电e宝app与H5交互开发指南

V1.0

**变更记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **版本号** | **修改内容** | **修改日期** |
| 1 | V1.0 | 无 |  |
|  |  |  |  |

**1、概述**

1、文档描述

随着公司业务发展，在开发过程中H5端需要跟APP端进行交互。

2、交互描述

APP端与H5端的交互，可分为APP调用H5方法和H5端调用APP接口。APP调用H5方法，是APP调用H5写好的方法名并传递相应数据。H5端调用APP接口,是H5调用APP接口，APP收到请求后，做出相应处理并通知H5端，如果有数据一起传递过去。

1. **H5调用APP方法**
2. 引入js文件

在需要与原生交互的页面引入js文件

<script type="text/javascript" src="jsBridge.js"></script>

<script type="text/javascript" src="js\_native\_bridge.js"></script>

1. 调用方法

|  |
| --- |
| **JsBridge.getSession({请求参数},function( data) {回调函数});** |

例getSession是请求的接口名称，如有请求参数放在{}内、格式{"key":"value"}，function( data)是回调函数，data是返回数据格式如下 {"status":"1","msg":"ok","values":{ ...... }；

|  |  |  |
| --- | --- | --- |
| 状态码（status） | 描述 | 错误信息 |
| 1 | 成功 | ok |
| 0 | 失败 | failed |
| -1 | 方法未定义 | method not defined |
| -2 | 参数错误 | param error |
| -3 | 超时 | time out |
| -4 | 用户取消 | user cancel |

返回数据为三种格式

1. 字符串格式

{"status":"1","msg":"ok","values":{"content":"我是返回的字符串"}

1. 数组格式

{"status":"1","msg":"ok","values":{"list":[1,2,3]}；

3.json格式

{"status":"1","msg":"ok","values":{"key","value"}

1. 接口
2. 获取请求头

|  |  |
| --- | --- |
| 方法名称 | getHttpHeader |
| 场景描述 | 获取请求头信息包含（APPVERSION， CLIENTVERSION，SESSIONID）  (此方法不需传值) |
| 调用示例 | JsBridge.getHttpHeader( {},  function( data) {    }); |

（2）获取sessionId

|  |  |
| --- | --- |
| 方法名称 | getSession |
| 场景描述 | 获取sessionId  (此方法不需传值) |
| 调用示例 | JsBridge.getSession({},  function( data) {    }); |

(3)退出登录

|  |  |
| --- | --- |
| 方法名称 | appLogout |
| 场景描述 | 退出登录  (此方法不需传值) |
| 调用示例 | JsBridge.appLogout({},  function( data) {    }); |

1. 跳转

|  |  |
| --- | --- |
| 方法名称 | goToActivity |
| 场景描述 | 跳转到指定控制器（controllerName传值需区分android和ios。内容根据实际情况而定）  (此方法需传值) |
| 调用示例 | JsBridge.goToActivity(  {"controllerName-android":"","controllerName-ios":""},  function( data) {    }); |

1. 请求数据

|  |  |
| --- | --- |
| 方法名称 | requestAysnc |
| 场景描述 | 请求后台数据  (此方法需传值) |
| 调用示例 | JsBridge.requestAysnc(  {"requestUrl":"请求路径","data":"请求参数"},  function( data) {  }); |

1. 获取userId

|  |  |
| --- | --- |
| 方法名称 | getUserId |
| 场景描述 | 获取getUserId  (此方法不需传值) |
| 调用示例 | JsBridge.getUserId({},  function( data) {    }); |

1. 获取getBaseData

|  |  |
| --- | --- |
| 方法名称 | getBaseData |
| 场景描述 | 获取getBaseData，获取基本信息。  (此方法不需传值) |
| 调用示例 | JsBridge.getBaseData({},  function( data) {    }); |

1. 获取当前环境

|  |  |
| --- | --- |
| 方法名称 | getConfiguration |
| 场景描述 | 获取当前是测试环境还是正式环境  (此方法不需传值) |
| 调用示例 | JsBridge.getConfiguration({},  function( data) {    }); |

1. 分享shareContent

|  |  |
| --- | --- |
| 方法名称 | shareContent |
| 场景描述 | 分享到微信好友、通讯录、朋友圈。  (此方法需传值) |
| 调用示例 | JsBridge.shareContent(  {"type":"","data":""},  function( data) {    }); |
| 描述 | type分享方式：0微信好友 1通讯录 2朋友圈  data 分享内容 |

1. 修改导航栏背景色／图片

|  |  |
| --- | --- |
| 方法名称 | setNavigationBarStyle |
| 场景描述 | 修改导航栏背景色／图片。  导航条内容使用标题（title）内容。  (此方法需传值) |
| 调用示例 | JsBridge.setNavigationBarStyle(  {"imageUrl":"","backgroundColor":"","alpha":""},  function( data) {    }); |
| 描述 | imageUrl 导航栏图片地址。  backgroundColor 导航条颜色，例：#ccc。 |

1. 修改导航栏按钮

|  |  |
| --- | --- |
| 方法名称 | setNavigationBarItems |
| 场景描述 | 修改导航栏按钮文字／图片。  (此方法需传值) |
| 调用示例 | JsBridge.setNavigationBarItems(  {"imageUrl":"","text ":"","methodName":""},  function( data) {    }); |
| 描述 | imageUrl 导航栏按钮图片地址。  text 导航条按钮文字。  methodName 参数为方法名，按钮被点击时，app通知js调用的方法。  （ios通过此方法传递方法名，安卓是直接告诉方法名） |

1. 银联支付

|  |  |
| --- | --- |
| 方法名称 | unionPay |
| 场景描述 | 银联支付  (此方法需传值) |
| 调用示例 | JsBridge.unionPay({},  function( data) {    }); |

1. 调用摄像头

|  |  |
| --- | --- |
| 方法名称 | openCamera |
| 场景描述 | 调用摄像头扫描二维码拍照。  (此方法需传值) |
| 调用示例 | JsBridge.openCamera(  {"type":"QR","methodName":"callBackCamera"},  function( data) {    }); |
| 描述 | type 打开类型QR或PICTURE，QR打卡二维码PICTURE打开照相  callBackName 扫描或拍照成功后调用js的函数名称默认是callBackCamera 不可更改。  回调结果返回成功或失败。成功表示打开摄像头成功，失败原因有打开摄像头错误或未获取到摄像头权限。 |

（14）获取地理定位

|  |  |
| --- | --- |
| 方法名称 | getLocaltion |
| 场景描述 | 获取地理定位  (此方法不需传值) |
| 调用示例 | JsBridge.getLocaltion(  {},function( data) {    }); |
| 描述 | 返回信息  longitude 经度 latitude 纬度 |

1. 增加接口

如定义的接口不能满足需求，可与APP进行联系新增接口。

新增接口分2部分。

（1） 增加接口

确定接口名称等问题后，再在插件中增加接口，新增接口需在jsBridge.js 文件下进行

格式如下：

方法名: function(params,cb){

this.checkUsable("方法名",params,cb);

}

例如增加getSession接口

getSession: function(params,cb){

this.checkUsable("getSession",params,cb);

}



1. 接口调用

JsBridge.新增接口名称(

{与app协商是否传参数以及参数格式},

function( data) {

回调函数

}

});

例：

JsBridge.getSession({},

function( data) {

}

});

1. Demo 实例

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width">  <title>JsBridge demo</title>  </head>  <body>  <div id="wrap">  <button type="button" name="button" onclick="test()">调用APP的getSession接口</button>  <button type="button" name="button" onclick="test1()">调用native的goToActivity接口</button>  </div>  <script type="text/javascript" src="jsBridge.js"></script>  <script type="text/javascript" src="js\_native\_bridge.js"></script>  <script type="text/javascript">  //无参数调用  function test(){  JsBridge.getSession({},function( data) {  if (data.status == 1) {  console.log('"getSession" bridge called success;');  }else{  console.log('"getSession" bridge called faild;');  }  });  }  //有参数调用  function test1(){  JsBridge.goToActivity(  {"key":"value","key1":"value1"},  function( data) {  if (data.status == 1) {  console.log('"getSession" bridge called success;');  }else{  console.log('"getSession" bridge called faild;');  }  });  }  </script>  </body>  </html> |

1. **APP调用H5方法**

1、引入js文件

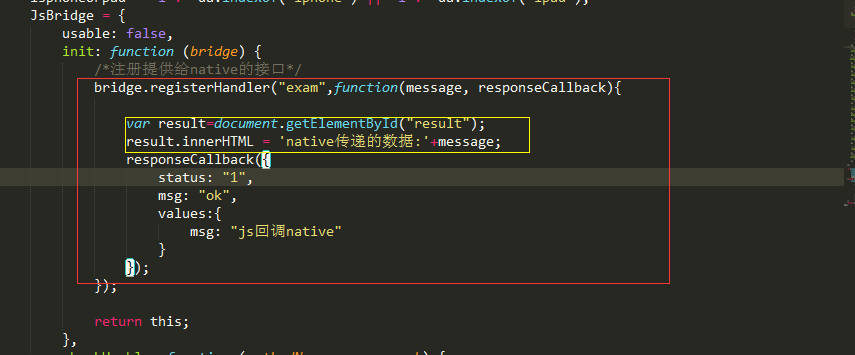
在需要与原生交互的页面引入js文件

<script type="text/javascript" src="jsBridge.js"></script>

<script type="text/javascript" src="js\_native\_bridge.js"></script>

1. 添加方法

H5端方法是写在jsBridge.js 中，格式如下



图中红色矩形方块内是一完整APP调用方法，exam是调用的方法名，黄色矩形是方法内容，

responseCallback是方法运行完成后回调APP，如有值需传递给APP放在values内。

新添加方法需放在JsBridge init方法内，格式如上图红色矩形方块内容。