SAP Commerce Klarna Payments

Add-on

Installation & User Guide

**Table of Contents**

[1. Introduction 3](#_Toc67522238)

[2. Supported in Integration 3](#_Toc67522239)

[3. Add-on Installation steps 6](#_Toc67522240)

[4. Unit Test Implementation 13](#_Toc67522241)

[5. Activating Klarna Payments through Backoffice 13](#_Toc67522242)

[6. Extension Model and Classes 24](#_Toc67522243)

[7. Klarna Payment Method Reference 25](#_Toc67522244)

[8. KlarnapaymentAPI Method Reference 25](#_Toc67522245)

[9. Uninstalling the Add-on 26](#_Toc67522246)

[10. Recommendation 27](#_Toc67522247)

[11. Modifying the order confirmation email 27](#_Toc67522248)

[12. Troubleshooting 30](#_Toc67522249)

[13. Installing the KOSM Add-on 31](#_Toc67522250)

[14. Support from Klarna 38](#_Toc67522251)

# Introduction

This document is a technical document to guide you through the installation process of the Klarna Payments (KP) extension on the SAP Commerce.

* 1. **Audience**

The expected audience for this document is technical specialists who will be involved in KP integration into SAP Commerce. These specialists are expected to be familiar with the SAP Commerce platform and installing extensions/add-ons for it.

* 1. **Supported Environments**

This add-on is designed and tested with version 1905 of SAP Commerce. It supports version 2005 also.

* 1. **Restrictions with this extension**

This version of the extension should be used in stores which use percentage tax rates and not absolute taxes. Note that it may still work as expected in certain circumstances with absolute taxes, in others it will not be possible to place the order.

This extension is designed to be compatible with out of the box SAP Commerce installations. Due to the nature and unlimited scope of potential customisations, we are unable to guarantee that compatibility and full functionality will be maintained on all customised systems. Testing may need to be performed on an individual basis to ensure that correct functionality is still maintained.

Testing has been conducted for the following country, currency, and locale combinations

UK-GBP-en

US-USD-en

DE-EUR-de

# Supported in Integration

The Klarna extension supports Klarna Payment method integration across multiple use-cases out of the box. The extension is designed utilizing the SAP integration patterns offered on the platform and tested against the standard b2c accelerator(s) offered by SAP. The integration extends the base models and classes where required and available for review in the section: Extension Model and classes.

The common checkout experience is enhanced with Klarna payment integration following the best practices mentioned on Klarna’s [developer portal](https://developers.klarna.com/documentation/klarna-payments/).

* 1. **Use-cases supported**

* + 1. **Pre-Purchase**

**Use Case: Display Klarna upstream on the website before payment**

The Klarna integration includes extension to support the configuration and display of On-site messaging placement(s) on product display pages and on cart page. The back-office configuration support enabling default themes or customized theme when configured by the merchant in Klarna [Merchant Portal.](https://www.klarna.com/us/business/the-merchant-portal/) Refer Installing the KOSM Add-on

* + 1. **Order Placement via Klarna Payments**

The Klarna extension integration configuration can be done by storefront to enable a specific payment method by Klarna market. The relevant Klarna configuration is stored by a specific payment config ID which can be attached to the respective Base-store configuration for the relevant storefront. Refer Activating Klarna Payments through Backoffice

**Use Case: Billing Address Collection from Address forms**

The Klarna extension integration utilizes the standard form and extends it for data entered by end-user in SAP Commerce checkout. The form data updates are supported for both billing and shipping address. The address forms have customization to align with required information for respective Klarna markets. Any person identifiable data is sent to Klarna once the customer chooses to authorise and pay with a Klarna payment method. Note that additional data maybe requested in a widget (Klarna modal) as part of Payment method authorization depending on market and the Klarna product.

**Use Case: Update Basket**

The Klarna extension integration enables updates made to the cart to be applied to the Klarna payment session. Updates to cart that trigger a Klarna session update will additionally lead to a re-authorization when customer checkouts with a Klarna payment method. Any additional authorization will return updated authorization token for new cart amount.

**Use Case: Management of Klarna decline - Recoverable**

The extension supports handling Klarna authorization decline gracefully. The payment methods are displayed again and re-direct to payment method selection step when rejected with recoverable error. The customer has the option to update the customer details to re-select Klarna payment methods. E.g. (Response: show\_form: false, approve: true)

**Use Case: Order creation**

On submission from payment Page, Authorization token is received from Klarna using client-side authorization call. Using this authorization token, Order is created at Klarna by calling create order API from SAP Commerce. Klarna Order is created just before Order creation in SAP Commerce and after authorization. Payment transaction entries are created for Klarna Authorization (PaymentTransactionType, AUTHORIZATION), Klarna Order Creation (PaymentTransactionType KLARNA\_ORDER\_PLACED).

**Use Case: Pending Orders**

The extension supports the processing of pending order status when the functionality is enabled. The pending order statuses are returned to the configured merchant notification URL, e.g., FRAUD\_RISK\_ACCEPTED, FRAUD\_RISK\_ACCEPTED, FRAUD\_RISK\_REJECTED. The order fraud status is verified utilizing the Order management API. The order process will be suspended until the application receives pending order update from Klarna. e.g., if notification URL receives an update with status FRAUD\_RISK\_ACCEPTED, then the order processing with resume.

**Use Case: Confirmation Email**

Extension enables sending of order confirmation email with “Paid by Klarna” message in the email body. Additional information regarding Klarna App and customer support links are included for customer to find additional information about the payment process and status. Refer Modifying the order confirmation email

* + 1. **Order Management (VCN)**

**Use Case: Post purchase update - Shipment of Goods / “Capture” / “Card Settlement”**

The extension does not support the capture or cancellation or other use-case post order placement out of the box. These use-cases can be enabled, or the extension extended to support them using the Klarnapaymentapi extension which includes support for Klarna Payment and Order Management API. A Sample Capture implementation is provided as part of accelerator Fulfilment -> TakePaymentAction for Merchant Reference

If standard order management is not a reasonable option for a Klarna integration, then Klarna’s Merchant Card Service based virtual card solution may be utilized. When order is settled via Virtual Card based settlement approach, the merchant needs to work with their Payment Service provider PSP in collaboration with Klarna Merchant success team to ensure that the VCN based settlement post purchase use-cases are handled appropriately.

This option can be enabled via backoffice Klarna configuration attribute “VCNEnabled”. By default, this option is disabled.

When a customer places an order, an order is created and accepted by Klarna, the addon integration creates a virtual card-based settlement, utilizing the merchant card services (MCSv3) API and creates order in Sap Commerce.

Once a settlement has been created (virtual card returned), the merchant platform can authorize the virtual card until the Klarna order is valid. Then, once the order has been fulfilled, the card funds should be captured. (For delays in capture, or other special use cases, please speak with the Klarna Key Account Manager in advance). While Klarna is the original payment method of the order, the order amount will be settled with a virtual card instead of direct bank account transfer.

A virtual card settlement request is triggered for new order or those orders with pending status resolved to FRAUD\_RISK\_ACCEPTED. An order placed with VCN settlement will have VCN specific data persisted in payment info.

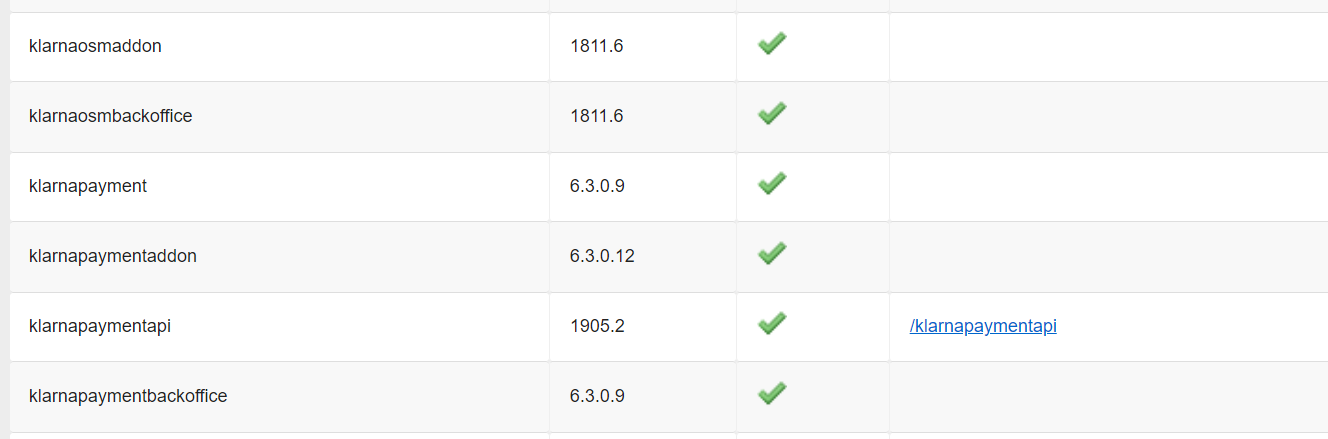
**Note**: If the Klarna order has a “**fraud\_status**” of “PENDING”, action is not taken on the order until receiving Klarna’s push notification that the “**fraud\_status**” has changed to “FRAUD\_RISK\_ACCEPTED”.

The virtual card issued is limited to 1 single successful authorization per order for a given MID

Refer VCN Set UP

# Add-on Installation steps

* 1. **Download Add-on**
* Get the addon archive from Klarna.
* Unpack archive and place folders klarnapaymentaddon, klarnapayment, klarnapaymentapi & klarnapaymentbackoffice into custom folder of SAP Commerce Suite (<HYBRIS\_HOME>/bin/custom).
* For SAP Commerce B2B implementations, copy the folder klarnapaymentb2baddon also to the custom folder.
  1. **Update localextensions.xml**
* Check the presence of <extension name=”addonsupport” /> in the localextensions.xml file.
* Add extension <extension name=” klarnapaymentaddon” />
* Add extension <extension name=” klarnapayment” />
* Add extension <extension name=” klarnapaymentbackoffice” />
* Add extension <extension name=” klarnapaymentapi” />
* For B2B, add extension <extension name=” klarnapaymentb2baddon” />
* Check the presence of the target storefront extension.
* Other OOB extensions required are mentioned in the localextension.xml available in archive.



* 1. **Run installation command**
* Stop SAP Commerce server if already running. Depending on the server start mode we can use one of following method to stop server.
* If the server was started with embedded mode, press Ctrl+C to stop the server.
* If the server was stared with service mode, use this command to stop:

Windows: hybrisserver.bat stop

Unix: ./hybrisserver.sh stop

* Go to <HYBRIS\_HOME>/bin/platform and run the following command if it was not applied in this terminal:

Windows: setantenv.bat

Unix: ./setantenv.sh

* Go to <HYBRIS\_HOME>/bin/platform and run the installation add-on with the command:

ant addoninstall -Daddonnames="klarnapaymentaddon"-DaddonStorefront.yacceleratorstorefront="yacceleratorstorefront"

* For B2B, klarnapaymentb2baddon should be installed to yb2bacceleratorstorefront. Go to <HYBRIS\_HOME>/bin/platform and run the installation add-on with the command:

ant addoninstall -Daddonnames="klarnapaymentb2baddon"-DaddonStorefront.yacceleratorstorefront="yb2bacceleratorstorefront"

**Note**:

* In some versions of SAP Commerce, you may be required to install liveeditaddon and b2ccheckoutaddon.
* Depending on the project, the storefront name may differ from the default storefront name that is provided by SAP Commerce.
  1. **Mandatory configuration changes**

It is highly recommended to use below configuration for support from Klarna

* Configuration for call back [URLs](https://developers.klarna.com/api/#payments-api__create-a-new-credit-session__merchant_urls) in the project.properties file of klarnapayment extension.
* Confirmation URL: The confirmation page url the customers will be finally redirected to

klarnapayment.merchant.url.confirmation==https://<your\_host>/klarna/order/confirmation

* Notification URL: The endpoint for receiving the fraud notifications

klarnapayment.merchant.url.notification=https://<your\_host>/klarna/payment/checkout/pending-update

In SAP Commerce, it is possible to activate KP for multiple sites simultaneously. If all sites are using KP, then the above configuration is to be used. If only certain sites are to use KP, then they must be specified individually with the below configuration:

klarnapayment.merchant.url.confirmation.<site\_uid>==https://<your\_host>/klarna/order/confirmation

klarnapayment.merchant.url.notification.<site\_uid>==https://<your\_host>/klarna/payment/checkout/pending-update

* Configuration for skipping CSRF verification when receiving a call back from Klarna - in the local.properties file.

csrf.allowed.url.patterns=/[^/]+(/[^?]\*)+(sop/response)$,/[^/]+(/[^?]\*)+(merchant\_callback)$,/[^/]+(/[^?]\*)+(hop/response)$,/[^/]+(/[^?]\*)+(language)$,/[^/]+(/[^?]\*)+(currency)$,/[^/]+(/[^?]\*)+(push)$,/[^/]+(/[^?]\*)+(pending-update)$,/[^/]+(/[^?]\*)+(confirmation)$

* Email server configuration for emails to be sent from SAP Commerce, add below properties in the local.properties file.

mail.from=<your\_email>

mail.replyto=<your\_email>

mail.smtp.server=<your\_mail\_server>

mail.smtp.port=<your\_mail\_port>

mail.smtp.user=<your\_email>

mail.smtp.password=<your\_password>

mail.use.tls=true

* User Agent Configurations

Below configuration are highly recommended for ensuring speedy support and issue identification from Klarna for production issues.

shoporplatform=<your\_platform> Eg: SAP\_Comm

platformversion=<your\_platform\_version> Eg: 1905

modulename=<your\_module> Eg: KP

moduleversion=<your\_module\_version> Eg: 7.0

* Currency Conversion

If multiple Currency is applicable in the merchant site, out of box settings for the conversion factor/delivery cost should be added with proper values. Those values are being used for sending the cost in purchase currency converted value to Klarna.

* 1. **Order Failed Notification**

If Klarna order is successful and failed to place order in Hybris, email notification is sent to merchant (email configured in Klarna configurations in backoffice). Email content is configured using \klarnapayment\resources\klarnapayment\email\_impex\FailedOrder\_Email\_Template.impex. This impex should be loaded after server start. Merchants can modify the email content (cms component “OrderFailedEmailBody”) either in impex or through back office later.

* 1. **Mandatory Code changes**

To enable Klarna in the storefront, the following code changes should be made in the storefront extension. For B2B implementations, changes should be made in the corresponding b2b storefront.

1. Add the below tag in the Merchant Payment Jsp e.g. - silentOrderPostPage.jsp in SAP commerce Out of the Box (OOB)

<%@ taglib prefix="klarnapayment" tagdir="/WEB-INF/tags/addons/klarnapaymentaddon/responsive/" %>

<klarnapayment:selectpayment/>

1. In CheckoutOrderSummary tag (SAP Commerce OOB code it is

4yacceleratorstorefront/web/webroot/WEB-INF/tags/responsive/checkout/multi/checkoutOrderSummary.tag)

<%@ taglib prefix="klarnacheckout" tagdir="/WEB-INF/tags/addons/klarnapaymentaddon/responsive/checkout" %>

Modify the paymentinfo tag to

<klarnacheckout:paymentInfo

If the merchant has a customized paymentInfo.tag, include the hidden parameters in the addon paymentinfo.tag to their own customized paymentinfo.tag

1. If Date of birth is required in the billing address section, add the below snippet in kpBillingAddressFormElements.tag. If any Merchant specific field is required in the billing address, it can be added in the same file.

<div class="form-element-icon datepicker date\_of\_birth" id="js-dateofbirth" data-date-format-for-date-picker="${dateFormatForDatePicker}">

<formElement:formInputBox idKey="dateOfBirth" labelKey="address.dateOfBirth" inputCSS="text" mandatory="false" path="dateOfBirth" placeholder="${dateFormatHint}"/>

<i class="glyphicon glyphicon-calendar js-open-datepicker-quote-expiration-time"></i></div>

1. In acc.klarnapayment.js, replace the form name **silentOrderPostForm** to merchant payment page form name if the paymentpageForm name/payment page jsp is different from silentOrderPostPage.jsp.

checkPaymentSelected: function () {

klarnaData={};

$("form input:radio[name='paymentMethod']").click(function () {

var paymentMethod = $('input[type=radio][name=paymentMethod]:checked').val();

if (paymentMethod == 'card') {

$('.submit\_silentOrderPostForm').bind('click');

$('#klarna\_container').empty();

$('#klarna\_billing').hide();

$('#**silentOrderPostForm'**).show();

ACC.silentorderpost.bindSubmitSilentOrderPostForm();

} else {

$('#klarna\_container').empty();

$('.submit\_silentOrderPostForm').unbind('click');

ACC.klarnaPayment.bindPaymentsubmit();

$('#**silentOrderPostForm'**).hide();

$('#klarna\_billing').show();

kpUpdateSessionUrl = $("#updateSessionUrl").val();

getKlarnaFormUrl = $("#getKlarnaFormUrl").val();

paymentOption = paymentMethod;

var clientToken = $("#clientToken").val();

ACC.klarnaPayment.getKlarnaPaymentForm(clientToken);

}

});

}

Eg : Form name in silentOrderPostPage.jsp

<form:form id="silentOrderPostForm" name="silentOrderPostForm" commandName="sopPaymentDetailsForm" action="${paymentFormUrl}" method="POST">

1. In acc.klarnapayment.js, replace the submit button class **submit\_silentOrderPostForm**to merchant payment page button name if the paymentpageForm submit button class name/pagement page jsp is different from silentOrderPostPage.jsp.

bindPaymentsubmit: function () {

$('.submit\_**silentOrderPostForm'**).click(function ()

{

checkPaymentSelected: function () {

klarnaData={};

$("form input:radio[name='paymentMethod']").click(function () {

var paymentMethod = $('input[type=radio][name=paymentMethod]:checked').val();

if (paymentMethod == 'card') {

$('.**submit\_silentOrderPostForm**').bind('click');

$('#klarna\_container').empty();

$('#klarna\_billing').hide();

$('#silentOrderPostForm').show();

ACC.silentorderpost.bindSubmitSilentOrderPostForm();

} else {

$('#klarna\_container').empty();

$('.**submit\_silentOrderPostForm**').unbind('click');

ACC.klarnaPayment.bindPaymentsubmit();

$('#silentOrderPostForm').hide();

$('#klarna\_billing').show();

kpUpdateSessionUrl = $("#updateSessionUrl").val();

getKlarnaFormUrl = $("#getKlarnaFormUrl").val();

paymentOption = paymentMethod;

var clientToken = $("#clientToken").val();

ACC.klarnaPayment.getKlarnaPaymentForm(clientToken);

}

});

}

Eg : submit button class name in silentOrderPostPage.jsp

<button type="button" class="btn btn-primary btn-block submit\_silentOrderPostForm checkout-next"><spring:theme code="checkout.multi.paymentMethod.continue"/></button>

1. Change the constant in KlarnaPaymentActiveFilter to the default checkout url in merchant application

public final static String DEFAULT\_CHECKOUT\_URL = "/checkout/multi/summary/placeOrder";

1. B2B implementations using b2bacceleratoraddon might have a Checkout step for selecting Payment Type. To display Klarna Payment Type, modify paymentTypeForm.tag file by replacing the line of code

*<form:radiobutton path="paymentType" id="PaymentTypeSelection\_${paymentType.code}" value="${paymentType.code}" label="${paymentType.displayName}" />*

With the following code snippet.

<c:choose>

<c:when test="${(paymentType.code eq 'CARD') && ('TRUE' eq isKlarnaActive)}">

<form:radiobutton path="paymentType" id="PaymentTypeSelection\_${paymentType.code}" value="${paymentType.code}" label="Card payment or Klarna Pay" />

</c:when>

<c:otherwise>

<form:radiobutton path="paymentType" id="PaymentTypeSelection\_${paymentType.code}" value="${paymentType.code}" label="${paymentType.displayName}" />

</c:otherwise>

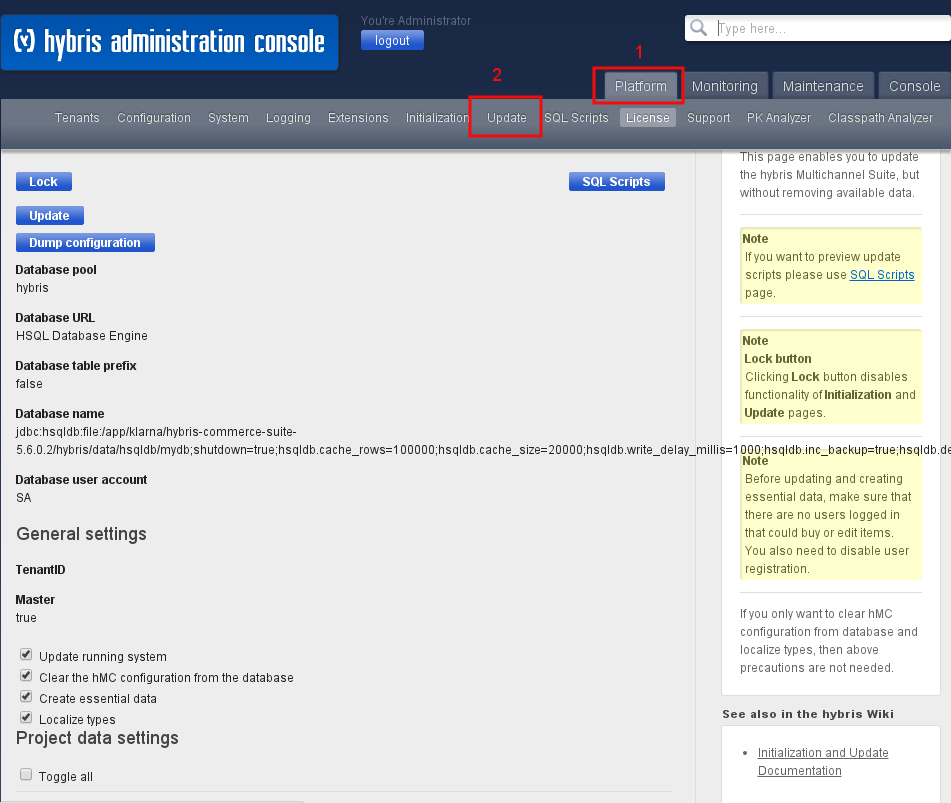
</c:choose>

* 1. **Rebuild the system**
* Go to <HYBRIS\_HOME>/bin/platform and rebuild the system with the command: ant clean all.
  + Update the system

After the previous step you may need to perform full initialization through Hybris Administration Console (HAC) if this is the first installation of SAP Commerce.

If you have already performed full initialization, then you need to update your SAP Commerce system as follows:

Open web browser, go to **HAC → 1 Platform → 2 Update**



* Check the 4 checkboxes under General Settings: Update running system, Clear the HMC configuration from the database(applicable for versions prior to 5.7 only), create essential data, Localize types.
* Check the klarnapaymentaddon, klarnapayment, klarnapaymentapi & klarnapaymentbackoffice checkbox further down under Project data settings.
* Check klarnapaymentaddon checkbox in case of B2B implementations.
* Click on the Update button to update the SAP Commerce system.



# Unit Test Implementation

* Junit

Junit test classes are implemented inside testsrc folder of klarnapayment and klarnapayment extensions. Mock test classes for all supported Klarna API calls are available in klarnapaymentapi extension.

* Integration Test

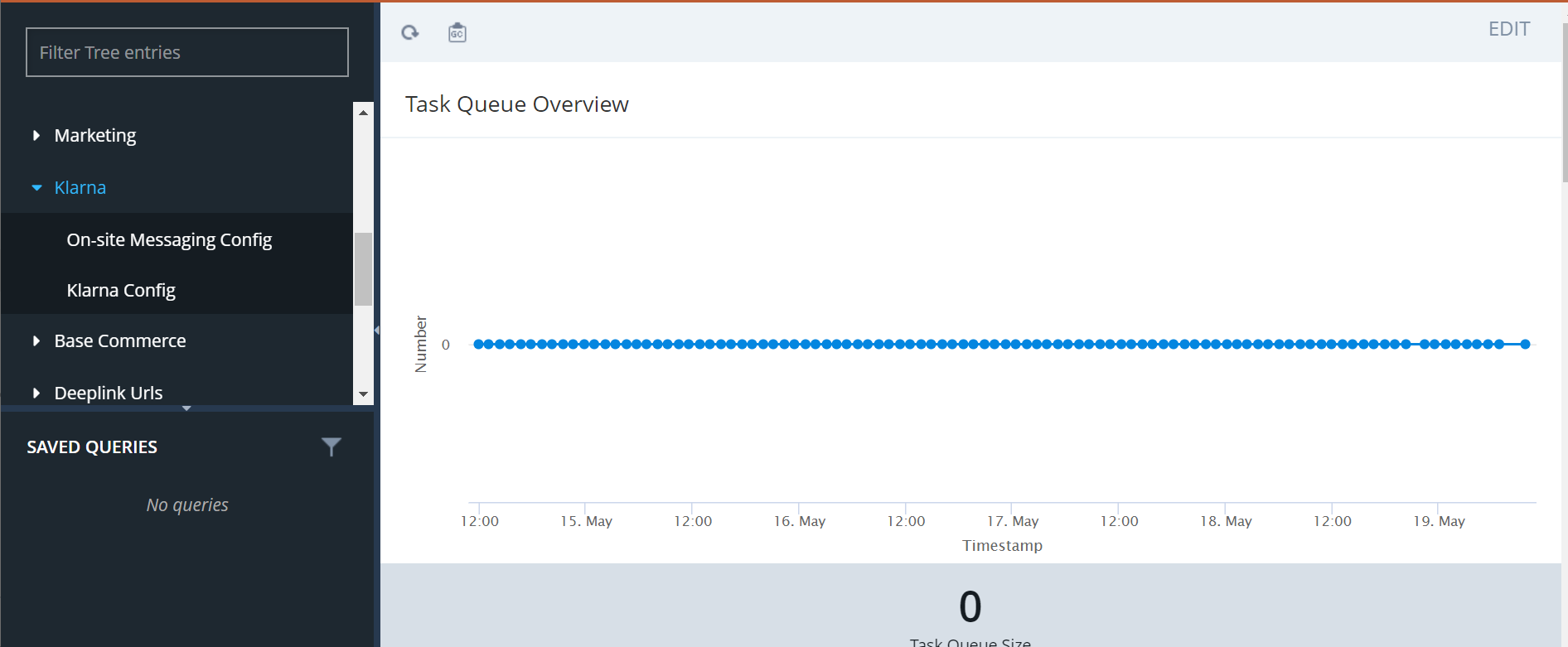
An integration test class, KlarnaClientConnectionTest, is implemented in klarnapaymentapi extension, which can be utilized to test the connection to Klarna server after the initial tests. Update the class with valid merchant test(playground) credentials, to identify issues in authentication or payment session creation during server start up. If SAP commerce can connect successfully to Klarna, server start-up will have “Klarna Integration Connection Test Successful” log generated. If this feature is utilized on production instances, ensure to remove the Klarna production API credentials from the test class.

# Activating Klarna Payments through Backoffice

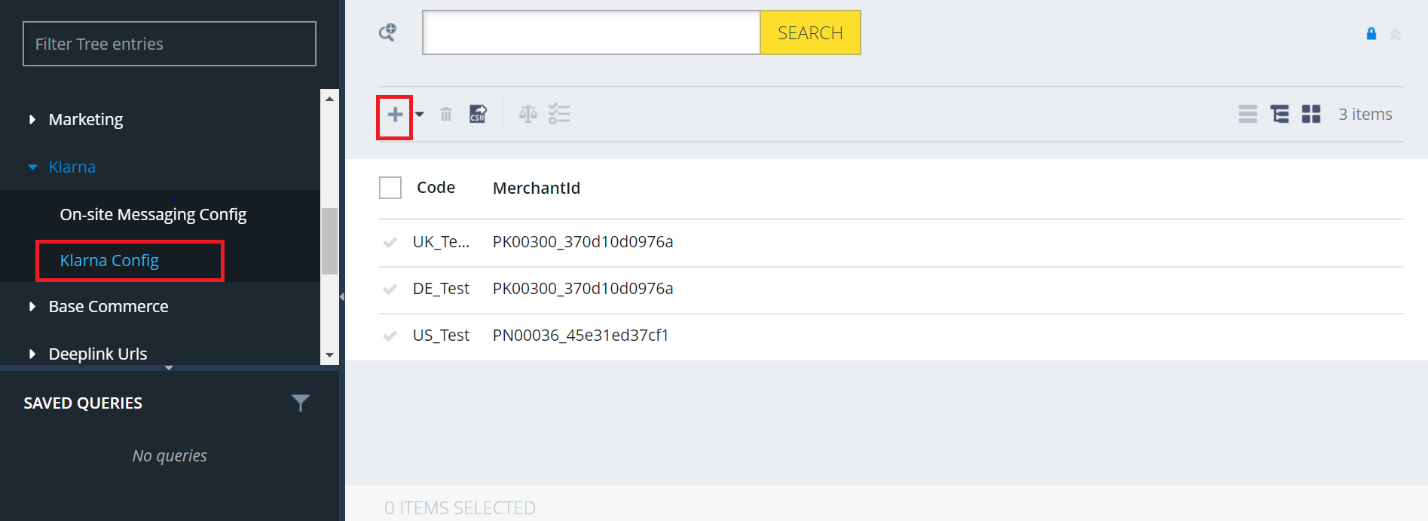
After installing klarnapaymentaddon successfully, you can now login to SAP Commerce Backoffice to setup a Klarna configuration and assign it to specific sites in SAP Commerce.

* 1. **Create Klarna Payments configuration**

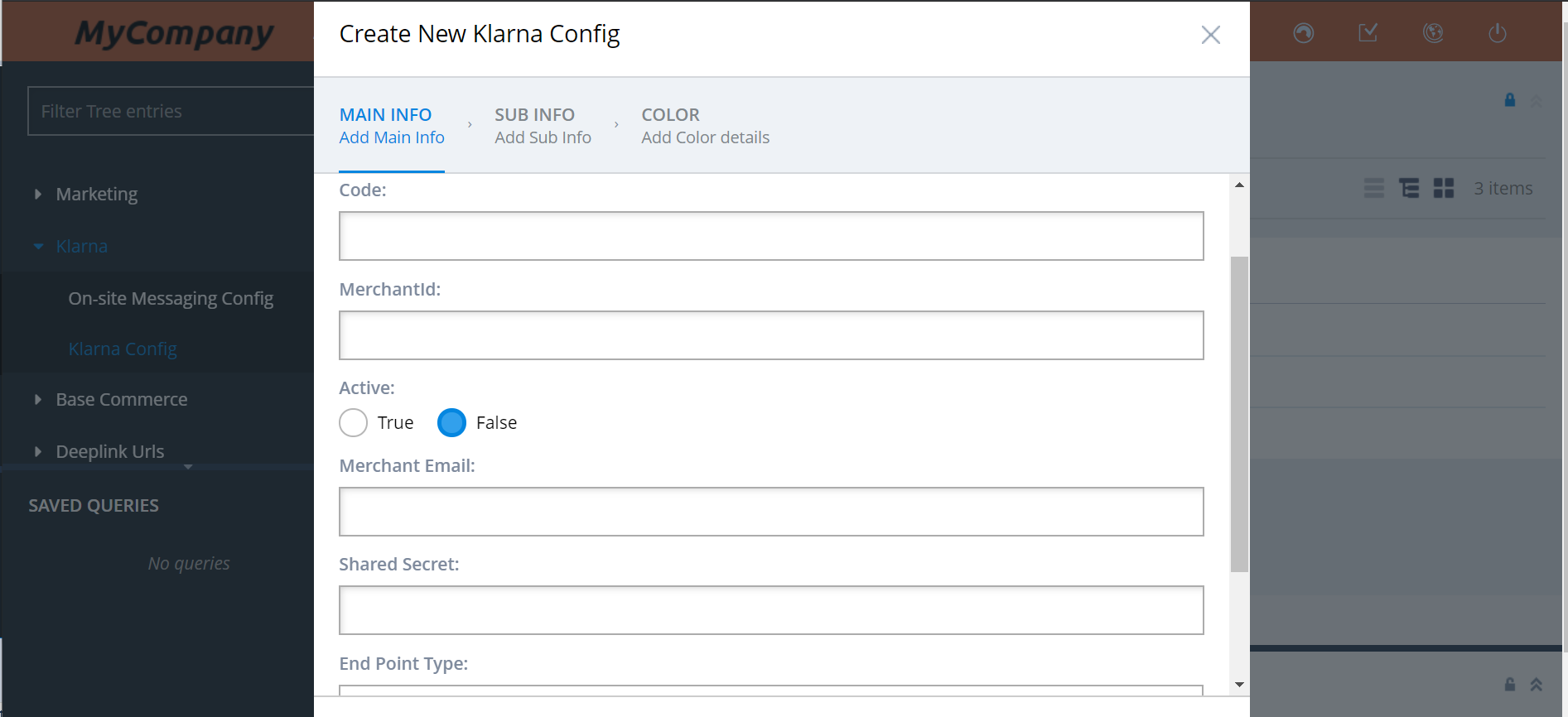
To configure Klarna, go to SAP Commerce-**Backoffice→ Klarna Payments → Klarna Config**



Chooseto create a new Klarna configuration



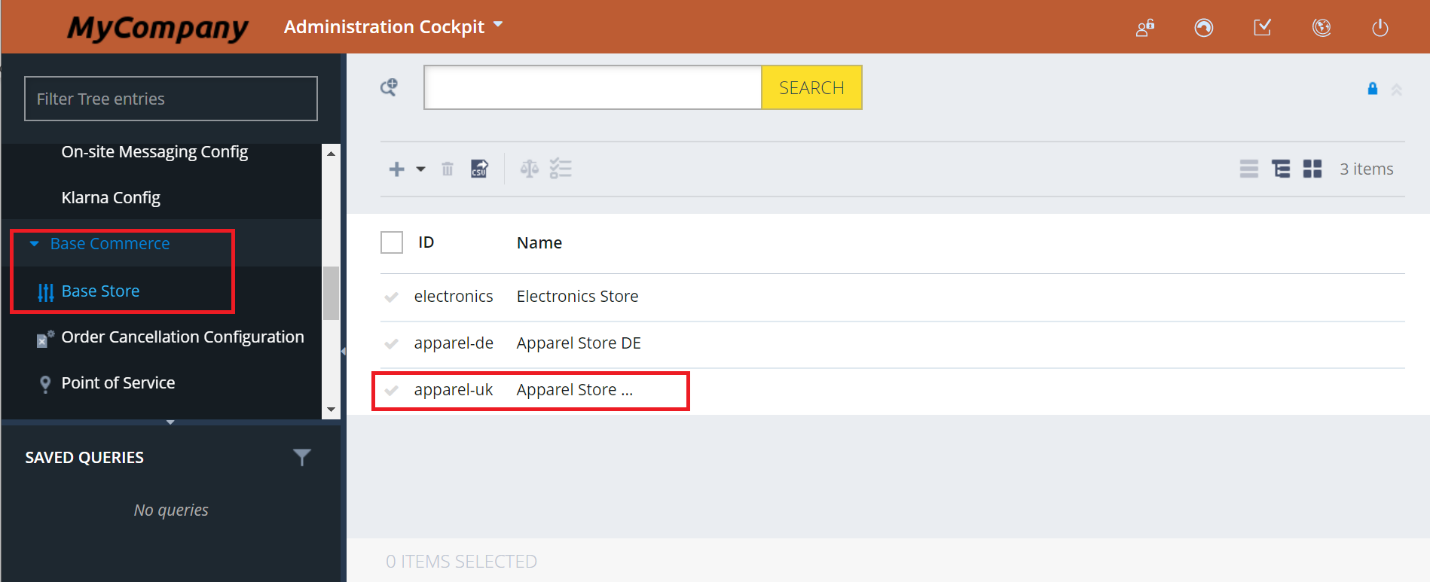
Fill the mandatory and where required optional fields. Fields information is given under Section 5.3.



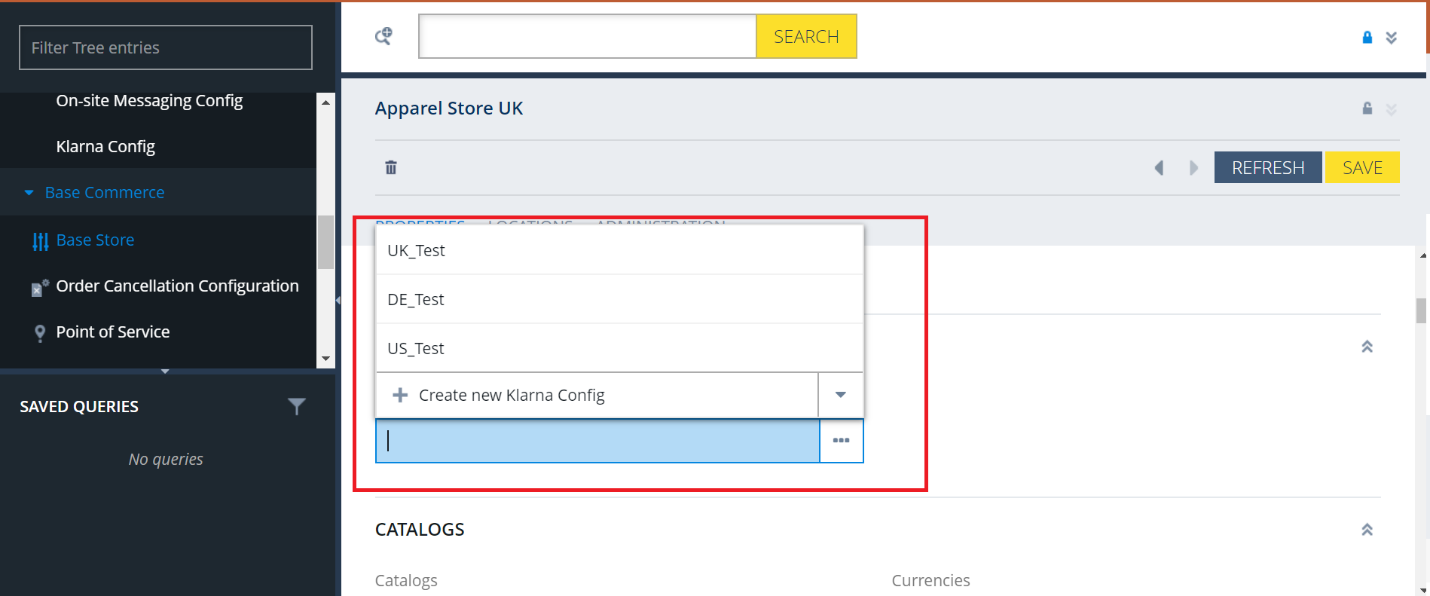
After providing the required data in the above fields for the first time, click on the Next button to provide further information on next page. After inputting all required fields click on Done button to save the changes.

* 1. **Active Klarna Payments config for site**

Go to SAP Commerce Backoffice → Base Commerce → Base Store → Search → double click on the base store which you want to assign the configuration to → Choose Properties tab.



Go to Klarna configuration, then click on search icon & select the configuration to activate Klarna config for the base store.



Result after assigning the selected Klarna config to the base store:

Click on Save.

* 1. **Klarna Configuration Fields**

**Payment Features Required**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Mandatory** | **Description** |
| Code | Yes | Input any name/code. This will be the internal reference used to recognise and manage Klarna configuration within SAP Commerce. |
| Active | No | Check to activate Klarna payments with this configuration; leave unchecked for it to remain inactive. |
| MerchantId (MID)/Klarna API username | Yes | Generated in Klarna Merchant Portal. |
| Shared Secret/Klarna API username | Yes | Generated in Klarna Merchant Portal. |
| End Point Type | Yes | Each MID support one of three end point types: EUROPE, NORTH\_AMERICA and OCEANIA. Select the appropriate one. |
| End Point Mode | Yes | Each MID support one of the two end point modes: TEST (playground) and LIVE (production). Select the appropriate one. |
| PurchaseCurrency | Yes | Choose currency to use with this Klarna configuration. |
| AttachmentRequired |  | Choose this if extended merchant data must be sent for certain type of orders. OOB customer info is sent, must be customised for other use-case (e.g., Ticket sales). |
| ProductUrlsRequired | Yes | Enable to send product and image URLs, highly recommended |
| MerchantReference2 | Yes | Select the merchant reference 2 to be sent to Klarna after Order placement. Merchant reference 1 is the SAP Commerce order ID. |
| AutoCapture | Yes | Allow merchant to trigger auto capturing during Order Placement. If Auto Capture is On, capture should be attempted after SAP Commerce Order creation. Otherwise Capture will be triggered in order process step. In case of VCN enables, AutoCapture setting will not be relevant. Reference for Capture Call in TakePayment Action is given in the package |

**Additional Payment Features**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Mandatory** | **Description** |
| colorBorder | No | Colour of the ‘border’ in the Klarna Payments iframe. |
| colorBorderSelected | No | Colour of the ‘border selected’ in the Klarna Payments iframe. |
| colorText | No | Colour of the ‘Text’ in the Klarna Payments iframe. |
| colorDetails | No | Colour for the details |
| radiusborder | No | Radius Border of Payment method borders in widget |
| ISVCNEnabled | No | To enable VCN based settlement for merchant integration |
| VCNPrivateKey | No | Private Key to decrypt PAN & CSV in VCN settlement. Recommended 4096-bit RSA keypair |
| VCNPublicKey | No | Public key given to Klarna (in JWK format) to encrypt PAN &CSV in VCN settlement. Applicable for testing and production. Recommended 4096-bit RSA keypair |
| VCN Key | No | Key ID (UUIDv4, included in JWK) specified for corresponding public key shared by the merchant with Klarna. Applicable for testing and production. |

* 1. **VCN Set UP**

To use the VCN settlement for Klarna orders, the merchant should:

* Enable VCN option in KP configuration through backoffice
* Generate a 4096-bit RSA key pair. Set the Klarna configuration for the property ‘vcnPublicKey’ with the value of the public key without the header and footer lines (begin and end public key) and the Klarna configuration for the property ‘vcnPrivateKey’ with the value of the private key without the header and footer lines (begin and end private key).

**How to generate a 4096-bit RSA key pair:**

To generate an RSA keypair with a 4096-bit private key you can use the following openssl command:

***openssl genpkey -algorithm RSA -out private\_key.pem -pkeyopt rsa\_keygen\_bits:4096***

To extract the public key from an RSA keypair, you can use the following openssl command:

***openssl rsa -pubout -in private\_key.pem -out public\_key.pem***

In the folder where you have executed the above commands two new files will be created - public\_key.pem and private\_key.pem.

The contents of the files should look something like:

**public\_key.pem**

-----BEGIN PUBLIC KEY-----

MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAoNYG7l2G8nZa+22oBYZk

tV228lw3UE9WO4oxfknJtKEdHn84x55ULt8KQTh9NVtdeKC8nTfTgyvMt/GNCa18

xuZV/lGYDftKt85hbV5EjOum+StAIufEXvlBX7nMOMc1KyWm9kp2kbqd88mFIX63

KV94OoNEXcNatRDFYR+qz53+ifadDQtQ1slVNStdroCZDJ1+LxtBy9V+BdmsBK1E

RLsKh/JLXyWE24FJKV+z00s7TQkdWW/5ET12OGQYZsWo1yqgi9HplNvrisve8vWP

xaL4m8iZ3I/9yYdg7yANQbTxSJcbbRCgaaagPo30CNxeqU6qafY5g8vY3E52CoXH

DdO4UslX1qcuYIDhqaDzey6W+b8m755xLi+rqQyM4PBWL0J0dM3FVid8+4YKILex

3AKBFciqRCMHSOGaEeyrXKTjlAsghr9RS8PifvQRrL440cHzqw2vX0DvpjSWcmUJ

tW4wUq5RNSsobrxnVmoV6fj1z67Q/1P+l5Ie+oowdahR5ztVqJlO+2PNoX4I5VDs

/Pkz3f8wWVc3Mp2oNT244o+/NIiyRfPFaJJx7JAgrcvZt2nFAmY4QApXLFJCpgEM

wYucE4AH4gJKsh3KZbxRERrrO72bL2rxvWqBp/0h7DcMsV9sQs4BvxxIl6CF506F

ThzmclaKLBAyd5LALiXiPfkCAwEAAQ==

-----END PUBLIC KEY-----

**private\_key.pem**

-----BEGIN PRIVATE KEY-----

MIIJQQIBADANBgkqhkiG9w0BAQEFAASCCSswggknAgEAAoICAQCg1gbuXYbydlr7

bagFhmS1XbbyXDdQT1Y7ijF+Scm0oR0efzjHnlQu3wpBOH01W114oLydN9ODK8y3

8Y0JrXzG5lX+UZgN+0q3zmFtXkSM66b5K0Ai58Re+UFfucw4xzUrJab2SnaRup3z

yYUhfrcpX3g6g0Rdw1q1EMVhH6rPnf6J9p0NC1DWyVU1K12ugJkMnX4vG0HL1X4F

2awErUREuwqH8ktfJYTbgUkpX7PTSztNCR1Zb/kRPXY4ZBhmxajXKqCL0emU2+uK

y97y9Y/FovibyJncj/3Jh2DvIA1BtPFIlxttEKBppqA+jfQI3F6pTqpp9jmDy9jc

TnYKhccN07hSyVfWpy5ggOGpoPN7Lpb5vybvnnEuL6upDIzg8FYvQnR0zcVWJ3z7

hgogt7HcAoEVyKpEIwdI4ZoR7KtcpOOUCyCGv1FLw+J+9BGsvjjRwfOrDa9fQO+m

NJZyZQm1bjBSrlE1KyhuvGdWahXp+PXPrtD/U/6Xkh76ijB1qFHnO1WomU77Y82h

fgjlUOz8+TPd/zBZVzcynag1Pbjij780iLJF88VoknHskCCty9m3acUCZjhAClcs

UkKmAQzBi5wTgAfiAkqyHcplvFERGus7vZsvavG9aoGn/SHsNwyxX2xCzgG/HEiX

oIXnToVOHOZyVoosEDJ3ksAuJeI9+QIDAQABAoICACRkaUsUNI22RB3yEPu3DiCP

pO6v+QAeA4gTW+GUdqR9dCZLaSCZ7bhxVVOuoX4qPzslO6hjUmOyzG6upFgVPk+P

HNQfyEUZoC148Eib9OziAXUN2URMpv1KbwVm+BO814X8zguai7uru0PHTG1oy677

4Ct1OknxAxxHQDIaxT6XJFo5SA4EinUfNz2Bo3/xry/QjxW/mCK0GwDd4PNp9TGM

FPTv2SgdSDOWzGQlOH5N3owuzMpI8NV6z74wv+i5Ptv41Dzu8WhyXpiYSsk00SRK

HPC68j2bAzTPghp5aSZ9976SGm2SPonJXyboXdiHbI/osdyqDxeIT3iB9GmrHX/i

kHPGJCh7fRZvqj39Hc+IxYjabwW3rDeDIPB7ab9z1KLF4z1D6AZOKCPyTaDRdQ1Q

eDi7LwDmk7NHEPrmF/nIcguQdqbIbmFO2zEs0TOe6y4uBMndRsbQprTNSMUdBkrA

lNaYVSTQ1Z0Y/8DZDpGcyS1OnJv74F15uDjKN6/ov991mZ1JrZ+V2sdS3EDUlmvP

6thQKwI7Ln6h+ApHtWUG1NmvQe5gJE0qAeJ9b45clUzIRUwhVmEp8NoIJh0kAjaN

d4lk7xy9ZRDUY5yekPeYrJPShjsHAyEoktJIjRufI2UUq3uxNjjICoQcOVGfNDIS

YTTPwpu1pmC0C+rh2fgBAoIBAQDRultRArvtc2JKhVOUyZk88zd9kvrI6fNiyKmi

HgiWf7qkTPD9xhOQWDw3iwRFQAD+YkgV5MCBO8wp8oO8GEsOCI+XZWExOcPT0Vfj

PZHiQrTFnlfG/+fAO14xLf3j3ED4YQXdHOKI3xoLknQx/EydLoctxgkkpgWLrsA7

DwdSAg1/0sBvaHY27ogAfdimHdaKZ5OAe4a9k1qP3xVZBuOe8Sd65unBavUJLDuv

ikeNmkSVgW1sm55/729JIr63USHF76It+vE1cdZ+vKg5vYotsQgPzvNBmUO/E8Gj

zMXQRfqfvEDlNXEX0rCupTkw1G6AGTwQc/NPzyr/LTpLe6UBAoIBAQDEUjTiG11V

hf7WjdG3gctRlr+mYapQHgXdVLx2QSaqUYid+0QXK11YfJlsRB6nwa+OED83RfP0

lIFqxpzudSLPmoDuIBT7Dl5c/aleyKs/siUusP8QVDXk6OAR84XSytC35sIRV7pE

VMuBL91jfkQ0Lf/PreslK/kI6Yvwwp4qrHK6/f9TgciHclYtf+/oti4ky6GJgfmP

fmuCqjxmUKbXXFPd5RbL2THGOowilb8zDLjf3RlbjlQFqogAk6H9hp2V0VZLiJHp

UWM3z3zxDWeDaqJ08sHuk/rA9QpsVTu8IGTQsxdj8JwluN1Q+YZiOuPiSENBqPzT

V3exexzo3sD5AoIBAGU3qEyPojz1+9D1SaI8LW2CABzlq4z9g84ABAZOslxX5q7W

x1PinZyDSQSRXg1B13jt29ZdIR79ygnQlg1YOBjcvtgVQHPuafk3RlBQbbCh+vaI

9dn/tUxMGqhnhunKaby1rovJHfdqnPpKwzNAjYUqaGkJ822xhmmke/fEyAanIPa4

stDRvIPEWPTLx5xcOCdx13khpKSnkgRvaLEfpwkVX7Vr7hK/2OSFaYTNmrzXYBQ7

c6D/9d3Oo4nLb/mu+Tq67S19t53Qg/GEgTfkpuRoVPi0KyhUnKKCGWlBMZLTwyIG

S9eTFDKoJ0cSTGipjW7bPua93wZ8eEbRABpf4QECggEANNhQBeEJ0aCdBVHtdrEI

crDaa8X0W1aJi5dol4hYCRajaKsfHAF/QfdgMQVxHwUC5YG4En/Q+DAVWhGWYpXD

RhC3zeFy5FVszyk0sx/fAOlKGvRn5BRW4YRR9GMRzbjsT+RcruBnckdE9ERXGpX9

c/JB3rxZBIt+oIiFM8yfWKtMwsrmNKtFuDftvJeok4KejycFF4eWDqsf828xjPT+

xA/FP4CQD1UqkcpmuFSIwAwXo6LXVY7NTS0nKMiUnTLkLlTIHtLnO9+9jmNapWRP

Tc+hZUuHKlpI8DHFmX2j87LgkFD05eD5lynY4RgZtU1W1C1RdVYwoA72WB7knEaB

uQKCAQAH9s67P/7fFX9dfEans3PHU4nGjD8dJ8eoNQ6DhBMydZpGWI5ZUeEBZDRk

0cBOeRs5BOcS43Em9kETpzawyCwxmnwzl+CzoPzMQcTw9tXomF9HG6RJ9XBdJfGA

ALAwCd4bASxmFM6guSP5GKnZ9aY3tR3tWWDfr7f9z8wOewzzpPclwRh009fPe4TC

NXoEm1MELJVeUieDSLKZgjgCw8WHGqLItONpA0/fwSM2gIcxETVV7qx3aPuJzCVh

LQZoBLQk3UMKsWDdpzeBdiERe66NAgVk92Xe7SY9EY2vymaq761i1x1vlprT27qp

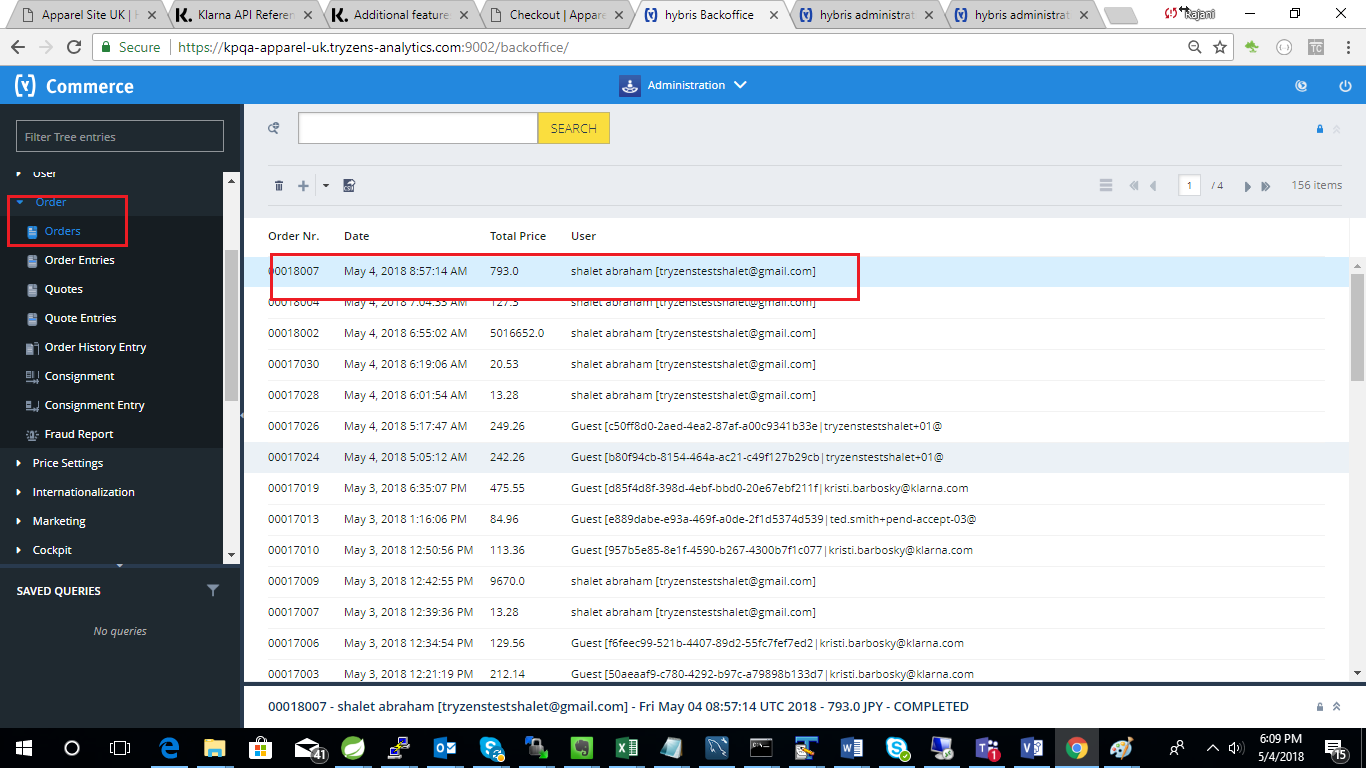
240LDJawqM0IraKmdCvWjofWSaOU

-----END PRIVATE KEY-----

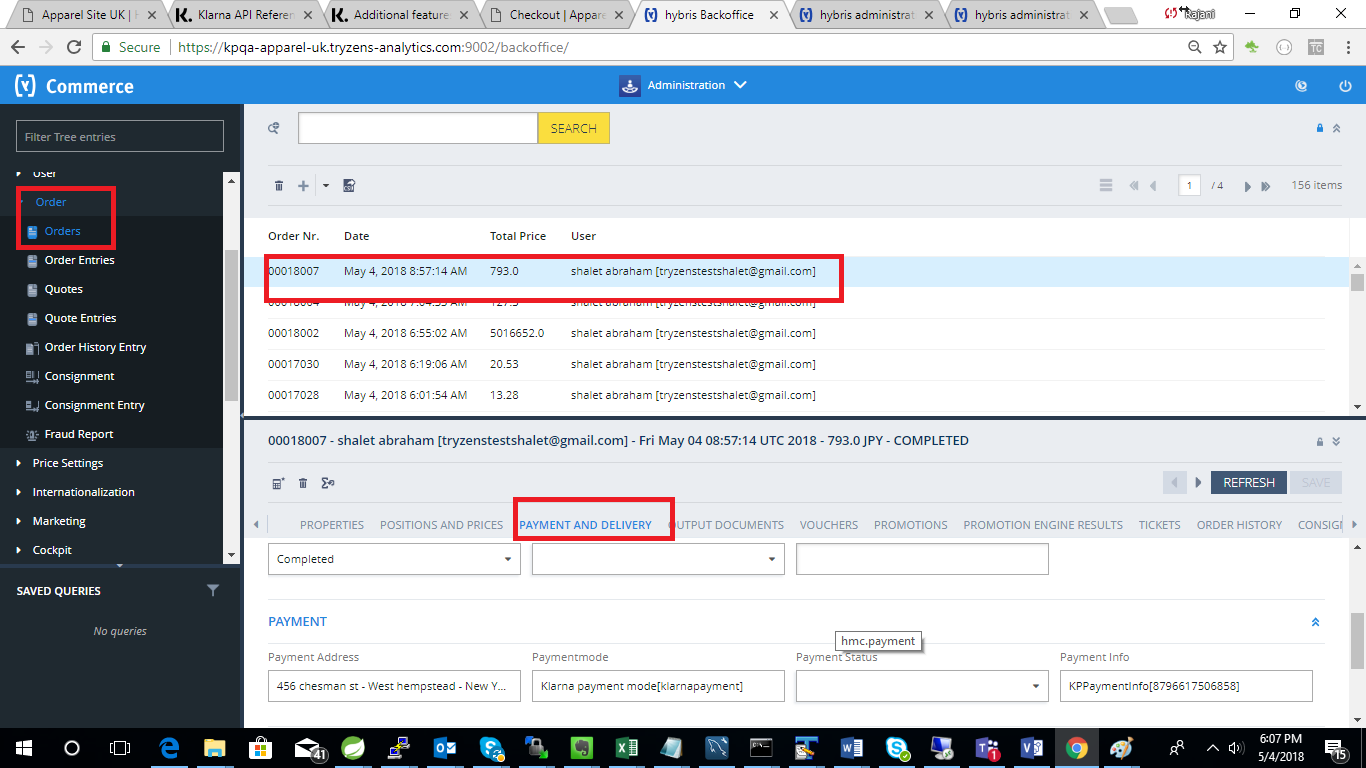
* The virtual card decryption is only a sample implementation in the KP extension. It is required that merchants only use it as reference for development but never store decrypted virtual card numbers/cvv (PCI\_DATA) in production environment.
* Important Note: **DO NOT SAVE DECRYPTED PCI DATA (incl. VCN/CVV) ON THE SERVER!** It is the responsibility of the merchant to ensure PCI-DSS compliance and to ensure the card data is handled securely in co-ordination with required partners/Payment Service Provider/Acquirer. Please review in advance the order export details required for virtual card based Klarna orders. Any historical decrypted PCI data should also be expunged, regardless of the validity date.
  1. **Verifying the VCN Data in Backoffice**

To verify the Klarna Orders in backoffice, follow the below step.

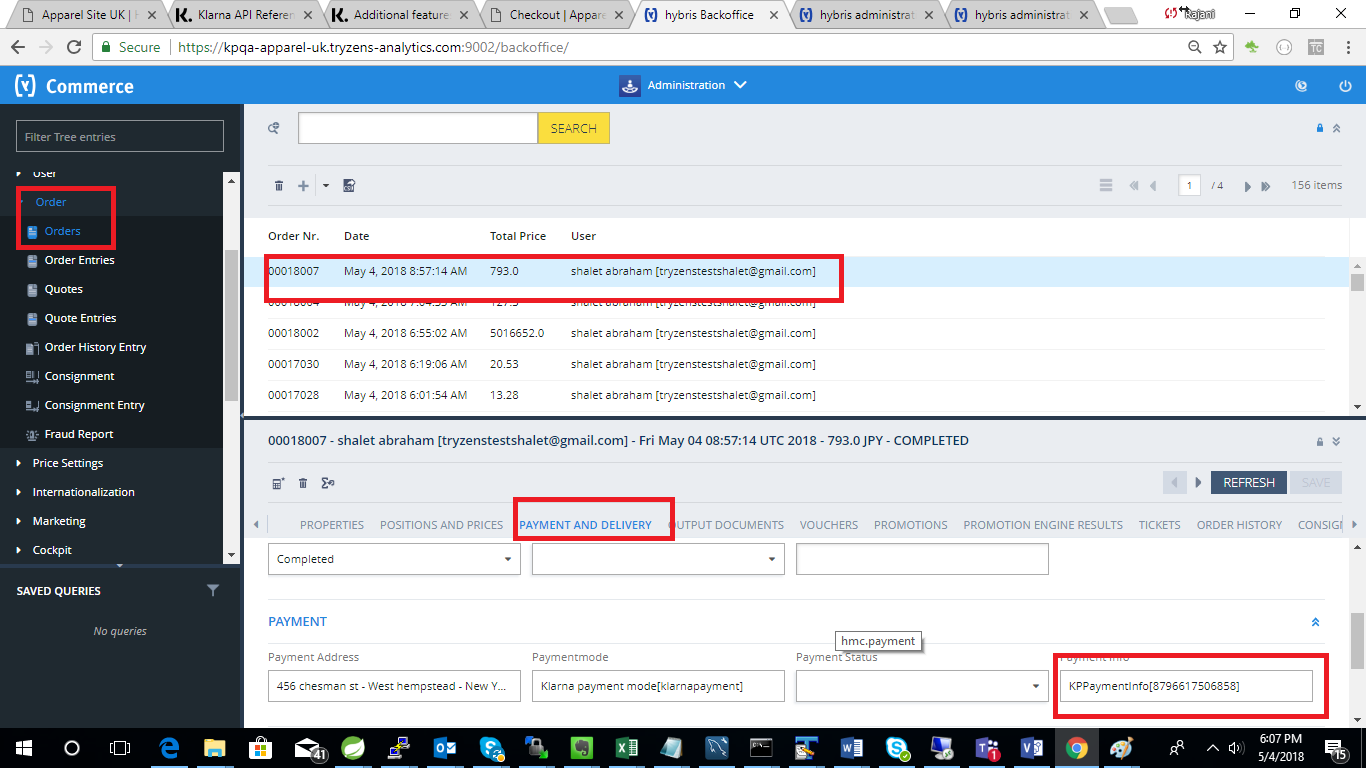
Go to backoffice & Click on order in the left side menu & Search for the order number.



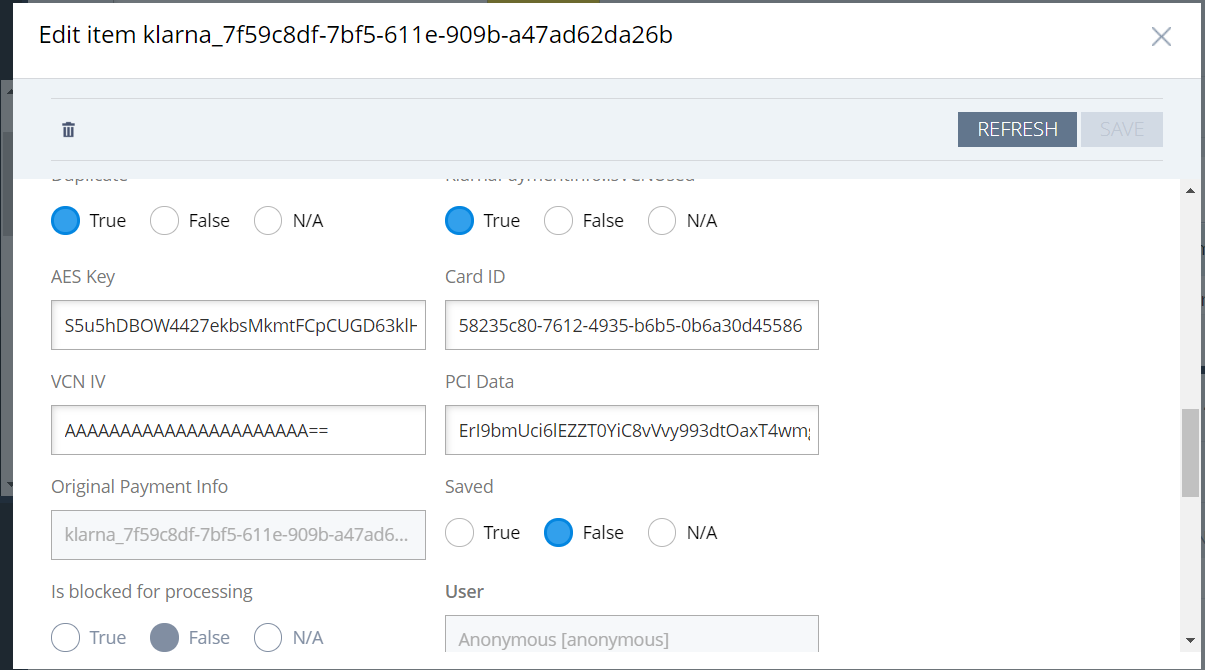
Click on the Order number and go to payment & delivery.



Select the payment information.



Verify the virtual card details.

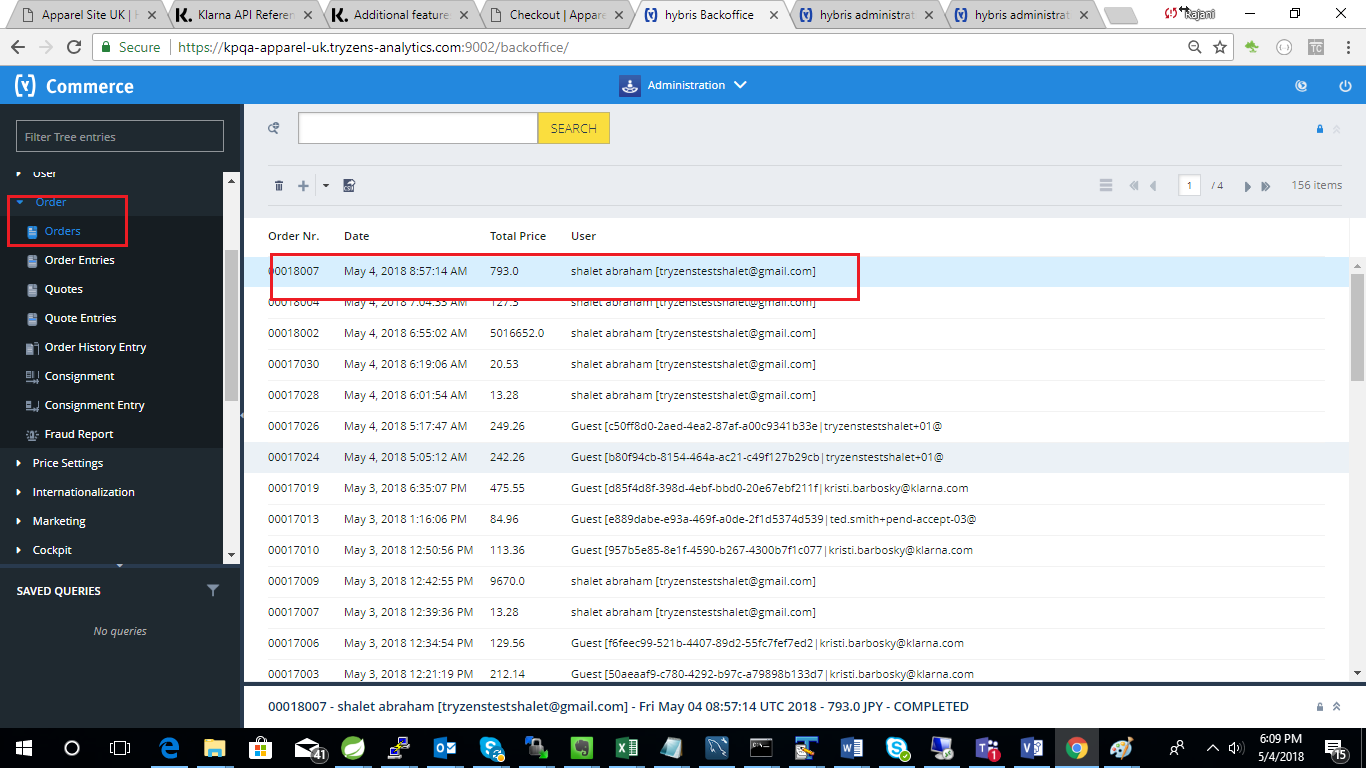


**Note:** The cards are one-time use cards which are specific to the order and merchant responsibility to securely store them. The above representation is only for reference. The data must never be stored decrypted in production.

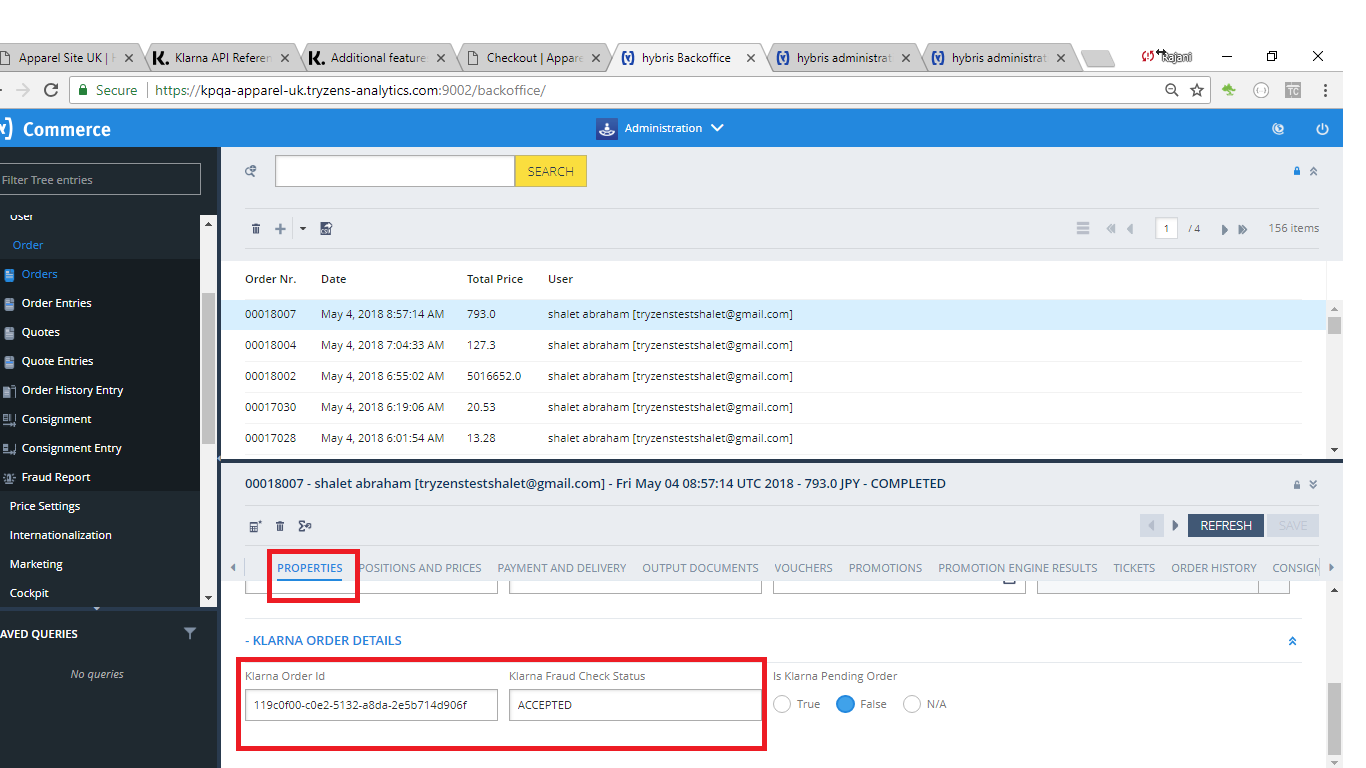
* 1. **Verifying Klarna Order ID in backoffice**

To verify the Orders in backoffice, follow the below step.

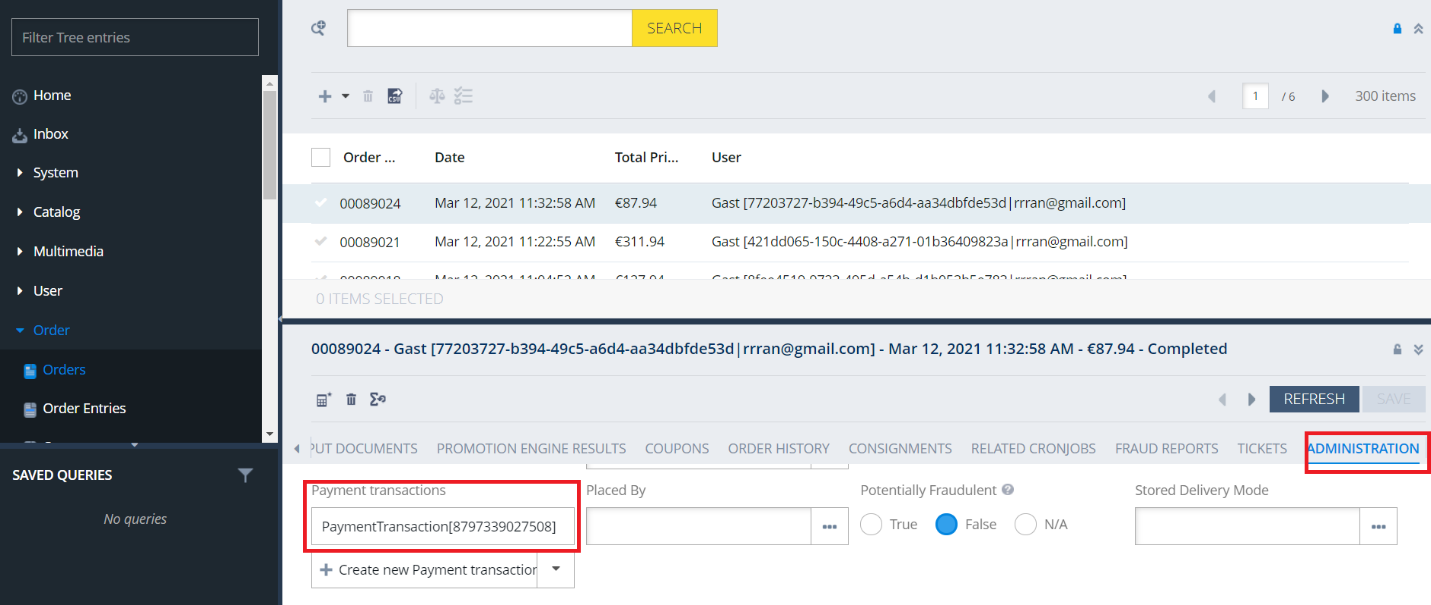
Go to backoffice & Click on order in the left side menu & Search for the order number

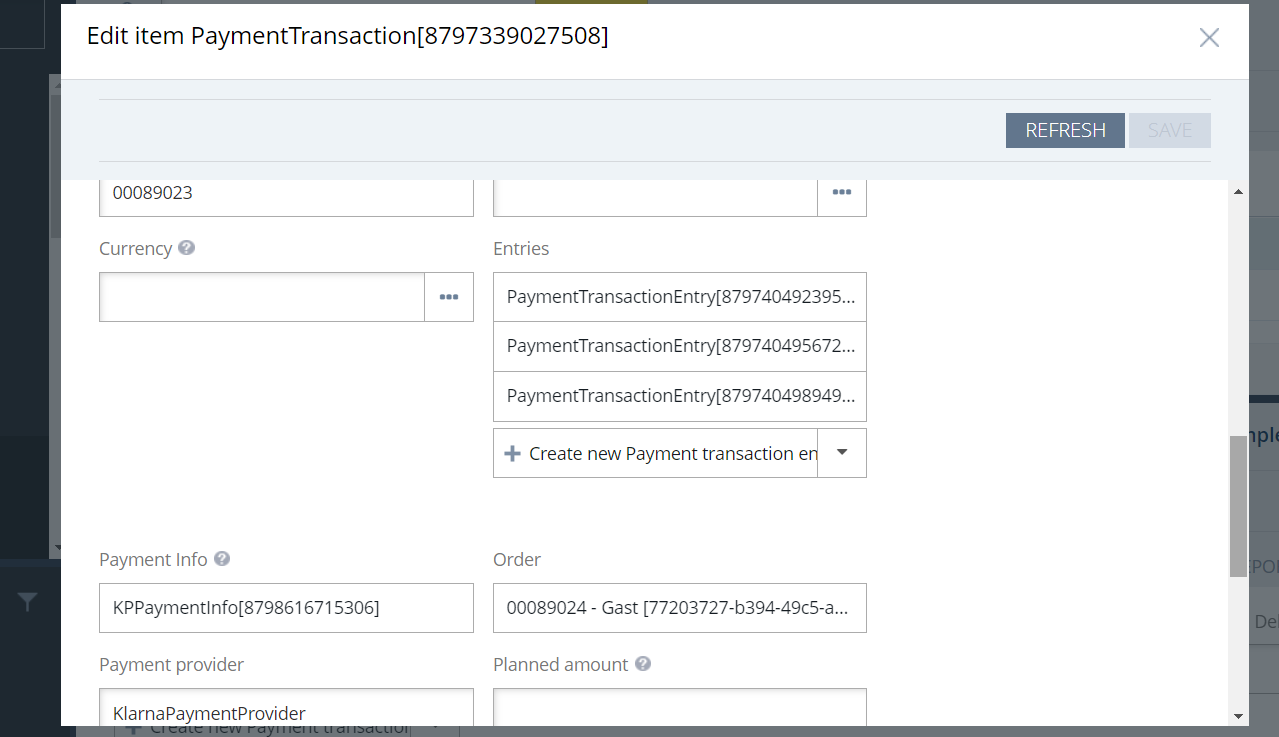


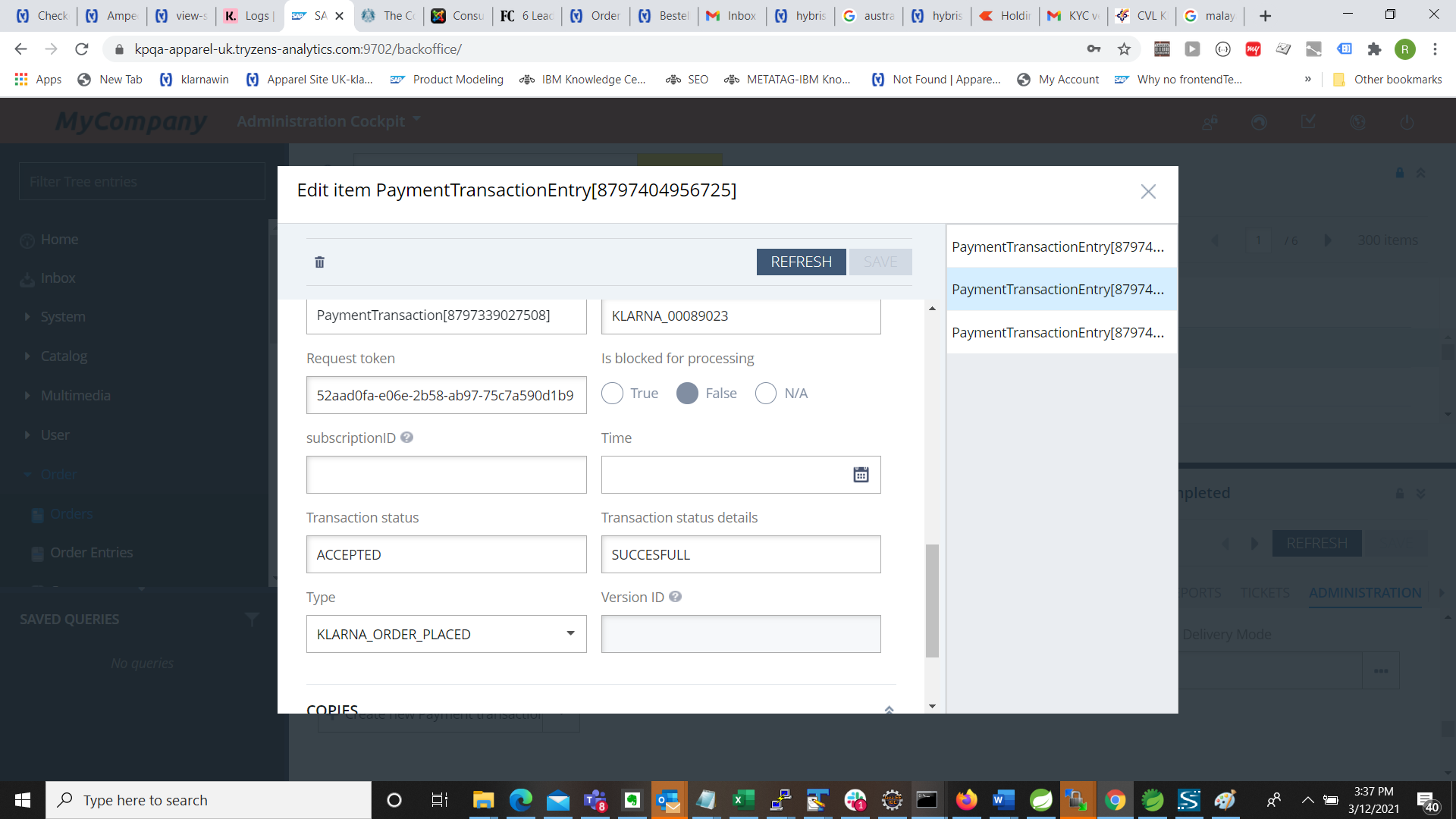
Verify the Klarna Order ID under properties tab



Verify the payment transactions under Administration Tab. Note that, Transaction Type KLARNA\_ORDER\_PLACED is custom for Klarna place order.

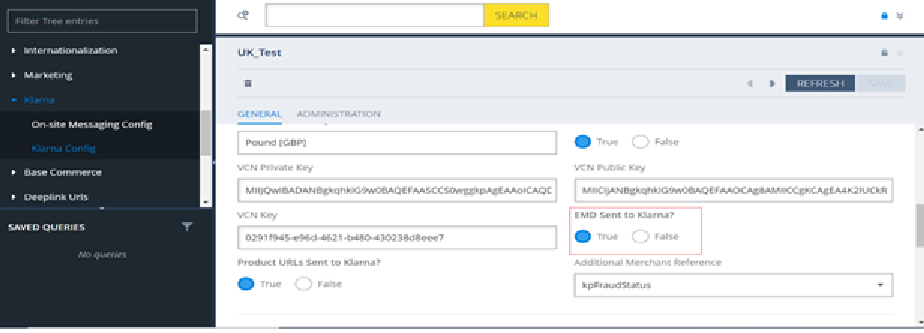






* 1. **Extra Merchant Data (EMD) Setup**

Additional merchant information can be sent to Klarna by enabling EMD option through backoffice. Additional details like customer information, alternative delivery address, reservation details can be configured in the code. Note that OOB only customer info for logged in customer can be sent.



In KPCreditSessionPopulator.java, method addAttachment is exposed to set the EMD data

# Extension Model and Classes

Models extended from OOB SAP (Hybris) Commerce platform

|  |  |  |
| --- | --- | --- |
| **Model** | **OOB Model** | **Attributes** |
| BaseStore | BaseStore | klarnaPayConfig |
| AbstractOrder | AbstractOrder | kpIdentifier kpAnonymousGUID  kpOrderId  kpFraudStatus  isKpPendingOrder  isKpAuthorised  isKpFraudRiskStopped |
| KPPaymentInfo | InvoicePaymentInfo | paymentOption  finalizeRequired  description  authToken  isVCNUsed  vcnBrand  vcnCSC  vcnValidToYear  vcnValidToMonth  vcnHolder  vcnPan  vcnCVV |

Classes extended from OOB SAP (Hybris) Commerce platform

|  |  |
| --- | --- |
| **Class** | **OOB Class** |
| KPPaymentMethodCheckoutStepController | PaymentMethodCheckoutStepController |
| KPSummaryCheckoutStepController | AbstractCheckoutStepController |
| KlarnaResponsiveSummaryCheckoutStepValidator | AbstractCheckoutStepValidator |
| KlarnaSummaryCheckoutStepValidator | AbstractCheckoutStepValidator |
| KPAddressPopulator | AddressPopulator |
| KPOrderPopulator | AbstractOrderPopulator |
| KlarnaEventPublishingSubmitOrderStrategy | SubmitOrderStrategy |
| KPPaymentTypeCheckoutStepController | AbstractCheckoutStepController |

# Klarna Payment Method Reference

|  |  |  |
| --- | --- | --- |
| **Use Case** | **Class** | **Method** |
| Create/Update payment session | DefaultKPPaymentFacade | getORcreateORUpdateSession |
| Authorize Payment | DefaultKPPaymentFacade | getPaymentAuthorization |
| Create Klarna Order | DefaultKPPaymentCheckoutFacade | saveKlarnaOrderId |
| Cancel Klarna Order | KPOrderConfirmationController | orderConfirmation |

# KlarnapaymentAPI Method Reference

|  |  |  |
| --- | --- | --- |
| Use Case | Class | Method |
| Get Klarna Payment session | Client | newPaymentsSessionsApi |
| Authorize Payment | Client | newPaymentsOrdersApi |
| Get Klarna Order by ID | Client | newOrderManagementOrdersApi |
| Create payment Settlement | Client | newVirtualCreditCardSettlementsApi |
| Delete Auth |  | newPaymentsOrdersApi |
| Capture Payment |  | newOrderManagementCapturesApi |

# Uninstalling the Add-on

* If SAP Commerce Server is running, stop it with Ctrl+C or by using this command, as explained in the Installation section of this document:

- Windows: hybrisserver.bat stop

- Unix: ./hybrisserver.sh stop

* Go to <HYBRIS\_HOME>/bin/platform and run the following command if it was not applied in this terminal:

- Windows: setantenv.bat

- Unix: ./setantenv.sh

* Go to <HYBRIS\_HOME>/bin/platform and run the following command:

ant addonuninstall -Daddonnames="klarnapaymentaddon" -DaddonStorefront.yacceleratorstorefront="yacceleratorstorefront"

* Go to <HYBRIS\_HOME>/config/localextensions.xml and delete <extension name=”klarnapaymentaddon” />, <extension name=”klarnapayment” />, <extension name=”klarnapaymentapi” /> & <extension name=”klarnapaymentbackoffice” /> .
  + Delete folder klarnapaymentaddon, klarnapayment, klarnapaymentapi, klarnapaymentbackoffice in the folder <HYBRIS\_HOME>/bin/custom.
* Remove the below from payment page.

<%@ taglib prefix="klarnapayment" tagdir="/WEB-INF/tags/addons/klarnapaymentaddon/responsive/" %>

<klarnapayment:selectpayment/>

* Rebuild the system (ant clean build) and restart the SAP Commerce server.
* Open web browser, go to hAC → Platform → Update.
* Check the 4 checkboxes under General Settings: Update running system, Clear the hMC configuration from the database, create essential data, Localize types.
* Click on the Update button to update Hybris.

# Recommendation

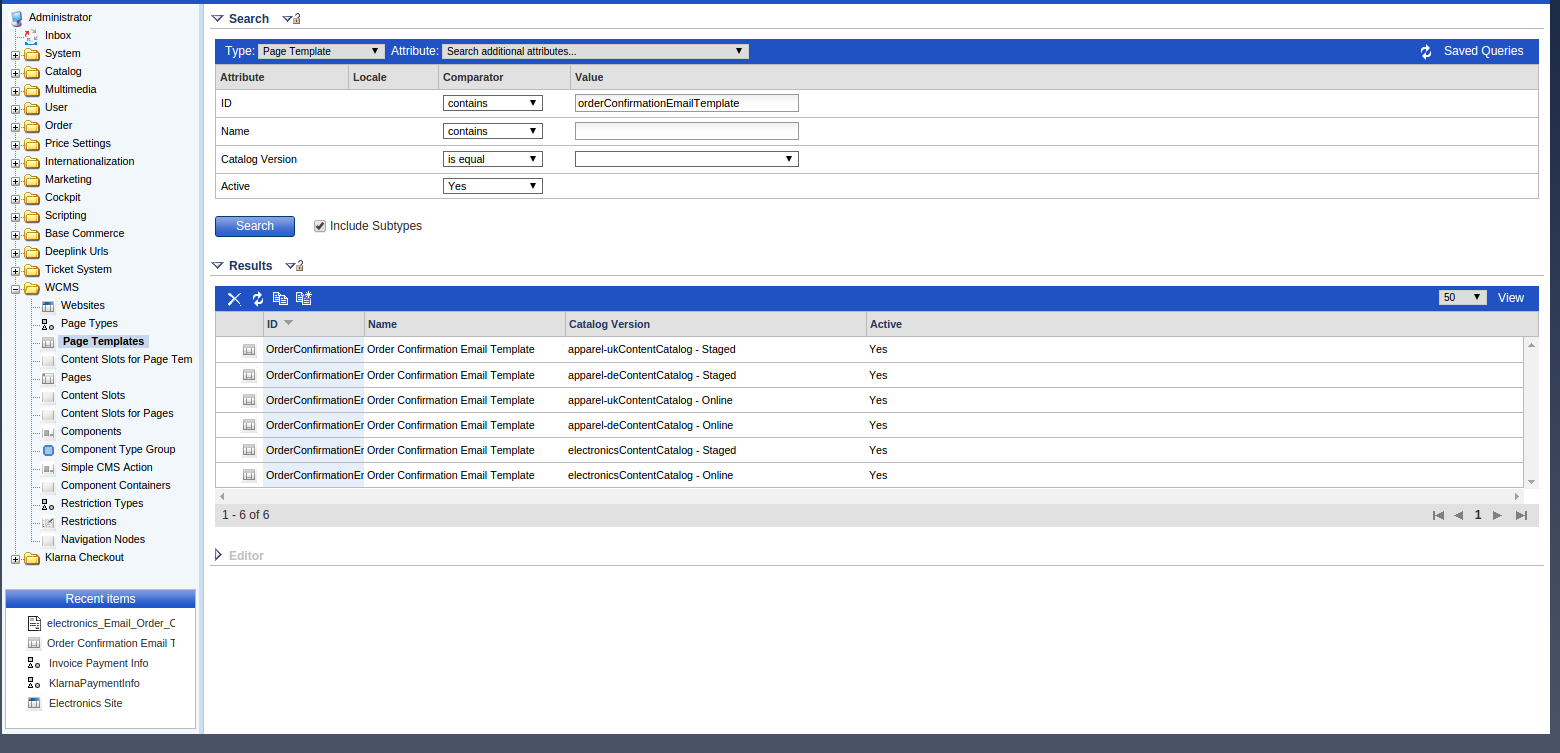
* To ensure that KP iframe and the Payment method text within it clearly viewable, please choose a min-width in your checkout such that it works well in conjunctions with majority of your mobile user’s screen sizes. Hence its preferred to have the billing page section with same min-width to get a better user experience while adjusting the browser width.

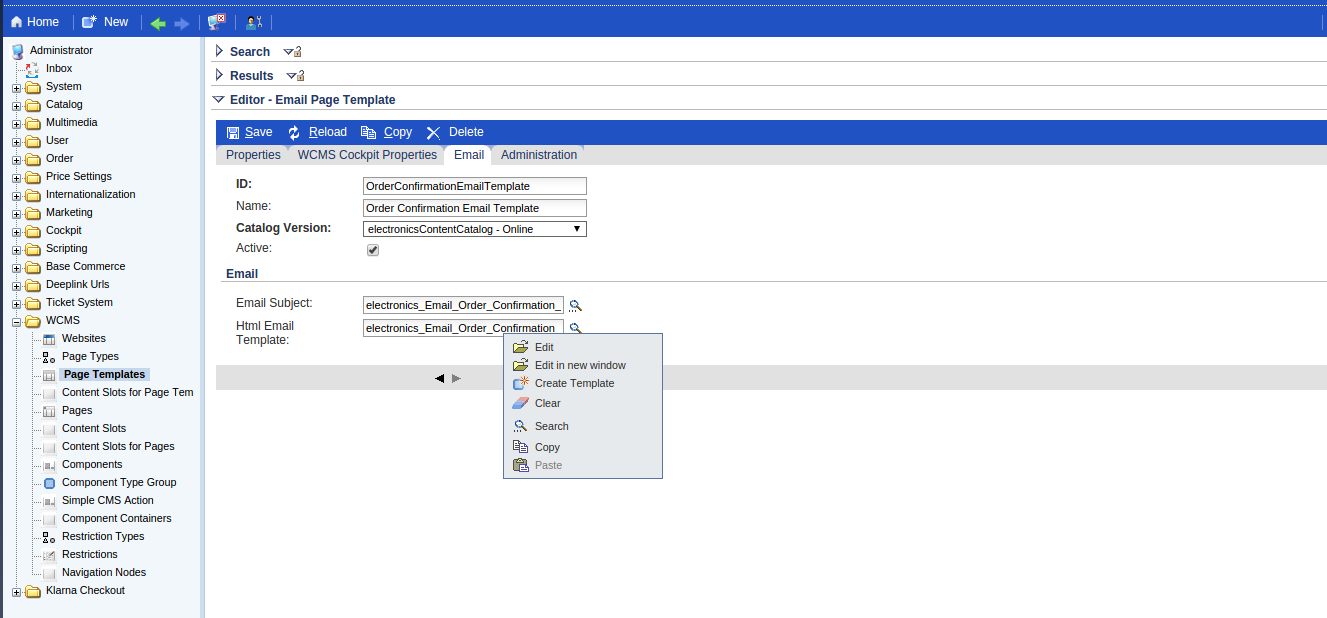
# Modifying the order confirmation email

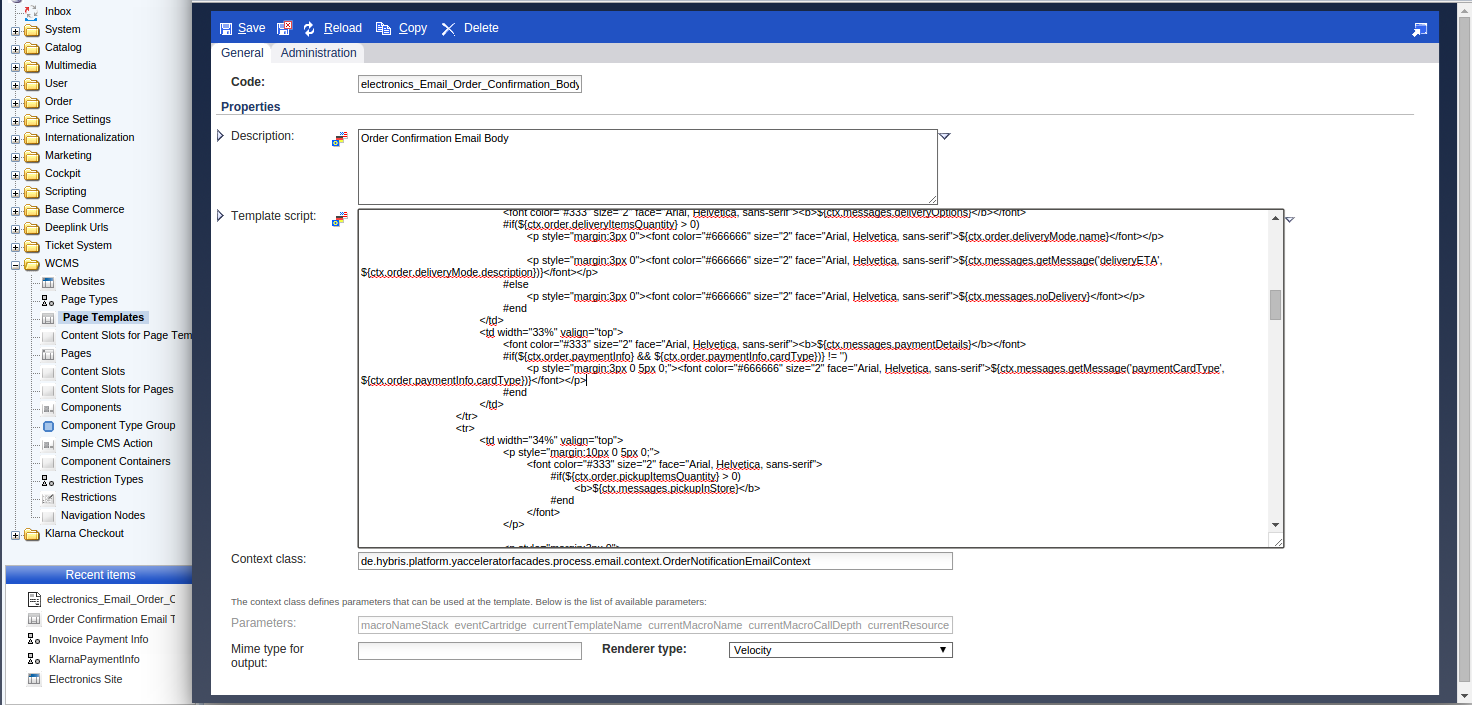
SAP Commerce will send the customer the order confirmation email once the order is placed. Payment instructions and notification for customers, will be sent directly from Klarna. We suggest that the ‘Payment Details’ area of the confirmation email be modified to inform the customer that payment has not been taken yet (except Pay Now payment methods), and that the customer is made aware that further updates will be received from Klarna. Customers have the option to check their purchases with Klarna and payments in the Klarna app(app.klarna.com).

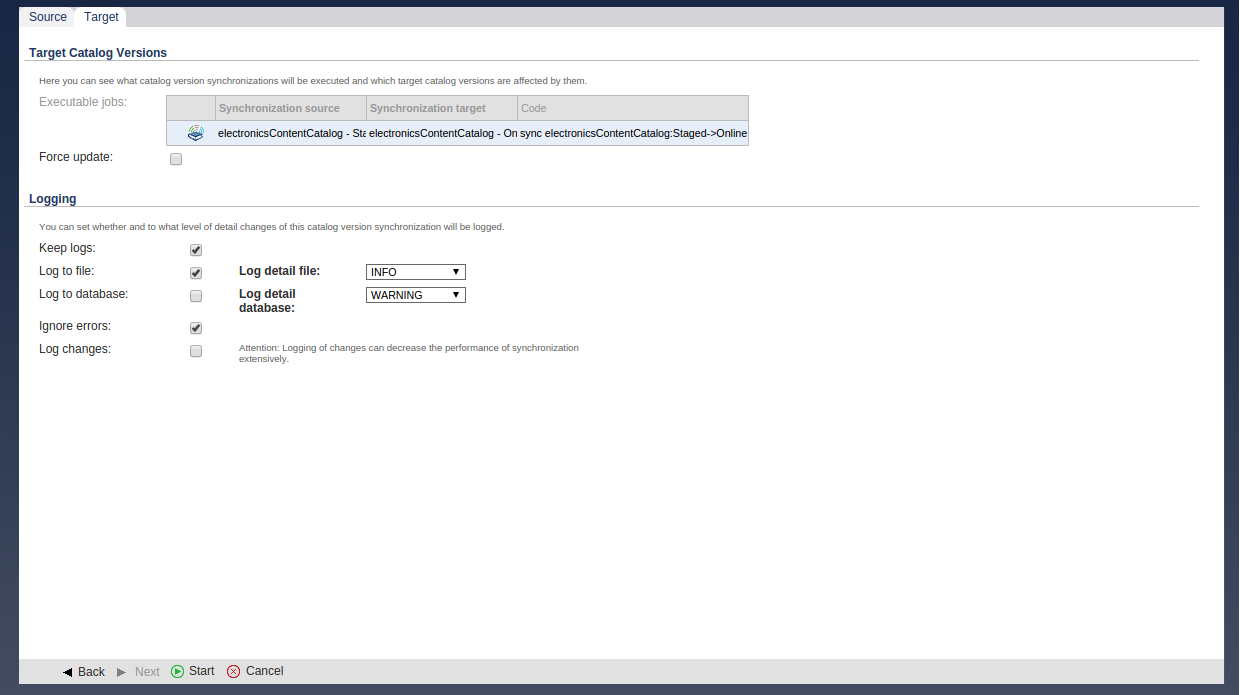
For having the payment method in order confirmation mail as ‘Purchased with Klarna’ the message property can be added. Sample Message Properties are provided in the folder /klarnapaymentaddon/resources/klarnapaymentaddon/messages. Merchants can modify the properties or add new property file of their own. Make sure that the property file is added in the EmailTemplate

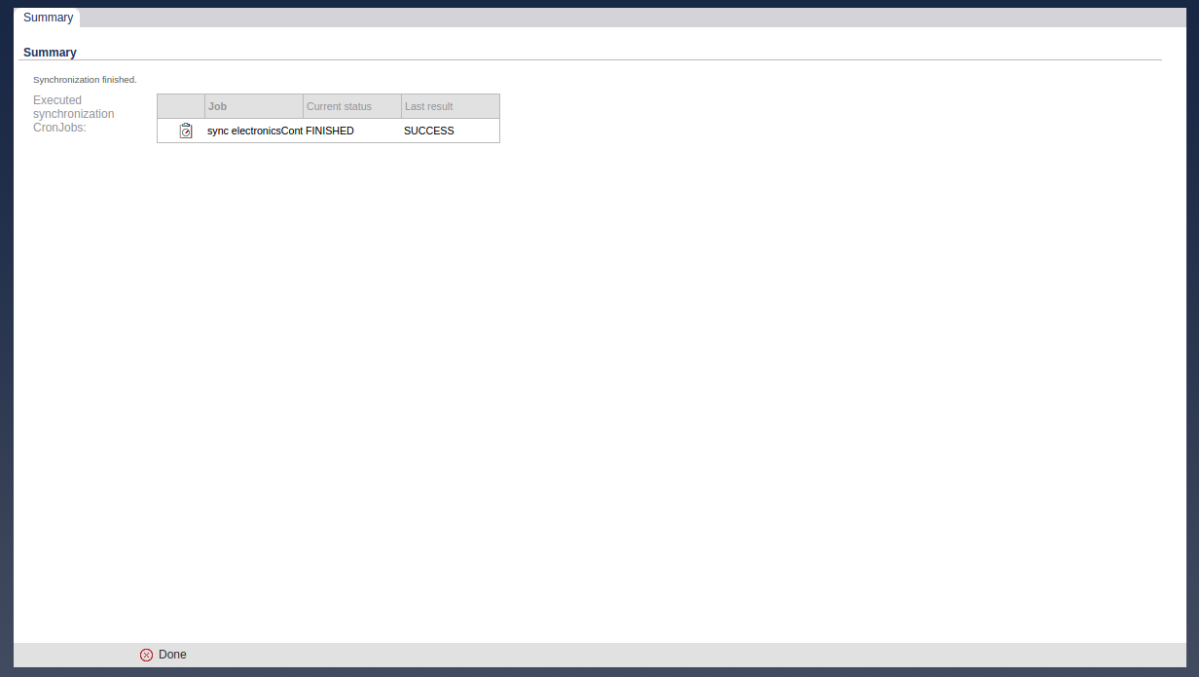
Below are the steps to follow to modify the email template:

* Login to HMC with an administrator account
* Navigate to WCMS -> Page templates.
* Find the email template which you wish to modify and select it or right click and select ‘Edit in new window’. Note that it is recommended that the Staged version is edited and then synchronized to the Online version.
* Select ‘Email’ tab and right click -> Edit in new window on ‘Html Email Template’



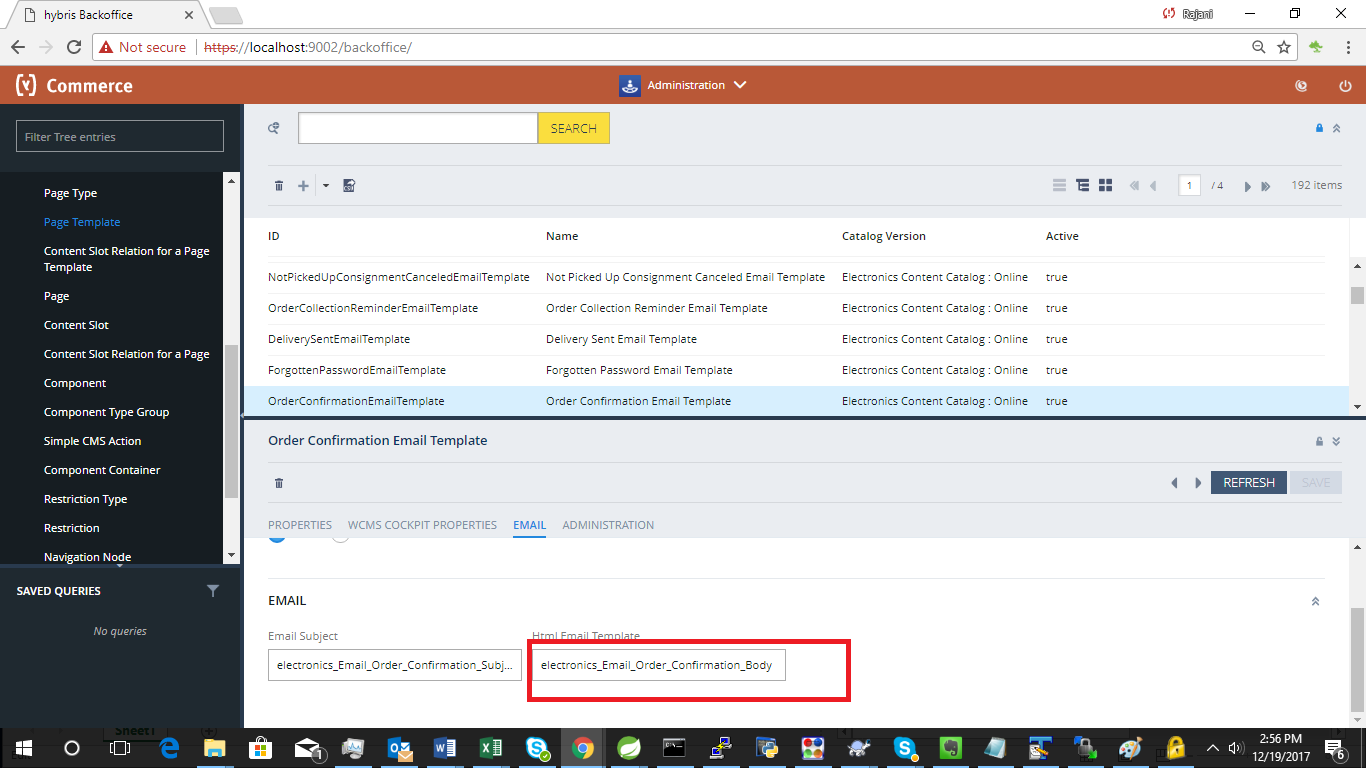
* Update the html content in ‘Template script’ as required.
* Navigate to Catalog -> Catalog Management Tools -> Synchronization, select catalog version to synchronize, then click Next and Start.



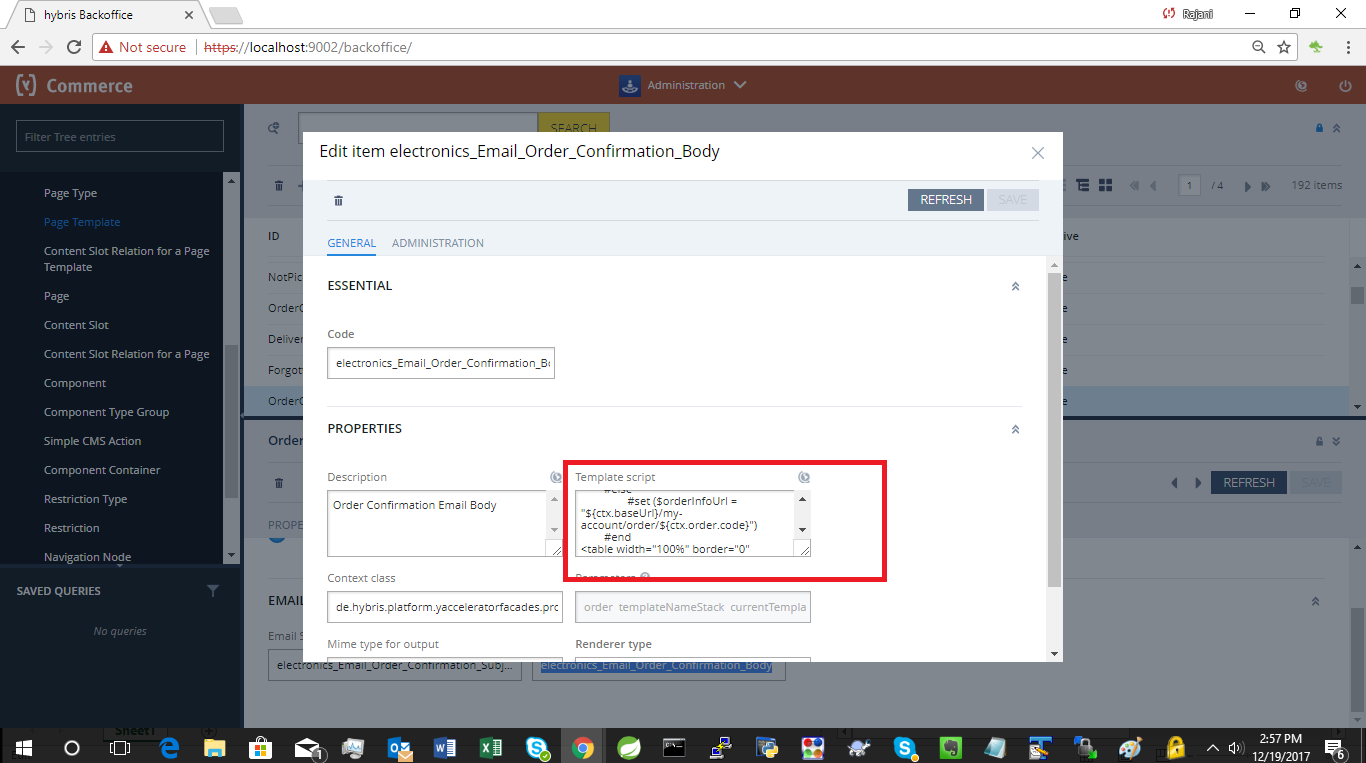


**Email Template can be updated using backoffice also using the following steps**

* Login to backoffice with an administrator account
* Navigate to WCMS -> Page templates.
* Find the email template which you wish to modify and select it or right click and select ‘Edit in new window’. Note that it is recommended that the Staged version is edited and then synchronized to the Online version.
* Select ‘Email’ tab and double click on htm email template.



* Update the html content in ‘Template script’ as required.



# Troubleshooting

* If Klarna Payments options are not available in the checkout flow, please make sure you have the correct Klarna configuration set up for the current base store: check that the config is marked as active and that the purchase country, language, and currency are valid for the base store chosen.
* ItemType typeCodes may have conflicts in the latest versions(2005 & 2011) of hybris based on the installation recipe. It is advised to make changes accordingly in case of conflicts.

# Installing the KOSM Add-on

* 1. **Download Add-on**
* Unpack archive and place folders klarnaosmaddon & klarnaosmbackoffice into custom folder of SAP Commerce Suite (<HYBRIS\_HOME>/bin/custom)
  1. **Update localextensions.xml**
  + Check the presence of <extension name=”addonsupport” /> in the localextensions.xml file
  + Add extension <extension name=” klarnaosmaddon” />
  + Add extension <extension name=” klarnaosmbackoffice” />
  + Check the presence of the target storefront extension.
  1. **Run installation command**
* If SAP Commerce Server is running, stop it. Depending on the server start mode we can use one of following method to stop server.
* If the server was started with embedded mode, press Ctrl+C to stop server.
* If the server was stared with service mode, use this command to stop:

- Windows: hybrisserver.bat stop

- Unix: ./hybrisserver.sh stop

* Go to <HYBRIS\_HOME>/bin/platform and run the following command if it was not applied in this terminal:

- Windows: setantenv.bat

- Unix: . ./setantenv.sh

* + - Go to <HYBRIS\_HOME>/bin/platform and run the installation add-on with the command:

ant addoninstall -Daddonnames=" klarnaosmaddon " -DaddonStorefront.yacceleratorstorefront="yacceleratorstorefront"

**Note**:

* Depending on the project, the storefront name may differ from the default storefront name that is provided by SAP Commerce.
  1. **Mandatory configuration changes**
     + Configure the script (library) url, uci, country, placement tag ids in the backoffice. This information is available in the Klarna Merchant Portal respectively for playground and production.
  2. **Mandatory code changes**
* Modify master tag to have the below script as the first entry in the body tag

<c:if test="${isPdpEnabled || isCartEnabled}">

<script async src="${scriptUrl}" data-client-id="${uci}" ></script>

</c:if>

* Modify the pages where the OSM content is required. Samples are given below

1. PDP Page – If KOSM marketing content is required in the PDP page, do the below steps in productPricePanel.tag

* Add the taglibrary

<%@ taglib prefix="osm" tagdir="/WEB-INF/tags/addons/klarnaosmaddon/responsive/osm/" %>

* Add osmproduct tag at the appropriate place

<osm:osmproduct price="${product.price.value}"/>

* If placement refresh is required,Include at the end of the page

<osm:osm/>

1. Cart Page - If KOSM is required in cart page, do the below steps in cartTotals.tag

* Add the tag library

<%@ taglib prefix="osm" tagdir="/WEB-INF/tags/addons/klarnaosmaddon/responsive/osm/" %>

* Set the final price like below based on tax setting.

<div class="col-xs-6 cart-totals-right text-right grand-total">

<ycommerce:testId code="cart\_totalPrice\_label">

<c:choose>

<c:when test="${showTax}">

<c:set var="finalPrice" value="${cartData.totalPriceWithTax}"/>

<format:price priceData="${cartData.totalPriceWithTax}"/>

</c:when>

<c:otherwise>

<c:set var="finalPrice" value="${cartData.totalPrice}"/>

<format:price priceData="${cartData.totalPrice}"/>

</c:otherwise>

</c:choose>

</ycommerce:testId>

</div>

* Add osmcart tag at the appropriate place

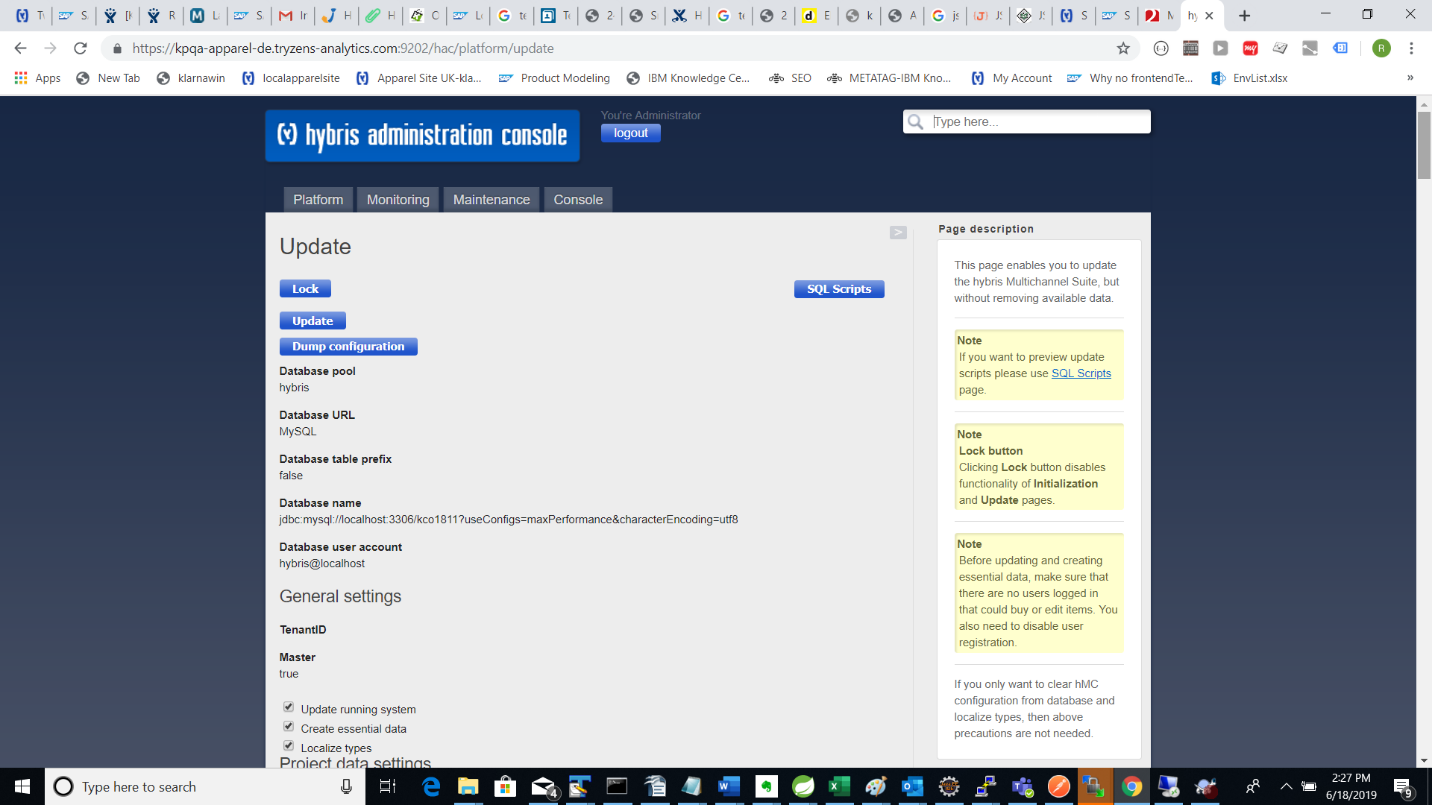
<osm:osmcart price="${finalPrice.value}"/>

* Do not update the content/media in the Klarna placements. Merchants have the flexibility to edit the placement design (look and feel) for the respective placement type in the Klarna Merchant portal using the “custom” theme option
  1. **Rebuild the system**
     + Go to <HYBRIS\_HOME>/bin/platform and rebuild the system with the command: ant clean all.
  2. **Update the system**

After the previous step you may need to perform full initialization through Hybris Administration Console (HAC) if this is the first installation of SAP Commerce.

If you have already performed full initialization, then you need to update your SAP Commerce system as follows:

* Open web browser, go to **HAC → 1 Platform → 2 Update**

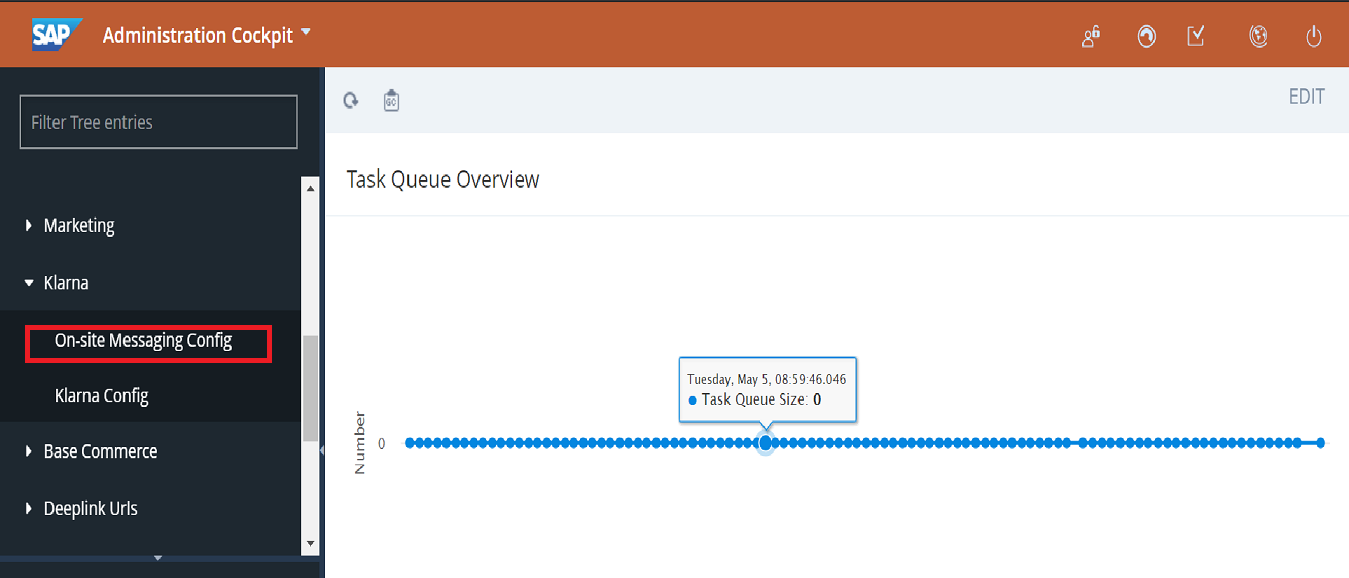


1. Check the 3 checkboxes under General Settings: Update running system, Localize types.
2. Check the klarnaosmaddon & klarnaosmbackoffice checkbox further down under Project data settings.
3. Click on the Update button to update the SAP Commerce system.
   1. **Activating Klarna On-site Messaging through Backoffice**

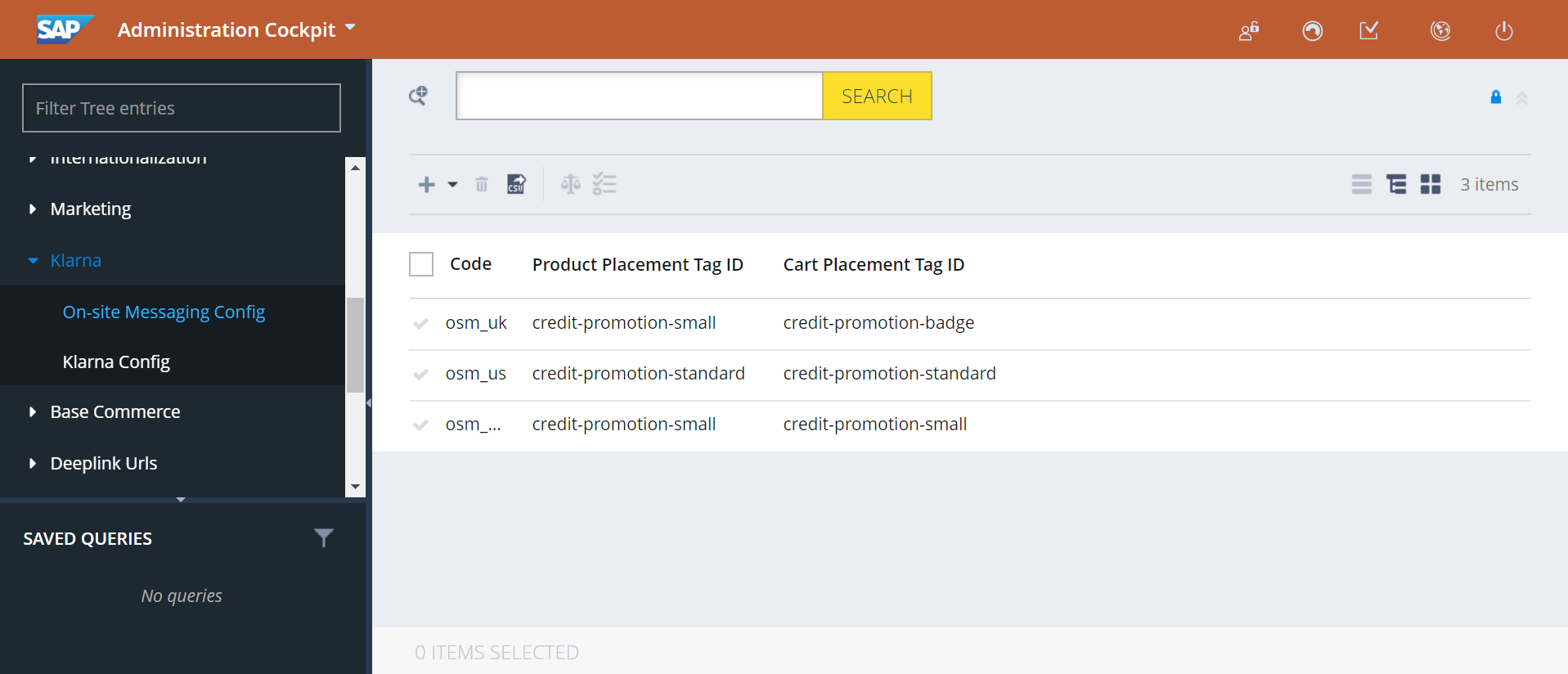
After installing klarnaosmaddon successfully, you can now login to SAP Commerce Backoffice to setup a Klarna configuration and assign it to specific sites in SAP Commerce.

* 1. **Create Klarna OSM configuration**

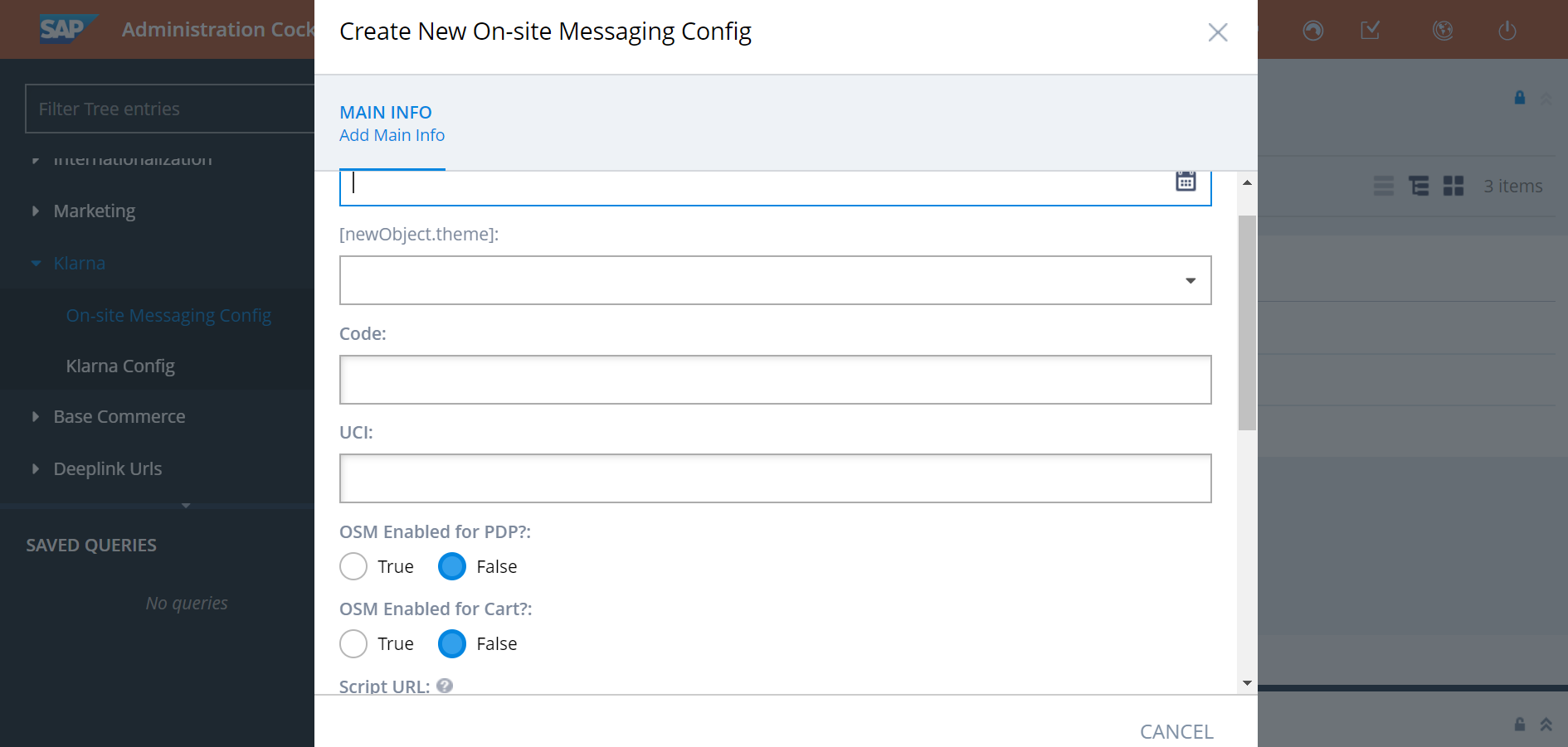
To configure Klarna, go to **hybris Backoffice→ Klarna → On-site Messaging Config**



Chooseto create a new On-site Mesaging configuration



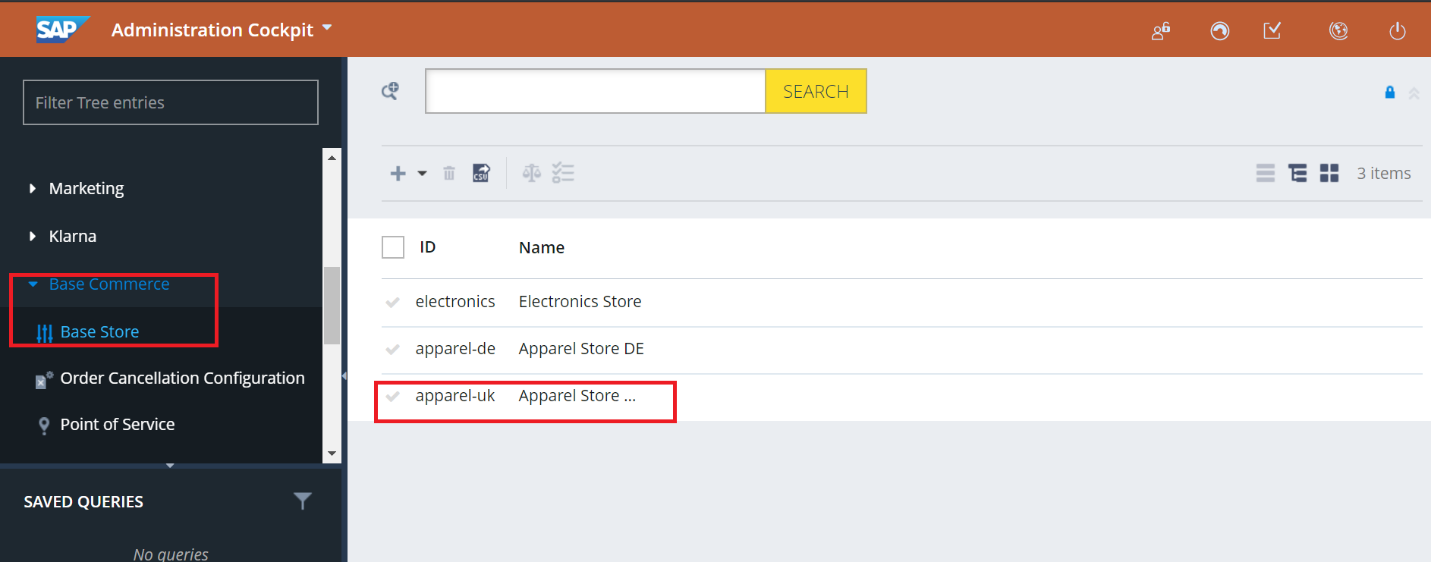
Fill in the mandatory and optional fields with the values fetched from the Klarna merchant portal. Available Fields information is given under Section 11.11.



After entering the required data in the above fields for the first time, click on the Done button.

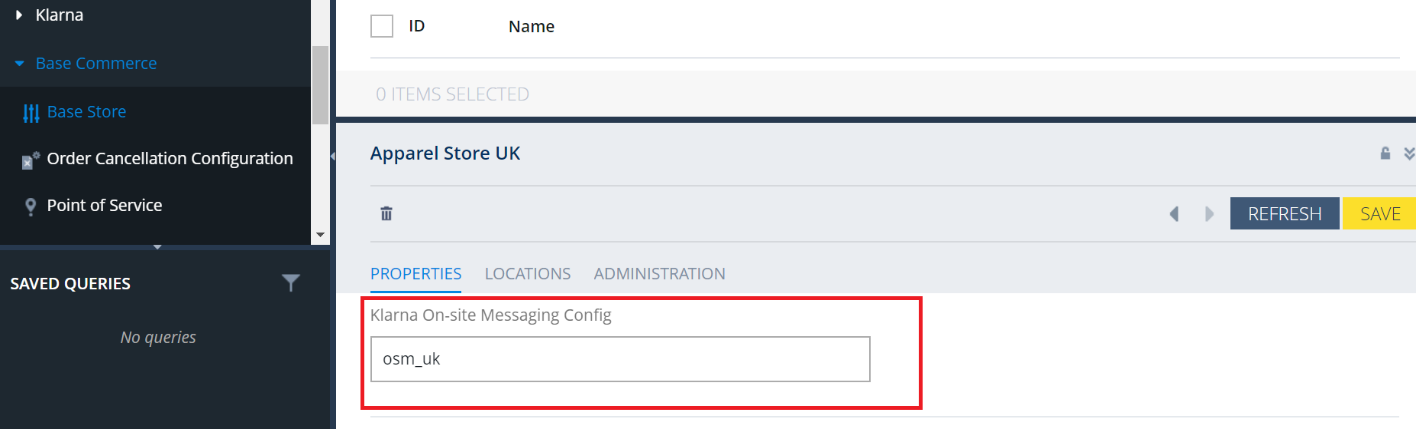
* 1. **Active Klarna OSM config for site**

Go to SAP Commerce Backoffice → Base Commerce → Base Store → Search → double click on the base store which you want to assign the configuration to → Choose Properties tab.



Go to Klarna OSM Configuration, then click on search icon & select the configuration to activate Klarna OSM config for the base store.

Result after assigning the selected Klarna config to the base store:



Click on Save.

* 1. **Klarna OSM Configuration Fields**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Mandatory** | **Description** |
| Code | Yes | Input any name/code. This will be the internal name used to recognise and manage this particular Klarna OSM configuration within SAP Commerce. |
| Active | Yes | Check to activate this configuration; leave unchecked for it to remain inactive. |
| Script URL | Yes | The script/library URL is fetched from the Klarna Merchant Portal (src) |
| Country | Yes | Country for which the OSM configuration is required |
| UCI | Yes | UCI value is fetched from the Klarna Merchant Portal (data-client-id) |
| Product Placement Tag ID | Yes | Placement Tag ID is fetched from the Klarna Merchant Portal (data-key) |
| Cart Placement Tag ID | Yes | Placement Tag ID is fetched from the Klarna Merchant Portal(data-key) |
| OSM enabled for Cart? | Yes | Flag for enabling Cart Placement |
| OSM enabled for PDP? | Yes | Flag for enabling PDP Placement |
| Is Data inline enabled? | No | Flag for enabling data inline attribute for OSM Tag. This attribute is specific to Paybright (Canada) only sites.  By default, it will be disabled. |
| Cart Theme | No | Select applicable theme from drop down for Cart Page. Use custom if configured in Klarna Merchant Portal |
| PDP Theme | No | Select applicable theme from drop down for PDP Page. Use custom if configured in Klarna Merchant Portal |
| Custom Style | No | This field provides an option to provide custom styling to OSM elements using ::part API. The custom style should be provided inside a style tag as shown in the below example:  *<style type='text/css'>*  *#osm-product-strip::part(osm-container) { background-color: #d9b259; }*  *#osm-product-strip::part(osm-cta) { font-size: 14px; }*  *#osm-cart-strip::part(osm-container) { background-color: #d4d122; }*  *</style>* |

* 1. **Uninstalling the OSM Add-on**
* If SAP Commerce server is running, stop it with Ctrl+C or by using this command, as explained in the Installation section of this document:

- Windows: hybrisserver.bat stop

- Unix: ./hybrisserver.sh stop

* Go to <HYBRIS\_HOME>/bin/platform and run the following command if it was not applied in this terminal:

- Windows: setantenv.bat

- Unix: ./setantenv.sh

* Go to <HYBRIS\_HOME>/bin/platform and run the following command:

ant addonuninstall -Daddonnames="klarnaosmaddon" -DaddonStorefront.yacceleratorstorefront="yacceleratorstorefront"

* Go to <HYBRIS\_HOME>/config/localextensions.xml and delete <extension name=”klarnaosmaddon” /> & <extension name=”klarnaosmbackoffice” /> .
  + Delete folder klarnaosmaddon, klarnaosmbackoffice in the folder <HYBRIS\_HOME>/bin/custom.
* Remove the osm tag library and tags from pd, cart pages.

<%@ taglib prefix="osm" tagdir="/WEB-INF/tags/addons/klarnaosmaddon/responsive/osm/" %>

<osm:osmproduct/>

<osm:osmcar/>

<osm:osm/>

* Rebuild the system (ant clean build) and restart the Hybris server.
* Open web browser, go to HAC → Platform → Update.
* Check the 3 checkboxes under General Settings: Update running system, create essential data, Localize types.

Click on the Update button to update SAP Commerce

# Support from Klarna

Klarna’s service center is responsible for handling operational tasks. The service center is divided into the following two teams: Customer Service and Merchant Support. A customer service workshop can be conducted during the implementation process before going live with our CS team to align the operational processes and ensure customer satisfaction.

**Klarna App**

Klarna provides all customers with the possibility to log into Klarna App via website - <https://app.klarna.com/login> or download the Klarna App (free) on a mobile where the customer can view their payment instructions, pay for their purchase, track delivery updates(when shipping references are updated) and prolong the due dates if they have chosen to pay after delivery.

**Klarna Status Page**

Klarna provides an external facing page - <https://status.klarna.com/>, where merchant can view the health of the Klarna solutions and products. The page also provides the opportunity to subscribe to specific ongoing incident by clicking on the issue subscribe button (via sms/emails)

|  |
| --- |
|  |

**Reporting an Incident**

Merchants should reach their Key Account manager (KAM) to report an incident if they have a suspicion about degraded performances or issues with Klarna's service. The KAM would then be able to report this internally to the incident management team who have established routines to handle and resolve reported incidents. The KAM can request additional information from the person reporting the problem to help internal team ascertain and identify the issue. The KAM may also advice the merchant to follow the updates on the status page if it is an existing incident with on-going updates.

Pre-requisite information to be provided by merchant when reporting incident to help with speedy investigation and resolution:

* Merchant's affected (MIDs/ Klarna API username) and market
* Examples of orders (Klarna order IDs or Klarna session IDs) if available. Additionally, screenshots, timeframe of issues, etc. may be requested.

**Klarna Customer Service**

Klarna Customer Service department answers payment specific questions from customer who made a purchase at the merchant’s site. Customers can do so via the Klarna App on desktop and via mobile.

|  |  |  |  |
| --- | --- | --- | --- |
| **Country** | **Phone** | **Web** | **Opening hours** |
| Austria | +43 (0)720 88 3811 | <https://www.klarna.com/at/kundenservice/> | Monday – Friday:  08:30 – 17:00 |
| Denmark | +45 69 91 8881 | <https://www.klarna.com/dk/kundeservice/> | Monday – Friday:  08:30 – 17:00 |
| Finland | +358 (0)9-425 99 771 | <https://www.klarna.com/fi/asiakaspalvelu/> | Monday – Friday:  09:00 – 18:00 |
| Germany | +49 (0)221 669 501 10 | <https://www.klarna.com/de/kundenservice/> | Monday - Thursday:  08:00 – 22:00  Friday:  08:00 – 17:00  Saturday – Sunday:  10:00 – 17:00 |
| The Netherlands | +31 (0)20 80 82 852 | <https://klarna.com/nl/klantenservice> | Monday – Friday:  08:00 – 17:00 |
| Norway | +47 21 01 89 91 | <https://www.klarna.com/no/kundeservice/> | Monday – Friday:  08:00 – 17:00 |
| Sweden | +46 (0)8 – 120 120 10 | <https://www.klarna.com/se/kundservice/> | Monday - Thursday:  08:30 – 22:00  Friday:  08:20 – 17:00  Saturday – Sunday:  10:00 – 17:00 |
| United Kingdom | Local Rate: +44 (0)20 300 50833  Freephone: +44 (0)808 189 3333 | <https://www.klarna.com/uk/customer-service/> | Support: 24/7 |
| Belgium | +32 (0) 02 808 06 01 | <https://www.klarna.com/be/klantenservice/> | Monday – Friday:  08:00 – 17:00 |
| United States | (844-KLARNA1) +1-844-552-7621) | <https://www.klarna.com/us/customer-service/> | Support: 24/7 |
| Switzerland | [+41445346180|tel:0041445346180] | https://www.klarna.com/en-ch/ Contact Form available in lang: FR,DE,IT  <https://www.klarna.com/en-ch/>  <https://www.klarna.com/fr-ch/contact-form/> | Monday – Friday: 09:00 – 18:00 |

**Klarna Merchant Support**

Klarna's Merchant Support department is available to answer purchase related questions from the merchant.

|  |  |  |  |
| --- | --- | --- | --- |
| **Country** | **Phone** | **Web/Email** | **Opening hours** |
| Austria | +43 720 883 820 | <https://www.klarna.com/at/verkaeufer/haendlersupport/>  Chatten Sie mit uns:(Click “Chat” and provide Merchant ID and other required information)  Contact Form:  <https://www.klarna.com/at/verkaeufer/handler-support-kontaktieren/> | Monday – Friday:  08:00 – 17:00 |
| Denmark | +45 69918883 | <https://klarna.com/dk/support>  Chat with us:(Click “Chat” and provide Merchant ID and other required information)  Contact Form: <https://www.klarna.com/dk/erhverv/kontakt-forhandlersupporten/> | Monday – Friday:  08:30 – 17:00 |
| Finland | +358 (0)9 – 425 99 773 | <https://www.klarna.com/fi/yritys/kauppiastuki/>  Chattaa kanssamme:(Click “Chat” and provide Merchant ID and other required information) | Monday - Friday:  09:00 – 18:00 |
| Germany | +49 (0)221-669 501 30 | <https://www.klarna.com/de/verkaeufer/haendlersupport/>  Chatten Sie mit uns(Click “Chat” and provide Merchant ID and other required information)  Contact Form: <https://www.klarna.com/de/verkaeufer/haendler-support-kontaktieren/> | Monday - Friday:  08:00 – 17:00 |
| The Netherlands | +31 (0)20 80 82 853 | <https://www.klarna.com/nl/zakelijk/webwinkelsupport/>  [webwinkel@klarna.nl](mailto:webwinkel@klarna.nl)  Chat met ons(Click “Chat” and provide Merchant ID and other required information)  Contact Form: <https://www.klarna.com/nl/zakelijk/contact-met-onze-webwinkelsupport/> | Monday - Friday:  08:00 – 17:00 |
| Norway | +47 21 04 96 00 | <https://www.klarna.com/no/bedrift/butikksupport/>  Chat med oss:(Click “Chat” and provide Merchant ID and other required information)  Contact Form:  <https://www.klarna.com/no/bedrift/kontakt-butikksupport/> | Monday - Friday:  08:30 – 17:00 |
| Sweden | +46 (0)8 – 120 120 30 | <https://www.klarna.com/se/foretag/butikssupport/>  Chat med oss:(Click “Chat” and provide Merchant ID and other required information)  Contact Form: <https://www.klarna.com/se/foretag/kontakta-butikssupporten/> | Monday - Thursday:  08:30 - 19:00  Friday:  08:30 - 17:00  Saturday - Sunday:  10:00 – 17:00 |
| United Kingdom | +44 (0)808 189 3444  +44 (0)20 300 50844  (local rate) | <https://www.klarna.com/uk/business/merchant-support/>  Chat with us:  Contact Form: <https://www.klarna.com/uk/business/contact-merchant-support/> | Monday - Friday:  08:00 - 17:00  Saturday:  10:00-18:00  Sunday:  10:00-16:00  Bank Holidays:  Closed |
| Belgium | +32 (0) 02 808 06 02 | <https://www.klarna.com/be/zakelijk/webwinkelsupport/>  Chat met ons:  Contact Form: <https://www.klarna.com/be/zakelijk/contact-met-onze-webwinkelsupport/> | Monday - Friday:  08:00 - 17:00 |
| United States | (844-KLARNA0)  +1-844-552-7620 | <https://www.klarna.com/us/business/merchant-support/>  Chat with us:  Contact Form: <https://www.klarna.com/us/business/merchant-support/> | Monday - Friday:  08:00 - 17:00 |