基于 CesiumJS 的 Android 端三维地球研究与开发

李　想[1,2], 罗　霄[1,2], 特日根[1,2,3]

(1.长光卫星技术有限公司,吉林 长春 130000;2.吉林省卫星遥感应用技术重点实验室,吉林 长春 130000;
3.中国科学院长春光学精密机械与物理研究所,吉林 长春 130000)

摘　要:CesiumJS 作为一个能在二维和三维两个维度展示和管理动态地理数据的开源 JavaScript 库,为遥感领域的开发人员针对 GIS 应用的开发提供了一个成本更低且更易共享的解决方案。本文基于 CesiumJS 和 Kotlin 语言,在 Android 平台上将 Android Native 与 Web 页面相结合,研究并开发了一种集多种遥感数据产品的展示与应用为一体的开发模式,包括多种卫星产品的查看与编辑,基于任意图层的标绘与测量等功能,可以使用户更便捷地在三维地球上访问和使用遥感数据。

关键词:CesiumJS;GIS 应用;Android;三维地球;遥感数据

中图分类号:P228;TP79　　**文献标识码**:A　　**文章编号**:1672-5867(2022)02-0166-05

Research and Development of 3D Earth Based on CesiumJS on Android

LI Xiang[1,2], LUO Xiao[1,2], TE Rigen[1,2,3]

(1.Changguang Satellite Technology Co., Ltd., Changchun 130000, China;
2. Key Laboratory of Satellite Remote Sensing Application Technology of Jilin Province, Changchun 130000, China;
3.Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130000, China)

Abstract:CesiumJS is an open source JavaScript library that can display and manage dynamic geographic data in two dimensions and three dimensions, providing developers in the field of remote sensing with a lower-cost and more easily shared solution for the development of GIS applications. Based on CesiumJS and Kotlin languages. This article combines Android Native and Web pages on the Android platform, and researches and develops a development model that integrates the display and application of multiple remote sensing data products, including viewing and editing of multiple satellite products, plotting and measuring based on arbitrary layers and other functions which can make users more convenient to access and use remote sensing data on the three-dimensional earth.

Key words:CesiumJS; GIS application; Android; 3D earth; remote sensing data

0　引　言

近年来,随着一系列遥感卫星的密集发射,我国遥感技术得到了蓬勃发展,产生的遥感数据也持续增长,为用户积累了长时间序列的对地观测数据,在国土、气象、交通和减灾等领域得到了广泛使用[1]。与传统的二维 GIS (Geographic Information System, GIS) 相比,三维 GIS 无论是在空间拓扑信息、几何位置信息和部分语义信息上,都能够更客观地反映出真实世界[2-4]。在当今社会,随着计算机技术的飞速发展,GIS 获得了新一轮发展机遇,WebGL 应运而生。CesiumJS 作为轻量级的三维地球 JavaScript 库,支持 WebGL 的硬件加速,在浏览器上可以流畅运行[5]。

根据 2019 年第二季度移动端操作系统市场份额表明,Android 以 77.14%的占比继续位居移动端手机系统的首位,同 PC 端相比,移动端具有更强的便携性、更低的使用功率,以及更便捷的互联网连接方式,使人们与外部世界的连接更加舒适。与此同时,移动端的地图服务也逐步进入人们的日常生活中,如日常出行导航、线路规划或搜索未知地点等,给人们的生活带来了极大的便利。

“长光卫星云极视”项目将 CesiumJS 与 Android 端相结合,将遥感数据作为移动地图服务的基础数据,将遥感信息以三维地球的形式展示给用户,使得用户可以随时随地查看并操作真实且立体的遥感数据,从真实性和生动性等方面都可以大幅度提高其使用感受。

收稿日期:2020-06-08

作者简介:李　想(1992-),男,黑龙江庆安人,助理研究员,硕士,2017 年毕业于吉林大学计算机技术专业,主要从事移动端研发及遥感大数据方面的研究工作。

1　系统设计

1.1　架构设计

“长光卫星云极视”项目由 Android 客户端和服务端组成,Android 端使用 MVP(Model-View-Presenter, MVP)架构,同时使用基于 Retrofit2+OKHttp3+RxJava2+Dagger2+Lifecycle 的网络请求模块,在原生 Fragment 上搭载自定义 WebView 控件,通过 Web 端的 CesiumJS 库显示三维地球,并使用 JavaScript Bridge 进行 Android 端 Kotlin 代码与 Web 端 JavaScript 代码之间的交互。

服务端使用 Spring Boot+Mybatis+Maven 架构开发,数据库使用 PostgreSQL+PostGIS 实现。其中图像处理使用 GDAL(Geospatial Data Abstraction Library,GDAL)开源库,图像的发布使用 GeoServer 服务器,并将已发布好的服务通过 CesiumJS 进行加载。同时服务器端还使用 Swagger、Apache Velocity 进行辅助开发。Android 客户端与服务端采用标准 JSON 格式进行数据传递,项目的系统架构如图 1 所示。

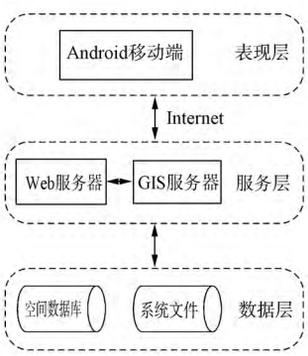

图 1　系统架构图

Fig.1　System architecture diagram

1.2　功能设计

“长光卫星云极视”项目的主要功能分为两部分,遥感产品的展示和应用,如图 2 所示。在展示的遥感产品中,主要包括卫星图(全色/多光谱)、卫星视频、专题数据(统计报告/PDF)、无人机数据、全景数据、标绘数据(标记点/标记路线/标记区域)和其他数据。在遥感产品的应用中,以标绘和测量为主,包含底图图层切换、自定义缩放、网格搜索、加载图层链接和其他应用(宜居指数/静态目标识别/地物分类/火电提取等)。

2　关键技术

2.1　MVP 架构

“长光卫星云极视”项目涉及的功能与模块较多,代码量较大,在体量上属于中大型 Android 项目,因此,项目开发架构的选择尤为重要。本项目主要使用 MVP(Model-View-Presenter,MVP)架构,与传统的 MVC(Model-View-Controller,MVC)架构相比,MVP 架构能够有效降低代码的耦合度,使整体结构更加清晰,能够大幅提升代码的可维护性。在 MVP 架构中,Model 层负责数据修改和操作,

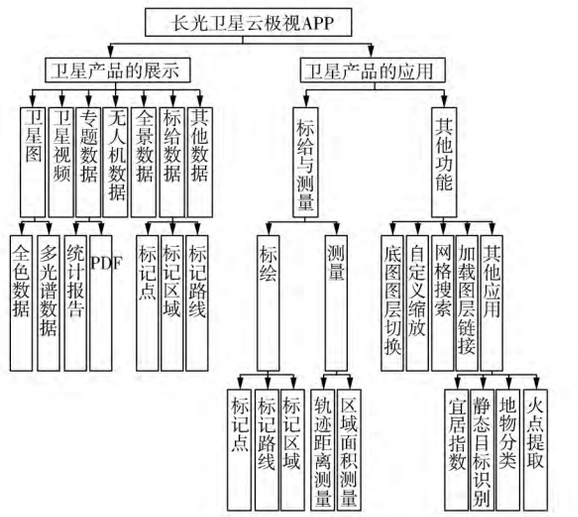

图 2　主要功能展示图

Fig.2　Main function display diagram

View 层负责全部 UI 组件以及与 Presenter 层的交互操作,Presenter 层负责全部的逻辑部分[6-7]。其主要特点如下:

1)各部分之间(View 层与 Presenter 层、Model 层与 Presenter 层)通信都是双向关系,View 层和 Presenter 层互相持有对象,Model 层在 Presenter 层中使用单例模式,而 Presenter 层实现 Model 层的回调接口,整体关系如图 3 所示;

图 3　MVP 模式示意图

Fig.3　MVP pattern diagram

2)View 层与 Model 层不直接发生联系,数据和逻辑通过 Presenter 层传递;

3)View 层不负责处理业务逻辑,被称为“被动视图”;

4)所有的交互都在 Presenter 层进行;

5)Model 层除获取数据外,还负责数据转换和数据存储。

以“长光卫星云极视”登录接口为例,一次完整的 MVP 架构的数据交互步骤如图 4 所示。

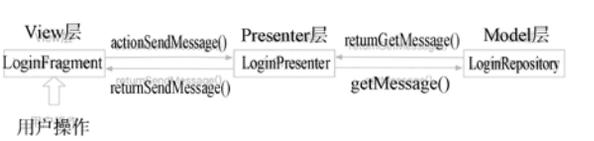

图 4　登录接口 MVP 模式示意图

Fig.4　Schematic diagram of the MVP mode of the login interface

1)用户在页面点击按钮,获取登录验证码;

2)View 层(LoginFragment)向 Presenter 层(LoginPresenter)发送委派函数 actionSendMessage();

3)Presenter 层向 Model 层(LoginReposity)发送检索数据的消息,即 getMessage()函数,从接口或数据库中获取数据,在此过程中 Presenter 层作为观察者,监听 Model

层的返回结果;

4)当 Model 层获取到数据后,Presenter 层将观察到该事件,即 returnGetMessage()函数;

5)Presenter 层将结果数据通过 returnSendMessage()函数返回给 View 层,至此一次交互结束。在此过程中界面(View 层)与数据(Model 层)未发生直接联系,体现出了更大的解耦性。

2.2 JavaScript 与 Android Native 交互

随着 Html5 的不断优化以及移动端对实时动态化的需求增大,开发者经常需要在原生 Android 应用中添加一些网页,并进行网页端和原生代码之间包括函数调用、参数传递等的交互操作。因编码语言不同,网页端和原生代码无法直接进行交互,而需通过 JavaScript Bridge 作为桥梁,当前使用范围最广泛的开源库主要有 JsBridge 和 DSBridge 等。

2.2.1 基于 JsBridge 的交互原理

JsBridge 是 Android Native 代码与 JavaScript 代码的通信桥梁。如图 5 所示,JsBridge 给 JavaScript 提供了调用 Android Native 的方法,同时也承接了 Android Native 调用 JavaScript 事件的封装;JsBridge 构建了双向的 JavaScript 和 Android Native 之间的通信[8]。

图 5 JsBridge 交互示意图

Fig.5 JsBridge interactive diagram

2.2.2 基于 DSBridge 的交互原理

DSBridge 是一款跨平台的 JavaScript Bridge,通过 DSBridge,用户可以在 JavaScript 和 Android Native 之间同步或异步调用彼此的函数[9]。DSBridge 作为最新开源的新框架,体现出了很多优势:

1)真正实现了跨平台(iOS/Android);

2)同时支持异步调用和同步调用,是唯一支持同步调用的 JavaScript Bridge 库;

3)逻辑清晰,使用简单,学习成本低;

4)有详细的中文文档和问题反馈渠道。

DSBridge 与 JsBridge 使用过程相似,但在稳定性上有大幅度的提升。JsBridge 在 Web 页面加载时,如果短时间内多次调用原生的方法,会遇到回调参数未被调用的情况。在“长光卫星云极视”项目的标绘与测量模块,Web 端需随着用户操作地球的移动,将两点之间的直线距离实时传递给 Android Native 端并通过 AppcompatTextView 组件展示给用户。在此调用过程中若使用 JsBridge 则会发生操作卡顿、更新不及时的现象,而使用 DSBridge 不会发生此类错误,因此,在“长光卫星云极视”项目中,使用 DSBridge 作为 Android Native 与 JavaScript 的通信库。

2.2.3 基于 DSBridge 的交互实现

2.2.3.1 Android 端传值给 Web 端

Android 端与 Web 端主要通过向 Web 端注入 JavaScript 代码的方式来间接调用 Html5 脚本中定义的方法[10]。DSBridge 在 Java 代码注入过程中,为了避免在垃圾回收时被清理,以及为了方便 Java 回调的时候能够找到,用 window 对象的_dsf 域保存 Java 调用 Javascript 的函数方法。在 Java 调用 JavaScript 时,DSBridge 中使用 callHandler()方法确定 window._dsf 的函数方法,对代码进行约束。整体流程如图 6 所示。

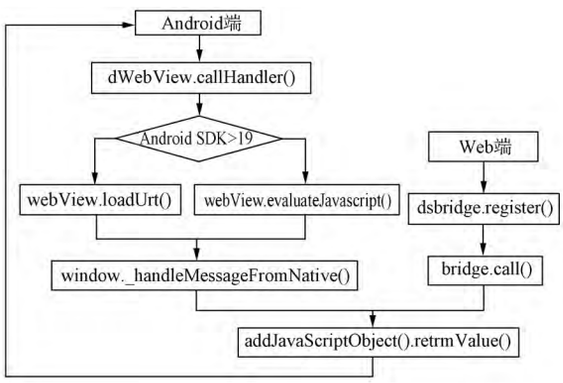

图 6 Android 端传值给 Web 端示意图

Fig.6 Schematic diagram of passing value from Android to Web

1)Android 端:调用 mDWebView.callHandler()方法传递函数名与参数;

Web 端:调用 dsbridge.register()函数定义交互的函数名称以及函数内容。

2)Android 端:当 Android SDK(Software Development Kit,SDK)版本号低于 19 时,Android 官方提供无回调方法:webView.loadUrl();当 Android SDK 版本号高于 19 时,Android 官方新增有回调的方法:webView.evaluateJavascript();

Web 端:调用 dsbridge.js 中 register()函数的 bridge.call()方法。

3)Android 端:通过 window._handleMessageFromNative()函数将函数名称和参数注入 JavaScript;

Web 端:通过 dsbridge.js 中的_handleMessageFromNative()函数接受函数名称和参数,并执行函数体,获得返回值;

4)通过反射机制将返回值传递给 window._dsb 对应的 addJavaScriptObject().returnValue()方法,Android 端获得返回值,一次交互结束。

2.2.3.2 Web 端传值给 Android 端

Web 端与 Android 端的交互方式为 Web 端获取到 Android 端注入的对象,通过该对象调用其方法。整体流程如图 7 所示。

1)Web 端:调用 dsBridge.call(),定义函数名与返回值;

Android 端:在初始化函数 DWebView.init()函数中将命名为_dsbridge 的 InnerJavascriptInterface 对象注入 JavaScript 中。

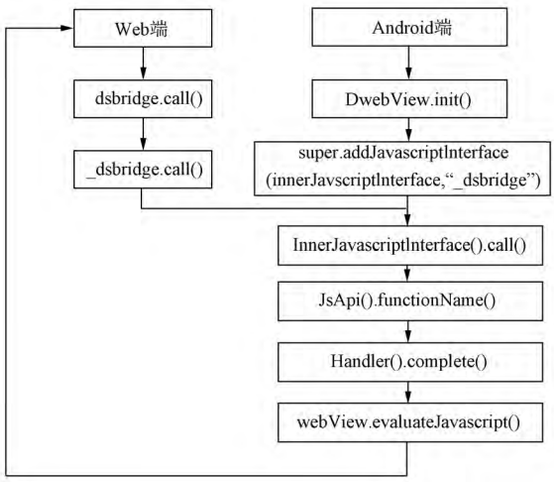

图 7　Web 端传值给 Android 端示意图

Fig.7　Schematic diagram of passing value from Web to Android

2)Web 端:通过 dsbridge.js 中的_dsBridge.call( )函数,使用 bridge 组件的 call( )方法将数据传递到 Android 端;

Android 端:InnerJavascriptInterface 类中的 call 函数接收到 JavaScript 传递过来的函数名和参数值。

3)Android 端:如果接收到的函数名称与 JsApi 类中的对象名称相匹配,则通过反射执行该方法。

4)Android 端:通过 Handler( ).complete( )函数,调用 webView.evaluateJavascript( )函数,将消息传回 Html5 页完成回调,一次交互结束。

2.2.4　基于 DSBridge 的应用示例

在"长光卫星云极视"项目中的卫星图展示模块,需要将 Android 端通过网络获取到的 JSON 数据传递给 Web 端用以显示出对应的影像数据,定义该函数名为 jShowLayer( );同时需要获取 Web 端三维地球的层级级数并在 Android 端页面显示,定义该函数名为 updateLevel( ); Android 端与 Web 端利用 DSBridge 进行交互的流程如图 8 所示,主要步骤如下:

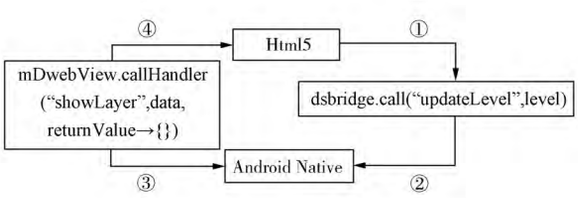

图 8　DSBridge 交互流程图

Fig.8　DSBridge interactive flow chart

步骤 1: Web 端调用 JavaScript 函数将调用 Android 端的函数名称和返回值传递,如下所示:

```
dsBridge.call("updateLevel", level);
```

步骤 2: Android 端接收到函数名称和返回值,在 JsApi 类中获取到对应的 updateLevel( )方法,在函数体内进行后续页面逻辑操作,如下所示:

```
@JavascriptInterface
public String updateLevel(final Object msg) {
    uiHandler.post(() -> Objects.requireNonNull(
            /**
             * 执行后续操作
             */
            fragment.getExploreUtils()).setUpdateLevel(msg.toString())
    );
    return msg + " [syn call] ";
}
```

步骤 3: Android 端通过调用 callHandler( )函数传递函数名称和参数,在成功获得回调后,在 OnReturnValue{ }中执行后续操作,如下所示:

```
mDWebView!!.callHandler(
    method: "jShowLayer",
    arrayOf(
        Gson().toJson(
            ShowLayerBean(
                offset: "0",
                layerName: "AdvancedProduct-KEY:beijing_after_dwfl",
                geo: "{\"type\": \"Polygon\",\"coordinates\": " +
                        "[[[115.4055647876,39.4241698206], [115.4055647876,41.0682903225]," +
                        "[117.511953845,41.0682903225],[117.511953845,39.4241698206]," +
                        "[115.4055647876,39.4241698206]]]}"
            ),
            ShowLayerBean::class.java
        )
    ),
    OnReturnValue<String> { it: String!
        //执行回调
        mDiwuShowFlag = true
        item.title = "关闭图层"
    })
```

步骤 4:Web 端通过注册与 Android 端相同的函数名来获取传递过来的参数并执行后续加载卫星图层操作,如下所示:

```
//显示卫星图
dsBridge.register('jShowLayer',function(r) {
        addImageLayer(r)
})
```

2.3　加载 GeoServer 服务的设计与实现

本项目在 GIS 端采用了 GeoServer+CesiumJS 的服务框架。其中 CesiumJS 作为基于 WebGL 的地图引擎,在不需要第三方插件的情况下提供三维地球渲染、矢量绘制数据显示和标准绘制影像图层加载等工作。GeoServer 作为地图数据发布平台,支持 WCS(Web Coverage Service, WCS)、WFS(Web Feature Service, WFS)、WMS(Web Map Service, WMS)、TMS(Tile Map Service, TMS)、WMS-C(Web Mapping Service - Cached, WMS-C)、WMTS(Web Map Tile Service, WMTS)等地图服务,将 Postgis、Shapefile、GeoTiff 等矢量与栅格数据输出为 jpeg、gif、png、svg、kml 等格式。

在本应用中,使用 WMTS 服务管理影像数据,业务逻辑如图 9 所示。

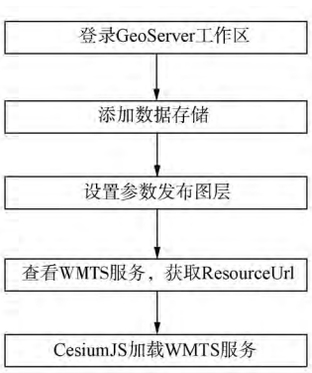

图 9　CesiumJS 加载 WMTS 服务流程图

Fig.9　CesiumJS loading WMTS service flow chart

1)登录 GeoServer Web 站点,进入工作区;2)新建数据源,选择数据接入方式并输入相关数据参数;3)发布图层,并输入参考坐标系、边框以及覆盖参数;4)在图层预览界面中搜索发布的图层,获取图层对象的 ResourceUrl 等相关信息;5)在 CesiumJS 中加载 WMTS 服务,在三维地球上展示出对应地图。

2.4 移动设备上的标绘与测量设计与实现

"长光卫星云极视"项目支持对任意两点间进行测距,对任意多边形进行周长计算以及面积计算,提示自动闭合矢量图形,同时支持对任意点、线和面的标记或轨迹进行备注,并且保存到用户的标绘库中。

在本应用中采用了固定标记点为手机屏幕中心,通过旋转、缩放操作三维地球来进行多点绘制的方式,与全屏幕内用户动态打点相比,在开发上逻辑更清晰,使用上的步骤也更简化。

在技术上主要使用 CesiumJS 的 ScreenSpaceEventHandler 接口进行交互,该接口提供手机端的拖拽、点击等交互方式。自动闭合矢量图形提示功能使用 TurfJS 的 pointsWithinPolygon 函数进行计算,并使用 PostGIS 的 ST_Area 方法进行面积计算。具体的交互方式如图 10 所示,步骤如下:

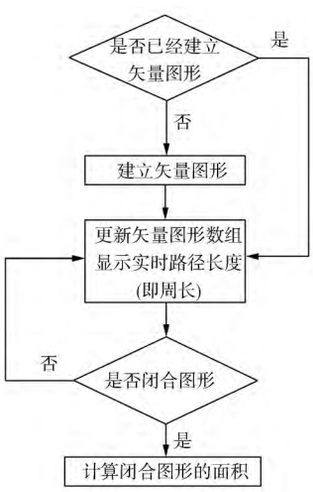

图 10 CesiumJS 多边形周长/面积计算流程图

Fig.10 CesiumJS polygon perimeter/area calculation flow chart

1)设置 CesiumJS 点击业务逻辑:首先判断用户是否已经建立矢量图形。如果未建立则新建矢量图形并将用户添加的点加入到矢量图形数组中;如果已新建矢量图形则更新矢量图形数组,并根据数组更新矢量多边形形状。

2)设置 CesiumJS 拖拽屏幕生成矢量图形业务逻辑:首先,根据矢量图形数组个数以及当前位置与矢量图形起始位置距离判断是否提示自动闭合图形;然后,更新矢量图形数组,并根据数组创建新的矢量图形,并调用 DSBridge 实时更新图形周长数据。

3)设置 CesiumJS 撤销点击业务逻辑:判断矢量图形数组长度,如果长度小于 3 则销毁 CesiumJS 矢量图形对象,同时对矢量图形数组进行 pop 操作;如果长度大于 3,直接对矢量图形数组进行 pop 操作,并调用 DSBridge 实时更新图形周长数据。

3 结束语

本文基于 Android 端和 CesiumJS 库,采用 DSBridge、GeoServer、MVP 等网络编程技术,分析并设计了基于 CesiumJS 的 Android 应用开发模式,并通过"长光卫星云极视"项目证明了该模式的可行性。该项目从遥感产品的展示和遥感产品的应用两个方面,提供了包括卫星图、卫星视频等在内的多种遥感产品的展示与交互,对三维地球多种可切换图层的任意标绘与测量等多种功能,充分发挥了 CesiumJS 在 Android 端的优势,为用户查看并操作遥感数据,对遥感数据的快速检索,对任意地点的标绘与测量以及遥感信息的发布和共享提供了一个高效且便捷的平台,在实际应用中也大幅度提升了操作效率。

本文基于 CesiumJS 设计的 Android 三维地球开发模式,为遥感技术在大众用户中的普及与应用提供了一个低成本的解决方案,具有一定的实际应用价值。

参考文献:

[1] TSAI F,LAI J S,LIU Y C.An alternative open source web-based 3D GIS: cesium engine environment[C]//Asian Conference on Remote Sensing: Fostering Resilient Growth in Asia. sn, 2016.

[2] 李桂华,陈畅曙,钟煜宏,等.基于 Cesium 的南中国海石油平台信息系统设计与开发[J].测绘与空间地理信息,2019,42(6):47-50.

[3] 朱栩逸,苗放.基于 Cesium 的三维 WebGIS 研究及开发[J].科技创新导报,2015,12(34):15-17,22.

[4] 高云成.基于 Cesium 的 WebGIS 三维客户端实现技术研究[D].西安:西安电子科技大学,2014.

[5] DI STASO U,SOAVE M,GIORI A,et al. Heterogeneous-resolution and multi-source terrain builder for CesiumJS WebGL virtual globe[J].International Journal of Civil and Architectural Engineering,2016,10(1):129-135.

[6] 倪红军.基于 MVP 模式的 Android 应用开发研究[J].电子设计工程,2018,26(11):6-9,13.

[7] 曾露. MVP 模式在 Android 中的应用研究[J].软件,2016(6):75-78.

[8] lZYZSD. JsBridge [EB/OL].[2019-11-04].https://github.com/lzyzsd/JsBridge.

[9] WENDUX. DSBridge-Android[EB/OL].[2019-01-03].https://github.com/wendux/DSBridge-Android.

[10] ROBINSUN.Hybrid 开发框架 DsBridge 原理分析[EB/OL].[2019-10-26].https://blog.csdn.net/RunningXiaoHei/article/details/102759400.

[编辑:刘莉鑫]