFraud sharedInstance] getDeviceId];
```

> Web

```
var deviceId = SMSdk.getDeviceId();
```

2. 然后，调用客户服务端登录接口

在原有登录参数（用户账号 ID 等）的基础上，加入 deviceId 参数。

## 2.4.2 客户服务端程序

执行登录动作前，按照 2.4.3 的说明，调用数美机器登录识别服务 API。

## 2.4.3 数美机器登录识别服务 API

### 请求 URL

http://antirobot-api.fengkongcloud.com/v2/saas/login

### 字符编码

请求及返回结果都使用 UTF-8 字符集进行编码

### 请求方法

POST

### 建议超时时长：

1s

### 请求参数

放在 HTTP Body 中，采用 Json 格式，具体参数如下：

字段	类型	说明	是否必须存在
accessKey	string	用于权限认证，由数美提供。获取方式参见 1.2 节。	是
appId	string	应用 id，用于区分相同公司的不同应用。内容自定义，支持 2-32 位字符。例如：“com.xxx.xxxx”	是
data	json_object	登录信息	是

其中，data 的内容如下：

字段	类型	说明	是否必须存在	可选值
tokenId	string	登录用户账号 id	是	
ip	string	用户登录 ip	是	
deviceId	string	数美设备指纹标识，由于用户行为分析。相比 mac、imei 等，数美设备指纹更难被篡改，当恶意用户篡改 mac、imei 等设备信息时，使用 deviceId 能够发现和识别此类恶意行为，同时，可用于比对数美设备黑名单。	是	
phone	string	用户手机号，用于比对通讯小号、数美黑手机号名单等	两者不能同时为空	
phoneMd5	string	注册账号手机号的 md5 值		
role	string	登陆用户等级，对有消费或高等级用户，置为 VIP，防止误伤，普通用户为 USER。	否	
channel	string	渠道信息	否	
valid	string	用户密码验证结果	否	0：不合法 1：合法 默认值为 1
userExist	string	用户是否存在	否	0：不存在 1：存在 默认值为 1
captchaValid	string	验证码是否通过	否	0：不通过 1：通过 默认值为 1

## 返回参数

放在 HTTP Body 中，采用 Json 格式，具体参数如下：

字段	类型	说明	是否必须存在	可选值
code	int	返回码	是	
message	string	详细描述	是	
score	int	风险分数，[ 0,1000 ] 整数，分数越高风险越大 ( code 为 1100 时存在 )	否	[0,1000]
riskLevel	string	风险级别 ( code 为 1100 时存在 )	否	PASS、REJECT、REVIEW，其中 PASS 为系统认定没有问题的，建议直接放行，REJECT 为系统认定有问题的，建议直接拦截，REVIEW 为系统不确定的，建议人工审核
detail	json_object	风险详情 ( code 为 1100 时存在 )	否	

其中，detail 的格式如下：

字段	类型	说明	是否必须存在

description	string	风险描述	是
model	string	模型编号	是

code 和 message 的列表如下：

code	message
1100	成功
1902	参数不合法
1903	服务失败
9100	余额不足
9101	无权限操作

示例：

**请求示例**

curl -d

```
'{"accessKey":"xxxxxxxxxxxxxxxxxxxxx","data":{"tokenId":"136386957","ip":"111.85.43.52","deviceId":"xxxxxxxx","phone":"xxxxxxxx"}}' http://antirobot-api.fengkongcloud.com/v2/saas/login'
```

**返回示例**

```
{
  "code":1100,
  "detail":{
    "description":"正常",
    "model":"M1000"
  },
}
```

```
"message": "成功",
"requestId": "e852cf2953d762449a947a05e86a1c4b",
"riskLevel": "PASS",
"score": 0
}
```

## 2.5 营销活动反作弊

### 2.5.1 客户端程序

1. 首先，获取 deviceId

> Android

```
String deviceId = SmAntiFraud.getDeviceId();
```

> IOS

```
NSString *deviceId = [[SmAntiFraud sharedInstance] getDeviceId];
```

> Web

```
var deviceId = SMSdk.getDeviceId();
```

2. 然后，调用客户服务端营销活动接口

在原有营销活动参数（用户账号 ID 等）的基础上，加入 deviceId 参数。

### 2.5.2 客户服务端程序

执行营销活动动作前，按照 2.5.3 的说明，调用数美营销活动反作弊服务 API；

### 2.5.3 数美营销活动反作弊服务 API

#### 请求 URL

<http://antirobot-api.fengkongcloud.com/v2/saas/hongbao>

## 字符编码

请求及返回结果都使用 UTF-8 字符集进行编码

## 请求方法

POST

## 建议超时时长：

1s

## 请求参数

放在 HTTP Body 中，采用 Json 格式，具体参数如下：

字段	类型	说明	是否必须存在
accessKey	string	用于权限认证，由数美提供。获取方式参见 1.2 节。	是
appId	string	应用 id，用于区分相同公司的不同应用。内容自定义，支持 2-32 位字符。例如：“com.xxx.xxxx”	是
data	json_object	营销活动信息	是

其中，data 传入参数如下所示：

字段	类型	说明	是否必须存在
channel	string	可选值： HONGBAO 抢红包渠道 FOLLOW 关注渠道 LIKE 点赞渠道 ENTER_ROOM 进入直播间 SEARCH 搜索渠道 DETAIL 详情渠道	是

tokenId	string	用户账号 id	是
ip	string	用户 ip	是
deviceId	string	数美设备指纹标识，由于用户行为分析。相比 mac、imei 等，数美设备指纹更难被篡改，当恶意用户篡改 mac、imei 等设备信息时，使用 deviceId 能够发现和识别此类恶意行为，同时，可用于比对数美设备黑名单。	是
phone	string	账号机号，用于比对通讯小号、数美黑手机号名单等	两者不能同时为空
phoneMd5	string	账号手机号的 md5 值	
receiveTokenId	string	被关注用户的 id，FOLLOW 渠道时需要传入	否
receiveObjec tId	string	被点赞主体的 id，例如：评论 id，帖子 id 等，LIKE 渠道时需要传入	否
roomId	string	房间 id，ENTER_ROOM 渠道时需要传入	否

## 返回参数

放在 HTTP Body 中，采用 Json 格式，具体参数如下：

字段	类型	说明	是否必须存在	可选值
code	int	返回码	是	
message	string	详细描述	是	
score	int	风险分数，[ 0,1000 ] 整数，分数越高风险越大（code 为 1100 时存在）	否	[0,1000]

riskLevel	string	风险级别 ( code 为 1100 时存在 )	否	PASS、REJECT、REVIEW，其中 PASS 为系统认定没有问题的，建议直接放行，REJECT 为系统认定有问题的，建议直接拦截，REVIEW 为系统不确定的，建议人工审核
detail	json_object	风险详情 ( code 为 1100 时存在 )	否	

其中，detail 的格式如下：

字段	类型	说明	是否必须存在
description	string	风险描述	是
model	string	模型编号	是

code 和 message 的列表如下：

code	message
1100	成功
1902	参数不合法
1903	服务失败
9100	余额不足
9101	无权限操作

示例：

#### 请求示例

```
curl -d
```

```
'{"accessKey":"xxxxxxxxxxxxxxxxxxxxx","data":{"tokenId":"136386957","ip":"111.85.43.52","deviceId":"xxxxxxxx","phone":"xxxxxxxx"}}' 'http://antirobot-api.fengkongcloud.com/v2/saas/hongbao'
```

#### 返回示例

```
{
  "code":1100,
  "detail":{
    "description":"正常",
    "model":"M1000"
  },
  "message":"成功",
  "requestId":"eb55f8a027dac433915843d8d9c70389",
  "riskLevel":"PASS",
  "score":0
}
```

## 3. 接入 demo

目前提供了 go、java、lua、php 的 demo，代码位置：  
<https://github.com/ishumei/api-demo>

## 4. 数美管理后台

数美管理后台用来查看各个服务的访问统计、历史记录，同时可进行服务相关的操作。

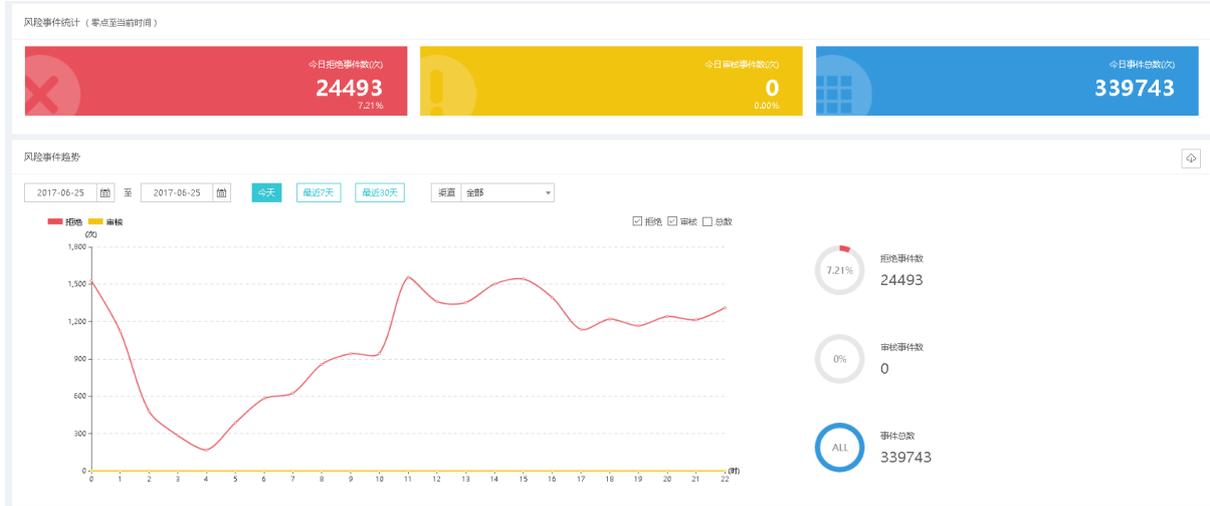
### 4.1 登录管理界面

进入 [www.fengkongcloud.com](http://www.fengkongcloud.com)



## 4.2 体验功能

### 4.2.1 趋势分析（注册&登录）



### 4.2.2 历史记录（注册&登录）

2017-06-25 00:00:00 至 2017-06-25 23:59:59

<input type="checkbox"/>	流水号	设备	tokenid	设备标识	原因	识别结果	昵称	文本内容	发生时间
<input type="checkbox"/>	bcd7a1b32...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404333.73	06-25 23:25:03
<input type="checkbox"/>	45d7857b2...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404316.84	06-25 23:25:16
<input type="checkbox"/>	024b0a1e...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404316.34	06-25 23:25:16
<input type="checkbox"/>	9cd92ee903...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404303.67	06-25 23:25:03
<input type="checkbox"/>	195f70a468...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404286.71	06-25 23:24:46
<input type="checkbox"/>	0aec6f03d...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404286.26	06-25 23:24:46
<input type="checkbox"/>	724256b0c...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404273.59	06-25 23:24:33
<input type="checkbox"/>	e9b05b50e...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404256.58	06-25 23:24:16
<input type="checkbox"/>	382d0f556...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404256.18	06-25 23:24:16
<input type="checkbox"/>	2ea0cccf40...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404243.52	06-25 23:24:03
<input type="checkbox"/>	8de47ea06...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404226.44	06-25 23:23:46
<input type="checkbox"/>	8eda0fd74...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404226.09	06-25 23:23:46
<input type="checkbox"/>	c6472d869...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404213.46	06-25 23:23:33
<input type="checkbox"/>	8f30f2125e...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404196.3	06-25 23:23:16
<input type="checkbox"/>	97f0ac346...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404196.0	06-25 23:23:16
<input type="checkbox"/>	fa8bb5113...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404183.39	06-25 23:23:03
<input type="checkbox"/>	59a945795...	评论	manitar_shumei_test	-	命中黑名单	拒绝	-	线上监控1498404166.16	06-25 23:22:46

### 4.2.3 设备信息查看（设备指纹）

1) 在界面左侧导航栏选择“设备指纹”的“历史记录”页面



## 2) 查看右侧的设备访问记录

数美id	平台 (All) ▼	设备标识	app版本	手机型号	系统版本	IP
20170531223823e7af4fce9468212...	android	5812390397af4ce9488212a...	6.9.2	HUAWEI TAG-TL00	5.1	[REDACTED]
201706282132388a42ffe0256dde...	android	[REDACTED]	6.9.2	MHA-AL00	7.0	[REDACTED]
20170628212456e8ec4d57205f1...	android	[REDACTED]	6.9.2	F100A	5.1	[REDACTED]
20170627210615669006ef7e78a...	android	[REDACTED]	6.9.2	GT-N7100	4.3	[REDACTED]

## 4.2.4 更多

请登录 [www.fengkongcloud.com](http://www.fengkongcloud.com) 网址，发现更多功能。

# 5. FAQ

## 5.1 数美 SDK

### 5.1.1 JS SDK 支持的最低浏览器版本？

答：支持最低是 IE6

### 5.1.2 JS SDK 为什么会访问 fp.ws flash 的文件？

答：JS 会采集 flash 的相关信息。

### 5.1.3 数美 JS SDK 能否可以绑定到客户的域下？

答：目前 JS 只支持放在数美域名下。

### 5.1.4 JS SDK 通过异步方式加载进来，网络比较慢的情况下，是否会出现无法生成设备唯一 ID 的情况？

答：JS 的文件在 CDN 上的，通过 CDN 进行加速。确实可能存在，但概率很低。目前客户还没出现问题。同时，JS 有降级处理逻辑，即使部分流程出现问题，也不会影响设备唯一 ID 的生成。

### 5.1.5 数美 accesskey 有效期是否有限制？是否分测试和正式环境？

答：数美对 accesskey 的有效期是没有限制的，也不区分测试和正式环境。

## 5.2 接口调用

### 5.2.1 调用接口返回参数错误（1902）

答：调用数美接口时，code 返回 1902 参数不合法，一般为客户输入的参数格式存在问题，客户可自行分析一下请求格式是否按照接口文档输入，或将请求的数据及返回数据反馈给数美分析解决。

### 5.2.2 调用接口返回无权限操作（9101）

答：调用数美接口时，code 返回 9101 无权限操作，一般为客户调用了未给客户开通的服务，沟通确认客户调用的服务接口，开通相应的服务。

### 5.2.3 调用接口超时问题

答：有如下两个常见问题：

#### 1. DNS 问题

客户通过公网调用数美接口进行测试，客户 DNS 解析域名较慢，导致第一次请求超时，建议客户更换 DNS，不建议客户在 host 中将域名和 ip 做绑定，数美更换接口 IP 导致无法请求接口。

#### 2) 网络问题

客户通过公网调用数美接口，公网网络延迟较长，导致少量请求存在超时。可以建议客户 ping 数美不同的集群网络，建议客户接入网络延迟较低的数美集群。

## 5.3 更多问题

可通过如下途径联系我们。

### 5.3.1 电邮

请发送数美服务邮箱 ( [api@ishumei.com](mailto:api@ishumei.com) )

### 5.4.2 运营电话

姓名	手机号
陈建	15011518732
王晓涛	18733198521
高松柏	18545801201