

TECHNICAL CARTRIDGE DOCUMENTATION

ScalaPay

LINK INTEGRATION (SFRA)

VERSION 22.9

TABLE OF CONTENTS

1. SUMMARY	3
2. COMPONENT OVERVIEW	4
2.1 Functional Overview	4
2.2 Use Cases	4
2.3 Limitations, Constraints	5
2.4 Compatibility	5
3. IMPLEMENTATION GUIDE	5
3.1 Set-Up	5
3.2 Configuration	5
3.2.1 Import Metadata	5
3.2.2 Site Preferences	6
3.3 Services	6
3.4 Payment Method Setup	7
3.5 Custom Code	7
3.6 Custom Metadata	9
4. INTEGRATION REFERENCES	10

1. SUMMARY

This document provides the guidelines for the implementation of the SCALAPAY StoreFront Reference Architecture (SFRA) enabled cartridge to Salesforce Commerce Cloud (SFCC). This document describes the various components: Scalapay Payment Method, Scalapay Widget.

2. COMPONENT OVERVIEW

2.1 Functional Overview

This cartridge extends store payment methods with ScalaPay payment method and payment method related functionality.


2.2 Use Cases

Scalapay widgets & payment options availability


Open SFCC instance BM(Business manager) - got to Merchant Tools > Site preferences -> Custom Preferences -> Find and open Scalapay custom preferences groups -> Enable/Disable scalapay widget per page or globally -> open the storefront related page/s(PDP, Cart, Payment) to verify the changes.

Expected result: Scalapay widget and payment options should be displayed if the feature/s configured with enabled status and not displayed or active otherwise.

Scalapay widget (Product Page):

Go to Product Detail Page(PDP), select product variation(color/size), find displayed Scalapay widgets below the product pricing and/or(depends on each widget availability) below the “add to cart” button - find and click on “” OR Scalapay logo. Expected: Dialog window with ScalaPay content should be opened, displayed content is localised according to customer locale(if locale is supported by the ScalaPay).

Scalapay widget (Cart Page):

Go to Product Detail Page(PDP), select product variation(color/size) and add product to shopping cart. Open shopping cart page - find displayed Scalapay widgets below the cart totals pricing - find and click on “” OR Scalapay logo. Expected: Dialog window with ScalaPay content should be opened, displayed content is localised according to customer locale(if locale is supported by ScalaPay).

Scalapay Payment Method: Open site storefront, go to any product details page, add product to shopping cart and proceed to checkout. On checkout payment page find and select Scalapay payment method, check if new payment options are present and selectable(pay in 3, pay in 4 and pay later) - select the payment option - place an Order with Scalapay payment method. Expected result: Order was placed, order payment method is Scalapay, Order contains all required address and products data.

Disable Scalapay for Products in Category: Login to Business manager, go to: Merchant Tools -> Products and Catalogs -> Catalog -> find and open storefront catalog -> find and open category settings. Find and open tab “Category Attributes” -> find checkbox with name “Disable ScalaPay displaying for products in category:” -> check the checkbox. Expected result: ScalaPay widget and text should not be displayed for products where current category selected as “primary category”

Enable/Disable Checkout Summary page: Login to Business Manager, go to: Merchant Tools -> Site Preferences -> Custom Preferences -> Find and open custom preference group with ID scalapay_general -> find field “Enable Summary Page” -> set to “Yes” to enable summary page or “No” to disable it -> apply modified settings

Enable/Disable Payment Capture Delay: Login to Business Manager, go to: Merchant Tools -> Site Preferences -> Custom Preferences -> Find and open custom preference group with ID scalapay_general -> find field “Delay Capture Enabled” -> set to “Yes” to enable payment capture delay or “No” to use immediate payment capture.

2.3 Limitations, Constraints

1. To support ScalaPay payment method - SFCC instance should be configured with EUR currency, since ScalaPay supports this currency only.
2. Product Sets and Product Bundles are not supported in scope of ScalaPay integration cartridge.

2.4 Compatibility

Built on Commerce Cloud 22.9 (Compatibility Mode 21.2) and Storefront Reference Architecture 4.1.0.

3. IMPLEMENTATION GUIDE

3.1 Set-Up

The following ScalaPay Integration tasks are included within the LINK Cartridge:

1. Installation of the cartridge
2. Import custom metadata
3. Set the newly-created metadata values (Site Preferences)
4. Configure services
5. Make storefront cartridge edits

3.2 Configuration

1. Import the int_scalapay cartridge into the Commerce Cloud Studio Workspace:
 - a. Open Commerce Cloud Studio
 - b. Go to: File -> Import -> General -> Existing Projects Into Workspace
 - c. Browse to the "int_scalapay" directory
 - d. Click Finish.
 - e. Click OK when prompted to link the cartridge to the sandbox
2. Assign cartridge to the site
 - a. Log into Business Manager
 - b. Go to: Administration -> Sites -> Manage Sites -> RefArch or required site
 - c. Select the "Settings" tab
 - d. Add "int_scalapay" to the beginning of the cartridge path (or ensure it is in front of app_storefront_base
 - e. Click Apply

3.2.1 Import Metadata

1. Metadata setup:
 - a. Within integration cartridge source folder find and ZIP "metadata" folder
 - b. Log in to Business Manager
 - c. Go to: Administration -> Site Development -> Site Import & Export
 - d. Using file selector form - select and upload archived "metadata" folder
 - e. After metadata archive uploaded -> select and import uploaded archive using file radio-button select and "import" button.

Note: Metadata package contains pre-configured ScalaPay payment method, payment processor, site preferences and services.

3.2.2 Site Preferences

1. Site Preferences Configuration:
 - a. Log into Business Manager
 - b. Go to: Merchant Tools -> Site preferences -> Custom Preferences
 - c. Find and open each of custom site preference group: “scalapay_general”, “scalapay_cart”, “scalapay_product”
 - d. Configure ScalaPay functionality using the following settings:
 - Scalapay Enabled** - set Yes/No to enable/disable Scalapay functionality(payment method and widgets displaying)
 - Scalapay Test Payment Authorization Token(required)** - Scalapay test Payment API Authorization token, used to authorize and capture the payment.
Used only if Scalapay Payment Interface set to Test.
 - Scalapay Production Payment Authorization Token(required)** - Scalapay production Payment API Authorization token, used to authorize and capture the payment.
Used only if Scalapay Payment Interface set to Production.
 - Scalapay Applicable Countries List(required)** - List of site country codes applicable for Scalapay.
 - Scalapay Applicable Currencies(required)** - List of Scalapay applicable currency codes.
Default applicable currency code: EUR
 - Scalapay Widget API URL(required)** - Scalapay widget API URL.
Default URL to add:
<https://cdn.scalapay.com/widget/v3/js/scalapay-widget.esm.js>
 - Scalapay Widget URL(required)** - Scalapay widget URL.
Default URL to add:
<https://cdn.scalapay.com/widget/v3/js/scalapay-widget.js>
 - Enable Summary Page** - Enable/Disable Checkout Summary page for scalapay payment method
 - Delay Capture Enabled** - Enable/Disable payment capture delay
 - Scalapay Payment Interface** - Select Scalapay payment interface type, Test/Production based on where the functionality should be used.
 - Price Selectors(scalapay_cart & scalapay_product groups)** - set the price selector in JS format for each of the scopes(PDP, Cart).
Cart Page Example: [".cart-page .totals .grand-total"]
Product Page Example: [".prices-add-to-cart-actions .price .sales .value", ".prices-add-to-cart-actions .price .list .value"]
 - Pay in Three, Pay in Four, Pay Later minimum/maximum value** - set the minimum and the maximum values for each of the payment option to enable/disable widget and the payment option displaying based on Product price or shopping cart totals.
- Note:** Attached metadata site preferences file can be used as a configuration reference

3.3 Services

1. Import integration cartridge metadata
 2. Set up ScalaPay services configuration
 - a. Login to Business manager
 - b. Go to: Administration -> Operations -> Services
 - c. Find and open tab “Credentials”
 - d. Find and open service credentials with name “scalapay.payment.authorize.credentials”
 - e. Set ScalaPay service URL(staging endpoint URL set by default)
 - f. Apply changes
 - g. Perform the same actions for payment service credentials with name “scalapay.payment.handle.credentials” (staging endpoint URL set by default) and “scalapay.payment.authorize.delay.credentials”
- Note:** username and password for both credentials not required and should be empty

3.4 Payment Method Setup

1. Login to SFCC BM
2. Go to: Merchant Tools -> Ordering -> Payment Processors
3. Create new payment processor with ID "SCALAPAY"
4. Go to: Merchant Tools -> Ordering -> Payment Methods
5. Create new payment method with the following settings:
Name: Scalapay
ID: SCALAPAY
Payment processor: SCALAPAY

3.5 Custom Code

Use customization files list provided below to customize site storefront cartridge with ScalaPay functionality.

1. File: cartridges/int_scalapay/cartridge/client/default/scss/checkout/checkout.scss
Changes: line 268, added the following code:
.scalapay-option-radio { float: left; margin-top: 5px; height: 15px; margin-right: 10px; width: 15px; }
2. File: cartridges/int_scalapay/cartridge/client/default/scss/product/detail.scss
Changes: line 196, added the following code:
.scalapay-widget-holder { margin-top: 15px; }
3. File: cartridges/int_scalapay/cartridge/templates/default/cart/cart.isml
Changes: line 6, added the following code:
<isinclude template="scalapay/widgetapi" />
4. File: cartridges/int_scalapay/cartridge/templates/default/cart/cartTotals.isml
Changes: line 49, added the following code:
<iscomment> ===== cart scalapay widget =====</iscomment> <isinclude url="{URLUtils.https('Scalapay-ShowWidget', 'wtype', 'pit', 'type', 'cart')}" /> <isinclude url="{URLUtils.https('Scalapay-ShowWidget', 'wtype', 'pif', 'type', 'cart')}" /> <isinclude url="{URLUtils.https('Scalapay-ShowWidget', 'wtype', 'pl', 'type', 'cart')}" /> <iscomment> ===== end cart scalapay widget =====</iscomment>
5. File: cartridges/int_scalapay/cartridge/templates/default/checkout/billing/paymentOptions/paymentOptionsContent.isml
Changes: line 5, added the following code:
<isif condition="{paymentOption.ID == 'SCALAPAY' && scalapayPaymentSettings.isEligible}" > <isinclude template="checkout/billing/paymentOptions/scalapayContentGeneral" /> </isif>
6. File: cartridges/int_scalapay/cartridge/templates/default/checkout/billing/paymentOptions/paymentOptionsSummary.isml
Changes: line 6, added the following code:
<isif condition="{payment.paymentMethod == 'SCALAPAY'}"> <isinclude template="checkout/billing/paymentOptions/scalapaySummary" /> </isif>
7. File: cartridges/int_scalapay/cartridge/templates/default/checkout/billing/paymentOptions/paymentOptionsTabs.isml
Changes: line 8, added the following code:
<isif condition="{paymentOption.ID == 'SCALAPAY' && scalapayPaymentSettings.isEligible}" > <isinclude template="checkout/billing/paymentOptions/scalapayTabGeneral" /> </isif>

8. File: cartridges/int_scalapay/cartridge/templates/default/checkout/billing/paymentOptions.isml
Changes: line 1, added the following code:

```
<isset name="scalapayHelper" value="{require('*/cartridge/scripts/helpers/scalapayHelper')}" scope="page"/> <isset name="scalapayPaymentSettings" value="{scalapayHelper.getScalaPayBillingSettings()}" scope="page"/>
```

9. File: cartridges/int_scalapay/cartridge/templates/default/product/productDetails.isml
Changes: line 121, added the following code:

```
<isif condition="{!isQuickView && !isProductSet}" >
<iscomment> ===== pay in 3 and pay in 4 product scalapay widgets
===== </iscomment>
template="scalapay/widgetapi" />
{URLUtils.https('Scalapay-ShowWidget', 'wtype', 'pit', 'type', 'product', 'pid',
product.id)}/>
<iscomment> ===== end pay in 3 and pay in 4 product scalapay widgets
===== </iscomment>
```

Changes: line 134, added the following code:

```
<isif condition="{!isQuickView && !isProductSet}" >
===== pay later product scalapay widget ===== </iscomment>
<isinclude url="{URLUtils.https('Scalapay-ShowWidget', 'wtype', 'pl', 'type', 'product',
'pid', product.id)}/>
<iscomment> ===== end pay later product
scalapay widget ===== </iscomment>
```

10. File: cartridges/int_scalapay/cartridge/templates/default/checkout/checkout.isml
Changes: line 48 after payment page errors block, added the following code:

```
<isif condition="{request.httpParameterMap.isParameterSubmitted('errorMessage') &&
request.httpParameterMap.errorMessage}" >
<div class="alert alert-danger" role="alert">
<p class="error-message-text">
<isprint value="{request.httpParameterMap.errorMessage}"
encoding="off" />
</p>
</div>
</isif>
```


3.6 Custom Metadata

List of customized system object types:

1. Type: **Site Preferences**

Custom Attributes Added:

scalaPayEnabled
scalaPayCountries
scalaPayCurrencies
scalapayEnableSummaryPage
scalapayDelayCaptureEnabled
scalapayWidgetAPIURL
scalapayWidgetURL
scalapayPiTMinValue
scalapayPiTMaxValue
scalapayPiFMinValue
scalapayPiFMaxValue
scalapayPLMinValue
scalapayPLMaxValue
scalapayEnablePiT
scalapayEnablePiF
scalapayEnablePL
scalapayCurrencyDisplay
scalapayCurrencyPosition
scalapayTheme
scalapayPiTFFrequency
scalapayPLFrequency
scalapayPiTInstallments
scalapayPiFInstallments
scalapayPLInstallments
scalaPayPaymentTokenTest
scalaPayPaymentTokenProd
scalaPayPaymentInterface
scalapayCartPiTEnabled
scalapayCartPiFEnabled
scalapayCartPLEnabled
scalapayCartPiTHidePrice
scalapayCartPiFHidePrice
scalapayCartPiTLogoSize
scalapayCartPiFLogoSize
scalaPayCartPriceSelector
scalapayProductPiTEnabled
scalapayProductPiFEnabled
scalapayProductPLEnabled
scalapayProductPiTHidePrice
scalapayProductPiFHidePrice
scalapayProductPiTLogoSize
scalaPayProductPriceSelector
scalapayProductPiFLogoSize

2. Type: **PaymentTransaction**

Custom Attributes Added:

- a. authorizedAmount
- b. authAddressData

3. Type: **Category**

Custom Attributes Added:

- a. scalaPayDisabled

4. INTEGRATION REFERENCES

Official ScalaPay API documentation can be found with links provided below:

Widget API: https://developers.scalapay.com/docs/widget_scalapay

Test Cards: <https://developers.scalapay.com/docs/test-cards>

Payment API: <https://developers.scalapay.com/reference/get-started-with-our-apis>

Capture Delay API: <https://documenter.getpostman.com/view/5023493/TVKFzwGq>