

# APP-H5 JS-Bridge 协议

H5应当在页面加载后的第一时间调用app的 `register` 方法

## H5可注册以下几个方法供APP调用

### onHandleIntent

```
1.  /**
2.  * 意图下发。开发者在DBP平台上面开发的意图，在匹配到对应用户query之后,会封装对应意图成为Intent下发下来。
3.  * APP通过{@link BotIntent#name}来确定意图名称，之后开发对应的逻辑。同时APP还可以通过{@link BotIntent#slots}
   解析参数
4.  *
5.  * @param token 指令的token,token是一个场景的标识符，在打开的时候，token会塞到intent中，key
   为"bot:accessToken"，可以通过token来区别这个指令是否是当前场景需要处理的指令
6.  * @param identity BotID参考{@link BotSdkImpl}包含packageName和accessToken
7.  * @param intent 意图内容，参考{@link BotIntent} 包含意图名称和槽位参数
8.  * @param customData 自定义数据，结构化数据，可选参数。参考BotSDK协议文档
   http://icode.baidu.com/repos/baidu/duer/open-platform-api-doc/blob/master:dueros-conversational-
   service/device-interface/bot-app-sdk-private.md
9.  */
10. bridge.registerHandler('onHandleIntent', function (payload, callback) {
11.     // payload 数据示例:
12.     {
13.         "app": {
14.             "packageName": {{string}}
15.         },
16.         "intent": {
17.             "name": {{string}}
18.         },
19.
20.         "token": {{string}},
21.         "customData": {{JSON}}
22.     }
23.     callback({Boolean}) 告知处理是否成功
24. });
```

### onClickLink

```
1.  /**
2.  * ClickLink事件下发,ClickLink是一种Directive,用户新增自定义交互之后，云端会解析用户定义的交互，下发对应的指令
3.  * 例如APP通过{@link BotSdk#updateUiContext(UiContextPayload)}新增自定义交互{@code <"试一试", url>},用户
   query"试一试"
4.  * 之后DuerOS会通过此接口下发上面定义的url
5.  * 如果用户引用系统内建自定义类型，用户query中可以包含参数，例如"输入北京"，这个query中北京可以被解析成参数，放到后
   面paramMap中下发
6.  * 系统内建类型参考: <a href="https://github.com/dueros/AndroidBotSdkDemo">DCS type文档</a>
7.  *
8.  * @param url 自定义交互描述中的url
9.  * @param paramMap 对于系统内建类型，参数列表。参数就是从query中通过分词取得的关键词。
10. */
11. bridge.registerHandler('onClickLink', function (payload, callback) {
12.     // payload数据示例
13.     {
```

```

14.         "url": "{{STRING}}",
15.         "params":{
16.             "{{STRING}}": "{{STRING}}"
17.         }
18.     }
19.     // 告知app是否处理成功
20.     callback({{Boolean}})
21. })

```

## onHandleScreenNavigatorEvent

```

1.
2. /**
3.  * 处理屏幕导航事件
4.  * @param event 屏幕导航事件值, 参考{@link BotMessageProtocol#NAV_GO_BACK}系列事件定义
5.  */
6.
7. const NAV_SCROLL_LEFT = 1;
8. const NAV_SCROLL_RIGHT = 2;
9. const NAV_SCROLL_UP = 3;
10. const NAV_SCROLL_DOWN = 4;
11. const NAV_NEXT_PAGE = 5;
12. const NAV_PREVIOUS_PAGE = 6;
13. const NAV_GO_BACK = 7;
14. const NAV_GO_HOMEPAGE = 8;
15.
16. bridge.registerHandler('onHandleScreenNavigatorEvent', function (payload, callback) {
17.     // payload参数上述常量中的一个 CONST
18.     callback({{Boolean}}) // 告知处理是否成功
19. })

```

## onLinkAccountSuccess

```

1. bridge.registerHandler('onLinkAccountSuccess', function (payload, callback) {
2.     // payload数据示例
3.     {
4.         "app":{
5.             "url":"{{STRING}}", // 可选
6.             "accessToken":"{{STRING}}" // 可选 ??? 为什么可选@matian
7.         }
8.     }
9.     // 告知app是否处理成功
10.    callback({{Boolean}})
11. })

```

## onChargeStatusChange

第三方开发者可以用此回调做页面的刷新。支付模块和第三方会有个服务端的通知通路，第三方开发者以服务端的通知订单数据为准。

```

1. bridge.registerHandler('onChargeStatusChange', function (payload, callback) {
2.     // payload数据示例
3.     {
4.         "token": "{{STRING}}",
5.         "app":{
6.             "packageName":"{{STRING}}",
7.             "accessToken":"{{STRING}}"

```

```

8.         },
9.         "authorizationDetails": {
10.             "authorizationAmount": {
11.                 "amount": "{{STRING}}",
12.                 "currencyCode": "CNY"
13.             },
14.             "capturedAmount": {
15.                 "amount": "{{STRING}}",
16.                 "currencyCode": "CNY"
17.             },
18.             "creationTimestamp": {{INT32}},
19.         },
20.         "baiduOrderReferenceId": "{{STRING}}"
21.         "sellerOrderId": "{{STRING}}",
22.         "purchaseResult": "{{ENUM}}",
23.         "message": "{{STRING}}"
24.     }
25.     // 告知app是否处理成功
26.     callback({{Boolean}})
27. })

```

## H5可调用APP的以下几个方法

### register

```

1.  /**
2.   * 向botsdk注册APP，注册信息会上传到云端进行校验，校验通过之后会收到注册成功{@link
   IBotMessageListener#onRegisterSucceed()}
3.   * 的回调。只有注册成功才可以进行语音交互
4.   * @param listener 回调对象{@link IBotMessageListener}
5.   * @param botId botId,参考DBP平台技能基础信息页面，botid
6.   * @param random1 随机数1，加密盐值，随便定义
7.   * @param signature1 签名1，计算方式{@code getMd5(rand + signKey)},其中signKey来自于DBP平台技能基础信息页
   面，签名key
8.   * @param random2 随机数2，加密盐值，随便定义，不应与random1相同
9.   * @param signature2 签名2，计算方式{@code getMd5(rand + signKey)},其中signKey来自于DBP平台技能基础信息页
   面，签名key
10.  * 具体格式如下
11.  // data数据格式
12.  {
13.      "random1";{{随机值}}
14.      "signature1":{{md5(random1 + appkey)}}
15.      "random2":{{随机值}}
16.      "signature2":{{md5(random2 + appkey)}}
17.  }
18.  */
19.  bridge.callHandler('register', data, function (payload) {
20.      // payload数据格式如下
21.      {
22.          "accessToken": "{{string}}"
23.      }
24.  });

```

### updateUiContext

```

1.  /**
2.   * 新增一个自定义交互元素描述{@link HyperUtterance}

```

```

3.      * @param url 用于确定用户query的url
4.      * @param utterances 支持的用户话术集合
5.      * @param type 类型, 自定义类型为link,系统还提供内建类型 input,select等等
6.      *      参考<a href="https://github.com/dueros/AndroidBotSdkDemo">自定义交互元素</a>
7.      * @param paramMap 参数列表
8.      * data格式如下
9.      {
10.         "enableGeneralUtterances": "{{Boolean}}",
11.         "hyperUtterances": [
12.             {
13.                 "url": "{{string}}"// 用于确定用户query的url
14.                 "utterances": "[{{string}}]", // 支持的用户话术集合
15.                 "type": "{{ENM}}", // 枚举类型, 自定义类型为link,系统还提供内建类型 input,select等等, 具体见下方链接
16.                 "parameters": "[{{"key": "{{string}}", "value": "{{string}}"}]} // 参数列表
17.             }
18.         ]
19.     }
20.     */
21.     bridge.callHandler('updateUiContext', data, function (result) {
22.         // result {{Boolean}} 是否调用成功
23.     })

```

## listen

```

1.     bridge.callHandler('listen', function (result) {
2.         // result {{Boolean}} 是否调用成功
3.     })

```

## speak

```

1.     /**
2.      * 播报一段文本
3.      data: 要播报的文本
4.      */
5.     bridge.callHandler('speak', data, function () {
6.         // TTS 播报完毕后调用此函数
7.     })

```

## requestClose

请求关闭APP容器

```

1.     {
2.         bridge.callHandler('requestClose');
3.     }

```

## requireLinkAccount

用户账号授权事件上报, 授权成功后会回调onLinkAccountSuccess

```

1.     bridge.callHandler('requireLinkAccount')

```

## requireCharge

发起购买，支付成功后会回调onChargeStatusChange指令，开发者可根据sellerOrderId关联订单号

```
1.  /**
2.  * data结构
3.  * {
4.  *   "packageName": "{{STRING}}",
5.  *   "token": "{{STRING}}",
6.  *   "chargeBaiduPay": {
7.  *     "authorizeAttributes": {
8.  *       "authorizationAmount": {
9.  *         "amount": "{{STRING}}",
10. *         "currencyCode": "CNY"
11. *       },
12. *       "sellerAuthorizationNote": "{{STRING}}"
13. *     },
14. *     "sellerOrderAttributes": {
15. *       "sellerOrderId": "{{STRING}}",
16. *       "productName": "{{STRING}}",
17. *       "productId": "{{STRING}}",
18. *       "description": "{{STRING}}",
19. *       "sellerNote": "{{STRING}}"
20. *     }
21. *   }
22. * }
23. */
24. bridge.callHandler('requireCharge', data)
```

## uploadLinkClicked

上报屏幕上的链接点击事件

```
1.  /**
2.  * data结构
3.  * {
4.  *   url: "{{string}}"
5.  * }
6.  */
7.
8.  {
9.    bridge.callHandler('uploadLinkClicked', data);
10. }
```