

## BatMobi 广告接口协议

一简介.....	2
二 API 地址.....	3
功能说明.....	3
请求说明.....	3
响应说明.....	4
样例数据.....	4
三数据模型.....	6
广告基本信息模型.....	6
响应码.....	6
四 POST BACK 设置.....	8
五应用时长数据上传.....	9
六补充说明.....	11
ua 参数获取.....	11
local 参数获取.....	11
lang 参数获取.....	11
aid 参数获取.....	12
adv_id 参数获取.....	12



# 一简介

这篇文档制定关于 BatMobi 广告协议，致力于多第三方广告平台支持和第三方统计平台支持，客户端或者第三方 publisher 可以通过 API 获取广告应用集成到他们的产品中，附有请求和返回数据的例子。



## 二 API 地址

请求通过 HTTP GET 的方式进行请求，数据以 JSON 的形式返回在 HTTP body，请求参数必须进行 URL encoded，必须传递参数 partnerid 和 token 进行访问。

API 服务统一域名：<http://api.batmobi.net>

## 功能说明

根据不同的渠道方以及用户的基础信息，结合机器学习，为用户实时推荐优质资源数据。

接口地址：</s2s/v1/online>

传参方式：GET

## 请求说明

通过 get 方式，进行参数的上传；如：

[http://api.batmobi.net/s2s/v1/online?token=8a8506b7-1767-41d4-a7b1-fa7a46d8f859&aid=4adc2ca8cb29afd7&adv\\_id=5afb5223-e7f3-4bdc-bab6-099d6a27c7a6&ads\\_num=50&aff\\_sub=xx\\_launcher&aff\\_sub2=app\\_picks&os=0&isp=72403&local=br&lang=pt\\_br&osver=4.4.2&net\\_name=WIFI&dl\\_type=2&request\\_id=20150528112012&ua=Mozilla%2F5.0%20%28Linux%3B%20Android%204.4.2%3B%20LG-D375%20Build%2FKOT49I%29%20AppleWebKit%2F537.36%20%28KHTML%2C%20like%20Gecko%29%20Version%2F4.0%20Chrome%2F30.0.0.0%20Mobile%20Safari%2F537.36](http://api.batmobi.net/s2s/v1/online?token=8a8506b7-1767-41d4-a7b1-fa7a46d8f859&aid=4adc2ca8cb29afd7&adv_id=5afb5223-e7f3-4bdc-bab6-099d6a27c7a6&ads_num=50&aff_sub=xx_launcher&aff_sub2=app_picks&os=0&isp=72403&local=br&lang=pt_br&osver=4.4.2&net_name=WIFI&dl_type=2&request_id=20150528112012&ua=Mozilla%2F5.0%20%28Linux%3B%20Android%204.4.2%3B%20LG-D375%20Build%2FKOT49I%29%20AppleWebKit%2F537.36%20%28KHTML%2C%20like%20Gecko%29%20Version%2F4.0%20Chrome%2F30.0.0.0%20Mobile%20Safari%2F537.36)

请求参数				
参数	必要	类型	描述	说明
token	Yes	String		1 用于权限验证，由服务方统一分配
ua	Yes	String	用户浏览器的 user-agent，需作 urlEncode 处理	例如: Mozilla/5.0 (Linux; U; Android 4.0.3; de-ch; HTC Sensation Build/IML74K) AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Mobile Safari/534.30, 获取方式见文档附录
device_ip	Yes	String	设备 ip	例如: 82.25.55.73 若服务器对接，则需要上传手机端的 IP; 若手机端对接，则不需要上传手机端的 IP，直接放空;
adv_id	Yes	String	用户唯一标识符	iOS: IDFA Android: Android Advertising ID 获取方式见文档附录
aid	Yes	String	Android ID	若 iOS 平台，则放空 获取方式见文档附录
request_id	Yes	String	uuid	用户产生一个跟踪的 id，如时间戳 20150528113725



os	Yes	int	操作系统类型	可选类型: 0 代表 ANDROID 1 -代表 iOS
osver	Yes	String	操作系统版本	例如: 4.1
device_type	Yes	int	设备类型	可选类型: 1 代表 MOBILE 2 代表 TABLET
ads_num	Yes	int	返回广告数量	最多 50 个, 默认 1 个
local	Yes	String	用户所属国家	例如 US 获取方式见文档附录
lang	Yes	String	用户手机语言	例如 en_US 获取方式见文档附录
dl_type	Yes	int	资源下载类型	可选类型: 1 代表 DDL(APK)下载 2 代表 GP 3 代表 iOS
aff_sub	No	String	用户自定义参数1	
aff_sub2	No	String	用户自定义参数2	
aff_sub3	No	String	用户自定义参数3	
ads_size	No	String	广告尺寸	例如 google play icon size 为: 320 * 320
loc	No	String	GPS 的经纬度	例如: 44.000000,-2.000000
isp	No	String	运营商 ID (i. e. MNC)	例如: 42501 Extraction Suggestion: ((TelephonyManager) context.getSystemService(Context.TELEPHONY_SERVICE)) .getSimOperator();
net_name	No	int	网络名称	如: GPRS, WIFI, LTE

## 响应说明

属性名	是否返回	类型	描述	说明
resp_code	Always	Int	HTTP <a href="#">响应码</a>	例如: 200, 具体请参考下面的响应代码表
resp_msg	Always	String	HTTP 响应消息	例如: OK
req_id	Always	String	请求 id	对应用户产生的 uuid
offers	Always	Array	<a href="#">广告基本信息数组</a>	包含广告的基本信息, json 数组
offers_cache_time	Always	Int	缓存时间, 单位: 小时	控制客户端广告缓存最大有效时间

## 样例数据

请求完整测试地址:



[http://api.batmobi.net/s2s/v1/online?token=ce7baf47-9010-4328-a275-552a536d5aee&aid=4adc2ca8cb29afd7&adv\\_id=5afb5223-e7f3-4bdc-bab6-099d6a27c7a6&ads\\_num=1&aff\\_sub=xx\\_launcher&aff\\_sub2=app\\_picks&os=0&isp=72403&local=br&lang=pt\\_br&osver=4.4.2&net\\_name=WIFI&dl\\_type=2&request\\_id=20150528112012&ua=Mozilla%2F5.0%20%28Linux%3B%20Android%204.4.2%3B%20LG-D375%20Build%2F%20KOT49I%29%20AppleWebKit%2F537.36%20%28KHTML%2C%20like%20Gecko%29%20Version%2F4.0%20Chrome%2F30.0.0.0%20Mobile%20Safari%2F537.36](http://api.batmobi.net/s2s/v1/online?token=ce7baf47-9010-4328-a275-552a536d5aee&aid=4adc2ca8cb29afd7&adv_id=5afb5223-e7f3-4bdc-bab6-099d6a27c7a6&ads_num=1&aff_sub=xx_launcher&aff_sub2=app_picks&os=0&isp=72403&local=br&lang=pt_br&osver=4.4.2&net_name=WIFI&dl_type=2&request_id=20150528112012&ua=Mozilla%2F5.0%20%28Linux%3B%20Android%204.4.2%3B%20LG-D375%20Build%2F%20KOT49I%29%20AppleWebKit%2F537.36%20%28KHTML%2C%20like%20Gecko%29%20Version%2F4.0%20Chrome%2F30.0.0.0%20Mobile%20Safari%2F537.36)

(android 平台无中国大陆广告，测试时可追加 device\_ip 参数 &device\_ip=195.112.172.147 模拟获取美国广告.)

响应数据：

```
JSON
{
  "resp_code": 200,
  "resp_msg": "success",
  "req_id": "",
  "offers": [
    {
      "camp_id": "17291",
      "creatives": [
        {
          "320x200_0": "http://cdn.batmobi.net/batmobi/pkgsimgs/1433745238483_320x200_me.lyft.android.jpg",
          "480x800_0": "http://cdn.batmobi.net/batmobi/pkgsimgs/1434075861277_480x800_me.lyft.android_01.jpg",
          "480x480_0": "https://lh3.googleusercontent.com/atFKR3usVJm12J3vtL2v7ElHpnt87rFxqPkZtMnpJgrZwXD4bywWD_ZOWiFsLpVcM=h31"
        }
      ],
      "imp_url": null,
      "click_url": "http://api.batmobi.net/call/v3/click?ads_id=17291&aff_id=1001571&recommend_id=3012",
      "click_callback_url": "http://api.batmobi.net/call/v3/click_callback?ads_id=17291&aff_id=1001571&recommend_id=3012",
      "mobile_app_id": "me.lyft.android",
      "mobile_platform": "android",
      "payout_amount": 3.84,
      "payout_currency": "$",
      "acquisition_flow": "CPA",
      "icon_gp": "https://lh5.ggpht.com/CfRup8ZMEbNpd4sf8PDUWzbnFIq0QINZpJ96M6V0-8wLmyqKD0ORSqPOq5EKdL1OskE=w300",
      "description": "Need a lift? Use Lyft to get an affordable ride in minutes. Instead of hailing a cab or waiting for the bus, just request a car with the",
      "name": "Lyft - Taxi & Bus Alternative",
      "rate": 4.3,
      "store_rating": 4.3,
      "installs": "1000000",
      "category": "Transportation",
      "dl_type": 2,
      "preload": 1
    }
  ],
  "totalrecord": 1,
  "totalpage": 1,
  "pageid": 0,
  "offers_cache_time": 4
}
```



# 三数据模型

## 广告基本信息模型

属性名	是否返回	类型	描述	说明
camp_id	Always	String	BatMobi 提供的 campaign ID	
creatives		json	图片 url <pre>{   "480x480_0":     "https://lh3.googleusercontent.com/19fFBuZsKC9R2F3ovoWtkXFKJyeZbdMhU38UEratfCemVL0oK2fuRw-GnU1TbpdQLdiD=h310" }</pre>	展示图片
imp_url		String	impression url	用于统计展示和第三方回调
click_url		String	click url	用于统计跟踪点击和第三方回调
click_callback_url		String		用户真实点击回调 URL
mobile_app_id		String	应用包名	
mobile_platform		String	系统版本	
payout_amount		Float	单价	
payout_currency		String	价格单位	
acquisition_flow		String	广告计费方式	如 CPA, CPI, CPC
icon_gp		String	图标(谷歌)	
description		String	应用简单描述	
name		String	应用名称	
rate		Float	应用星级	
store_rating		float	评分	
installs		String	安装量	
category		String	广告类别	见下面应用分类

## 响应码

响应码	描述
200	OK
204	服务器成功处理，返回无结果
400	Bad request.
401	Unauthorized. 验证不通过
403	Forbidden. 禁止访问-可能要加入白名单
404	404
429	同一时间请求太多次
500	Server error. 广告服务出错



503

服务不可用



## 四 POST BACK 设置

广告点击 URL 可支持在后面动态拼凑三个参数支持回调

参数名	描述	备注
aff_sub	可自定义的参数	
aff_sub2	可自定义的参数	
aff_sub3	可自定义的参数	

Demo:

[http://api.batmobi.net/call/v3/click?ads\\_id=19981&aff\\_id=1001568&aff\\_sub={x1}&aff\\_sub2={x2}&aff\\_sub3={x3}](http://api.batmobi.net/call/v3/click?ads_id=19981&aff_id=1001568&aff_sub={x1}&aff_sub2={x2}&aff_sub3={x3})

开发者可以通过两种方式进行自定义参数的拼装:

- 1 在请求广告的时候, 在请求信息里面添加上(具体见“请求说明”), 广告服务器会默认帮忙把所有的 click\_url 拼装上自定义参数信息;
- 2 开发者把广告请求回来本地后, 在本地进行 click\_url 的自定义参数拼装;

POSTBACK URL 设置, 需要你们提供如下 demo 的 URL 提供给我们在后台设置

[http://yourserver/postback?x1={aff\\_sub}&x2={aff\\_sub2}&x3={aff\\_sub3}](http://yourserver/postback?x1={aff_sub}&x2={aff_sub2}&x3={aff_sub3})

x1,x2,x3 参数名开发者可自定义



## 五应用时长数据上传

每天收集一次用户手机应用的使用时长情况，以满足后续更深层次的用户级别广告推荐，手机端每天数据上传的时机，为手机当前时间 00 点~ 02 点之间进行随机上传。

上传方式: **Post**

上传地址: <http://sts.batmobi.net/log?typeid=1>

Post 上传参数说明:

属性名	类型	说明
data	String	<p>提交的 json 格式请求数据。</p> <p><b>json.toString().getBytes("UTF-8"); 然后对 bytes 进行 BASE64加密处理; 最后对加密结果 bytes 采用 ISO-8859-1编码成字符串。</b></p> <p>json.toString() 数据 demo:</p> <pre>{   "token": "1437191900904f5600b03e644c0b9",   "date": "2015-07-22",   "aid": "4f5600b03e644c0b9"   "events": [     {       "count": 2,       "usage": 91273,       "pkg": "com.example.test1"     },     {       "count": 22,       "usage": 91273,       "pkg": "com.example.test2"     }   ] }</pre>

应用时长信息体

属性名	类型	说明
token	String	由服务端统一分配
date	String	所属日期, 格式: 2015-06-12
aid	String	安卓 ID
events	JSONArray	事件列表, 见下述 事件单元信息体, json 数组



## 事件单元信息体

属性名	类型	说明
pkg	String	应用包名
count	int	当天的应用打开次数
usage	int	当天应用的总使用时长，单位:毫秒



## 六补充说明

### ua 参数获取

```
//Android平台获取方式
public void getUserAgent() {
    WebView wv = new WebView(this);
    String ua = wv.getSettings().getUserAgentString();
    return ua;
}
```

### local 参数获取

```
//Android平台获取方式
public static String getSimCountry(Context context) {
    String ret = null;
    Locale locale = Locale.getDefault();
    try {
        TelephonyManager telManager =
            (TelephonyManager) context.getSystemService(Context.TELEPHONY_SERVICE);
        ret = telManager.getSimCountryIso();
    } catch (Throwable e) {
        e.printStackTrace();
    }
    if (TextUtils.isEmpty(ret)) {
        ret = locale.getCountry().toLowerCase();
    }
    return null == ret ? "error" : ret;
}
```

### lang 参数获取

```
// 获取语言和国家地区的方法 格式: SIM卡方式: cn 系统语言方式: zh-CN
public static String getLanguage(Context context) {
    String ret = null;
    Locale locale = Locale.getDefault();
    try {
        TelephonyManager telManager =
            (TelephonyManager) context.getSystemService(Context.TELEPHONY_SERVICE);
        ret = telManager.getSimCountryIso();
        if (!TextUtils.isEmpty(ret)) {
```





```

        ret = String.format("%s_%s", locale.getLanguage().toLowerCase(), ret.toLowerCase());
    }
} catch (Throwable e) {
    e.printStackTrace();
}
if (TextUtils.isEmpty(ret)) {
    ret = String.format("%s_%s", locale.getLanguage().toLowerCase(), locale.getCountry().toLowerCase());
}
return null == ret ? "error" : ret;
}

```

## aid 参数获取

```

// 获取安卓 ID
public String getAndroidId(Context context) {
    if (null == sAndroidId) {
        sAndroidId = Settings.Secure.getString(context.getContentResolver(),
            Settings.Secure.ANDROID_ID);
    }
    return sAndroidId;
}

```

## adv\_id 参数获取

```

import java.io.IOException;
import java.util.concurrent.LinkedBlockingQueue;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.ServiceConnection;
import android.content.pm.PackageManager;
import android.os.IBinder;
import android.os.IInterface;
import android.os.Looper;
import android.os.Parcel;
import android.os.RemoteException;

public class AdvertisingIdUtil {

    //adv_id 获取
    public static String getAdvertisingId(Context context) {
        try {
            AdInfo adInfo = getAdvertisingIdInfo(context);
            if(adInfo != null) {

```



```

        return adInfo.getId();
    }
} catch (Exception e) {
    e.printStackTrace();
}
return null;
}

public static AdInfo getAdvertisingIdInfo(Context context) throws Exception {
    if (Looper.myLooper() == Looper.getMainLooper())
        throw new IllegalStateException("Cannot be called from the main thread");
    try {
        PackageManager pm = context.getPackageManager();
        pm.getPackageInfo("com.android.vending", 0);
    } catch (Exception e) {
        throw e;
    }
    AdvertisingConnection connection = new AdvertisingConnection();
    Intent intent = new Intent("com.google.android.gms.ads.identifier.service.START");
    intent.setPackage("com.google.android.gms");
    if (context.bindService(intent, connection, Context.BIND_AUTO_CREATE)) {
        try {
            AdvertisingInterface adInterface = new AdvertisingInterface(connection.getBinder());
            AdInfo adInfo = new AdInfo(adInterface.getId(), adInterface.isLimitAdTrackingEnabled(true));
            return adInfo;
        } catch (Exception exception) {
            throw exception;
        } finally {
            context.unbindService(connection);
        }
    }
    throw new IOException("Google Play connection failed");
}

private static final class AdvertisingConnection implements ServiceConnection {
    boolean retrieved = false;
    private final LinkedBlockingQueue<IBinder> queue = new LinkedBlockingQueue<IBinder>(1);

    public void onServiceConnected(ComponentName name, IBinder service) {
        try {
            this.queue.put(service);
        } catch (InterruptedException localInterruptedException) {
        }
    }

    public void onServiceDisconnected(ComponentName name) {
    }
}

```



```

    public IBinder getBinder() throws InterruptedException {
        if (this.retrieved)
            throw new IllegalStateException();
        this.retrieved = true;
        return (IBinder) this.queue.take();
    }
}

private static final class AdvertisingInterface implements IInterface {
    private IBinder binder;

    public AdvertisingInterface(IBinder pBinder) {
        binder = pBinder;
    }

    public IBinder asBinder() {
        return binder;
    }

    public String getId() throws RemoteException {
        Parcel data = Parcel.obtain();
        Parcel reply = Parcel.obtain();
        String id;
        try {
            data.writeInterfaceToken("com.google.android.gms.ads.identifier.internal.IAdvertisingIdService");
            binder.transact(1, data, reply, 0);
            reply.readException();
            id = reply.readString();
        } finally {
            reply.recycle();
            data.recycle();
        }
        return id;
    }

    public boolean isLimitAdTrackingEnabled(boolean paramBoolean) throws RemoteException {
        Parcel data = Parcel.obtain();
        Parcel reply = Parcel.obtain();
        boolean limitAdTracking;
        try {
            data.writeInterfaceToken("com.google.android.gms.ads.identifier.internal.IAdvertisingIdService");
            data.writeInt(paramBoolean ? 1 : 0);
            binder.transact(2, data, reply, 0);
            reply.readException();
            limitAdTracking = 0 != reply.readInt();
        } finally {

```



```
        reply.recycle();
        data.recycle();
    }
    return limitAdTracking;
}

}

public static final class AdInfo {
    private final String advertisingId;
    private final boolean limitAdTrackingEnabled;

    AdInfo(String advertisingId, boolean limitAdTrackingEnabled) {
        this.advertisingId = advertisingId;
        this.limitAdTrackingEnabled = limitAdTrackingEnabled;
    }

    public String getId() {
        return this.advertisingId;
    }

    public boolean isLimitAdTrackingEnabled() {
        return this.limitAdTrackingEnabled;
    }
}
}
```