

# 移动应用开发集成多包的案例

---

移动应用开发中，分模块开发已经是开发流程中，众所周知的方式。我们经常会遇到在APP的混合开发中的，集成了多个H5的模块。

## 前提

APP中集成的H5模块，需要符合hatom2-cli脚手架[成果物规范](#)，同时基于移动应用开发平台（以下简称，Hatom平台），打包编译服务，才能集成。采用该方案需要提前了解[Hatom平台](#)，以及[hatom2-cli脚手架](#)开发移动应用APP。

## 新建应用

在移动应用开发平台中新建应用（Android 或 iOS），然后选择模板、配置应用信息（包括应用名称、应用图标、是否显示状态栏、是否横竖屏、启动页）、配置证书、上传H5资源包（如有需要）、选择组件、配置开放配置项，最后执行云编译，完成应用构建。

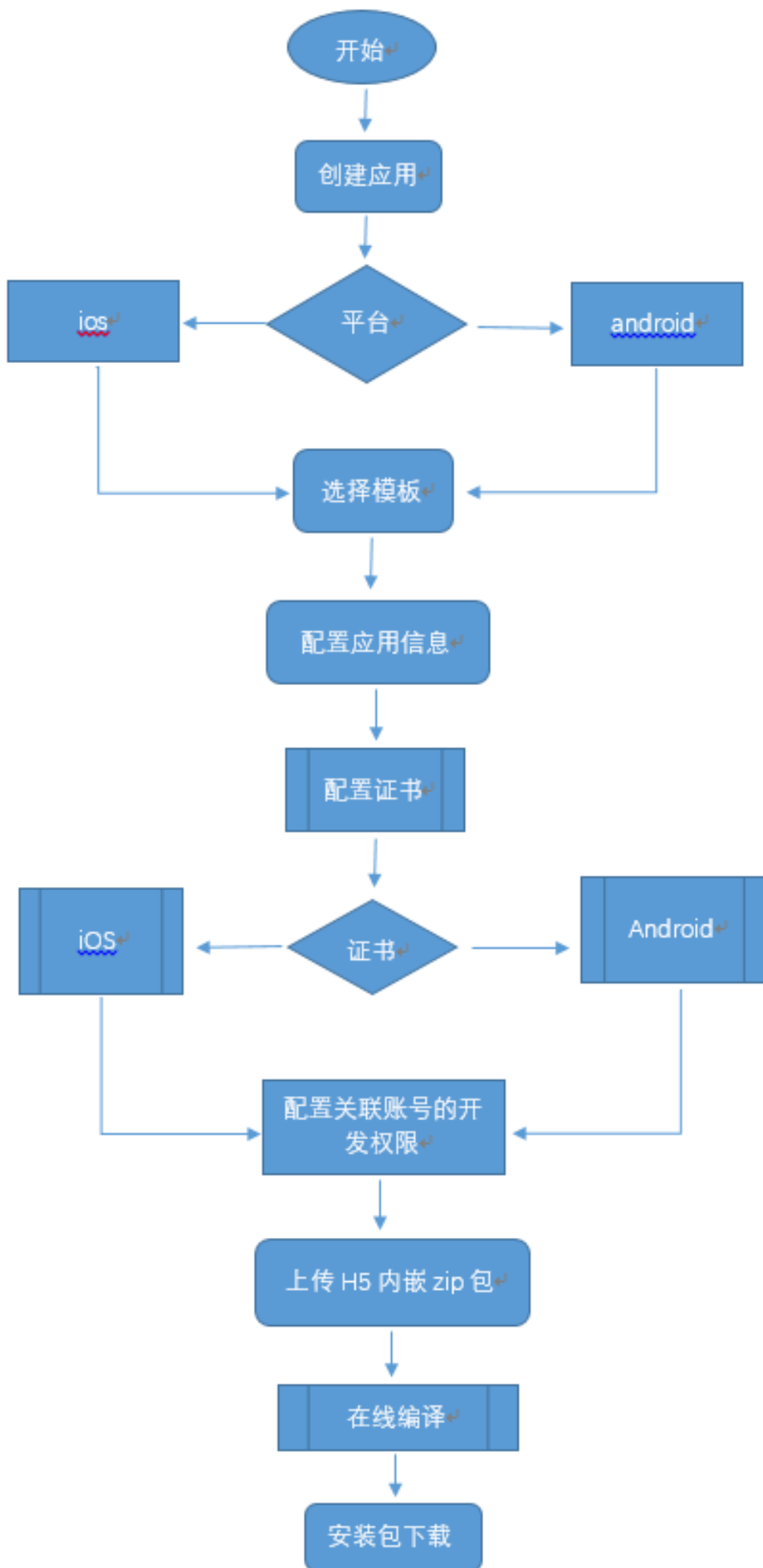


图1.1 新建应用流程

# 上传H5.zip

在应用详情里面，应用开发菜单里面H5资源包模块，如下图：

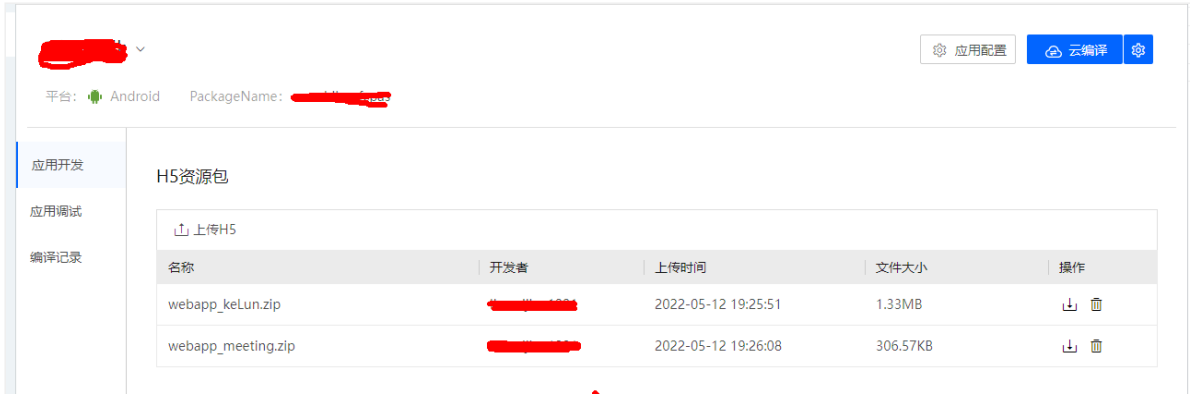


图1.2 应用H5.zip资源包

## H5资源包交互组件

在资源包之间的跳转的交互，当需要从一个H5资源包跳转到另外一个H5资源包中，则需要集成基础引擎插件：

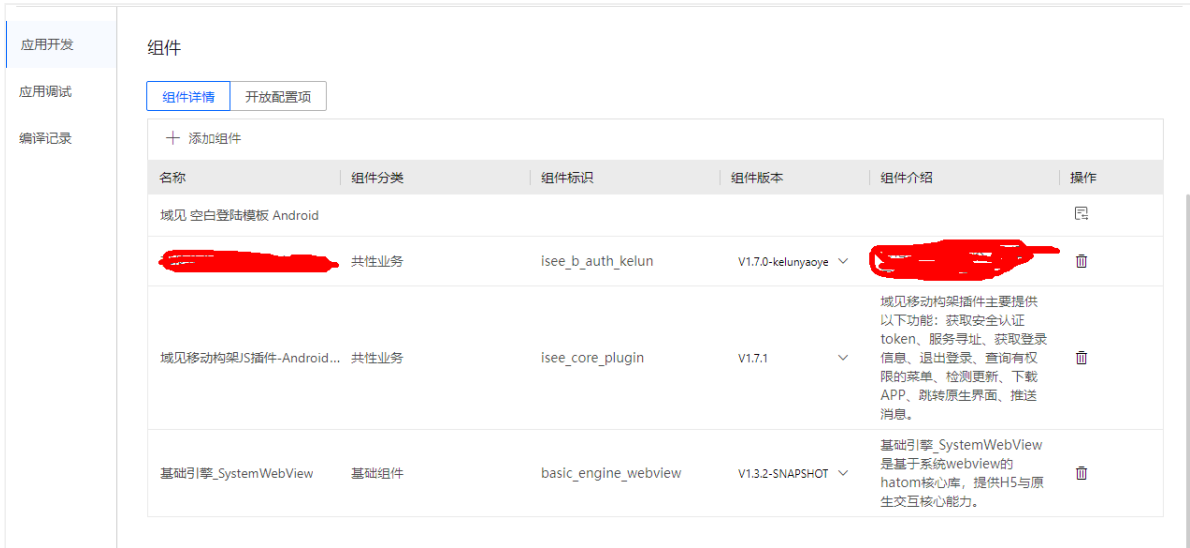


图1.3 基础引擎应用组件

## H5资源包交互API

在资源包之间的跳转的交互的API，则需要查询[帮助文档](#)路由API

## 多页面跳转

常用于[多页面应用](#)模板

使用示例：

```
/** pageName为跳转页面名称 */
const pageName = 'page1.html'
const data = {
  params: {
    'page': pageName
  },
  target: `html:hatom://${pageName}`
};
/** 跳转到page1.html */
hatom.page.pushPage(data);
```

参数说明：

参数	类型	必填	描述
params	Object	是	跳转页面参数，包含页面名称
target	String	是	这个字段中的 hatom 要与 webApp.json 文件中的 h5packCode 保持一致

params参数说明：

参数	类型	必填	描述
page	Object	是	页面名称

图1.4 路由API文档

其中params是透传参数，用于[登录组件](#)的拓展参数routeData，暂无需理解，需要分析的是target参数

html + h5packCode + html文件路径

其中html代表协议，跳转的目标是H5资源包，h5packCode 资源包的唯一标识，html文件路径是成果物中html文件路径名，解析中注意的是pageRouter.json文件中的配置，同时html文件名需要匹配上。

```
{
  "routes": [
    {
      "type": "standard",
      "name": "首页",
      "path": "index.html",
      "tabBar": [],
      "data": {}
    },
    {
      "type": "standard",
      "name": "首页",
      "path": "html路径 + 静态参数", // 此处的参数与target参数中 html文件路径全匹配上
      "tabBar": [],
      "data": {}
    }
  ]
}
```

```
}
```

## 拓展：嵌入远程包

### 方法一

我们在集成的模块提供方，除了是静态资源包之外，还有远程链接，对于这种情况，我们H5组件中需要加载远程的内容嵌入其中。这需要用到一个配置文件webApp.json中的url字段。步骤如下：

先安装hatom2-cli脚手架，通过脚手架建立一个空的工程，找到webApp.json中的h5packCode字段，修改包的标识

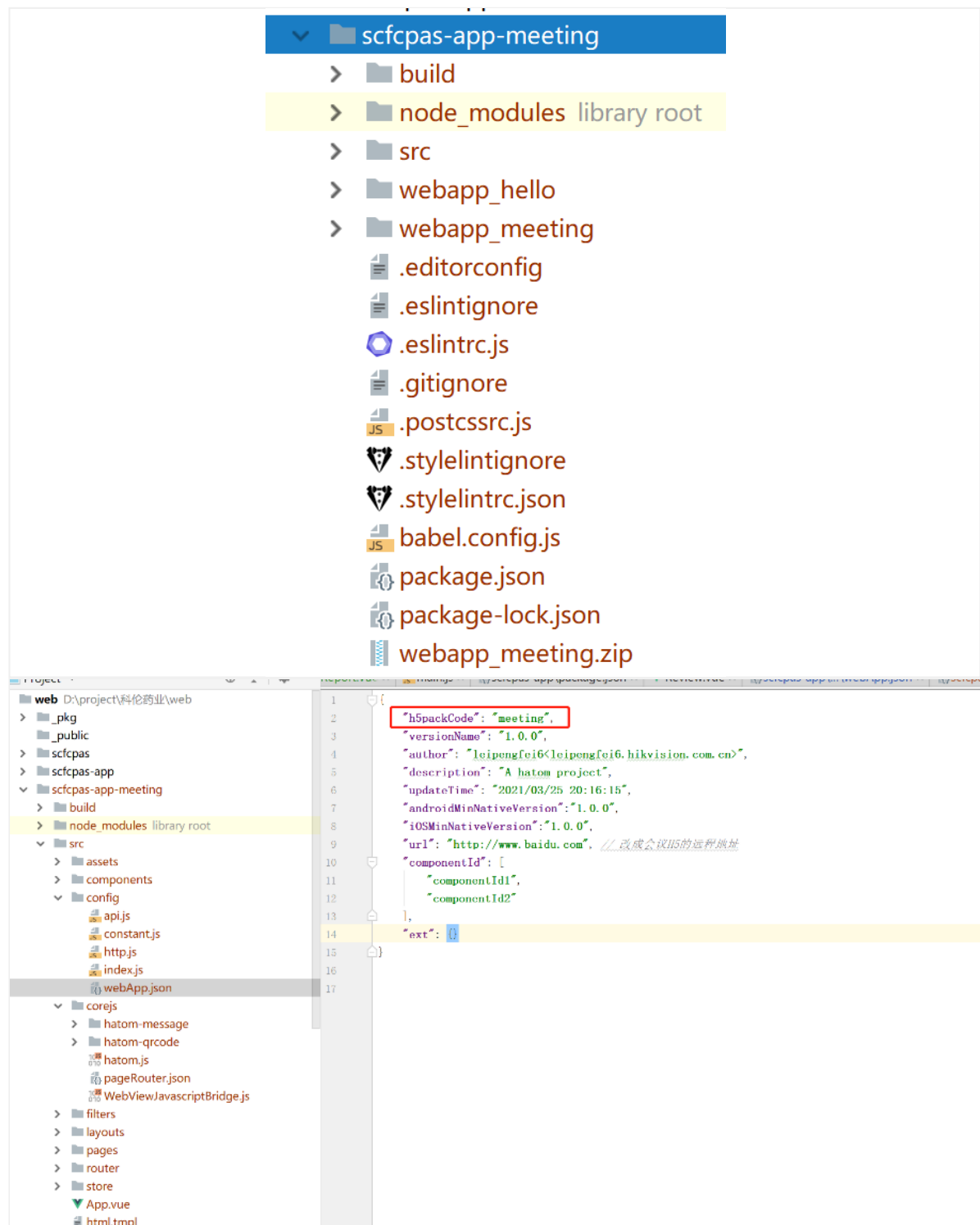


图1.5 新建空包

接着找到webApp.json中的url字段，修改url内容为加载的远程地址：

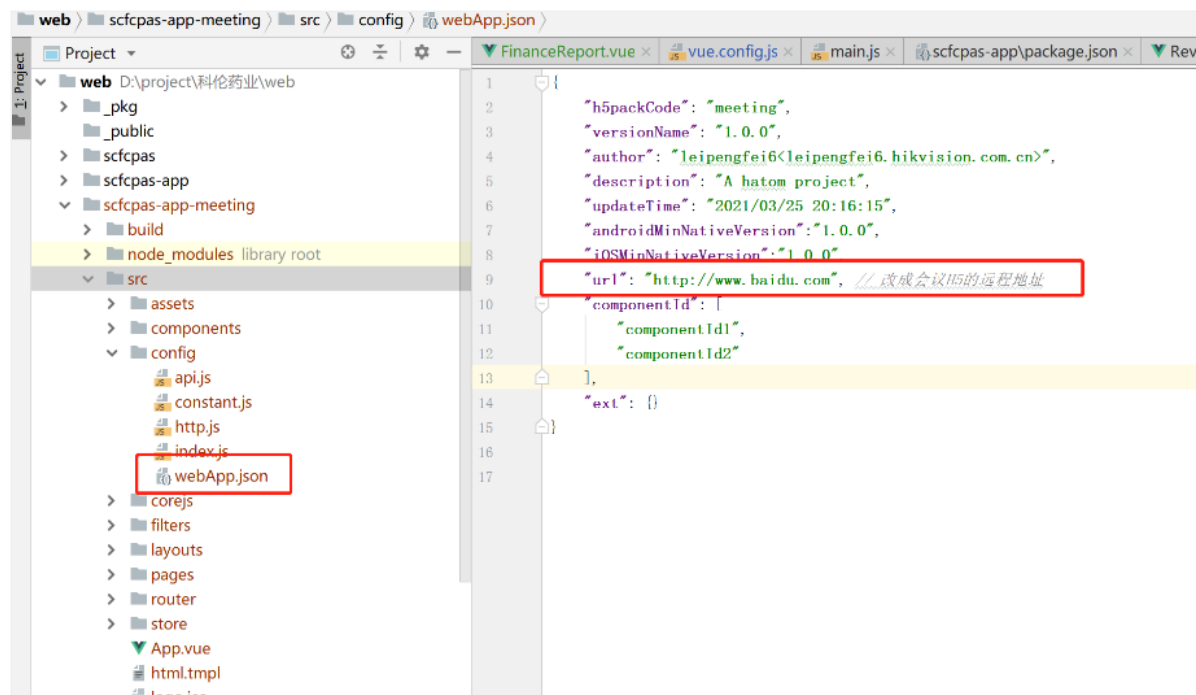


图1.6 加载远程链接

再打包生成zip包成果物

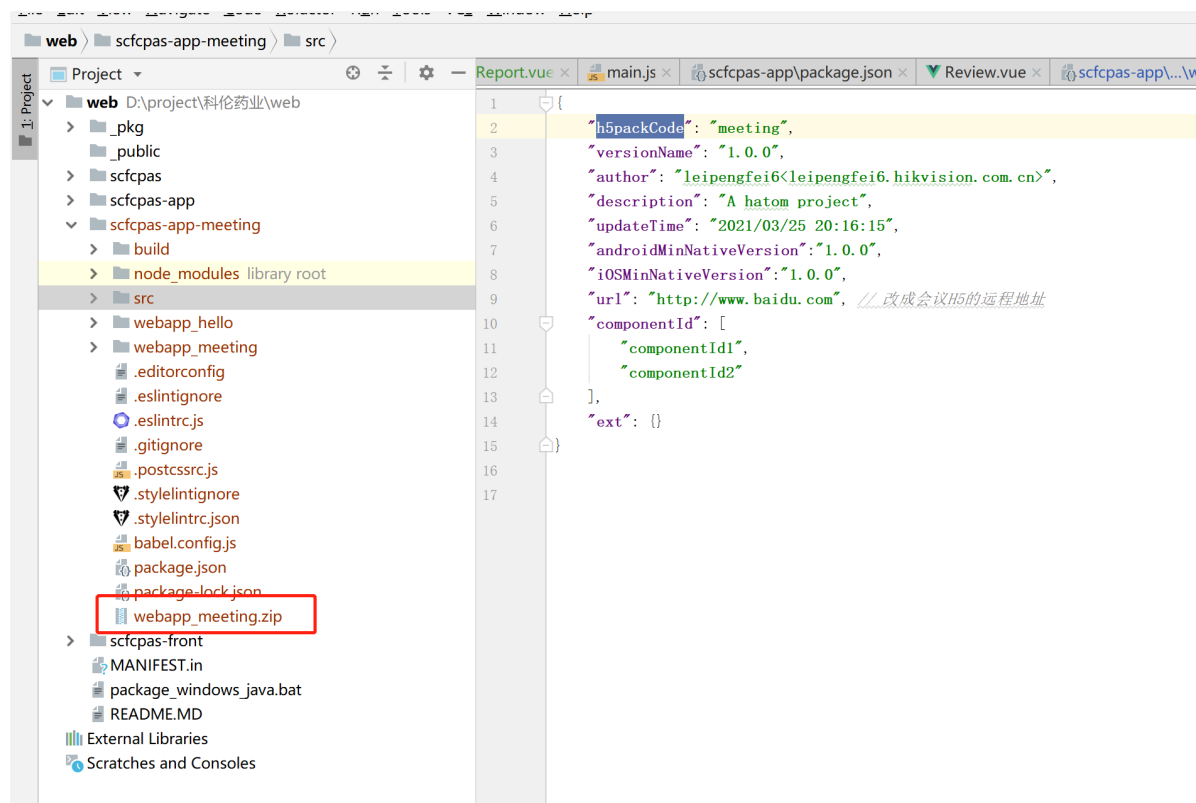


图1.7 成果物zip包

最后上传资源包，见步骤上传H5.zip资源包。

该方法中可以配置白名单，在webApp.json中添加字段whitelist, 见[文档](#)

方法二

相比于方法一，该方法不是很推荐，对于嵌入的内容有很多不可控因素，在保证资源可靠的情况下，可以使用

同样是基于方法一中的API

多页面跳转

常用于多页面应用模板

使用示例:

```
/** pageName为跳转页面名称 */
const pageName = 'page1.html'
const data = {
  params: {
    'page': pageName
  },
  target: `html:hatom://${pageName}`
};
/** 跳转到page1.html */
hatom.page.pushPage(data);
```

参数说明:

参数	类型	必填	描述
params	Object	是	跳转页面参数，包含页面名称
target	String	是	这个字段中的 hatom 要与 webApp.json 文件中的 h5packCode 保持一致

params参数说明:

参数	类型	必填	描述
page	Object	是	页面名称

图1.4 路由API文档

target参数格式需要变化一下 html + http/https:// + 远程路径 如下图:

```
/**
 * 打开页面
 *
 * @param fragment 格式:
 * @param json {
 *   target: "html:webAppName://index.html" 本地相对路径 或 "html:http://www.baidu.com" 远程路径
 *   或者通过 "native:/xxx/xxx" 调用原生页面 "native:/三方app的包名" or "native:/三方app的包名/指定类名"
 *   params: {
 *     "param1": "",
 *     "param2": "",
 *   }
 * }
 * @param function
 */
```

图1.8 参数变化

总结：该文档是基于H5包之间的跳转流程，其他跳转到APP原生界面的API见[帮助文档](#)，跳转第三方应用的[文档API](#)