

Avazu Native Ad SDK Setup for iOS

Date	Version	Description	Developer
2015.03.12	v1.0.0	First Version	曹诗聪
2015.03.25	v1.0.1	Optimization and issue fix	曹诗聪
2015.03.27	v1.0.2	Sample Project Optimization	曹诗聪
2015.05.22	v1.1.0	Change Server	曹诗聪
2015.07.03	v1.2.0	Add C2S API	曹诗聪
2015.07.06	v1.2.1	Add Parameter to get creative picture data	曹诗聪

1. 简介	3
1.1 SDK功能介绍	3
1.2 兼容性	3
1.3 开发环境要求	3
2. 接入配置	4
2.1 Traffic Source ID	4
2.2 导入SDK文件	4
2.3 添加依赖库	4
2.4 添加linker flag	5
3. 展示广告	5
3.1 代码方式	5
3.1.1 创建AvazuADView实例	5
3.1.2 选择一个适合您的广告形式	6
3.1.3 定制化广告	6
3.2 xib或storyboard方式	8
4. 实现delegate	9
5. C2S接口	9

1.简介

1.1 SDK功能介绍

Avazu ADSDK 能够提供以下功能

- 展示单个广告banner (推荐尺寸: 320 * 100)
- 展示透明banner (推荐尺寸: 400 * 110)
- 展示广告banner墙
- 展示单行矩形广告墙
- 展示多行矩形广告墙

以上这些广告展现方式, 开发者可以根据SDK提供的方法实现广告外观和展示应用数量的定制

1.2 兼容性

支持 iOS6.0 以上的系统

1.3 开发环境要求

操作系统: macosx lion及以上版本

开发工具: Xcode 5.0及以上版本

2. 接入配置

2.1 Traffic Source ID

确保您已注册Avazu APX账号且已有可用的Traffic Source ID for Display

2.2 导入SDK文件

请将以下文件添加到您的项目中：













- libAvazuAdSDK.a
- AvazuADView.h

2.3 添加依赖库

为了保证SDK能够正确运行，您需要添加以下依赖库到项目中：

- CoreTelephony.framework
- UIKit.framework
- Foundation.framework
- AdSupport.framework

▼ **Link Binary With Libraries** (5 items)

Name	Status
 CoreTelephony.framework	Required 
 libAvazuAdSDK.a	Required 
 UIKit.framework	Required 
 Foundation.framework	Required 
 AdSupport.framework	Required 
<div>   </div> <div>Drag to reorder frameworks</div>	

2.4 添加linker flag

请在Other Linker Flags选项中添加：-ObjC



3. 展示广告

[AvazuADSDK](#) 提供了5种不同种类的广告展现形式，包括单个广告banner，广告banner墙单行矩形广告墙，多行矩形广告墙，透明banner。

作为开发者，您只需要通过[AvazuADView](#)这个接口来创建这些广告类型，[AvazuADView](#)是UIView的子类，可以展现HTML5的广告。

您可以选择以下两种方式来成功展示广告：

3.1 代码方式

3.1.1 创建AvazuADView实例

您需要确认您的 Traffic Source ID是可用的，每次创建广告实例时都需要用到它，否则将无法正确展示广告。

创建步骤如下：

1. Import [AvazuADView.h](#) 文件.
2. 在头文件中声明一个实例，您的头文件类似如下所示：

```
#import <UIKit/UIKit.h>
#import "AvazuADView.h"

@interface ViewController : UIViewController <AvazuADViewDelegate>
@property (strong, nonatomic) AvazuADView *adView;
@end
```

3. 在实现文件中创建实例，范例如下：

```
self.adView = [[AvazuADView alloc]
initWithFrame:adFrame
adType:AVAZU_SINGLE_BANNER
sourceID:@"input your Traffic Source ID here"];
```

4. 创建实例后，在载入广告之前您需要在设置该实例的delegate, 用来通知应用广告载入成功或失败。

5. 将adView添加到主view中

5. 根据您需求定制广告的展现形式和展示应用数量，我们将会3.3章节中做详细说明

7. 用此方法来载入广告：[self.adView loadAD]

以下是Viewcontroller中的一个完整范例：

```

- (void)viewDidLoad {
    [super viewDidLoad];
    CGRect adframe = CGRectMake(0.0, 0.0, 320.0, 100.0);
    self.adView = [[AvazuADView alloc]
        initWithFrame:adframe
        adType:AVAZU_SINGLE_BANNER
        sourceID:@"1234"];
    self.adView.delegate = self;
    [self.view addSubview:self.adView];
    //Insert Customized Adview setup here
    [self.adView loadAD];
}
    
```

8. 在ViewController中, 我们推荐您在dealloc函数将delegate设为nil, 或在任意释放adView的地方您也需要将其设为nil

```

- (void)dealloc {
    self.adView.delegate = nil;
}
    
```

3.1.2 选择一个适合您的广告形式

广告形式	值	变量名
单个广告banner	9	AVAZU_SINGLE_BANNER
banner广告墙	10	AVAZU_BANNER_APPWALL
透明banner	11	AVAZU_TRANSPARENT_SINGLE_BANNER
单行矩形广告墙	12	AVAZU_SINGLE_BUTTON_APP_WALL
多行矩形广告墙	13	AVAZU_MULTIPLE_BUTTON_APP_WALL

3.1.3 定制化广告

AvazuADSDK支持丰富的定制化广告，您可以选择展示的广告的颜色搭配，展示内容以及展示应用的数量，只需要像如下的代码进行设置就可：

```
self.adView.isNeedIcon = 0;
```

1. 广告元素配置

变量名	类型	默认值	作用
isNeedIcon	BOOL	1	设置广告中是否显示应用图标
isNeedTitle	BOOL	1	设置广告中是否显示应用名称
isNeedCat	BOOL	1	设置广告中是否显示应用类别
isNeedSize	BOOL	1	设置广告中是否显示应用尺寸
isNeedRating	BOOL	1	设置广告中是否显示应用评分
isNeedInstallButton	BOOL	1	设置广告中是否显示安装按钮
isNeedReviewNumber	BOOL	1	设置广告中是否显示应用查看次数
isNeedLoadingIndicator	BOOL	1	设置加载广告时时是否显示载入时圆圈

2. 广告颜色配置

变量名	类型	默认值	作用
blockBackColor	NSString	null	设置广告块背景色
appTitleColor	NSString	null	设置应用名字体颜色
buttonBackColor	NSString	null	设置下载按钮颜色
buttonTextColor	NSString	null	设置下载按钮上的字体颜色
mainBackColor	NSString	null	设置应用墙背景色

注意: 配置广告颜色时, 所有颜色值的设置方式为 “#”加上6位十六进制数RGB格式的字符串, 如“#FFFFFF”表示白色, “#000000”表示黑色

3. 展示应用数量配置

变量名	类型	默认值	作用
appCount	INT	1	设置广告中出现的应用数量

注意: 以下广告类型必须设置appCount: 广告banner墙, 单行矩形广告墙, 多行矩形广告墙

4. 设置广告透明度

变量名	类型	默认值	作用
transparentBannerAlpha	INT	0	设置透明广告的透明度

注意: 只有在**透明banner**的广告形式下才用设置transparentBannerAlpha, 合法取值范围0~100,0表示完全透明, 100表示完全不透明

以下是一个定制广告banner墙的完整代码:

```
- (void)viewDidLoad {
    [super viewDidLoad];
    CGRect adframe = CGRectMake(0.0, 0.0, 320.0, 100.0);
    self.adView = [[AvazuADView alloc]
        initWithFrame:adframe
        adType:AVAZU_BANNER_APPWALL
        sourceID:@"1234"];
    self.adView.delegate = self;
    [self.view addSubview:self.adView];
    self.adView.isNeedSize = 0;
    self.adView.isNeedReviewNumber = 0;
    self.adView.appCount = 4;
    [self.adView loadAD];
}
```

3.2 xib或storyboard方式

如果您的项目使用nib或者storyboard实现界面布局, 您可以非常简单的完成广告接入工作, 请按以下步骤实现:

1. 在xib或storyboard中创建合适尺寸的UIview用来显示广告
2. 在右上角的identity inspector中, 将Custom Class设为AvazuADView
3. 打开辅助编辑器, 通过连线方式创建一个IBOutlet.
4. 可以使用Auto layout设置该view的各项约束
5. 在对应的viewController.m文件中的viewDidLoad方法里加入如下代码

```
- (void)viewDidLoad {
    [super viewDidLoad];
    self.sourceID=@"6395";
    self.adView.adType = AVAZU_SINGLE_BANNER;
    self.adView.delegate = self;
    [self.adView loadAD];
}
```

注: 由于使用xib或storyboard创建广告, 系统会自动调用init方法, 所以在调用loadAD方法前必须设置adType和sourceID, 否则将无法正确显示广告

您仍然可以按照3.1.3章节中的方法定制您的广告

4. 实现delegate

如果您需要知道广告显示结果的回调，请在应用中实现如下两个delegate方法，在delegate中您可以知道这次广告是否展示成功，您可以根据应用所需逻辑重写该方法。

```
- (void)avazuADViewLoadAdSucess:(AvazuADView *)adview
{
    NSLog(@"avazuADViewLoadAdSucess");
}

- (void)avazuADView:(AvazuADView *)adview
didFailToReceiveAdWithError:(NSError *)error
{
    NSLog(@"avazuADViewLoadAdFail");
    NSLog(@"error:%@", error);
}
```

5. C2S接口

您可以通过以下方式使用C2S接口：

在ViewController中的viewDidLoad中添加如下代码：

```
- (void)viewDidLoad {
    [super viewDidLoad];
    //initialize
    self.nativdAd = [[AvazuNativeAd alloc] init];
    //set delegate
    self.nativdAd.delegate = self;
    //load ad
    [self.nativdAd loadNativeAdWithSourceId: @"15353"
                                     excludePackages: @"
                                     limitNumber: 10
                                     incent: 0
                                     creatives: 1
    ];
}
```

参数说明：

Source ID: Traffic Source ID

excludePackages: 传入不想显示的广告的 campaign_id, 用于翻页时不显示重复广告, 若有多条以“,”分隔, 若无则传入“”, 例如: “6184, 3241”

limitNumber: 本次想要拉取广告个数的上线

incent: 是否选取激励广告, $\text{incent} = 0$ 时选取非激励广告, $\text{incent} = 1$ 时选取激励广告, $\text{incent} = 2$ 是选择所有广告, 默认为0

creatives: 是否返回多尺寸图片素材, 值为 1 表示返回 图片素材, 默认为 0(不返回图片素材)。可能返回的图片尺寸有:

尺寸	描述
320x50	Banner
320x480	Phone Fullscreen
480x320	Phone Fullscreen Landscape
1024x768	Tablet Fullscreen
768x1024	Tablet Fullscreen Landscape 300x250 - Medium
728x90	Leaderboard
160x600	Skyscraper
1200x627	Content Stream Image

然后, 在delegate中监听拉取广告的Callback:

```
#pragma avazuNativeDelegate

- (void)avazuGetRawdataFailWithError:(NSError *) error
{
    NSLog(@"avazuADViewLoadAdFail");
    NSLog(@"error:%@", error);
}

- (void)avazuGetRawdataSuccess:(NSArray *)dataArray
{
    NSLog(@"avazuADViewLoadAdSuccess");
    NSLog(@"dataArray:%@", [dataArray description]);
}
```

dataArray为获取的NSArray形式的广告数组, 其数据结构如下:

字段	描述
campaignid	广告 ID
payout	一个转化的价格(CPI),单位为美元
pkgname	App 的包名(package name)
title	App 的标题
description	App 的描述(不超过 100 个英文字母)
icon	App 的 icon 地址(100x100 的 png 图片)
appcategory	App 分类
apprating	App 的评分(0-5 星)
appreviewnum	App 的评论数
appinstalls	App 的安装量
appsize	App 的安装文件大小
creatives	App 的多尺寸图片素材地址
countries	定向的国家,格式采用 ISO-3166-1 标准的国家代 字符串 码,多个国家使用” ”分割,例如:USIMY
os	定向的操作系统,[ioslandroid]
minosv	定向的最低操作系统版本号,格式为“主版本号.次版本号”,如:1.2(表示 1.2 和 1.2 版本以上都允许)
connectiontype	定向的网络类型,值为:wifi(只允许 WiFi)、3g (只允许 2G/3G/4G 等运营商网络)、all(不限网络类型)
devicetype	定向的设备类型,[allphonetablet]
incent	是否为激励广告 值为:yes(激励广告)、no(非激励广告)
clkurl	广告的点击 URL