目錄

[目錄 1](#_Toc78536776)

[1. 文件更動記錄 3](#_Toc78536777)

[2. APP 溝通注意事項 4](#_Toc78536778)

[3. webview url 參數說明 7](#_Toc78536779)

[API : 1-1 取得手機Device ID 7](#_Toc78536780)

[API : 1-2 顯示Bank List Dialog 8](#_Toc78536781)

[API : 1-3 取得常用帳單筆數 10](#_Toc78536782)

[API : 1-4 顯示常用帳單列表Dialog 11](#_Toc78536783)

[API : 1-5 顯示可付款的他行信用卡列表 Dialog 14](#_Toc78536784)

[API : 1-6 顯示指定的html Dialog(只有確認) 16](#_Toc78536785)

[API : 1-7 顯示指定的html Dialog(有確認及取消) 18](#_Toc78536786)

[API : 1-8 顯示message Dialog(只有確認) 20](#_Toc78536787)

[API : 1-9 顯示Message Dialog(有確認及取消) 22](#_Toc78536788)

[API : 1-10 檢核是否已登入 24](#_Toc78536789)

[API : 1-11 檢核是否已登入，如未登入會顯示登入Dialog 25](#_Toc78536790)

[API : 1-12 登出並回到首頁 26](#_Toc78536791)

[API : 1-12 顯示支付密碼驗證Dialog 26](#_Toc78536792)

[API : 1-13 轉跳到自動扣款設定頁 27](#_Toc78536793)

[API : 1-14 依menuID轉跳到指定頁 28](#_Toc78536794)

[API : 1-15 轉跳到註冊頁 29](#_Toc78536795)

[API : 1-16 是否已註冊 30](#_Toc78536796)

[API : 1-17 取得登入資訊 31](#_Toc78536797)

[API : 1-18 關閉webview 32](#_Toc78536798)

[API : 1-19 關閉webview並回到首頁 33](#_Toc78536799)

[API : 1-20 關閉webview並轉跳到繳費記錄頁 35](#_Toc78536800)

[API : 1-21 是否顯示App的Header 36](#_Toc78536801)

[API : 1-22 是否顯示App的下方BottomNav 37](#_Toc78536802)

[API : 1-23 啟動 OCR 38](#_Toc78536803)

[API : 1-24 更改APP個人電話號碼 38](#_Toc78536804)

[API : 1-25 更改APP個人Email 39](#_Toc78536805)

[API : 1-26 通知APP更新個人會員等級 40](#_Toc78536806)

[API : 1-27 取得手機是否有綁定等資訊 41](#_Toc78536807)

[API : 1-28 (iOS)取得Wallet中信用卡狀態 42](#_Toc78536808)

[API : 1-29 (iOS)Apple Pay加卡 44](#_Toc78536809)

[API : 1-31 掃描身分證正面 47](#_Toc78536810)

[API : 1-32 掃描信用卡 48](#_Toc78536811)

[API : 1-33 掃描3碼瓦斯帳單bar code 49](#_Toc78536812)

[API : 1-34 掃描1碼水費帳單bar code 50](#_Toc78536813)

## 1. 文件更動記錄

|  |  |  |  |
| --- | --- | --- | --- |
| 更動日期 | 版號 | 作者 | 更動內容 |
| 2021/06/04 | 0.1 | 王世豪 | 初版 |
| 2021/06/07 | 0.2 | 吳君儀 | 文件格式，文字補充 |
| 2021/06/07 | 0.3 | 范嫺芸 | 資料範例補充 |
| 2021/06/08 | 0.4 | 王世豪 | 新增userAgent補充說明 |
| 2021/06/11 | 0.5 | 王世豪 | 1. 調整1.5 showCreditCardList更名為showPayCreditCardList 2. 新增1.12 logoutToHome API 3. 修改1.19 exitWebToHome API |
| 2021/07/02 | 0.6 | 王世豪 | 1. 新增1-24更改APP個人電話號碼 2. 新增1-25更改APP個人Email 3. 新增1-26通知APP更新個人會員等級 |
| 2021/07/16 | 0.7 | 王世豪 | 1.修改1-17 登入資料欄位 |
| 2021/07/21 | 0.8 | 王世豪 | 1.修改1-2顯示Bank List Dialog可指定是否包含永豐銀行 |
| 2021/07/26 | 0.9 | 王世豪 | 1.新增1-27取得手機是否有綁定的資訊 |
| 2021/07/28 | 1.0 | 范嫺芸 | 1. 1.新增 webview url 參數說明 |
| 2021/07/29 | 1.1 | 吳君儀 | 1.新增1-28 (iOS)取得Wallet中信用卡狀態   1. 2.新增1-29 (iOS)Apple Pay加卡 |
| 2021/07/30 | 1.2 | 王世豪 | 1. 新增1-30掃描身分證正面 2. 新增1-31掃描信用卡卡號 3. 新增1-32掃描瓦斯3碼bar code 4. 新增1-33掃描水費1碼bar code |
| 2021/08/02 | 1.3 | 吳君儀 | 1.更新1-28 (iOS)取得Wallet中信用卡狀態  2.更新1-29 (iOS)Apple Pay加卡 |

## APP 溝通注意事項

※ 在CaWhoPay APP 中user agent格式如下

"sinopac://CaWhoPay?appVer=${BuildConfig.VERSION\_NAME}&osType=Android&osVer=${Build.VERSION.SDK\_INT}&devName=${Build.MANUFACTURER} ${Build.MODEL}"

appVer->CaWhoPay版本

osType->Android or iOS

osVer->os的版本

devName->手機型號

※ 判斷裝置是 Android 或 IOS 再決定執行的 function

* Webview call App

Android 可直接呼叫 function，APP.function name(參數)

範例：

APP.getDeviceUUID()

IOS 因使用 WebViewJavascriptBridge，所以需用該套件方式呼叫

範例：

setupWebViewJavascriptBridge((bridge) **=>** {

      bridge.callHandler('getDeviceUUID', '', (response) **=>** {

        return resolve(response);

      });

    });

※ 本文件範例以 Android 為主

* App call Webview

提供給 Android 呼叫

範例：

dialogCallBack(id, action, data) {

    switch (id) {

      case 'bankList': {

*// 轉換獲取的資料*

**const** { code, name } = translateData(data, 1);

*// 0 表示使用者 click 確定*

        if (Number(action) === 0) {

*// do something…*

        }

        break;

      }

      default:

        break;

    }

  }

IOS 同樣需用套件方式呼叫

範例：

**function** setupWebViewJavascriptBridge (callback) {

  if (window.WebViewJavascriptBridge) { return callback(WebViewJavascriptBridge); }

  if (window.WVJBCallbacks) { return window.WVJBCallbacks.push(callback); }

  window.WVJBCallbacks = [callback];

**var** WVJBIframe = document.createElement('iframe');

  WVJBIframe.style.display = 'none';

  WVJBIframe.src = 'https://\_\_bridge\_loaded\_\_';

  document.documentElement.appendChild(WVJBIframe);

  setTimeout(**function** () { document.documentElement.removeChild(WVJBIframe) }, 0)

}

setupWebViewJavascriptBridge((bridge) **=>** {

  bridge.registerHandler("dialogCallBack", **function** (res) {

*// 轉換獲取的資料*

**const** { id, action, data } = JSON.parse(res);

*// 同上方 Android 範例執行 function*

  })

});

## webview url 參數說明

電支註冊頁面帶 cust\_id 為身分證，tele\_no 為電話號碼，直接帶入註冊 input

繳費網址帶 comm\_id 可由此參數查詢帳單進行後續流程，另有 add\_common 0 不加入 ( 常用、待繳 ) 1 加入 ( 掃碼 )

## API : 1-1 取得手機Device ID

定義：getDeviceUUID():String

Input：

無

Output：

Device ID

Sample：

APP.getDeviceUUID()

## API : 1-2 顯示Bank List Dialog

定義：showBankListDialog(params:String)

呼叫後會顯示各銀行列表資訊，使用者操作後會call back dialogCallBack(id: String, action: Int, data: String)，所以呼叫的網頁需準備dialogCallBack(id: String, action: Int, data: String)。

Input：

Param->是一個Json String內容如下

{“Id”:”id” //callback用來判斷動作的id

“hasSinoac”:true //是否包含永豐銀行}

Output：

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Description | Remark |
| id | 呼叫時所帶入的id | 用來判斷動作使用 |
| action | 動作 | 0選擇成功 |
| data | 選擇的銀行資料  Json format(android)  String with Json format(iOS) | {  “code”:銀行代碼,  “name”:銀行名稱  }  “\”bankCode\”:\”銀行代碼”,\”bankName\”:\”銀行名稱\” |

Sample：

APP.showBanklistDialog('bankList')

dialogCallBack(id, action, data) {

  switch (id) {

    case 'bankList': {

*// 轉換獲取的資料*

**const** { code, name } = translateData(data, 1);

*// 0 表示使用者 click 確定*

      if (Number(action) === 0) {

*// do something…*

      }

      break;

    }

    default:

      break;

  }

}

## API : 1-3 取得常用帳單筆數

定義：getBillSettingSize(menuIDs:String,id:String)

呼叫後會查看App cache是否有常用帳單資訊，如果沒有app會發送取得資料，在一個menuIDs來作過濾得知筆數，callback dataCallBack(id:String,data:String)，所以網頁要準備dataCallBack(id:String,data:String)讓app呼叫。

Input：

menuIDs->使用,分隔可傳入，可傳入需要過濾的常用帳單的menuid

Id->callback用來判斷動作的id

Output：

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Description | Remark |
| id | 呼叫時所帶入的id | 用來判斷動作使用 |
| data | 回覆資料筆數 | “3” ->筆數 |

Sample：

※ 參數為物件會使用 JSON.stringify 先做轉變

APP.getBillSettingSize({

  menuIDs: '2328A3C1-D755-46F6-8C62-5B268B27E221',

  id: 'getBillSettingSize'

})

## API : 1-4 顯示常用帳單列表Dialog

定義：showBillSettingDialog(menuIDs:String,id:String)

呼叫後會顯示依menuIDs過濾常用帳單，再顯示指定類別的常用帳單列表Dialog，使用者操作後會callback dialogCallBack(id: String, action: Int, data: String)，所以呼叫的網頁需準備dialogCallBack(id: String, action: Int, data: String)。

Input：

menuIDs->使用逗號分隔傳入需要過濾的常用帳單的menuid。

Id->callback用來判斷動作的id。

Output：

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Description | Remark |
| id | 呼叫時所帶入的id | 用來判斷動作使用 |
| action | 動作 | 0選擇成功 |
| data | 選擇的常用帳單資料  Json format(android)  String with Json format(iOS) | {  “menu\_id”:menu id,  “comm\_id”:common id,  “common\_type”:0, //0帳單 1帳戶  “comm\_name”:帳單名稱,  “comm\_acc\_id”:繳費帳戶/信用卡  “Notify\_info”:{  “id”:提醒id,  “name”:提醒名稱,  “months”:”11100001111”, //12個月1代表通知0代表不通知  “day”:3 //每月的第幾天通知  },  “propertyType”:”custid” //欄位種類 custid 身分證/birth 出生/cardno 信用卡/powerid 電號/waterid 水號 // /noticeno 銷帳編號/phone 電話號碼/carno 車號/vehicletype 車輛類型1汽車2機車  } |

Sample：

※ 參數為物件會使用 JSON.stringify 先做轉變

APP.showBillSettingDialog({

  menuIDs: '2328A3C1-D755-46F6-8C62-5B268B27E221',

  id: 'getCommonBillInfo'

})

## API : 1-5 顯示可付款的他行信用卡列表 Dialog

定義：showPayCreditCardList(id:String)

呼叫後會顯示可以被繳信用卡費的信用卡列表Dialog，使用者操作後會call back dialogCallBack(id: String, action: Int, data: String)，所以呼叫的網頁需準備dialogCallBack(id: String, action: Int, data: String)。

Input：

Id->callback用來判斷動作的id

Output：

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Description | Remark |
| id | 呼叫時所帶入的id | 用來判斷動作使用 |
| action | 動作 | 0選擇成功 |
| data | 選擇的可繳信用卡費資料  Json format(android)  String with Json format(iOS) | {  “menuID”:menu id,  “bankCode”:銀行code,  “bankName”:銀行名稱,  “fromMessage”:註記說明  } |

Sample：

APP.showPayCreditCardList('credCardList')

## API : 1-6 顯示指定的html Dialog(只有確認)

定義：showHtmlDialog(param:String)

呼叫後會顯示Dialog，Dialog中的內容依指定的資料呈現，使用者操作後會call back dialogCallBack(id: String, action: Int, data: String)，所以呼叫的網頁需準備dialogCallBack(id: String, action: Int, data: String)。

Input：

Param->是一個Json String內容如下

{

“Id”:”id” //callback用來判斷動作的id

“title”:title //dialog title文字

“url”:”http://google” //網頁位置，與htmlStr擇一提供

“htmlStr”:”” //html字串，與url擇一提供

“btnStr”:’確定’ //確認按鈕的字串

“data”:”data” //任一字串資料，會在callback時帶回

}

Output：

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Description | Remark |
| id | 呼叫時所帶入的id | 用來判斷動作使用 |
| action | 動作 | 0選擇成功 |
| data | 傳入的data String |  |

Sample：( 目前還未使用該 API )

## API : 1-7 顯示指定的html Dialog(有確認及取消)

定義：showHtmlSelDialog(param:String)

呼叫後會顯示Dialog，Dialog中的內容依指定的資料呈現，使用者操作後會call back dialogCallBack(id: String, action: Int, data: String)，所以呼叫的網頁需準備dialogCallBack(id: String, action: Int, data: String)。

Input：

Param->是一個Json String內容如下

{

“Id”:”id” //callback用來判斷動作的id

“title”:title //dialog title文字

“url”:”http://google” //網頁位置，與htmlStr擇一提供

“htmlStr”:”” //html字串，與url擇一提供

“btnNO”:’取消’ //取消按鈕的字串

“btnOK”:’確定’ //確認按鈕的字串

“data”:”data” //任一字串資料，會在callback時帶回

}

Output：

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Description | Remark |
| id | 呼叫時所帶入的id | 用來判斷動作使用 |
| action | 動作 | 0選擇成功  1選擇取消 |
| data | 傳入的data String |  |

Sample：( 目前還未使用該 API )

## API : 1-8 顯示message Dialog(只有確認)

定義：showMsgDialog(param:String)

呼叫後會顯示Dialog，Dialog中的內容依指定的資料呈現，使用者操作後會call back dialogCallBack(id: String, action: Int, data: String)，所以呼叫的網頁需準備dialogCallBack(id: String, action: Int, data: String)。

Input：

Param->是一個Json String內容如下

{

“Id”:”id” //callback用來判斷動作的id

“title”:title //dialog title文字

“msg”:”message” //文字訊息

“btnStr”:’確定’ //確認按鈕的字串

“data”:”data” //任一字串資料，會在callback時帶回

}

Output：

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Description | Remark |
| id | 呼叫時所帶入的id | 用來判斷動作使用 |
| action | 動作 | 0選擇成功 |
| data | 傳入的data String |  |

Sample：

※ 參數為物件會使用 JSON.stringify 先做轉變

APP.showMsgDialog({

  id: 'addCommonBillSuccess',

  title: '已加入常用帳單',

  msg: '已儲存「中華電信1」銷帳編號',

  btnOK: '確 定',

  data: {}

})

## API : 1-9 顯示Message Dialog(有確認及取消)

定義：showMsgSelDialog(param:String)

呼叫後會顯示Dialog，Dialog中的內容依指定的資料呈現，使用者操作後會call back dialogCallBack(id: String, action: Int, data: String)，所以呼叫的網頁需準備dialogCallBack(id: String, action: Int, data: String)。

Input：

Param->是一個Json String內容如下

{

“id”:”id” //callback用來判斷動作的id

“title”:title //dialog title文字

“msg”:”文字訊息” //文字訊息

“btnNO”:’取消’ //取消按鈕的字串

“btnOK”:’確定’ //確認按鈕的字串

“data”:”data” //任一字串資料，會在callback時帶回

}

Output：

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Description | Remark |
| id | 呼叫時所帶入的id | 用來判斷動作使用 |
| action | 動作 | 0選擇成功  1選擇取消 |
| data | 傳入的data String |  |

Sample：

※ 參數為物件會使用 JSON.stringify 先做轉變

APP.showMsgSelDialog({

  id: 'addCommonBill',

  title: 'Oops!',

  msg: '我知道你很懶，但我比你更懶，加入我們吧!!',

  btnNO: '先繳再說',

  btnOK: '立即註冊',

  data: {}

})

## API : 1-10 檢核是否已登入

定義：isLogin():Boolean

Input：

無

Output：

Boolean->true已登入 false未登入

Sample：

APP.isLogin()

## API : 1-11 檢核是否已登入，如未登入會顯示登入Dialog

定義：checkLogin():Boolean

Input：

無

Output：

Boolean->true已登入 false未登入

Sample：

APP.checkLogin()

## API : 1-12 登出並回到首頁

定義：logoutToHome()

Input：

無

Output：

無

Sample：

APP.logoutToHome()

## API : 1-12 顯示支付密碼驗證Dialog

定義：showPayVertify()

呼叫後會顯示支付密碼驗證的dialog，驗證成功會callback payVertifyPass()，

呼叫的網頁需準備payVertifyPass()來得知驗證成功。

Input：

無

Output：

無

Sample：( 還未實作該 API，屆時補充 )

## API : 1-13 轉跳到自動扣款設定頁

定義：routeAutoPaySetting(menuID:String,subMenuID:string,thirdMenuID:string,primaryData:String)

呼叫後會轉跳到新增設定自動扣款，並將對應的資料帶入。

Input：

menuID->menu id 如水費的menuid

subMenuID->子階層的menu id，如台北自來水的menu id

thirdMenuID->第三階的menu id，目前還沒有用到

primaryData->主欄位資料，如水費的水號

Output：

無

Sample：( 還未實作該 API，屆時補充 )

## API : 1-14 依menuID轉跳到指定頁

定義：routebyMenuID(menuID:String)

依menu id呼叫後會轉跳到指定的頁面（menu id 只能帶入在getconfig中menu有定義的項目）。

Input：

menuID->menu id

Output：

無

Sample：

APP.routeByMenuID('register')

## API : 1-15 轉跳到註冊頁

定義：routeToRegister()

轉跳到註冊頁面

Input：

無

Output：

無

Sample：( 還未實作該 API，屆時補充 )

## API : 1-16 是否已註冊

定義：isReged():Boolean

檢核是否已註冊

Input：

無

Output：

Boolean->true已註冊 false尚未註冊

Sample：( 還未實作該 API，屆時補充 )

## API : 1-17 取得登入資訊

定義：getLoginData():String

取得登入資訊

Input：

無

Output：

Json String，資料如下

{

“token”:”token”, //登入token

“userType”:”會員類別”, //None ,MMA, PayMember

“loginMethod”:”登入方法”, //Password, Graphic, FingerPrint, FaceID

“has\_sinopac\_acct”:boolean, //是否有永豐銀行帳戶

“has\_sinopac\_card”:Boolean, //是否有永豐信用卡

“custID”:”身分證字號”,

“mobile”:”手機號碼”,

“email”:”email”

}

Sample：

APP.getloginData()

## API : 1-18 關閉webview

定義：exitWeb()

關閉webview

Input：

無

Output：

無

Sample：

APP.exitWeb()

## API : 1-19 關閉webview並回到首頁

定義：exitWebToHome(param:String)

關閉webview並回到首頁，並依param的內容來決定首頁的動作。

Input：

Param->是一個Json String內容如下

{

“needLogin”:ture //用來指定回到首頁後是否要顯示登入dialog(如果已登入，給true也不會顯示)

“needQuickLogin”:true //到了首頁登入後是否顯示快速登入設定詢問

}

needLogin->

Output：

無

Sample：( 還未實作該 API，屆時補充 )

## API : 1-20 關閉webview並轉跳到繳費記錄頁

定義：exitWebToPayDetail(needLogin:boolean)

關閉webview並回到繳費記錄頁，並依needLogin來決定到繳費記錄頁後是否要顯示登入dialog。

Input：

needLogin->用來指定到繳費記錄頁後是否要顯示登入dialog(如果已登入，給true也不會顯示)

Output：

無

Sample：( 還未實作該 API，屆時補充 )

## API : 1-21 是否顯示App的Header

定義：showHeader(isShow:boolean)

在webview中是否要呈現header(在iOS中是指top navigation bar)。

Input：

isShow->是否呈header ,true呈現 false不呈現

Output：

無

Sample：

APP.showHeader(true)

## API : 1-22 是否顯示App的下方BottomNav

定義：showBottomNav(isShow:boolean)

在webview中是否要呈現下方的Bottom navigation(在ios指的是tab bar controller)。

Input：

isShow->是否呈下方bottom navigation,true呈現 false不呈現

Output：

無

Sample：

APP.showBottomNav(true)

## API : 1-23 啟動 OCR

定義：startOCRPersonID()

## API : 1-24 更改APP個人電話號碼

定義：updateMobile(mobile:String)

在webview中更新了會員的電話號碼，需要呼叫此API更新APP cache的電話號碼。

Input：

mobile->電話號碼 如:0953123456

Output：

無

Sample：

APP.updateMobile(‘0953123456’)

## API : 1-25 更改APP個人Email

定義：updateEmail(email:String)

在webview中更新了會員的email，需要呼叫此API更新APP cache的email。

Input：

email->郵件地址 如:freeman@sinopac.com

Output：

無

Sample：

APP.updateEmail(‘freeman@sinopac.com’)

## API : 1-26 通知APP更新個人會員等級

定義：levelupPayMember()

在webview中更新了會員等級，需要呼叫此API通知APP更新會員等級。

Input：

無

Output：

無

Sample：

APP.levelupPayMember()

## API : 1-27 取得手機是否有綁定等資訊

定義：getLoginTypeInfo()

在webview中取得綁定相關資訊。

Input：

無

Output：

Json String，資料如下

{

“user\_type”:”None”, //空置串或是None代表裝置無綁定,

PayMember綁定電支登入,MMA綁定mma登入

“loginMethod”:”登入方式”, //Password,Graphic,FingerPrint,FaceID,Bio

“cust\_id”:”綁定的身分ID”, //身分證

}

Sample：

APP.getLoginTypeInfo()

## API : 1-28 (iOS)取得Wallet中信用卡狀態

定義：getPassStatus(cardNoSuffixs:String):String

Webview呼叫iOS Native，依傳入cardNoSuffix(信用卡卡號後四碼)取得使用者的Wallet(手機錢包)中該信用卡的狀態和資料。若無資料則回覆空字串。

Input：

cardNoSuffixs->使用者所有信用卡卡號後四碼，以逗號區隔。

Ex : ”1234,4567,6789”

Output：

Json String，資料如下

[{

“PANS”:”信用卡後4碼”,

“DANS”:”虛擬卡後4碼”,

“device”:”加卡裝置來源”, //iPhone, Apple Watch,

“state”:”該卡狀態”//已啟動卡片(activated):**0**, 被發行者啟動卡片(requiresActivation):**1**,正在啟動卡片(activating):**2**, 使用者或發行者取消卡片或無法使用(suspended):**3**, 發行者撤銷卡片(deactivated):**4**

},{…}]

Sample：

setupWebViewJavascriptBridge((bridge) **=>** {

      bridge.callHandler(' getPassStatus ', '', (response) **=>** {

        return resolve(response);

      });

## API : 1-29 (iOS)Apple Pay加卡

定義：applePayAddCard(param:String)

使用者點選Webview加卡按鈕，Webview呼叫Native帶入該信用卡的Json， iOS Native 進行初始化Apple Pay加卡流程，如系統異常則初始化失敗。

使用者指定該信用卡Apple Pay加卡流程操作結束後Native會callback addCardFinishCallBack(action: Int, data: String)，回覆加卡流程成功或失敗，成功則回覆action與該卡狀態資料param，失敗僅回覆action，所以呼叫的Webview需準備addCardFinishCallBack (action: Int, data:String)。

Input：

param->是一個Json String內容如下

{

“cardHolderName”:”持卡人姓名”,

“cardName”:”卡片名稱”,

“cardNo”:”信用卡號”

}

Output：

Boolean->true初始化成功 false初始化失敗

Web Sample：

setupWebViewJavascriptBridge((bridge) **=>** {

      bridge.callHandler(' applePayAddCard ', '', (response) **=>** {

        return resolve(response);

      });

APP Sample：

[\_bridge callHandler:@”addCardFinishCallBack”data:”action:0,data:{}”]

Web Sample：

bridge.registerHandler("addCardFinishCallBack", **function** (res) {

*// 轉換獲取的資料*

**const** { id, action, data } = JSON.parse(res);

  })

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Description | Remark |
| action | 動作 | 0:加卡成功  1:加卡失敗 |
| data | 傳入的data String | {  “PANS”:”信用卡後4碼”,  “DANS”:”虛擬卡後4碼”,  “device”:”加卡裝置來源”, //iPhone, Apple Watch,  “state”:”該卡狀態”//已啟動卡片(activated):**0**, 被發行者啟動卡片(requiresActivation):**1**,正在啟動卡片(activating):**2**, 使用者或發行者取消卡片或無法使用(suspended):**3**, 發行者撤銷卡片(deactivated):**4**  } |

## API : 1-31 掃描身分證正面

定義：scanPersonID(id:String)

呼叫後會顯示掃描畫面，使用者操作後會call back dataCallBack(id: String,data: String)，所以呼叫的網頁需準備dataCallBack(id: String,data: String)。

Input：

Id->callback用來判斷動作的id

Output：

dataCallBack的資料會以Json String回覆，格是如下

{

“name”:”姓名”,

“birthday\_roc”:”yyyMMDD”,(0710101)

“issurance\_date\_roc”:”yyyMMDD”,(0940701) //發證日期

“issurance\_city”:”北市” //換證城市

“issurance\_type”:1 //換補發類型 0:初發 1:補發 2:換發

}

Sample：

APP.scanPersonID('personID')

## API : 1-32 掃描信用卡

定義：scanCreditCard(id:String)

呼叫後會顯示掃描畫面，使用者操作後會call back dataCallBack(id: String,data: String)，所以呼叫的網頁需準備dataCallBack(id: String,data: String)。

Input：

Id->callback用來判斷動作的id

Output：

dataCallBack的資料會以Json String回覆，格是如下

{

“card\_no”:”信用卡號碼”,

“holder\_name”:”test”, //持卡人姓名

“expire\_date”:”1024”// 卡片校期MMyy

}

Sample：

APP.scanCreditCard('personID')

## API : 1-33 掃描3碼瓦斯帳單bar code

定義：scanBarCodeGas(id:String)

呼叫後會顯示掃描畫面，使用者操作後會call back dataCallBack(id: String,data: String)，所以呼叫的網頁需準備dataCallBack(id: String,data: String)。

Input：

Id->callback用來判斷動作的id

Output：

dataCallBack的資料會以Json String回覆，格是如下

{

“code1”:”9碼字串”,

“code2”:”16碼字串”,

“code3”:”15碼字串”

}

Sample：

APP.scanBarCodeGas('gas')

## API : 1-34 掃描1碼水費帳單bar code

定義：scanBarCodeWater(id:String)

呼叫後會顯示掃描畫面，使用者操作後會call back dataCallBack(id: String,data: String)，所以呼叫的網頁需準備dataCallBack(id: String,data: String)。

Input：

Id->callback用來判斷動作的id

Output：

dataCallBack的資料會以Json String回覆，格是如下

{

“code”:”10碼水號字串”

}

Sample：

APP.scanBarCodeWater('water')