Klarna for SAP Commerce Cloud

Version 9.0



# Content

[Content 1](#_Toc180141887)

[1. Introduction 4](#_Toc180141888)

[1.1. Audience 4](#_Toc180141889)

[1.2. Supported Environments 4](#_Toc180141890)

[1.3. Restrictions 4](#_Toc180141891)

[1.4. Tested Environments 5](#_Toc180141892)

[1.5. Data Migration 5](#_Toc180141893)

[2. Klarna 9](#_Toc180141894)

[Payments 9](#_Toc180141895)

[2.1. Integration Support 9](#_Toc180141896)

[2.2 Use Cases supported 9](#_Toc180141897)

[2.2.1 Pre-purchase 9](#_Toc180141898)

[2.1.1. Order placement via Klarna Payments 9](#_Toc180141899)

[2.1.2. Order Management 11](#_Toc180141900)

[2.1.3. Card Settlement - Virtual Card Number (VCN) 11](#_Toc180141901)

[2.2. Add-on Installation steps 12](#_Toc180141902)

[2.2.1. Download Add-on 12](#_Toc180141903)

[2.2.2. Update localextensions.xml 12](#_Toc180141904)

[2.2.3. Run installation command 13](#_Toc180141905)

[**2.3.4** **Mandatory configuration changes** 14](#_Toc180141906)

[**2.3.5** **Order Failed Notification** 17](#_Toc180141907)

[**2.3.6** **Mandatory Code changes** 17](#_Toc180141908)

[**2.3.7** **Rebuild the system** 20](#_Toc180141909)

[2.4 Unit Test Implementation 21](#_Toc180141910)

[2.5 Activating Klarna Payments 22](#_Toc180141911)

[2.5.1 Create Klarna Payments configuration 22](#_Toc180141912)

[**2.5.2 Klarna Payments Configuration Fields** 27](#_Toc180141913)

[**2.5.3 Setting Up Virtual Card Numbers (VCN)** 28](#_Toc180141914)

[**2.5.4** **Verifying the VCN Data in Backoffice** 30](#_Toc180141915)

[**2.5.5** **Verifying Klarna Order ID in Backoffice** 32](#_Toc180141916)

[**2.5.6** **Extra Merchant Data (EMD) Setup** 33](#_Toc180141917)

[2.6 Extension Model and Classes 34](#_Toc180141918)

[2.7 Klarna Payment Method Reference 35](#_Toc180141919)

[2.8 KlarnapaymentAPI Method Reference 35](#_Toc180141920)

[2.9 Uninstalling the Add-on 36](#_Toc180141921)

[2.10 Modifying the order confirmation email 37](#_Toc180141922)

[2.11 Troubleshooting 41](#_Toc180141923)

[3 Klarna On-site messaging 42](#_Toc180141924)

[3.5 Installation 42](#_Toc180141925)

[**3.5.1** **Download and Unpack Add-on** 42](#_Toc180141926)

[**3.5.2** **Update localextensions.xml** 42](#_Toc180141927)

[**3.5.3** **Stop SAP Commerce Server** 42](#_Toc180141928)

[**3.5.4** **Set Up Environment** 43](#_Toc180141929)

[**3.5.5** **Install Add-on** 43](#_Toc180141930)

[**3.5.6** **Configure Klarna URLs and Tags** 43](#_Toc180141931)

[**3.5.7** **Modify Code for KOSM Integration** 43](#_Toc180141932)

[**3.5.8** **Rebuild the System** 45](#_Toc180141933)

[**3.5.9** **Update the System** 45](#_Toc180141934)

[3.6 Configuration 45](#_Toc180141935)

[**3.6.1** **Create Klarna OSM Configuration** 45](#_Toc180141936)

[**3.6.2** **Mapping to KOSM to Common Config, Credentials and Base Store** 47](#_Toc180141937)

[**3.6.3** **Klarna OSM Configuration Fields** 49](#_Toc180141938)

[3.7 Uninstalling the OSM Add-on 49](#_Toc180141939)

[4 Klarna Express checkout 52](#_Toc180141940)

[4.5 Use Cases 52](#_Toc180141941)

[**4.5.1** **Express Checkout from Cart / Minicart** 52](#_Toc180141942)

[**4.5.2** **Express Checkout from PDP** 52](#_Toc180141943)

[**4.5.3** **Single Step vs Multistep Checkout** 52](#_Toc180141944)

[**4.5.4** **Finalize and Create Order** 53](#_Toc180141945)

[**4.5.5** **Collect Shipping Address** 54](#_Toc180141946)

[4.6 Installation 54](#_Toc180141947)

[**4.6.1** **Add-on Installation Steps** 54](#_Toc180141948)

[**4.6.2** **Mandatory Code Changes** 55](#_Toc180141949)

[4.7 Activation and configuration 57](#_Toc180141950)

[**4.7.1** **Configure Klarna Express Checkout in Backoffice** 57](#_Toc180141951)

[5 Sign in with Klarna 60](#_Toc180141952)

[5.1 Mandatory Code Changes: 60](#_Toc180141953)

[**5.2** Building and Updating the Database: 61](#_Toc180141954)

[**5.3 Backoffice Reset:** 61](#_Toc180141955)

[**5.4 Activation and Configuration** 61](#_Toc180141956)

[**5.4.1 Setting up Sign-in configuration in Back office:** 61](#_Toc180141957)

[**5.4.2 Sign in With Klarna configuration fields** 63](#_Toc180141958)

[**5.5 How to Sign in with Klarna:** 63](#_Toc180141959)

[5 Support 65](#_Toc180141960)

[**5.1** Reporting an Incident 65](#_Toc180141961)

[**5.1.1** Pre-requisite Information for Reporting an Incident: 65](#_Toc180141962)

[**5.2** Klarna Customer Service 66](#_Toc180141963)

[**5.3** Klarna Merchant Support 68](#_Toc180141964)

# Introduction

This document is a technical guide for installing the Klarna extensions on the SAP Commerce platform.

## Audience

The intended audience for this document is technical specialists involved in integrating *Klarna Payments (KP), Klarna On-site messaging (OSM), Klarna Express checkout (KEC)* and *Sign in with Klarna (SIWK)* into SAP Commerce. These specialists should be familiar with the SAP Commerce platform and the process of installing extensions or add-ons.

## Supported Environments

* **KP**: Designed and tested with version 1905 of SAP Commerce, and supports version 2005.
* **KEC**: Designed and tested with version 2211 of SAP Commerce.
* **SIWK:** Designed and tested with version 2211 of SAP commerce.

## Restrictions

* **Klarna Payments**:
  + **Tax Rates**: Use in stores with percentage tax rates, not absolute taxes. It may work in some cases with absolute taxes but can prevent order placement in others.
  + **Customizations**: Compatible with out-of-the-box SAP Commerce installations. Customizations may affect compatibility and functionality. Individual testing may be required to ensure proper functionality.
* **Klarna Express checkout**: Not a standalone extension and is included in the KP extension from version 7.2 onwards. Installing KP is a prerequisite for KEC. For KP installation, refer to the ‘KP Addon Installation and User Guide’.
* **Sign in With Klarna:** Not a standalone extension and is included in the KP extension from version 9.0 onwards. Installing KP is a prerequisite for SIWK. For KP installation, refer to the ‘KP Addon Installation and User Guide’.

## Tested Environments

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

* UK-GBP-en
* US-USD-en
* DE-EUR-de

## Data Migration

**1.5.1 Migration Script**

1. Login to SAP Commerce Admin Console.
2. Navigate to ‘**Console**’ Tab → Select ‘Scripting Languages’ Menu

A screenshot of a computer

Description automatically generated

1. Under script Type select ‘Groovy’ and run the below script after replacing the below values in the Script with the baseStoreId, marketCountryIsoCode and the marketCountryIsoCode for the merchant specific site.

|  |
| --- |
| String baseStoreId = "apparel-uk";  String marketCountryIsoCode = "GB";  String marketRegionCode = "EU"; |



**1.5.2 Manual Migration Using Back office**

To Ease the Migration Process, we have retained the old Configuration Data for manual migration of data. The old Configuration Data for Klarna Payment, Klarna OSM and Klarna Express Checkout can be found under Klarna Config OLD, Klarna On-site Messaging OLD and Klarna Express Checkout Config OLD.

**Backoffice Reset:**To reflect the new Configurations in Back office, reset needs to be done. Follow the below steps,

1. Login to Backoffice and Press F4. You will see the below screen once the spinner stops.A screenshot of a computer

   Description automatically generated
2. Click on the hamburger icon on the top right corner

A screenshot of a computer

Description automatically generated

1. Select the Reset Everything Option from the Dropdown menu.

A screenshot of a computer

Description automatically generated

1. Once the reset is completed. Click on the Hamburger icon again and Select Refresh from the dropdown menu.  
   A screenshot of a computer

   Description automatically generated

**Klarna Payment configuration:**

Backoffice > Klarna > Klarna Config OLD.

A screenshot of a computer

Description automatically generated

**Klarna On-site Messaging configuration:**

Backoffice > Klarna > Klarna On-site messaging OLD.

A screenshot of a computer

Description automatically generated

**Klarna Express Checkout configuration:**

Backoffice > Klarna > Klarna Express Checkout Config OLD.

A screenshot of a computer

Description automatically generated

# Klarna

# Payments

## Integration Support

The Klarna extension seamlessly integrates the Klarna Payment method across various use cases. It utilizes SAP integration patterns and has been thoroughly tested with SAP's standard B2C accelerators. The integration extends the base models and classes as necessary, which you can review in the "Extension Model and Classes" section.

The Klarna payment integration enhances the common checkout experience, following best practices outlined in Klarna’s developer portal.

## Use Cases supported

### Pre-purchase

**Use Case: Display Klarna Upstream on the Website Before Payment**

The Klarna integration supports configuring and displaying On-site Messaging (OSM) placements on product display pages and the cart page. This includes back-office configuration to enable default themes or customized themes set up by the merchant in the [Klarna Merchant Portal](https://www.klarna.com/us/business/the-merchant-portal/). For detailed instructions, refer to the "[Klarna On-site messaging installation](#_on1esd9x4p49)" section.

### Order placement via Klarna Payments

The Klarna extension can be configured by the storefront to enable specific payment methods 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. For detailed instructions, refer to the "[Activating Klarna Payments](#_muzvrzdmy3rr)" section.

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

The Klarna extension integration uses the standard form and extends it for data entered by the end-user during the SAP Commerce checkout process. The form data updates support both billing and shipping addresses. The address forms are customized to align with the required information for respective Klarna markets. Any person identifiable data is sent to Klarna once the customer authorizes and pays with a Klarna payment method. Additional data may be requested in a widget (Klarna modal) as part of the payment method authorization, depending on the market and Klarna product.

**Use Case: Update Basket**

The Klarna extension integration ensures that updates made to the cart are applied to the Klarna payment session. Updates to the cart that trigger a Klarna session update will lead to re-authorization when the customer checks out with a Klarna payment method. Any additional authorization will return an updated authorization token for the new cart amount.

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

The extension handles Klarna authorization declines gracefully. If a payment method is declined with a recoverable error, the payment methods are displayed again, and the customer is redirected to the payment method selection step. The customer has the option to update their details and re-select Klarna payment methods (e.g., Response: show\_form: false, approve: true).

**Use Case: Order Creation**

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

**Use Case: Pending Orders**

The extension supports processing pending order statuses when the functionality is enabled. Pending order statuses are returned to the configured merchant notification URL, such as FRAUD\_RISK\_ACCEPTED, FRAUD\_RISK\_ACCEPTED, and FRAUD\_RISK\_REJECTED. The order fraud status is verified using the Order Management API. The order process is suspended until the application receives a pending order update from Klarna. For example, if the notification URL receives an update with the status FRAUD\_RISK\_ACCEPTED, then order processing will resume.

**Use Case: Confirmation Email**

The extension enables the sending of order confirmation emails with a “Paid by Klarna” message in the email body. Additional information regarding the Klarna App and customer support links are included for customers to find more information about the payment process and status. For more details, refer to the "[Modifying the Order Confirmation Email](#_gs5ikl5mhmz1)" section.

### Order Management

**Use Case: Post Purchase Update - Shipment of Goods / “Capture” / “Card Settlement”**

The extension does not support capture, cancellation, or other post-order placement use cases out of the box. These use cases can be enabled or extended using the klarnapaymentapi extension, which includes support for the Klarna Payment and Order Management API. A sample capture implementation is provided as part of the accelerator Fulfillment under the TakePaymentAction for Merchant Reference.

### Card Settlement - Virtual Card Number (VCN)

If standard order management is not feasible for a Klarna integration, Klarna’s merchant card service-based virtual card solution may be utilized. When an order is settled via the Virtual Card-based settlement approach, the merchant must work with their Payment Service Provider (PSP) in collaboration with the Klarna Merchant Success team to ensure proper handling of VCN-based settlement post-purchase use cases.

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

When a customer places an order that is created and accepted by Klarna, the addon integration creates a virtual card-based settlement using the Merchant Card Services (MCSv3) API and creates an 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. Once the order has been fulfilled, the card funds should be captured. For delays in capture or other special use cases, please consult with the Klarna Key Account Manager in advance. While Klarna is the original payment method for the order, the order amount will be settled with a virtual card instead of a direct bank account transfer.

A virtual card settlement request is triggered for new orders or those with a 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, no action is taken on the order until Klarna's push notification updates the “fraud\_status” to FRAUD\_RISK\_ACCEPTED. The virtual card issued is limited to one single successful authorization per order for a given merchant ID. |

For more details, refer to the [Setting Up Virtual Card Numbers (VCN)](#_o3ro4vz5txwf) section.

## Add-on Installation steps

### Download Add-on

1. **Get the Add-on Archive:** Obtain the add-on from [Klarna’s GitHub repository](https://github.com/klarna/SAP-Commerce-klarna-payments).
2. **Unpack Archive:** Extract the archive and place the following folders into the custom folder of the SAP Commerce Suite (<HYBRIS\_HOME>/bin/custom):
   * klarnapaymentaddon
   * klarnapayment
   * klarnapaymentapi
   * klarnapaymentbackoffice
   * For SAP Commerce B2B implementations, also copy klarnapaymentb2baddon to the custom folder.

### Update localextensions.xml

**Verify Add-on Support Extension:** Ensure <extension name="addonsupport" /> is present in the localextensions.xml file.

**Add Klarna Extensions:**

* <extension name="klarnapaymentaddon" />
* <extension name="klarnapayment" />
* <extension name="klarnapaymentbackoffice" />
* <extension name="klarnapaymentapi" />
* For B2B, add <extension name="klarnapaymentb2baddon" />



**Verify Target Storefront Extension:** Ensure the target storefront extension is listed.

**Check Required OOB Extensions:** Verify the presence of other out-of-box extensions mentioned in the localextensions.xml file available in the archive.

### Run installation command

**Stop SAP Commerce Server:**

* If the server is running in embedded mode, press Ctrl+C to stop.
* If the server is running in service mode, use the following command to stop:
  + Windows: hybrisserver.bat stop
  + Unix: ./hybrisserver.sh stop

**Set Up Environment:**

* Navigate to <HYBRIS\_HOME>/bin/platform and run:
  + Windows: setantenv.bat
  + Unix: ./setantenv.sh

**Install Add-on:**

For general installation, run the following command:

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

For B2B installation, run:

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

|  |  |
| --- | --- |
|  | **Note:**   * In some versions of SAP Commerce, you may need to install liveeditaddon and b2ccheckoutaddon. * Depending on the project, the storefront name may differ from the default name provided by SAP Commerce. |

### **Mandatory configuration changes**

**Configuration for Callback URLs**

To ensure proper support from Klarna, add the following configurations in the project.properties file of the klarnapayment extension:

* **Confirmation URL:** This URL is where customers will be redirected after the payment confirmation.

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

* **Notification URL:** This URL is for receiving fraud notifications.

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

* **Authorization Callback URL:** This URL is for receiving authorization call back.

|  |
| --- |
| klarnacheckout.merchant.url.authorization=https://<your\_host>/klarna/payment/auth-callback |

If Klarna Payments (KP) is activated for multiple sites simultaneously, use the above configurations. If only certain sites will use KP, specify them individually:

|  |
| --- |
| 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  klarnacheckout.merchant.url.authorization.<site\_uid>=https://<your\_host>/klarna/payment/auth-callback |

**Configuration for Skipping CSRF Verification**

Add the following configuration to the local.properties file to skip CSRF verification when receiving a callback from Klarna:

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

#### Email Server Configuration

Configure the email server for sending emails from SAP Commerce by adding the following properties to 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**

For efficient support and issue identification by Klarna, add the following configurations:

|  |
| --- |
| shoporplatform=<your\_platform> # Eg: SAP\_Comm  platformversion=<your\_platform\_version> # Eg: 1905  modulename=<your\_module> # Eg: KP  moduleversion=<your\_module\_version> # Eg: 7.0 |

#### Klarna Market and Klarna Region Configuration for site

Configure the Market and the Region specific to each site:

|  |
| --- |
| klarna.market.country.<site\_id>=<market\_code>  # E.g. klarna.market.country.apparel-uk=GB  klarna.market.region.<site\_id>=<region\_code>  # E.g. klarna.market.region.apparel-uk=EU |

**Klarna Market and Regions**

|  |  |  |
| --- | --- | --- |
| **Market** | **Country Code**  **for Market** | **Region** |
| Australia | AU | OC |
| Austria | AT | EU |
| Belgium | BE | EU |
| Canada | CA | NA |
| Czech Republic | CZ | EU |
| Denmark | DK | EU |
| Finland | FI | EU |
| France | FR | EU |
| Germany | DE | EU |
| Greece | GR | EU |
| Hungary | HU | EU |
| Ireland | IE | EU |
| Italy | IT | EU |
| Mexico | MX | NA |
| Netherlands | NL | EU |
| New Zealand | NZ | OC |
| Norway | NO | EU |
| Poland | PL | EU |
| Portugal | PT | EU |
| Romania | RO | EU |
| Slovakia | SK | EU |
| Spain | ES | EU |
| Sweden | SE | EU |
| Switzerland | CH | EU |
| United Kingdom | GB | EU |
| United States | US | NA |

**Currency Conversion**

If multiple currencies are applicable on the merchant site, ensure that the out-of-box settings for the conversion factor and delivery cost are properly configured. These values are used to send the cost in the purchased currency converted value to Klarna.

Adjust these configurations based on your specific project requirements to ensure smooth operation and support from Klarna.

### **Order Failed Notification**

When a Klarna order is successfully processed but fails to place an order in Hybris, an email notification is sent to the merchant. The recipient's email address is configured in the Klarna settings within the back office. The email content for this notification is managed using the FailedOrder\_Email\_Template.impex file, located in \klarnapayment\resources\klarnapayment\email\_impex\. This impex file must be loaded after the server starts. Merchants have the flexibility to modify the email content, specifically the CMS component OrderFailedEmailBody, either by editing the impex file or through the back office at a later time.

### **Mandatory Code changes**

To integrate Klarna into your SAP Commerce storefront, apply the following code changes. For B2B implementations, ensure these changes are applied to the corresponding B2B storefront.

1. **Adding Klarna Payment Tag:** Insert the following tag into the Merchant Payment JSP (e.g., silentOrderPostPage.jsp in SAP Commerce Out of the Box).

|  |
| --- |
| <%@ taglib prefix="klarnapayment" tagdir="/WEB-INF/tags/addons/klarnapaymentaddon/responsive/" %>  …  <!-- Insert the below snippet after <ycommerce:testId code="paymentDetailsForm">-->  <klarnapayment:selectpayment/> |

1. **Modifying CheckoutOrderSummary Tag:** Update the checkoutOrderSummary.tag file (located at yacceleratorstorefront/web/webroot/WEB-INF/tags/responsive/checkout/multi/checkoutOrderSummary.tag in SAP Commerce OOB) by adding the following.

|  |
| --- |
| <%@ taglib prefix="klarnacheckout" tagdir="/WEB-INF/tags/addons/klarnapaymentaddon/responsive/checkout" %>  <!-- Insert the below snippet after order:appliedVouchers order="${cartData}" -->  <klarnacheckout:paymentInfo cartData="${cartData}" paymentInfo="${cartData.paymentInfo}" showPaymentInfo="${showPaymentInfo}" /> |

1. **Customizing Payment Info Tag:** If the merchant uses a customized paymentInfo.tag, include the hidden parameters from the addon paymentinfo.tag into the customized paymentInfo.tag.
2. **Adding Date of Birth Field:** If the billing address requires a date of birth field, add the following snippet to kpBillingAddressFormElements.tag.

|  |
| --- |
| <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. Updating JavaScript for Form Names: In acc.klarnapayment.js, update the form name from silentOrderPostForm to the merchant payment page form name if it differs. For example:

|  |
| --- |
| 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);  }  });  } |

Example form name in silentOrderPostPage.jsp:

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

1. **Updating JavaScript for Submit Button Class:** In acc.klarnapayment.js, update the submit button class from submit\_silentOrderPostForm to the merchant payment page button name if it differs:

|  |
| --- |
| bindPaymentsubmit: function () {  $('.submit\_silentOrderPostForm').click(function () {  // your logic here  });  } |

Example submit button class 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. **Updating Default Checkout URL:** Change the constant in KlarnaPaymentActiveFilter to the default checkout URL in the merchant application:

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

1. **B2B Implementation - Payment Type Selection:** For B2B implementations using b2bacceleratoraddon, modify the paymentTypeForm.tag file to display Klarna Payment Type by replacing.

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

With:

|  |
| --- |
| <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> |

### **Rebuild the system**

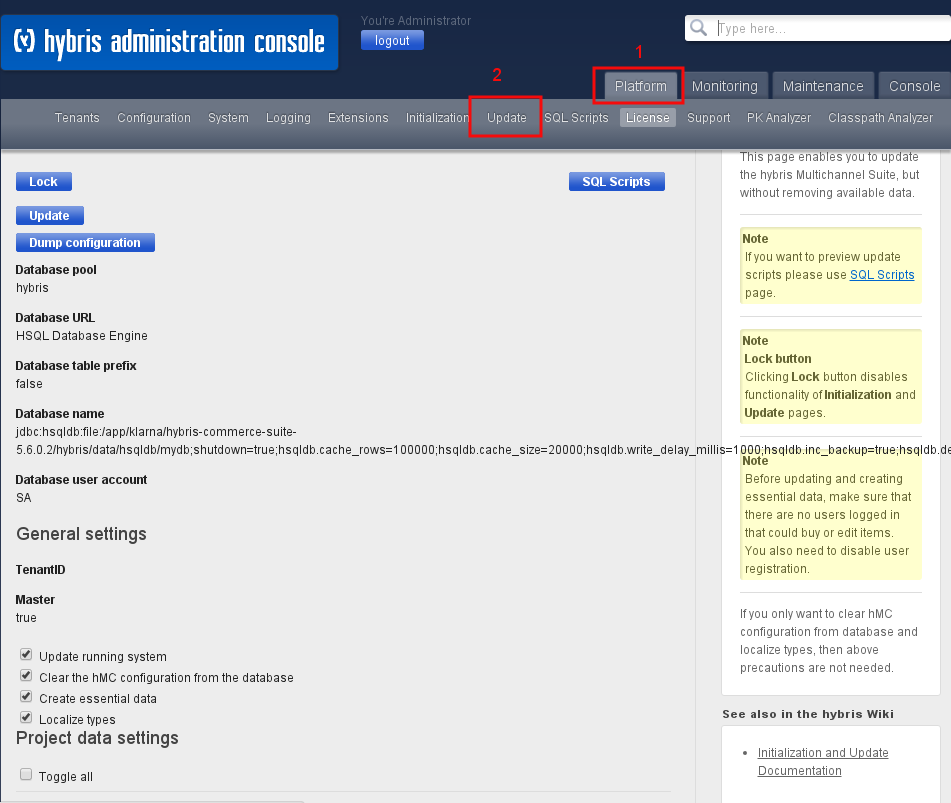
**Rebuild the system**

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

**Update the System**

After rebuilding the system, you may need to perform a full initialization through the Hybris Administration Console (HAC) if this is the first installation of SAP Commerce. If you have already completed a full initialization, proceed with updating your SAP Commerce system as follows:

1. **Open the Hybris Administration Console (HAC):** Open your web browser and navigate to the HAC URL.
2. **Navigate to the Update System Page:** Go to 1 Platform → 2 Update.

****

1. **Update System Settings:**
   1. Under **General Settings**, check the following boxes:
      1. Update running system
      2. Clear the HMC configuration from the database (applicable for versions prior to 5.7 only)
      3. Create essential data
      4. Localize types
   2. Further down, under **Project data settings**, check the following boxes:
      1. *klarnapaymentaddon*
      2. *klarnapayment*
      3. *klarnapaymentapi*
      4. *klarnapaymentbackoffice*
   3. For B2B implementations, also check the *klarnapaymentaddon* checkbox.
2. **Update the System:** Click the **Update** button to apply the changes to the SAP Commerce system.

## Unit Test Implementation

**JUnit Tests**

JUnit test classes are located in the testsrc folder of both the klarnapayment and klarnapayment extensions. Mock test classes for all supported Klarna API calls are available in the klarnapaymentapi extension.

**Integration Test**

The KlarnaClientConnectionTest integration test class is implemented in the klarnapaymentapi extension. This class is used to test the connection to the Klarna server after initial tests.

**Steps:**

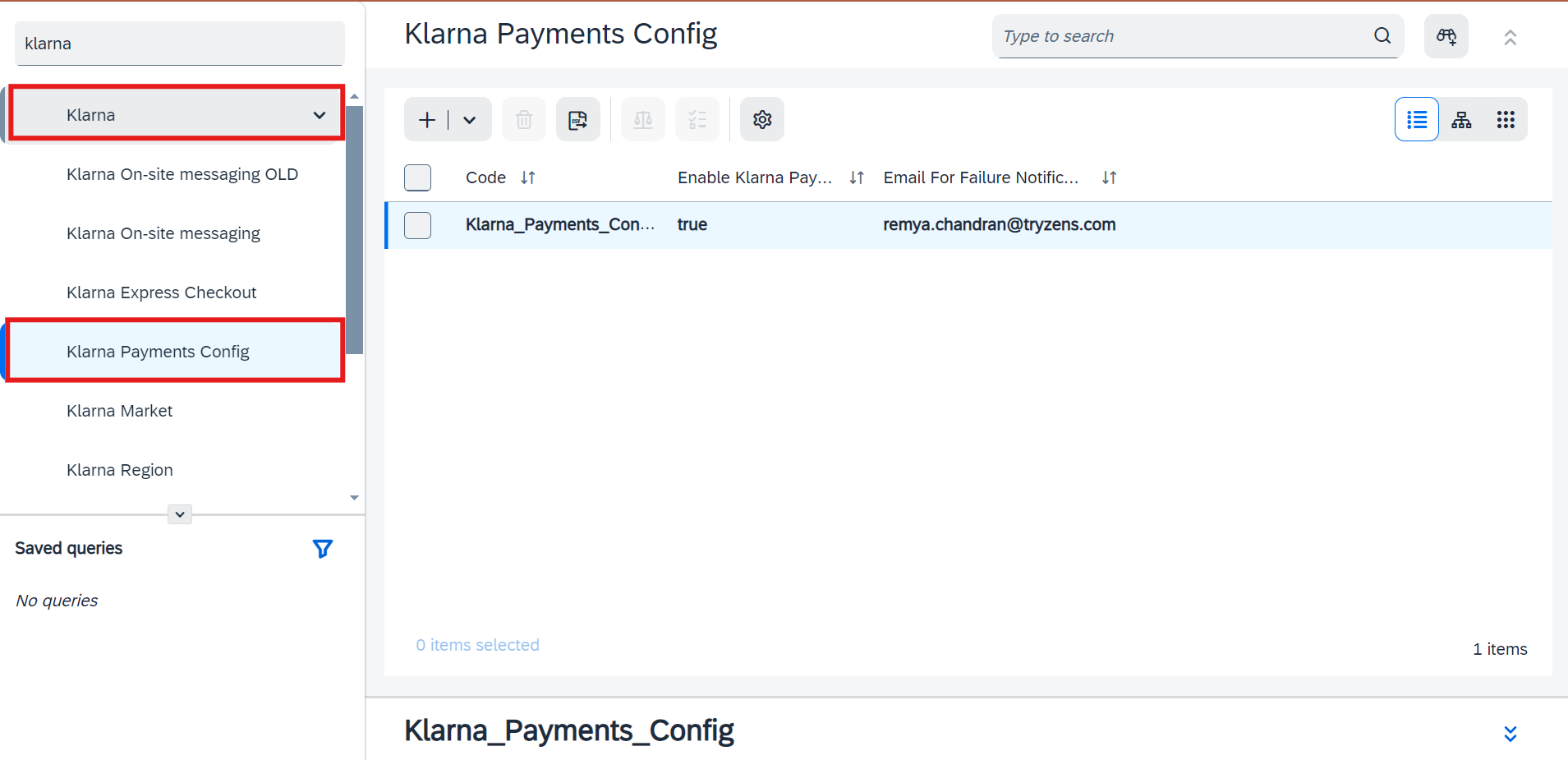
1. **Update Credentials**:
   * Modify the KlarnaClientConnectionTest class with valid merchant test (playground) credentials. This helps identify any issues with authentication or payment session creation during server startup.
2. **Verify Connection**:
   * Upon starting the server, look for the log message: "Klarna Integration Connection Test Successful". This confirms that SAP Commerce has successfully connected to Klarna.
3. **Production Use**:
   * If using this feature in production, ensure to remove the Klarna production API credentials from the test class to maintain security.

## Activating Klarna Payments

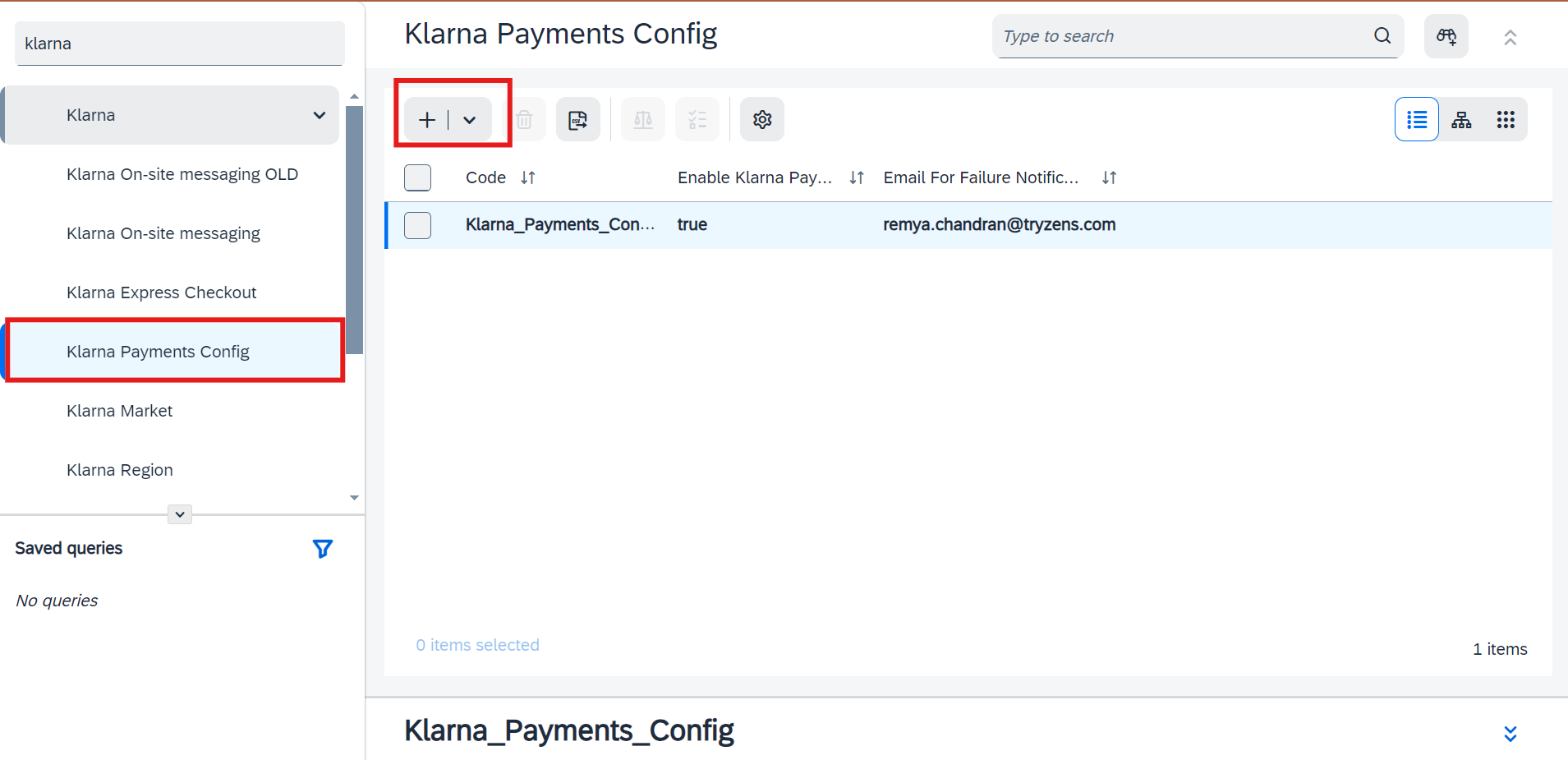
After successfully installing klarnapaymentaddon, you can log in to SAP Commerce Backoffice to set up a Klarna configuration and assign it to specific sites.

### Create Klarna Payments configuration

1. **Navigate to Klarna Payment Configuration:** Go to SAP Commerce Backoffice → Klarna → Klarna Payments Config.

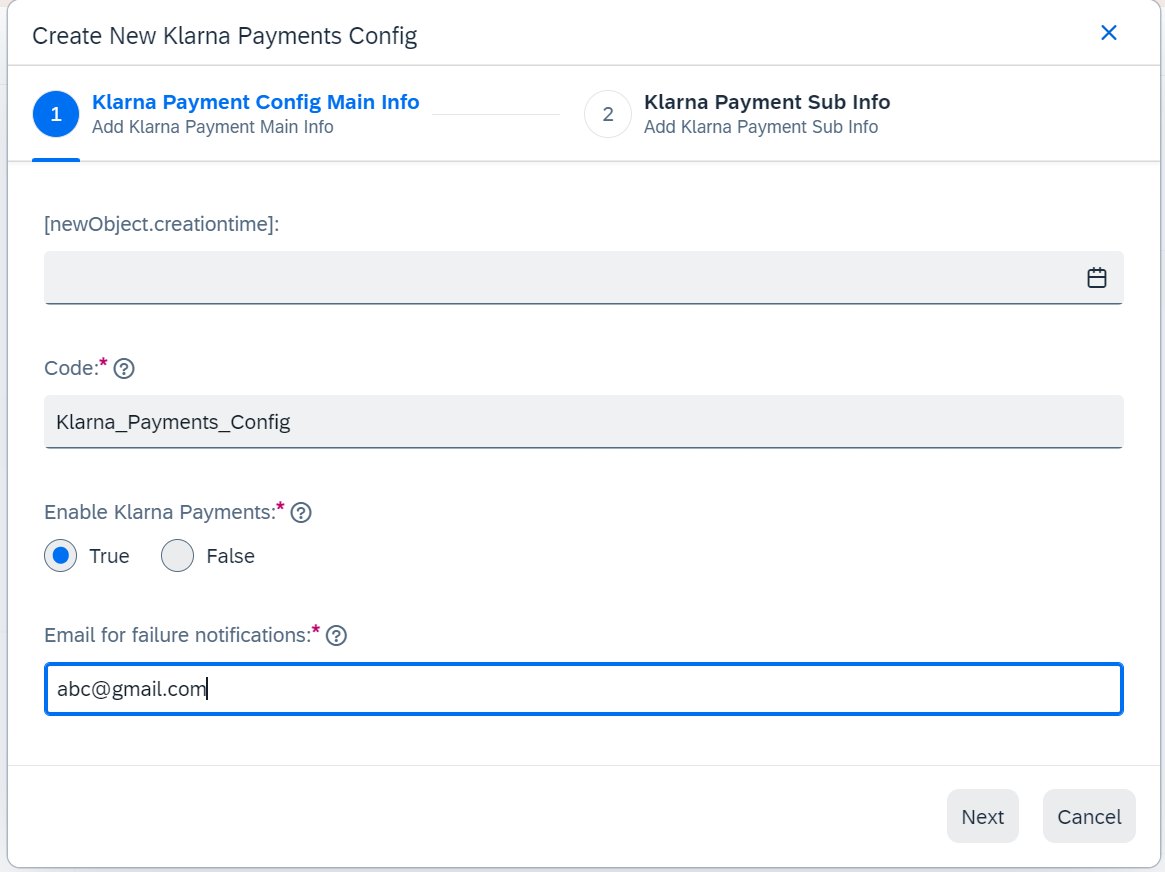


1. **Create a New Klarna Configuration:** Choose the option to create a new Klarna configuration.

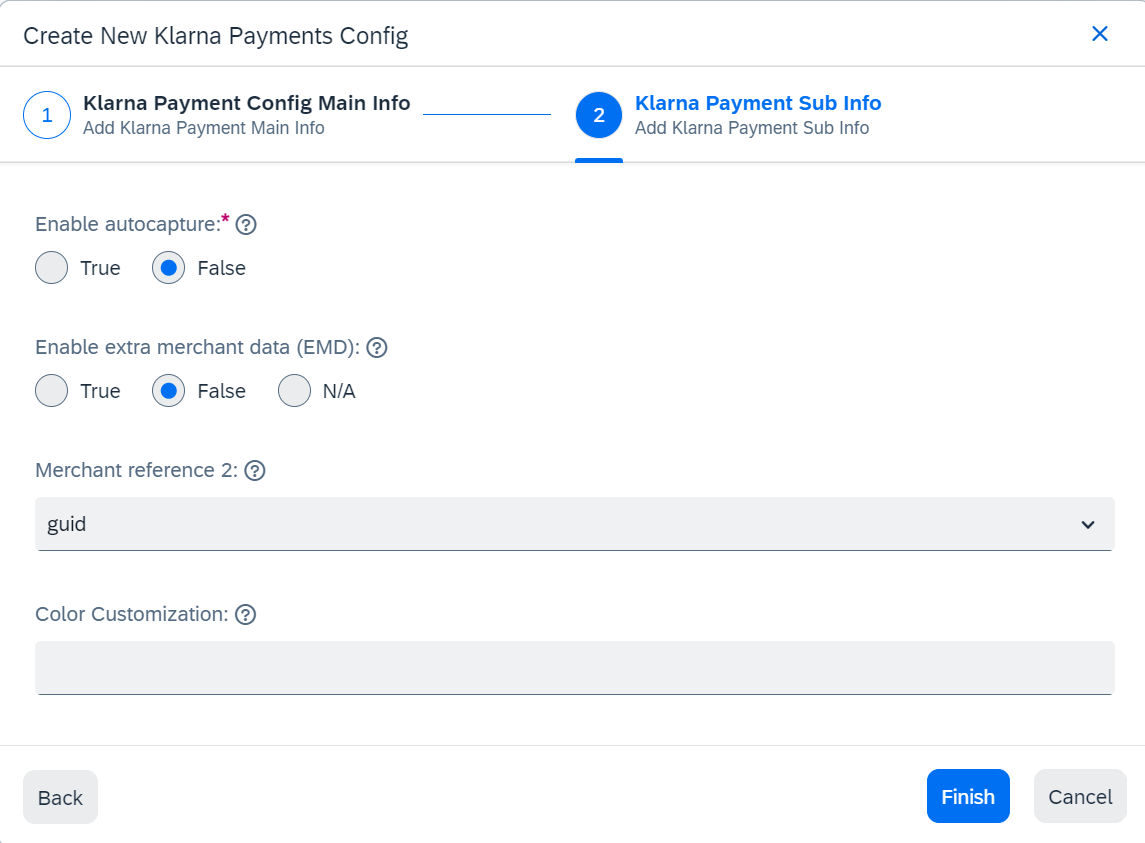


1. **Fill in the Klarna Payments Specific Fields:**

**Step1:** Fill in the necessary data under the fields in step 1: Code, Enable Klarna Payments and Email for Failure Notifications

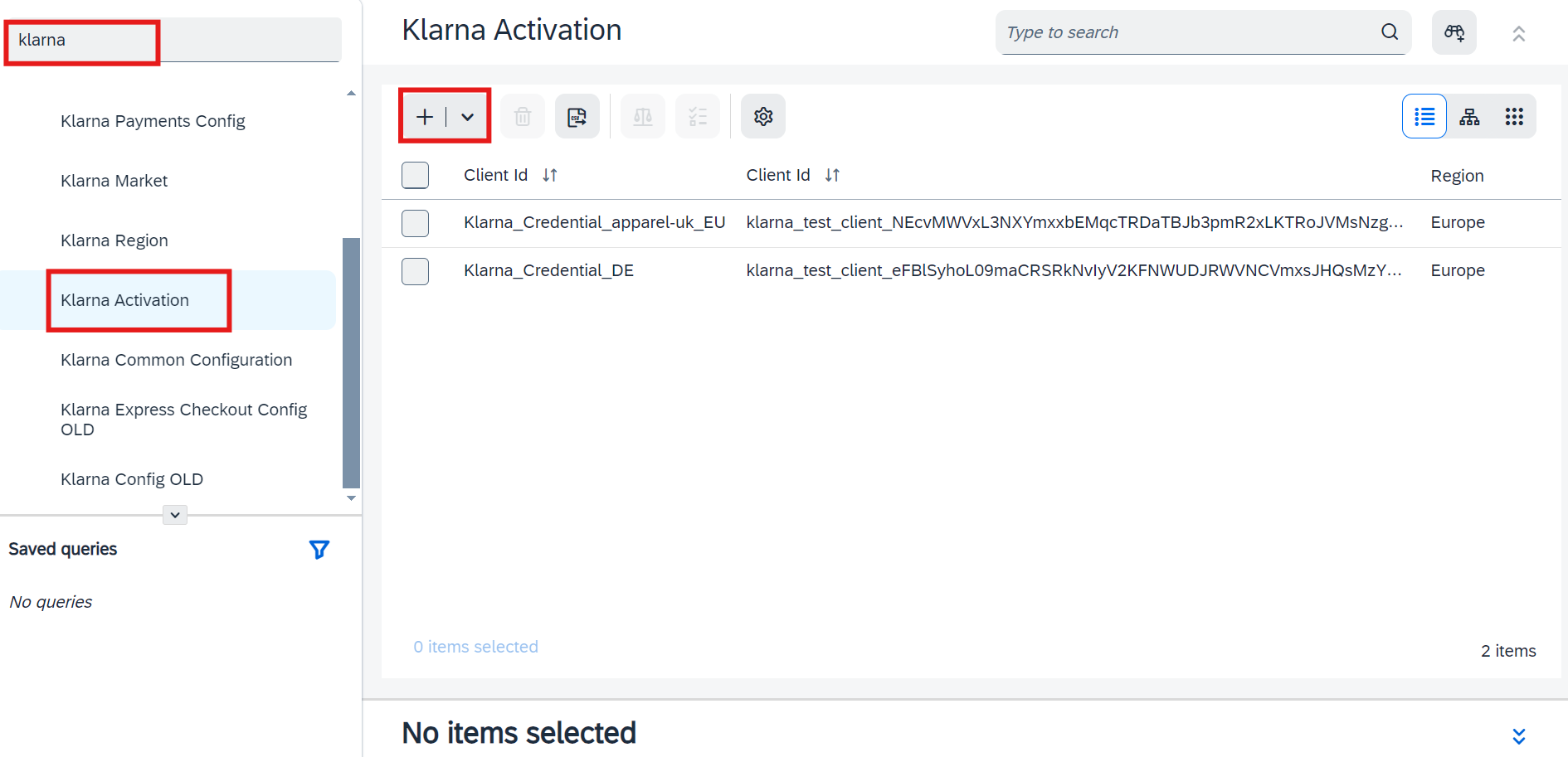


**Step 2:** Navigate to the next step after clicking next which appears after providing mandatory fields and provide the information for the remaining fields: Enable AutoCapture, Enable Extra merchant Data, Merchant Reference 2 and color customization.

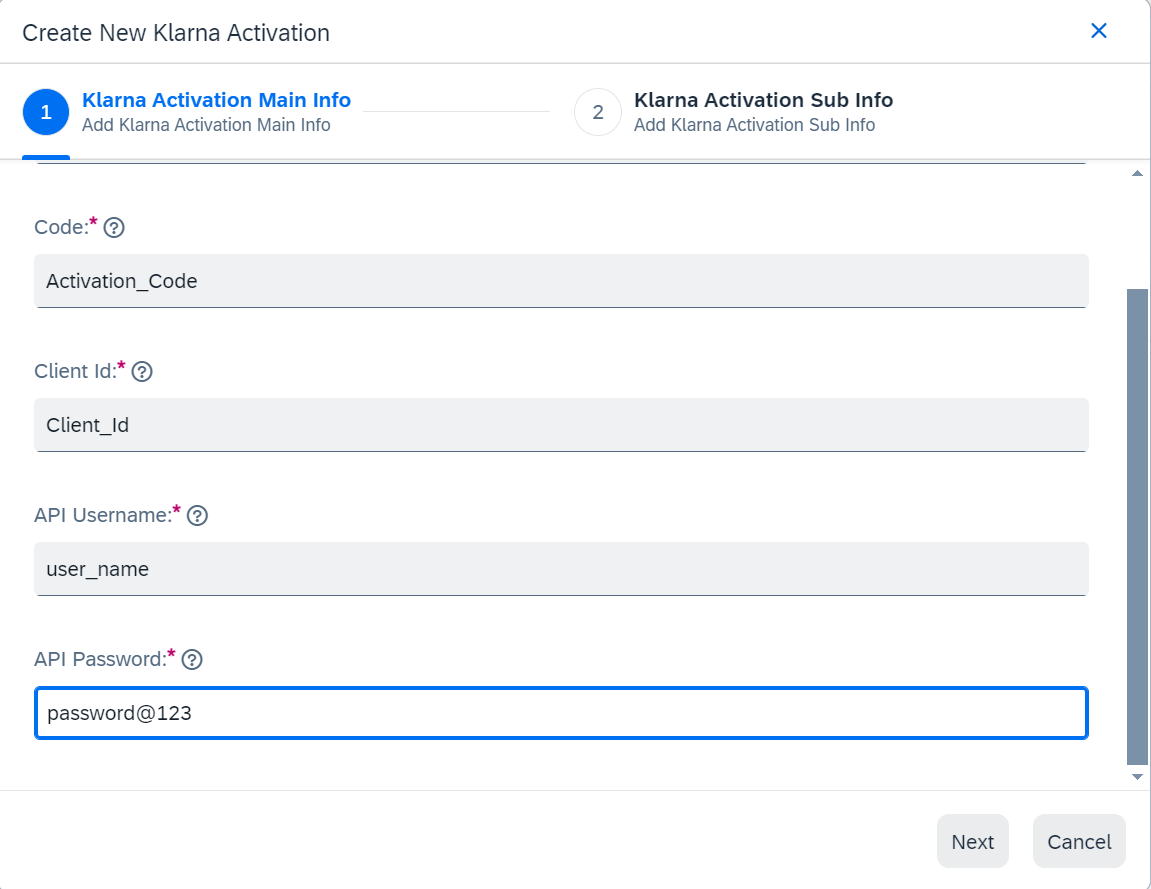


1. **Create Credential Data for Klarna Payments:**

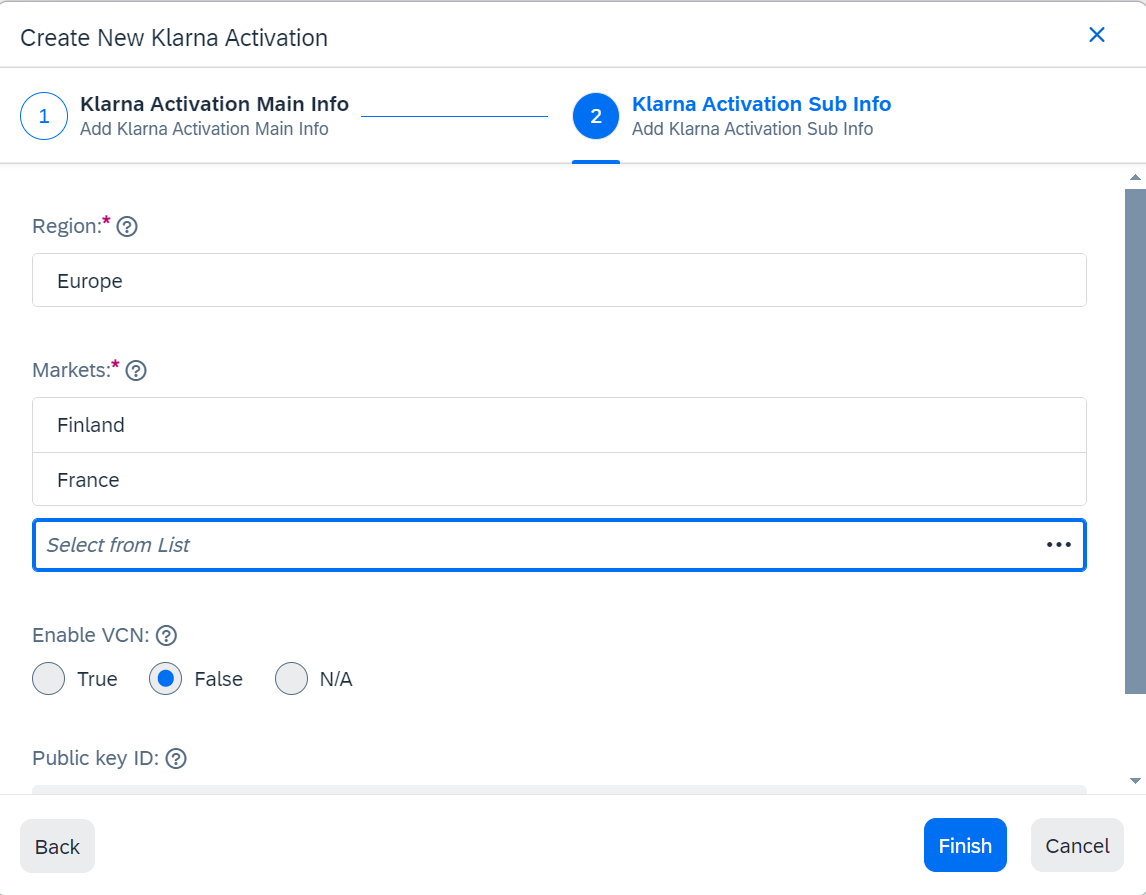
In SAP Commerce Backoffice, Navigate to Klarna → Klarna Activation, on the left corner below the section name ‘Klarna Activation’ click on the ‘+’ sign.

****

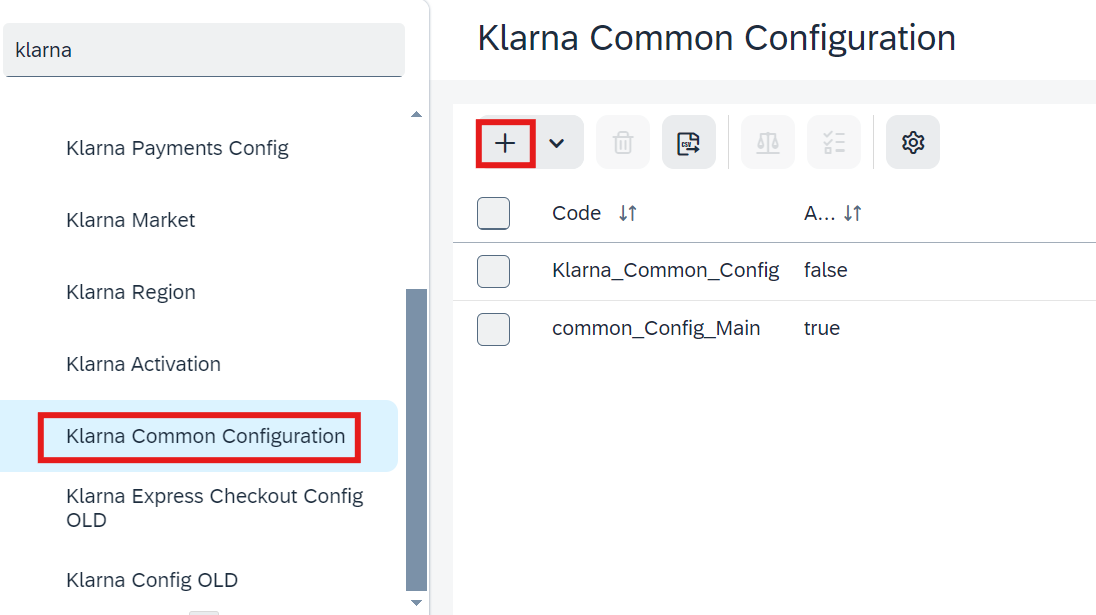
Provide the Credential information necessary for Klarna Payments: Code, Client ID, API username and API Password.

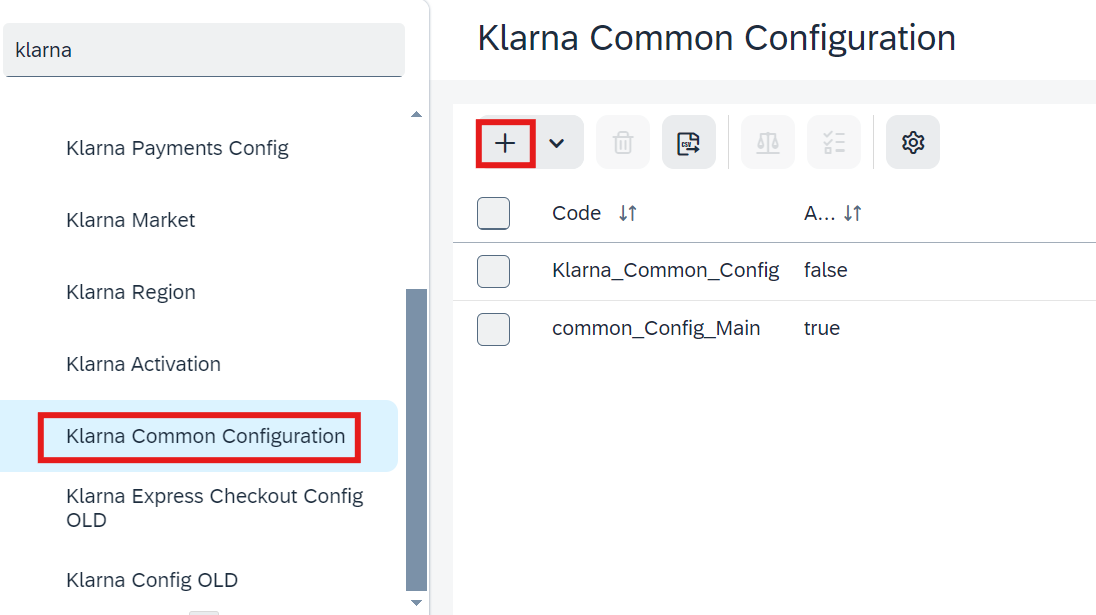


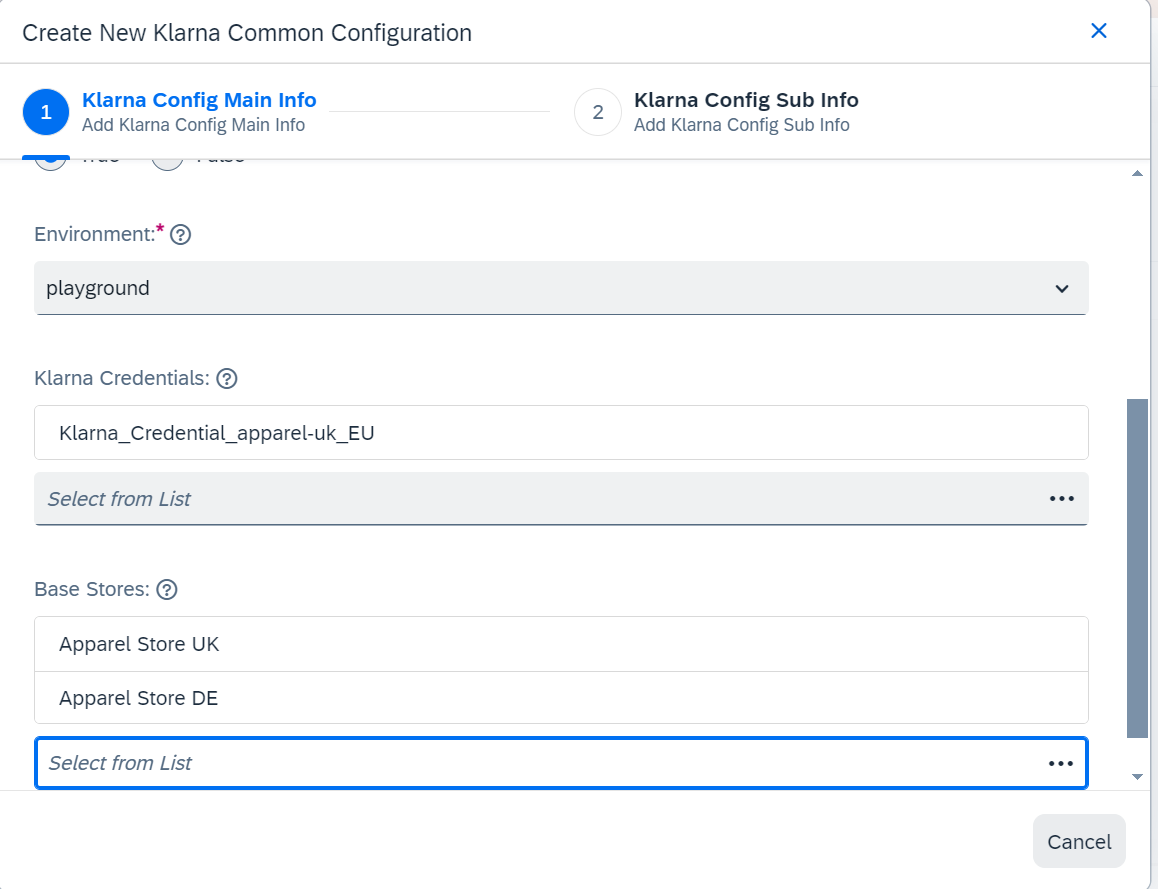
Once the mandatory fields are provided, click on the next button and provide the information for Region, Markets, Enable VCN and public Key ID.



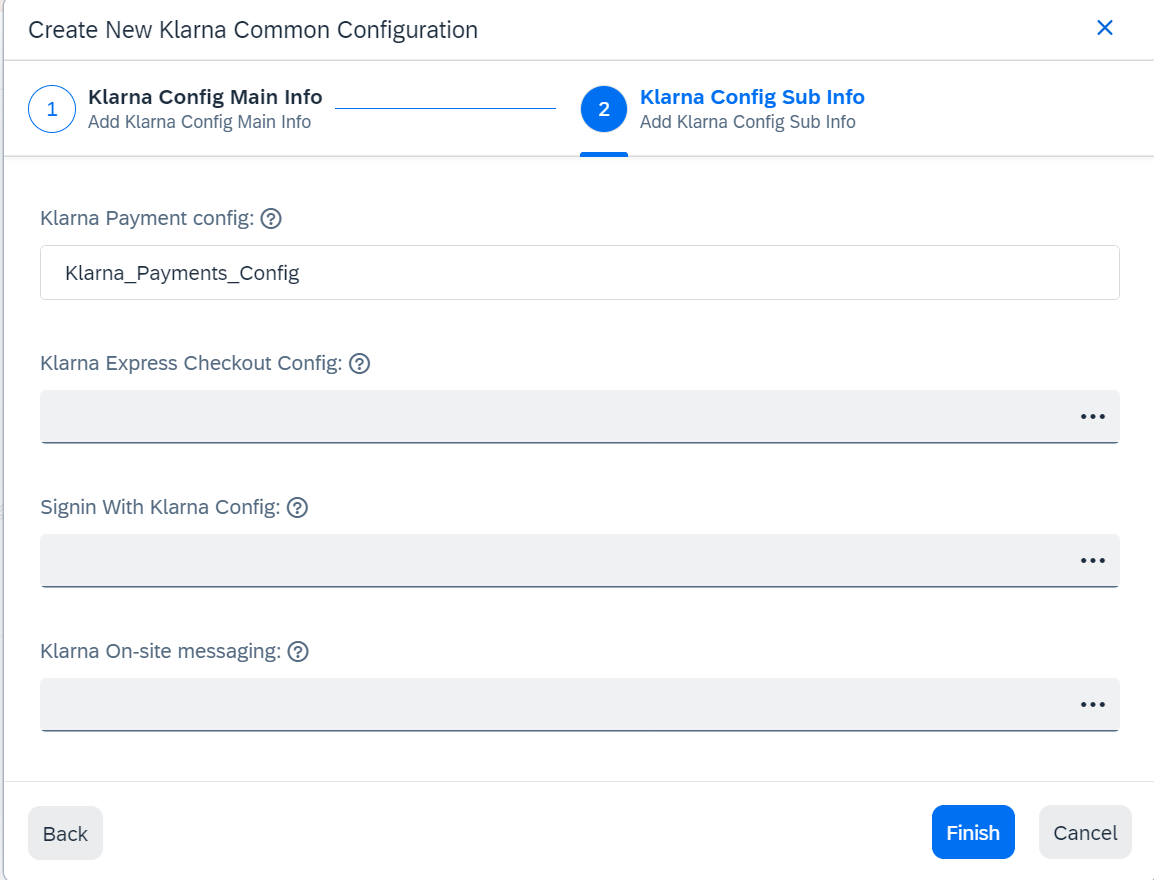
1. **Mapping the Credentials and Klarna Payments to the Klarna Common Configuration:**

In SAP Commerce Backoffice, Navigate to Klarna → Klarna Common Configuration, Click on the **** icon.

****

Provide the necessary information for: code, Active, Environment, Klarna Credentials, Base Store.  


Once the mandatory fields are provided the next button will be enabled, click on the next button and select the created Klarna Payment Config.



### **2.5.2 Klarna Payments Configuration Fields**

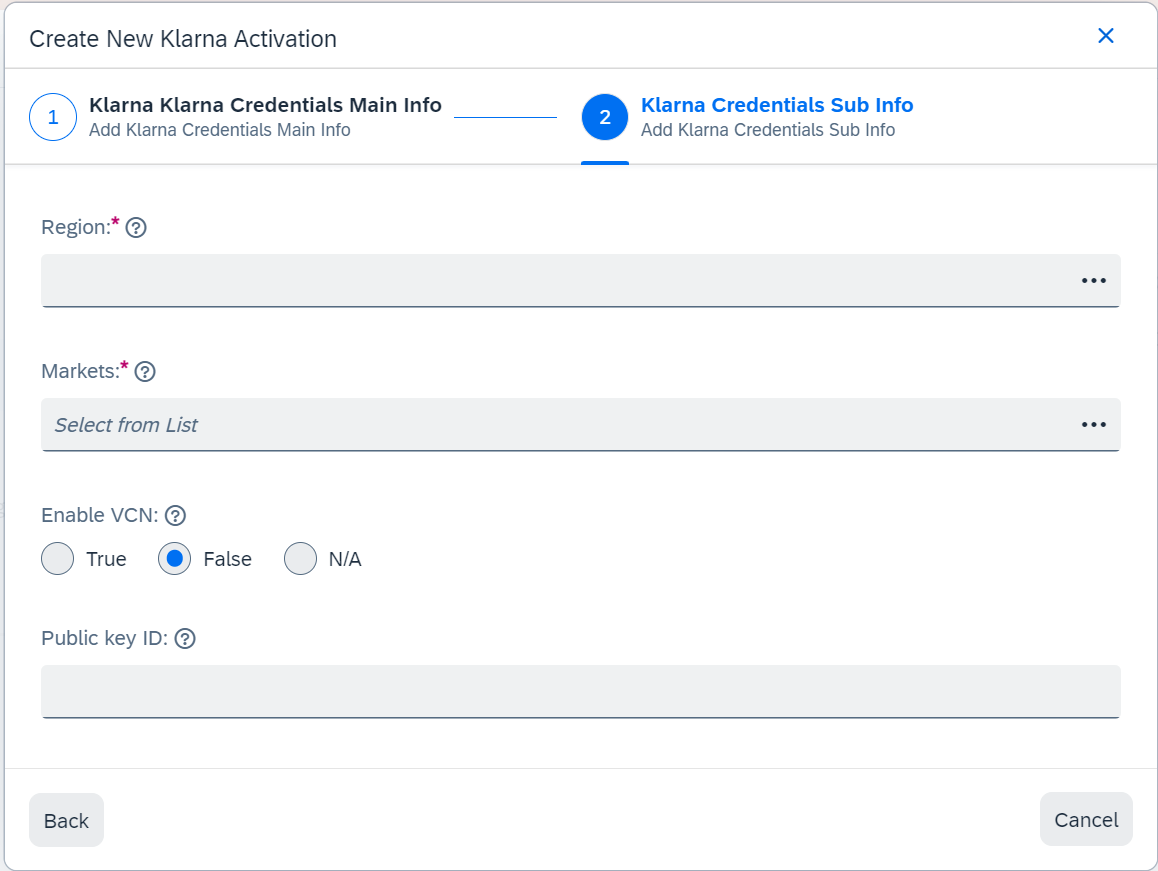
|  |  |  |
| --- | --- | --- |
| **Attribute** | **Mandatory** | **Description** |
| Klarna API username | Yes | Generated in Klarna Merchant Portal. |
| Klarna API username | Yes | Generated in Klarna Merchant Portal. |
| Region | Yes | Each MID supports one of three end point types: EUROPE, NORTH\_AMERICA and OCEANIA. Select the appropriate one. |
| Markets | Yes | Countries that are supported under the Regions. |
| Environment | Yes | Each MID supports one of the two end point modes: TEST (playground) and LIVE (production). Select the appropriate one. |
| 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 | Allows merchants to trigger auto capturing during Order Placement. If VCN is enabled, AutoCapture settings will not be relevant. |
| Enable VCN | No | To enable VCN based settlement for merchant integration |
| 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 |

|  |  |
| --- | --- |
|  | Note:  To ensure that the Klarna Payments (KP) iframe and the payment method text within it are clearly visible, please select a minimum width for your checkout page that works well with the majority of your mobile users' screen sizes. It is recommended to apply the same minimum width to the billing page section. This will provide a consistent and optimal user experience as the browser width is adjusted.  By maintaining a uniform minimum width, you can ensure that all critical payment information remains accessible and legible, thereby enhancing the overall usability of the checkout process for mobile users. |

### **2.5.3 Setting Up Virtual Card Numbers (VCN)**

To use the VCN settlement for Klarna orders, follow these steps:

1. **Enable VCN in Klarna Configuration:**
   * Go to **SAP Commerce Backoffice**.
   * Navigate to **Klarna** → **Klarna Activation**.
   * Select the ‘Enable VCN’ option under Klarna Activation.



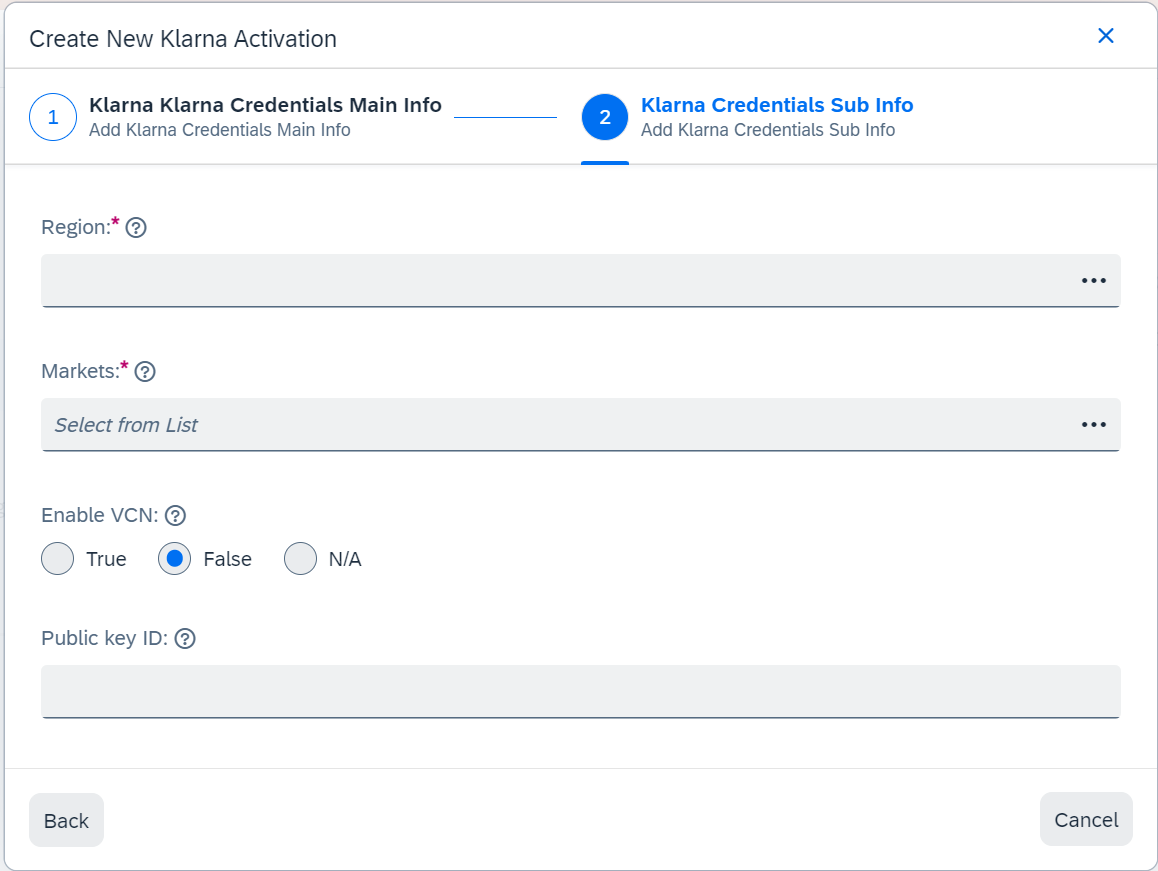
1. **Generate a 4096-bit RSA Key Pair:**
   * Generate the RSA key pair using the following OpenSSL commands:

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

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

* + This will create two files: public\_key.pem and private\_key.pem.

1. **Configure Klarna with RSA Keys:**
   * Set the Public Key id field of the Klarna Activation



**Example contents of the public\_key.pem file:**

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----- |

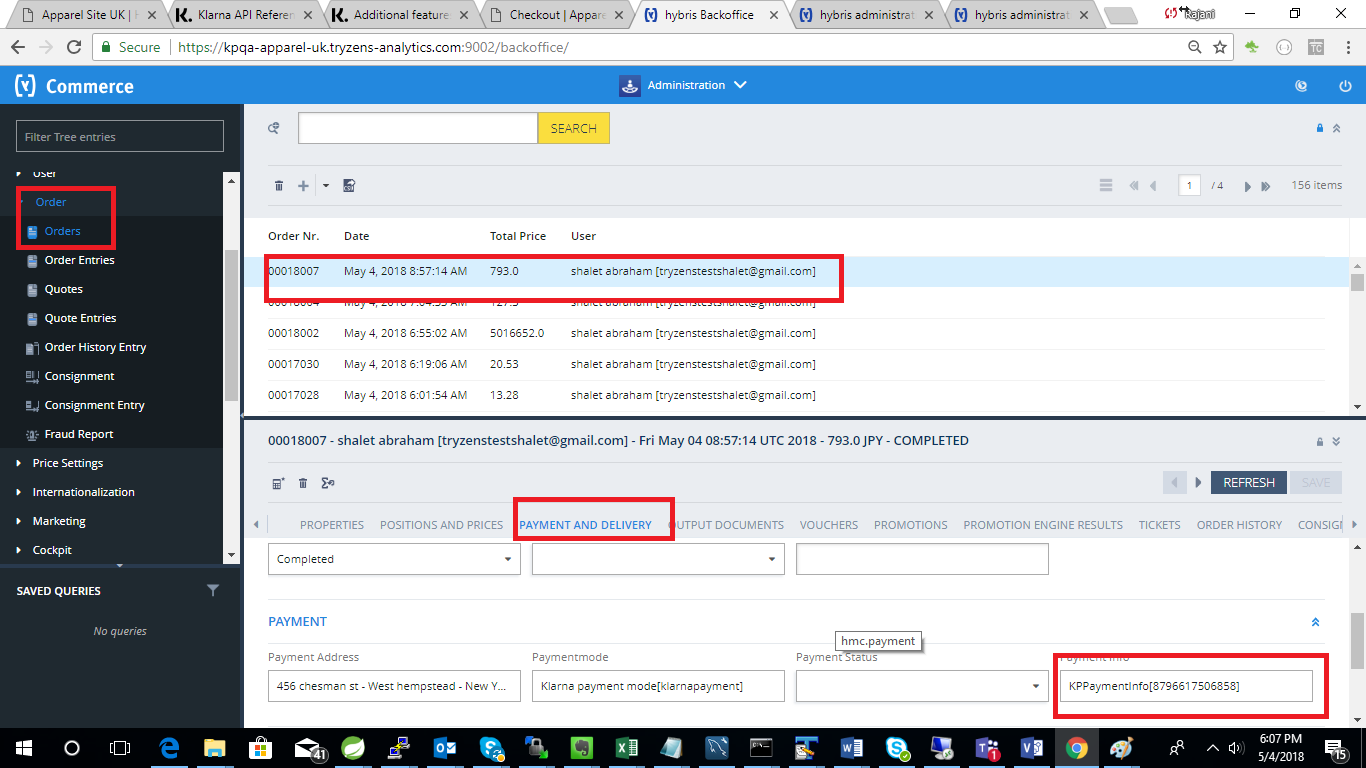
The virtual card decryption is only a sample implementation in the KP extension. Merchants must use this as a reference for development and never store decrypted virtual card numbers (VCN) or CVV (PCI data) in the production environment.

|  |  |
| --- | --- |
|  | **Important! Do not save decrypted PCI data (Incl. VCN/CVV) on the server!**   * Ensure PCI-DSS compliance by handling card data securely in coordination with required partners, Payment Service Providers (PSP), and Acquirers. * Review the order export details required for virtual card-based Klarna orders in advance. * Expunge any historical decrypted PCI data, regardless of the validity date. |

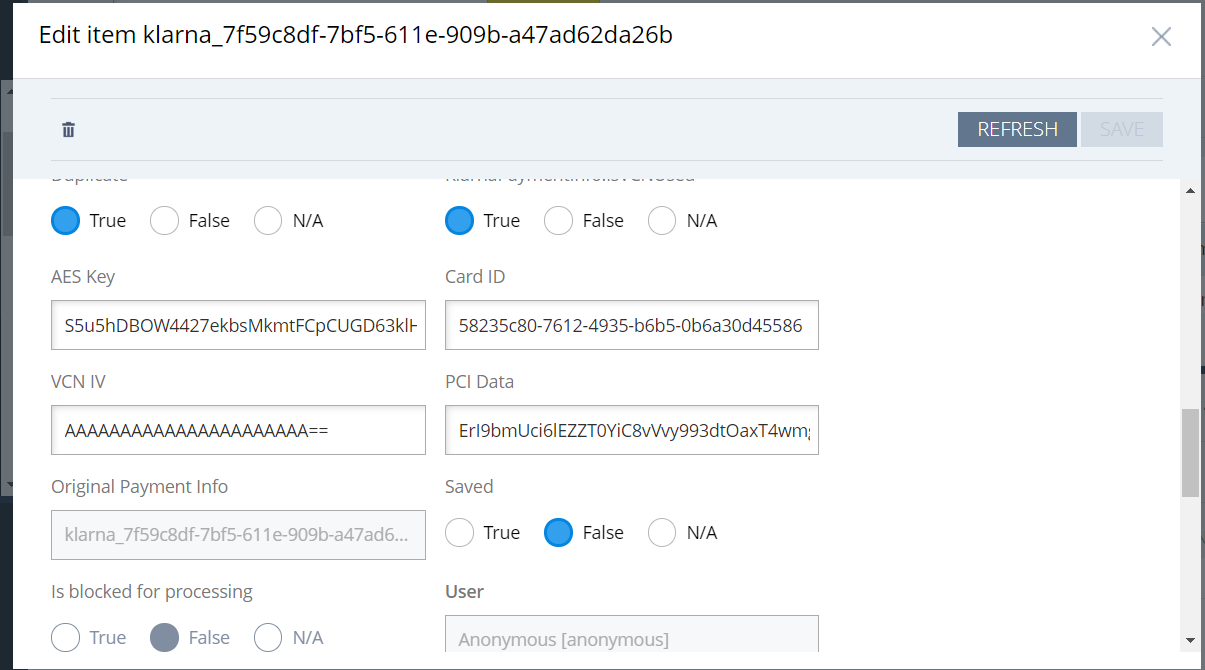
### **Verifying the VCN Data in Backoffice**

To verify the Klarna orders in Backoffice, follow these steps:

1. **Access the Order:**
   * Open **SAP Commerce Backoffice**.
   * Click on **Order** in the left-side menu.
   * Search for the specific order number.
2. **View Order Details:**
   * Click on the order number.
   * Navigate to the **Payment & Delivery** section.
3. **Select Payment Information:**
   * Select the payment information related to the order.



1. **Verify Virtual Card Details:**
   * Review the virtual card details displayed.

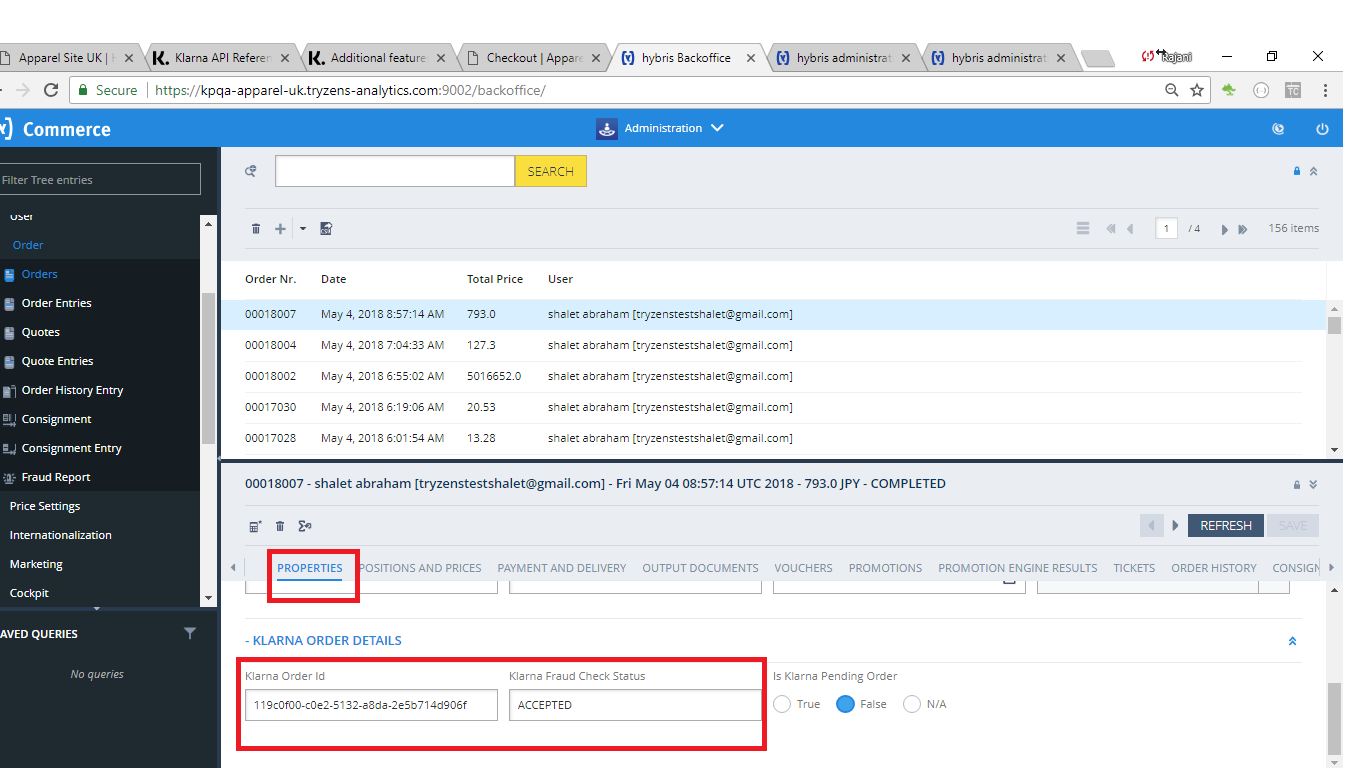


|  |  |
| --- | --- |
|  | **Important!**   * The cards are one-time use cards specific to the order. It is the merchant's responsibility to securely store them. * The above representation is for reference only. Do not store decrypted PCI data (including VCN/CVV) in the production environment. * Ensure all PCI-DSS compliance requirements are met when handling card data. |

### **Verifying Klarna Order ID in Backoffice**

To verify orders in the Backoffice, follow these steps:

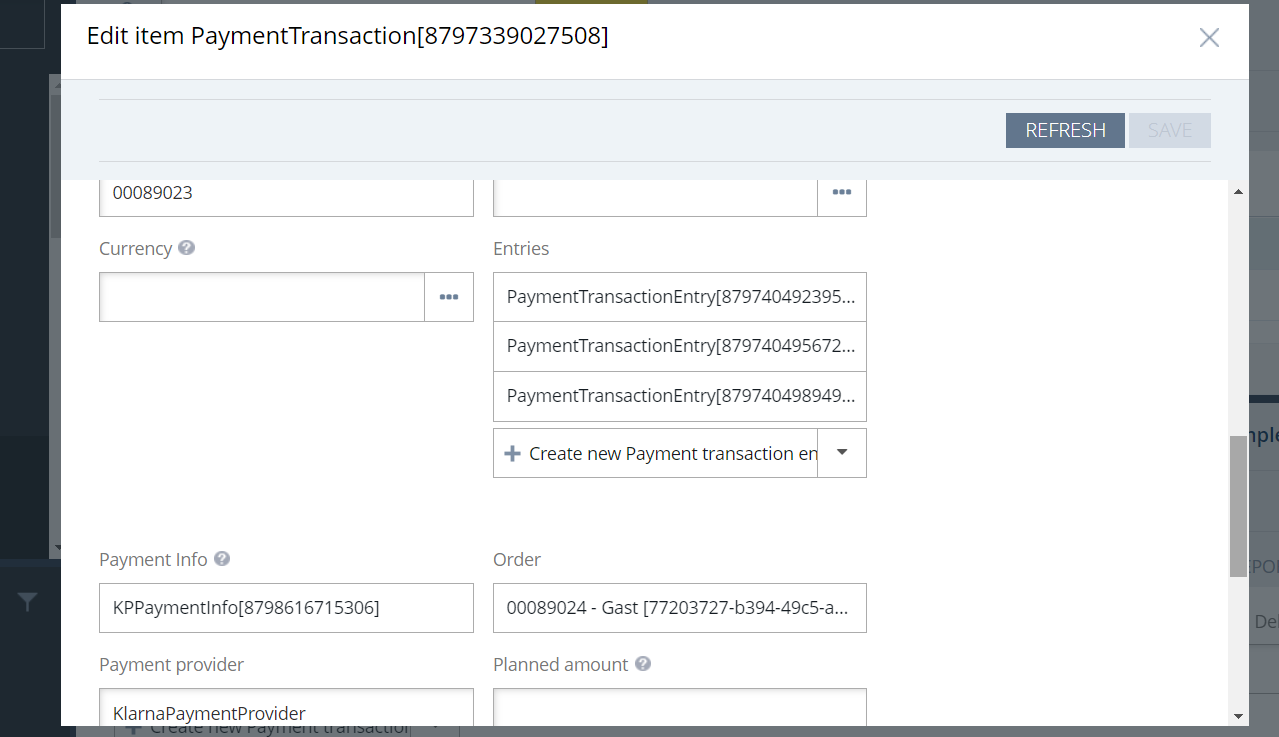
1. **Navigate to Orders**
   * Log in to the Backoffice.
   * Click on **Orders** in the left-side menu.
2. **Search for the Order Number**
   * Enter the specific order number in the search bar.
3. **Verify Klarna Order ID**
   * Go to the **Properties** tab to confirm the Klarna Order ID.



1. **Verify Payment Transactions**
   * Check the **Administration** tab.



* + Look for the transaction type **KLARNA\_ORDER\_PLACED**, which is custom for Klarna orders.



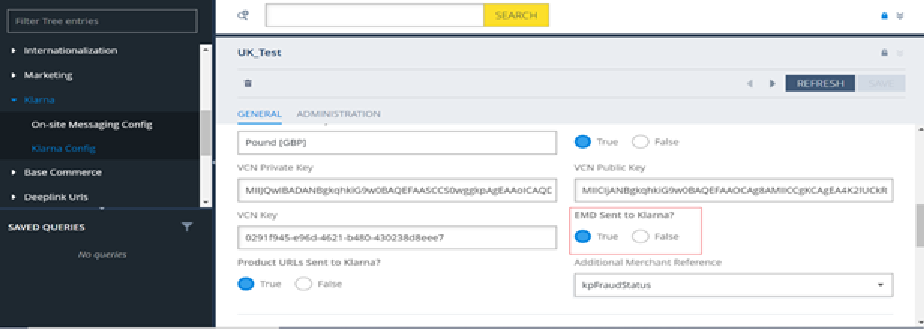
### **Extra Merchant Data (EMD) Setup**

The Extra Merchant Data (EMD) feature allows merchants to send additional information to Klarna. This can be configured through the back office by enabling the EMD option. The additional details that can be sent include customer information, alternative delivery addresses, and reservation details. Note that by default, only customer information for logged-in customers (OOB) can be sent.

To configure EMD in the code, use the addAttachment method in the KPCreditSessionPopulator.java file. This method is exposed specifically for setting EMD data.

#### Steps to Enable EMD:

1. **Enable EMD in Back Office**:
   * Navigate to the back office.
   * Locate the EMD option and enable it.



1. **Configure EMD in Code**:
   * Open the KPCreditSessionPopulator.java file.
   * Locate the addAttachment method.
   * Use this method to set the required 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

1. **Stop the SAP Commerce Server**: If the server is running, stop it using Ctrl+C or the following commands, as described in the Installation section:
   * **Windows**: hybrisserver.bat stop
   * **Unix**: ./hybrisserver.sh stop
2. **Set Up the Environment**: Go to <HYBRIS\_HOME>/bin/platform and run the environment setup script if it has not been applied in this terminal:
   * **Windows**: setantenv.bat
   * **Unix**: ./setantenv.sh
3. **Uninstall the Add-on**: Navigate to <HYBRIS\_HOME>/bin/platform and run the following command:

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

###### 

1. **Remove Extensions:** Open <HYBRIS\_HOME>/config/localextensions.xml and delete the following lines:

|  |
| --- |
| <extension name="klarnapaymentaddon" />  <extension name="klarnapayment" />  <extension name="klarnapaymentapi" />  <extension name="klarnapaymentbackoffice" /> |

###### 

1. **Delete Extension Folders:** Delete the corresponding folders from <HYBRIS\_HOME>/bin/custom:
   * klarnapaymentaddon
   * klarnapayment
   * klarnapaymentapi
   * klarnapaymentbackoffice
2. **Remove Payment Page References:** Open the payment page and remove the following lines.

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

1. **Rebuild and Restart the System**:
   * Rebuild the system using:

|  |
| --- |
| ant clean build |

###### 

* + Restart the SAP Commerce server.

1. **Update SAP Commerce**:
   * Open a web browser and go to hAC → Platform → Update.
   * Check the four checkboxes under General Settings:
     + Update running system
     + Clear the hMC configuration from the database
     + Create essential data
     + Localize types
   * Click the **Update** button to complete the process.

By following these steps, you will successfully uninstall the Klarna add-on from your SAP Commerce environment.

## Modifying the order confirmation email

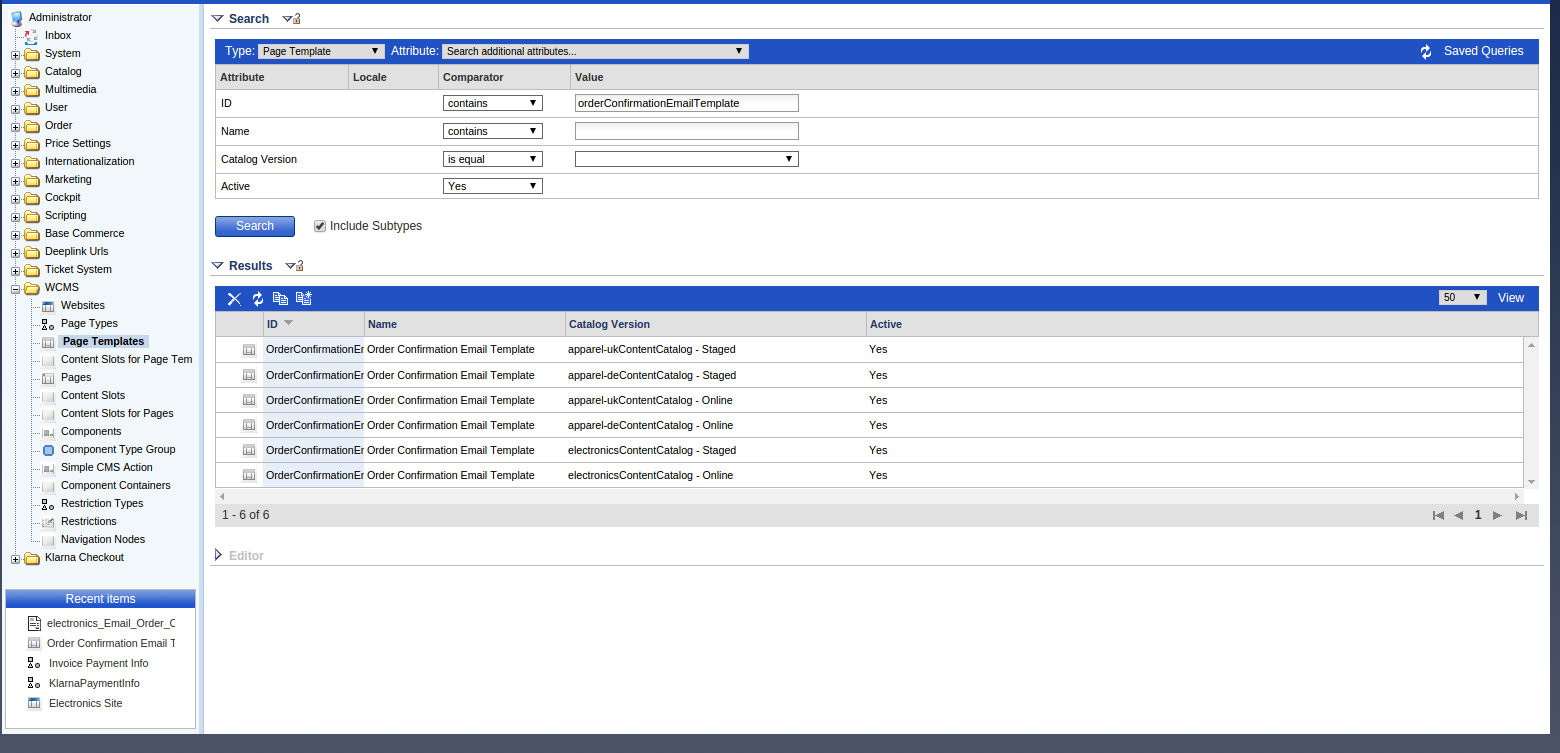
SAP Commerce will send the customer an order confirmation email once the order is placed. Payment instructions and notifications for customers will be sent directly from Klarna. It is recommended to modify the ‘Payment Details’ section of the confirmation email to inform the customer that payment has not been taken yet (except for Pay Now payment methods). Additionally, notify the customer that further updates will be received from Klarna. Customers can check their purchases and payments in the Klarna app (app.klarna.com).

To display the payment method as ‘Purchased with Klarna’ in the order confirmation email, you can add a message property. Sample message properties are provided in the folder /klarnapaymentaddon/resources/klarnapaymentaddon/messages. Merchants can modify these properties or add new property files. Ensure that the property file is included in the EmailTemplate.

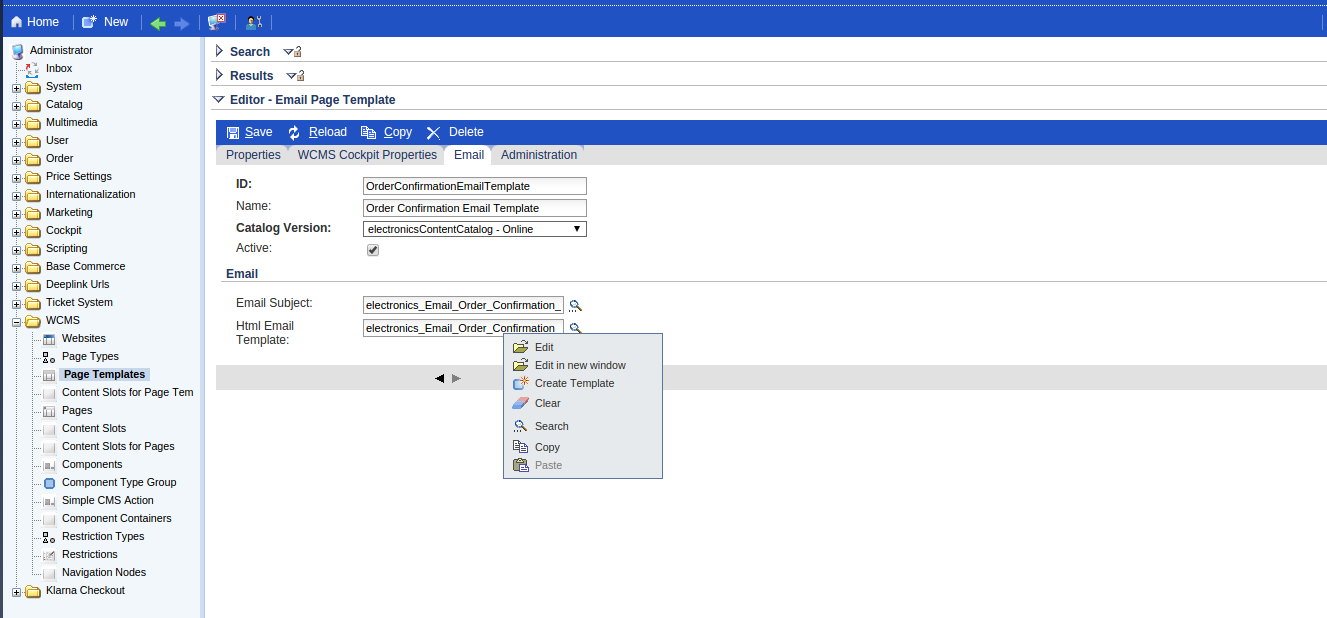
Follow the steps below to modify the email template:

#### Using HMC:

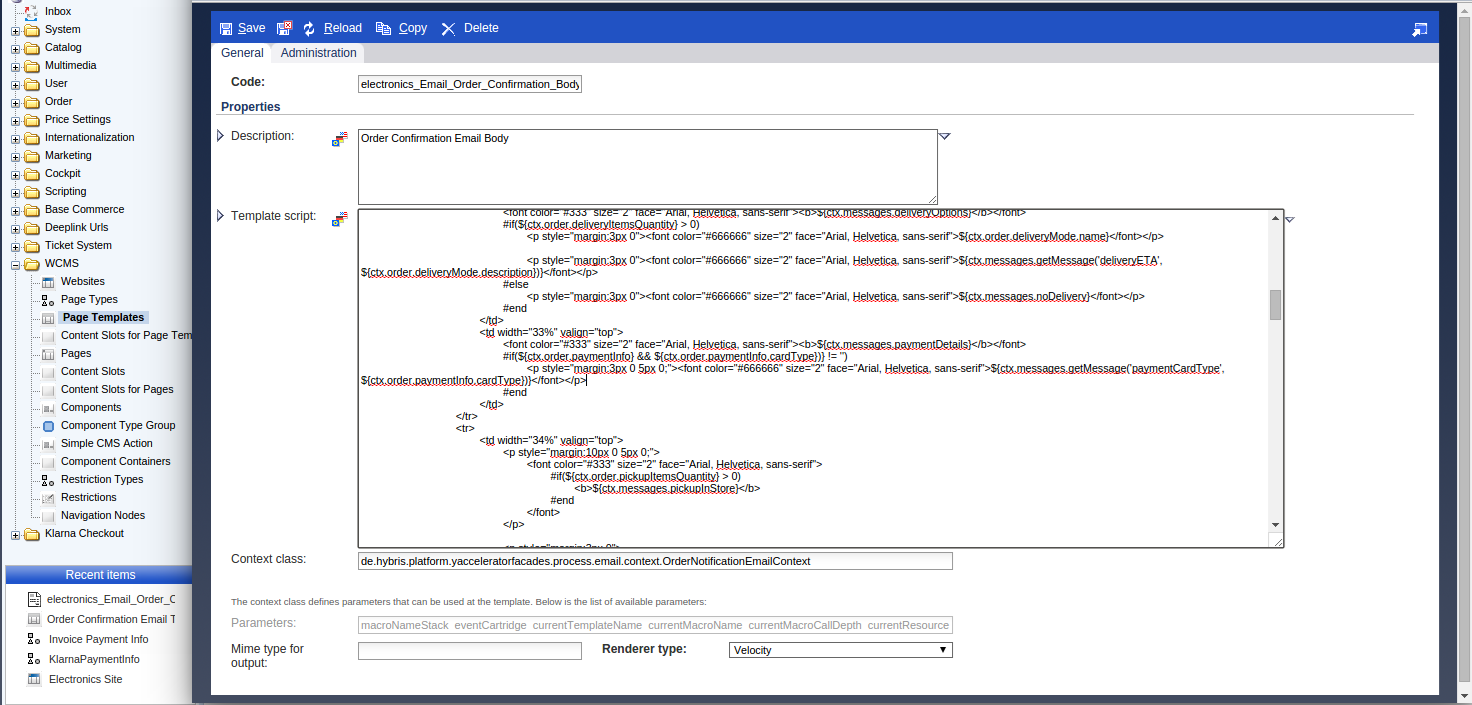
1. **Login to HMC**:
   * Use an administrator account to log in to HMC.
2. **Navigate to Email Template**:
   * Go to **WCMS** → **Page templates**.
   * Find and select the email template you wish to modify. You can also right-click and select ‘Edit in new window’. Note that it is recommended to edit the Staged version and then synchronize it to the Online version.



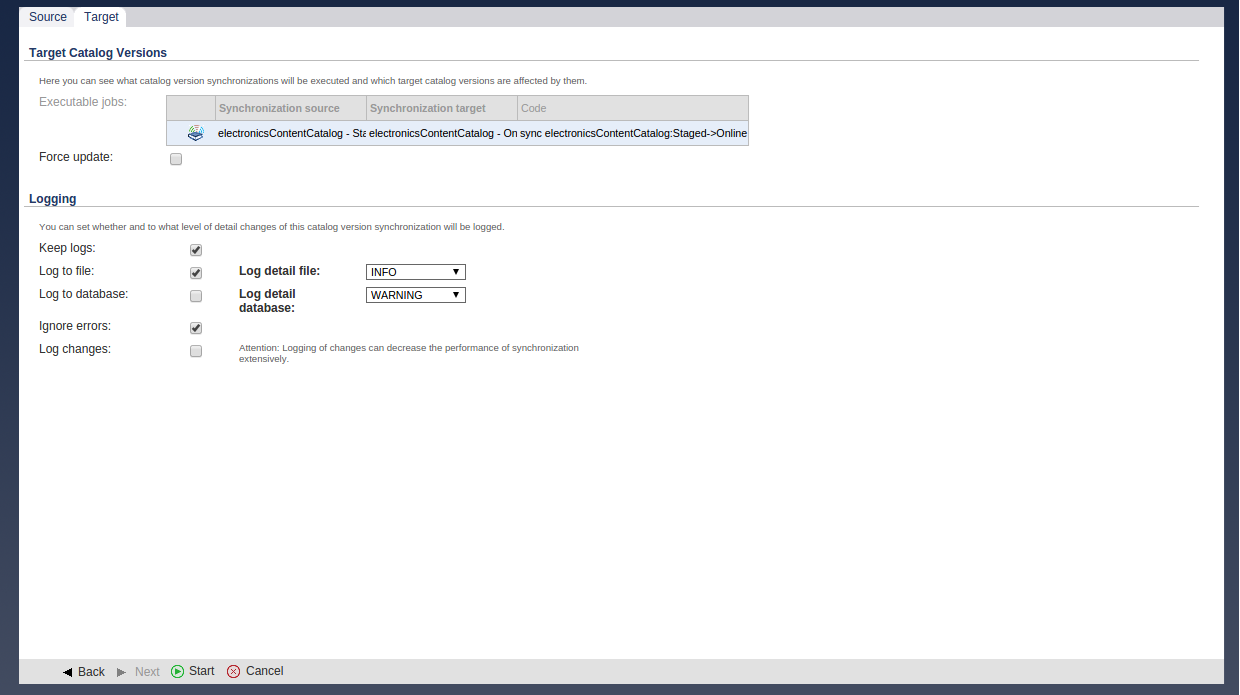
1. **Edit HTML Email Template**:
   * Select the ‘Email’ tab.
   * Right-click on ‘Html Email Template’ and select ‘Edit in new window’.

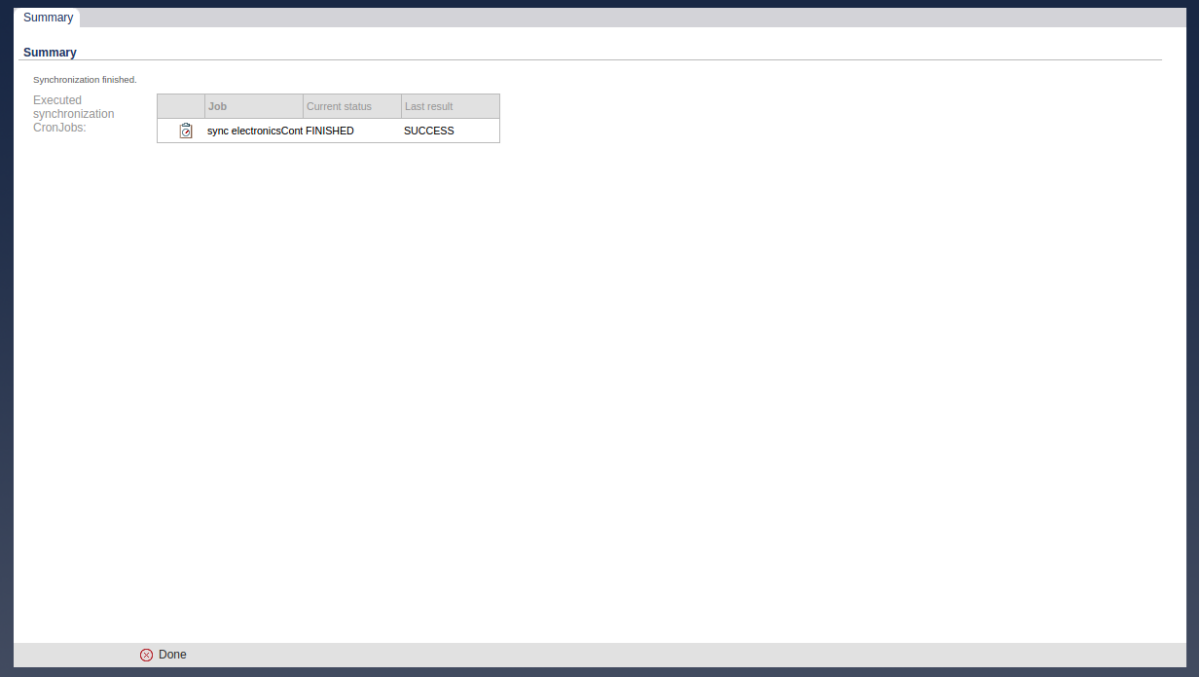


1. **Update HTML Content**:
   * Update the HTML content in the ‘Template script’ as required.



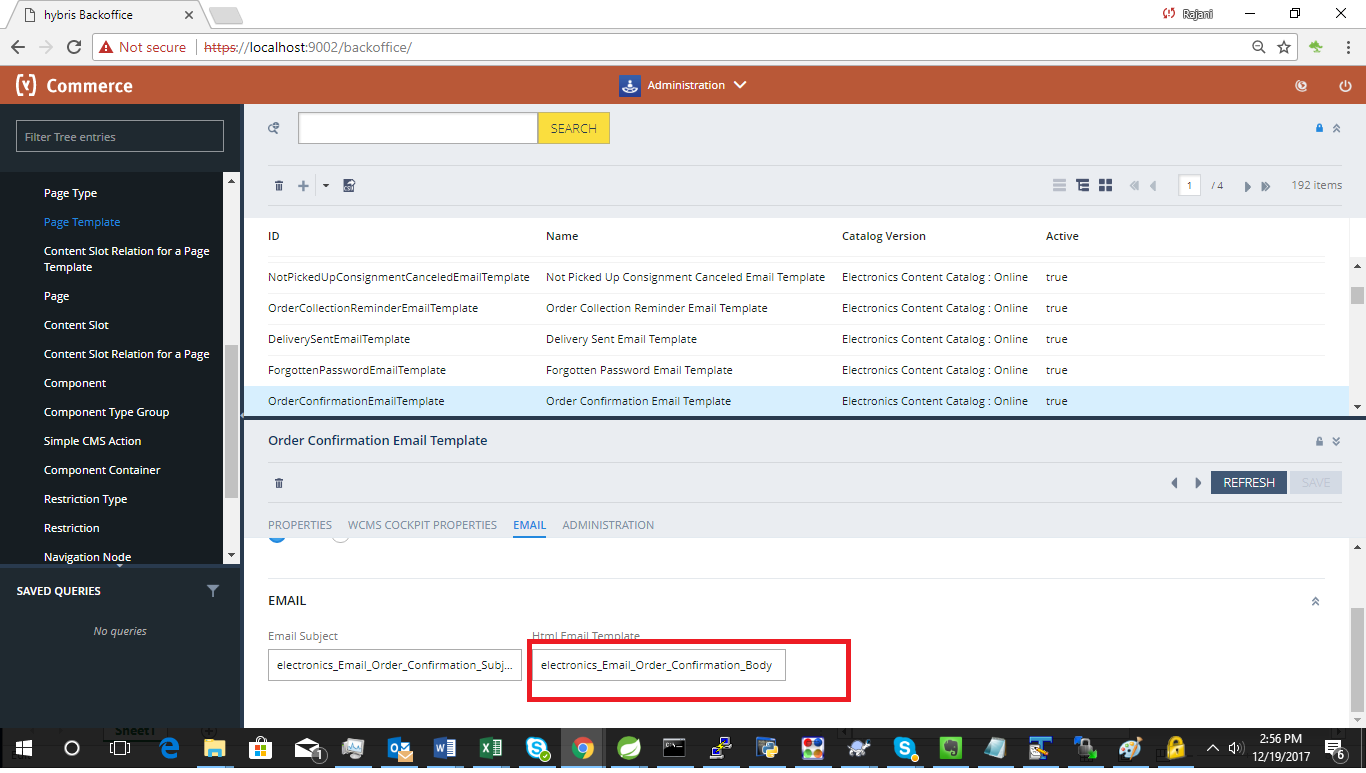
1. **Synchronize Changes**:
   * Navigate to **Catalog** → **Catalog Management Tools** → **Synchronization**.
   * Select the catalog version to synchronize, then click **Next** and **Start**.



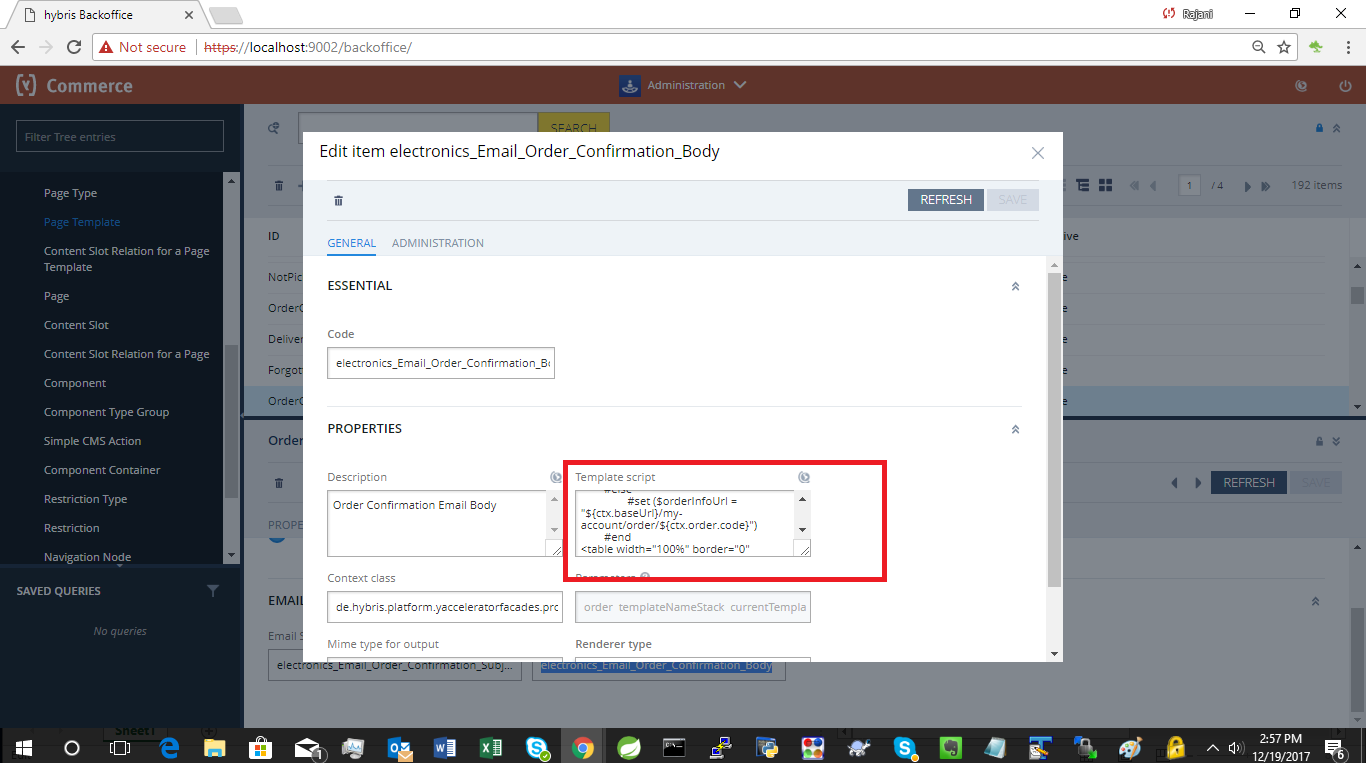


#### Using Backoffice:

1. **Login to Backoffice**:
   * Use an administrator account to log in to Backoffice.
2. **Navigate to Email Template**:
   * Go to **WCMS** → **Page templates**.
   * Find and select the email template you wish to modify. You can also right-click and select ‘Edit in new window’. Note that it is recommended to edit the Staged version and then synchronize it to the Online version.
3. **Edit HTML Email Template**:
   * Select the ‘Email’ tab and double-click on the ‘Html Email Template’.



1. **Update HTML Content**:
   * Update the HTML content in the ‘Template script’ as required.



## Troubleshooting

If Klarna Payment options are not available in the checkout flow, follow these troubleshooting steps:

1. **Verify Klarna Configuration**:
   * Ensure that the correct Klarna configuration is set up for the current base store.
   * Check that the configuration is marked as active.
   * Confirm that the purchase country, language, and currency are valid for the selected base store.
2. **Resolve ItemType Conflicts**:
   * ItemType typeCodes may have conflicts in the latest versions (2005 & 2011) of SAP Commerce based on the installation recipe.
   * It is advised to review and adjust the ItemType definitions to resolve any conflicts.

# Klarna On-site messaging

## Installation

### **Download and Unpack Add-on**

1. **Download Add-on**: Obtain the Klarna KOSM add-on archive.
2. **Unpack Archive**:
   * Unpack the archive.
   * Place the folders klarnaosmaddon and klarnaosmbackoffice into the custom folder of the SAP Commerce Suite (<HYBRIS\_HOME>/bin/custom).

### **Update localextensions.xml**

1. **Verify Add-on Support**: Ensure <extension name="addonsupport" /> is present in the localextensions.xml file.
2. **Add Klarna Extensions**: Add the following extensions to the localextensions.xml file:

|  |
| --- |
| <extension name="klarnaosmaddon" />  <extension name="klarnaosmbackoffice" /> |

1. **Check Target Storefront Extension**:
   * Ensure that the target storefront extension is correctly referenced.

### **Stop SAP Commerce Server**

1. **Stop Server**:
   * If the server is running, stop it using the appropriate method:
     + **Embedded Mode**: Press Ctrl+C.
     + **Service Mode**:
       - **Windows**: hybrisserver.bat stop
       - **Unix**: ./hybrisserver.sh stop

### **Set Up Environment**

Go to <HYBRIS\_HOME>/bin/platform and run the environment setup script if it has not been applied in this terminal:

* **Windows**: setantenv.bat
* **Unix**: ./setantenv.sh

### **Install Add-on**

**Run Add-on Installation Command**: Run the following command to install the add-on.

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

|  |  |
| --- | --- |
|  | **Note:** The storefront name may differ based on the project. |

### **Configure Klarna URLs and Tags**

Configure the script (library) URL, UCI, country, and placement tag IDs in the backoffice. This information is available in the Klarna Merchant Portal for both playground and production environments.

### **Modify Code for KOSM Integration**

1. **Update Master Tag**: Modify the master tag to include the following script as the first entry in the body tag:

|  |
| --- |
| <c:if test="${osmConfigData.active == true}">  <script async src="${scriptUrlKOSM}" data-client-id="${uci}"></script>  </c:if> |

1. **Modify PDP Page**: For the PDP page, update productPricePanel.tag as follows:
   * **Add Tag Library**:

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

###### 

* + **Add osmproduct Tag**:

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

* + **Include osm Tag at the End**:

|  |
| --- |
| <osm:osm/> |

1. **Modify Cart Page**: For the cart page, update cartTotals.tag as follows:
   * **Add Tag Library**:

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

###### 

* + **Set Final Price 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**:

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

### **Rebuild the System**

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

|  |
| --- |
| ant clean all |

### **Update the System**

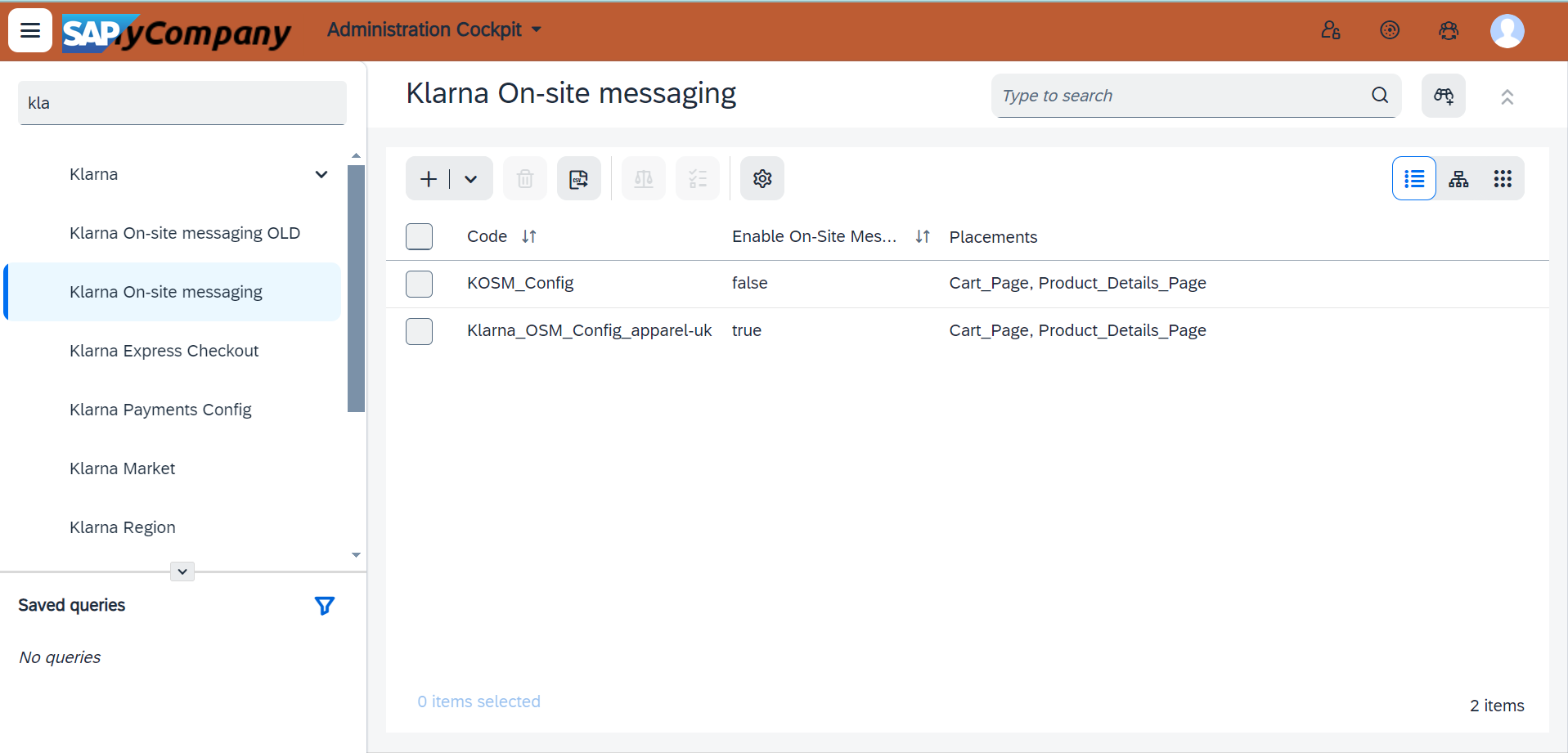
1. **Full Initialization (if first installation)**: If this is the first installation of SAP Commerce, perform a full initialization through the Hybris Administration Console (HAC).
2. **Update SAP Commerce System**:
   * Open a web browser and go to HAC → Platform → Update.
   * Check the following checkboxes under General Settings:
     + Update running system
     + Localize types
   * Check the klarnaosmaddon and klarnaosmbackoffice checkboxes under Project Data Settings.
   * Click the **Update** button to update the SAP Commerce system.

## Configuration

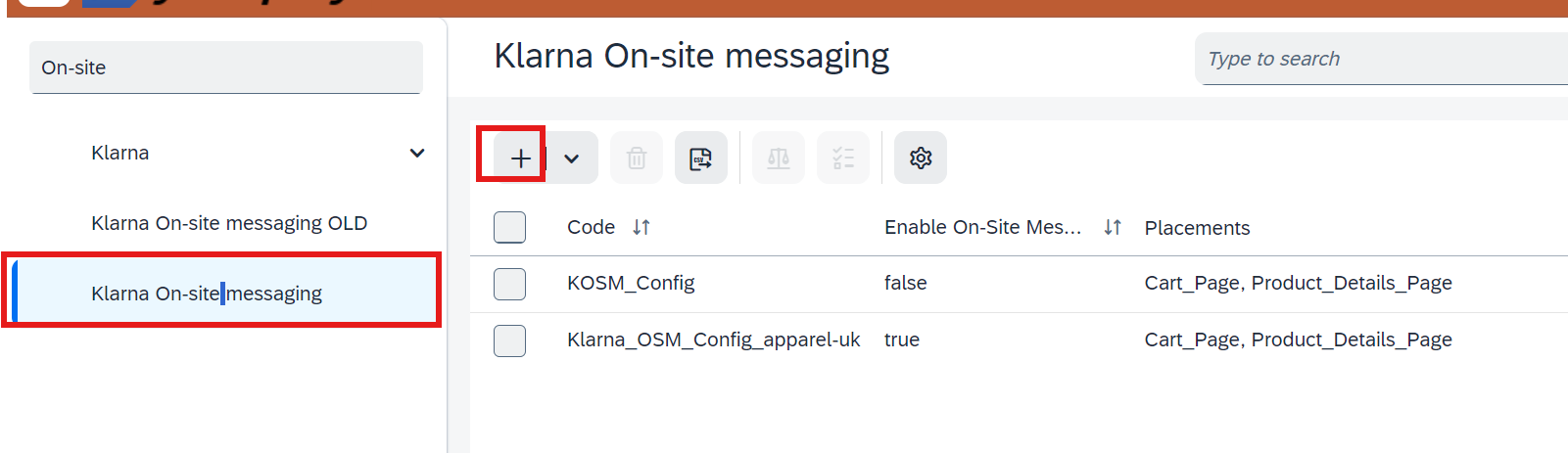
After successfully installing the klarnaosmaddon, you can log in to SAP Commerce Backoffice to set up a Klarna configuration and assign it to specific sites in SAP Commerce.

### **Create Klarna OSM Configuration**

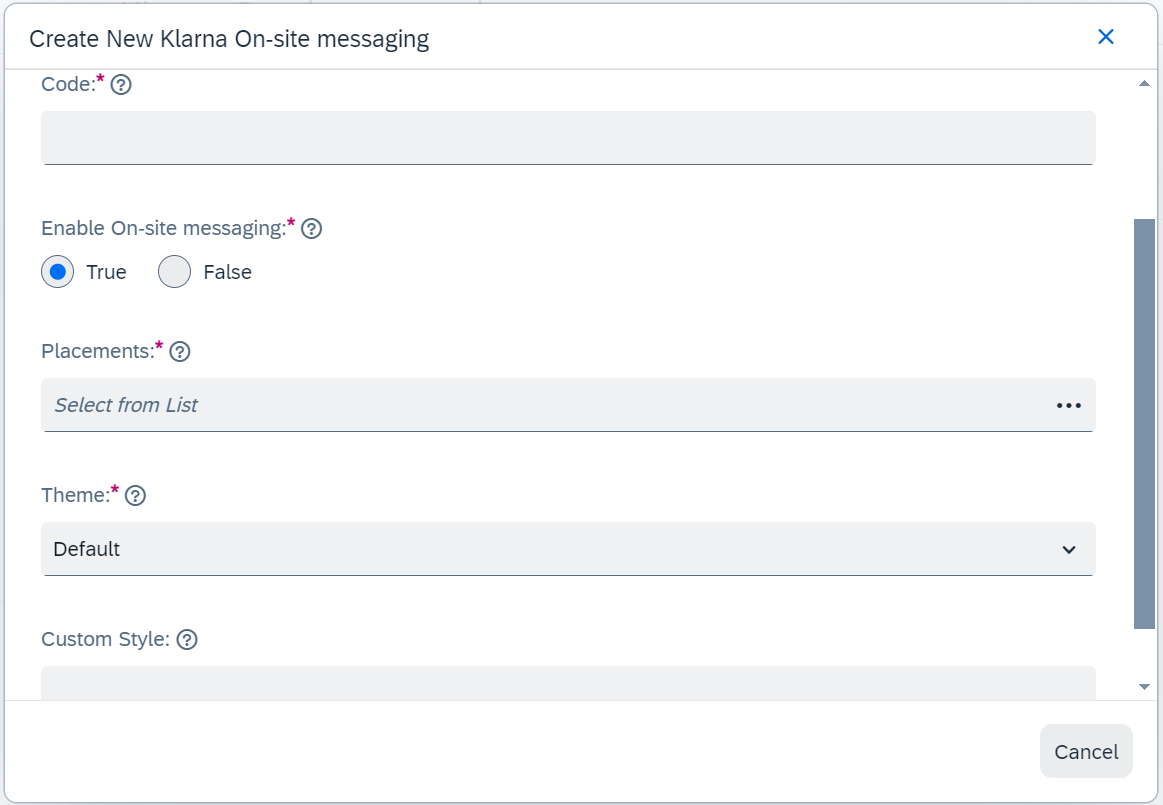
1. **Login to Backoffice**:
   * Open a web browser and navigate to the SAP Commerce Backoffice login page.
   * Log in using your administrator account credentials.
2. **Navigate to Klarna** **On-site Messaging**:
   * In the Backoffice, go to **Klarna** → **Klarna** **On-site Messaging**



1. **Create a New Configuration**:
   * Click on **Create** to add a new Klarna On-site Messaging configuration.



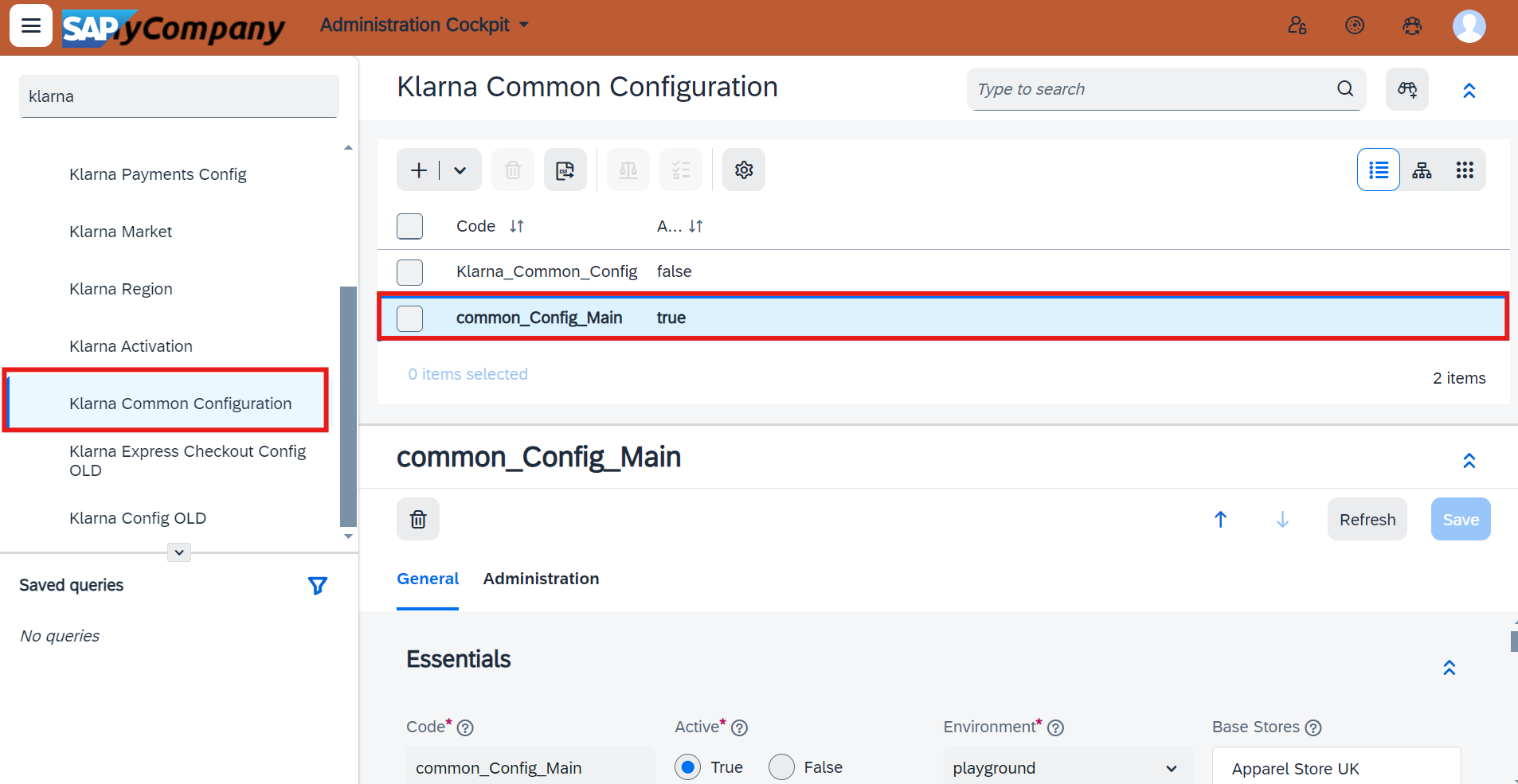
1. **Fill in Configuration Details**:
   * Enter the values for configuring the OSM.



1. **Save Configuration**:
   * After entering the required data, click on the **Done** button to save the configuration.

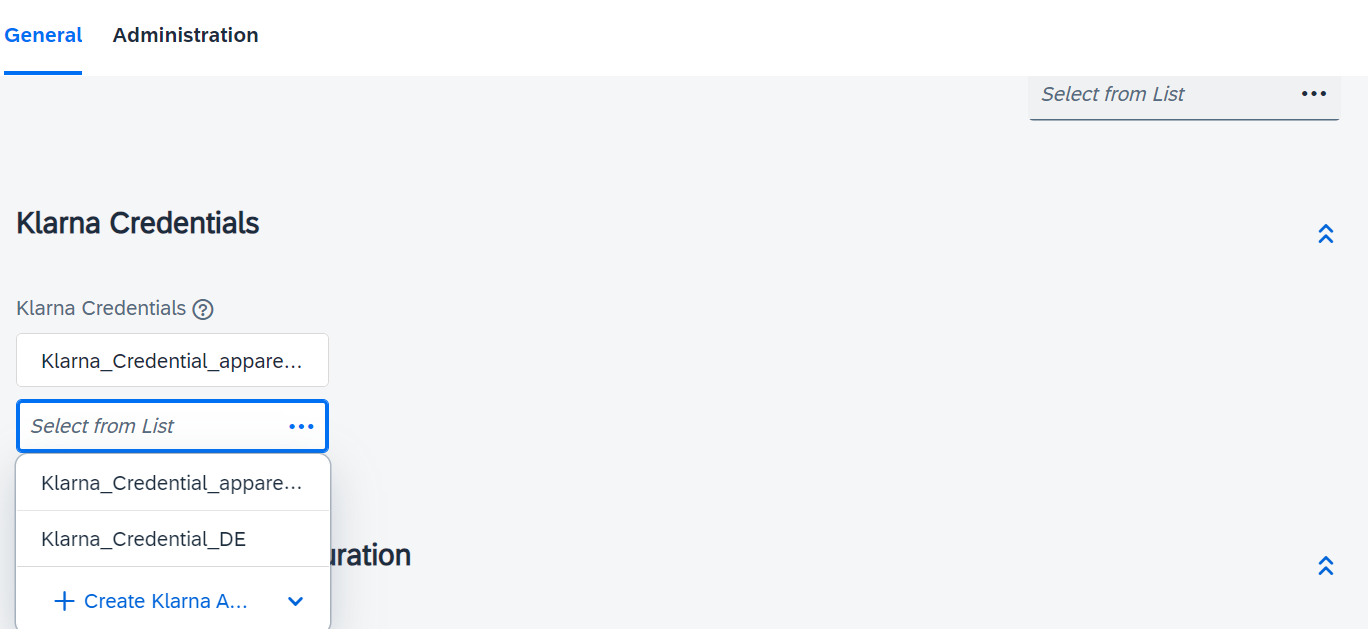
### **Mapping to KOSM to Common Config, Credentials and Base Store**

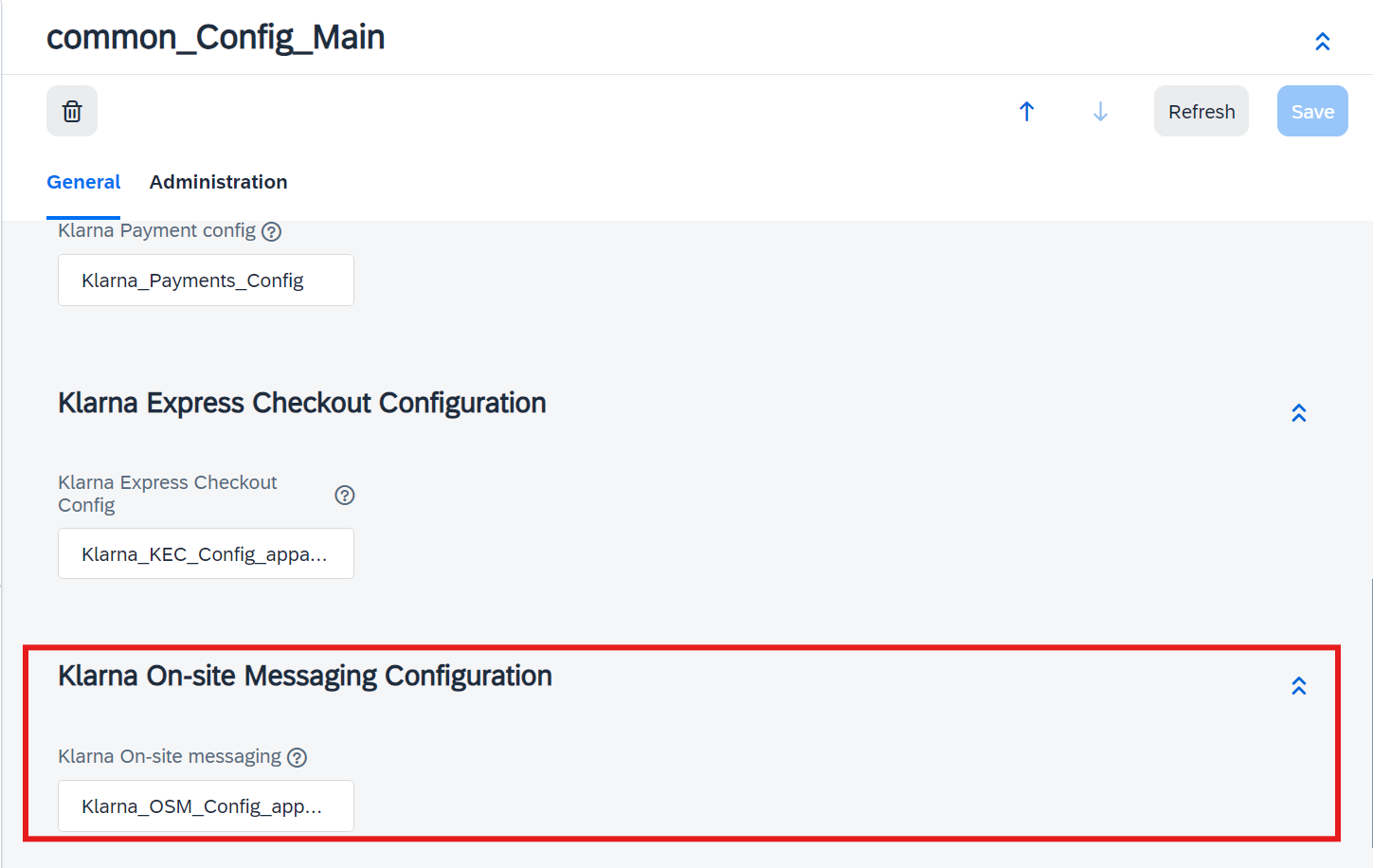
1. **Navigate to Klarna Common Configuration**:
   * In the Backoffice, go to **Klarna** → **Klarna Common Configuration**.
   * Double-click on the Klarna Common Configuration to edit it.



1. **Assign Base Store**:
   * Go to the **General** tab.
   * Under the **Essentials** section, select the appropriate Base Stores to show **KOSM**.



1. **Assign Credentials**:
   * Under the **Klarna Credentials** section, select the appropriate Credentials to show **KOSM**. (In case you wish to create new Credentials, click on the + Create Klarna Activation)  
     
2. **Assign Klarna On-site Messaging Configuration**:
   * Scroll down to the **Klarna On-site Messaging Configuration** section, select the newly created OSM configuration



1. **Save Changes**:
   * Click on **Save** to apply the changes to the Klarna Common Configuration.

### **Klarna OSM Configuration Fields**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Mandatory** | **Description** |
| Code | Yes | Input any name/code. This will be the internal name used to recognize and manage this particular Klarna OSM configuration within SAP Commerce. |
| Active | Yes | Check to activate this configuration; leave unchecked for it to remain inactive. |
| Client Id | Yes | UCI value is fetched from the Klarna Merchant Portal (data-client-id) to be provided under  Klarna > Klarna Activation |
| Placements | Yes | Multi Select Option to show in Cart or PDP |
| Theme | Yes | Select applicable theme from drop down for Cart Page and PDP. 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>* |

## Uninstalling the OSM Add-on

**Step 1: Stop SAP Commerce Server**

If the SAP Commerce server is running, stop it using Ctrl+C or the following commands:

* **Windows**: hybrisserver.bat stop
* **Unix**: ./hybrisserver.sh stop

**Step 2: Set Up Environment**

Go to <HYBRIS\_HOME>/bin/platform and run the environment setup script if it has not been applied in this terminal:

* **Windows**: setantenv.bat
* **Unix**: ./setantenv.sh

**Step 3: Uninstall Add-on**

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

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

**Step 4: Update localextensions.xml**

Open <HYBRIS\_HOME>/config/localextensions.xml and delete the following lines:

|  |
| --- |
| <extension name="klarnaosmaddon" />  <extension name="klarnaosmbackoffice" /> |

**Step 5: Delete Extension Folders**

Delete the following folders from <HYBRIS\_HOME>/bin/custom:

* klarnaosmaddon
* klarnaosmbackoffice

**Step 6: Remove Tag Libraries and Tags**

Update the PDP and cart pages to remove the following tag library and tags:

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

**Step 7: Rebuild the System**

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

|  |
| --- |
| ant clean all |

**Step 8: Update the System**

Open a web browser and go to HAC → Platform → Update. Check the following checkboxes under General Settings: Update running system, Create essential data, Localize types. Click on the **Update** button to update SAP Commerce.

# Klarna Express checkout

Klarna Express checkout (KEC) is a new feature that displays an Express Checkout Button on the Product Detail Page (PDP), Cart, and Mini Cart. When the KEC button is clicked, a Klarna popup is displayed where customers can view their account details and select shipping and payment options. Upon successful validation, the customer is redirected to the merchant’s website, where the cart is prepopulated with the shipping and payment data received from Klarna for a faster checkout experience.

For more information, refer to the [Klarna Documentation](https://docs.klarna.com/express-checkout/).

## Use Cases

### **Express Checkout from Cart / Minicart**

The KEC button is displayed on the Cart page and in the Minicart popup. When the button is clicked, an authorization request is sent to Klarna with a payload containing the cart details. If the request is valid, a popup appears where the customer can enter their phone number and verification code to validate the payment details. Upon successful validation, an authorization response with a client token is sent back to the merchant’s site.

### **Express Checkout from PDP**

The KEC button on the Product Detail Page (PDP) works similarly to the one on the Cart page and Minicart. However, clicking the button on the PDP creates a new session cart, regardless of whether the customer already has a session cart. The product is added to the new cart, and the details of this cart are sent as payload to Klarna.

### **Single Step vs Multistep Checkout**

#### Single Step Checkout

Klarna offers an option for single step checkout where the authorization response from Klarna contains the authorization token, allowing the customer to be taken directly from the PDP or Cart page to the Order Confirmation page.

#### Multistep Checkout

Klarna also supports multistep checkout, which requires an additional finalize call after the authorize call to obtain the authorization token. This is enabled by sending the parameter autoFinalize as false during the authorize call. This method is suitable for websites that display an order review/summary page before order placement. Customers can review and edit their checkout details if necessary. In this scenario, the authorization response from Klarna contains a client token used to make a finalize call from the order review/summary page. The response from the finalize call contains the authorization token, which is sent to Klarna as part of the create order request triggered by clicking the Place Order button.

For SAP Commerce, orders must include the delivery mode, which is not available in the Klarna user profile. Therefore, the delivery mode must be added to the cart from the merchant's checkout page. Once the delivery address is set, a default delivery mode can be programmatically selected in SAP Commerce, and customers should be able to view the selected delivery mode before order placement. The order total can change based on the shipping charges for the selected delivery mode. Thus, the customer should be redirected to the order review page before confirming the order. Consequently, the autoFinalize flag should be set to false, and KEC will always be Multistep in SAP Commerce.

### **Finalize and Create Order**

In multistep checkout, clicking the Place Order button initiates a finalize call to get the authorization token. The client token received in the initial authorize call response is required to finalize the payment.

Customers can edit the address and/or delivery mode before placing the order. The payment option will be preselected as Klarna Express Checkout.

The updated order details are sent to Klarna during the finalize call. If there is a significant difference in the order amount sent in the finalize call compared to the amount approved initially during the authorize call, a popup will prompt the customer to acknowledge and approve the changed order amount.

The authorization token received in the finalize call response is used during the create order call. KEC follows the same process as KP for finalizing and creating orders.

### **Collect Shipping Address**

During the authorize call, a parameter called collectShippingAddress indicates whether the customer's shipping address should be included in Klarna's response. This parameter is configurable from the Backoffice.

* **When collectShippingAddress is true**:
  + The shipping address received from Klarna is set as both the delivery and billing address in the cart.
  + If the user is not logged in, a guest checkout user is created using the email address from the collected shipping address.
* **When collectShippingAddress is false**:
  + No shipping address is received from Klarna.
  + Logged-in customers are redirected to the checkout delivery address step to add the address, instead of going to the checkout summary page. They must then select the preferred delivery mode, but will not need to submit payment details again, as Klarna Express Checkout will be preselected in the payment step.
  + If the user is not logged in and no address is received from Klarna, a guest user cannot be created automatically without the customer's email address. The customer is redirected to the checkout login page and must resubmit payment details by selecting Klarna Payment in the checkout payment step.

It is recommended to set the collectShippingAddress flag to true to maintain the express checkout experience. If the customer wants to use a different address, they can modify it from the order summary page before placing the order.

## Installation

### **Add-on Installation Steps**

No separate installation for Klarna Express checkout (KEC) is required. However, if KEC is being enabled on top of an existing Klarna add-on installation, a clean build followed by a system update of the Klarna extensions from HAC is required to add the KEC configuration table KlarnaExpCheckoutConfig to the database schema.

### **Mandatory Code Changes**

To enable KEC, the following code changes should be made in the storefront extension.

1. **Update master.tag File**

Add the following entry to load the Klarna script library and define the configurations required for initializing the KEC button.

|  |
| --- |
| <c:if test="${isKlarnaExpCheckoutEnabled}">  <input type="hidden" class="text" name="kecButtonTheme" id="kecButtonTheme" value="${kecButtonTheme}" >  <input type="hidden" class="text" name="kecButtonShape" id="kecButtonShape" value="${kecButtonShape}" >  <input type="hidden" class="text" name="klarnaLocale" id="klarnaLocale" value="${klarnaLocale}" >  <input type="hidden" id="klarnaClientId" name="klarnaClientId" value="${klarnaClientId}"/>  <script src="${fn:escapeXml(scriptUrlKEC)}" async></script>  <spring:url value="/klarna/express-checkout/create-authorize-payload" var="expCheckoutAuthorizePayloadUrl"/>  <input type="hidden" name="expCheckoutAuthorizePayloadUrl" id="expCheckoutAuthorizePayloadUrl" value="${expCheckoutAuthorizePayloadUrl}"/>  <spring:url value="/klarna/express-checkout/process-authorize-response" var="expCheckoutProcessAuthorizeResponseUrl"/>  <input type="hidden" name="expCheckoutProcessAuthorizeResponseUrl" id="expCheckoutProcessAuthorizeResponseUrl" value="${expCheckoutProcessAuthorizeResponseUrl}"/>  <spring:message code="klarna.expcheckout.error" var="klarnaExpCheckoutErrorMessage"/>  <input type="hidden" id="klarnaExpCheckoutErrorMessage" name="klarnaExpCheckoutErrorMessage" value="${klarnaExpCheckoutErrorMessage}"/>  </c:if> |

1. **Update productDetailsPanel.tag File**

To display the KEC button on the PDP, add the following code:

|  |
| --- |
| <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>  <c:if test="${isKlarnaExpCheckoutEnabled and (showKECInPDP == true) and (product.stock.stockLevelStatus.code ne 'outOfStock')}">  <div id="klarna\_exp\_checkout\_container\_default">  <input type="hidden" class="text" name="klarnaExpCheckoutProductCode"  id="klarnaExpCheckoutProductCode" value="${product.code}" >  <div id="klarnaErrorMessage" title="Klarna Error Message" >  </div>  </div>  </c:if> |

1. **Update cartDisplay.jsp File**

To display the KEC button at the top section of the Cart page, add the following code:

|  |
| --- |
| <c:if test="${isKlarnaExpCheckoutEnabled and (showKECInCartPage == true) }">  <div class="col-sm-4 col-md-3 pull-right">  <div id="klarna\_exp\_checkout\_container\_default">  <div id="klarnaErrorMessage" title="Klarna Error Message" >  </div>  </div>  </div>  </c:if> |

1. **Update checkoutDisplay.jsp File**

To display the KEC button at the bottom section of the Cart page, add the following code:

|  |
| --- |
| <c:if test="${isKlarnaExpCheckoutEnabled and (showKECInCartPage == true)}">  <div class="col-sm-4 col-md-3 pull-right">  <div id="klarna\_exp\_checkout\_container\_checkout\_display">  </div>  </div>  </c:if> |

1. **Update cartPopup.jsp File**

To display the KEC button on the Minicart popup, add the following code:

|  |
| --- |
| <br>  <c:if test="${isKlarnaExpCheckoutEnabled and (showKECInCartPopup == true) }">  <div id="klarna\_exp\_checkout\_container\_in\_cartpopup">  <script>  ACC.klarnaexpcheckout.klarnaButtonLoad("#klarna\_exp\_checkout\_container\_in\_cartpopup");  </script>  <div id="klarnaErrorMessage" title="Klarna Error Message" >  </div>  </div>  </c:if> |

###### 

1. **Update acc.checkout.js File**

To avoid duplicate form submission, remove or comment out the following line in the toggleActionButtons() function in the acc.checkout.js file:

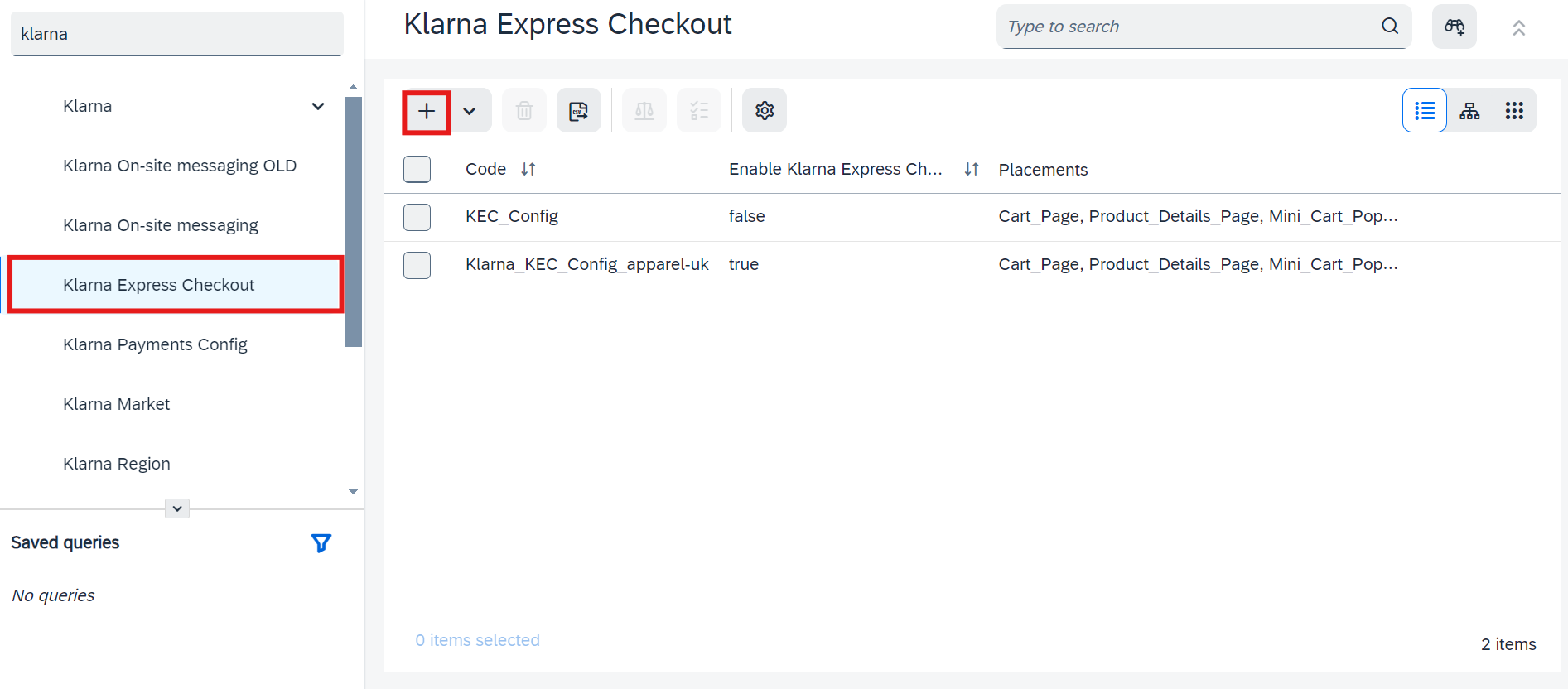
|  |
| --- |
| //$('#placeOrderForm1').submit(); |

## Activation and configuration

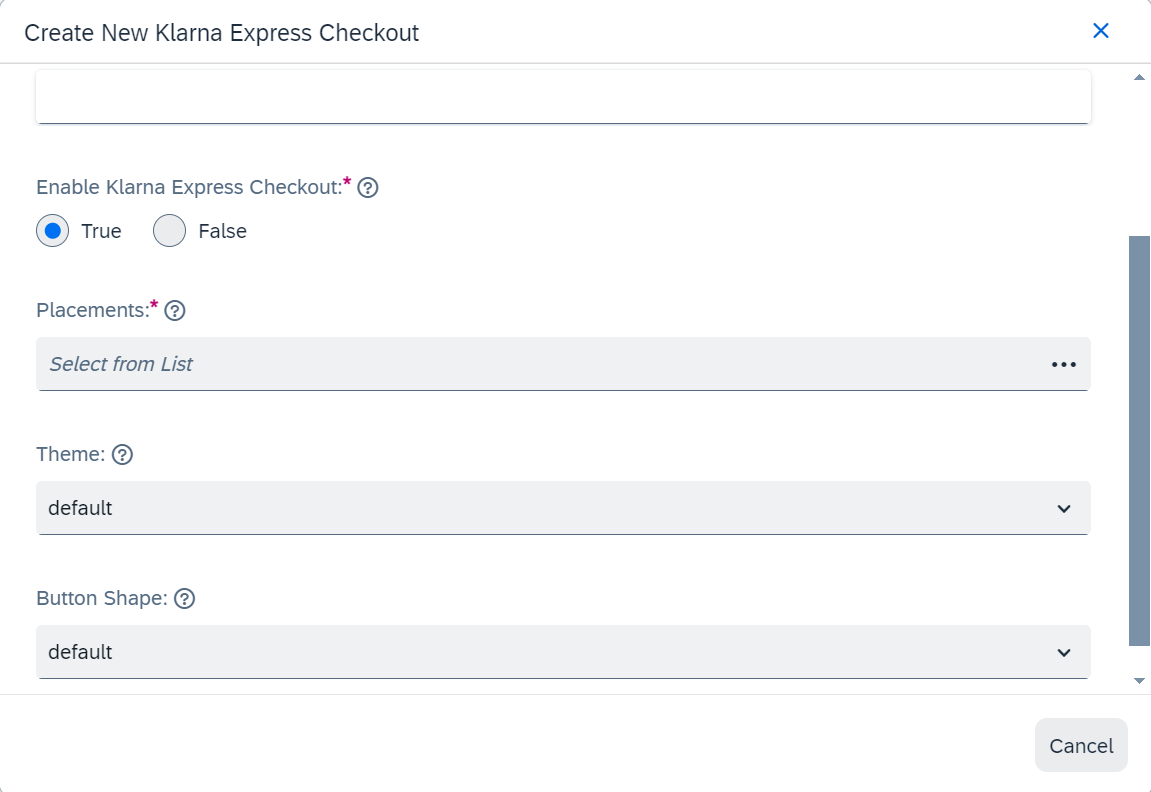
### **Configure Klarna Express Checkout in Backoffice**

To configure Klarna Express Checkout (KEC), follow these steps:

1. **Access KEC Configuration**:
   * Go to SAP Commerce Backoffice.
   * Navigate to **Klarna** → **Klarna Express Checkout**
2. **Create KEC Configuration**:
   * Create New KEC configuration by selecting the + icon.



* Provide the details for: Code, Enable Klarna Express Checkout, Placements, Theme and Button shape and click on ‘Done’ to save the changes.



1. **Mapping to KEC to Common Config, Credentials and Base Store**:
   * Mapping of credentials and base store is very similar to the steps mentioned for KOSM set up. Refer to the section of 3.2.2. Assign Klarna OSM to the common Configuration, map the Credentials and Base Store for further details.
   * In SAP Commerce Back office, navigate to **Klarna** → **Klarna Common Configuration.**
   * Under the **General** Tab, scroll down to reach the **“Klarna Express Checkout Configuration”**, now select the created Klarna Express Checkout and click on ‘Save’ 

|  |  |
| --- | --- |
|  | **Note:** Even if KEC is activated and linked to the Base Store, it will function only if there is an active KP configuration associated with the same Base Store. This is because KEC uses the KP configurations for creating orders in Klarna. |

**4.7.2 Klarna Express checkout configuration fields**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Mandatory** | **Description** |
| Code | Yes | Input any name/code. This will be the internal reference used to recognise and manage KEC configuration within SAP Commerce. |
| Active | Yes | Check to activate KEC configuration; leave unchecked for it to remain inactive. |
| Button Theme | No | Colour theme of the KEC button: default/light/dark |
| Button Shape | No | Shape of the KEC button: default/rect/pill |

## 

## 

# Sign in with Klarna

Sign in With Klarna is included along with the klarnapayment plugin so a separate plugin installation is not necessary. Although a clean build of the package and updating of database scheme is required to generate new Data Models and Data Transfer Objects.

# 5.1 Mandatory Code Changes:

* + - 1. **Making Changes to the master.tag file:**
* Open the master.tag file found in the path: <HYBRIS\_HOME>\bin\modules\baseaccelerator\deprecated\yacceleratorstorefront\web\webroot\WEB-INF\tags\responsive\template\master.tag
* Paste the below portion below <jsp:doBody/>

|  |
| --- |
| <c:if test="${isKlarnaSignInEnabled}">  <input type="hidden" id="currentLocale" name="currentLocale" value="${currentLocale}" >  <input type="hidden" id="clientId" name="clientId" value="${klarnaConfigData.credential.clientId}"/>  <input type="hidden" id="environment" name="environment" value="${klarnaConfigData.environment}"/>  <input type="hidden" id="country" name="country" value="${klarnaConfigData.credential.marketCountry}"/>  <input type="hidden" id="scopeData" name="scopeData" value="${klarnaConfigData.siwkConfig.scopeData}"/>  <input type="hidden" id="redirectUri" name="redirectUri" value="${klarnaConfigData.siwkConfig.redirectUri}"/>  <input type="hidden" id="buttonTheme" name="buttonTheme" value="${klarnaConfigData.siwkConfig.buttonTheme}"/>  <input type="hidden" id="buttonShape" name="buttonShape" value="${klarnaConfigData.siwkConfig.buttonShape}"/>  <input type="hidden" id="buttonLogoAlignment" name="buttonLogoAlignment" value="${klarnaConfigData.siwkConfig.buttonLogoAlignment}"/>  <input type="hidden" id="showInLoginPage" name="showInLoginPage" value="${klarnaConfigData.siwkConfig.showInLoginPage}" />  <input type="hidden" id="showInRegisterPage" name="showInRegisterPage" value="${klarnaConfigData.siwkConfig.showInRegisterPage}" />  <input type="hidden" id="showInCheckoutLoginPage" name="showInCheckoutLoginPage" value="${klarnaConfigData.siwkConfig.showInCheckoutLoginPage}" />  <script src="${fn:escapeXml(scriptUrlSIWK)}" defer></script>  </c:if> |

* + - 1. **Making Changes to the login.tag file:**
* Open the login.tag file located in the path: <HYBRIS\_HOME>\bin\modules\base-accelerator\deprecated\yacceleratorstorefront\web\webroot\WEB-INF\tags\responsive\user\login.tag
* Paste the below portion inside the <form:form action="${action}" method="post" modelAttribute="loginForm"> segment and below the <ycommerce:testId code="loginAndCheckoutButton"> … </ycommerce:testId> segment.

|  |
| --- |
| <c:if test="${isKlarnaSignInEnabled}">  <br>  <div id="klarna-signin-container" style="overflow:auto"><sec:csrfInput/></div>  <c:set var="signinErr"><spring:message code="klarna.signin.error"/></c:set>  <input id="signinErrHidden" type="hidden" value="${signinErr}"/>  <div id="klarna-signin-err" title="Klarna Error Message"><span></span></div>  <br>  </c:if> |

## **5.2** Building and Updating the Database:

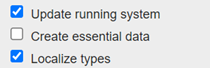
* Once the latest KP\_8.0 package has been updated to the *HYBRIS\_HOME*/bin/custom and *HYBRIS\_HOME*/config folders.
* Navigate to *HYBRIS\_HOME*/bin/platform and do a clean build

|  |
| --- |
| ant clean all |

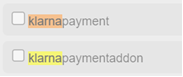
* Once the build is successful, start the server.

|  |
| --- |
| hybrisserver.bat |

* Once the server is up and running. Update the system by selecting the below. (hac > platform > update)



Select the below extensions





### **5.3 Backoffice Reset:**

* Once the update system is successful. Login to backoffice and Press F4. On the top right corner select the drop down and select “Reset All”.
* Once the Reset is completed switch back to the normal backoffice mode by clicking F4 again.

### **5.4 Activation and Configuration**

### **5.4.1 Setting up Sign-in configuration in Back office:**

**1.Creating Sign in with Klarna Configuration**

* Now search for ‘Sign in with Klarna’ and click on ‘Sign in with Klarna’ config.
* Click on ‘Create New Item’ Icon (+ sign). Provide the information for: Unique Identifier / Code, Scope: *openid , offline\_access, profile\_name, profile\_email, profile\_phone, profile\_billing\_address*, Redirect URL, Placements, Theme, Button Shape, Button Alignment.

A screenshot of a computer

Description automatically generated

|  |  |
| --- | --- |
|  | **Note:** While selecting the Scope for the SIWK configuration ***openid*** and ***offline\_access*** should be always selected as these are mandatory. |

**2. Mapping to SIWK to Common Config, Credentials and Base Store:**

* Mapping of credentials and base store is very similar to the steps mentioned for KOSM set up. Refer to the section of 3.2.2. Assign Klarna OSM to the common Configuration, map the Credentials and Base Store for further details.
* In SAP Commerce Back office, navigate to Klarna → Klarna Common Configuration.
* Under the General Tab, scroll down to reach the “Sign in With Klarna Configuration”, now select the created “Sign in With Klarna” and click on ‘Save’.

### **5.4.2 Sign in With Klarna configuration fields**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Mandatory** | **Description** |
| Code | Yes | Input any name/code. This will be the internal reference used to recognize and manage SIWK configuration within SAP Commerce. |
| Scope | openid,  offline\_access, profile\_name, profile\_email, profile\_phone, profile\_billing\_address,  profile\_country,  payments\_create\_session,  profile\_national\_id | Select Scope to get the information of the user. Mandatory to select (openid ,  offline\_access, profile\_name, profile\_email, profile\_phone) |
| Active | Yes | Check to activate SIWK configuration; leave unchecked for it to remain inactive. |
| Redirect URL | Yes | This is for redirection in case the storefront is accessed from mobile devices. |
| Placements | Yes | Multi select option to show the button in Login Page and Checkout Login Page. |
| Theme | No | Color theme of the SIWK button: default/light/dark |
| Button Shape | No | Shape of the SIWK button: Default, Rect, pill |
| Button Alignment | No | Button Logo Alignment: Left, Right, Center |

### **5.5 How to Sign in with Klarna:**

* Open Incognito window and close all other windows of that browser for incognito to work.
* Navigate to the login page of any storefront

*DE:* [*https://klarnawin-apparel-de.tryzens-analytics.com/de/login*](https://klarnawin-apparel-de.tryzens-analytics.com/de/login)

*UK:* [*https://klarnawin-apparel-uk.tryzens-analytics.com/en/login*](https://klarnawin-apparel-uk.tryzens-analytics.com/en/login)

* Click on continue with Klarna Button



* Use the email Id and phone number mentioned in the below sample Data

*UK:* [*https://docs.klarna.com/resources/test-environment/sample-customer-data/#united-kingdom*](https://docs.klarna.com/resources/test-environment/sample-customer-data/#united-kingdom)

*DE:* [*https://docs.klarna.com/resources/test-environment/sample-customer-data/#europe-germany*](https://docs.klarna.com/resources/test-environment/sample-customer-data/#europe-germany)

* If the account is already present in Klarna and present in SAP storefront, the user will be logged in successfully and the accounts will be merged automatically.
* If the account is already present in Klarna but not present in SAP storefront, then a consent page will be shown and upon consent the account will be created in the storefront.
* If the account is not present in Klarna, the login will fail, and the user shall go ahead with the normal login flow.

# Support

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

**Klarna App**

Klarna provides all customers with access to the Klarna App via:

* Website: [Klarna App Login](https://app.klarna.com/login)
* Mobile App: Download the free Klarna App on mobile devices to view payment instructions, pay for purchases, track delivery updates (when shipping references are updated), and extend due dates if they have chosen to pay after delivery.

**Klarna Status Page**

Klarna offers an external status page where merchants can view the health of Klarna solutions and products:

* [Klarna Status Page](https://status.klarna.com/)
* The page also allows merchants to subscribe to specific ongoing incidents via SMS or email by clicking the issue subscribe button.

## Reporting an Incident

Merchants should contact their Key Account Manager (KAM) to report any incidents or suspicions of degraded performance or issues with Klarna's services. The KAM will then report this internally to the incident management team, who will handle and resolve reported incidents. The KAM may request additional information to help identify and address the issue. Merchants may also be advised to follow updates on the status page if the incident is ongoing.

### Pre-requisite Information for Reporting an Incident:

To aid in the speedy investigation and resolution of issues, merchants should provide the following information when reporting an incident:

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

## Klarna Customer Service

The Klarna Customer Service department addresses payment-specific questions from customers who have made purchases at merchant sites. Customers can contact Klarna Customer Service through the Klarna App, available on both desktop and mobile platforms.

#### Accessing Klarna Customer Service:

1. **Via Desktop**:
   * Customers can log into the Klarna App via the website: [Klarna App Login](https://app.klarna.com/login).
2. **Via Mobile**:
   * Customers can download the free Klarna App on their mobile devices. Through the app, they can view payment instructions, pay for purchases, track delivery updates, and extend due dates if they choose to pay after delivery.

By using the Klarna App, customers have easy access to support for any payment-related queries they may have.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 merchants.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |
| 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 |