

**Global Payments**

Global Payments Storefront Reference Architecture Integration Guide

**Table of Contents**

[1. Introduction 3](#_Toc107568390)

[2. API Flow Diagram 4](#_Toc107568391)

[3. Install the cartridge, node modules and import the metadata. 5](#_Toc107568392)

[Step 1: Install the cartridge 5](#_Toc107568393)

[Step 2: Install Node modules 5](#_Toc107568394)

[Step 3: Import the metadata 5](#_Toc107568395)

[4. Set up the cartridge and Workspace 6](#_Toc107568396)

[Step 1: Set up the cartridge path 6](#_Toc107568397)

[Step 2: Changes in Business Manager for site preferences 6](#_Toc107568398)

[Step 3: Set up an API key and Merchant key 6](#_Toc107568399)

[Step 4: Set up workspace with Visual Studio code 8](#_Toc107568400)

[Failover/Recovery Process 9](#_Toc107568401)

[Supported Locales 10](#_Toc107568402)

[5. Configure the Payment Methods 10](#_Toc107568403)

[1. Hosted fields: 10](#_Toc107568404)

[2. Google Pay 22](#_Toc107568405)

[3. PayPal 24](#_Toc107568406)

[4. Apple Pay 29](#_Toc107568407)

[6. Order Management 40](#_Toc107568408)

[1. Refund Procedure: 40](#_Toc107568409)

[2. Capture Procedure: 41](#_Toc107568410)

[*Global payments document links* 42](#_Toc107568411)

[*Release History* 42](#_Toc107568412)

# Introduction

The Intent of this document is to give end user a step by step guide to integrate the Global payments in SFRA cartridge. It will support to integrate the hosted fields, PayPal, Apple Pay and Google Pay in SFRA cartridge.

Global payments is payment provider which supports credit, debit cards and the digital wallets. You can find out more about what Global Payments does [here](https://www.globalpayments.com/en-ap).

**System requirements**

* **Categories:** Payment Processing
* [**LINK Cartridge version**](https://documentation.b2c.commercecloud.salesforce.com/DOC1/topic/com.demandware.dochelp/content/b2c_commerce/topics/getting_started/b2c_welcome.html)**:** 22.6.0
* **API Fleet and version:** Global paymentsversion 2021-03-22
* **Commerce Platform Version:** This package was implemented against SF B2C version 22.6 and tested against compatibility mode 21.7. This package was developed against SFRA versions 6.0.0. This document covered the link cartridge support for SFRA.

Please contact Global Payments Inc. team before you install the Salesforce B2C LINK Cartridge.

Email on [salesforce@globalpay.com](mailto:salesforce@globalpay.com)

Request credentials, configure the available payment options within Drop-in and receive your LINK key. Additionally: Use of the Global Payments cartridge requires credentials and keys from Global Payments (contact on salesforce@globalpay.com).

The cartridge is developed and tested for the US locale and is compatible with any locale if you add the localization strings.

# API Flow Diagram

Table

Description automatically generated

# Install the cartridge, node modules and import the metadata.

### Step 1: Install the cartridge

The Global Payments LINK cartridge can be used with Storefront Reference Architecture (SFRA).

1. Using SFRA, install **int\_globalpay** and **int\_globalpay\_sfra** using Commerce Cloud UX-studio.
2. Using a controller-based integration, install **int\_globalpay** , and **int\_globalpay\_controllers** cartridges using Commerce Cloud UX-studio.

### Step 2: Install Node modules

From the project's root directory, install Node modules using your command line:

npm install

### Step 3: Import the metadata

To add new configuration items, import the predefined metadata:

1. Download the [installation package](https://github.com/globalpayments/globalpayments-salesforce-commerce-cloud-b2c/tree/main/Link_globalPay) on our GitHub page.
2. Open the **package/metadata/site\_import/sites/** folder.
3. Rename the **yourSiteId**folder to the ID of your site in the Business Manager.
4. Zip the **site\_import** folder.
5. In the Business Manager, go to **Administration** > **Site Development** > **Site Import & Export** and import the zipped file.

# Set up the cartridge and Workspace

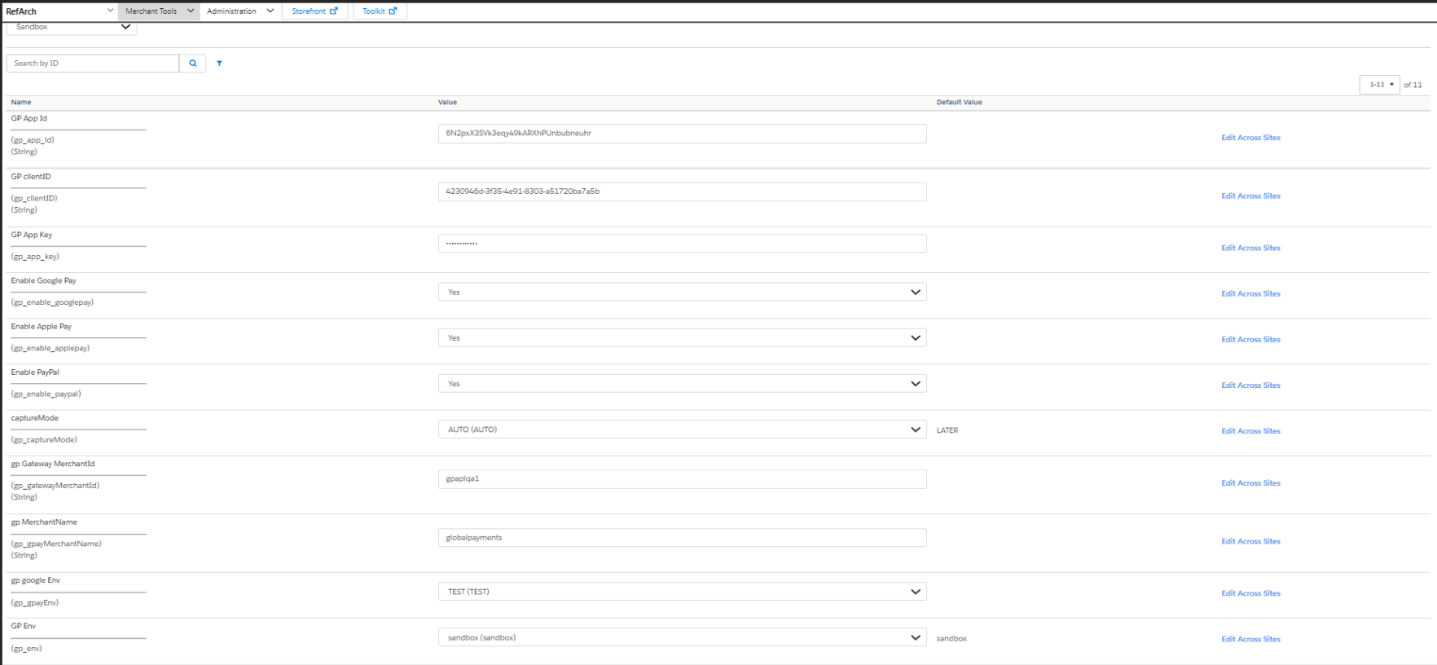
1. [Create a test account with Global Payments](https://developer.globalpay.com/). With this, you will be able to access the [test Customer Area](https://developer.globalpay.com/resources/test-card-numbers), and apply for a live account. While you wait for your live account credentials, you can start integrating by using your test account.
2. Finish the [metadata import](https://docs.adyen.com/plugins/salesforce-commerce-cloud/install-the-cartridge-and-import-the-metadata).

### Step 1: Set up the cartridge path

1. Go to **Administration** > **Sites** > **Manage Sites**>**[yourSite]**>**Settings.**
2. In the**Cartridges**box:
   * 1. If you are integrating in SFRA, the cartridge path is as-  
        **int\_globalpay\_sfra:int\_globalpay:app\_storefront\_base**
3. Select **Apply**.

### Step 2: Changes in Business Manager for site preferences

1. In the Business Manager, go to Merchant tools > Site Preferences > Custom Preferences > GlobalPay.
2. In the merchant Code box, enter the name of your Globalpay merchant account for which the payments will be processed.
3. Set Test/Production mode to (TEST).
4. Select Save in the upper-right corner of the screen.

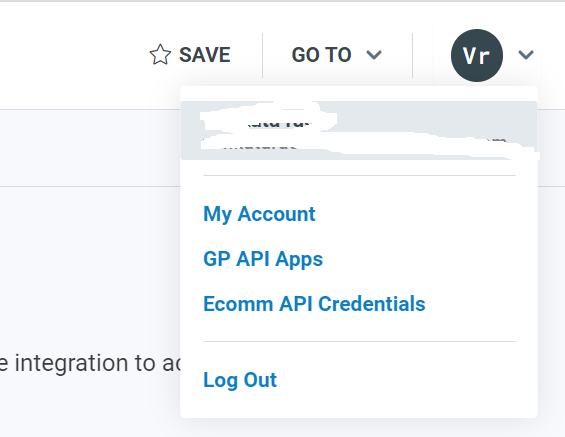


### Step 3: Set up an API key and Merchant key

To connect your Commerce Cloud store to the Globalpay API, you need to get an API key from Globalpay need to update into the Business Manager Site preferences.

Generate an API key in the Globalpay Customer Area

1. Log in to your [Customer Area](https://developer.globalpay.com/), if you do not have account, you can create the account in development portal.
2. Click on following profile icon.



Click on ‘GP API Apps’, you can get merchant id and api key and Account & Permissions.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. Copy the API key and Merchant key to the Business Manager site preference.

### Step 4: Set up workspace with Visual Studio code

1. Create a folder “Globalpay” in your workspace and place the cartridge (int\_globalpay\_sfra and int\_globalpay) downloaded from GitHub.
2. If you have a different project set-up, you will need to open the file ‘/package.json’ and modify the paths.base value to point to your ‘app\_storefront\_base’ cartridge. This path is used by the JS and SCSS build scripts.
3. If using VSCode, install the extension Prophet Debugger link or any other SFCC extension and include below in dw.json ().

NOTE: If you are using different IDE, refer respective developer guide to setup the workspace.

Step 1: Build and Upload the code

Prerequisite: Install node under " Globalpay" folder.

Install sgmf-scripts Compile the Code Upload the code

Upload the code to Salesforce Commerce Cloud instance

{

"hostname": "your-sandbox-hostname.demandware.net",

"username": "yourlogin",

"password": "yourpwd",

"version": "version\_to\_upload\_to"

]

}

Step 2: Build and Upload the code Prerequisite:

Install node under "Link\_globalPay" folder.

install sgmf-scripts

Install sgmf-scripts with following command

npm install sgmf-scripts

Compile the Code

Compile JS and SCSS with following command

npm run compile:js && npm run compile:scss

Upload the code

npm run uploadCartridge

### Failover/Recovery Process

There are no single points of failure. There are multiple reasons why you might need to recover a payment. But it needn’t be a manual process. We have solution which is auto-retries it will attempt immediately after the first decline. In case of any failover, please open support case @ <https://developer.globalpay.com/support/terminal>

### Supported Locales

Globalpay cartridge support only for the locale like English (United States).

### Unit testing

To run unit and integration test cases following are the commands you need to run.

>npm run test

>npm run test:integration

# Configure the Payment Methods

Payment methods

After the [metadata import](https://docs.adyen.com/plugins/salesforce-commerce-cloud/install-the-cartridge-and-import-the-metadata), Globalpay has been added as the payment processor for credit cards and for the local payment methods that have been [set up for your merchant account](https://docs.adyen.com/payment-methods#add-payment-methods-to-your-account) in the Adyen [Customer Area](https://ca-test.adyen.com/).

To make sure that this has been set up correctly:

1. In the Business Manager, go to Merchant tools > Ordering > Payment Methods.
2. In the ID column, select CREDIT*\_CARD*  and check that Payment Processor is set to GLOBALPAY\_CREDITCARD.

### 1. Hosted fields:

Hosted fields are available in some countries. To learn more about their availability, see  [Hosted fields](https://developer.globalpay.com/ecommerce/hosted-fields-guide).

1. To configure credit card payment: In the Business Manager, go to Merchant tools > Ordering > Payment Methods.
2. Select the ‘CREDIT\_CARD’ id and enable or disable
3. In the Business Manager, go to Merchant tools > Ordering > Payment Methods.
4. In the ID column, select CREDIT\_CARD and check that Payment Processor is set to GLOBALPAY\_CREDITCARD

Globalpaycreditcards.isml is included in creditCardForm.isml (OOO file). All card validations are happening through Globalpay libraries from Globalpay end.

Text

Description automatically generated

After updating the above changes, the credit card form looks like as below,

Graphical user interface, text, application

Description automatically generated

After entering credit card details (card#, exp date, CVV) in checkout page in payment form, you can see the call (payment-methods) in browser console in network section.

With help of GlobalPay plugin (globalpayments-3ds.js), below 2 controllers are invoked.

1. Globalpay-Authentication
2. Globalpay-Initiation

**3DS2**:

3D Secure 2 (3DS2) is the new authentication protocol for online card payments. 3DS2 is designed to improve upon 3D Secure 1 (3DS1) by addressing the old protocol's pain points, and delivering a much smoother and integrated user experience.

3DS2 popup will appear only for specific cards. On checkout page, if the card is 3DS2 type card then the following popup appears for the payment verification.

Graphical user interface, application

Description automatically generated

If the specified card is 3DS2 card type, then making an additional API call to Global Pay. (Url-https://apis.sandbox.globalpay.com/ucp/authentications/<AuthId>/result).

**3DS1**:

The original version, 3DS1 can support SCA compliance for PSD2. It can also provide merchant fraud liability protection, but only until October 2021 for Visa Secure - 3DS1.

For the 3DS1 card, user can see below popup to process with the payment.

Graphical user interface, application

Description automatically generated

Globalpay test cards:   
<https://developer.globalpay.com/resources/test-card-numbers>

API URL:   
<https://documenter.getpostman.com/view/12989945/UVJeGwg4>

**Sequence of API calls for hosted fields:**

**1. Access Token**

Uri: https://apis.sandbox.globalpay.com/ucp/accesstoken

**Request**:

{ "app\_id": "5KnUFba9o2ZcGrFp71PKL897ne0Hlbpx",

"nonce": "\*\*\*\*\*",

"secret": "0e1f3250de675c023fec16139256e082e7ed7366673d5b5069445828e3caac9f35252536e85e0b973603d6f88b46996c38f1afbbdadfdc437e3e78128654f67d",

"grant\_type": "client\_credentials"

}

**Response**:

{

"token": "LbmF3L5pfwgwJu849VIPdugSvI9S",

"type": "Bearer",

"scope": {

"merchant\_id": "MER\_7e3e2c7df34f42819b3edee31022ee3f",

"merchant\_name": "Sandbox\_merchant\_3",

"accounts": [

{

"id": "TRA\_c9967ad7d8ec4b46b6dd44a61cde9a91",

"name": "transaction\_processing",

"permissions": [

"TRN\_POST\_Reauthorize",

"TRN\_POST\_Capture\_Multiple",

"TRN\_POST\_Authorize",

"TRN\_POST\_Adjust",

"TRN\_POST\_Refund",

"TRN\_GET\_Single",

"TRN\_POST\_Initiate",

"TRN\_POST\_Refund\_Standalone",

"TRN\_POST\_Reverse",

"TRN\_POST\_Confirm",

"TRN\_POST\_Capture",

"TRN\_GET\_List",

"ACT\_POST\_Multiple",

"ACT\_GET\_Single",

"ACT\_GET\_List",

"ACC\_GET\_List",

"ACC\_GET\_Single",

"GET\_Single",

"VER\_POST\_Verify",

"AUT\_POST\_Results",

"AUT\_POST\_Initiate",

"AUT\_POST\_Check\_Availability",

"CCS\_POST\_DCC"

]

},

{

"id": "TRA\_2f92c641eb7f43449541a038baea9154",

"name": "dcc\_rate",

"permissions": [

"TRN\_POST\_Reauthorize",

"TRN\_POST\_Authorize",

"TRN\_POST\_Adjust",

"TRN\_POST\_Refund",

"TRN\_GET\_Single",

"TRN\_POST\_Initiate",

"TRN\_POST\_Refund\_Standalone",

"TRN\_POST\_Reverse",

"TRN\_POST\_Confirm",

"TRN\_GET\_List",

"ACT\_POST\_Multiple",

"ACT\_GET\_Single",

"ACT\_GET\_List",

"ACC\_GET\_List",

"ACC\_GET\_Single",

"GET\_Single",

"VER\_POST\_Verify",

"AUT\_POST\_Results",

"AUT\_POST\_Initiate",

"AUT\_POST\_Check\_Availability",

"CCS\_GET\_Bin\_Ranges",

"CCS\_POST\_DCC"

]

},

{

"id": "TRA\_baa3d4f50fda4d7c9341815c79576f27",

"name": "dcc",

"permissions": [

"TRN\_POST\_Reauthorize",

"TRN\_POST\_Authorize",

"TRN\_POST\_Refund",

"TRN\_POST\_Adjust",

"TRN\_GET\_Single",

"TRN\_POST\_Initiate",

"TRN\_POST\_Refund\_Standalone",

"TRN\_POST\_Confirm",

"TRN\_POST\_Reverse",

"TRN\_GET\_List",

"ACT\_POST\_Multiple",

"ACT\_GET\_Single",

"ACT\_GET\_List",

"ACC\_GET\_List",

"ACC\_GET\_Single",

"GET\_Single",

"VER\_POST\_Verify",

"AUT\_POST\_Results",

"AUT\_POST\_Initiate",

"AUT\_POST\_Check\_Availability",

"CCS\_GET\_Bin\_Ranges",

"CCS\_POST\_DCC"

]

},

{

"id": "TKA\_b3a46f0f351f43cfad20acf5c32fea50",

"name": "tokenization",

"permissions": [

"ACT\_POST\_Auto\_Action",

"ACT\_POST\_Multiple",

"ACT\_GET\_Single",

"ACT\_GET\_List",

"ACC\_GET\_List",

"ACC\_GET\_Single",

"GET\_Single",

"PMT\_POST\_Create\_Single",

"PMT\_POST\_Detokenize",

"PMT\_POST\_Search",

"PMT\_GET\_List",

"PMT\_POST\_Create",

"PMT\_GET\_Single",

"PMT\_PATCH\_Edit"

]

},

{

"id": "DIA\_d4f75884b1e54ae4a5e904155f629f26",

"name": "dispute\_management",

"permissions": [

"ACT\_POST\_Multiple",

"ACT\_GET\_Single",

"ACT\_GET\_List",

"ACC\_GET\_Single",

"ACC\_GET\_List",

"GET\_Single",

"DIS\_POST\_Challenge",

"DIS\_POST\_Accept",

"DIS\_GET\_Lists",

"DIS\_GET\_Single"

]

},

{

"id": "DAA\_fabf29a777724d83998f6cc4747b7d9a",

"name": "settlement\_reporting",

"permissions": [

"TRN\_GET\_List\_Funded",

"TRN\_GET\_Single\_Funded",

"ACT\_POST\_Multiple",

"ACT\_GET\_Single",

"ACT\_GET\_List",

"ACC\_GET\_List",

"ACC\_GET\_Single",

"GET\_Single",

"DIS\_GET\_List\_Funded",

"DIS\_GET\_Single\_Funded",

"DEP\_GET\_List",

"DEP\_GET\_Single"

]

}

]

},

"app\_id": "5KnUFba9o2ZcGrFp71PKL897ne0Hlbpx",

"app\_name": "vipul\_sandbox",

"time\_created": "2022-05-27T09:28:15.756Z",

"seconds\_to\_expire": 86399,

"email": "vipul.nanavati@Mindtree.com"

}

**2. Payment-methods**

URL: https://apis.sandbox.globalpay.com/ucp/payment-methods

**Request**:

{

"reference": "000000001",

"card": {

"number": "4012001038488884",

"expiry\_month": "12",

"expiry\_year": "25"

}

}

**Response**:

{

"id":"PMT\_039a07d3-d0ad-4c66-af3f-17be287569a0",

"time\_created":"2022-06-16T14:22:41.785Z",

"status":"ACTIVE",

"usage\_mode":"SINGLE",

"merchant\_id":"MER\_7e3e2c7df34f42819b3edee31022ee3f",

"merchant\_name":"Sandbox\_merchant\_3",

"account\_id":"TKA\_b3a46f0f351f43cfad20acf5c32fea50",

"account\_name":"tokenization",

"reference":"f654b7f7-e947-a63d-a547-6abce6126029",

"card":{

"masked\_number\_last4":"XXXXXXXXXXXX8884",

"brand":"VISA",

"expiry\_month":"12",

"expiry\_year":"30"

},

"action":{

"id":"ACT\_loHFoJyX1uDW5GEceXLwEFmNPBSduG",

"type":"PAYMENT\_METHOD\_CREATE",

"time\_created":"2022-06-16T14:22:41.785Z",

"result\_code":"SUCCESS",

"app\_id":"6N2pxX3SVk3eqy49kARXhPUnbubneuhr",

"app\_name":"sample\_app\_CERT"

}

}

3. **Authentications**

Url: <https://apis.sandbox.globalpay.com/ucp/authentications>

**Request**:

{"account\_name":"transaction\_processing","channel":"CNP","country":"US","reference":"93459c79-f3f9-427d-84d9-ca0584bb55bf","amount":15997,"currency":"USD","source":"BROWSER","payment\_method":{"id":"PMT\_039a07d3-d0ad-4c66-af3f-17be287569a0"},"notifications":{"challenge\_return\_url":"https://zyem-006.sandbox.us01.dx.commercecloud.salesforce.com/on/demandware.store/Sites-RefArch-Site/en\_US/GlobalPay-ThreeDSSecureChallenge","three\_ds\_method\_return\_url":"https://zyem-006.sandbox.us01.dx.commercecloud.salesforce.com/on/demandware.store/Sites-RefArch-Site/en\_US/GlobalPay-ThreeDsMethod"},"token":"tQZ1YadUkrn2pHkKsm2OW6WkdoOe"}

**Response**:

{

"id": "AUT\_57f7abd6-ee0a-45da-ac53-71352f1ea758",

"time\_created":"2022-06-16T14:22:44.941Z",

"transaction\_type": "SALE",

"status": "AVAILABLE",

"channel": "CNP",

"amount": "15997",

"currency": "USD",

"country": "US",

"merchant\_id": "MER\_7e3e2c7df34f42819b3edee31022ee3f",

"merchant\_name": "Sandbox\_merchant\_3",

"account\_id": "TRA\_c9967ad7d8ec4b46b6dd44a61cde9a91",

"account\_name": "transaction\_processing",

"reference":"93459c79-f3f9-427d-84d9-ca0584bb55bf",

"three\_ds" :{

"server\_trans\_ref": "57f7abd6-ee0a-45da-ac53-71352f1ea758",

"message\_version": "2.1.0",

"ds\_protocol\_version\_start": "2.1.0",

"ds\_protocol\_version\_end": "2.1.0",

"acs\_protocol\_version\_start": "2.1.0",

"acs\_protocol\_version\_end": "2.1.0",

"acs\_info\_indicator": ["NOT\_AVAILABLE"],

"enrolled\_status": "ENROLLED",

"eci": "",

"liability\_shift": "",

"challenge\_model": "",

"challenge\_status": "",

"session\_data\_field\_name":"threeDSSessionData",

"message\_type": "creq",

"challenge\_value": "",

"redirect\_url": "",

"acs\_challenge\_request\_url":"",

"method\_url": "https://test.portal.gpwebpay.com/pay-sim/sim/acs",

"method\_data": {

"three\_ds\_server\_trans\_id": "57f7abd6-ee0a-45da-ac53-71352f1ea758",

"three\_ds\_method\_return\_url": "https://zyem-006.sandbox.us01.dx.commercecloud.salesforce.com/on/demandware.store/Sites-RefArch-Site/en\_US/GlobalPay-ThreeDsMethod",

"encoded\_method\_data": "ewogICJ0aHJlZURTU2VydmVyVHJhbnNJRCIgOiAiNTdmN2FiZDYtZWUwYS00NWRhLWFjNTMtNzEzNTJmMWVhNzU4IiwKICAidGhyZWVEU01ldGhvZE5vdGlmaWNhdGlvblVSTCIgOiAiaHR0cHM6Ly96eWVtLTAwNi5zYW5kYm94LnVzMDEuZHguY29tbWVyY2VjbG91ZC5zYWxlc2ZvcmNlLmNvbS9vbi9kZW1hbmR3YXJlLnN0b3JlL1NpdGVzLVJlZkFyY2gtU2l0ZS9lbl9VUy9HbG9iYWxQYXktVGhyZWVEc01ldGhvZCIKfQ"

}

},

"notifications": {

"challenge\_return\_url": "https://zyem-006.sandbox.us01.dx.commercecloud.salesforce.com/on/demandware.store/Sites-RefArch-Site/en\_US/GlobalPay-ThreeDSSecureChallenge"

},

"action":{

"id":"ACT\_Z4RtwQONzCkbLj6MUkGxnsMJci7DE9",

"type":"CHECK\_AVAILABILITY",

"time\_created":"2022-06-16T14:22:44.941Z",

"result\_code":"SUCCESS",

"app\_id":"6N2pxX3SVk3eqy49kARXhPUnbubneuhr",

"app\_name":"sample\_app\_CERT"

}

}

**4. Initiate**

URL: <https://apis.sandbox.globalpay.com/ucp/authentications/AUT_57f7abd6-ee0a-45da-ac53-71352f1ea758/initiate>

**Request**:

{"three\_ds":{"source":"BROWSER","preference":"NO\_PREFERENCE"},"method\_url\_completion\_status":"YES","auth\_id":"AUT\_57f7abd6-ee0a-45da-ac53-71352f1ea758","merchant\_contact\_url":"http://www.vacationtoplan.in/shopping/contact/","order":{"time\_created\_reference":"2022-06-16T14:22:50.652Z","amount":15997,"currency":"USD","address\_match\_indicator":true,"shipping\_address":{"line1":"address","city":"Bangalore","postal\_code":"12345"}},"payment\_method":{"id":"PMT\_039a07d3-d0ad-4c66-af3f-17be287569a0"},"browser\_data":{"accept\_header":"\*/\*","color\_depth":"TWENTY\_FOUR\_BITS","ip":"136.226.254.114","java\_enabled":false,"javascript\_enabled":true,"language":"en-US","screen\_height":720,"screen\_width":1280,"challenge\_window\_size":"WINDOWED\_600X400","timezone":-5.5,"user\_agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.0.0 Safari/537.36"},"token":"tQZ1YadUkrn2pHkKsm2OW6WkdoOe"}

**Response**:

{"id":"AUT\_57f7abd6-ee0a-45da-ac53-71352f1ea758","time\_created":"2022-06-16T14:22:53.094Z","time\_last\_updated":"2022-06-16T14:22:44.941Z","transaction\_type":"SALE","status":"CHALLENGE\_REQUIRED","channel":"CNP","amount":"15997","currency":"USD","country":"US","source":"BROWSER","three\_ds":{"server\_trans\_ref":"57f7abd6-ee0a-45da-ac53-71352f1ea758","acs\_trans\_ref":"ced86380-ed7f-41ec-8c10-436a798d313b","acs\_reference\_number":"3DS\_LOA\_ACS\_INTE\_020200\_00293","ds\_trans\_ref":"20e337d2-1a0a-4e9f-acce-13eed714aa34","eci":"","liability\_shift":"","status":"CHALLENGE\_REQUIRED","status\_reason":"","authentication\_source":"BROWSER","message\_version":"2.1.0","authentication\_value":"","cardholder\_response\_info":"","message\_category":"PAYMENT\_AUTHENTICATION","redirect\_url":"https://apis.sandbox.globalpay.","acs\_challenge\_request\_url":"https://acs2p.test.gpe.cz/tds/challenge/brw/ced86380-ed7f-41ec-8c10-436a798d313b","challenge\_status":"MANDATED","challenge\_model":"OUT\_OF\_BAND\_CHALLENGE","session\_data\_field\_name":"threeDSSessionData","message\_type":"creq","challenge\_value":"ewogICJtZXNzYWdlVHlwZSIgOiAiQ1JlcSIsCiAgIm1lc3NhZ2VWZXJzaW9uIiA6ICIyLjEuMCIsCiAgInRocmVlRFNTZXJ2ZXJUcmFuc0lEIiA6ICI1N2Y3YWJkNi1lZTBhLTQ1ZGEtYWM1My03MTM1MmYxZWE3NTgiLAogICJhY3NUcmFuc0lEIiA6ICJjZWQ4NjM4MC1lZDdmLTQxZWMtOGMxMC00MzZhNzk4ZDMxM2IiLAogICJjaGFsbGVuZ2VXaW5kb3dTaXplIiA6ICIwNCIKfQ"},"notifications":{"challenge\_return\_url":"https://zyem-006.sandbox.us01.dx.commercecloud.salesforce.com/on/demandware.store/Sites-RefArch-Site/en\_US/GlobalPay-ThreeDSSecureChallenge"},"action":{"id":"ACT\_QwwdivLG9uLezbJnzQSkcdv1D10w5U","type":"INITIATE","time\_created":"2022-06-16T14:22:53.094Z","result\_code":"SUCCESS","app\_id":"6N2pxX3SVk3eqy49kARXhPUnbubneuhr","app\_name":"sample\_app\_CERT"}}

**5. Result**

URL: <https://apis.sandbox.globalpay.com/ucp/authentications/AUT_57f7abd6-ee0a-45da-ac53-71352f1ea758/result>

**Request**:

{"auth\_id":"AUT\_57f7abd6-ee0a-45da-ac53-71352f1ea758","token":"tQZ1YadUkrn2pHkKsm2OW6WkdoOe"}

**Response**:

{"id":"AUT\_57f7abd6-ee0a-45da-ac53-71352f1ea758","time\_created":"2022-06-16T14:23:12.224","status":"SUCCESS\_AUTHENTICATED","channel":"CNP","amount":"15997","currency":"USD","country":"US","source":"BROWSER","three\_ds":{"acs\_trans\_ref":"ced86380-ed7f-41ec-8c10-436a798d313b","ds\_trans\_ref":"20e337d2-1a0a-4e9f-acce-13eed714aa34","server\_trans\_ref":"57f7abd6-ee0a-45da-ac53-71352f1ea758","acs\_reference\_number":"","liability\_shift":"YES","authentication\_type":"OUT\_OF\_BAND\_CHALLENGE","authentication\_value":"AAkBBzlkEQAAAD59hAFndAoPFww=","eci":"05","status":"AUTHENTICATION\_SUCCESSFUL","status\_reason":"","redirect\_url":"https://apis.sandbox.globalpay.","acs\_challenge\_request\_url":"https://acs2p.test.gpe.cz/tds/challenge/brw/ced86380-ed7f-41ec-8c10-436a798d313b","authentication\_source":"BROWSER","challenge\_value":"ewogICJtZXNzYWdlVHlwZSIgOiAiQ1JlcSIsCiAgIm1lc3NhZ2VWZXJzaW9uIiA6ICIyLjEuMCIsCiAgInRocmVlRFNTZXJ2ZXJUcmFuc0lEIiA6ICI1N2Y3YWJkNi1lZTBhLTQ1ZGEtYWM1My03MTM1MmYxZWE3NTgiLAogICJhY3NUcmFuc0lEIiA6ICJjZWQ4NjM4MC1lZDdmLTQxZWMtOGMxMC00MzZhNzk4ZDMxM2IiLAogICJjaGFsbGVuZ2VXaW5kb3dTaXplIiA6ICIwNCIKfQ","message\_category":"PAYMENT\_AUTHENTICATION","message\_version":"2.1.0","challenge\_status":"NOT\_MANDATED","authentication\_request\_type":"","acs\_decoupled\_response\_indicator":"","whitelist\_status":"","message\_extension":[]},"action":{"id":"ACT\_mvVTLvyRQuvwLkaMgiQ6xVAKq7MYiL","type":"GET\_CHALLENGE\_RESULT","time\_created":"2022-06-16T14:23:12.224","result\_code":"SUCCESS","app\_id":"6N2pxX3SVk3eqy49kARXhPUnbubneuhr","app\_name":"sample\_app\_CERT"}}

**6. Transaction**

https://apis.sandbox.globalpay.com/ucp/transactions

**Request**:

{"account\_name":"transaction\_processing","channel":"CNP","capture\_mode":"AUTO","type":"SALE","amount":"15997","currency":"USD","reference":"00002616","country":"US","payment\_method":{"entry\_mode":"ECOM","id":"PMT\_039a07d3-d0ad-4c66-af3f-17be287569a0","authentication":{"id":"AUT\_57f7abd6-ee0a-45da-ac53-71352f1ea758"}},"token":"tQZ1YadUkrn2pHkKsm2OW6WkdoOe"}

**Response**:

{"id":"TRN\_q0gq9uobdpD9gzbiKGhoXlsQAFslUu\_00002616","time\_created":"2022-06-16T14:25:02.306Z","type":"SALE","status":"CAPTURED","channel":"CNP","capture\_mode":"AUTO","amount":"15997","currency":"USD","country":"US","merchant\_id":"MER\_7e3e2c7df34f42819b3edee31022ee3f","merchant\_name":"Sandbox\_merchant\_3","account\_id":"TRA\_c9967ad7d8ec4b46b6dd44a61cde9a91","account\_name":"transaction\_processing","reference":"00002616","payment\_method":{"result":"00","message":"[ test system ] Authorised","entry\_mode":"ECOM","authentication":{"cavv\_result":""},"card":{"funding":"CREDIT","brand":"VISA","masked\_number\_last4":"XXXXXXXXXXXX8884","authcode":"12345","brand\_reference":"2qaJwgYCLt5XuSVL","brand\_time\_created":"","tag\_response":"","cvv\_result":"NOT\_CHECKED","avs\_address\_result":"NOT\_CHECKED","avs\_postal\_code\_result":"NOT\_CHECKED","avs\_action":"","provider":{"result":"00","cvv\_result":"U","avs\_address\_result":"U","avs\_postal\_code\_result":"U"}}},"batch\_id":"BAT\_1105046","action":{"id":"ACT\_q0gq9uobdpD9gzbiKGhoXlsQAFslUu","type":"AUTHORIZE","time\_created":"2022-06-16T14:25:02.306Z","result\_code":"SUCCESS","app\_id":"6N2pxX3SVk3eqy49kARXhPUnbubneuhr","app\_name":"sample\_app\_CERT"}}

### 

### 2. Google Pay

**Prerequisite:**

Before proceeding to Business Manager Configuration, a Merchant Account needs to be created with Google. Please follow the [link](Amit_Global%20Payments%20Storefront%20Reference%20Architecture_v1.0.docx) to create the merchant account with google.

Google Pay is enabled by default in Test environment. To enable or disable Google Pay payment option, please follow below steps to perform.

1. In the Business Manager, go to Merchant tools > Ordering > Payment Methods.
2. Select the ‘GP\_DW\_GOOGLE\_PAY’ id and enable or disable
3. In the Business Manager, go to Merchant tools > Ordering > Payment Methods.
4. In the ID column, select *GP\_DW\_GOOGLE\_PAY* and check that Payment Processor is set to GOOGLE\_PAY

Graphical user interface, application

Description automatically generated

Once you click on ‘Buy with G Pay’, then user navigates to next screen to continue with the payment method selection,

Graphical user interface, text, application, Teams

Description automatically generated

Note - In lower Environment, after choosing any of the above listed test cards and click on ‘continue’ button, ‘Place Order’ button gets enabled on confirmation page.

**Sequence of API calls for Google Pay:**

**1. Transactions**

URL: https://apis.sandbox.globalpay.com/ucp/transactions

**Request**:

{"account\_name":"transaction\_processing","channel":"CNP","capture\_mode":"AUTO","type":"SALE","amount":3000,"currency":"USD","reference":"00002618","country":"US","payment\_method":{"entry\_mode":"ECOM","name":"venkat rao","digital\_wallet":{"provider":"PAY\_BY\_GOOGLE","payment\_token":{"signature":"MEYCIQCTZBLY03vv07u2QWOBx5zU7VDYFb3LXFat1pRxC+AauQIhAJ6COccsh0FJ+I/eq3sB5q7E9/eqpV/qVcWietEKJ/1i","protocolVersion":"ECv1","signedMessage":"{\"encryptedMessage\":\"UfiWtkYs3vws6LPqOrS1r2UyTsMp3kVTq4LHMTQISOIqfHd/BleeQTivTbPKWp9U/FQGyqT5x8eU4Yf77ADQujOnSwHVQC7Db0x+eqtLHC85tqAP8rx2eywMvf+3xJVMghoYQUZ7jLrBEKxltQCLZ4AsP1j8Neq/bDDxz0Pdv5T3SqUbufUKEqvjQHIDPx9vtL5vvchs6koicyudXql2yqnz71CBWnyEDbeMqbiAZGQlH39x4NVCE0RYpG9EEBPRDdrbeLD1S3Kl80wkMnLpjQmSTVi+yYHW7n9eqe1BcBIh/CURkSViLxZSjT+ryAmcYkSkFrydgaHz7JlUO19gbYs8unWVRknle8vfIQt7IRuTkIsw1B+HwGdAkkH92jvFdr7f2Dcm3HfSWOpngyscQVNpmyTqvNPCQJFqDHCTX4rq+EytVq0XFmHNYAY9nAXgjzWH\",\"ephemeralPublicKey\":\"BECHW3fG/MAMX0A1pm7S26KCqIuIPP856yLg8wg0x8sn9J/FObGoLoR1CPyWTiVyf2SYBHB2n3J8fSMRUUL/FpA\\u003d\",\"tag\":\"BfRUmUVCT/J5b5fAkeDQxbDJFNqaaHo41PVpTf+56XQ\\u003d\"}"}}},"token":"tQZ1YadUkrn2pHkKsm2OW6WkdoOe"}

**Response**:

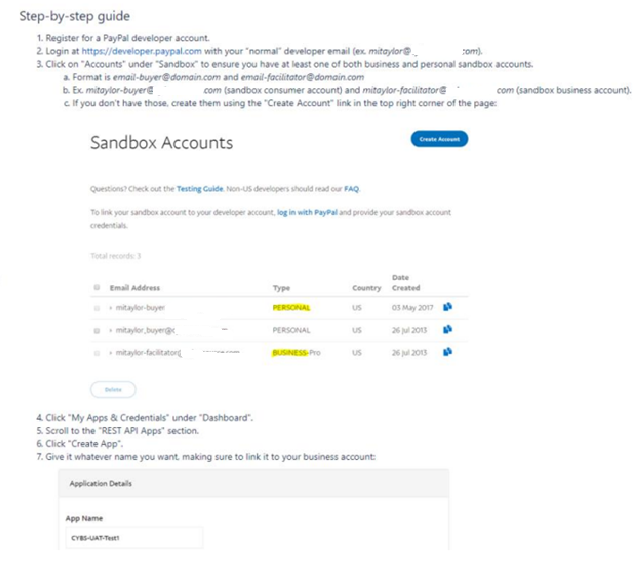
{"id":"TRN\_ysWmWpAW2gTKq4RRcLISvmYzmiyLMK\_00002618","time\_created":"2022-06-16T14:33:44.643Z","type":"SALE","status":"CAPTURED","channel":"CNP","entry\_mode":"ECOM","capture\_mode":"AUTO","amount":"3000","currency":"USD","country":"US","merchant\_id":"MER\_7e3e2c7df34f42819b3edee31022ee3f","merchant\_name":"Sandbox\_merchant\_3","account\_id":"TRA\_c9967ad7d8ec4b46b6dd44a61cde9a91","account\_name":"transaction\_processing","reference":"00002618","payment\_method":{"first\_name":"","last\_name":"","result":"00","message":"[ test system ] Authorised","digital\_wallet":{"provider":"PAY\_BY\_GOOGLE","token\_format":"","eci":"","brand":"","authcode":"12345","brand\_reference":"5q45D0geEXCYdbJJ"}},"batch\_id":"BAT\_1105046","action":{"id":"ACT\_ysWmWpAW2gTKq4RRcLISvmYzmiyLMK","type":"AUTHORIZE","time\_created":"2022-06-16T14:33:44.643Z","result\_code":"SUCCESS","app\_id":"6N2pxX3SVk3eqy49kARXhPUnbubneuhr","app\_name":"sample\_app\_CERT"}}

### 3. PayPal

**Prerequisite:**

Prior to development phase, there are set of configurations which development team need to take care. These configurations include:

1. PayPal developer account.
2. PayPal sandbox account Screenshot of the detailed set of configurations for #1 & #2.



1. Linking developer and sandbox account. On creating a PayPal developer account, get in touch with the GlobalPay team, share the developer account details and get the developers’ details configured on GlobalPay (BackOffice Configuration tool). Share the following keys with GlobalPay:

In the Business Manager, go to Merchant tools > Ordering > Payment Methods.

1. In the Business Manager, go to Merchant tools > Ordering > Payment Methods.
2. Select the ‘GP\_DW\_PAYPAL’ id and enable or disable
3. In the Business Manager, go to Merchant tools > Ordering > Payment Methods.
4. In the ID column, select *GP\_DW\_PAYPAL* and check that Payment Processor is set to GLOBALPAY\_PAY .
5. In the Business Manager, go to Merchant tools > Site Preferences > Custom Preference > Globalpay
6. You can enable and the payment method through site preference.

Following screen is for enabling the payment options on checkout page.



If the PayPal is enabled on payment methods and custom site preferences, then on checkout page below PayPal button is enabled as below screen.

Graphical user interface

Description automatically generated with medium confidence

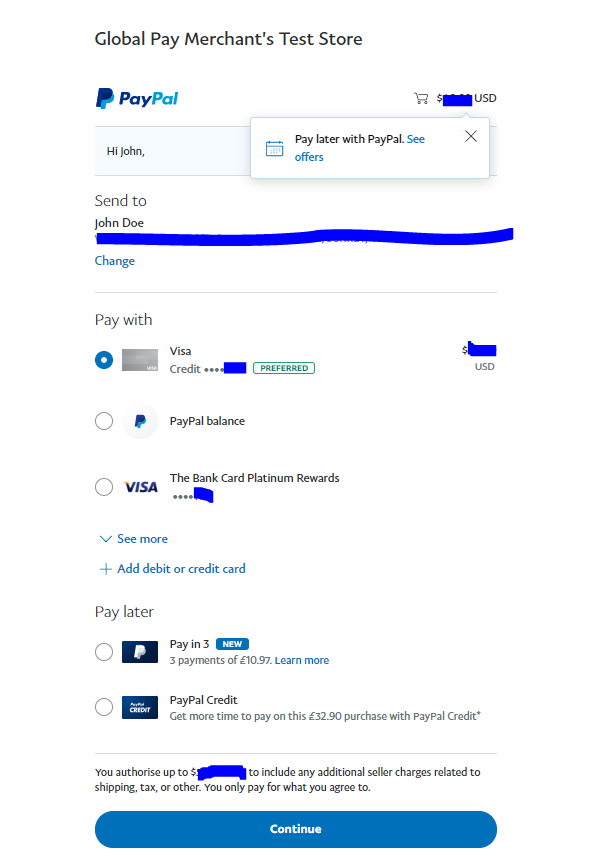
Once you click on ‘Buy with PayPal’ button, then below screen will appear, after entering your PayPal credentials and clicks on ‘Log In’ button.

Graphical user interface, application

Description automatically generated

User has leverage to add the cards of his/her choice or else user can continue the payment with existing card. Order will be created and Payment status changes to ‘Paid’ once done with the payment.

By clicking on Cancel button on PayPal screen will redirect user to Merchant portal for continuing the shopping.



**Sequence of API calls for Pay Pal:**

1. **Transactions**

https://apis.sandbox.globalpay.com/ucp/transactions

**Request**:

{"account\_name":"transaction\_processing","channel":"CNP","capture\_mode":"AUTO","type":"SALE","amount":"7599","currency":"USD","reference":"00002617","country":"US","payment\_method":{"entry\_mode":"paypal","apm":{"provider":"paypal"}},"notifications":{"return\_url":"https://zyem-006.sandbox.us01.dx.commercecloud.salesforce.com/on/demandware.store/Sites-RefArch-Site/en\_US/GPPayPal-PayPalReturn","status\_url":"https://zyem-006.sandbox.us01.dx.commercecloud.salesforce.com/on/demandware.store/Sites-RefArch-Site/en\_US/GPPayPal-PayPalStatus","cancel\_url":"https://zyem-006.sandbox.us01.dx.commercecloud.salesforce.com/on/demandware.store/Sites-RefArch-Site/en\_US/GPPayPal-PayPalCancel"},"token":"tQZ1YadUkrn2pHkKsm2OW6WkdoOe"}

**Response**:

{"id":"TRN\_IT51SclCDA2Mi360e5I4Kvlf6zsbmP\_00002617","time\_created":"2022-06-16T14:28:21.208Z","type":"SALE","status":"INITIATED","channel":"CNP","capture\_mode":"AUTO","amount":"7599","currency":"USD","country":"US","merchant\_id":"MER\_7e3e2c7df34f42819b3edee31022ee3f","merchant\_name":"Sandbox\_merchant\_3","account\_id":"TRA\_c9967ad7d8ec4b46b6dd44a61cde9a91","account\_name":"transaction\_processing","reference":"00002617","payment\_method":{"result":"00","message":"SUCCESS","entry\_mode":"PAYPAL","redirect\_url":"https://apis.sandbox.globalpay.com/ucp/postback/transactions/TRN\_IT51SclCDA2Mi360e5I4Kvlf6zsbmP\_","apm":{"provider\_redirect\_url":"https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=\_express-checkout&token=EC-20U74772571882923","provider":"paypal","session\_token":"EC-20U74772571882923","ack":"SUCCESS","correlation\_reference":"46920ff3ca64d","version\_reference":"98.0","build\_reference":"56929629"}},"batch\_id":"","action":{"id":"ACT\_IT51SclCDA2Mi360e5I4Kvlf6zsbmP","type":"INITIATE","time\_created":"2022-06-16T14:28:21.208Z","result\_code":"SUCCESS","app\_id":"6N2pxX3SVk3eqy49kARXhPUnbubneuhr","app\_name":"sample\_app\_CERT"}}

**2. Confirmation**

URL: https://apis.sandbox.globalpay.com/ucp/transactions/TRN\_IT51SclCDA2Mi360e5I4Kvlf6zsbmP\_00002617/confirmation

**Request**:

{"transaction\_id":"TRN\_IT51SclCDA2Mi360e5I4Kvlf6zsbmP\_00002617","token":"tQZ1YadUkrn2pHkKsm2OW6WkdoOe"}

**Response**:

{"id":"TRN\_IT51SclCDA2Mi360e5I4Kvlf6zsbmP\_00002617","time\_created":"2022-06-16T14:30:14.443Z","type":"SALE","status":"CAPTURED","amount":"7599","currency":"USD","country":"US","merchant\_id":"MER\_7e3e2c7df34f42819b3edee31022ee3f","merchant\_name":"Sandbox\_merchant\_3","account\_id":"TRA\_c9967ad7d8ec4b46b6dd44a61cde9a91","account\_name":"transaction\_processing","reference":"00002617","payment\_method":{"result":"00","message":"SUCCESS","entry\_mode":"PAYPAL","apm":{"provider":"paypal","session\_token":"EC-20U74772571882923","seller\_reference":"testmerchant@globalpay.com","time\_created\_reference":"2022-06-16T14:30:14Z","transaction\_reference":"1P191098V8195591Y","ack":"SUCCESS","version\_reference":"98.0","build\_reference":"56929629","correlation\_reference":"4a7037af4d6a","secure\_account\_reference":"CUEXTCPF4MNQY","reason\_code":"none","pending\_reason":"none","gross\_amount":"7599","payment\_time\_reference":"2022-06-16T14:30:13Z","payment\_type":"INSTANT","payment\_status":"COMPLETED","type":"express-checkout","protection\_eligibilty":"ELIGIBLE"}},"action":{"id":"ACT\_qNAEqIWH78EtvNvNK7r9W9gLFoHILP","type":"CONFIRM","time\_created":"2022-06-16T14:30:14.443Z","result\_code":"SUCCESS","app\_id":"6N2pxX3SVk3eqy49kARXhPUnbubneuhr","app\_name":"sample\_app\_CERT"}}

### 4. Apple Pay

**Configuration:**

1. In the Business Manager, go to Merchant tools > Ordering > Payment Methods.
2. In the ID column, select ***GP\_DW\_APPLE\_PAY***and check that Payment Processor is set to Apple\_PAY

**1. Description**

This document details the configurations required for Apple Pay through SFCC Business Manager and Apple Merchant Account. It will also brief

The setup required on Apple devices to use Apple Pay

**2. Supported Devices**

iPhone: iPhone SE, iPhone 6, iPhone 6Plus & higher with iOS 10 or later

iPad: iPad Pro, iPad Air 2, iPad mini 4, iPad mini 3 & higher with iOS 10 or later

Macbook: macOS v. 10.12. The user must have an iPhone or Apple Watch that can authorize the payment

Note: To troubleshoot issues from other regions, change the region setting on the device to a region where Apple Pay is supported

**3. Apple Pay Requirements**

You must have an Apple Developer Account (enrolled for development)

All paged that incorporate Apple Pay must be served over HTTPS

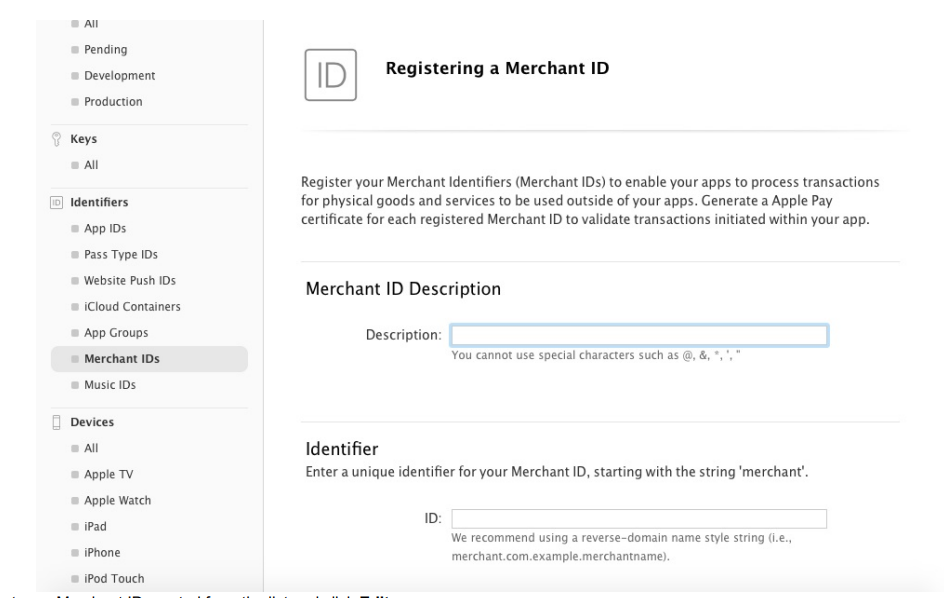
Your website must comply with the Apple Pay guidelines. For more information see Apple Pay on the Web Acceptable Use Guideline

Setting up Sandbox Domain with Apple Pay Merchant

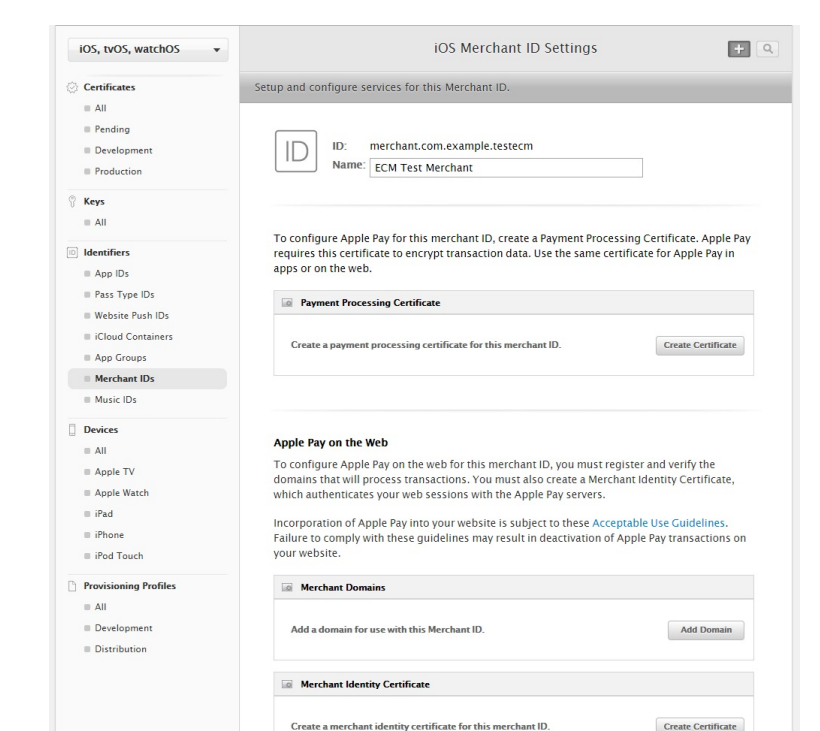
Below are the steps mentioned to configure the SFCC Sandbox and Apple Pay account along with the Payment processor.

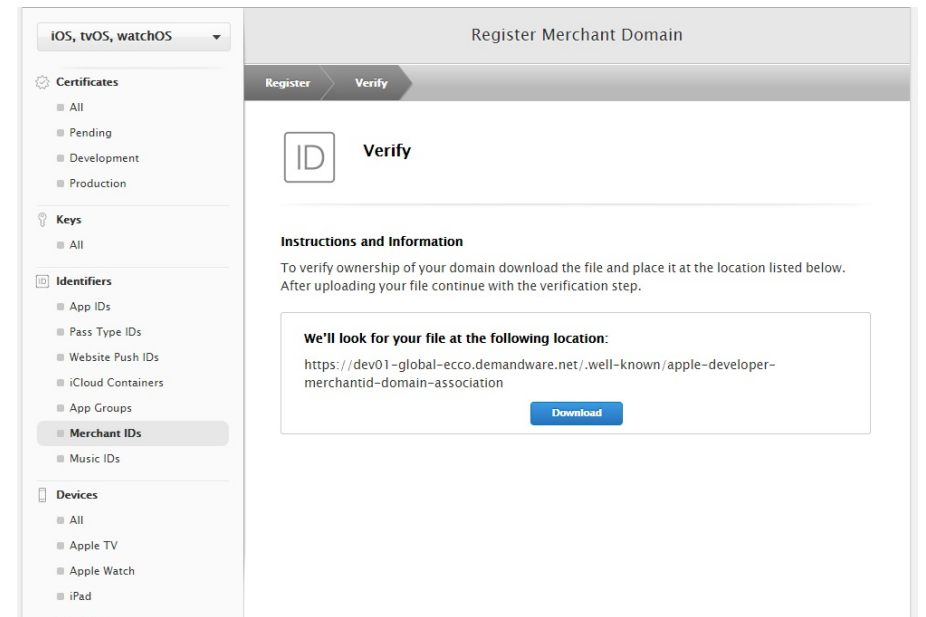
Setting up Apple Pay Merchant

1. Login to Apple Merchant Center with registered Developer Account
2. In Member Center, select Certificates, Identifiers & Profiles
3. Under Identifiers, select Merchant IDs
4. Click on "+" sign on the Right Top corner to add new Merchant ID
5. Enter Merchant ID Description and Identifier on Registering a Merchant ID Page
   1. Don't use special characters like @, &, \*, ', "
   2. Click on Next
   3. Click Register on subsequent page



1. Select your Merchant ID created from the list and click Edit
2. In the Apple Pay on the Web Section, click on Add Domain
3. Enter your fully qualified domain name and click on Continue
4. This creates a file and makes it available to download and host it at the provided location i.e. **Error! Hyperlink reference not valid.**





**4. Adding Verification File on Domain**

1. Adding verification file in Catalog

1. In SFCC Business Manager, go to Merchant Tools > Products & Catalogs > Catalogs
2. Select storefront catalog
3. Create Category structure .well-known > apple-developer-merchantid-domain-association under root category
4. Use xml file similar to the one attached here:
5. Use a Category level custom attribute to add data present in the domain verification file downloaded from Apple Merchant Account mentioned in above steps
6. Sample rendering template attached here: apple-domain-verification.isml
7. Use a rendering template to include content from the above mentioned attribute on category landing page for .well-known/apple-developer-merchantid-domain-association

2. Creating Alias to register domain

1. Go to Merchant Tools > Site URLs > Aliases
2. Add alias as below in SFCC Business Manager

{

"\_\_version":"1",

"settings":{

"http-host":"development-pentland-ecommera.demandware.net",

"https-host":"development-pentland-ecommera.demandware.net"

},

"development-pentland-ecommera.demandware.net":[

{

"pipeline" : "Home-Show"

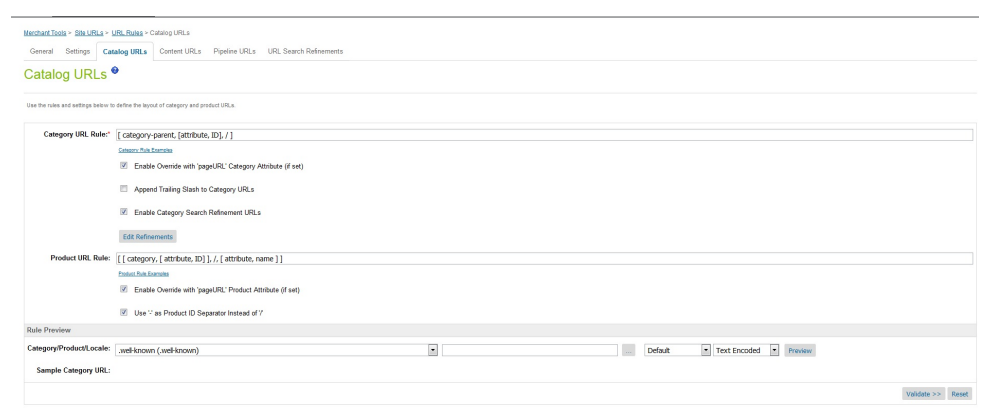
}

]

}

3. Updating URL Rules

* 1. Go to Merchant Tools > Site URLs > URL Rules > Catalog URLs
  2. Add rules as in below screenshot:



3. Verifying domain

* 1. Hit site url as https://<domain>/.well-known/apple-developer-merchantid-domain-association and check if content from domain verification file are getting loaded with a 200 OK status
  2. To check the network response, use one of the below:
     1. Use Browser extension for Link Redirect Trace:http://lrt.li/rtlogoappchrome
     2. Check online at http://www.redirect-checker.org/index.php
  3. Once this is done, go back to Apple Merchant Account and click on Verify

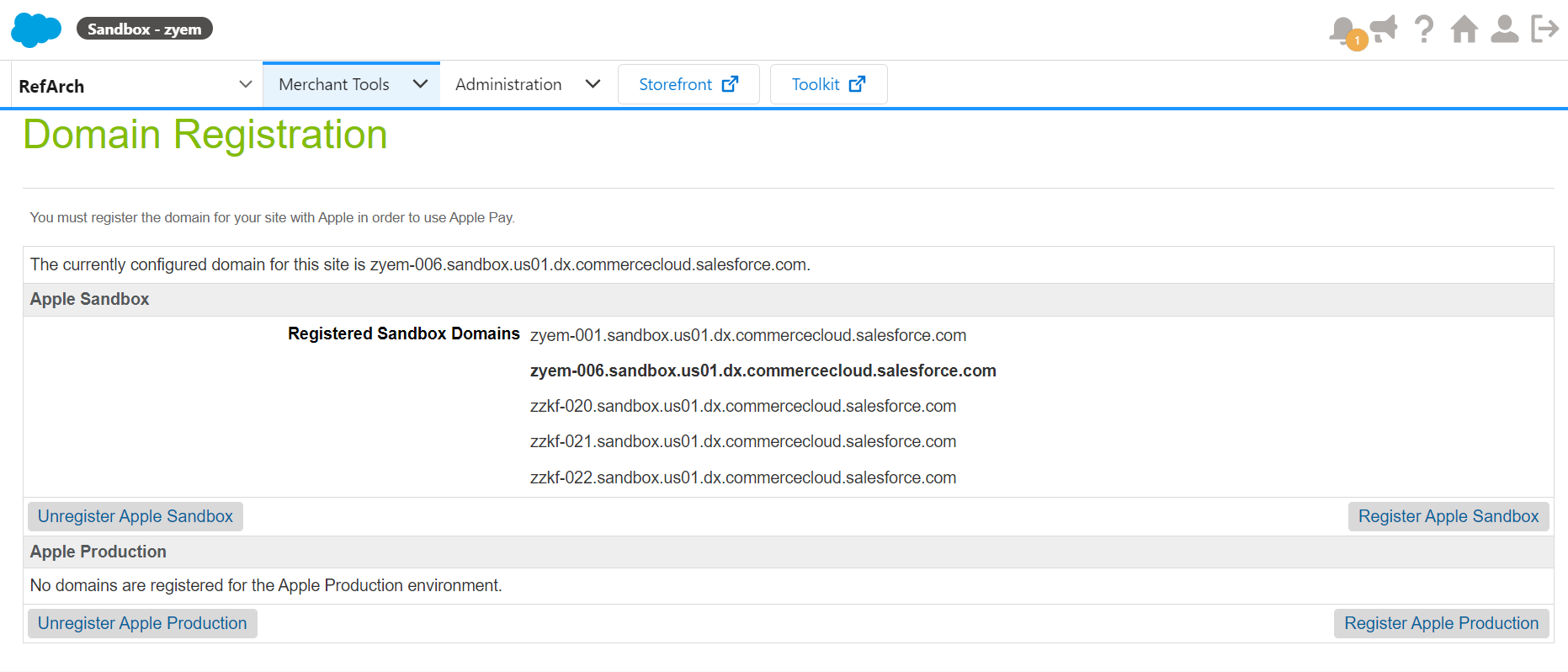
4. Domain Registration in SFCC Business Manager

1. Go to “Merchant Tools > Site Preferences > Apple Pay

2. Under Domain Registration section

a. Click on Register Apple Sandbox under Apple Sandbox section for registering SFCC to Apple Sandbox account 16 Visa Confidential

b. Click on Register Apple Production under Apple Production section for registering SFCC to Apple Production account.



1. Configure Apple Pay in SFCC Business Manager Business Manager Configuration

1. Go to “Merchant Tools > Site Preferences > Apple pay

2. Check “Apple Pay Enabled?”

3. Fill in the “Onboarding” form: • Ensure “Apple Merchant ID” and “Apple Merchant Name” match settings in your Apple account

4. Fill in the “Storefront Injection” form:

• Selects where Apple Pay buttons should display on your site.

5. Fill in “Payment Integration” form:

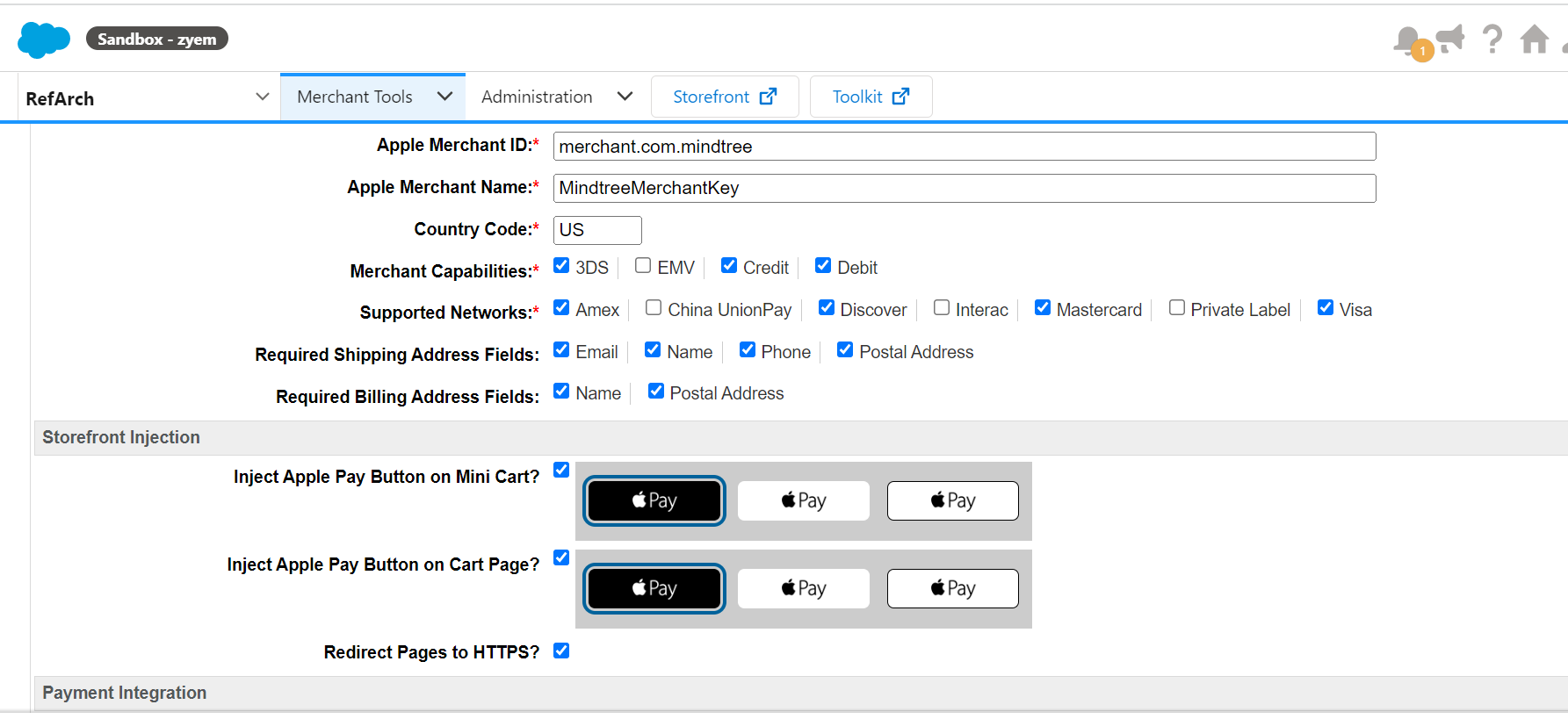
• Fill in Payment Provider URL

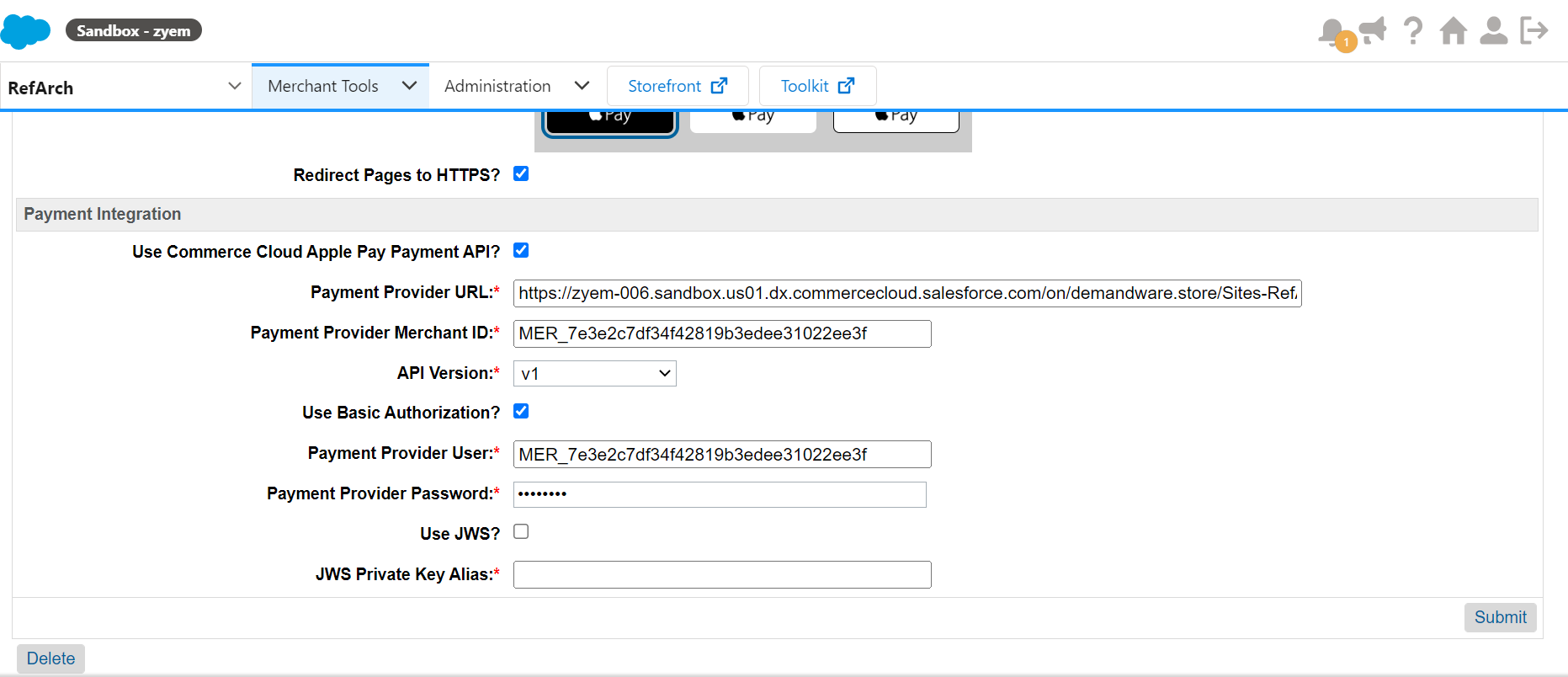
• Fill in Payment Provider User

• Fill in Payment Provider Password (some random Value)

• Ensure “Use Basic Authorization” is checked

6. Click "Submit"





1. Apple Pay on Demandware Storefront

To test ApplePay on Demandware site, following files need to be updated:

* + 1. Script – applepay.js

Update the file with below changes

var Status = require('dw/system/Status');

var server = require('server');

var Transaction = require('dw/system/Transaction');

var ApplePayHookResult = require('dw/extensions/applepay/ApplePayHookResult');

var paymentMethodID = 'DW\_APPLE\_PAY';

/\*\*

 \* @function getRequest hook is called whenever there is a new request on the site

 \*/

 exports.getRequest = function (basket, response) {

    session.custom.applepaysession = 'yes';   // eslint-disable-line

    var status = new Status(Status.OK   );

    var result = new ApplePayHookResult(status, null);

    return result;

};

exports.authorizeOrderPayment = function (order, responseData) {

    var status = Status.ERROR;

    var authResponseStatus;

    var paymentMethod = require('dw/order/PaymentMgr').getPaymentMethod(paymentMethodID);

    setBillingAddress(responseData.payment.billingContact);

    setShippingAddress(responseData.payment.shippingContact);

    // eslint-disable-next-line

    Transaction.wrap(function () {

        //  lineItemCtnr.paymentInstrument field is deprecated.  Get default payment method.

        var paymentInstrument = null;

        // eslint-disable-next-line

        if (!empty(order.getPaymentInstruments())) {

            paymentInstrument = order.getPaymentInstruments()[0];

            paymentInstrument.paymentTransaction.paymentProcessor = paymentMethod.getPaymentProcessor();

        } else {

            return new Status(status);

        }

        paymentInstrument.paymentTransaction.paymentProcessor = paymentMethod.getPaymentProcessor();

    });

   // service logic import

  var globalpayAuthorization = require('../globalpayapplepay');

  var token = responseData.payment.token.paymentData;

  authResponseStatus= globalpayAuthorization.Authorize( order ,token);

    if (authResponseStatus) {

        status = Status.OK;

    }

    return new Status(status);

};

2. hooks.json Add hook for applepay at the end of file present at /cartridge/script

  {

            "name": "dw.extensions.applepay.getRequest",

            "script": "./cartridge/scripts/hooks/payment/processor/applepay"

        },

        {

            "name": "dw.extensions.applepay.paymentAuthorized.authorizeOrderPayment",

            "script": "./cartridge/scripts/hooks/payment/processor/applepay"

        }

3. Controller – GlobalPay-Authorization.js

/\*\*

 \* GlobalPay-Authorization : The GlobalPay-Authorization endpoint invokes authorization call from applepay

 \* @name Base/GlobalPay-Authorization

 \* @function

 \* @memberof GlobalPay

 \*/

 server.post('Authorization', function (req, res) {

  //Returning Success in the basic Auth method

  return { success: true }

});

1. Actual Authorization Using GlobalPay API:

Update the below file :

/\*\*

 \* Authorizes a payment using a apple pay.

 \* @param {number} orderNumber - The current order'\

 \* s number

 \* @param {dw.order.PaymentInstrument} paymentInstrument -  The payment instrument to authorize

 \* @param {dw.order.PaymentProcessor} paymentProcessor -  The payment processor of the current

 \*      payment method

 \* @return {Object} returns an error object

 \*/

function Authorize(order, paymentdata) {

    var globalpayconstants = require('\*/cartridge/scripts/constants/globalpayconstants');

    var globalPayPreferences = require('\*/cartridge/scripts/helpers/globalPayPreferences');

    var globalPayHelper = require('\*/cartridge/scripts/helpers/globalPayHelper');

    var URLUtils = require('dw/web/URLUtils');

    var BasketMgr = require('dw/order/BasketMgr');

    var currentBasket = BasketMgr.getCurrentBasket();

    var preferences = globalPayPreferences.getPreferences();

    var captureMode = preferences.captureMode;

    var HookManager = require('dw/system/HookMgr');

    var Locale = require('dw/util/Locale');

    var serverErrors = [];

    var applePayData = {

        account\_name: globalpayconstants.applePay.account\_name,

        channel: globalpayconstants.applePay.channel,

        type: globalpayconstants.applePay.type,

        capture\_mode: captureMode.value,

        amount: (order.totalGrossPrice) \* 100,

        currency: order.currencyCode,

        reference: order.orderNo,

        country: 'US',

        payment\_method: {

            name: order.customerName.replace(' ', ''),

            entry\_mode: globalpayconstants.applePay.entryMode,

            digital\_wallet: {

                provider: globalpayconstants.applePay.provider,

                //need to be removed once we get the solution for payment token

                payment\_token: {

                    version: paymentdata.version,

                    data: paymentdata.data,

                    header: {

                        ephemeralPublicKey: paymentdata.header.ephemeralPublicKey,

                        transactionId: paymentdata.header.transactionId,

                        publicKeyHash: paymentdata.header.publicKeyHash

                    }

                }

            }

        }

    }

    var globalPayHelper = require('\*/cartridge/scripts/helpers/globalPayHelper');

    var PaymentInstrumentUtils = require('\*/cartridge/scripts/utils/PaymentInstrumentUtils');

    var applePayresp = globalPayHelper.applePay(applePayData);

    var orderUpdateResult = PaymentInstrumentUtils.ApplePaymentOrderUpdate(order, applePayresp);

    if (!orderUpdateResult) {

        var error = true;

             serverErrors.push(

                 Resource.msg('error.technical', 'checkout', null)

             );

    }

    return orderUpdateResult;

}

exports.Authorize = Authorize;

* 1. Now Applepay Can be tested in the DW storefront.

# Order Management

### 1. Refund Procedure:

1. In the Business Manager, go to Merchant tools > Site Preferences > Custom Preference > Globalpay
2. You can select captureMode as either ‘Auto’ or ‘Later’.
3. If the captureMode as ‘Auto’, if you can place the order through storefront,  
   then the order payment status as ‘paid’ otherwise payment status will be ‘not paid’.

If you want to refund the amount, then you can refund the with following url. Once refund completes the order status will be updated as ‘cancelled’. You can refund the amount only for the order, which has the payment status as paid and order status as ‘open’.  
  
**Refund**

curl --location --request POST 'https://dwhost/on/demandware.store/Sites-RefArch-Site/en\_US/OrderTransactions-RefundTransaction?orderID=00000513' \

--header 'clientid: 4230946d-3f35-4e91-8303-a51720ba7a5b' \

--header 'Authorization: Bearer WfAhe1msHSuGHaSAG3OJGupZKGtC' \

--header 'Cookie: dwanonymous\_1f866598df3d23bd96d7fbc0ff91985f=abxxA87LbXDJZ06kg840282GHi; dwanonymous\_b5d8a8e2b741ecb26e2f400a02c69ec5=ac2HKz9TnA5RBARSORIh4EaeRq'

**Cancel**

curl --location --request POST 'https://dwhost/on/demandware.store/Sites-RefArch-Site/en\_US/OrderTransactions-CancelTransaction?orderID=00000516' \

--header 'clientid: 4230946d-3f35-4e91-8303-a51720ba7a5b' \

--header 'Authorization: Bearer Vt2DOw2mvl3DqGTbgGgy4EFAq9qG' \

--header 'Cookie: dwanonymous\_1f866598df3d23bd96d7fbc0ff91985f=abxxA87LbXDJZ06kg840282GHi; dwanonymous\_b5d8a8e2b741ecb26e2f400a02c69ec5=ac2HKz9TnA5RBARSORIh4EaeRq'

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

### 2. Capture Procedure:

1. In the Business Manager, go to Merchant tools > Site Preferences > Custom Preference > Globalpay
2. You can select captureMode as either ‘Auto’ or ‘Later’.
3. If the captureMode as ‘Later, if you can place the order through storefront,  
   then order payment status as ‘not paid’.
4. Below is the url for capturing amount from 3rd party system.

Capture Url:

https://yoursandboxurl/on/demandware.store/Sites-RefArch-Site/en\_US/OrderTransactions-CaptureTransaction?orderID=<orderId>

**Capture**

curl --location --request POST 'https://dwhost/on/demandware.store/Sites-RefArch-Site/en\_US/OrderTransactions-CaptureTransaction?orderID=00000513' \

--header 'clientid: 4230946d-3f35-4e91-8303-a51720ba7a5b' \

--header 'Authorization: Bearer WfAhe1msHSuGHaSAG3OJGupZKGtC' \

--header 'Cookie: dwanonymous\_1f866598df3d23bd96d7fbc0ff91985f=abxxA87LbXDJZ06kg840282GHi; dwanonymous\_b5d8a8e2b741ecb26e2f400a02c69ec5=ac2HKz9TnA5RBARSORIh4EaeRq'

# *Global payments document links*

1. Global Payments Developer Guide - <https://developer.globalpay.com/>

2. Postman Collection - <https://documenter.getpostman.com/view/12989945/UVJeGwg4>

3. 3DS Implementation Guide - <https://developer.globalpay.com/docs/browser-auth-3DS#hosted-fields>

4. Test Cards Detail - <https://developer.globalpay.com/resources/test-card-numbers>

# *Release History*

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Change** |
| 1.0. | 06/16/2022 | Initial Release |
|  |  |  |
|  |  |  |
|  |  |  |