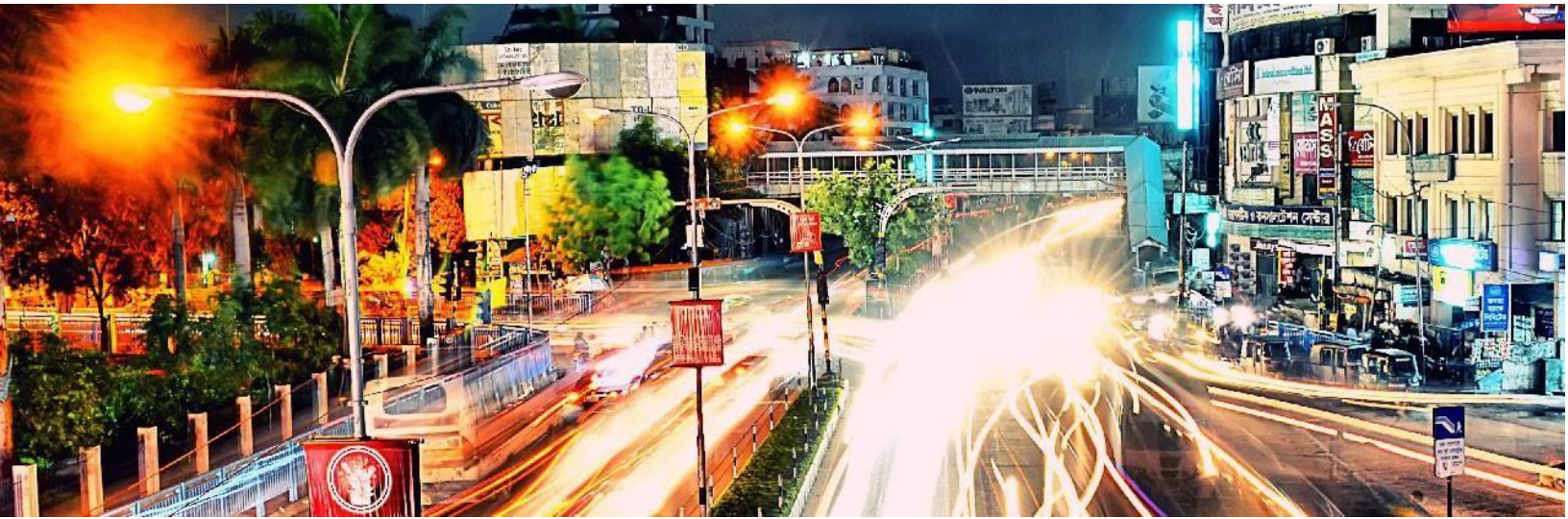


Cybersource®



ISV - Oracle Commerce Cloud Gateway User Installation Guide

October 2023



Contents

1.	Introduction.....	4
1.1.	Cybersource Configuration	5
1.2.	Generating API Keys in Business Center	5
1.3.	Oracle Commerce Cloud Configuration	6
1.3.1.	Payment Gateway Installation Details	6
1.3.2.	SSE (server-extension) installation details	8
1.3.3.	Plugin installation details	9
2.	Version History	10
3.	Configuration Details.....	12
3.1.	General Settings.....	12
3.2.	Fraud Management Settings.....	14
3.2.1.	Enabling Payer Authentication	14
3.2.2.	Enabling Strong Customer Authentication	16
3.2.3.	Enabling Device fingerprint.....	17
3.2.4.	Advanced Fraud Screening with Decision Manager	18
4.	Reporting.....	18
4.1.	Reporting configuration	18
5.	Shipping Region.....	19
6.	Placing order from Storefront	20
6.1.	Placing an order from storefront using Credit Card	20
7.	Oracle Commerce Cloud Storefront Cancel an order	25
8.	Apple Pay Configuration.....	29
8.1.	Create a Merchant ID.....	29
8.2.	Create Payment Processing Certificate.....	31
8.3.	Domain Validation.....	32
8.4.	Create Merchant Identity Certificate.....	34
9.	Network Tokenization	37
10.	Support.....	38
11.	Abbreviations	39

Copyright:

© 2023 Cybersource Corporation. All rights reserved. Cybersource Corporation (including its subsidiaries, “Cybersource”) furnishes this document and the software and/or code described in this document under the applicable agreement between the reader of this document (“You”) and Cybersource (“Agreement”). You may use this document and/or software and/or code only in accordance with the terms of the Agreement, except as expressly set forth in the Agreement, the information contained in this document is subject to change without notice and therefore should not be interpreted in any way as a guarantee or warranty by Cybersource. Cybersource assumes no responsibility or liability for any errors that may appear in this document. The copyrighted software and/or code that accompanies this document is licensed to you for use only in strict accordance with the Agreement. You should read the Agreement carefully before using the software and/or code. Except as permitted by the Agreement, you may not reproduce any part of this document, store this document in a retrieval system or transmit this document in any form or by any means, electronic, mechanical, recording or otherwise without the prior written consent of Cybersource.

Network Capability: By accepting this document, you acknowledge and accept that you are responsible for and assume liability for the functionality, maintenance and availability of your software and network. At all times, it is your responsibility to ensure the accuracy, technical sufficiency and functionality of your software, network, plug-ins, configurations, applications, code, application program interfaces (APIs), software development kits and all other technology (“Your Network”). You are responsible for Your Network’s ability to use and/or access the Cybersource network, any Cybersource API and receive the benefit of Cybersource’s services. You are responsible for all costs, fees, expenses and liabilities associated with Your Network’s ability to access and interface with the Cybersource network and receive the benefit of Cybersource’s services. Cybersource will not be responsible or liable for loss or costs associated with or that results from Your Network’s inability to connect to or process transactions on the Cybersource network.

Release: October 2023

Version: 23.3.0

1. Introduction

This document contains the details of configuring the ISV OCC payment plugin in Oracle Commerce Cloud. The configuration steps are related to Payment Acceptance, Payment Security, Fraud Management, Order Management and Commerce services for Credit Card, Google Pay and Apple Pay Payment Methods.

The purpose of this manual is to guide a user to configure and to use the ISV OCC payment plugin for the Oracle Commerce Cloud platform. The Oracle Commerce Cloud platform includes the following Cybersource payment management capabilities.

Credit Cards (Flex Microform)

- a) Payment Acceptance
 - Authorization
 - Sale (Authorization & Settlement)
- b) Payment Security
 - Tokenization -Create Payment Token for New Payment Methods
 - Tokenization - Make a Payment Using a Stored Token
 - Network Token Updates
- c) Fraud Management
 - Payer Authentication
 - Strong Customer Authentication
 - Decision Manager with Device Fingerprint
 - Advanced Fraud Screening with Decision Manager
- d) Order Management
 - Capture
 - Refund
 - Void (Authorization Reversal)
- e) Commerce service
 - On-demand conversion
 - Daily conversion

Google Pay

- a) Payment Acceptance
 - Authorization
- b) Fraud Management
 - Decision Manager with Device Fingerprint
 - Advanced Fraud Screening with Decision Manager
- c) Order Management
 - Capture
 - Refund
 - Void (Authorization Reversal)
- d) Commerce service
 - On-demand conversion

- Daily conversion

Apple Pay

- a) Payment Acceptance
 - Authorization
- b) Fraud Management
 - Decision Manager with Device Fingerprint
 - Advanced Fraud Screening with Decision Manager
- c) Order Management
 - Capture
 - Refund
 - Void (Authorization Reversal)
- d) Commerce service
 - On-demand conversion
 - Daily conversion

Note:

- Saved Card feature is supported only during checkout
- Services triggered using OMS will not be updated in OCC

1.1. Cybersource Configuration

To use the Cybersource services, the Merchant needs to procure an account from Cybersource. The Merchant will be provided with the Merchant key ID and Shared secret key. This Merchant key ID and Shared secret key should be configured in Oracle Commerce Cloud to enable the integration between Cybersource and Oracle Commerce Cloud.

A Cybersource account can be created from [Cybersource.com](https://cybersource.com). For more information on creating an account, Merchant can contact the Cybersource Customer support.

Key features of the Oracle Commerce Cloud and Cybersource Official Payment Add-on Integration:

- Enable the users to accept and manage payments in Oracle Commerce Cloud.
- Enable hassle-free, completely secure, PCI Compliant, Fraud management enabled - end to end payment transaction for Credit Card, Google Pay and Apple Pay Payment Methods.
- Supports Tokenization which eliminates electronic CHD from being stored in the Merchant environment thereby reduces the scope of Payment Card Industry (PCI) compliance considerations.

1.2. Generating API Keys in Business Center

This section provides the detailed steps to generate API Keys (Merchant key ID and Shared secret key) required to configure in the Oracle Commerce Cloud Back Office.

Oracle Commerce Cloud – ISV Gateway User Installation Guide

Step 1: Go to [Cybersource.com](https://cybersource.com) and then login to business center. Click on “Key Management” in “Payment Configuration” Tab.

Step 2: Click on “Generate Key” button.

Step 3: Select “REST - Shared Secret” and click on “Generate Key”.

Step 4: A Shared secret key will be generated.

Step 5: Go to “Key Management”, note the Key ID displayed. These keys can be used in Oracle Commerce Cloud Back Office Payment Settings for the MID in which these keys are generated.

1.3. Oracle Commerce Cloud Configuration

This section provides the plugin installation steps to enable integration between Oracle Commerce Cloud and Cybersource.

1.3.1. Payment Gateway Installation Details

The steps to install the plugin from Oracle Commerce Cloud Admin are:

1.3.1.1. Create an extension ID

To upload an extension into Commerce Cloud, you must generate an ID for the extension and update the same in `packages/payment-gateway/ext.json` file

To create an extension ID:

- Log into Commerce Cloud.
- Navigate to Settings -> Extensions -> Developer tab.
- Click on Generate ID button.
- Enter a name for the extension and click Save.
- Copy and update the generated extension ID

1.3.1.2. Upload Extension

Before uploading the extension, zip up all the files within your `packages/payment-gateway` directory excluding `settings.json`. This is the file you upload to Commerce Cloud to make the extension available for use.

Step 1: In the “Settings” tab on the left panel.

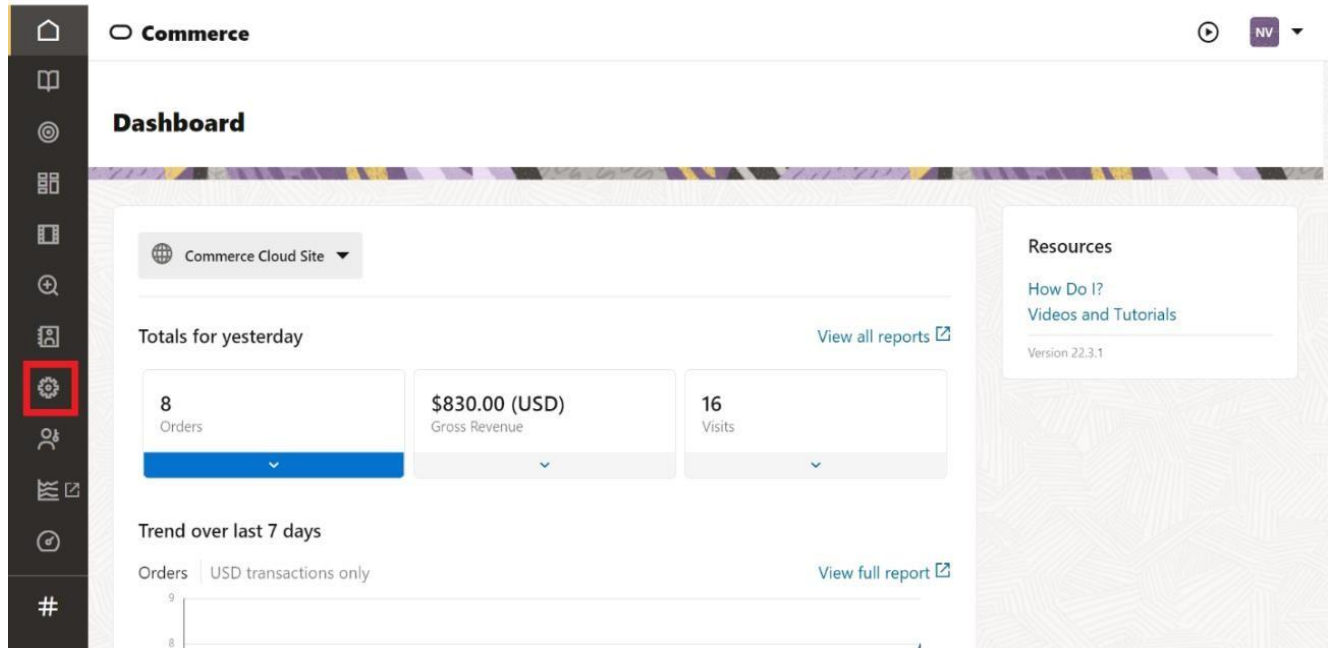


Figure 1: Oracle Commerce Cloud Back Office Dashboard

Step 2: In settings, click on “Extension” button.

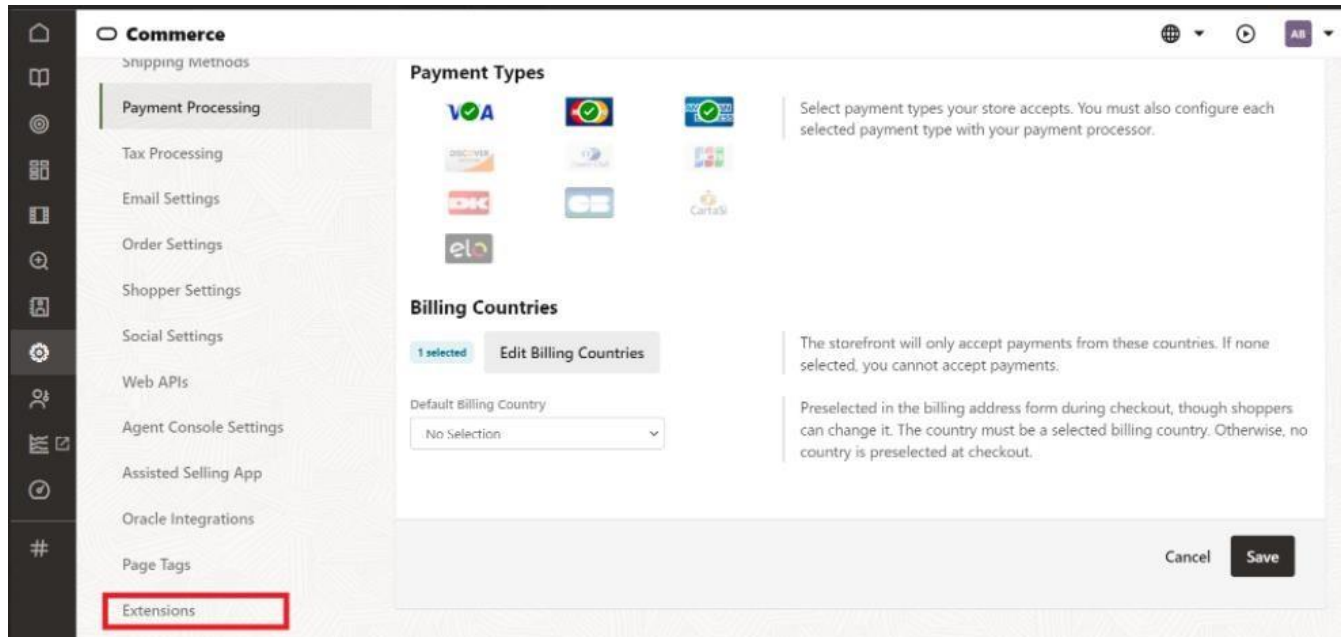


Figure 2: Extension button

Step 3: Click the Upload Extension button and select the extension zip file from your local file system.

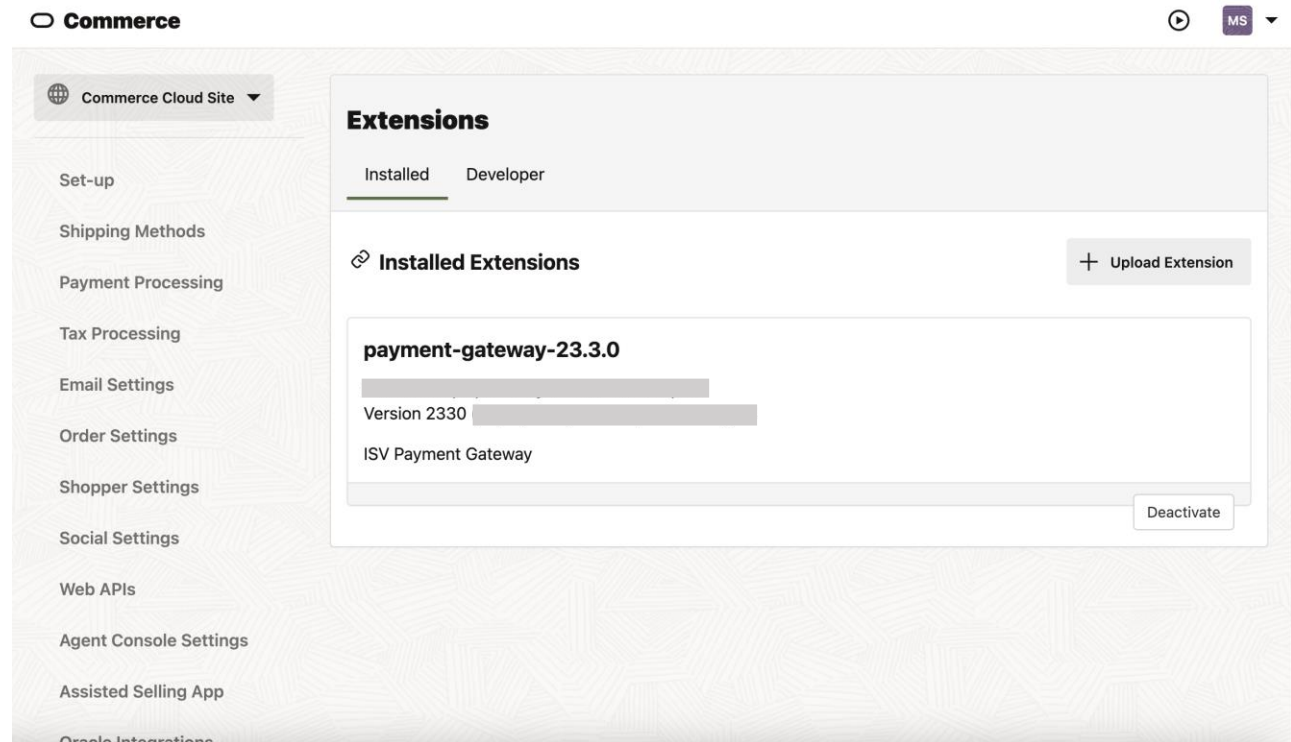


Figure 3: Upload Extension

Once the module is installed, head back to the Oracle Commerce Cloud Admin settings to configure it.

1.3.2. SSE (server-extension) installation details

Configure production settings in the following file `packages/server-extension/config/app.prod.json`:

- `cache.service.ttl.secs` - Default caching TTL, can be zero value
- `cache.gatewaysettings.ttl.secs` - Caching TTL for gateway settings call (see `packages/serverextension/src/middlewares/gatewaySettings.ts`). You might want to use TTL value '1' while testing SSE so that changes in gateway settings performed in OCC Admin become immediately available to SSE and Payment Widget respectively
- `crypto.service.key` - Random key which is used to encrypt data so that it is not tampered in UI
- `partner.developerId` - Leave the value as is
- `partner.solutionId` - Leave the value as is
- `logging.webhook.http` - Enable webhook request/response logging
- `logging.api.error` - Enable logging for errors
- `logging.api.access` - Enable logging for incoming requests
- `payments.secret.key` - Webhook secret key (SHA512)

1.3.3. Plugin installation details

- Copy the contents from `cybersource-plugins-oraclecommerce/plugins` into the `plugins` directory of your storefront (OSF workspace) code.

Copy `plugins/actions` into your storefront code and export the actions in the index and meta files: `plugins/actions/index.js`

```
export * from '@oracle-cx-commerce/actions';

export const flexMicroformAction = () => import('./flex-microform-action');
export const applePayValidationAction = () => import('./apple-pay-validation-action');
export const getPayerAuthSetupAction = () => import('./get-payer-auth-setup-action');
```

`plugins/actions/meta.js`

```
export * from '@oracle-cx-commerce/actions/meta';

export {flexMicroformAction} from './flex-microform-action/meta';
export {applePayValidationAction} from './apple-pay-validation-action/meta';
export {getPayerAuthSetupAction} from './get-payer-auth-setup-action/meta';
```

Copy `plugins/components` into your storefront code and export the components in the index and meta files: `plugins/components/index.js`

```
export * from '@oracle-cx-commerce/react-widgets';
export const IsvPaymentMethod = () => import('./isv-payment-method/index');
export const IsvCheckoutContinueToReviewOrderButton = () => import('./isv-checkout-continue-to-review-order-button');
export const IsvCheckoutPlaceOrderButton = () => import('./isv-checkout-place-order-button');
```

`plugins/components/meta.js`

```
export * from '@oracle-cx-commerce/react-widgets/meta';
export {default as IsvPaymentMethod} from './isv-payment-method/meta';
export {default as IsvCheckoutContinueToReviewOrderButton} from './isv-checkout-continue-to-review-order-button/meta';
export {default as IsvCheckoutPlaceOrderButton} from './isv-checkout-place-order-button/meta';
```

Copy `plugins/endpoints` into your storefront code and export the endpoints in the index and meta files: `plugins/endpoints/index.js`

```
export * from '@oracle-cx-commerce/endpoints';
export * from '@oracle-cx-commerce/oce-endpoints';
export const flexMicroformEndpoint = () => import('./flex-microform-endpoint');
export const paymentMethodConfigEndpoint = () => import('./payment-method-config-endpoint');
export const applePayValidationEndpoint = () => import('./apple-pay-validation-endpoint');
export const payerAuthSetupEndpoint = () => import('./payer-auth-setup-endpoint');
```

`plugins/endpoints/meta.js`

```
export * from '@oracle-cx-commerce/endpoints/meta';
export * from '@oracle-cx-commerce/oce-endpoints';
export {default as flexMicroformEndpoint} from './flex-microform-endpoint/meta';
export {default as paymentMethodConfigEndpoint} from './payment-method-config-endpoint/meta';
export {default as applePayValidationEndpoint} from './apple-pay-validation-endpoint/meta';
export {default as payerAuthSetupEndpoint} from './payer-auth-setup-endpoint/meta';
```

Copy `plugins/selectors` into your storefront code and export the selector in the index file: `plugins/selectors/index.js`

```
export * from './flex-microform-selector';  
export * from './payment-method-config-selector';
```

Copy plugins/fetchers into your storefront code and export the fetchers in the hook, index and meta files:

```
export {default as useFlexMicroformFetcher} from './flex-microform-fetcher/hook';  
export {default as usePaymentMethodConfigFetcher} from './payment-method-config-fetcher';
```

plugins/fetchers/index.js

```
export {default as flexMicroformFetcher} from './flex-microform-fetcher';  
export {default as paymentMethodConfigFetcher} from './payment-method-config-fetcher';
```

plugins/fetchers/meta.js

```
export {default as flexMicroformFetcher} from './flex-microform-fetcher/meta';  
export {default as paymentMethodConfigFetcher} from './payment-method-config-fetcher/meta';
```

Note: Install jwt-decode package by running 'yarn add jwt-decode -W'

- Deploy with the following command:

yarn occ deploy

2. Version History

This section gives details on the release notes of the ISV OCC Gateway.

Version 23.3.0

1. Network Tokenization
2. Updated the user guide

Compatible with latest OSF: v5.0.0

OCC Version: v23.08.01

Version 23.2.0

1. Allow merchants to configure when to enforce Strong Consumer Authentication.
2. Module compatible with latest Oracle 23C upgrade.

Compatible with OSF: v5.0.0

OCC Version: v23.08.01

Version 23.1.0

1. Payer Authentication from Hybrid model to direct connection API
2. Included custom properties in the request
3. Addressed Checkmarx issues
4. Updated the user guide with support items

Note: Local instance doesn't support Payer Authentication with saved card due to product limitation.

Compatible with OSF: v4.4.0

OCC Version: v23.1.1.1

Version 22.1.0

1. Implemented Payment Acceptance, Fraud Management and Payment Security Services for Credit Card, Google Pay and Apple Pay for OSF framework.
2. Security and vulnerability issues, Checkmarx issue and
3. Handled promise rejection with error logs
4. Request - response logs for webhook and API calls

Compatible with OSF: 3.7

3. Configuration Details

This section provides the details about the steps for configuring the extension with Merchant Details, Payment Method enabling for Credit Card, Google Pay and Apple Pay

3.1. General Settings

Step 1: Go to OCC Admin -> Settings ->Payment Processing and then click on “Payment Gateways”.

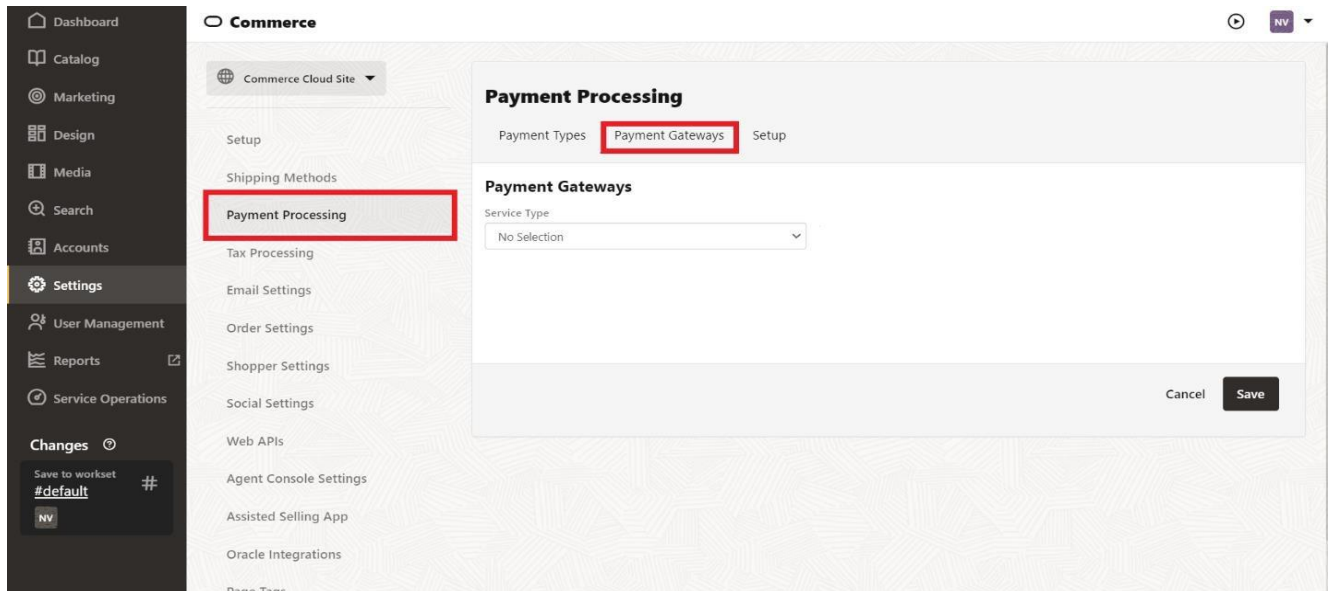


Figure 4: Payment Gateways

Step 2: Under Payment Gateways, select the Service Type “ISV OCC Gateway” and enable the Payment Gateway

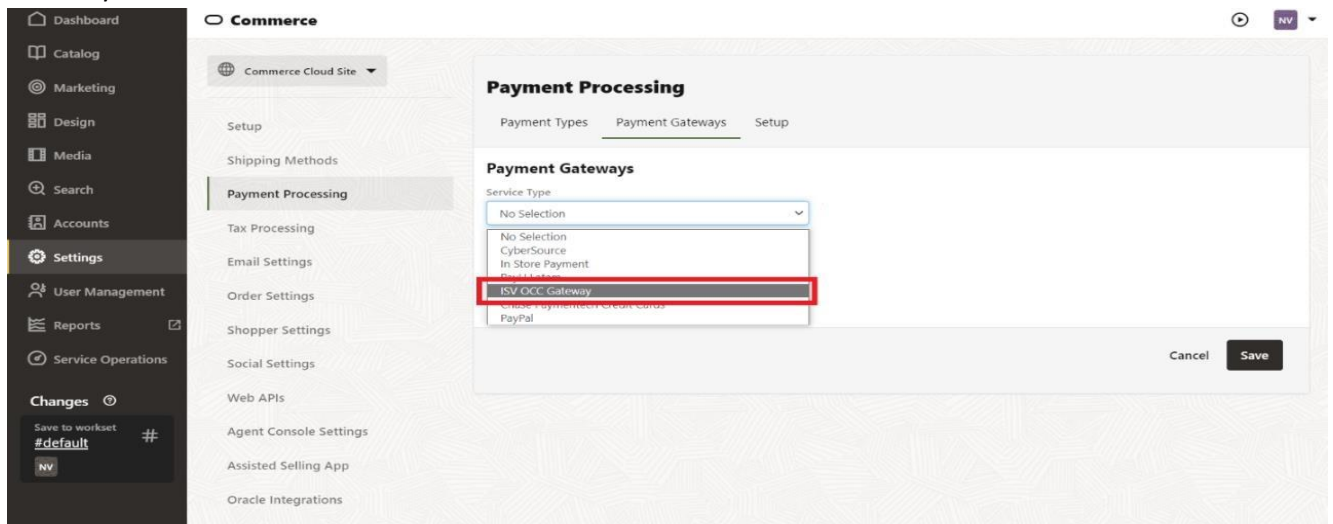


Figure 5: ISV OCC Gateways

Step 3: Under ISV OCC Gateway, enter the details for Preview, StoreFront and Agent Configurations

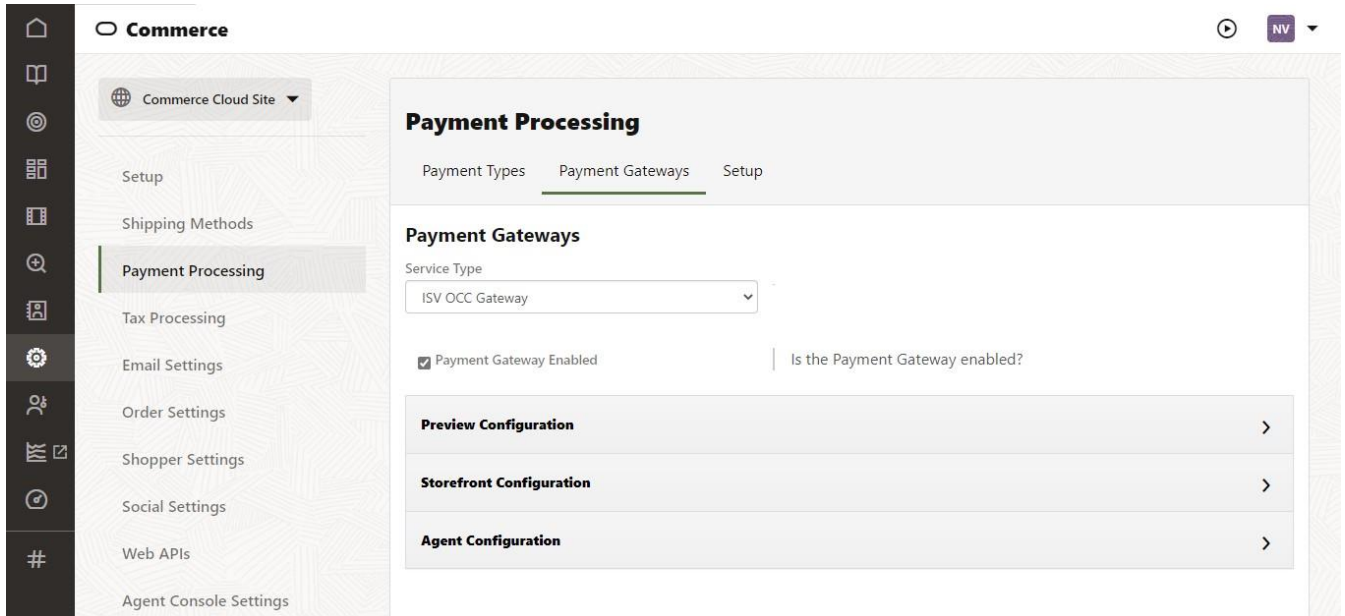


Figure 6: Preview Configuration

Step 4: Save and publish the changes

Section	Description
Merchant ID	Enter the Merchant ID details
Key id	Enter the Merchant Key ID
Secret key	Enter the Merchant Secret Key
Key alias	Key Alias (in case authentication type = jwt)
Key pass	Key Pass (in case authentication type = jwt)
Key file name	Key File Name (in case authentication type = jwt)
Authentication type	Choose the authentication type from the drop down
Environment	PSP REST API environment to send requests to
Google Pay Gateway	To retrieve payment and customer information from a payment gateway that's supported by the Google Pay API. Gateway's identifier, which is issued by Google
Google Pay Gateway merchant ID	To retrieve payment and customer information from a payment gateway that's supported by the Google Pay API. Your gateway account ID, which is provided by the gateway

Google Pay Merchant ID	A Google merchant identifier issued after registration with the Google Pay Business Console. Required when Payments Client is initialized with an environment property of PRODUCTION. See Request production access for more information about the approval process and how to obtain a Google merchant identifier
Google Pay Merchant Name	Merchant name encoded as UTF-8. Merchant name is rendered in the payment sheet. In TEST environment, or if a merchant isn't recognized, a "Pay Unverified Merchant" message is displayed in the payment sheet
Google Pay Supported Networks	Google Pay Supported networks
Apple Pay Merchant ID	Apple Pay Merchant ID
Apple Pay initiative context	Fully qualified domain name associated with your Apple Pay Merchant Identity Certificate
Apple Pay supported network	Apple Pay Supported Networks
Apple Pay display name	Apple Pay Display Name

Table 1: Configuration Fields

3.2. Fraud Management Settings

3.2.1. Enabling Payer Authentication

Step 1: Login to OCC Admin dashboard and click on Settings.

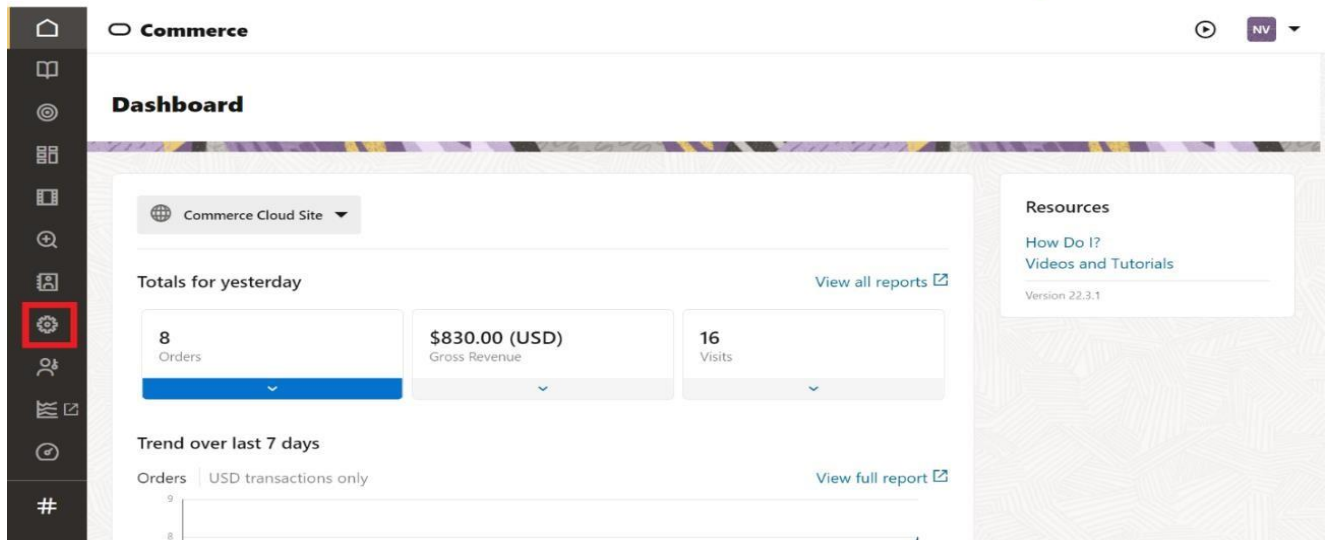


Figure 7: Enabling Payer Authentication

Step 2: Go to Settings -> Payment Processing and then click on “Payment Gateways”.

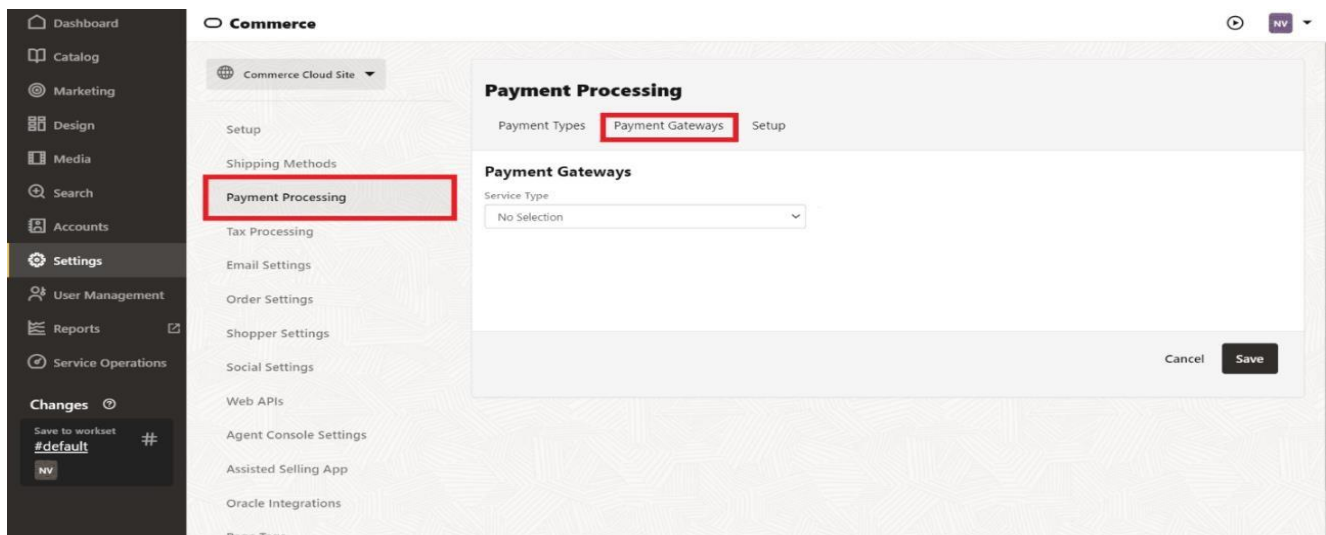


Figure 8: Payment Gateways

Step 3: Under Payment Gateways, select the Service Type “ISV OCC Gateway”. Select the Credit Card Payer Authentication Enabled checkbox. Save the changes.

Commerce

Key File Name (in case authentication type = jwt) (required)

keyFileName

Authentication Type (required)

http_signature

Environment (required)

environment

☒ Credit Card Payer Authentication Enabled

Indicates whether Payer Authentication (3D Secure) will be enabled

☒ Enforce SCA for Saving Card

If enabled, card holder will be 3DS challenged when saving a card (enforcing Strong Customer Authentication)

☐ Network Token Updates

Subscribe to Network Token life cycle updates

Sale Enabled

☐ Card

Indicates if authorizing and taking payment will be done at the same time for a particular payment mode

Figure 9: Enabling Payer Authentication

Note: Local Instance doesn't support Payer Authentication with saved cards

3.2.2. Enabling Strong Customer Authentication

When payer authentication is enabled, if a transaction gets declined with the reason as Strong Customer Authentication required, then another request will be sent from Oracle Commerce Cloud automatically for the same order and the customer will be 3DS challenged

This section covers information on how to enable a Strong Customer Authentication Service. Under Payment Gateways -> "ISV OCC Gateway", select the Enforce Strong Customer Authentication checkbox to enable the Strong Customer Authentication. Save the changes.

Commerce

Key File Name (in case authentication type = jwt) (required)

Authentication Type (required)

Environment (required)

☒ Credit Card Payer Authentication Enabled
 Indicates whether Payer Authentication (3D Secure) will be enabled

☒ Enforce SCA for Saving Card
 If enabled, card holder will be 3DS challenged when saving a card (enforcing Strong Customer Authentication)

☐ Network Token Updates
 Subscribe to Network Token life cycle updates

Sale Enabled
☐ Card
 Indicates if authorizing and taking payment will be done at the same time for a particular payment mode

Figure 10: Enabling Strong Customer Authentication

Note: The “Enforce SCA for Saving Card” setting will be only available if “Payer Authentication” is enabled

3.2.3. Enabling Device fingerprint

This section covers information on how to enable Device Fingerprint Service.

Under Payment Gateways -> “ISV OCC Gateway”, select the Device Fingerprint Enabled checkbox to enable the Device Fingerprint and enter the details for Device Fingerprint URL & Device Fingerprint Organization Id. Save the changes.

Commerce

DM Decision Skip
☐ Card
☐ Apple Pay
☐ Google Pay
 Indicates which payment modes should skip the decision manager step

Daily Report Name (required)

Device Fingerprint URL (required)

Device Fingerprint Organization Id (required)

☒ Device Fingerprint Enabled

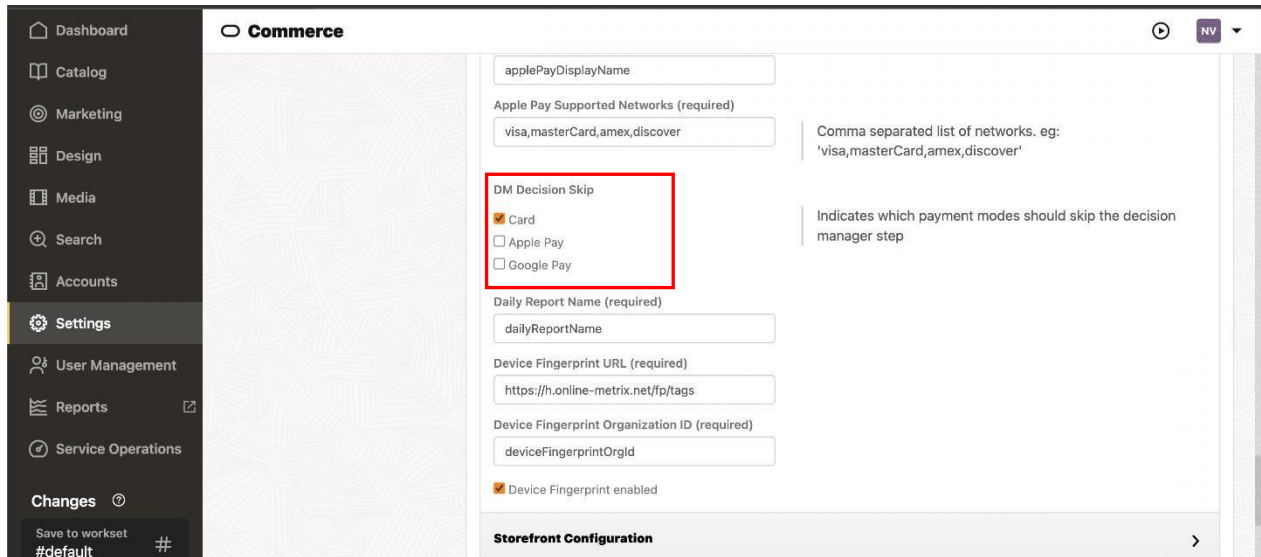
Cancel Save

Figure 11: Enabling Device fingerprint

3.2.4. Advanced Fraud Screening with Decision Manager

This section provides information on Configuring Decision Manager Services in OCC.

Under Payment Gateways -> “ISV OCC Gateway”, uncheck the DM Decision Skip to enable Decision Manager for the desired Payment Service.



The screenshot shows the Oracle Commerce Cloud Settings page for the ISV OCC Gateway. The left sidebar contains navigation links: Dashboard, Catalog, Marketing, Design, Media, Search, Accounts, Settings (highlighted), User Management, Reports, and Service Operations. The main content area is titled 'Commerce' and contains various configuration fields. The 'DM Decision Skip' section is highlighted with a red box, showing three options: 'Card' (checked), 'Apple Pay' (unchecked), and 'Google Pay' (unchecked). The 'Card' option is selected, indicating that the decision manager step should be skipped for card payments. Other fields include 'applePayDisplayName', 'Apple Pay Supported Networks (required)' (with a comma-separated list of networks), 'Daily Report Name (required)', 'Device Fingerprint URL (required)', 'Device Fingerprint Organization ID (required)', and 'Device Fingerprint enabled' (checked). A 'Storefront Configuration' section is visible at the bottom right.

Figure 12: DM Decision Skip

4. Reporting

This section covers the details of the Reports imported from Cybersource to Oracle Commerce Cloud. Following Reports are generated in Cybersource and are imported in Oracle Commerce Cloud:

1. On-demand conversion
2. Daily conversion

4.1. Reporting configuration

This section covers the configuration to be made for Reporting:

The types of Reports supported are:

1. **On-demand conversion:** Daily transaction level report that provides details related to each individual transaction.
2. **Daily conversion:** Report that can be scheduled for daily execution which returns conversion report for a given date.

In settings, give the daily report name as **ConversionDetailReport_Daily_Classic**

The screenshot shows the 'Commerce' settings page in the Oracle Commerce Cloud Admin interface. The left sidebar contains navigation links: Dashboard, Catalog, Marketing, Design, Media, Search, Accounts, Settings (highlighted), User Management, Reports, and Service Operations. The 'Changes' section shows 'Save to workset #default'. The main content area is titled 'Commerce' and contains several configuration fields:

- Google Pay Supported Networks (required):** A text input field containing 'AMEX,DISCOVER,INTERAC,JCB,MASTERCARD,VISA'.
- Apple Pay Merchant ID (required):** A text input field containing 'merchant.com.cybs.wiprofitd'.
- Apple Pay initiative (required):** A text input field containing 'web'.
- Apple Pay initiative context (required):** A text input field containing 'www.occ.isvplugins.com'.
- Apple Pay Supported Networks (required):** A text input field containing 'visa.masterCard.amex.discover'.
- DM Decision Skip:** A section with checkboxes for 'Card' (checked), 'Apple Pay' (checked), and 'Google Pay' (unchecked).
- Daily Report Name (required):** A text input field containing 'dailyReportName', which is highlighted with a red box.
- https://honline-metricx.net/lp/tags.js**: A text input field at the bottom.

Figure 13: Enter the Daily Report Name

5. Shipping Region

This section covers the details about changing the Shipping Region in OCC Admin

Step 1: Under Settings -> Shipping Methods, click on New Shipping Region

The screenshot shows the 'Shipping Methods' page in the Oracle Commerce Cloud Admin interface. The left sidebar contains navigation links: Dashboard, Catalog, Marketing, Design, Media, Search, Accounts, Settings (highlighted), User Management, Reports, and Service Operations. The 'Changes' section shows 'Save to workset'. The main content area is titled 'Shipping Methods' and contains a table of shipping methods:

Shipping Method	Type	Selected Site
Shipping Cost	Internally Priced	✓
Shipping Cost External	Externally Priced	✓
US shipping	Internally Priced	✓

Below the table, there is a 'Default Shipping Country' dropdown menu set to 'United States'. To the right of this dropdown is a text box explaining: 'Preselected in the shipping address form during checkout, though shoppers can change it. The country must be part of an enabled shipping method. Otherwise, no country is preselected at checkout.'

At the bottom of the page, there is a section titled 'Shipping Regions' with a '+ New Shipping Region' button highlighted.

Figure 14: Shipping methods

Step 2: Enter the display name as per your preference and select the shipping country and save

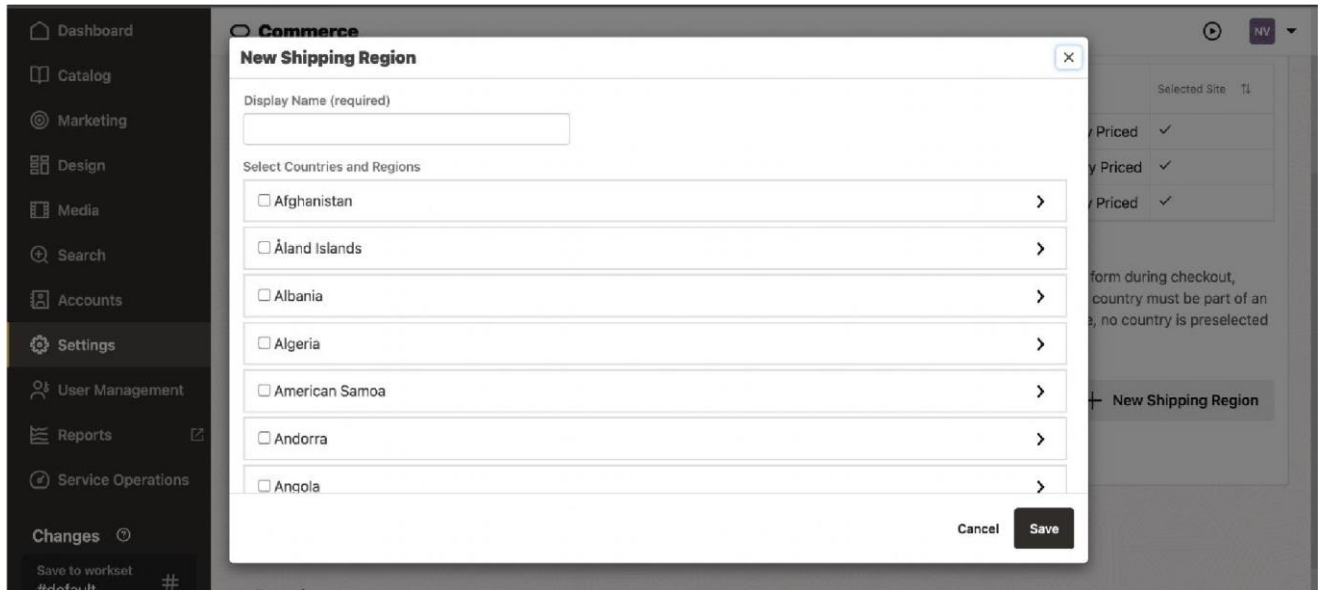


Figure 15: New Shipping Region

6. Placing order from Storefront

6.1. Placing an order from storefront using Credit Card

Step 1: Open the Oracle Commerce Cloud Storefront

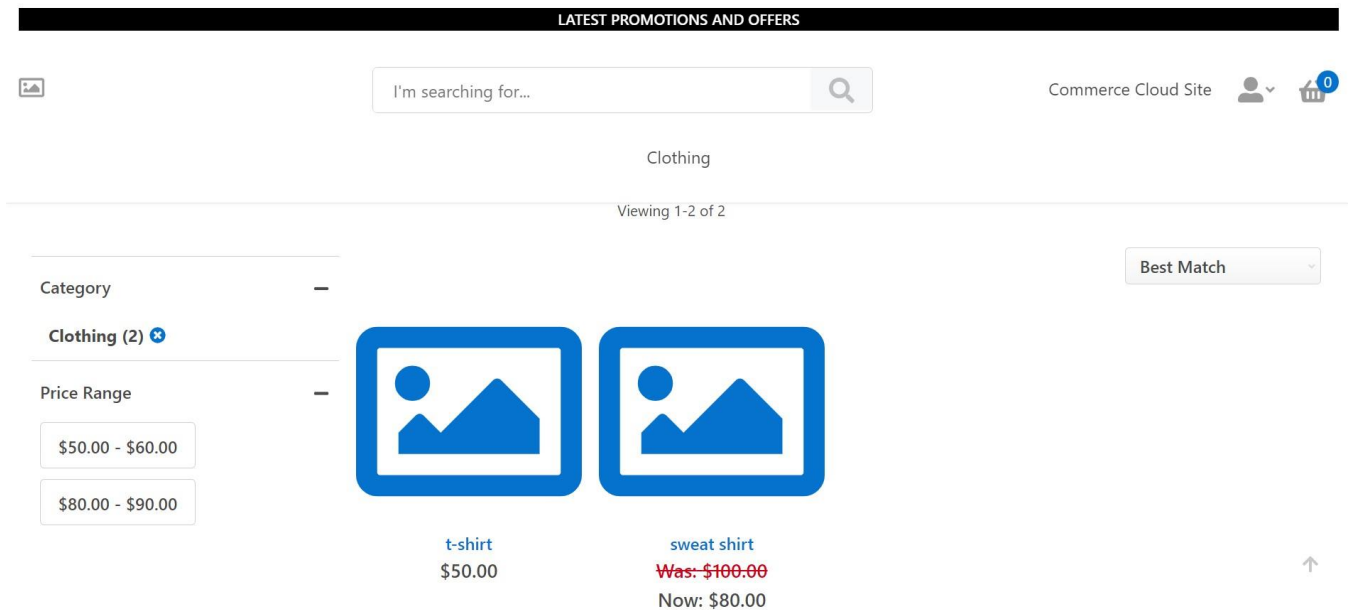


Figure 16: Oracle Commerce Cloud StoreFront

Step 2: Add an item in to cart

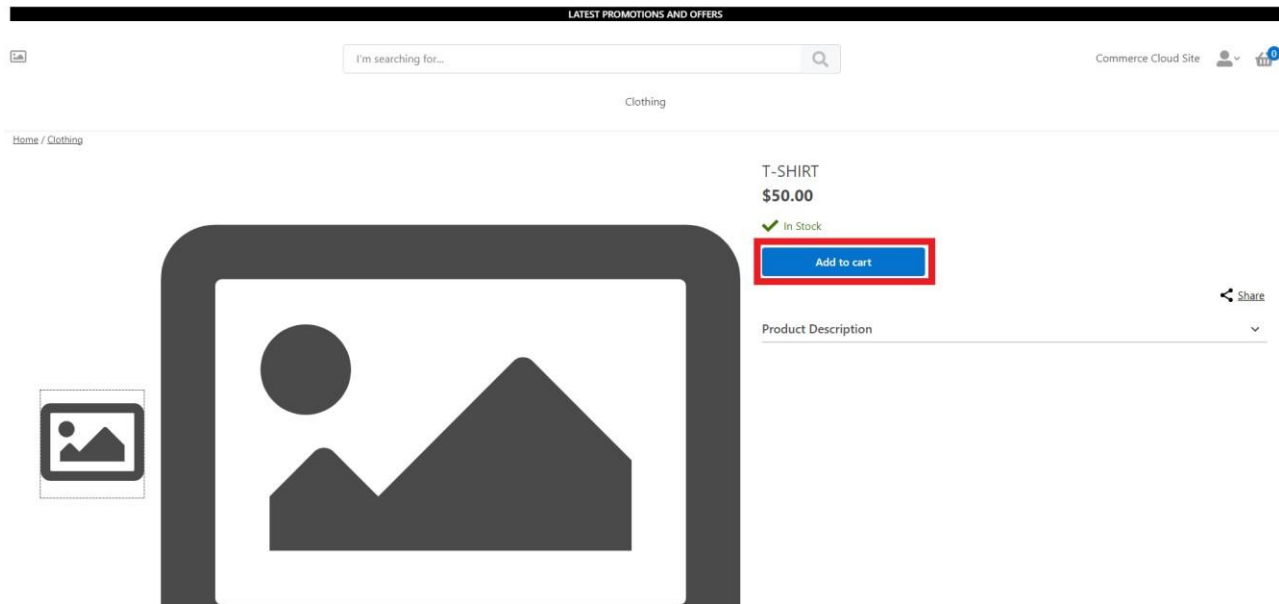


Figure 17: Oracle Commerce Cloud Add to Cart

Step 3: After adding an item to cart, click on “Checkout” option.

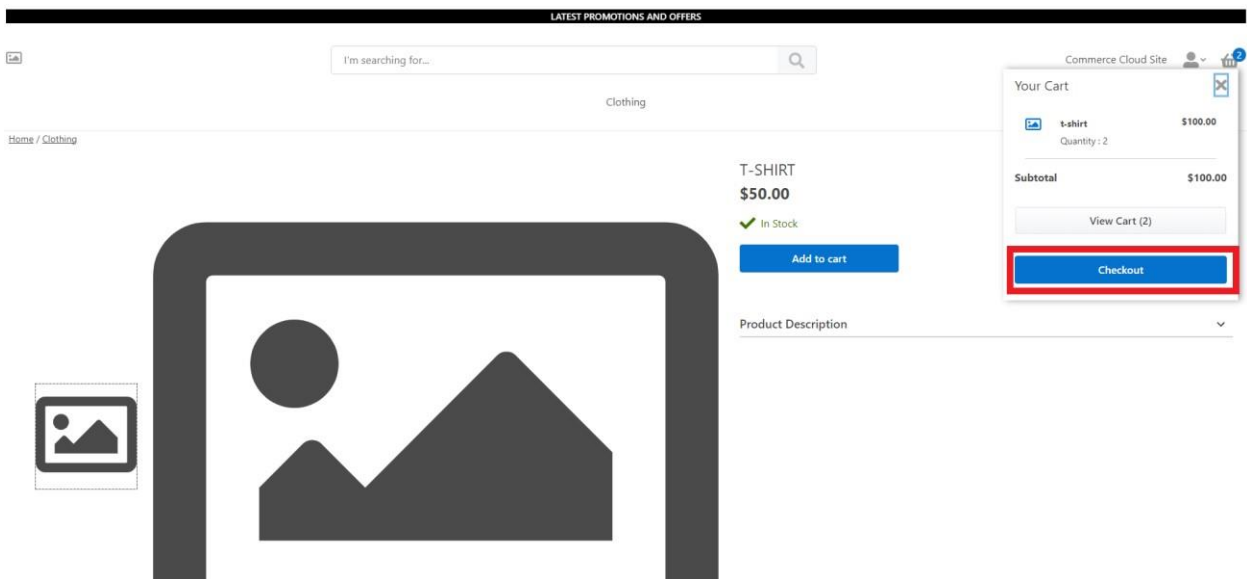


Figure 18: Oracle Commerce Cloud Checkout

Step 4: Click on “Checkout as Guest”

Oracle Commerce Cloud – ISV Gateway User Installation Guide

LATEST PROMOTIONS AND OFFERS

Commerce Cloud Site

Home / Checkout Login

Returning Customer?

Email Address

Password

[Forgot your password?](#)

Checkout

Checkout As Guest

Help Customer Services Contact Us Shipping Information Returns & Refunds

Copyright © 2020 Oracle. All rights reserved.

Figure 19: Oracle Commerce Cloud Checkout as Guest

Step 5: Fill in the Shipping details.

Checkout

1 SHIPPING 2 PAYMENT 3 REVIEW

SHIPPING

Delivery Address

All fields required unless marked as optional.

First Name

Last Name

Country

United States

ZIP Code

State

Alabama

Street Address

Town/City

Phone Number (optional)

Continue

Order Summary

Subtotal	\$100.00
Shipping	Free
Tax	\$0.00
Total	\$100.00

Help Customer Services Contact Us Shipping Information Returns & Refunds

Copyright © 2020 Oracle. All rights reserved.

Figure 20: Shipping address

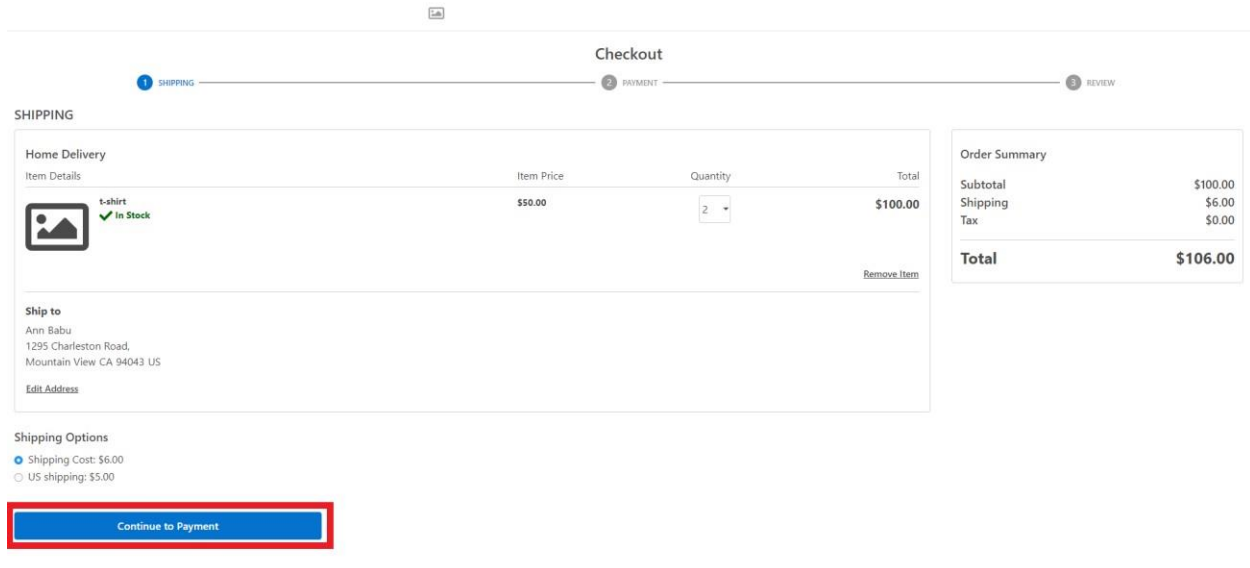
Step 6: Select the desired shipping option & click “Continue to Payment”

Checkout

1 SHIPPING 2 PAYMENT 3 REVIEW

SHIPPING

Home Delivery

Item Details	Item Price	Quantity	Total
 t-shirt ✓ In Stock	\$50.00	2	\$100.00

[Remove Item](#)

Ship to
 Ann Babu
 1295 Charleston Road,
 Mountain View CA 94043 US
[Edit Address](#)

Shipping Options
☒ Shipping Cost: \$6.00
☐ US shipping: \$5.00

Order Summary

Subtotal	\$100.00
Shipping	\$6.00
Tax	\$0.00
Total	\$106.00

[Continue to Payment](#)

Figure 21: Continue to Payment

Step 7: Select the required Payment method and enter the necessary details. For Credit Card, click on Continue to Review Order button and respective buttons for Google Pay & Apple Pay.

PAYMENT

☒ Credit Card

Card Number VISA MASTERCARD AMERICAN EXPRESS

Expiry Date (MM/YY) / CVV Number

Name on Card

Billing Address
 1295 Charleston Road,
 Mountain View ABE 94043 GB
[Edit Address](#)

☐ Google Pay

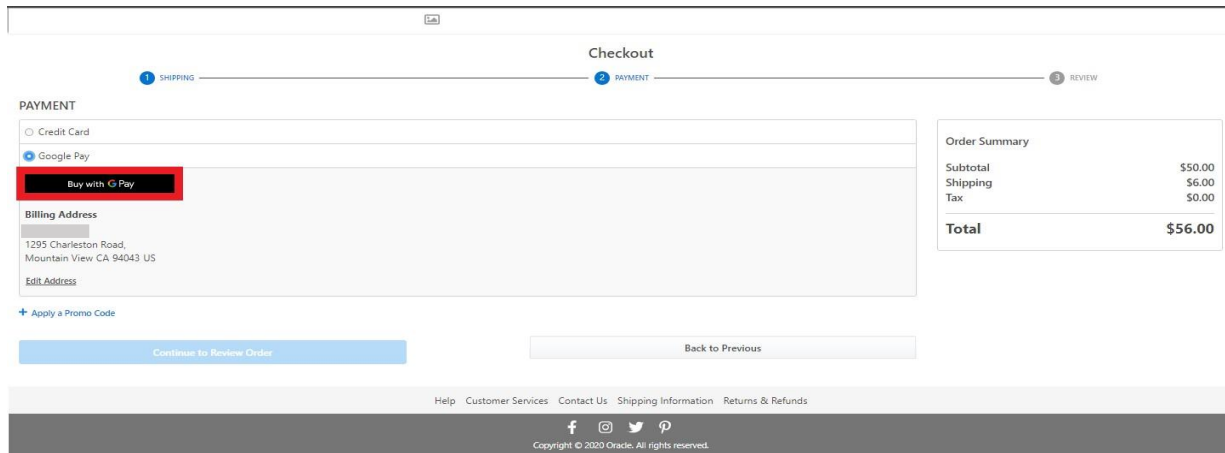
[+ Apply a Promo Code](#)

[Continue to Review Order](#) [Back to Previous](#)

Order Summary

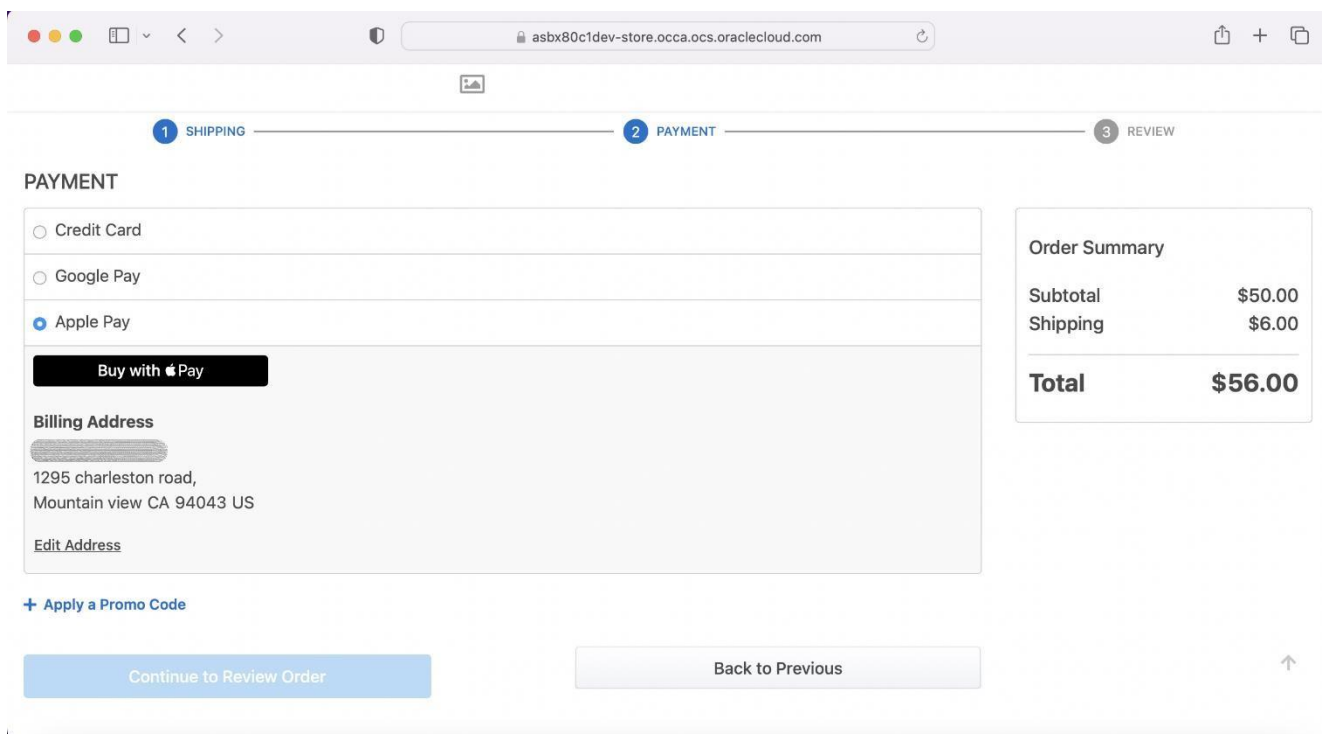
Subtotal	\$2,101.00
Shipping	Free
Tax	\$0.00
Total	\$2,101.00

Figure 22.1: Credit Card Payment Method



The screenshot shows the checkout process with three steps: 1 SHIPPING, 2 PAYMENT, and 3 REVIEW. The PAYMENT step is active. Under the PAYMENT section, the Google Pay option is selected, and a red box highlights the "Buy with G Pay" button. Below this, the Billing Address is displayed: 1295 Charleston Road, Mountain View CA 94043 US, with an "Edit Address" link. At the bottom of the payment section, there is a link to "Apply a Promo Code" and two buttons: "Continue to Review Order" and "Back to Previous". On the right side, the Order Summary is shown with a Subtotal of \$50.00, Shipping of \$6.00, Tax of \$0.00, and a Total of \$56.00. The footer contains links for Help, Customer Services, Contact Us, Shipping Information, and Returns & Refunds, along with social media icons and a copyright notice for 2020 Oracle.

Figure 22.2: Google Pay Payment Method



The screenshot shows the checkout process with three steps: 1 SHIPPING, 2 PAYMENT, and 3 REVIEW. The PAYMENT step is active. Under the PAYMENT section, the Apple Pay option is selected, and a black box highlights the "Buy with Apple Pay" button. Below this, the Billing Address is displayed: 1295 charleston road, Mountain view CA 94043 US, with an "Edit Address" link. At the bottom of the payment section, there is a link to "Apply a Promo Code" and two buttons: "Continue to Review Order" and "Back to Previous". On the right side, the Order Summary is shown with a Subtotal of \$50.00, Shipping of \$6.00, and a Total of \$56.00. The browser address bar shows the URL: asbx80ctdev-store.occa.ocs.oraclecloud.com.

Figure 22.3: Apple Pay Payment Method

Step 8: Enter the email address, click on “Place Order” button

Oracle Commerce Cloud – ISV Gateway User Installation Guide

The screenshot displays the checkout process in Oracle Commerce Cloud. At the top, a progress bar shows three steps: 1. SHIPPING, 2. PAYMENT, and 3. REVIEW. The 'REVIEW ORDER' section is active. It includes 'Contact Information' with an email address 'test@gmail.com' highlighted in a red box. Below this is 'Shipping Details' for 'Home Delivery' of a 't-shirt' for \$50.00. The 'Ship to' address is 'Ann Babu, 1295 Charleston Road, Mountain View CA 94043 US'. The 'Payment Details' section shows a VISA card ending in 1111, expiring 05/2025. An 'Order Summary' on the right lists Subtotal (\$50.00), Shipping (\$6.00), Tax (\$0.00), and a Total of \$56.00. A red box highlights the 'Place Order' button at the bottom right. The footer contains links for Help, Customer Services, Contact Us, Shipping Information, and Returns & Refunds, along with social media icons and a copyright notice for 2020 Oracle.

Figure 23: Place Order

The screenshot shows the 'Thank you for your order' confirmation page. It states that the order has been submitted and a confirmation email has been sent to 'test@gmail.com'. The order number is 'o60142'. Below this, there is a section titled 'Make checkout faster and easier' with a prompt to 'Create an account to store addresses, payment methods and orders that are placed while you are signed in.' The form includes fields for 'First Name' (Ann), 'Last Name' (Babu), and 'Email Address' (test@gmail.com). A checkbox for 'I want to get email updates.' is present. A blue 'Create an Account' button is highlighted, and a 'Continue Shopping' button is also visible. The footer is identical to the previous figure, showing navigation links and a copyright notice for 2020 Oracle.

Figure 24: Order placed

7. Oracle Commerce Cloud Storefront Cancel an order

The Customer can cancel the order from Oracle Commerce Cloud.

Step 1: Click on “Agent Console Settings” and then “Remorse Period”.

Oracle Commerce Cloud – ISV Gateway User Installation Guide

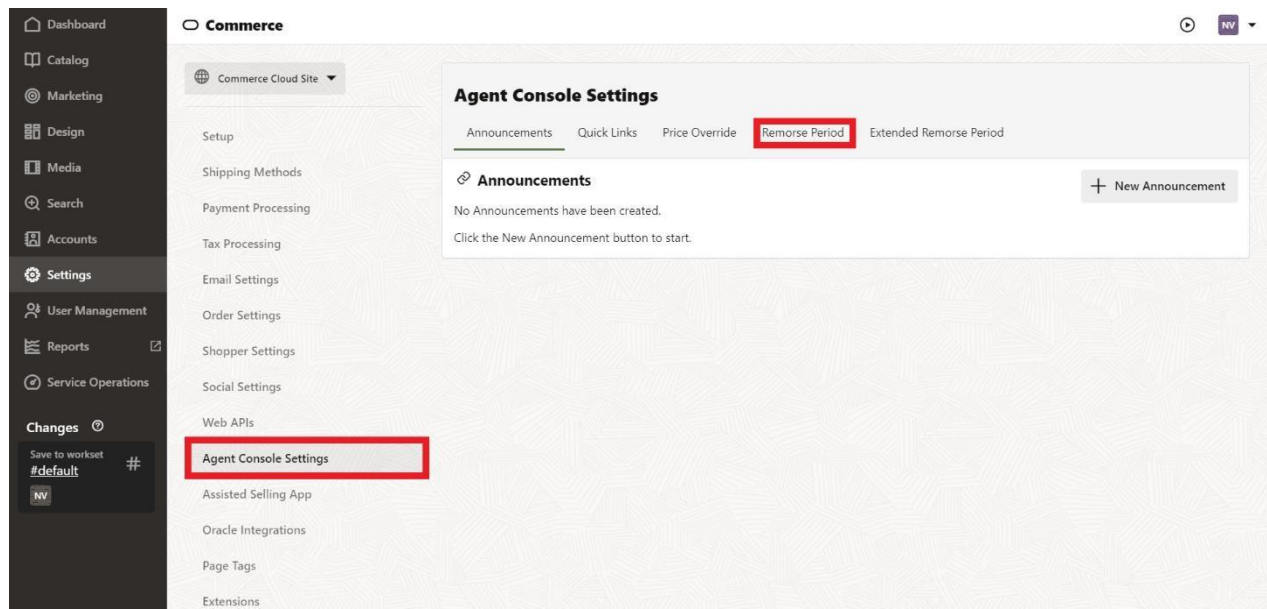


Figure 25: Remorse Period

Step 2: Specify the Time for Remorse Period and save.

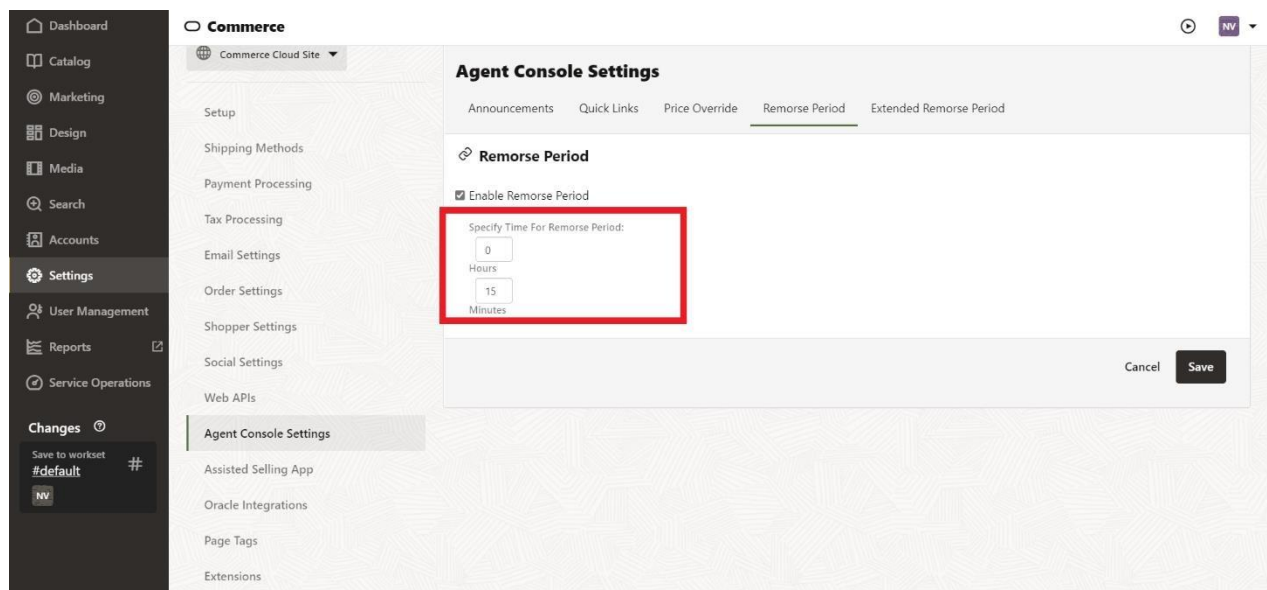


Figure 26: Enable Remorse Period

Step 3: Login into user account and select Order History from the profile

Oracle Commerce Cloud – ISV Gateway User Installation Guide

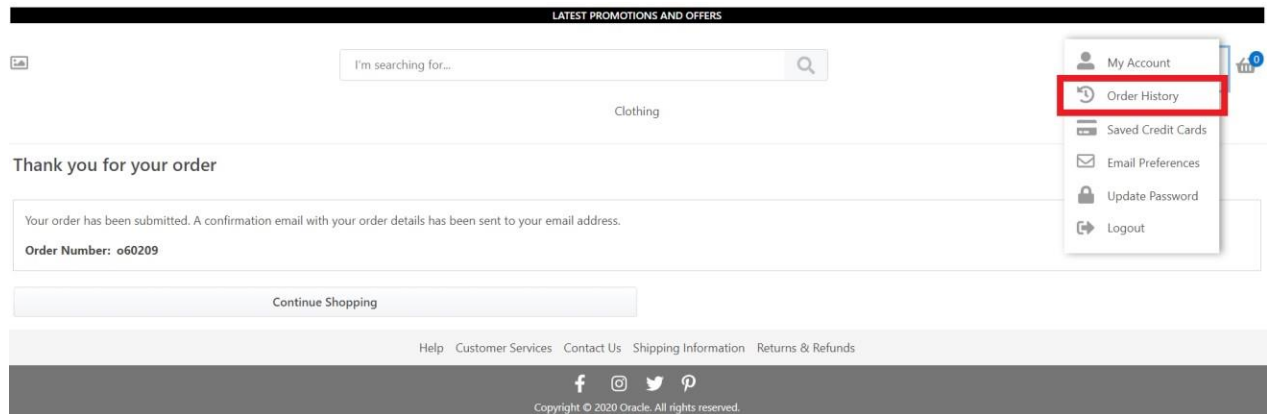


Figure 27: Order History

Step 4: Click on the order to be cancelled

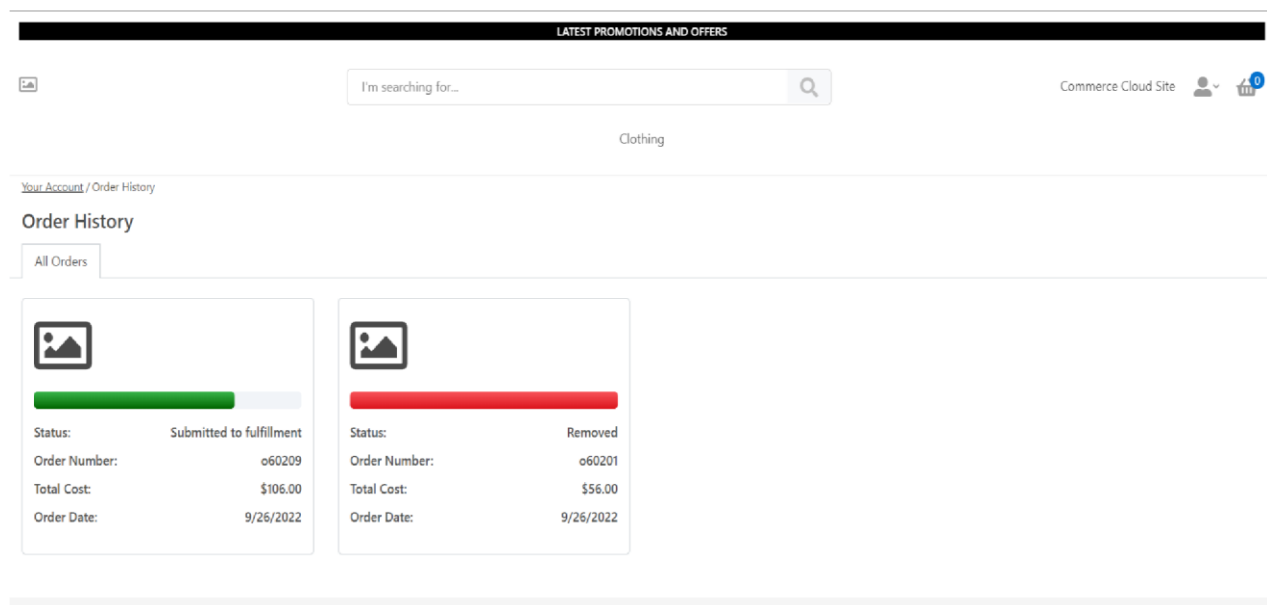


Figure 28: Orders page

Step 5: Click on "Cancel This Order" button

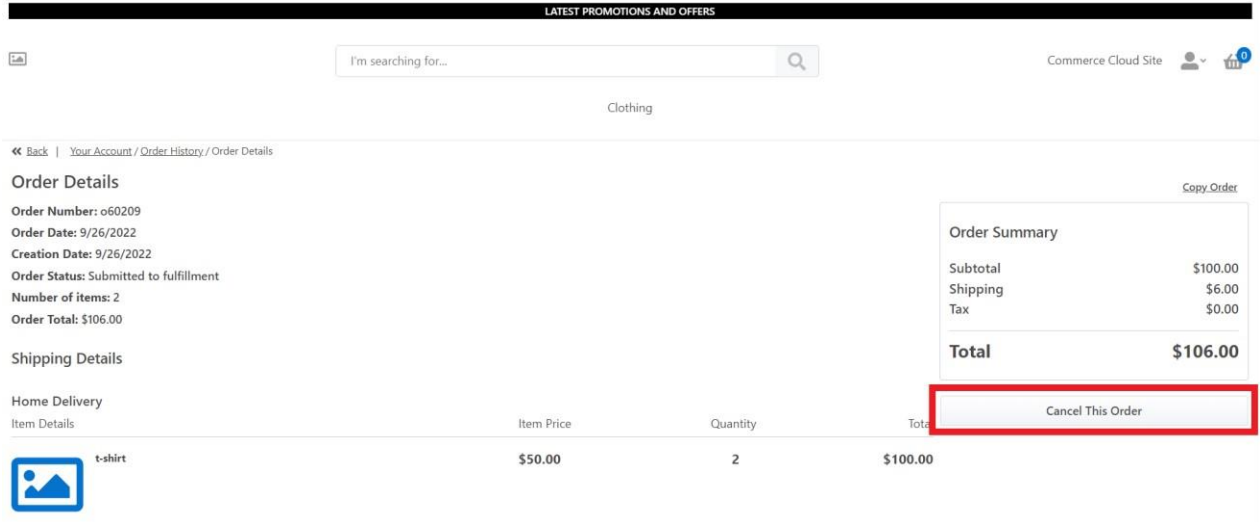


Figure 29: Cancel This order

Step 6: Select the reason for the cancellation & click on “Submit Cancellation”

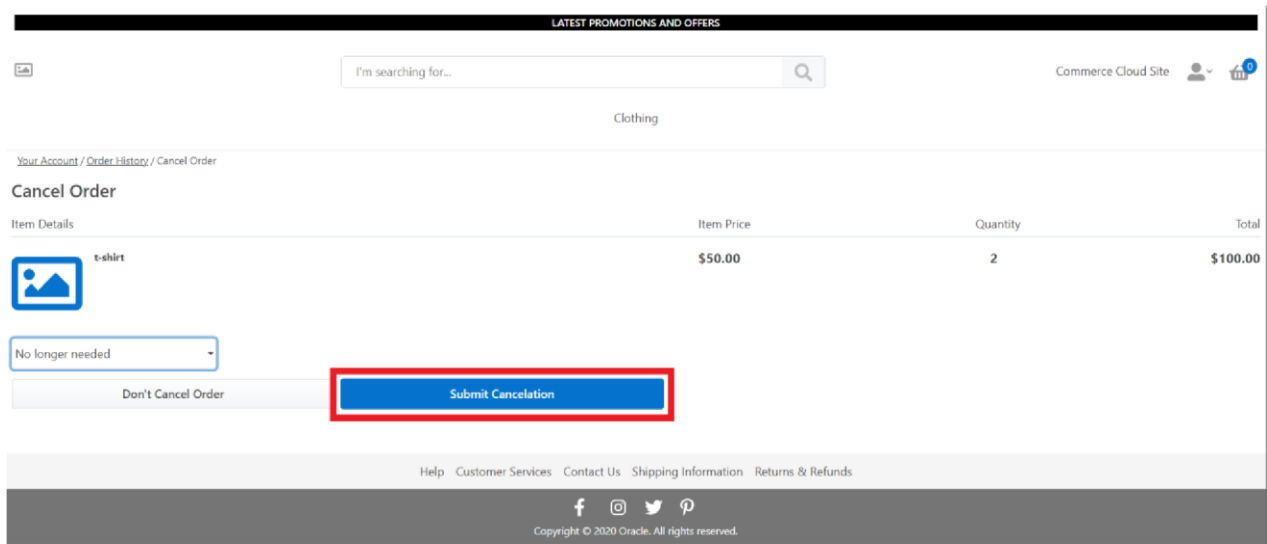


Figure 30: Submit cancellation

Step 7: The Order Cancellation screen.

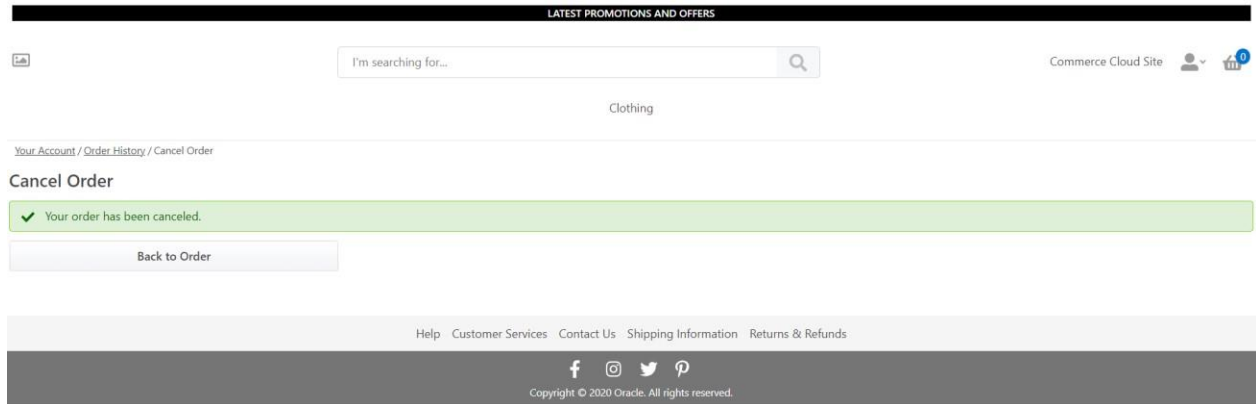


Figure 31: Order Canceled successfully

8. Apple Pay Configuration

In order to use Apple Pay there are few assumptions and prerequisites one should take into consideration.

- You must have an Apple Developer Account.
- All pages that incorporate Apple Pay must be served over HTTPS.
- Your website must comply with the Apple Pay guidelines. [Click here](#) for more information.
- Your website must have HTTPS mode enabled. [Click here](#) to know more about server requirements

In order to configure Apple Pay added in the ISV OCC Gateway plugin, you need to perform the following actions:

1. Register an Apple Pay Merchant ID
2. Validate your Store domain in Apple Pay.
3. Create a Payment Processing Certificate.
4. Create a Merchant Identity Certificate.

8.1. Create a Merchant ID

[Click here](#) to visit Apple's official article

1. Go to [Certificates, Identifiers & Profiles](#) page.
2. Select Identifiers from the sidebar. Click the "+" button.
3. Find and select the Merchant IDs from the list.



Figure 32: Register a New Identifier Page

4. Fill in the Description and the Identifier field values. Record the value of the Identifier as it is required in the following configuration process. Click the Continue button.

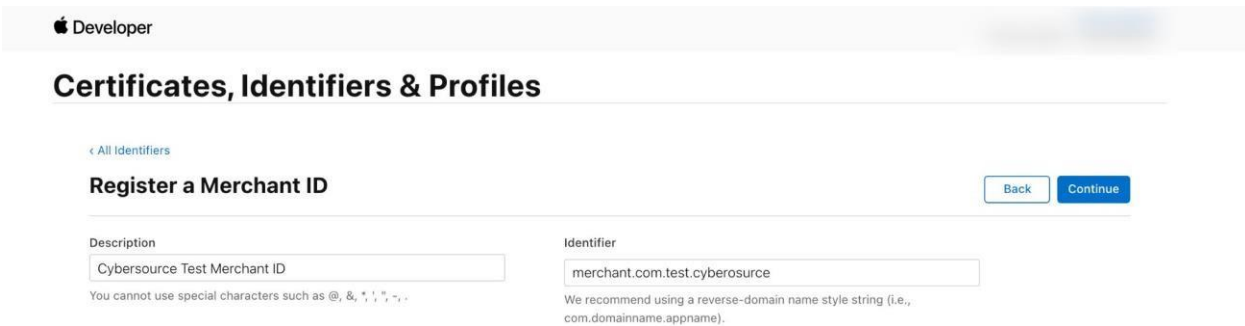


Figure 33: Register a Merchant ID Page

5. Click the Register button to finish the Merchant ID creation process.

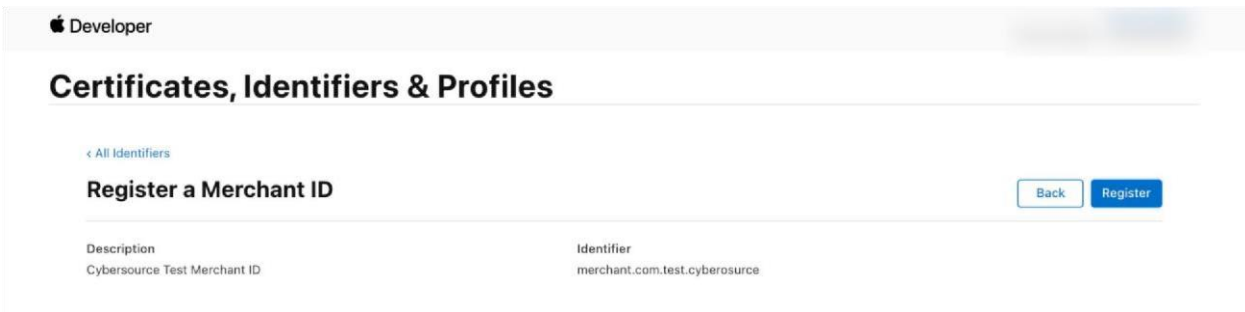


Figure 34: Finishing a New Merchant ID Registration Page

8.2. Create Payment Processing Certificate

A Payment Processing certificate is used to establish secure communication between Apple Pay and Cybersource.

1. Log in to your Cybersource Enterprise Business Center account.
2. On the left navigation panel go to “Payment Configuration” > “Digital Payment Solutions”.
3. Click “Configure” button near “Apple Pay”.
4. Enter the value of your Apple Pay Merchant ID in the Apple Merchant ID field.

Step 1: Generate the Certificate Signing Request

The Certificate Signing Request (CSR) contains an identifier and a public key. Apple requires this file to encrypt sensitive payment credentials.

To use the CSR generated on this page, download and submit it to Apple (see step 2). Your old certificate can be used until its expiration, 25 months from the date of generation or until the new CSR is submitted to Apple.

Enter the Apple Merchant ID you registered in the certificates, identifiers, and profiles area of the member center on the Apple Website.

Apple Merchant ID

GENERATE NEW CERTIFICATE SIGNING REQUEST

Step 2: Submit the CSR to Apple

Submit the CSR to Apple to get the required Apple Pay Certificate. [Learn more about the submission process.](#)

Step 3: Generate Transaction Security Key

Complete this step if you use the SDK. When using the SDK, each transaction request originating from your iOS application must include a unique signature.

At least one CyberSource SOAP Toolkit API transaction security key exists. If you do not want to reuse an existing key, [generate a new transaction security key here.](#)

Important - To use Apple Pay, your processor must support payment network tokenization.

Figure 35: EBC Apple Pay Registration Page

5. Click Generate New Certificate Signing Request button.
6. Save the generated CSR on your disk.
7. Go to [Certificates, Identifiers & Profiles](#) page on your Apple Developer portal.
8. Select Identifiers from the sidebar.
9. Select your Merchant ID from the list.
10. Under Apple Pay Payment Processing Certificate click the Create Certificate button.
11. Select the CSR file you have downloaded from EBC in the previous step.

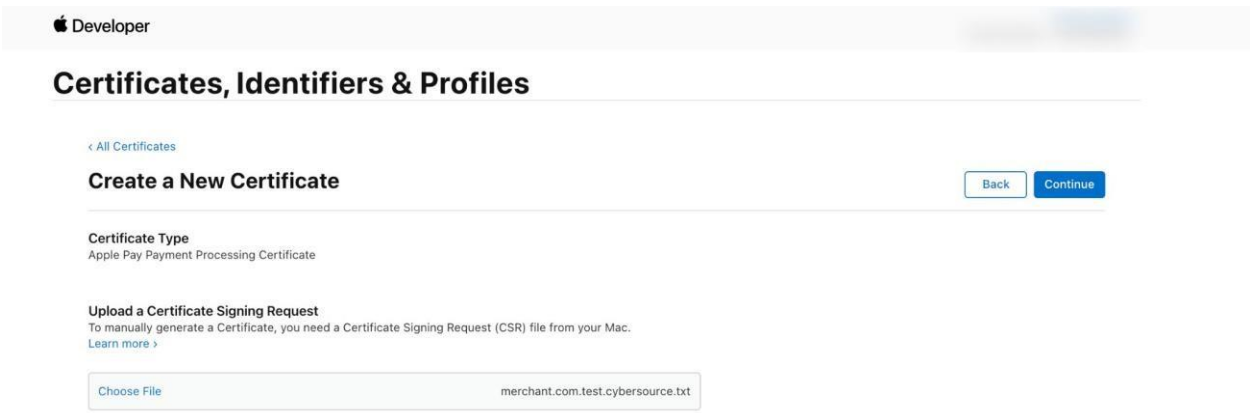


Figure 36: Uploading Payment Processing Certificate Request

12. Click Continue button.
13. Click Download button.

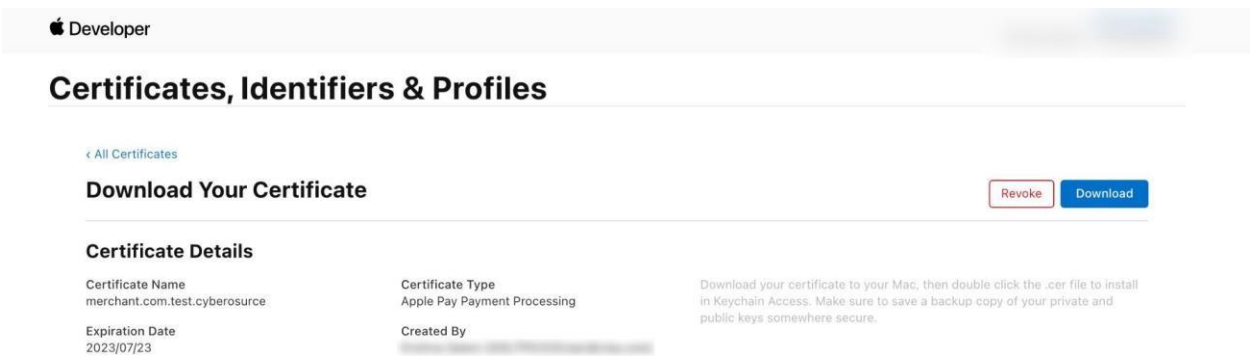


Figure 37: Downloading Payment Processing Certificate Request

8.3. Domain Validation

1. Go to [Certificates, Identifiers & Profiles](#) page on your Apple Developer portal.
2. Select your Merchant ID from the list.
3. Click Add Domain button under Merchant Domains.

The screenshot shows the Apple Developer console interface. At the top, the 'Certificates, Identifiers & Profiles' section is selected. Below this, the 'Edit or Configure Merchant ID' page is displayed. The 'Name' field contains 'Cybersource Test Merchant ID' and the 'Identifier' field contains 'merchant.com.test.cybersource'. A note states: 'You cannot use special characters such as @, &, *, !, ", ~, -, .'. Below this, the 'Apple Pay Payment Processing Certificate' section is visible, showing a certificate for 'merchant.com.test.cybersource' with a type of 'Apple Pay Payment Processing' and an expiration date of 'Jul 23, 2023 (Active Certificate)'. There are 'Revoke' and 'Download' buttons for this certificate. A 'Create Certificate' button is also present. The 'Apple Pay Payment Processing on the Web' section follows, with instructions on registering and verifying domains. A note mentions that incorporation of Apple Pay Payment Processing into a website is subject to Apple's Terms and Conditions. The 'Merchant Domains' section has an 'Add Domain' button. Finally, the 'Apple Pay Merchant Identity Certificate' section has a 'Create Certificate' button.

Figure 38: Configure Apple Pay Merchant ID Page.

4. Enter your Admin domain name into the field.

The screenshot shows the Apple Developer console interface, specifically the 'Register' section under 'Certificates, Identifiers & Profiles'. The 'View Identifier' link is visible. The 'Register' section has a 'Save' button. Below this, a message states: 'To use Apple Pay Payment Processing on the web you must register and verify your fully qualified domain name. To begin enter your domain below.' The 'Enter the domain you wish to register' section has a text input field with 'https://' followed by a blue box for the domain name.

Figure 39: Domain Registration Page.

5. Click Save button.
6. Download the validation file and save on your disk.

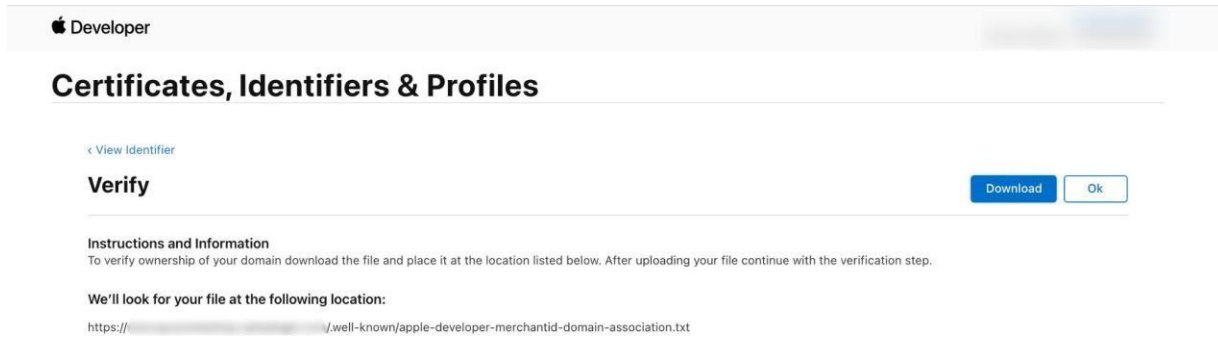


Figure 40: Domain Verification Page

7. The [Upload an Apple Pay merchant identity certificate](#) guide can be used in order to setup Apple Pay for testing in the sandbox environment. `yarn occ` CLI tool has a command for your convenience which can upload domain association file for you:`

```
yarn occ upload-apple-domain-association -u ${OCC_ADMIN_HOST} -k  
${APPLICATION_KEY} <appledeveloper-merchantid-domain-association>
```

Where

- ``APPLICATION_KEY`` - Application Key created in Settings -> Web APIs -> Registered Applications □ ``OCC_ADMIN_HOST`` - your OCC specific environment, e.g. ``asbx80c1dev-admin-{env}.oraclecloud.com``
 - ``apple-developer-merchantid-domain-association`` - domain association file downloaded from ApplePay dev account
8. When you verify domain make sure it is accessible from ApplePay network. OCC Admin (e.g. ``https:// asbx80c1dev-admin-{env}.oraclecloud.com/thirdparty/.well-known/apple-developer-merchantiddomain-association``) is often protected by basic authentication which might fail the domain verification process. You can use [updateBasicAuthConfiguration](#) to update your storefront access control settings by removing basic authentication or whitelisting [Apple Pay IP Range](#).
 9. Click the Ok button to finish the Domain Validation process.

8.4. Create Merchant Identity Certificate

Merchant Identity certificate is used to authenticate your sessions on Apple Pay servers. The certificate and its private key files must be uploaded to your server and full paths to these files must be entered in Cybersource Official add-on settings.

1. Go to [Certificates, Identifiers & Profiles](#) page on your Apple Developer portal.
2. Select your Merchant ID from the list.

3. Generate your CSR following [Apple Developer Help article](#).
4. Click Create Certificate under Apple Pay Merchant Identity Certificate.

Apple Pay Payment Processing on the Web

To configure Apple Pay Payment Processing on the web for this merchant ID, you must register and verify the domains that will process transactions. You must also create a Apple Pay Merchant Identity, which authenticates your web sessions with the Apple Pay Payment Processing servers.

Incorporation of Apple Pay Payment Processing into your website is subject to these [Apple Pay Payment Processing Web Merchant Terms and Conditions](#) and [Acceptable Use Guidelines](#). Failure to comply with any of these Terms and Conditions or guidelines may result in deactivation of Apple Pay Payment Processing transactions on your website.

Merchant Domains

Domain: [www.qa.prestashop.cybsplugin.com](#)

Status: **Verified**

Verification Expires: Aug 3, 2021

[Remove](#)[Verify](#)

Add a domain for use with this Merchant ID.

[Add Domain](#)

Apple Pay Merchant Identity Certificate

Create an Apple Pay Merchant Identity Certificate for this Merchant ID.

[Create Certificate](#)

Figure 41: Create Merchant Identity Certificate Page.

5. Upload the CSR and click Continue.

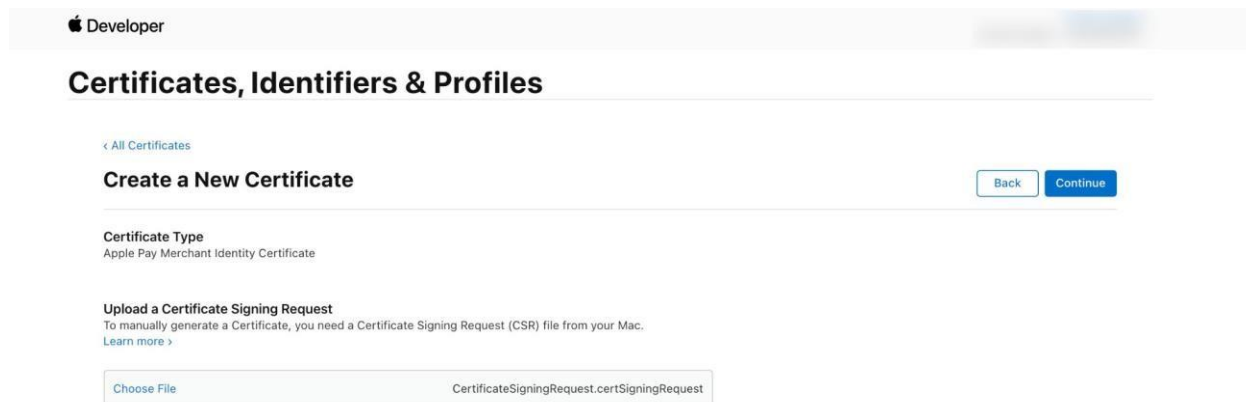


Figure 42: Create a New Merchant Identity Certificate Page.

6. Click the Download button and save the Certificate on your disk.

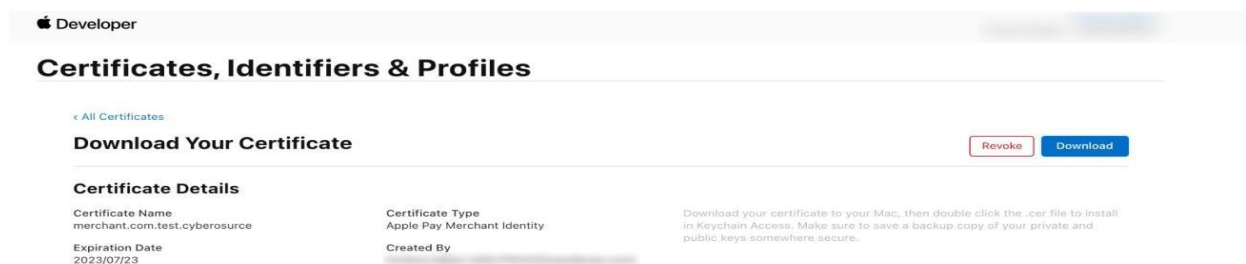


Figure 43: Download Merchant Identity Certificate Page.

- Convert the Certificate into PEM format.

```
openssl x509 -inform der -in merchant_id.cer -out merchant_id.pem
```

- Install the certificate by double clicking on it. Certificate will be available under Keychain Access.

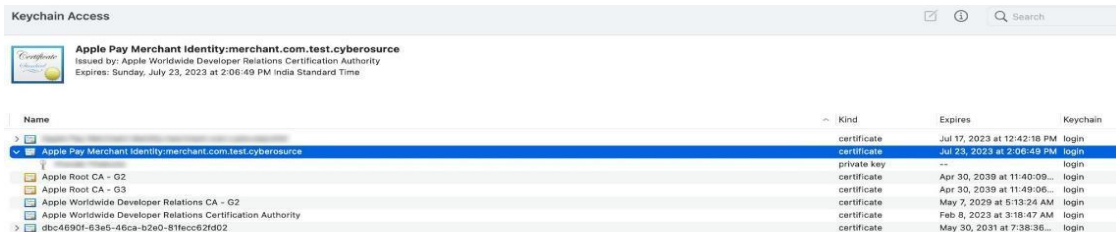


Figure 44: Certificates Under Keychain Access.

- Right click on the private key file and export as merchant_id.p12
- Convert your private key into KEY format.

```
openssl pkcs12 -in merchant_id.p12 -out merchant_id.key -nodes
```

- Identity certificate file is located in packages/server-extension/certs/applePayIdentityCert.pem. Private key file is located in packages/server-extension/certs/applePayIdentityKey.key. **Please make sure you update the file with identity certificate downloaded from your Apple dev account.**

The screenshot shows the 'Commerce' section of the ISV Gateway Payment Processing Page. It contains several required configuration fields for Apple Pay:

- Apple Pay Merchant Id (required):**
- Apple Pay Initiative (required):** A predefined value that identifies the e-commerce application making the request. For Apple Pay on the web use 'web'
- Apple Pay Initiative Context (required):** Fully qualified domain name associated with your Apple Pay Merchant Identity Certificate
- Apple Pay Display Name (required):** Name to be displayed on Apple Pay payment sheet
- Apple Pay Supported Networks (required):** Comma separated list of networks. eg: 'visa,masterCard,amex,discover'

Figure 45: ISV OCC Gateway Payment Processing Page.

9. Network Tokenization

A Network Token is a network scheme generated token, that represents customer card information for secure transactions that references a customer’s actual PAN.

Before a MID can be enabled for Network Tokenization, it must be provisioned with a Token Requestor ID (TRID) for each card scheme.

Plug-in would need to subscribe to the necessary webhook notifications and ingest them for changes to the card. Subscription is created automatically when Authorization is processed, while the Webhook Subscription feature is enabled in the Backoffice configuration.

The following Token updates are processed in Oracle Commerce Cloud:

- Expiry month
- Expiry year
- Card suffix

This section covers information on how to enable a Network Tokenization Service. Under Payment Gateways -> “ISV OCC Gateway”, select the Network Token Updates checkbox to enable the Network Tokenization. Save the changes.

The screenshot shows the 'Commerce' configuration page for the 'ISV OCC Gateway'. The left sidebar contains navigation icons. The main content area has the following settings:

- Key File Name (in case authentication type = jwt) (required):** A text input field containing 'keyFileName'.
- Authentication Type (required):** A dropdown menu with 'http_signature' selected.
- Environment (required):** A text input field containing 'environment'.
- Credit Card Payer Authentication Enabled:** An unchecked checkbox. A tooltip indicates: 'Indicates whether Payer Authentication (3D Secure) will be enabled'.
- Enforce SCA for Saving Card:** An unchecked checkbox. A tooltip indicates: 'If enabled, card holder will be 3DS challenged when saving a card (enforcing Strong Customer Authentication)'.
- Network Token Updates:** A checked checkbox. A tooltip indicates: 'Subscribe to Network Token life cycle updates'.
- Sale Enabled:** A section header for the following checkbox.
- Card:** An unchecked checkbox. A tooltip indicates: 'Indicates if authorizing and taking payment will be done at the same time for a particular payment mode'.

Figure 46: Enabling Network Tokenization

10. Support

If you require support with this software, please contact **GlobalPartnerSolutionsCS@visa.com** and provide the following details:

- Summary of the issue
- Steps to reproduce the issue
- Oracle Commerce Cloud Platform version: You can find Oracle Commerce Cloud Platform Version in Oracle Commerce Cloud Backoffice dashboard.

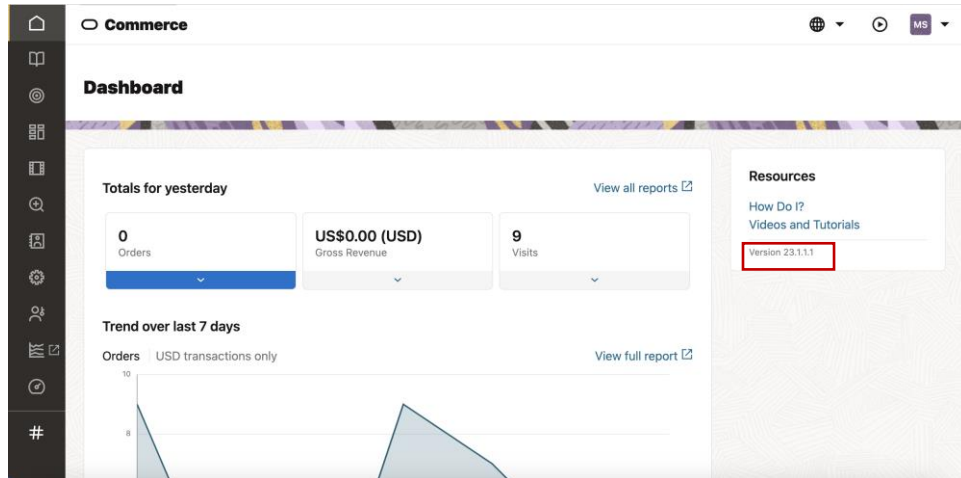


Figure 47: ISV OCC Gateway Payment Processing Page.

- Plugin/ Extension version: Under Settings->Extension, find the version of the installed ISV Payment Gateway extension.

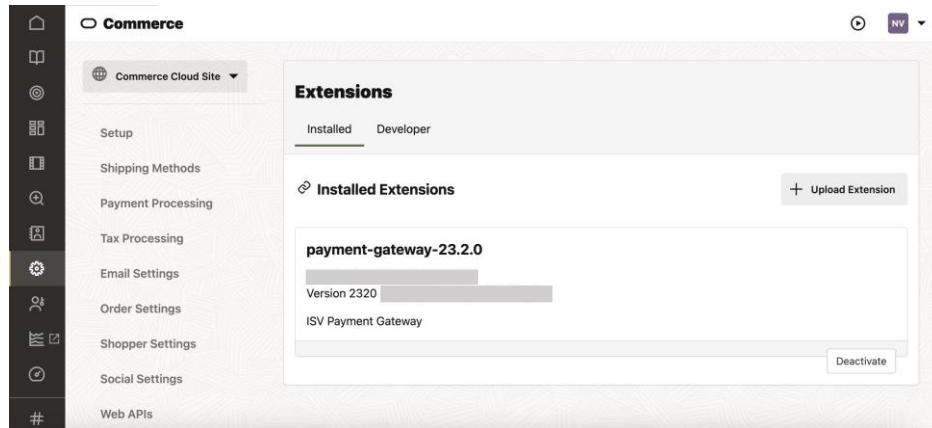


Figure 48: ISV OCC Gateway Extension Page.

- Cybersource Merchant ID: Under Settings->Payment Processing, Select the ISV OCC Payment from the Service Type dropdown, find the Merchant Id.

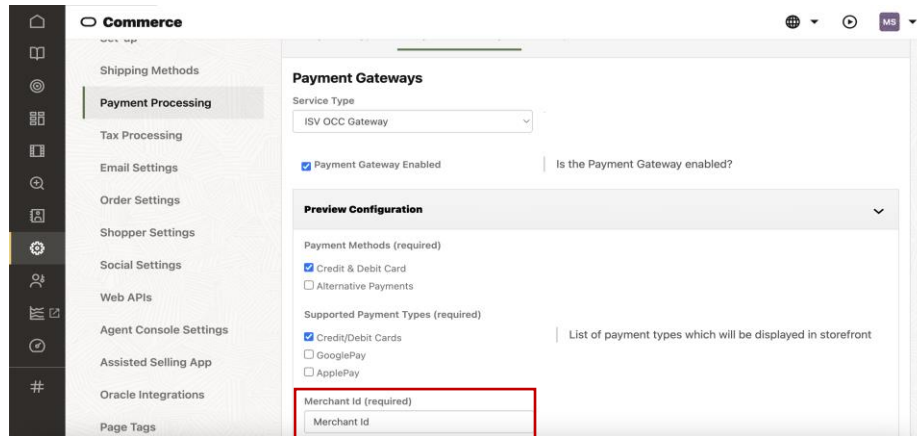


Figure 49: ISV OCC Gateway Payment Processing Page.

- Order ID/ Merchant Reference Number: Order ID can be found in Order Confirmation Page or Order History.
- Configuration screenshots: Please provide screenshots of ISV OCC Gateway Configurations.
- Log file and other relevant data: Download the **debug** and **error** logs from Oracle Commerce Cloud using getExtensionServerLogs admin endpoint.

11. Abbreviations

This section covers the full form of all the abbreviations used in the document.

Abbreviation	Full Form
MID	Merchant ID
EBC	Enterprise Business Center
Org ID	Organization ID
API	Application Programming Interface
PCI	Payment Card Industry

Table 2: Abbreviations