

1. JS 接口 for SDK

1.1 背景

MMA China 对于广告监测有一个 MMA 通用监测 SDK。当可见曝光的概念推出后，MMA China 也针对相应的准则推出了通用（广告）验证 SDK 用于可见曝光的监测。

对于 MMA 通用（广告）验证 SDK，MMA China 选择原生 code 和 JS 轻 SDK 并存的方案。即广告验证 SDK 中融合了 Native 和 JS 两种：

- MMA 通用监测 SDK
- MMA 通用(广告)验证 SDK：
 - 原生 code 方案，只需配置 xml
 - JS 轻 SDK，JS 需要媒体审核各家第三方 JS 并配置 xml

针对 JS 轻 SDK 方案，需要各家第三方需要各自提供 JS 代码来计算原生 code 采集的可见曝光原始数据，计算是否可见，并判断是否需要上报数据并结束该广告位的可见曝光监测。

以下内容主要介绍 JS 轻 SDK 方案中，关于 JS 部分的说明。

1.2 JS 代码主要负责的功能

- 提供接口，SDK 会回传获取到的相关可见曝光原始数据
- 计算 SDK 采集的原始数据，判断是否可见
- 负责数据上报
- 调用 SDK 接口，通知 SDK 停止相关监测

1.3 JS 上报逻辑

各家第三方自行定义上报逻辑

1.4 JS 上报方式

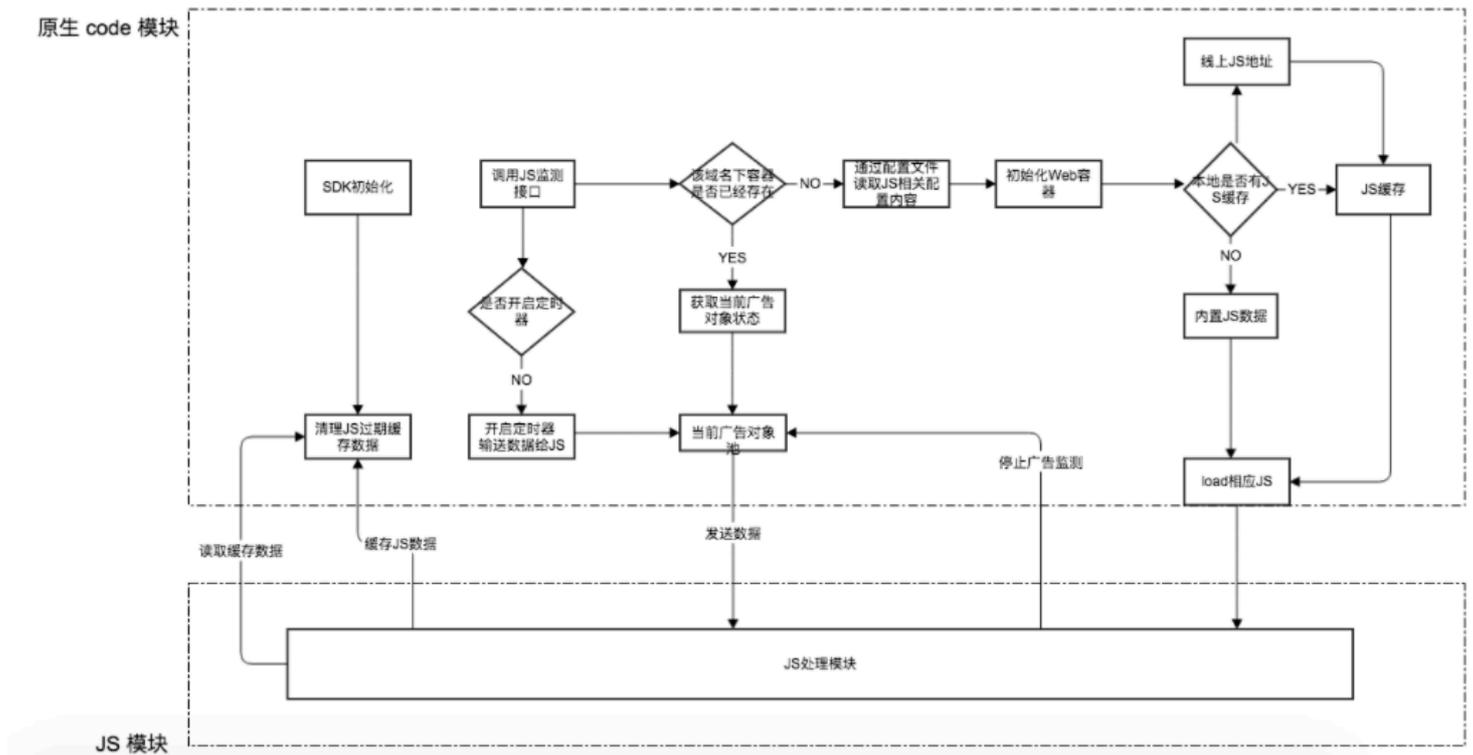
各家第三方自行定义上报的代码格式

1.5 JS 计算逻辑

各家第三方自行定义 JS 计算该广告是否可见的规则

2. MMA viewability SDK JS 部分说明

2.1 基本流程图如下图所示



2.2 基本点

- JS 监测方式与原生 code 监测模式完全隔离。使用 JS 监测模式，原生 code 不会发送普通曝光监测代码，如需发送普通曝光监测代码，请使用 JS 进行补发。原生代码的主要任务是将监测链接及当前广告位信息及时准确的传输给 JS
- JS 监测逻辑的开启，是以接口调用某一家 JS 监测接口为起点的
- MMA SDK 初始化后，为了防止数据积累对用户 App 造成的影响，会对本地缓存的 JS 数据做数据过期清理（暂定 3 天）
- 当 SDK 接收到 JS 监测接口请求时，需要根据当前 JS 监测的主域名地址，对应到 xml 配置文件中，加载相应的 JS 信息到 Web 容器中，一个主域名下的 JS 在一次 App 生命周期内只允许加载一次
 - XML 文件中，在每一个 company 节点下，增加一个 JSURL 的配置，目前一家第三方在一个媒体下只允许使用一个 JSURL 配置
 - 在 SDK 一个生命周期内，不同媒体 / 第三方暂时使用不同的 Web 容器。这样可以天然的把各家监测独立开来（使用相同的 web 容器会带来不同 JS 之前数据安全性的隐患，同时开启几个 web 对性能的影响不太大）
 - 在 SDK 一个生命周期内，同一个媒体 / 第三方同时监测多个广告位的可见曝光，那么同一个域名的监测使用同一个 web 容器
- JS 监测的整个模块，运行开启一个定时器，SDK 需要根据不同第三方域名区分不同的 Web 和广告监测对象，从而将广告对象状态转的传给对应的第三方 JS
- 每一个第三方广告平台，同时监测多个广告对象时，每次定时轮询时，需要将多个对象整理成数组，一次性传入（保证媒体 JS 接受的可控性）
- 开启定时 viewability 监测的广告对象，每隔固定时间将数据以 JSonSting 形式发送给 JS
 - Viewability 信息
 - DeviceMessage
 - 监测 URL
- JS 与原生代码目前提供 5 个接口
 - 发送定时数据接口（原生代码调用 JS 提供的接口，给 JS 发送数据）
 - 停止广告监测接口（JS 调用原生代码提供的接口，通知原生代码停止监测）
 - 存储 JS 数据接口（JS 调用原生代码提供的接口，通知原生代码缓存数据）
 - 读取 JS 缓存数据接口（JS 调用原生代码提供的接口，通知原生代码需要读取缓存数据）
 - 发送缓存数据接口（原生代码调用 JS 提供的接口，给 JS 发送缓存数据）

2.3 广告位监测信息

key	value 含义	是否必须	备注	格式
adurl	广告位的监测代码	YES		string
AdviewabilityID	对应每条监测的 URL	YES	广告监测链接+当前时间戳（毫秒）的 MD5值	string
deviceMessage	原生 code 采集的信息	YES	包含 os,ts,mac,idfa,idfamd5,imei,androidid,wifi,akey,aname,scwh,openudid,term,osvs,lbs,sdkvs 信息	array
viewabilityMessage	原生 code 采集的信息	YES	包含 AdviewabilityTime, AdviewabilityFrame, AdviewabilityPoint, AdviewabilityAlpha, AdviewabilityHide, AdviewabilityCoverRate, AdviewabilityShowFrame, AdviewabilityLight, AdviewabilityForground, AdviewabilityType	array
os	操作系统	YES	0: Android; 1: iOS; 2: WP; 3: other	string
ts	时间戳	YES	客户端触发监测的时间，用于计算离线监测时间	string
mac	MAC 地址	YES	全部大写，冒号分隔符去掉，MD5 值	string
idfa	iOS IDFA	YES	适用于 iOS6 及以上，原始值	string
idfamd5	iOS IDFA md5 加密	YES	32 位 MD5 加密后小写	string
imei	用户终端的 IMEI	YES	32 位 MD5 加密后小写	string
androidid	用户终端的 AndroidID	YES	MD5 加密	string
wifi	客户端触发监测时是否使用 wifi	YES	0: 2/3G ;1: wifi ;2: 无网状态	string
akey	媒体 APP Key	YES	字符串，需 escape 转义	string
aname	媒体 APP Name	YES	字符串，需 escape 转义	string
scwh	用户终端的屏幕分辨率	YES	宽 x 高，用字母 x 分割	string
openudid	iOS OpenUDID	YES	原始值	string
term	用户终端的机型	YES	字符串，需 escape 转义	string
osvs	用户终端的操作系统版本	YES	字符串，需 escape 转义	string
	用户终端设备的经		十进制保留 6 位小数，用字母 x 分割纬度与精度（先纬后经，最后精度），如	

lbs	纬度地理坐标	YES	35.758134x122.406722x100.0 Android 端不获取 LBS 数据，iOS 端按照配置文件开关开启	string
sdks	SDK 版本	YES	字符串，需 escape 转义	string
AdviewabilityTime	当前时间戳	YES	时间格式为毫秒	string
AdviewabilityFrame	当前广告尺寸	YES	当前广告位真实广告尺寸	string
AdviewabilityPoint	广告在屏幕中坐标	YES	广告在屏幕里的坐标，坐标形式回传，横坐标 x 纵坐标，中间以小写 x 分隔	string
AdviewabilityAlpha	广告透明度	YES	0-1 标示，0 标示透明，1 标示不透明	string
AdviewabilityHide	广告是否被隐藏	YES	0 表示隐藏，1 表示没有隐藏	string
AdviewabilityCoverRate	广告是否被遮挡	YES	coversize/adframe,0-1，保留两位小数，如果未被覆盖该参数为 0	string
AdviewabilityShowFrame	当前广告显示尺寸	YES	当前广告在屏幕中的显示尺寸	string
AdviewabilityLight	屏幕是否点亮	YES	0 为未点亮，1 为点亮	string
AdviewabilityForground	app 是否在前台运行	YES	0 为在后台运行，1 为在前台运行	string
AdviewabilityType	广告位是 banner 还是 video	YES	0 为 banner，1 为 video	string

2.4 设备信息（与配置文件一致）

2.5 发送数据格式

```
{
  adurl: http://xxxxxx,
  AdviewabilityID:asd,
  deviceMessage:{
    idfa:ddd,
    imei:adfad,
  },
  viewabilityMessage:{
    AdviewabilityTime:ddd x ddd,
  }
}
```

当 viewabilityMessage 回传内容为空时，为不可测量

2.6 JS 交互协议

- viewability 信息发送接口
 - 函数名称：**sendViewabilityMessage(viewabilityMessage)**
 - 函数描述：SDK 开启定时器后，每次检测到广告位信息后，通过调用 JS 提供的此接口，将数据发送给 JS
 - 调用逻辑：原生代码调用，JS 添加监听方法被动使用。
 - 数据格式：JSON 字符串
 - 参数说明：viewabilityMessage 为本次监测广告位数据包，详细描述参考《广告位 viewability 信息发送格式》

- 调用示例:

```
[webView stringByEvaluatingJavaScriptFromString:[NSString stringWithFormat:@"sendViewabilityMessage(%@)",viewabilityMessage]];
```

- viewability 停止监测接口

- 函数名称: **mmaViewabilitySDK://stopViewability?adviewabilityid=asd**
- 函数描述: JS 通知原生代码, 停止对某一个广告位的监测。
- 调用逻辑: JS 代码调用, 原生代码添加监听方法被动使用。
- 数据格式: 字符串, value 需要编码一次。
- 参数说明: 调用此接口, JS 需要将停止监测的 AdviewabilityID 回传给原生代码
- 调用示例:
mmaViewabilitySDK://stopViewability?adviewabilityid=asd

- 缓存 JS 数据接口

- 函数名称: **mmaViewabilitySDK://saveJSCacheData?data=XXXXXX**
- 函数描述: JS 通知原生代码, 实现对 JS 数据缓存。
- 调用逻辑: JS 代码调用, 原生代码添加监听方法被动使用。
- 数据格式: 字符串, value 需要编码一次。
- 参数说明: 调用此接口, JS 将需要缓存的数据编码后传输给原生代码, 原生代码收到后直接存储在本地, 等待原生代码后续读取。**JS 每次调用该接口, 原生代码会覆盖之前的数据。**
- 注意: JS 处理此逻辑时, 需要调用读取缓存接口, 将之前缓存数据读取出来。**
- 调用示例:
mmaViewabilitySDK://saveJSCacheData?data=XXXXXX

- 读取本地缓存数据接口

- 函数名称: **mmaViewabilitySDK://getJSCacheData? clear=true**
- 函数描述: JS 通知原生代码, 实现对 JS 数据缓存的读取。
- 调用逻辑: JS 代码调用, 原生代码添加监听方法被动使用。
- 参数说明: JS 调用此接口,通知原生代码, 将当前缓存数据读出,clear 字段可缺省, 默认为 ture
- 注意: JS 调用此接口后, 如果 clear 字段缺省或者为 ture, 客户端会将相应的数据清理, 以免造成数据重复传输。**
- 调用示例:
mmaViewabilitySDK://getJSCacheData

- 本地缓存数据发送接口

- 函数名称: **sendCacheMessage(cacheData)**
- 函数描述: JS 通知原生代码需要读取本地缓存数据后, 原生代码通过此接口将数据同步给 JS 代码。
- 调用逻辑: 原生代码调用, JS 添加监听方法被动使用。
- 数据格式: JSon 字符串
- 参数说明: 本地数据会将之前缓存的数据以字符串的形式传输给 JS
- 调用示例:
[webView stringByEvaluatingJavaScriptFromString:[NSString stringWithFormat:@"sendCacheMessage(%@)",cacheMessage]];

2.7 JS 监测注意事项

- SDK 不会发送普通曝光代码, 但是会把监测的 URL 发给 JS, JS 需要自行发送曝光代码, 否则会出现普通曝光计数缺失
- Demo 里面的 JS 为基本的数据收集功能示例, 各家第三方或媒体在实际联调时请用各自的 JS 在 XML 配置本地 JS 以及远程 JS 地址
- 关于 SDK 对于缓存数据的处理是提供了缓存接口供 JS 调用, JS 可以把需要缓存的数据通过调用相关接口让 SDK 缓存。但是建议 JS 不要一次性向 SDK 发送太多数据进行缓存, 如果缓存不及时可能导致数据丢失
- 如果本地离线没有添加 JS 文件, 并且缓存也不存在, 只能依赖于在线获取 JS:
 - 如果在此次可见曝光监测结束前, JS 尚未加载完全, 则本次可视化曝光会失败 (SDK 没有缓存数据, JS 不存在的话, 每次的点都空发出来)
 - 如果 JS 一直未加载成功, 则在下一次发生可见曝光监测时 SDK 会再去加载 JS, 若还未加载完全, 则本次可视化曝光依然失败

建议本地配置 JS 文件

- SDK 会提供停止监测的接口, JS 调用此接口通知 SDK 对某个对象的可见曝光监测。如果不停止监测, SDK 会不断的向 JS 发送数据, 性能资源会

有影响；

- JS 把停止监测的协议回传给 SDK 时，SDK 得到指令后，从队列中移除该对象的监测，这个过程 SDK 还会有机会一直发送数据给 JS 端，所以对 JS 的代码在 Stop 某对象的可见曝光监测后，SDK 再向 JS 发送该对象的数据时，JS 应该丢弃；
- 如果配置文件同时配置了多个公司的 JS 文件，并同时存在多个公司的 JS 可视化监测，会生成多个 WEBVIEW，对性能有一定要求；
- Android 端由于不同厂商的定制版本可能使用的 WEBVIEW 内核版本不一，JS 代码最好不要使用过于先进的语法，以免在低端机型上出现加载 JS 失败，导致监测无效，需要各个公司对 JS 部分有详细的测试；