



# **Hosted Page Integration Guide**

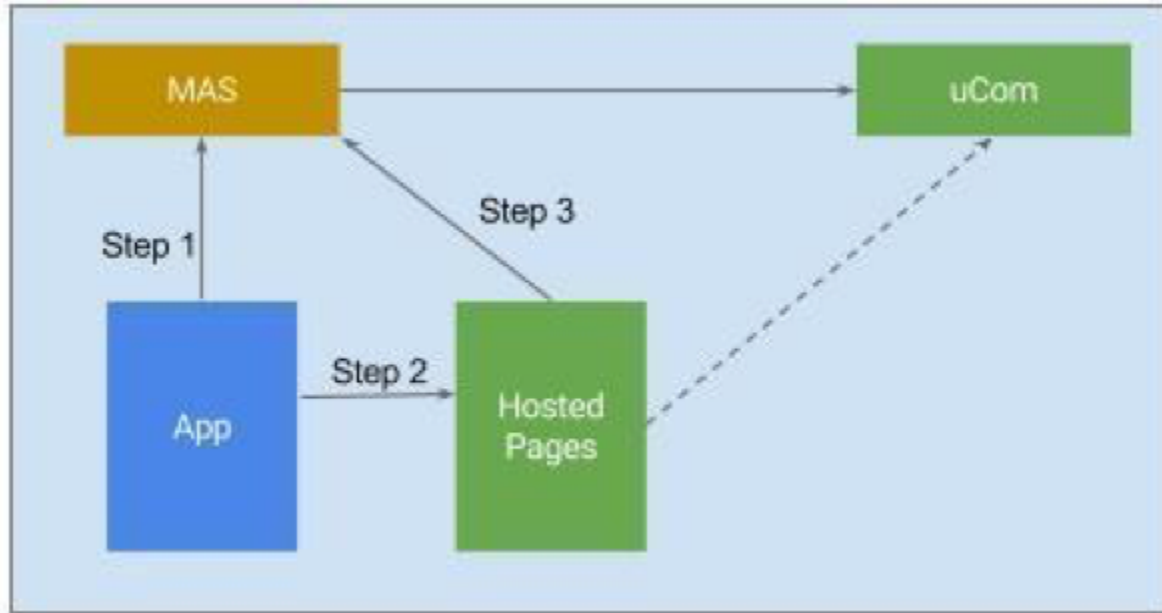
**Version 3.11**

The scope of the document is to describe how to use the Hosted Page. It is not intended to describe all the interactions between App, Mobile Application Server, and uCom Server.

<b>1. Roles .....</b>	<b>3</b>
1.1 Mobile Application (App).....	3
1.2 Mobile Application Server (MAS).....	3
1.3 Hosted Page (HP) .....	3
1.4 uCom Server (uCom).....	3
<b>2. Setup .....</b>	<b>4</b>
2.1. Api-Key.....	4
2.2. Api-Secret.....	4
2.3. Redirect Url/MAS Url (Asynchronous).....	4
2.4. FDCustomerId.....	4
2.5. PageLink (url and relation).....	4
<b>3. Flow .....</b>	<b>5</b>
3.1 Start a New Session .....	5
3.1.1. App calls MAS .....	6
3.1.2. MAS calls uCom to getToken .....	6
3.2 Load Hosted Page .....	8
3.2.1 Mobile Webview Integration Steps .....	8
3.2.2. Website Integration Steps .....	13
3.2.3. SDK Configuration property Value.....	16
3.3 Handle POST from Hosted Pages.....	17
3.3.1 Hosted Pages Response Payload .....	18
<b>4. Events .....</b>	<b>24</b>
4.1.1 onReady .....	24
4.1.2 onChange .....	24
4.1.3 onError .....	25
4.1.4 onSuccess .....	25
<b>5. API Error Status Codes.....</b>	<b>26</b>
5.1 Error Responses Before Screen Render .....	26
5.2 Error Responses After Form Submission .....	27
<b>6. Native/Web Button Submit.....</b>	<b>28</b>
<b>7. FAQ's .....</b>	<b>29</b>
7.1 How to enable CORS in server?.....	29
7.2 Multiple Languages Support .....	29
7.3 How to enable Threatmetrix feature on Hosted Pages.....	29
7.6 How to control the look and feel?.....	30
7.7 Local Validation by Hosted Page .....	31

# 1. Roles

There are 4 roles



**1.1 Mobile Application (App)** - This could be a native mobile application running on IOS or Android devices. This could also be a web application running on a browser.

**1.2 Mobile Application Server (MAS)** - This is a server which is a communication bridge between the App and uCom. It also stores uCom developer account information like app id, app secret, and encryption key securely.

**1.3 Hosted Page (HP)** - This is a web page rendered on native webview or browser which will display the UI to let user enter their account information like Credit card, debit card, gift card etc. This page securely captures the account information from user and passes it to uCom server. When finished, the page sends the result to returnUrl.

**1.4 uCom Server (uCom)** - This is the Fiserv solution server which provides all the apis. This server sits behind Apigee.

## 2. Setup

We need following parameters to use HP

**2.1. Api-Key** - This will be generated for you when you create an Apigee account. This should be saved securely on MAS and shared with App. Api-key is fixed for a merchant.

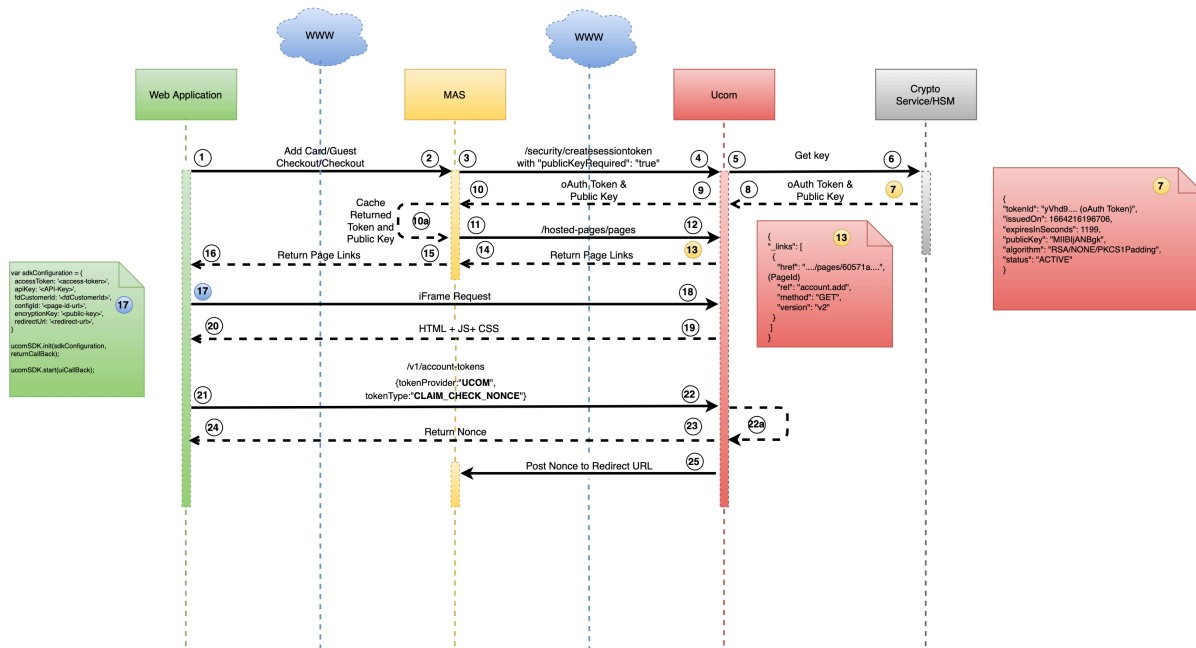
**2.2. Api-Secret** - This will be generated for you when you create an Apigee account. This should be saved securely on the application server. This should NOT be shared with app. Api-key is fixed for a merchant.

**2.3. Redirect Url/MAS Url (Asynchronous)** - All the Hosted pages responses(error/success) will be responded back to JavaScript main return callback only. Responses should be parsed and handled from JavaScript callback. Additionally same responses will be delivered to your MAS URL by Hosted pages via HTTP POST (Ajax Call) asynchronously. This API should be provided by MAS. MAS has to enable CORS for Fiserv origin "int.api.firstdata.com/cat.api.firstdata.com /prod.api.firstdata.com". This can be used for auditing purposes when web browser or app got crashed accidentally.

**2.4. FDCustomerId** - This must be obtained using other uCom apis. This is optional when you initiate SDK with guest checkout option.

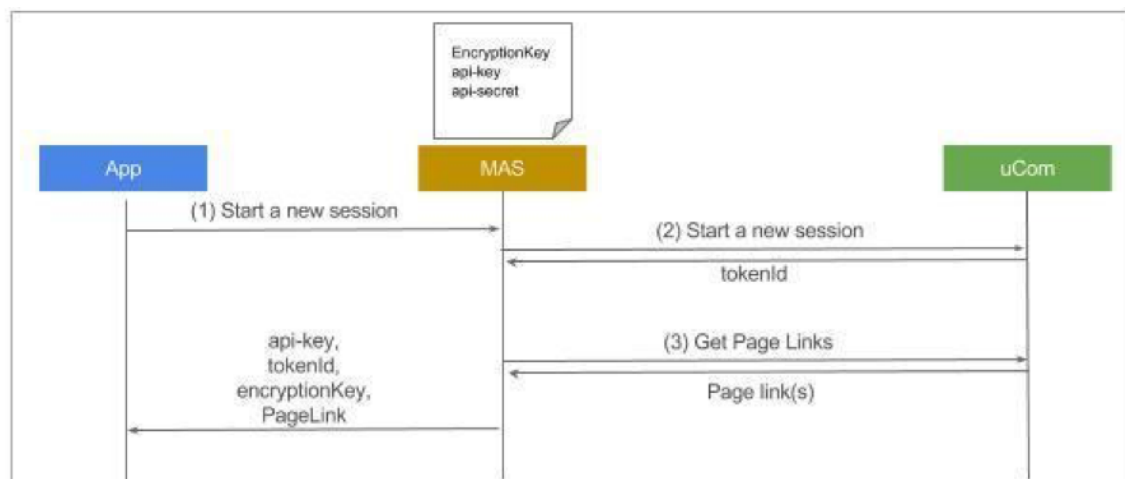
**2.5. PageLink (url and relation)** - This is the unique page which is going to display the use case. Url is the address where page is hosted, and Relation is the name of the use case. PageLink can be retrieved run time via the api (ucom/v1/hosted-pages/pages) and can be cached. We prefer that PageLink should be freshly fetched. The page contents are configured offline.

### 3. Flow



#### 3.1 Start a New Session

App calls MAS to get tokenId, encryptionKey and pageLink. tokenId and encryptionKey should not be cached or stored on the app and should be fetched from MAS. The tokenId and encryptionKey expires frequently and therefore this step should be done every time user starts the flow.



**3.1.1. App calls MAS** The api between app and MAS is not part of this document. It's up to the merchant to decide this part of the transaction.

**3.1.2. MAS calls uCom to getToken** MAS has to call uCom to get a tokenId. MAS should not cache the tokenId. getToken call will provide the one time session token and public key which needs to be passed to SDK to launch HP.

Api (HTTP POST)	<b>CAT:</b> <a href="https://int.api.firstdata.com/ucom/v1/tokens">https://int.api.firstdata.com/ucom/v1/tokens</a> <b>PREPROD:</b> <a href="https://cat.api.firstdata.com/ucom/v1/tokens">https://cat.api.firstdata.com/ucom/v1/tokens</a> <b>PROD:</b> <a href="https://prod.api.firstdata.com/ucom/v1/tokens">https://prod.api.firstdata.com/ucom/v1/tokens</a>
Document	
Headers	Content-Type = application/json Api-Key = {api-key} Authorization = HMAC {signature} Timestamp = {time UTC in milliseconds}
Body	{ "token" : { "fdCustomerId" : "123456789" } }, publicKeyRequired: true }
Sample Response HTTP 201	{ "tokenId": "cZq0YBz0ZuhS90Udzl8Nlp35uq4w", "fdCustomerId": "1234", "issuedOn": 1507684425488, "expiresInSeconds": 599, "publicKey": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAWoEfMkE6Ly12XoucblT1f7HvnBLSXZv0owzMKg57EqpWeB1F4JmeZTsqiC8X2t0xnhaY6SjD1xBEPfsFXN07smIsntWfzENPxyPhbtwqXtDauhfr1yqTxHRCz0393KwotFio6tkwLUr76zsQJ4tIm49zp4JAzE8gHK4S371k/X6Y0IF0efuzc9mLcg+L+fakRcVOMhF/HldKyw+tda4TBPE+S/RMdksoF+IYFaD668hzUrwMKoBYg1ZCc6YmnthWTIM1mWr5wGKYoQnMDWPsWAcG6N5r28vk4YHBfA9gnuXC7EehDkDk4CR3TjrIhg+W2yTkew8YJYRbKwUeEhZqQIDAQAB", "algorithm": "RSA/NONE/PKCS1Padding", "status": "ACTIVE" }

## How to generate HMAC signature

```
var key = ""; //the 'Api-Key'
var secret = ""; //the 'Api-Secret';

var time = new Date().getTime(); // use this time in header
var method = request.method;

var rawSignature = key + ":" + time;
var requestBody = request.data;

if(method != 'GET' && method != 'DELETE'){
    var payload_digest = CryptoJS.SHA256(requestBody);
    var b64BodyContent = CryptoJS.enc.Base64.stringify(payload_digest);
    rawSignature = rawSignature + ":" + b64BodyContent;
}

var signature =
CryptoJS.enc.Base64.stringify(CryptoJS.HmacSHA256(rawSignature, secret));
// use this in header
```

**3. MAS calls uCom to get page link** MAS can cache the page link for future reference though we do not recommend that. Merchant may have configured multiple pages and therefore this api will return all of them. Each page can be identified by the relation.

Api (HTTP GET)	<a href="https://int.api.firstdata.com/ucom/v1/hosted-pages/pages">https://int.api.firstdata.com/ucom/v1/hosted-pages/pages</a>
Document	
Headers	Content-Type = application/json Authorization = HMAC {signature} Api-Key = {api-key} Timestamp = {time UTC in milliseconds}
Sample Response	{ "_links": [ { "href": " <a href='\"https://int.api.firstdata.com/ucom/v1/hosted-pages/pages/f74237ec-112a-4204-a37b-2cfe9daa904f\"'>https://int.api.firstdata.com/ucom/v1/hosted-pages/pages/f74237ec-112a-4204-a37b-2cfe9daa904f</a> ", "rel": " account.add", "method": "GET", "version": "v2" } ] }

```
]
}
```

## 3.2 Load Hosted Page

To start the hosted page, app needs api-key, pageLink, tokenId, fdCustomerId, encryptionKey and redirectUrl. Mobile apps must make sure that they have disabled webview caching and enabled loading javascript in webview.

### 3.2.1 Mobile Webview Integration Steps

#### Step1:

Load the below URL in WebView

Environment	WebView URL
CAT	<a href="https://int.api.firstdata.com/ucom/v2/static/v2/mobile/int/ucom-sdk.html">https://int.api.firstdata.com/ucom/v2/static/v2/mobile/int/ucom-sdk.html</a>
PRE-PROD	<a href="https://cat.api.firstdata.com/ucom/v2/static/v2/mobile/cat/ucom-sdk.html">https://cat.api.firstdata.com/ucom/v2/static/v2/mobile/cat/ucom-sdk.html</a>
PROD	<a href="https://prod.api.firstdata.com/ucom/v2/static/v2/mobile/prod/ucom-sdk.html">https://prod.api.firstdata.com/ucom/v2/static/v2/mobile/prod/ucom-sdk.html</a>

#### Step2:

Call **uComClient.init()** javascript method with configuration objects after WebView is loaded

```
uComClient.init({
    "accessToken": "<token-id>",
    "apiKey": "<API-key>",
    //If guest checkout then do not pass fdCustomerId
    "fdCustomerId": "<fdCustomerId>",
    //Get it from pageLink
    "pageId": "<page-id>",
```



```
"encryptionKey": "<Encryption-Key>",
"redirectUrl": "<MAS-URL>"
});
```

Note: Please refer additional params section 3.2.3

### Step3:

Set local redirection listener

A redirection listener should be set on the webview to catch the event that HP has finished its work. HP will call this redirection in case of permanent failures and final success(nonce). A permanent failure is if js fails to load or Ajax call fails or tokenId has expired or encryptionKey invalid.

Once HP get response from uCom then it will do URL redirect with encoded URI.

### Redirection Listener URL:

ucom://finish?response=response-payload-object-string

**Note:** Webview redirection listener URL should be decoded before handle it.

#### 3.2.1.1 IOS Sample Code Snippets

```
1. #Load HP static URL in webview
2. - (void) successfulMASRequestWithResponseDictionary:(NSDictionary *)response {
3.     // Step 2 - Load Hosted Pages
4.     NSString *tokenId = response[@"tokenId"];
5.     NSString *apiKey = response[@"apiKey"];
6.     NSString *fdCustomerId = response[@"fdCustomerId"];
7.     NSString *url = response[@"href"];
8.     NSString *requestURL = [NSString stringWithFormat:@"https://%.api.firstdata.com/u
com/v2/static/v2/ucom-sdk.html", MAS.masSharedManager.environment];
9.     hostedPageRequest = [[NSMutableURLRequest alloc] initWithURL:[NSURL URLWithString:
requestURL]];
10.    // Load
11.    dispatch_async(dispatch_get_main_queue(), ^{
12.        [self->webView loadRequest:hostedPageRequest];
13.        [self->webView reloadFromOrigin];
14.    });
15. }
16.
17. #Once webview loaded then pass configuration object and call HP javascript method
18. - (void) webView:(WKWebView *)webView didFinishNavigation:(WKNavigation *)navigation
    {
19.     MAS *sharedInstance = [MAS masSharedManager];
20.     NSString *execTemplate =
21.     @"uComClient.init({
22.         \"apiKey\": \"%@\",
```

```

23.         \"redirectUrl\": \"%@\",
24.         \"fdCustomerId\": \"%@\",
25.         \"pageId\": \"%@\",
26.         \"accessToken\": \"%@\",
27.         \"encryptionKey\": \"%@\",
28.         \"env\": \"%@\"});";
29.     NSString *exec = [NSString stringWithFormat:execTemplate,sharedInstance.apiKey,
    _redirectUrl,
30. sharedInstance.fdCustomerId, sharedInstance.pageId, sharedInstance.bearer,
31. sharedInstance.encryptionKey, sharedInstance.environment];
32.     [_webView evaluateJavaScript:exec completionHandler:nil];
33.     if ([_activityIndicatorView isAnimating]) {
34.         [_activityIndicatorView stopAnimating];
35.     }
36. }
37.
38. #pragma mark WKNavigationDelegate methods (Redirection Listener Methods)
39. - (void) webView:(WKWebView *)webView decidePolicyForNavigationAction:(WKNavigation
Action *)navigationAction decisionHandler:(void (^)(WKNavigationActionPolicy))decisionH
andler {
40.     //Step 4 - Handle finish redirection
41.     NSString *requestURL = navigationAction.request.URL.absoluteString;
42.     NSLog(@"Redirect has occurred ...");
43.     NSLog(@"Request Url: %@", requestURL);
44.     if ([requestURL hasPrefix:@"ucom://finish"]) {
45.         // This is the clue that Hosted Posted is finished
46.         NSLog(@" -----
");
47.         NSLog(@"| REQUEST URL: %@", requestURL);
48.         NSLog(@" -----
");
49.         decisionHandler(WKNavigationActionPolicyCancel);
50.         NSString *resultStr = [[requestURL componentsSeparatedByString:@"ucom://fin
ish?response="].lastObject stringByRemovingPercentEncoding];
51.         UIAlertController *alertController = [UIAlertController alertControllerWith
Title:@"Result" message:resultStr preferredStyle:UIAlertControllerStyleAlert];
52.         UIAlertAction *okButton = [UIAlertAction actionWithTitle:@"OK" style:UIAler
tActionStyleDefault handler:^(UIAlertAction * _Nonnull action) {
53.             [self.navigationController popViewControllerAnimated:YES];
54.         }];
55.         [alertController addAction:okButton];
56.         [self presentViewController:alertController animated:YES completion:nil];
57.     } else {
58.         // Otherwise don't intercept
59.         decisionHandler(WKNavigationActionPolicyAllow);
60.     } }

```

### 3.2.1.2 Android Sample Code Snippets

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_web);
    // Load Hosted Pages
    String url = "";
    url =
String.format("https://%s.api.firstdata.com/ucom/v2/static/v2/mobile/%s/ucom-
sdk.html", getEnvironmentString(env), getEnvironmentString(env));
    FDLogger.INSTANCE.d(TAG,url);

    HashMap<String, String> headers = new HashMap<>();
    headers.put(Constants.Header.API_KEY, apiKey);
    headers.put(Constants.Header.AUTHORIZATION, Constants.Header.BEARER +
tokenId);
    headers.put(Constants.Header.TIMESTAMP, String.valueOf(new Date().getTime()));
    FDLogger.INSTANCE.map(TAG,headers);
    webView.loadUrl(url, headers);

    webView.setWebViewClient(new WebViewClient(){

        @Override
        public void onPageFinished(WebView view, String url) {
            super.onPageFinished(view, url);
            progressBar.setVisibility(View.GONE);
            String javascript = "";
            //Initialize the SDK with SDK configuration params by calling this
ucomSDK.init() method and pass required parameters
            javascript = String.format(
                "uComClient.init({" +
                "\"%s\":\"%s\"," +
                "\"%s\":\"%s\"," +
                "\"%s\":\"%s\"," +
                "\"%s\":\"%s\"," +
                "\"%s\":\"%s\"," +
                "\"%s\":\"%s\"," +
                "\"%s\":\"%s\"," +
                "\"%s\":\"%s\"" +
                "});",
                Constants.BundleKey.API_KEY, apiKey,
                Constants.BundleKey.REDIRECT_URL, redirectUrl,
                Constants.BundleKey.FD_CUSTOMER_ID, fdCustomerId,
                Constants.BundleKey.PAGE_ID, pageId,
```

```

        Constants.BundleKey.ACCESS_TOKEN, tokenId,
        Constants.BundleKey.ENCRYPTION_KEY, encryptionKey,
        Constants.BundleKey.ENV, getEnvironmentString(env));

        FDLogger.INSTANCE.d(TAG, javascript);
        webView.evaluateJavascript(javascript, null);
    }

    @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
    @Override
    public boolean shouldOverrideUrlLoading(WebResourceRequest
request) {
        Uri uri = request.getUrl();
        FDLogger.INSTANCE.d(TAG, uri.getScheme());
        FDLogger.INSTANCE.d(TAG, uri.getPath());
        // Handle finish redirection
        String resultString = request.getUrl().toString();
        String decodedstr = null;
        if (resultString.startsWith("ucom://finish")) {
            // This is the clue that Hosted Posted is finished
            Log.i(TAG, "REQUEST URL: " + resultString);

            try {
                decodedstr = URLDecoder.decode(resultString, "UTF-8");
            } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
            }
            Log.i(TAG, "Decoded String: " + decodedstr);

            if (!resultString.startsWith("ucom://finish?result=failure")) {
                finish();
                return true;
            }
        } else {
            // Otherwise don't intercept
            Log.i(TAG, "REQUEST URL: " + resultString);
        }
        return super.shouldOverrideUrlLoading(view, request);
    }
});
}

```

### 3.2.2. Website Integration Steps

1. Include the uCom SDK library on head tag on your html page

**URL:** <https://<env>.api.firstdata.com/ucom/v2/static/v2/js/ucom-sdk.js>

**Environment Variable:** <int/cat/prod>

Variable	Environment
INT	CAT/CERT
CAT	PRE-PROD
PROD	PRODUCTION

2. Initialize the SDK with SDK configuration params by call this **ucomSDK.init()** method
  - a. Pass access token
  - b. Pass API Key
  - c. Pass fdCustomerId
  - d. Pass pageURL
  - e. Pass Mount Id where SDK needs to be mounted on the screen
  - f. Pass EncryptionKey
  - g. Pass RedirectUrl
3. Call **uComSDK.start()** method to render SDK on mounted element on your page
4. Call **uComSDK.stop()** whenever you want to kill the SDK from the page.

**Note:** The javascript method should be called after the web content is loaded. Refer the sample code below

### 3.2.2.1 Web Sample Code

```
<html>

<head>
    <!-- uCom SDK -->
    <script src="https://int.api.firstdata.com/ucom/v2/static/v2/js/ucom-
sdk.js"></script>
</head>

<body>
    <!-- uCom SDK container -->
    <div id="ucom-container"></div>
    <script>
        /*
        // return Callback method
        // Hosted Pages Response (Error/Success) will be returned to this callback
once the hosted page actions get completed.
        */
        function returnCallBack(response) {
            console.log('UCOM response:', response);
            /* Kill the uCom UI */
            ucomSDK.stop();
            /* Do rest of the flow here */
        }

        //Configuration object
        var configs = {
            "accessToken": "<access-token>",
            "apiKey": "<API-KEY>",
            "fdCustomerId": "<transaction-id>",
            "pageUrl": "https://int.api.firstdata.com/ucom/v1/hosted-
pages/pages/<your-page-id>",
            "mountId": "ucom-container",
            "encryptionKey": "<encryption-key>",
            "redirectUrl": "<redirect-url>",

            /*Below attribute should be enabled when you integrate and test it on
http://localhist in lower (CAT) environment. This attribute should be removed in
higher environment*/
            /* "debug": true */

        }
    </script>
</body>
</html>
```

```

    ucomSDK.init(<configs>, <callbackMethod>, <isLoggingEnabled>);
    @params
        configs          - configurations object,
        callbackMethod    - callback method,
        isLoggingEnabled - pass true to enable console debug logging(true/false)
    */
    //Initialize the uCom SDK with configurations
    ucomSDK.init(configs, returnCallBack);

    //Event Subscribe
    ucomSDK.on('ready', function() {
        //Handle ready event
    });

    ucomSDK.on('change', function(event) {
        if (event.formValid) {
            //Enable pay button
        } else {
            //Disable pay button
        }
    });

    ucomSDK.on('error', function(response) {
        //Handle Error Response
    });

    ucomSDK.on('success', function(response) {
        //Handle nonce
    });

    //Start the uCom UI
    ucomSDK.start();
</script>
</body>
</html>

```

Please refer below table for additional sdk configurations properties

### 3.2.3. SDK Configuration property Value

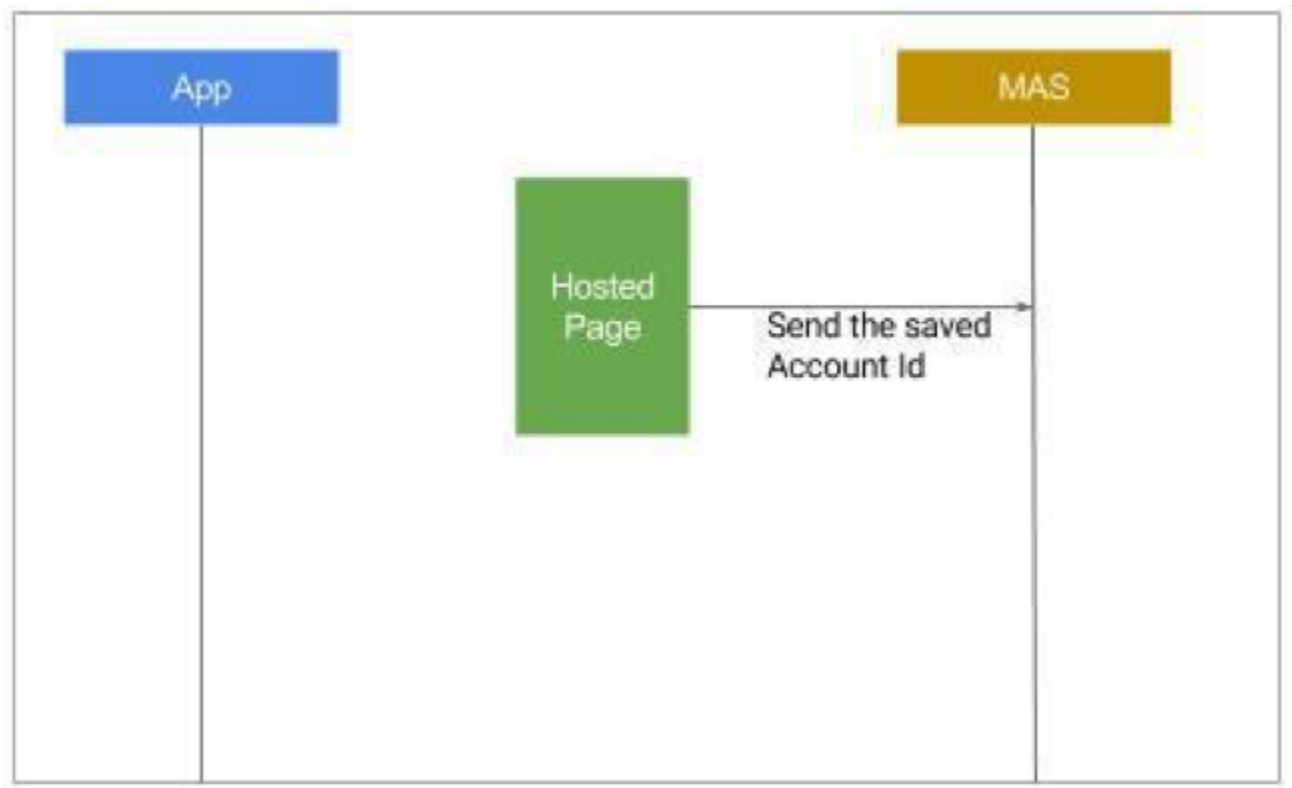
SDK Params	Required/Optional	Description
accessToken	R	Session Token Id
apiKey	R	API Key
fdCustomerId	R	First data Customer Id
PageURL	R	Page link url of the page
mountId	R	Mount Id where HP should be rendered
encryptionKey	R	Public key from session token response to encrypt the data
redirectUrl	O	MAS URL to capture all the hosted pages response for auditing purpose.
orgId	O	Org id should be passed if Threatmetrix should be enabled on HP
sessionId	O	Session id should be passed if Threatmetrix should be enabled on HP
extraObject	O	Additional details can be passed to HP to generate nonce as part of card detail. Eg: "billingAddress": { "type": "work", "streetAddress": "100 universal city plaza", "locality": "Hollywood", "region": "CA", "postalCode": "98290", "country": "US", "formatted": "100 universal city plaza, Hollywood, CA 98290 USA", "primary": true }
debug	O	Eg: debug: true This attribute should be passed if you develop and integrate it on localhost ( <a href="http://localhost">http://localhost</a> ) and bypass



		the https error on CAT environment. Note: This attribute should be removed in higher environment.
--	--	--

### 3.3 Handle POST from Hosted Pages

Once HP is finished, it will send the result app callback URL and redirectUrl(POST URL). MAS must implement this api to receive the result body. MAS will have to enable CORS on their end to allow access from javascript originating from “int.api.firstdata.com / prod.api.firstdata.com”.



### 3.3.1 Hosted Pages Response Payload

#### 3.3.1.1 New Card

##### 3.3.1.1.1 Failure

This is the failure response payload from uCom API

```
{
  "response": {
    "code": "279912",
    "message": "Decryption failed.",
    "category": "common",
    "developerInfo": {
      "developerMessage": "Decryption failed due to invalid/expired keyId.",
      "fieldError": [
        {
          "field": "KeyId/Algorithm error.",
          "message": "crypto-service: Single use key has already been used (CR008): Key#9bff66d610b48efb829a409c8d619f1dc8306da8fb10d997abbed44fc353fa21"
        }
      ]
    }
  }
}
```

##### 3.3.1.1.2 Success

This is the success response payload from uCom API

```
{
  "type": "CREDIT",
  "token": {
    "tokenType": "CLAIM_CHECK_NONCE",
    "tokenProvider": "UCOM",
    "tokenId": "4f0dd98e-bf56-499c-b562-7936ca20964c"
  },
  "isSaveCard": true,
  "credit": {
    "nameOnCard": "Michael John",
    "cardType": "VISA",
    "alias": "2345"
  }
}
```

### 3.3.1.1.3 Success Response with SDK Error

This is the success response with sdk errors payload. Sometimes card will be enrolled successfully but SDK will fail to post the response into redirect URL due to some reason. In this case SDK will send back with success response with SDK errors.

```
{
  "type":"CREDIT",
  "token":{
    "tokenType":"CLAIM_CHECK_NONCE",
    "tokenProvider":"UCOM",
    "tokenId":"4f0dd98e-bf56-499c-b562-7936ca20964c"
  },
  "isSaveCard":true,
  "credit":{
    "nameOnCard":"Michael John",
    "cardType":"VISA",
    "alias":"2345"
  },
  "developerInfo":{
    "developerMessage":"Redirect URL is failure",
    "fieldError":[{
      "code": 1013,
      "field":"Redirect URL",
      "message":"Redirect URL is failure"
    }]
  }
}
```

### 3.3.1.1.4 Success with Threatmetrix Details

This is the enrollment response with TM(Threatmetrix) payload from uCom API

```
{
  "type":"CREDIT",
  "token":{
    "tokenType":"CLAIM_CHECK_NONCE",
    "tokenProvider":"UCOM",
    "tokenId":"4f0dd98e-bf56-499c-b562-7936ca20964c"
  },
  "isSaveCard":true,
  "credit":{
    "nameOnCard":"Michael John",
    "cardType":"VISA",
    "alias":"2345"
  },
  "userSysDetails":{
```

```

    "orgId":"8cz43sdv",
    "sessionId":"2fb1a98a-7182-497f-bc2a-79c37e556cb2"
  }
}

```

### 3.3.1.1.5 Success with Extra Params Details

Merchant has the ability to pass the billing address into SDK. If they inject the billing address into SDK then that information will be part of the response.

```

{
  "type":"CREDIT",
  "token":{
    "tokenType":"CLAIM_CHECK_NONCE",
    "tokenProvider":"UCOM",
    "tokenId":"4f0dd98e-bf56-499c-b562-7936ca20964c"
  },
  "isSaveCard":true,
  "credit":{
    "nameOnCard":"Michael John",
    "cardType":"VISA",
    "alias":"2345",
    "billingAddress": {
      "type": "work",
      "streetAddress": "100 universal city plaza",
      "locality": "Hollywood",
      "region": "CA",
      "postalCode": "98290",
      "country": "US",
      "formatted": "100 universal city plaza, Hollywood, CA 98290 USA",
      "primary": true
    }
  }
}

```

### 3.3.1.2 Vaulted Card

#### 3.3.1.2.1 Failure

This is the failure response payload from uCom API

```
{
  "response":{
    "code":"279912",
    "message":"Decryption failed.",
    "category":"common",
    "developerInfo":{
      "developerMessage":"Decryption failed due to invalid/expired keyId.",
      "fieldError":[
        {
          "field":"KeyId/Algorithm error.",
          "message":"crypto-service: Single use key has already been used
(CR008): Key#9bff66d610b48efb829a409c8d619f1dc8306da8fb10d997abbed44fc353fa21"
        }
      ]
    }
  }
}
```

#### 3.3.1.2.2 Success

This is the success response payload from uCom API

```
{
  "type":"VAULTED_ACCOUNT",
  "token":{
    "tokenType":"CLAIM_CHECK_NONCE",
    "tokenProvider":"UCOM",
    "tokenId":"4f0dd98e-bf56-499c-b562-7936ca20964c"
  },
  "credit":{
    "cardType":"VISA",
    "alias":"2345"
  }
}
```

### 3.3.1.2.3 Success Response with SDK Error

This is the success response with sdk errors payload. Sometimes card will be enrolled successfully but SDK will fail to post the response into redirect URL due to some reason. In this case SDK will send back with success response with SDK errors.

```
{
  "type": "VAULTED_ACCOUNT",
  "token": {
    "tokenType": "CLAIM_CHECK_NONCE",
    "tokenProvider": "UCOM",
    "tokenId": "4f0dd98e-bf56-499c-b562-7936ca20964c"
  },
  "credit": {
    "cardType": "VISA",
    "alias": "2345"
  },
  "developerInfo": {
    "developerMessage": "Redirect URL is failure",
    "fieldError": [{
      "code": 1013,
      "field": "Redirect URL",
      "message": "Redirect URL is failure"
    }]
  }
}
```

### 3.3.1.2.4 Success with Threatmetrix Details

This is the enrollment response with TM(Threatmetrix) payload from uCom API

```
{
  "type": "VAULTED_ACCOUNT",
  "token": {
    "tokenType": "CLAIM_CHECK_NONCE",
    "tokenProvider": "UCOM",
    "tokenId": "4f0dd98e-bf56-499c-b562-7936ca20964c"
  },
  "credit": {
    "cardType": "VISA",
    "alias": "2345"
  },
  "userSysDetails": {
    "orgId": "8cz43sdv",
  }
}
```

```
    "sessionId":"2fb1a98a-7182-497f-bc2a-79c37e556cb2"
  }
}
```

### 3.3.1.2.5 Success with Extra Params Details

Merchant has the ability to pass the billing address into SDK. If they inject the billing address into SDK then that information will be part of the response.

```
{
  "type":"VAULTED_ACCOUNT",
  "token":{
    "tokenType":"CLAIM_CHECK_NONCE",
    "tokenProvider":"UCOM",
    "tokenId":"4f0dd98e-bf56-499c-b562-7936ca20964c"
  },
  "credit":{
    "cardType":"VISA",
    "alias":"2345",
    "billingAddress": {
      "type": "work",
      "streetAddress": "100 universal city plaza",
      "locality": "Hollywood",
      "region": "CA",
      "postalCode": "98290",
      "country": "US",
      "formatted": "100 universal city plaza, Hollywood, CA 98290 USA",
      "primary": true
    }
  }
}
```

## 4. Events

The only way to communicate with HP is by listening to an event. HP will emit and communicate back if you are subscribed with those events.

```
ucomSDK.on(event, handler);
```

### 4.1.1 onReady

Triggered when iFrame is fully rendered and can accept user's inputs

```
ucomSDK.on('ready', function() {  
  //Handle ready event  
});
```

### 4.1.2 onChange

Triggered when form value changed. The event payload always contains object with form valid status.

```
ucomSDK.on('change', function() {  
  if (event.formValid) {  
    //Enable pay button  
  } else {  
    //Disable pay button  
  }  
});
```

#### Handler Event Object

```
{  
  "elementType": "payment",
```



```
"formValid": false
}
```

### 4.1.3 onError

Triggered when HP's API errors. The event payload object contains API error which needs to be handled on app. Please refer the **section 3** for error payload structure.

```
ucomSDK.on('error', function(response) {
  //Handle Error Response
});
```

### 4.1.4 onSuccess

Triggered when HP's nonce generated. The event payload object contains nonce which needs to be handled on app. Please refer the **section 3** for error payload structure.

```
ucomSDK.on('success', function(response) {
  //Handle Nonce
});
```

## 5. API Error Status Codes

Below error status code needs to be handled from client side. These API error responses will be communicated back to App JavaScript main callback to handle the errors and show the appropriate error dialog.

### 51 Error Responses Before Screen Render

Following errors will be thrown before hosted pages screen render

Status Code	Transaction Status Desc	Comments
269904	Configuration record(s) not found.	When we pass the invalid page id then API will throw this error
401	Unauthorized	This error will occur when we pass invalid access token id.

Example response payload

```
{
  "response": {
    "code": "269904",
    "message": "Configuration record(s) not found.",
    "category": "common",
    "developerInfo": {
      "developerMessage": "Configuration record(s) not found."
    }
  }
}
```

```
{
  "response": {
    "code": 401,
    "message": "Unauthorized",
    "category": "common",
    "developerInfo": {
      "developerMessage": "Unauthorized"
    }
  }
}
```

```
}  
}  
}
```

## 5.2 Error Responses After Form Submission

Following errors will be thrown after hosted pages screen render

Status Code	Transaction Status Desc	Comments
279912	Decryption failed.	When we pass the invalid or already used public key into SDK then API will throw this error.
269901	Unable to process your request, please try again later, if problem persist, contact sys admin.	This error will occur when our backend is down.
401	Access Token expired / Unauthorized	This error will occur when user idle and try to submit the rendered form with expired access token.

Example response payload:

```
{
  "response": {
    "code": "279912",
    "message": "Decryption failed.",
    "category": "common",
    "developerInfo": {
      "developerMessage": "Decryption failed due to invalid/expired
keyId.",
      "fieldError": [{
        "field": "KeyId/Algorithm error.",
        "message": "crypto-service: Single use key has already been
used (CR008): Key#9bff66d610b48efb829a409c8d619f1dc8306da8fb10d997abbed44fc353fa21"
      }]
    }
  }
}
```

```
{
  "response": {
    "code": 401,
    "message": "Unauthorized",
    "category": "common",
    "developerInfo": {
      "developerMessage": "Unauthorized"
    }
  }
}
```

## 6. Native/Web Button Submit

HP will allow to submit the form through mobile native button or website button from outside iFrame. Following command will trigger the save action

```
//Trigger form save from outside iFrame or web view
ucomSDK.triggerSaveAction();
```

## 7. FAQ's

### 7.1 How to enable CORS in server?

Browser/App will be sending the preflight request (OPTIONS method) instead of actual method (POST/GET) to server when access different domain in ajax call. In this case, we have to handle and enable CORS in server to handle this preflight request. OPTIONS method has to be handled and respond with following request:

```
access-control-allow-headers →ACCEPT, fdCustomerId, ORIGIN,
AUTHORIZATION, CONTENT-TYPE
access-control-allow-methods →POST, GET, OPTIONS
access-control-allow-origin →*
allow →GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, PATCH
connection →Keep-Alive
content-language →en-US
content-length →0
date →Tue, 19 Dec 2017 23:14:23 GMT
keep-alive →timeout=10, max=100
x-powered-by →Servlet/3.0
```

Once browser/app receives the above as a response header then, it will subsequently invoke actual methods.

### 7.2 Multiple Languages Support

HP supports all texts in multiple languages. This process is manual and someone has to provide all texts displayed by HP in different languages. Please note that error messages from uCom server do not support multiple languages in current version.

### 7.3 How to enable Threatmetrix feature on Hosted Pages

Threatmetrix feature can be enabled on HP when you pass the orgId and sessionId in SDK configuration parameters.

## 7.6 How to control the look and feel?

HP customization is limited. But it provides option to change the following style properties


1. Background Color, Text color
2. Label Changes
3. Native Button (Form can be triggered from outside of iFrame)
4. Fields position changes
5. Label position (Above or floating) Changes
6. Hint text changes

### Sample Form

#### Default View

Name on Card

Card Number



Expiry Date

CVV



Zip Code

☐ By saving this card, you agree that this account will be saved and used for purchases. You can unregister the card at any time in the account settings.

All card information is securely encrypted for your protection.

Done

## Error View

Name on Card

Name on Card is required

Card Number



Card number is required

Expiry Date

Expiry date is required

CVV

Security Code

CVV code is required

Zip Code

What is your billing zip code?

- ☒ By saving this card, you agree that this account will be saved and used for purchases. You can unregister the card at any time in the account settings.

All card information is securely encrypted for your protection.

Done

## 7.7 Local Validation by Hosted Page

HP will do its own local validation of fields and show error to user. Local validation is configurable.