通过移动浏览器启动 APP 的技术说明

一、URL

URL 也被称为网址，全称 Uniform Resource Locator，即统一资源定位符。它遵守以下的语法规则： [1]

```
scheme://host.domain:port/path/filename
```

- scheme - 定义因特网服务的类型（最常见的类型是 http）
- host - 定义域主机（http 的默认主机是 www）
- domain - 定义因特网域名（比如 w3school.com.cn）
- :port - 定义主机上的端口号（http 的默认端口号是 80）
- path - 定义服务器上的路径（如果省略，则文档必须位于网站的根目录中）
- filename - 定义文档/资源的名称

二、URL Schemes

URL 中的服务类型，即 URL Scheme，可大致分为三类：

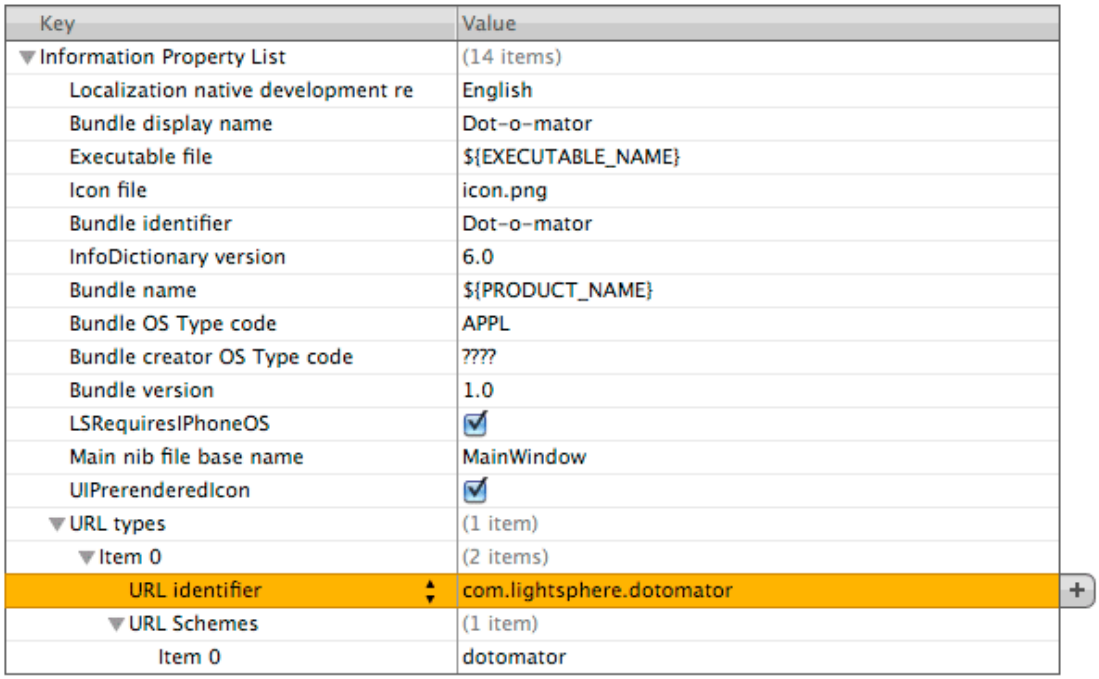
1. 访问网络/本地资源 - http、https、ftp、file ...
2. 启动系统应用服务 - mailto、tel、sms ...
3. **自定义的服务类型** - alipay、taobao、weixin ...

三、Custom URL Schemes

URL 的自定义服务类型 ,是启动 APP 的技术关键。所以从本质上说 ,是“通过自定义 URL Schemes 启动 APP” ，而不限于浏览器。只要是能够执行 URL Schemes 的程序，都可以启动 APP。

第一步：在 APP 内注册自定义 URL Scheme

1. **IOS**：打开 info.plist 文件，添加新的 Key，选择“URL Types”。[2]



- URL identifier 里填写的是下文所指的 pkgKey
- URL Schemes 里填写的就是自定义的 URL Scheme
- 通过实现 application:handleOpenURL 方法，处理 URL Scheme 调用事件：

```
- (BOOL)application:(UIApplication *)application handleOpenURL:(NSURL *)url {
    // handler code here
}
```

2. **Android**：打开 AndroidManifest.xml，添加<intent-filter>元素。[3]

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="(your.pkg.key)"
    android:versionCode="5"
    android:versionName="1.0.5">
    <activity android:name="(YourActivity)"
        android:launchMode="singleTask"
        android:label="@string/app_name"
        android:screenOrientation="portrait"
        android:theme="@android:style/Theme.NoTitleBar.Fullscreen">
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
        <intent-filter>
            <action android:name="android.intent.action.VIEW"/>
            <category android:name="android.intent.category.DEFAULT"/>
            <category android:name="android.intent.category.BROWSABLE"/>
            <data android:scheme="(yourscheme)"
                  android:host="(yourhost)"
                  android:path="/(yourpath)"/>
        </intent-filter>
    </activity>
</manifest>
```

- <manifest> 的 package 属性是下文所指的 pkgKey
- <intent-filter> 的 <data> 元素的 android:scheme 属性就是自定义的 URL Scheme
- 需要设置 android.intent.category.MAIN
- 需要设置 android.intent.category.LAUNCHER
- 需要设置 android.intent.category.BROWSABLE

四、Launch APP

接下来介绍如何在 Javascript 中触发自定义 URL Scheme，从而达到启动 APP 的目的。

第二步： 组装 URL Scheme 字符串

一般来说，只需要输入自定义的 Scheme 加上“://” ，在浏览器中执行，即可启动相应的 APP。例如，微信的 Scheme 是“weixin”，在浏览器里输入“weixin://”并执行，即可启动微信 APP。

而“://”后面，理论上来讲可以是任意字符串，只要在 APP 里有相应的处理代码即可。不过在项目中，最好约定一个语法规则（默认）：

```
scheme://path/params
```

- scheme - 自定义服务类型
- path - 访问路径（操作/方法）
- params - 传递给 APP 的参数

需要注意的是，Android 浏览器在某个版本之后，不再支持上述使用 URL Scheme 的方式，而改为使用 Intent 协议，具体定义如下：[4]

```
intent:
   HOST/URI-path           // HOST is optional
   #Intent;
      package=[string];
      action=[string];
      category=[string];
      component=[string];
      scheme=[string];
   end;
```

据此约定的语法规则（Android Intent）：

```
intent://(path)/(params)#Intent;scheme=(scheme);package=(pkgKey);end
```

- path - 访问路径（操作/方法）
- params - 传递给 APP 的参数
- scheme - 自定义服务类型
- pkgKey - APP 包名

Javascript 伪代码如下：

```
function getSchemeUrl () {
    if (canIntent) {
        return "intent://" + path + params + "#Intent;scheme=" + scheme
+ ";package=" + pkgKey + ";end";
    } else {
        return scheme + "://" + path + params;
    }
}
```

第三步：执行 URL Scheme

Javascript 触发 URL 执行有三种方式：

	方式	优点	缺点
1	window.location	正常浏览器都支持	不能灵活控制 会弹出警告框
2	iframe.src	不会弹出警告框	不能灵活控制 不支持 IOS9 及 Android Intent
3	achor.href	正常浏览器都支持 可以灵活控制	会弹出警告框

最终选择的是以 3 为主，与 2 结合的方案，以下是部分核心代码：

```
<a id="btnLaunch" href="javascript:;">Launch APP</a>
<iframe name="iframe" style="display:none;"></iframe>

<script>
  // 页面加载之后，自动触发启动 APP
  document.addEventListener("DOMContentLoaded", launchAPP, true);
  // 设置 URL Scheme
  btnLaunch.href = getSchemeUrl();
  // 设置通过 iframe 执行
  if (canIntent || isIOS && "9" > iosVersion) {
    btnLaunch.target = "iframe";
  }
  // 启动 APP
  function launchAPP () {
    var clickEvent = document.createEvent("Event");
    clickEvent.initEvent("click", true, true);
    btnLaunch.dispatchEvent(clickEvent);
  }
</script>
```

兼容性说明：

某些第三方浏览器屏蔽或阉割了 URL Scheme 的功能，导致无法启动 APP，也无法通过 Javascript 判断（不过可以穷举），此类问题目前没有很好的解决方案。

五、参考资料

[1] HTML 统一资源定位器

[2] Objective-C: Custom URL Schemes

[3] Intents and Intent Filters

[4] Android Intents with Chrome