

Implementation Guide - Saferpay Payment Solution for SAP Commerce Cloud

Table of contents

1	Introduction	3
2	Functional Architecture	4
3	Technical Overview	5
4	Technologies and Tools	6
5	Plugin requirements Scope	7
5.1	Payment Page.....	7
5.2	Transaction Interface (Only for Business license)	7
5.3	Refund & Cancellation	8
5.4	3D Secure & DCC.....	8
6	Additional Features considered	9
7	Implementation Guide	10
7.1	Prerequisite	10
7.2	Installation and usage	10
7.3	Configuring the properties	11

Index of Figures

Figure 2.1: Functional Diagram	4
--------------------------------------	---

Index of Tables

Table 1: Application Software	6
Table 2: Development Tools.....	6

1 Introduction

The purpose of providing an extension for Saferpay Payment Solution for SAP Commerce Cloud is to aid the integration of Saferpay into a Hybris implementation. Hybris Store accelerator will use the newly developed extension to interact with Saferpay. The extension will transform the Hybris cart model to Saferpay data structures. The payment result from Saferpay is captured in Hybris and relevant messages if any are displayed to the storefront.

The Saferpay Payment Solution for SAP Commerce Cloud extension provides a configurable spring component which helps shorten the implementation cycle and reduce system development, testing efforts and maintenance costs for payment integration. The extension package contains most of the backend code needed for integrating Saferpay into the hybris commerce platform. The Merchants can use the Saferpay Payment Solution for SAP Commerce Cloud extension with their hybris project to integrate Saferpay.

This document describes the functionalities and Payment Integration of the Saferpay Payment Solution for SAP Commerce Cloud.

2 Functional Architecture

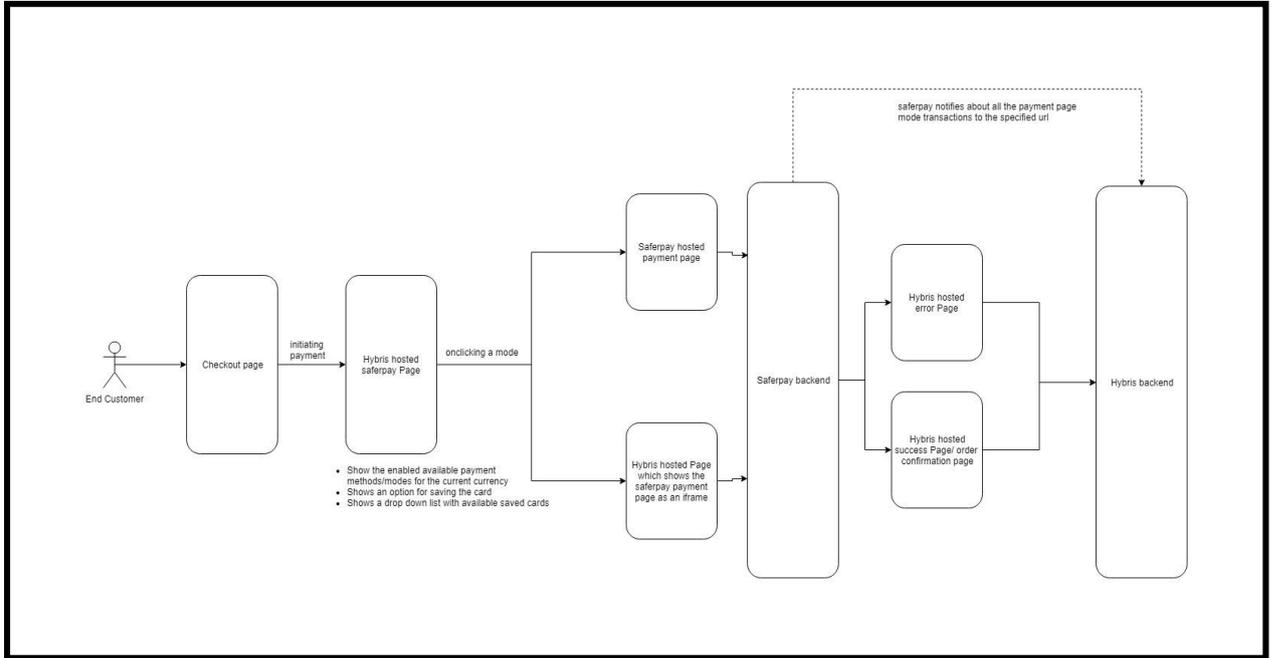


Figure 2.1: Functional Diagram

3 Technical Overview

This extension is built on top of Hybris version 1808. Thus, make sure you have the project running with version 1808 or later.

4 Technologies and Tools

The following table summarizes the proposed platform and software recommended for the system hosting.

Sl.No.	Component	Description
1	Operating Platform	Windows 10
2	Web Server	Tomcat
3	Technology	SAP Hybris

Table 1: Application Software

Sl. No	Component	Description
1	Development Environment	Intellij IDE
2	Source Control	Git

Table 2: Development Tools

5 Plugin requirements Scope

The following functionalities are covered as part of the “Saferpay Payment Solution for SAP Commerce Cloud Plugin”

5.1 Payment Page

The Payment Page Interface provides a simple and easy integration of Saferpay into an e-commerce web shop. The Saferpay Payment Page can be used with a Saferpay e-commerce license as well as with a Saferpay Business license. It allows the processing of all payment methods that are available with Saferpay.

- The process begins with the Payment Page Initialize request. With this request, you forward all data necessary for the payment to Saferpay which generates the RedirectUrl for the Payment Page.
- The Token that is generated from Payment Page Initialize request is cached for subsequent processing of the transaction.
- Redirect to the RedirectUrl.
- Return to returnUrl depending on the outcome of the transaction.
- If the payment capture status is differed, enable manual capture.
- Depending on the information about the payment, finalize or cancel the transaction.
- We have considered support for these payment methods: VISA/Mastercard, Maestro, American Express, Bancontact, Diners/Discover, JCB, Bonus Card, MyOne, SEPA ELV, PayPal, "Przelewy", "eps", iDeal, BillPay Kauf auf Rechnung, Sofort, Twint, Unionpay, Wallet Masterpass.

5.2 Transaction Interface (Only for Business license)

The Transaction Interface is an extension to the Payment Page. Credit card payments can be seamlessly integrated into the merchant's shop with the Transaction Interface.

- The process begins with Transaction Initialize. With this request, you forward all data necessary for the payment to Saferpay which generates the RedirectUrl for the iframe Integration.
- Open the RedirectUrl inside an HTML-iframe, to show the hosted card entry form, which can be used to capture sensitive card details in a PCI-compliant manner.
- Return to Return Url depending on the outcome of the 3D Secure procedure.
- Authorize the card and finalize or cancel the transaction.
- If the payment capture status is differed, enable manual capture.
- We have considered only these payment methods VISA/Mastercard, Maestro, American Express, Bancontact, Diners/Discover, JCB, Bonus Card, MyOne, Wallet: Materpass.

5.3 Refund & Cancellation

Refunds are basically normal transactions where instead of debiting the cardholders account, you are crediting it. That means, that refunds go through the same authorization-steps as a normal (debit) authorization. The refund amount value can differ from the original debited amount. It is also possible to do multiple refunds on one transaction via Payment page & Transaction interface.

- We have considered only these payment methods AmericanExpress, Bancontact, Diners/Discover, JCB, Bonus Card, VISA, Maestro, Mastercard, MyOne, PayPal, "Przelewy", iDeal, Sofort, Twint, Unionpay, Paydirect (COD).

5.4 3D Secure & DCC

- 3-D Secure – 3DS for short is used to make fraud free transactions and liability shift mechanism in transaction. 3DS have the following options.
- Auto capture: All transactions are captured automatically.
- Manual: Transactions with Liability Shift false are not captured, but held inside the transaction overview, so the merchant can capture, or cancel them to his liking. (This should be default!).
- Auto Cancel: Transactions with Liability Shift false are cancelled automatically.

Dynamic Currency Conversion (DCC) is a dynamic currency converter that allows international customers to pay the purchase price in the local currency or their home currency.

6 Additional Features considered

- Possibility to Switch between productive and test systems.
- Pre-selection of payment means in the shop.
- Possibility to choose between direct or delayed capture (adjustable for every mean of payment) depends on payment methods feature.
- Plugin must work on most common browsers by supported the shop-system.
- Error Handling, validation of License, Credentials & Error log etc.
- Currencies: Payment methods can be activated and deactivated per Currency/Currencies can be applied to a given Payment Method. The shop then filters, if a currency is used, that is not configured for a payment method, so the shop does not access this method, if the currency is not available.
- Provide Saferpay full access to place it on its own portals. Can be provided as ZIP file.
- Update saved card expiry date.
- Delete a saved card.

7 Implementation Guide

The purpose of providing an extension for Saferpay is to aid the integration of Saferpay into a Hybris implementation. This document describes how to install and configure the extension to work with any Hybris implementation.

As Hybris is built on the spring framework this makes it highly customizable and extensible. The extension also utilizes this framework, so it can also be easily extended to add specific behavior if required.

7.1 Prerequisite

This extension is built on top of Hybris version 1808. Thus, make sure you have the project running with version 1808 or later.

7.2 Installation and usage

The plugin is supplied as a zip file. Take the following steps to include the steps to your Hybris application.

1. Unzip the supplied zip file.
2. Unzip the supplied zip file. Copy the extracted folder to `${HYBRIS_CUSTOM_DIR}` of your Hybris installation.

```
<extension name='saferpay' />  
<extension name='saferpaycustomaddon' />  
<extension name='saferpaycustombackoffice' />
```

3. Add the Customweb extensions to your *localextensions.xml* file.
4. Install *saferpaycustomaddon* to the storefront in order to make the changes applicable to the storefront.

```
ant addoninstall -Daddonnames="saferpaycustomaddon" -  
DaddonStorefront.yacceleratorstorefront="XXXstorefront"
```

5. Add the Saferpay credentials into project data:
`~\saferpay\resources\impex\projectdata_saferpay.impex.`
6. Run the 'ant cleanupdatesystem' command from your
`_${HYBRIS_PLATFORM_DIR}.`
7. Run the `hybrisserver.sh/bat` to start up the hybris server.
8. Import the project data of the Saferpay extension from `hac`.
9. Or skip the steps from 6-8 and do an initialize.

After initializing to get the component updates, change the `$contentCatalog` value in
`~\saferpaycustomaddon\resources\saferpaycustomaddon\import\contentCatalogs\XXXXXXcontentCatalog \cms-content.impex` file to the catalog versions used in your project.

Note: You can add more accounts in to the system via the backoffice, but that won't be available in all other environments unless created manually to them also.

7.3 Configuring the properties

Saferpay Payment Solution for SAP Commerce Cloud uses some properties in order to manage the process. Those properties can be seen in the `project.properties`. Some of the main properties are listed below:

```
successURL=checkout/multi/summary/saferpay-response/  
failureURL=response/failureResponse/  
transactionSuccessURL=response/successResponse/  
aliasSuccessUrl=my-account/payment-details/success  
aliasFailureUrl=my-account/payment-details/aliasInsert  
notifyURL=https://your_notify_url/  
notifyEmail=  
specVersion=1.18  
clientInfo=client_info  
shopInfo=shop_info  
osInfo=os_info  
productionUrl=https://www.saferpay.com/api  
testUrl=https://test.saferpay.com/api
```

Eg. In our context, it is `ShopInfo:clientInfo = Hybris_1808:PITs_1.18`

Those properties have names that speak for itself. Note: Important point to note is that about the `saferpay.notify.url`. This property holds the url to which the Shop will be notified of a successful transaction, by Saferpay. This URL needs to be available to reach from Saferpay servers.