Citcon UPI WebSDK

Version 0.1.0

Version No.	Modify Activity	Modify Description	Editor	Modify Date
0.1.0	Creation	Paypal & Venmo Support	Long Zhou	2022-05-04

1. Introduction

Citcon UPI SDK was designed for online merchants to integrate Citcon payment solutions effortlessly into their own app. By using the SDK, merchant developers can focus on business logic without having to understand the plumbing of payment transactions. This version of the SDK supports PayPal , Venmo.

2. Integration Steps

As an example, this guide uses jQuery framework for frontend and PHP for backend.

Please Note:

Citcon UPI has two kinds of access token, one is client access token, which has inquiry and charge permissions. This is the correct access token for WebSDK / Mobile SDK use.

The other one is server access token, which has full permissions: inquiry, charge, refund, capture, vault.

In Citcon WebSDK. Only client access token is needed.

Step 1: Load Core JS in html page

Load Web SDK core at the bottom of the html page, for example before </body> or after </html>. You can either use the latest version or specify a version

UAT

```
<script
src="https://cdn.uat01.citconpay.com/latest/core/citconpay.core.js"></scri
pt>
```

Version base

```
<script
src="https://cdn.uat01.citconpay.com/v0.1.0/core/citconpay.core.js"></scri
pt>
```

Production

```
<script
src="https://cdn.citconpay.com/latest/core/citconpay.core.js"></script>
```

Version base

```
<script
src="https://cdn.citconpay.com/v0.1.0/core/citconpay.core.js"></script>
```

Step 2: Create access token and pending charge from server back end

Frontend Sample Code:

```
$.ajax({
  url: merchantUrl + '?action=create_transaction',
  type:'post',
  dataType: 'json',
  data: JSON.stringify({
    reference: transaction_reference,
```

```
    totalAmount: parseInt($("#txtAmount").val()),
    currency: $("#currency").val(),
    countryCode:$("#country").val()
  }),
  success:function(resp){
    console.log('create pending transaction...' + JSON.stringify(resp));
    if(resp.status === 'success'){
      access_token = resp.data.access_token;
      chargeToken = resp.data.charge_token;
      transactionId = resp.data.transaction_id;
    }else{
      console.log(resp.data);
    }
  },
  async:false
});
```

This only a PHP sample code, you may need to implement by your own

Back End Sample Code

```
.....
case 'create_transaction':
  {
      //create a pending transaction
      $resp = get_access_token();
      $data_array = json_decode($resp,true);
      if($data_array['status']=='success'){
        $token = $data_array['data']['access_token'];
        //do create pending transaction
        $transaction = create_pending_transaction($token);
        $transaction_array = json_decode($transaction,true);
        if($transaction_array['status']=='success'){
          $resp_token = array(
            'charge_token' => $transaction_array['data']['charge_token'],
            'transaction_id' => $transaction_array['data']['id'],
            'access_token' => $token
          );
          echo make_response(
            $transaction_array['status'],
            $resp_token
          );
        }else{
```

```php
            echo make_response(
                $transaction_array['status'],
                $transaction_array['data']['message']
            );
        }
    }else{
        echo make_respone($data_array['status'], $data_array['data']['message']);
    }
  }
  break;
 ........
 function make_response($status,$data){
   $resp = array(
       'status' => $status,
       'data' => $data
   );
   return json_encode($resp);
}

function get_access_token(){
   $data = array(
       "token_type" => "client"
   );
   $url = API_URL .'/access-tokens';

   //xxxx is your merchant private key which get from citcon
   $resp = do_http_post_with_token($url,'xxxx',json_encode($data));

   return $resp;
}
function create_pending_transaction($token){

   $json = file_get_contents('php://input');
   $json_data = json_decode($json,true);
   $urls = array (
       "ipn_url" => "https://dev.citconpay.com",
       "success_url" => "https://dev.citconpay.com",
       "fail_url" =>  "https://dev.citconpay.com",
       "mobile_url" => "https://dev.citconpay.com"
   );
   $transaction = array(
         "reference" => $json_data['reference'],
         "amount"=> $json_data['totalAmount'],
         "currency"=> $json_data['currency'],
```

```php
        "auto_capture"=> false,
        "country"=> $json_data['countryCode'],
        "urls"=> $urls
    );
    $data_array['transaction'] = $transaction;
    $data = json_encode($data_array);
    $url = API_URL ./'/charges';
    $resp = do_http_post_with_token($url,$token,$data);
    return $resp;
}
function do_http_post_with_token($url,$token,$data){

    $curl = curl_init($url);
    curl_setopt($curl, CURLOPT_URL, $url);
    curl_setopt($curl, CURLOPT_POST, true);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);

    $headers = array(
    "Accept: application/json",
    "Authorization: Bearer ".$token,
    "Content-Type: application/json",
    );
    curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);

    curl_setopt($curl, CURLOPT_POSTFIELDS, $data);

    //for debug only!
    curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);

    $resp = curl_exec($curl);
    curl_close($curl);
    // var_dump($resp);
    return $resp;
}
```

Step 3: Initialize WebSDK Core with access token

Add a element in the HTML file to host WebSDK widgets, such as

```
<div id="citcon-client-container"></div>
```

Web SDK then can mount a widget.

Sample code

```
// the payment method is what merchant choosed at onboarding.
let paymentMethodArray =['paypal','venmo'];
let defaultPaymentMethod = 'paypal';

const configObj = {
  accessToken: access_token,
  environment:'uat', ///uat/prod,This is optional. default is production
  debug: true, // true/false, This is optional. default is false
  consumerID: "18000"
};

citconpay.client.core(configObj).then( clientInstance=>{
  console.log(' Init SDK...' + JSON.stringify(clientInstance));
  // mount UI
  // class name is css class, you can define your own class at your style sheet
  citconInstance = clientInstance;
  clientInstance.mount('#citcon-client-container',{
    classname: 'payment-method-select-component',
    paymentMethods: paymentMethodArray,
    selectedPaymentMethod: defaultPaymentMethod,
  }, sdkUIDidInitialized).then(function(instance) {

    console.log('instance ...' + JSON.stringify(instance));
    //Do Events Register after WebSDK core init complete
    registerEvents();
  }).catch(error=>{
    console.log(' mount error:' + JSON.stringify(error));
  });
}).catch(error=>{
  console.log(' Init SDK error:' + JSON.stringify(error));
});
function sdkUIDidInitialized(e){
```

```
 console.log(' sdk ui initialized .....' + JSON.stringify(e));
}
```

Step 4: Register event listeners

Supported events include the following:

1.`payment-method-selected`

This event is triggered when consumer clicks on the Radio Button

2.payment-method-submitted

This event is triggered when consumer clicks the "Pay Now" button

3.payment-status-changed

This event is triggered when payment is completed. It calls back the listener with payment results.

Sample Return Object in JSON:

Success:

```
{
   "status":"success",
   "data":{
       "object":"charge",
       "id":"da949a90cb1011ecb9a3550ac676a90f",
       "reference":"0wqd3i0s5p8483q62f5x",
       "amount":180,
       "amount_captured":null,
       "amount_refunded":null,
       "currency":"USD",
       "time_created":1651603379000,
       "time_captured":null,
       "auto_capture":false,
       "status":"authorized",
       "country":"US",
       "payment":{
          "method":"paypal"
       }
       ...
    }
}
```

fail:

```
{
    "status":"fail",
    "data":{
        "code":xxxx,
        "message":"xxxxx",
     }
```

```
}
```

Sample Code:

```
function registerEvents(){

  citconInstance.on('payment-method-selected', function (e) {
    console.log('Inside `payment-method-select` .........' + JSON.stringify(e));
    //Do you code here
  });
   citconInstance.on('payment-method-submitted', function(e) {
    // UI Event
    // For those payment method Popup Or Redirect, Sucha as Paypal, Venmo
    // merchant handles selection of payment methods through Citcon's UI
    console.log('....paynow..click.....');
    $("body").addClass("loading");
    const requestOptions ={
        payment: {
          chargeToken:chargeToken,
          countryCode:$("#country").val(),
          transactionReference: transaction_reference
        },
        billing_address: {
          street: $("#address").val(),
          street2:$("#address2").val(),
          city: $("#txtCity").val(),
          state: $("#state").val(),
          zip: $("#zip").val(),
          country: $("#country").val()
        },
        consumer:{
         id:"18000",
         reference: "consumer_test_1",
         firstName: $("#firstName").val(),
         lastName: $("#lastName").val(),
```

```
          phone: $("#phone").val(),
          email: $("#email").val(),
        }
      }

citconInstance.onPaymentMethodSubmitted(e.paymentMethod,requestOptions).then(rest=>{
        console.log('pay now click, return..' + JSON.stringify(rest));
      }).catch(error=>{
        $("body").removeClass("loading");
        console.log('pay now click, error..' + JSON.stringify(error));
      });
    });
    //on payment status change, this is the event for charge result
    //status: status, status will return "success" or "failed"
    //retObj: retObj
    citconInstance.on('payment-status-changed', function(e) {
      const status = e.status;
      const res = e.retObj;

      console.log('payment-status-change status..' + status + JSON.stringify(res));
      if(status == 'success'){
        //payment success
        // do your code here....
      }else{
        $("body").removeClass("loading");
        if (res.code === 'VENMO_CANCELED') {
          console.log('App is not available or user aborted payment flow');
        } else if (res.code === 'VENMO_APP_CANCELED') {
          console.log('User canceled payment flow');
        } else {
          console.log('An error occurred:', res.message);
        }
      }
    });
}
```

Additional functions

Charge inquiry

You can use this function to inquire charge status

Sample code

```
$("body").addClass("loading");
console.log('inquire order... id=' + transactionId);
citconInstance.inquire(transactionId).then(resp=>{
  console.log('inquire...return' + JSON.stringify(resp));
  $("body").removeClass("loading");
}).catch(error=>{
  console.log('inquire order error...' +
JSON.stringify(error));
  $("body").removeClass("loading");
});
```

That's everything, happy coding!