

ironSource 可玩广告开发指南 V1.1

更新时间：2018 年 8 月 22 日

目录

1. ironSource可玩广告对接流程概览
2. 附录：可玩广告内容设计最佳实践
3. 第一步：技术对接
4. 第二步：使用ironSource网页测试工具进行自测
5. 第三步：向ironSource交付可玩广告素材
6. 第四步：安装ironSource测试APP进行自测
7. 第五、六、七步：交付最终版本后经ironSource测试后上线

版本说明

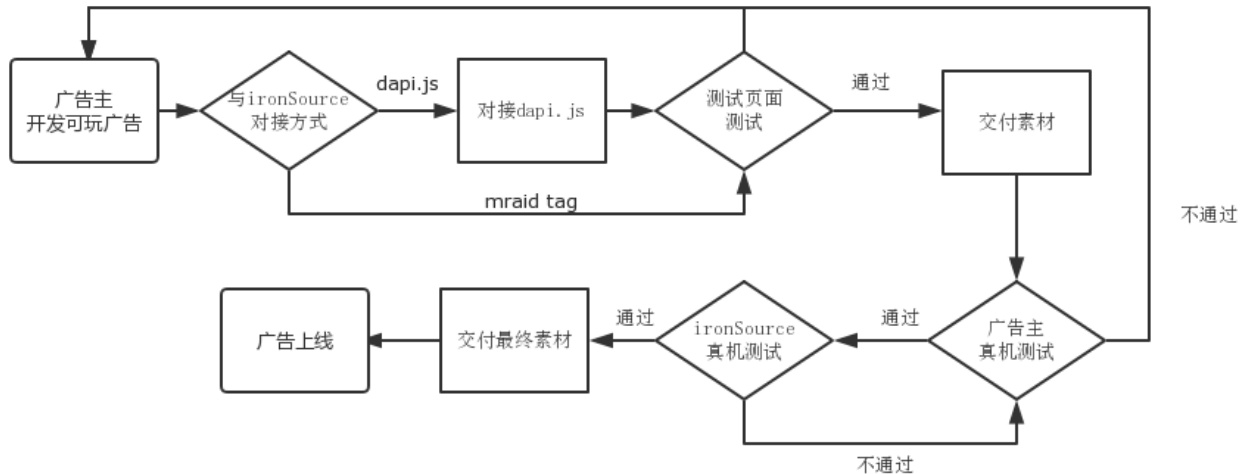
2018.08.09 优化了中文翻译 V0.0

2018.08.16 优化了dapi.js 部分的说明 V0.5

2018.08.22 添加完整测试流程说明 + 修改部分细节说明 V1.0

2018.10.23 增加了general link的说明

1. ironSource可玩广告对接流程



第一步、技术对接

可玩广告开发完成后，广告主按照本文档第三部分《技术对接》接入ironSource API(dapi.js)（如您使用的是mraid tag，请直接进入第二步）

第二步、广告主使用ironSource网页测试工具进行自测

使用ironSource测试页面对完成开发的url/mraid tag进行测试。使用url进行测试时需要使用https链接。您需要通过所有页面上的测试项才可进入下一步

第三步、广告主交付可玩广告链接与素材

向您的ironSource客户经理交付所有的素材（.html,.js,.css和图片素材）

第四步、广告主安装ironSource测试APP进行自测

ironSource客户经理会向您提供ironSource测试APP和测试Application Key。您需要在尽可能多的机型和操作系统版本上进行测试，并且告知您的客户经理您已测试的机型和操作系统版本。

第五步、广告主交付最终版本可玩广告链接与素材

当ironSource判定当前版本可以进行最后QA后，您需要向您的ironSource客户经理交付最终版本的所有的素材（.html,.js,.css和图片素材）

第六步、ironSource进行QA

ironSource会对您最终交付的素材和链接进行真机测试

第七步、ironSource安排广告上线

2. 附录：可玩广告内容设计最佳实践

介绍

可玩广告是向用户提供交互式游戏片段的广告，也称为“小游戏”。广告最后有一个用户自愿进行的行为（例如，安装应用程序），通常时长不到 30 秒。

游戏设计建议

- **不要复制您的游戏**您的可玩广告不应该是您游戏的复制品。而应该是一个简化版本。
- **将难度级别设置为容易-中等**
- **降低用户学习成本**必须在3-4秒内使用户了解如何与可玩广告互动。通常可以向用户提供手指游标，并指示用户滑动屏幕的位置。或者突出显示关键按钮，提供提示，并提供简洁明确的说明。

加载页面

- 加载可播放广告时，**必须包含加载动画**（例如转圈、进度条）
- **包含应用程序图标或LOGO以提高品牌知名度**

Call to Action(动作唤起)文本

- **在游戏过程中** - “现在安装”
- **Endcard** - “继续游戏”或“免费下载”

3. 第一步、技术对接指南

对接

有两个对接选项：

1. [MRAID Tag](#)（直接进入第二步）

或

2. [ironSource API](#)（推荐选用）

❗ 重要

- 为了运行广告，必须使用 **ironSource API（推荐）** 或 **MRAID Tag**
- 必须支持横屏和竖屏
- 当网络环境不佳时，广告必须包含加载页面（loader screen）
- 广告将会使用ironSource自带计时器和X按钮，**请勿自行添加相关组件**
- 您交付的链接必须为**HTTPS**
- 若您使用的并非MRAID Tag，则.html文档中不可以包含 `<script src="mraid.js"></script>`

❗ 使用 ironSource API (dapi.js)

- 需以Base64将所有素材和文件（JS/CSS/图像）嵌入单个 HTML 文件中
- 文件最大为 2MB
- 当用户点击“下载”按钮时，请使用以下功能：**dapi.openStoreUrl()**；

托管

ironSource 可以托管您的可玩广告，您也可以将可玩广告置于自己的服务器上

大小限制

可玩广告所有的素材加起来最大包体大小为 2MB

方向

可玩广告应具有交互功能，支持横屏和竖屏，并支持下列比例：

1. 全屏
2. 16:9
3. 4:3

内存消耗

在批准新的素材之前，我们将对内存消耗进行严格的测试。请注意，使用第三方游戏引擎通常会增加迷你游戏的 RAM消耗 和包体大小

最佳实践

1. 所有图形素材均应以 sprite sheet 的形式嵌入同一个文档、JS 和 CSS 适当地缩小
2. 减少网络调用 - 在主 HTML 文件内以 Base64 嵌入 JS/CSS/图像文档

ironSource Campaign API 对接 (dapi.js)

dapi.js API - 设置指南

可播放和交互式广告通过 iFrame 在 ironSource SDK 内运行。**您需要在您的 HTML 代码中使用 dapi.js 来与我们的 SDK 进行通信**（这对于 ironSource 接收屏幕大小、广告可见度、发送点击事件等基本信息是必须的）请确保您阅读并参照本节所有备注。

示例：

第 1 步：在 <head> 标签内添加下面的代码

```
<script>
function getScript(e,i){var
n=document.createElement("script");n.type="text/javascript",n.async=!0,i&&(n.onload=i),n.src=e,docum
ent.head.appendChild(n)}function parseMessage(e){var
i=e.data,n=i.indexOf(DOLLAR_PREFIX+RECEIVE_MSG_PREFIX);if(-1!==n){var t=i.slice(n+2);return
getMessageParams(t)}return{}}function getMessageParams(e){var
i,n=[],t=e.split("/"),a=t.length;if(-1===e.indexOf(RECEIVE_MSG_PREFIX)){if(a>=2&&a%2===0)for(i=0;a>i
;i+=2)n[t[i]]=t.length<i+1?null:decodeURIComponent(t[i+1])}else{var
o=e.split(RECEIVE_MSG_PREFIX);void 0!==o[1]&&(n=JSON&&JSON.parse(o[1]))}return n}function
getDapi(e){var i=parseMessage(e);if(!i||i.name===GET_DAPI_URL_MSG_NAME){var
n=i.data;getScript(n,onDapiReceived)}}function invokeDapiListeners(){for(var e in
dapiEventsPool)dapiEventsPool.hasOwnProperty(e)&&dapi.addEventListener(e,dapiEventsPool[e])}function
onDapiReceived(){dapi=window.dapi,window.removeEventListener("message",getDapi),invokeDapiListeners(
)}function
init(){window.dapi.isDemoDapi&&(window.parent.postMessage(DOLLAR_PREFIX+SEND_MSG_PREFIX+JSON.stringify(
{state:"getDapiUrl"}),"*"),window.addEventListener("message",getDapi,!1))}var
DOLLAR_PREFIX="$$",RECEIVE_MSG_PREFIX="DAPI_SERVICE:",SEND_MSG_PREFIX="DAPI_AD:",GET_DAPI_URL_MSG_NA
ME="connection.getDapiUrl",dapiEventsPool={},dapi=window.dapi||{isReady:function(){return!1},addEven
tListener:function(e,i){dapiEventsPool[e]=i},removeEventListener:function(e){delete
dapiEventsPool[e]},isDemoDapi:!0};init();
</script>
```

第 2 步：使用下面列出的方法/事件

dapi.js API - 方法

dapi.isReady

描述：检查 dapi 是否准备就绪（如果没有，则不应使用 dapi 方法播放并运行可玩广告）

传递参数：无

返回值：布尔值（如果就绪 - 将返回 true，如果未就绪 - 将返回 false）

dapi.isViewable

描述：如果广告目前可见，则返回 true（例如，如果用户按下返回主页按钮，则广告不可见）

传递参数：无

返回值：布尔值（如果可见 - 将返回 true，如果不可见 - 将返回 false）

dapi.getScreenSize

描述：返回确切的屏幕大小

传递参数：无

返回值：对象（例如：{width: 200, height: 400}）

dapi.openStoreUrl

描述：当用户点击“安装”或“下载”按钮时，调用此方法可将用户引导至应用商店。此方法还将上报点击。

传递参数：回调 [function] 可选，参数 [object] 可选

返回值：无

dapi.open

概述：调用此方法打开 URL 链接

传递参数：url [string] 必选，参数 [object] 可选，回调 [function] 可选。

返回值：无

dapi.js API - 事件和监听器

dapi.addEventListener

描述：添加事件监听器（请参阅下面的监听器列表）

传递参数：事件名称 [string] 必选，回调 [function] 必选

返回值：无

dapi.removeEventListener

描述：移除事件监听器（请参阅下面的监听器列表）

传递参数：事件名称 [string] 必选，回调 [function] 必选

返回值：无

事件（为这些事件添加监听器）：

- **ready** - 监听此事件以确定 dapi.js 何时准备就绪并可供使用
- **viewableChange** - 监听此事件以了解广告何时可见/不可见

- **adResized** - 监听此事件以重新计算广告UI

dapi.js API - 示例代码

//加载游戏，但在广告可见之前不要启动游戏

```
window.onload = function(){
    (dapi.isReady()) ? onReadyCallback() : dapi.addEventListener("ready",
onReadyCallback);
    //您可以在此处放置与 dapi 逻辑无关的其他代码
};

function onReadyCallback(){
    //无需再监听此事件
    dapi.removeEventListener("ready", onReadyCallback);

    if(dapi.isViewable()){
        adVisibleCallback({isViewable: true});
    }

    dapi.addEventListener("viewableChange", adVisibleCallback);
    dapi.addEventListener("adResized", adResizeCallback);
}

function adVisibleCallback(event){
    console.log("isViewable " + event.isViewable);
    if (event.isViewable){
        screenSize = dapi.getScreenSize();
        //启动或恢复广告
    } else {
        //暂停广告并静音
    }
}

function adResizeCallback(event){
    screenSize = event;
    console.log("ad was resized width " + event.width + " height " + event.height);
}

//当用户点击下载按钮时 - 使用 openStoreUrl 函数
function userClickedDownloadButton(event){
    dapi.openStoreUrl();
}
```

❗ dapi.js API - 备注（请务必遵循）

- **监听器** - 您甚至可以在 window.onload 之前添加事件监听器
- **viewableChange** - 使用此事件以了解广告何时显示，以便启动/暂停可玩广告
 - 您应该在显示广告之前在后台加载所有素材 - 但仅在广告可见时才能开始可玩广告
 - 当广告不可见时（用户点击主页按钮/关闭广告） - 您应该暂停可玩广告并静音
- **adResized** - 当调整屏幕大小时，使用此事件以接收通知（例如，横屏到竖屏）
- **dapi.getScreenSize()** - 当广告位于后台或隐藏时，此事件将返回 {width: 0, height: 0}。建议仅在广告可见时使用。
- 您必须在运行游戏脚本之前调用**dapi.isReady();**和**dapi.getScreenSize();**以确保通过QA
- 当用户点击下载按钮时，请使用**dapi.openStoreUrl();**以确保通过QA。
- 您可以使用下列链接来测试您的html是否符合要求：
<https://demos.ironsrc.com/test-tool/?adUnitLoader=dapi&mode=testing>

4. 第二步、测试页面自测

ironSource测试页面

ironSource测试页面可以帮助您测试您的可玩广告是否调用所有必须的方法以及是否具备必须的交互功能。测试界面地址如下：

<https://demos.ironsrc.com/test-tool/?adUnitLoader=dapi&mode=testing>

测试步骤

- 1.您需要在页面上方将选项选至testing，并根据交付素材情况选择Mraid（仅在交付mraid tag的情况使用）或者DAPI（交付素材为url时通用）
- 2.输入您的url或者mraid tag并按下测试
- 3.待广告加载出来后，通过左侧工具栏对广告进行转向和缩放测试
- 4.游戏结束时，点击下载按钮。您需要页面所提出的所有的要求才能通过测试。

1. Enter the **HTTPS://** URL of your DAPI ad (.html file) and click 'TEST AD'
2. The ad should pass **all tests** listed on the **pre-flight checklist** at the bottom
3. **Rotate the device** to ensure different ad orientations work correctly
4. Use the **Tablet/Phone icon switch** to ensure that the ad runs on 4:3 screen ratio correctly

Guidelines

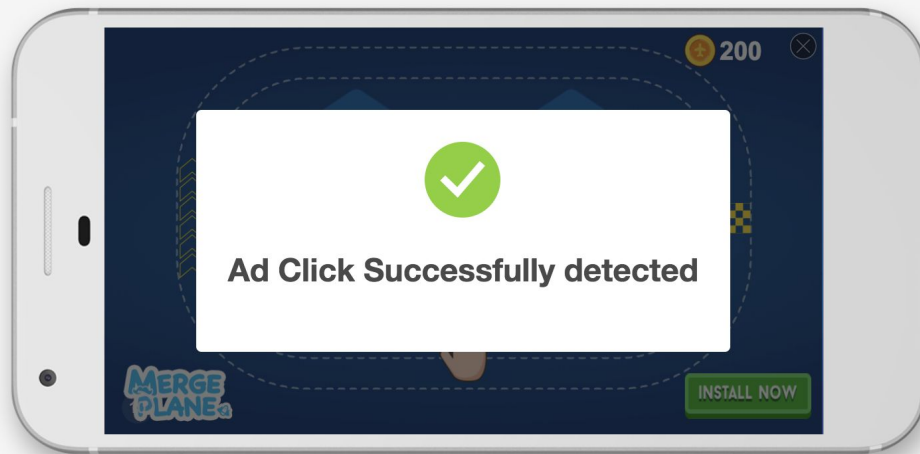
- Ad must support [DAPI protocol](#)
- Ad URL must be in **HTTPS://** (not **HTTP://**)
- Ad must embed all assets and files (JS/CSS/Images) in base64 in a single inline HTML file
- Network requests should be **absolute URLs** instead of relative URLs and **HTTPS** instead of HTTP
- The maximum ad size is **2MB**
- Ad should support **both** landscape and portrait orientations (recommended)
- ironSource will render the timer and the close button. Do not add your own
- Ad must include within <head> the DAPI script from the [implementation step from DAPI protocol](#)
- Clicks leading to **store** page (Google Play Store, Apple App Store) must use `dapi.openStoreUrl()`. To redirect to a different **landing page**, use `dapi.open(<your non-store url>)`



Paste your hosted file URL here

TEST AD

使用DAPI时之成功界面



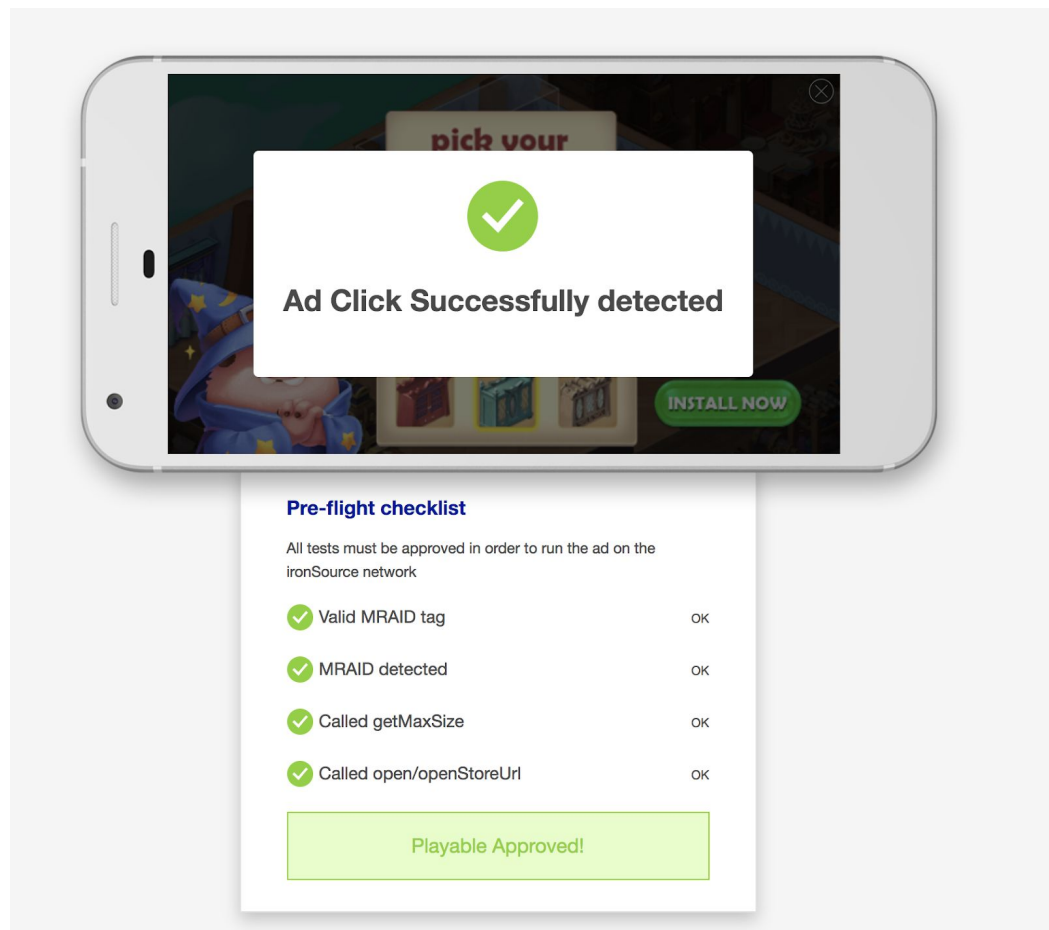
Pre-flight checklist

All tests must be approved in order to run the ad on the ironSource network

- | | |
|----------------------------|----|
| ✓ Valid URL | OK |
| ✓ Called isReady | OK |
| ✓ Called getScreenSize | OK |
| ✓ Uses native close button | OK |
| ✓ Called open/openStoreUrl | OK |

Playable Approved!

使用Mraid Tag时之成功界面



4. 第三步、交付可玩广告材料

向您的ironSource客户经理交付所有的素材：playable链接，.html文档,.js文档（如果有），.css文档（如果有）和图片素材（如果有）

如您有多个待测试可玩广告，**请交付一个通用url**，并在广告上线之后将链接内容替换为下一个待测playable，以便快速进行真机测试。

通用链接最佳实践：

1. 样例：<https://gamecompany.com/playable/ironsource-test/index.html>
2. 链接内容**请勿包含**游戏名或者是素材名，以免引起混淆
3. 链接必须为https

5. 第四步、安装ironSource测试APP进行自测

测试步骤

1. 安装ironSource测试APP（安卓测试需使用VPN）

请在尽可能多的机型（新、旧手机、平板）和操作系统上进行测试

iOS 测试应用程序 -<https://ssa.public.s3.amazonaws.com/OTA/IronSource-DemoApp-New/index.html>

Android 测试应用程序 -<https://drive.google.com/file/d/1rZ6Qs3KqIKuuQzT-dnGpJK231ycVgnzs/view?usp=sharing>

- 2.使用Application Key测试

ironSource客户经理/ironSource委托之第三方将向您提供一组Application Key（iOS和安卓各一个）

您只需要在测试程序输入Application Key后选择launch，并点选INIT。如加载成功，即刻点击Show RV查看真机测试效果。（注：安卓设备测试时需要使用VPN）

- 3.测试标准

下列是ironSource进行QA使用的标准，请务必确认您达到下列标准

- a. 可玩广告大小不超过2mb
- b. 广告加载时应有转圈/进度条/进度百分比
- c. 用户离开当前页面（前往商店、按下home键或者锁屏）后再重返时，游戏应继续进行
- d. 设备转向时界面切换流畅（水平、竖直）
- e. 用户跳出广告界面时（前往商店、按下home键或者锁屏）游戏画面和声音应中止，并在用户返回游戏界面时重启
- f. 所有手势动作表现正常
- g. 没有闪退、卡顿的情况出现

- 4.按下下载按钮

当您按下下载按钮时，ironSource会将用户带至预设之下载界面，跟操作系统不同。
若商店中显示的APP与您的APP不匹配，无需担心，正式上线时会设置成正确的下载界面。

5. 上报测试机型和操作系统版本

待您完成真机测试且确认没有问题时，请向您的ironSource客户经理提供您测试过所

有的机型和操作系统版本。

6. Debug

若您在真机测试中反复出现错误，您可以使用我们的测试程序项目进行debug

iOS

<https://drive.google.com/file/d/1pMwYKWd6uei2WazHfAE8JokybuDYhA/view?usp=sharing>

Android

https://drive.google.com/file/d/1FiU9Ean5Te3D3C_iclH8MI1r4bLhJqjb/view

如您仍无法解决问题，需要我们的支持，请发送一封包含下列内容的邮件给您的客户经理：

1. Bug重现步骤
2. 预期结果
3. 漏洞重现的设备和 OS 版本
4. 视频/屏幕截图

5. 第五、六、七步：交付最终版本，ironSource测试上线

真机自测通过后，向您的ironSource客户经理交付所有的素材：.html文档,.js文档（如果有）,.css文档（如过有）和图片素材（如果有）

ironSource将在内部真机测试通过之后，安排广告上线。

上线完成后，若您有下一个待测的可玩广告，请直接将通用链接的内容更改为下一个待测 playable（需通过PTT），即可使用先前ironSource客户经理交付的appkey继续进行真机测试。（重复第四~七步）

如有任何疑问，请联系您的客户经理。