

魔窗mLink JS集成文档

1. 引用mLink JS文件.

特别提示: 请直接使用此链接, 不要将此文件下载到您项目中, 否则, 我们在更新数据结构或API接口后有可能导致mLink失效。

```
<script src="https://static.mlinks.cc/scripts/dist/mlink.min.js"> </script>
```

2. 在HTML页面中准备一个或多个用于打开APP的a元素

一般开发者会使用CSS样式重置这个链接的外观, 使其看起来像一个按钮, 建议使用“打开APP”、“在APP中购买”、“在APP中阅读”等对用户有实际引导意义的文案, 请尽量避免使用“下载APP”, 在以往的用户使用习惯中我们发现使用“下载APP”的链接对比使用“打开APP”的链接的点击率会低好几倍;

```
<a id= "btnOpenApp">打开 APP</a>
```

3. 魔窗后台配置好mLink必要参数并且生成一条短链

请在后台配置mLink服务获取短链接, 短链看上去是这样的: <https://a.mlinks.cc/ABCD>, 其中ABCD为短链KEY;

4. 初始化mLink

```
new Mlink({
  mlink:'https://a.mlinks.cc/ABCD',//短链地址
  button:document.querySelector('a#btnOpenApp')
});
/* ----- 或 ----- */
var link='https://a.mlinks.cc/ABCD',//短链地址
var btn_1=document.querySelector('a#btnOpenApp1');
var btn_2=document.querySelector('a#btnOpenApp2');
var btn_3=document.querySelector('a#btnOpenApp3');

var options = [
```

```
    {
      mlink: link+'?name=1',
      button: btn_1
    }, {
      mlink: link+'?name=2',
      button: btn_2
    }, {
      mlink: link+'?name=3',
      button: btn_3
    }
  ];
```

```
new Mlink(options);
```

5. 在H5页面中使用默认短链接进行动态传参

短链接可以从外部直接唤起App指定页面，那么就可以应用在H5页面中，从而实现在H5中看到什么内容，在App中看到对应的内容。



而且当有很多同类的H5内容时, 可以**只使用一个默认短链接做到动态传递参数**, 在App对应打开该参数的对应内容。

实现方法:

这里, 给出用带有“打开app” banner链接的代码例子。H5页面里, 请务必使用我们的mLink JS 代码加载按钮的点击事件, 而不是直接使用短链, 否则即使是iOS9 并且使用了Universal Link, 也会出现一个中间页面才能唤醒App, 而不是一键唤醒。

```

<!DOCTYPE html>
<html>
  <head>
    <title>唤醒demo</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0,
    minimum-scale=1.0, maximum-scale=1.0, user-scalable=no">
    <link rel="shortcut icon" href="icon.icon">
  </head>
  <body>
    <div class="main">
      <div class="banner">
        <a id="appopen" href="">下载app</a>
      </div>
    </div>
    <script type="text/javascript" src="https://static.mlinks.cc/scripts/dist/mlink.min.js">
    </script>
    <script type="text/javascript">
      new Mlink({
        mlink: 'http://a.mlinks.cc/Aadb?bannerlink=123', //bannerlink为魔窗后台配置的短链参数
        button: document.querySelector('a#appopen')
      });
    </script>
  </body>
</html>

```

6. options 选项

```

{
  mlink: "短链 KEY",
  button: document.querySelector('a#btnOpenApp'),
  autoLaunchApp : false,
  autoRedirectToDownloadUrl: true,
  downloadWhenUniversalLinkFailed: false,
  inapp : false,
  params: {}
}

```

1、options.mlink

类型: String

必填

描述: 短链地址或 KEY, 推荐使用短链 KEY, 如您的短链是“https://a.mlinks.cc/ABCD”, 那么 options.mlink 的值就是 **ABCD**

注意: 旧版本的 mLink JS API 支持 options.mlink 直接传入一个短链的 URL, 而在

最新版本中已经不再推荐直接传短链的 URL，但是依然会兼容，更好的做法是传短链 KEY。

2、options.button

类型: HTMLAnchorElement (传说中的<a/>标签)

必填

描述: 打开 APP 的 a 标签，一般会使用 CSS 定义 a 标签的外观，使其看起来像一个按钮；

注意: options.button 是必要参数并且只能是 a 标签，不能是 button、input 或其他 HTML 元素，否则 Mlink 无法初始化成功。为什么 button 是必要参数？因为很多浏览器禁止通过 JS 直接打开 APP，必须通过用户手动点击才能打开应用；

3、options.autoLaunchApp

类型: Boolean

选填

默认: false

描述: 此选项配置为 true 时，用户打开您集成了 mlink.js 的页面时，在支持自动打开 APP 的浏览器和设备中，将自动唤起 APP；但遗憾的是有一些浏览器因为限制通过 JS 直接唤起 APP，即使此选项设为 true 也不能自动打开应用，此时需要用户手动点击由 options.button 指定的链接才能打开 APP；

提示: 此选项设置为 true 此时会出现以下情况，请您自行斟酌是否有必要设置为 true；

①在 Android 以及 iOS 低于 9.0 的系统中，微信、QQ、QQ 浏览器，如果在魔窗后台的 mLink 高级设置中开启了"应用宝跳转"，可以自动唤起应用，如果没有开启"应用宝跳转"，会重定向到一个"请在浏览器中打开"的引导页面；

②在微博内，无论是否安装过 APP，都会重定向到一个"请在浏览器中打开"的引导页面；

③在当前设备未安装 APP 时，iOS 9+ 的 Safari 会提示："Safari 打不开该网页，因为该网址无效。"，此提示无法通过 JS 代码阻止；

④在魔窗后台开启了 Universal Link 开关时，不能自动唤起 APP，因为 Universal Link 必须由用户点击链接才能唤起 APP；

⑤Android 系统的 Chrome、QQ(不是 QQ 浏览器)和部分基于 Chrome 内核的浏览器，不能自动唤起应用，因为这些浏览器或 Web 容器需要用户手动点击才能唤起 APP；

⑥在已安装 APP 情况下，UC、QQ 浏览器和 iOS 9+的 Safari 内，在您的页面加载完成后会自动弹出"是否打开 APP"的对话框，用户点击确认按钮后能打开 APP，点击取消按钮后会无效；

⑦在一些不支持 Scheme 和 Universal Link 的浏览器或 Web 容器内, 都会因为无法打开 APP 而引导用户去往下载地址; 如何引导用户下载安装根据 options.autoRedirectToDownloadUrl 的选项走不同的流程;

4、options.autoRedirectToDownloadUrl

类型: Boolean

选填

默认: true

描述: 此选项值默认为 True, 在打开 APP 失败的情况下(包括未安装 APP、APP 没有实现 Deeplink、浏览器不支持 Deeplink 等情况), 自动跳转到下载地址; 如果为 false, 在打开 APP 失败时, Android 和 iOS7、8 会继续停留在您的页面并弹出一个包含下载按钮的 Dialog; iOS9 以上系统并且开启了 Universal Link 时, 如果打开 APP 失败, 系统仍然会跳转: 在微信内会显示"请在浏览器打开的提示"、在 Safari 中会跳转到下载页面;

5、options.inapp

类型: Boolean

选填

默认: false

描述: 当您的 HTML 页面被嵌入在 APP 中的 Webview 时, 因为 APP 中调用的 Webview 与常规浏览器的功能有差异, mLink 所依赖的 Universal Link 在 Webview 中会有缺陷, 但是 mLink 又无法准确检测当前打开的环境是 Webview 还是 APP, 所以, 如果在 APP 中的 Webview 使用 mLink 时, 为了确保 mLink 能在 iOS 9.0 以上系统中的 APP 也能正确唤起指定的 APP 界面, 请将此选项设置为 true, 设为 true 时, mLink 会优先使用 Custom URI Scheme 来唤起指定页面。

6、options.params

类型: Object

选填

描述: 传给 APP 的参数, 这些参数必须是您在魔窗后台的 mLink 服务配置中设置好的参数占位符, 否则会被过滤掉; 如果 APP 已安装, 那么唤起应用时从打开 APP 的 URI Scheme 或 Universal Link 中就能获得参数, 如果是第一次安装, 这些参数可以通过魔窗 SDK 的 API 从我们的服务器取回; 关于 SDK 如何实现 mLink 请参考:

[iOS mLink 集成文档](#)

[android mLink 集成文档](#)

示例:

您在后台 mLink 服务配置中填的 URI Scheme: myapp://path/article?articleId=**id**

options.params: { **id**: '123456' }

最终的 URI Scheme 是这样的: myapp://path/article?articleId=**123456**

7、options.cparams

类型: Object

选填

描述: 传给 APP 的参数, 用户可以通过这个参数从 H5 页面传递一些参数到 APP 内进行使用。这些参数不是魔窗后台的 mLink 服务配置中设置的参数占位符。

注意: SDK V3.9.20161010 之后的版本支持。

8、options.tparams

类型: Object

选填

描述: 使用这个参数追踪来源, 现在支持的参数如下 (非下面的参数无效) u_id: 来源用户 ID

示例: tparams: {"u_id": "uid-123456"}

注意: SDK V3.9.20161010 之后的版本支持。