



Klarna Payments for SFRA

Version 25.4.0



Table of Contents

1.	Summary	6
2.	Component Overview	7
2.1.	Functional Overview	7
2.2.	Locales	9
2.3.	Use Cases	10
2.3.1.	Enable Klarna Payments Across International Sites (NA, EU, OC).....	10
2.3.2.	Multitude of Payment Options for Customers	10
2.3.3.	Klarna Display Conditions and Authorization Handling	11
2.3.4.	Authorize and Place Klarna Order in Checkout	11
2.3.5.	Refusal of Klarna Payments on Payment Method – Authorization	13
2.3.6.	Klarna Payment Option Not Available for Current Purchase - Billing Page	15
2.3.7.	Klarna Payments Not Available - Checkout	15
2.3.8.	Handling Notifications.....	15
2.3.9.	Virtual Cards Settlements	17
2.3.10.	Auto-Capture	21
2.3.11.	Widget Customizations	21
2.3.12.	Customizing Payment Method Name	22
2.3.13.	Klarna On-Site Messaging	23
2.3.14.	Klarna Express Button	29
2.3.15.	Klarna Payment Method Based Promotions	32
2.3.16.	Price Adjustment Taxation Handling	33

2.3.17.	Buy Online, Pickup in Store (BOPIS).....	34
2.3.18.	Configuration Support for Service Rate Limits	35
2.3.19.	Klarna Subscriptions	36
2.3.20.	Klarna Express Checkout	41
2.3.21	Klarna SIWK (Sign in with Klarna).....	42
2.4.	Compatibility.....	45
2.5.	Privacy, Payment.....	48
2.5.1.	GDPR Compliance.....	48
2.5.2.	EMD (Extra Merchant Data)	48
2.5.3.	PCI-DSS Compliance	53
3.	Implementation Guide	55
3.1.	Setup of Business Manager.....	55
3.1.1.	Cartridge Upload & Assignment.....	55
3.1.2.	Metadata Import.....	56
3.1.3.	Build Klarna JS & CSS.....	56
3.2.	Configuration	57
3.3.	Extended Controllers.....	60
3.4.	Template Updates	61
3.5.	Jobs.....	61
3.5.1.	Job “OrderCleanUp” (Optional)	61
3.5.2.	Job “RecurringOrders”.....	65
3.6.	Custom Code.....	66
3.6.1.	<code>app_storefront_base\cartridge\client\default\js\checkout\checkout.js</code> (optional) ...	66
3.6.2.	<code>app_storefront_base\cartridge\client\default\js\cart\cart.js</code> (required)	67
3.7.	Integration with Other Payment Cartridges	68
3.8.	External Interfaces.....	69
4.	Testing.....	69
5.	Operations, Maintenance	70
5.1.	Data Storage	70

5.1.1.	System Object Extensions.....	70
5.1.2.	Custom Objects.....	81
5.1.3.	Session Attributes & Cookies	88
5.1.4.	Library.....	89
5.1.5.	Services	90
5.2.	Logs.....	90
5.3.	Availability	90
5.4.	Failover/Recovery Process.....	90
5.5.	Support	90
5.5.1.	Merchant Support.....	91
6.	User Guide	92
6.1.	Cartridge upgrade.....	92
6.2.	Upgrade Process.....	92
6.2.1.	Review Release Notes.....	92
6.2.2.	Assess Custom Code Changes	92
6.2.3.	Backup Custom Files	92
6.2.4.	Compare Code Differences	92
6.2.5.	Update Integration Code	92
6.2.6.	Test and Validate	93
6.2.7.	Address Compatibility Issues	93
6.2.8.	Deploy Changes	93
6.3.	Roles, Responsibilities	93
6.4.	Storefront Functionality	93
7.	Known Issues	99
8.	Release History	100
9.	Additional Information	103
9.1.	Klarna API Information.....	103
9.1.1.	Live Environment	103
9.1.2.	Testing Environment.....	104

9.2.	Generate Key Pair and Key Id for Virtual Card Settlements (VCN)	104
9.3.	Decrypt VCN Card Details	107
9.4.	Update KlarnaCountries Definition	108

1. Summary

The **Klarna Payments SFRA cartridge** enables integration of Klarna Payment solution on Commerce Cloud Storefront. The integration provides merchants the flexibility to offer choice of multiple Klarna Payment products on the Commerce Cloud checkout.

This document contains the instructions for a developer to install the cartridge and integrate it on the Salesforce Commerce Cloud site. The cartridge is fully compatible with the Storefront Reference Architecture (SFRA).

Merchant teams are required also to configure the cartridge with the valid merchant credentials and site configurations in Commerce Cloud Business Manager to enable Klarna payments methods in the checkout.

The integration consists of an archive, which contains the following contents:

- Cartridge called “int_klarna_payments” and “int_klarna_payments_sfra” to be imported.
- A site-template archive containing new attributes and settings.
- This document for SFRA (Klarna Payments SFRA Integration Guide).
- The integration is based on the SFRA demo store provided by Commerce Cloud.

It is a requirement that Merchant sign a contract for integration support and production go-live with Klarna.

Klarna offers a playground (test) environment, so the integration can be tested before switching to the Klarna production environment. Based on the contract, Klarna shall provide assistance with integration and testing prior to sign-off for go-live.

2. Component Overview

2.1. Functional Overview

Key Features:

- Integrate Klarna Payments using best practices on international sites based (markets in North America, Europe, Oceania)
- Enable multiple payment products for customer in Pay Now, Pay Later and Pay Over Time categories
- Fast integration/go-live with virtual card-based integration approach for settlement
- Handle Notification: pending status updates (reject/accept) for suspected orders post review
- Site managers can customize the Klarna Payments widget styling displayed in checkout, to match the style guide of merchant website(s)
- GDPR (EU) compliant checkout flow
- Multi Shipping Address support
- Supports Klarna authorize with finalize for Bank Transfer methods (Pay Now)
- Enable Onsite Messaging placements on PDP, Cart, Header, Footer, and dedicated Info page
- Enable Klarna's Express Button on Cart page (*US) for faster checkout
- BOPIS (Buy Online, Pickup in Store) support including extra merchant data
- Support for Klarna Payment Method based promotions
- Support for adjusted price promotions with Gross Tax Policy
- Support disabling Payment method for authorization rejection
- Support Auto-Capture
- Support for custom service rate limits configuration

Klarna Payment cartridge makes use of the Klarna Payments JSON REST API and a JavaScript SDK to integrate on the storefront. Klarna Payment enables consumers to choose from the different payment method products offered by Klarna. Multiple Klarna products are available within the categories “Pay Now”, “Pay Later” and “Pay Over Time”. The cartridge integration displays payment options via a widget (iframe) added inline on the billing page, referred to as

Klarna widget or just “the widget”. The widget with information about the payment method is displayed to the customer when the individual clicks on the Klarna payment method.

Customers can authorize the payment after reviewing the payment method terms and clicking Place Order button. The Klarna order is created, and customer is re-directed to the confirmation page.

Orders successfully placed with Klarna return a Fraud Status: ACCEPTED and displayed in Business manager (BM).

With ACCEPTED status, order creation in SCC proceeds as usual. Klarna payment status is saved in a custom attribute with id kpFraudStatus in the PaymentTransaction system object, and can be seen in BM on the order details Payment tab as below:

The screenshot shows the Business Manager interface for an order. The URL in the address bar is Merchant Tools > Ordering > Orders > Order: 00005307(RefArchGlobal). Below the address bar, there are tabs: General, Attributes, Payment (which is selected), Notes, and History. The main content area is titled "Payment Information for Order '00005307'". The payment details are listed in a table:

Order Total:	£25.19
Amount Paid:	£0.00
Balance Due:	£25.19
Invoice Number:	00024502
Payment Status:	Paid
Payment Method:	KLARNA_PAYMENTS Processor: KLARNA_PAYMENTS Transaction: 40ab1ccf-1a51-266b-98a2-0d00fb59261d Amount: £25.19 Klarna Payment Category ID: pay_over_time Klarna Payment Category Name: Buy now, pay later Fraud Status: ACCEPTED

At the bottom left, there is a link "<< Back to List".

Figure 1 Klarna Payment Details in BM

An alternate flow when PENDING status is returned for Klarna order creation, the SCC order creation proceeds with modified statuses. If later a Klarna notification with updated fraud status FRAUD_RISK_ACCEPTED is returned, SCC order status is updated and returns to the usual flow. Klarna payment status is saved in a custom attribute with id kpFraudStatus in the PaymentTransaction system object, and can be seen in BM on the order details Payment tab as below:

General Attributes **Payment** Notes History

Payment Information for Order '00005338'

Order Total:	£191.99
Amount Paid:	£0.00
Balance Due:	£191.99
Invoice Number:	00024524
Payment Status:	Paid
Payment Method:	KLARNA_PAYMENTS Processor: KLARNA_PAYMENTS Transaction: 08aef0f0-22f0-2138-9334-978b637d74dc Amount: £191.99 Klarna Payment Category ID: pay_over_time Klarna Payment Category Name: Buy now, pay later Fraud Status: FRAUD_RISK_ACCEPTED

[**<< Back to List**](#)

Figure 2 Klarna Payment Details in BM

2.2. Locales

The cartridge supports most locales including:

- English
- German
- Danish
- Spanish
- Finnish
- French
- Italian
- Dutch
- Polish
- Mexican

For a list of more [supported](#) locales, contact Klarna.

2.3. Use Cases

2.3.1. Enable Klarna Payments Across International Sites (NA, EU, OC)

Klarna Payments on SFRA can be configured independently on each site by locale.

2.3.2. Multitude of Payment Options for Customers

On the checkout billing step, configured Klarna payment options are dynamically loaded based on customer's cart information and the payment method categories returned for the current Klarna session.

The screen below shows an example of the payment options displayed in the Checkout billing step:

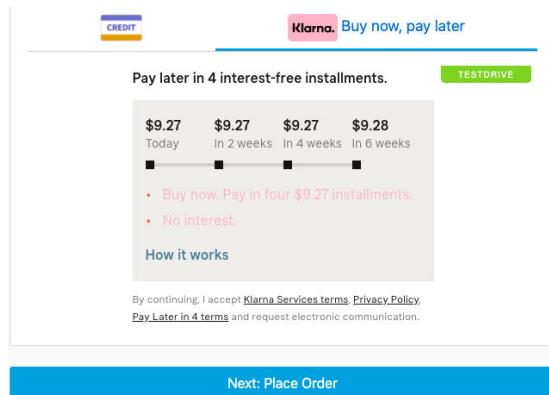


Figure 3 Payment Option

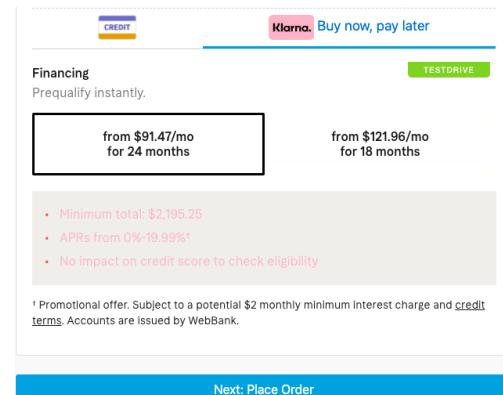


Figure 4 Payment Option

When customer selects (clicks) payment method, a widget with additional information of the Klarna product is displayed as shown above.

Note: The payment methods displayed are based on market availability and/or contractual agreement with Klarna.

2.3.3. Klarna Display Conditions and Authorization Handling

2.3.3.1. Display Conditions

Klarna payment methods may not be displayed under the following conditions:

1. **Payment Categories Not Returned:** Klarna is not displayed if payment categories are not returned in the session response.
2. **Session Linking Issue:** Payment method content is not displayed if we have payment methods in the session, but no session is linked to the basket.
3. **Authorization Denial:** Payment method category is hidden or greyed out if `show_form` is returned as false in the Klarna response on an authorization request and the payment method is denied.
4. **Session Creation Failure:** Overall, if Klarna is unavailable and a session is not created, payment categories will not be displayed.

2.3.3.2. Authorization Handling Preference

We've implemented a site preference, `kpRejectedMethodDisplay`, allowing merchants to choose how to handle authorization rejections (refer to section 2.3.5 Refusal of Klarna Payments on Payment Method – Authorization). The available options are:

- **No action:** No specific handling for authorization rejections.
- **Hide:** Hide the payment method upon authorization rejection.
- **Gray out:** Gray out the payment method upon authorization rejection.

2.3.4. Authorize and Place Klarna Order in Checkout

Cartridge provides best practice implementation and options to include Extra Merchant Data (EMD) to optimize acceptance rate for Klarna Payment (methods) products. This includes customer info and (Buy Online, Pickup in Store) BOPIS details included as merchant data when custom site preference “attachments” is enabled for the site. The EMD data sent can be extended but should be reviewed case by case and optimized and validated based on merchant data & privacy requirements prior to go-live.

The customer's information (personally identifiable information) is sent once customer chooses a Klarna payment method and authorizes (clicks: “Continue to Place Order”). The required customer information facilitates assessment and verification of customer data to display payments method options available for the customer.

When customer clicks “Continue to Place Order” button, the authorization is initiated, and a successful authorization takes customer to summary page. Note that customer may be prompted for additional information prior to completion of authorization based on the payment method selected and market. If the authorization is not successful (“approved = false”), then the customer

will stay on the billing page, the “Continue to Place Order” button will be disabled and depending on the Business Manager settings the payment option may be hidden or grayed out (refer to section 2.3.5 Refusal of Klarna Payments on Payment Method – Authorization).

In some cases, the Pay Now payment category requires additional “finalize” call to be triggered when the customer clicks on the “Place Order” button in the review page. This is needed to ensure that the funds will be transferred (e.g.: Bank Transfer payment methods which initiate fund transfer) only when the customer has decided to place the order on the merchant site.

Please, note that in such cases cookie called “**selectedKlarnaPaymentCategory**” is set in the customer’s browser. This cookie contains the selected payment method from the previous step and is used to initialize the “finalize” call to Klarna.

For more information regarding the authorize & finalize calls, refer to the developer documentation [here](#)

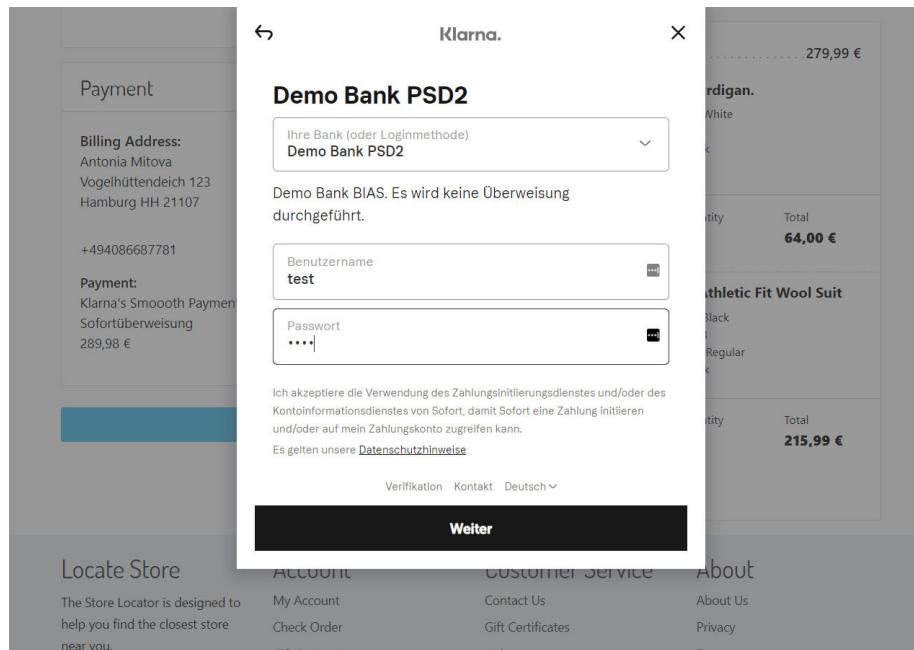


Figure 5 Finalize call screen

The Klarna Order is only placed prior to final step before SFCC order is created.

Successful payment authorizations (Klarna Payments authorization status: APPROVED) followed by placement of Klarna and SCC orders leads to a “Paid” order payment status and “Ready for Export” export status. Refused and pending Klarna Payment orders (Klarna Payments order statuses REJECTED and PENDING) lead to a “Not Paid” order payment status, and “Not Exported” export status.

In case of VCN settlement errors, Klarna orders are cancelled and SFCC orders are failed and set to a “Not Exported” export status.

When customers have selected Klarna payment option as a payment method for the order, successfully authorized the amount on order and later return to the billing page to choose a different payment method that is non-Klarna payment method. In such scenarios, when customer authorizes (non-Klarna payment method) and reaches the review page – automatic “cancelAuthorization” call will be triggered to release the authorized funds (Klarna related) & free up the available purchase amount for this customer.

Merchants have the option to utilize this function to cancel prior authorization when required for specific use-cases apart from the above scenario. If enabled, the checkout flow should be tested thoroughly as part of integration, considering the valid checkout scenarios (e.g. relevant Klarna session flow, Klarna payment method switching, order amount updates, checkout with external payment method, etc.).

```
36 server.get('SaveAuth', function(req, res){  
37   ...  
38   ...  
39   ...  
40   ...  
41   ...  
42   ...  
43   ...  
44   ...  
45   ...  
46   ...  
47   ...  
48   ...  
49   ...  
50 } );  
51 ...
```

Figure 6 Cancel Authorization Call

2.3.5. Refusal of Klarna Payments on Payment Method – Authorization

Upon selecting one of Klarna’s options as the payment method on billing step of checkout and based on the customer information provided, Klarna Payments can be refused as a payment method.

If the payment method was rejected with “**show_form=false**” & “**approved=false**” (i.e., hard reject), the merchant has the option to choose what happens with the payment option display in Billing page using BM preference “Hide Payment Methods on Deny” (**kpRejectedMethodDisplay**):

No – Leave the payment visible to the customers

Hide – The payment option will be hidden from customer

Grey Out – The payment option will be greyed out and not clickable

Please note that reloading the page will show the denied Klarna payment method again.

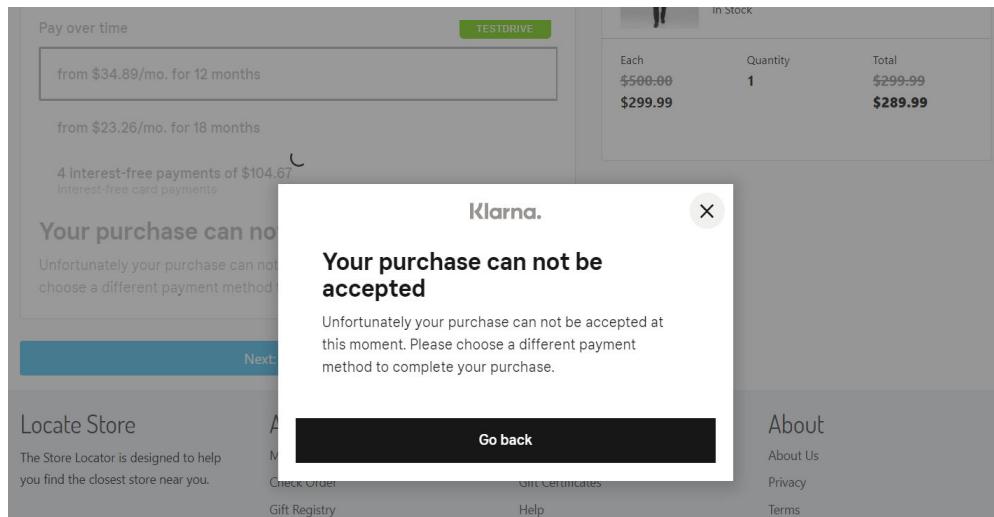


Figure 7 Denied Order Popup

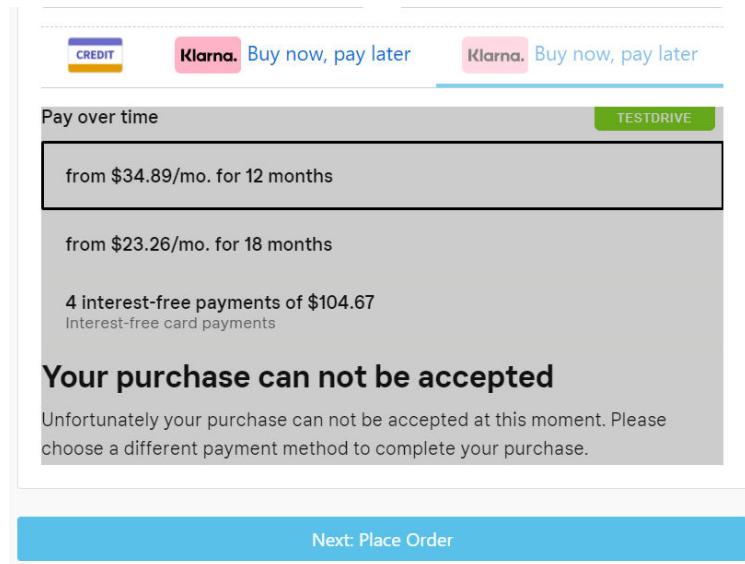


Figure 8 Greyed Out Payment Option

2.3.6. Klarna Payment Option Not Available for Current Purchase - Billing Page

The customer is presented with an appropriate message in the Klarna widget when customer attempts to choose Klarna in the billing step.

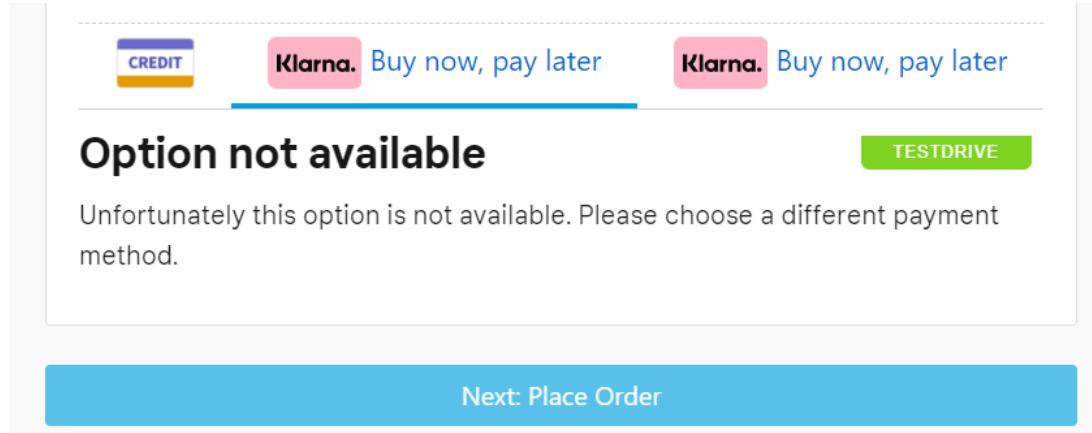


Figure 9 Option Not Available

2.3.7. Klarna Payments Not Available - Checkout

If Klarna API is not available or the site/storefront is not applicable, Klarna is not presented as a payment option in the billing page. It is recommended that Klarna session must not be created when customer chooses a non-Klarna market or merchant store, as well as in cases when merchants have multicurrency storefront with basket currency that is not supported by Karna.

2.3.8. Handling Notifications

In scenarios where the Klarna Order has been created but instead of immediately accepting the order, the Klarna is flagged for additional review. This results in Commerce Cloud order staying in “Created” status with Fraud Status: PENDING. This order is marked with EXPORT_STATUS_NOTEEXPORTED, confirmation status NOTCONFIRMED and NOTPAID.

For Klarna orders with Fraud Status PENDING, once review is complete, updates are sent to the pre-configured notification_url on the merchant Commerce Cloud site. The updated Fraud status depends on the fraud screening, the returned Fraud Status (e.g., FRAUD_RISK_ACCEPTED) is displayed in BM. The push notification is repeatedly sent (up-to 24 hours, every 10 mins) until the POST request is acknowledged with a 200 response.

Klarna sends one of the following event types in the notification to SFCC to update risk status: FRAUD_RISK_ACCEPTED, FRAUD_RISK_REJECTED, FRAUD_RISK_STOPPED.

The notification updates are generally received within 4-24 hours. The order's payment transaction is updated (see **kpFraudStatus**). This can be seen in BM on the order details Payment tab as below:

Merchant Tools > Ordering > Orders > Order: 00005338(RefArchGlobal)

General Attributes **Payment** Notes History

Payment Information for Order '00005338'

Order Total:	£191.99
Amount Paid:	£0.00
Balance Due:	£191.99
Invoice Number:	00024524
Payment Status:	Paid
Payment Method:	KLARNA_PAYMENTS Processor: KLARNA_PAYMENTS Transaction: 08aef0f0-22f0-2138-9334-978b637d74dc Amount: £191.99
	Klarna Payment Category ID: pay_over_time Klarna Payment Category Name: Buy now, pay later Fraud Status: FRAUD_RISK_ACCEPTED

<< Back to List

Figure 10 Fraud Status in Payment Details

If the order is “FRAUD_RISK_ACCEPTED” upon notification, the order will be placed in SFCC (order status changes to OPEN). The order is marked with status CONFIRMATION_STATUS_CONFIRMED and export status EXPORT_STATUS_READY. If the order was placed with auto-capture, the payment status will be set to PAYMENT_STATUS_PAID and the full order amount will be captured.

If the order is “FRAUD_RISK_REJECTED” or “FRAUD_RISK_STOPPED” upon notification, the order is failed (FAIL) in SFCC.

Note: The Klarna pending functionality availability is dependent on markets and enabled based on the contractual agreement with Klarna.

2.3.9. Virtual Cards Settlements

This option is disabled by default. However, 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 via site preference “**kpVCNEnabled**”:

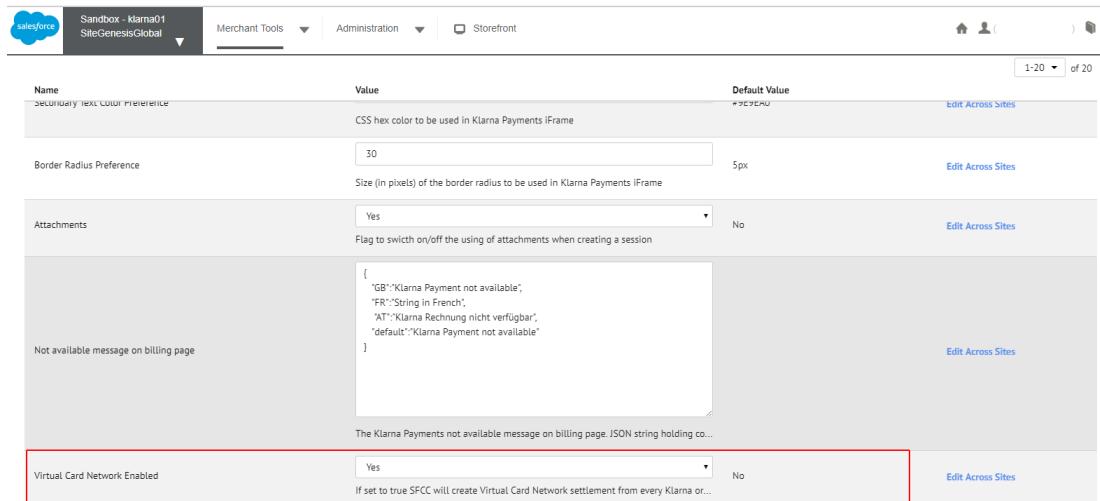


Figure 11 VCN Enablement Setting

The virtual card (see note) is issued against an Klarna order, for the purpose of capturing the Klarna authorized order amount using the standard card rails.

When a customer places an order, the order is first booked in SFCC. Once an order has been accepted by Klarna, the Klarna cartridge integration creates a virtual card-based settlement, utilizing the merchant card services (MCSv3) API.

Once a settlement has been created (virtual card returned), the encrypted card details are saved in SFCC and can be later utilized by the merchant OMS platform or custom PSP integration to authorize the virtual card until the Klarna order is valid. On successful fulfillment of the order, the funds on the virtual card maybe 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 the merchant using the issued virtual credit card instead of direct bank account transfer.

Please review the procedures with Klarna team for details of settlement process using virtual cards.

Refer to the below code in “**scripts/payments/processor.js**” and update accordingly to the integrated cards processor.

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

For decrypting the credit card details refer to **9.3 Decrypt VCN Card Details**.

```

118 /**
119 * Call Credit Card Authorization Hook (for VCN settlement)
120 * @param {dw.order.order} order DW Order
121 * @returns {processorResult} authorization result
122 */
123
124 function callCreditCardAuthorizationHook(order) {
125   var processorResult = null;
126   var paymentInstrument = order.getPaymentInstruments(PAYMENT_METHOD)[0];
127   var paymentProcessor = PaymentMgr[
128     .getPaymentMethod(paymentInstrument.paymentMethod);
129     .paymentProcessor;
130   ];
131   var transactionID = paymentInstrument.getPaymentTransaction().getTransactionID();
132   var hook = 'app.payment.processor.' + CREDIT_CARD_PROCESSOR_ID;
133   if (!HookMgr.hasHook(hook)) {
134     throw new Error(`File of app.payment.processor.${CREDIT_CARD_PROCESSOR_ID} hook is missing or the hook is not configured`);
135   }
136
137   processorResult = HookMgr.callHook('app.payment.processor.' + CREDIT_CARD_PROCESSOR_ID, 'Authorize', transactionID,
138     paymentInstrument, paymentProcessor);
139   return processorResult;
140 }
141

```

Figure 12 Credit Card Authorization Call

To utilize virtual card integration option the merchant should:

- Enable VCN option in Site Preferences as shown above
- Enter the VCN Public Key ID. Unique UUIDv4 value, which should be different for playground testing and Production (live site)

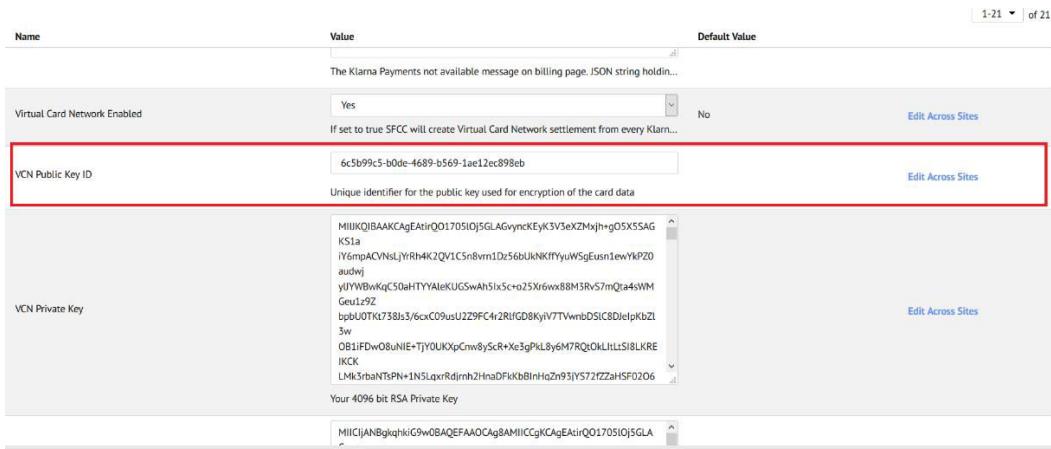


Figure 13 VCN Public Key ID

- Generate a 4096-bit RSA key pair (Refer to section Generate Key Pair and Key Id for Virtual Card Settlements). Set the custom preference “**vcnPublicKey**” with the value of the public key without the header and footer lines (begin and end public key) and the custom preference “**vcnPrivateKey**” with the value of the private key without the header and footer lines (begin and end private key)

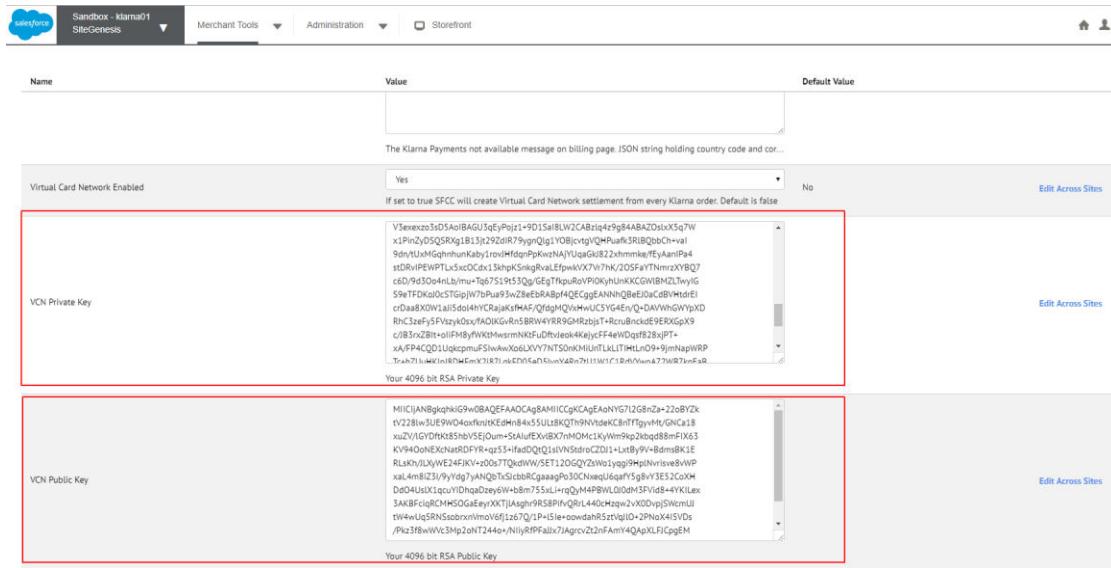


Figure 14 VCN Public & Private Keys

- Update the VCN settlement retry setting “**kpVCNRetryEnabled**” (Deprecated with v24.4.0) “**kpVCNRetry**” . By default, this is disabled. However, if enabled and in cases when a VCN creation error is returned, the application will retry the settlement request once again with `order_id` as the idempotency key.
- Finally, you need to send the generated unique `key_id` + public key combination in JWK format to Klarna prior to testing and go-live. It will be used to encrypt the aes key which is used for encrypting the pci data on Klarna side when settlement request is made. After confirmation from Klarna that the key has been successfully added to your merchant profile you would be able to use virtual card-based settlement option for Klarna payment methods

If enabled and fully configured, virtual card settlement request is made successfully. For orders placed with the VCN settlement option, the related custom attributes are shown below:

Merchant Tools > Ordering > Orders > Order: 00049309(RefArch)

General Attributes Payment Notes History

Attributes for Order '00049309'

On this page you can edit the attributes of the order. Fields with a red asterisk (*) are mandatory. Click **Apply** to save changes. Click **Reset** to revert your changes.

Klarna Payments	
Klarna Payments Order ID:	72bf2c96-6523-2add-8c50-f2af87712019
Is VCN Used:	<input checked="" type="checkbox"/>
VCN Card ID:	befbb0e9-5e98-4e39-9c00-75aba0c3372b

[<< Back to List](#)

Figure 15 VCN Details in Order

If required, the additional virtual card details can be assigned to this group in Administration > Site Development > System Object Types > select “Order”. In the Attribute Grouping tab select Klarna_Payments and click on “edit”. Assign the new attributes and save the data.

Administration > Site Development > System Object Types > Order - Attribute Groups > Klarna Payments

Object Type 'Order' - Attribute Definition Assignments

On this page you can assign existing attribute definitions to your attribute group.

Assign Attribute Definition						
ID*	ID	Name	Type	Attribute Settings	Sorting	
<input type="checkbox"/>	kpoOrderID	Klarna Payments Order ID	String			
<input type="checkbox"/>	kplIsVCN	Is VCN Used	Boolean			
<input type="checkbox"/>	kplVCNcardID	VCN Card ID	String			
<input type="checkbox"/>	kplVCNHolder	VCN Holder	String			
<input type="checkbox"/>	kplVCNBrand	VCN Brand	String			
<input type="checkbox"/>	kplVCNPIData	VCN PCI Data	String			
<input type="checkbox"/>	kplVCNIV	VCN Initialization Vector	String			
<input type="checkbox"/>	kplVCNAESKey	VCN AES Key	Text			

[Unassign](#)

[<< Back](#)

Figure 16 Full List of VCN Attributes

Please work with Klarna Account Manager and Delivery contact in advance to select the appropriate virtual card product based on your business requirements and use-cases. You can find information [here](#) around other use cases supported.

Important Note!

DO NOT SAVE DECRYPTED PCI DATA 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 (see section **3.5.1** Job “OrderCleanUp”).

2.3.10. Auto-Capture

Auto-capture is enabled via a site preference “**kpAutoCapture**” located in “**Klarna_Payments**” preference group.

When the preference is enabled (disabled by default), a full amount capture is attempted. If the capture is successful, the SFCC order’s payment transaction is marked as Paid, and viewable in the Business Manager.

The order will be marked as “*Captured*” in the Klarna’s Merchant Portal

The screenshot shows the Klarna Merchant Portal interface. At the top, it displays basic order information: Merchant reference I (00029205), Klarna reference (BJJGH9P1), Created (Jun 25, 2019, 5:00 PM), Expires (Jul 23, 2019, 5:00 AM), and Merchant ID (K500726). Below this, the "Customer" section shows shipping and billing addresses for a customer named Angela Gill. The "Order lines (2)" section lists two items: "Black Flat Front Wool Suit" and "Наземен транспорт". The "PAYOUT DETAILS" section shows the payment method as "Initial Payment Method" and the status as "Captured". The "ORDER TOTAL" section shows a total of £199.98. The "Activity Log" at the bottom tracks three events: "Order placed: £199.98" (5:00 PM, Jun 25, 2019, Via API), "Captured: £199.98" (5:00 PM, Jun 25, 2019, Via API), and "Order acknowledged" (5:00 PM, Jun 25, 2019, Via API).

Figure 17 Order Details in Klarna Portal

If the capture is unsuccessful, an error will be logged in the custom error log. The setting must be reviewed with Klarna delivery team before testing and go-live.

Note: Auto-capture is possible for orders when VCN is not enabled!

2.3.11. Widget Customizations

Note: The merchant will need a configured Klarna Payments account.

The merchant can style the Klarna Payments widget (skin), to match the marketing and branding needs of their store. The list with the graphic elements that can be customized out of the box through site preferences are listed below:

```
"color_details" (site preference kpColorDetails): "#C0FFEE"  
"color_button" (site preference kpColorButton): "#C0FFEE"  
"color_button_text" (site preference kpColorButtonText): "#C0FFEE"  
"color_checkbox" (site preference kpColorCheckbox): "#C0FFEE"  
"color_checkbox_checkmark" (site preference kpCheckboxCheckmark): "#C0FFEE"  
"color_header"(site preference kpColorHeader): "#C0FFEE"  
"color_link"(site preference kpColorLink): "#C0FFEE"  
"color_border"(site preference kpColorBorder): "#C0FFEE"  
"color_border_selected"(site preference kpBorderSelected): "#C0FFEE"  
"color_text"(site preference kpColorText): "#C0FFEE"  
"color_text_secondary"(site preference kpColorTextSecondary): "#C0FFEE"  
"radius_border"(site preference kpRadiusBorder): "0px"
```

Note : Site preferences kpColorDetails, kpColorButton, kpColorButtonText, kpColorCheckbox, kpCheckboxCheckmark, kpColorHeader, kpColorLink, kpColorBorder, kpBorderSelected, kpColorText, kpColorTextSecondary, kpRadiusBorder are deprecated .A new site preference kpColorCustomization is created under klarna_KP which includes all the parameters mentioned in a JSON

2.3.12. Customizing Payment Method Name

The payment method name “*Klarna Payments*” may be customized via the “*Merchant Tools > Ordering > Payment Methods*” section in Business Manager.

The screenshot below shows the “*Klarna*” method selected and the administrator choosing a language from the drop-down.

Payment Methods

The screenshot shows a list of payment methods: BANK_TRANSFER, BML, CREDIT_CARD, DW_ANDROID_PAY, DW_APPLE_PAY, and GIFT_CERTIFICATE. Below this, there is a table with two rows. The first row has 'Klarna' in the ID column and 'Rechnung und Ratenkauf' in the Name column, both of which are also highlighted with red boxes. The second row is for 'PayPal'. On the right side of the screen, a language selection dropdown is open, showing a list of languages. The 'German (Germany)' option is selected and highlighted with a red box.

Figure 18 Customize Payment Name

The payment method name would then be visible in the mini summary & confirmation screens, the confirmation emails and My Account Order Details section.

Item	Quantity	Price	Shipping To
Straight Fit Shorts	1	112,00 €	Testperson-de Approved

Figure 19 Payment Method Name in Email

2.3.13. Klarna On-Site Messaging

On-site messaging is a platform that enables you to add tailored messaging to your website. With On-site messaging you can let shoppers know about the different payment options you have available as they browse your site. By using Klarna, customers have access to our flexible

payment options in the checkout; On-site messaging is a great way to let them know even before they decide to buy.

The Klarna Payment cartridge provides multiple options in the standard implementation based on the reference architecture:

- Product Page and Cart based promotions
- Sitewide
 - Top Banner strip
 - Footer logo
- Custom Info-page

Klarna On-Site Messaging (OSM) is configured by site and by locale via the **KlarnaCountries** custom object.

To configure the OSM settings for a locale, you must visit “**Merchant Tools – Custom Object Editor**” and search for **KlarnaCountries** custom object. Select the country Key, e.g. “US” .

In the custom country specific configuration, provide a valid locale for the OSM tag based on the country being configured.

The OSM Data Client ID and Data keys required are available in Klarna Merchant Portal within the On-site Messaging App:

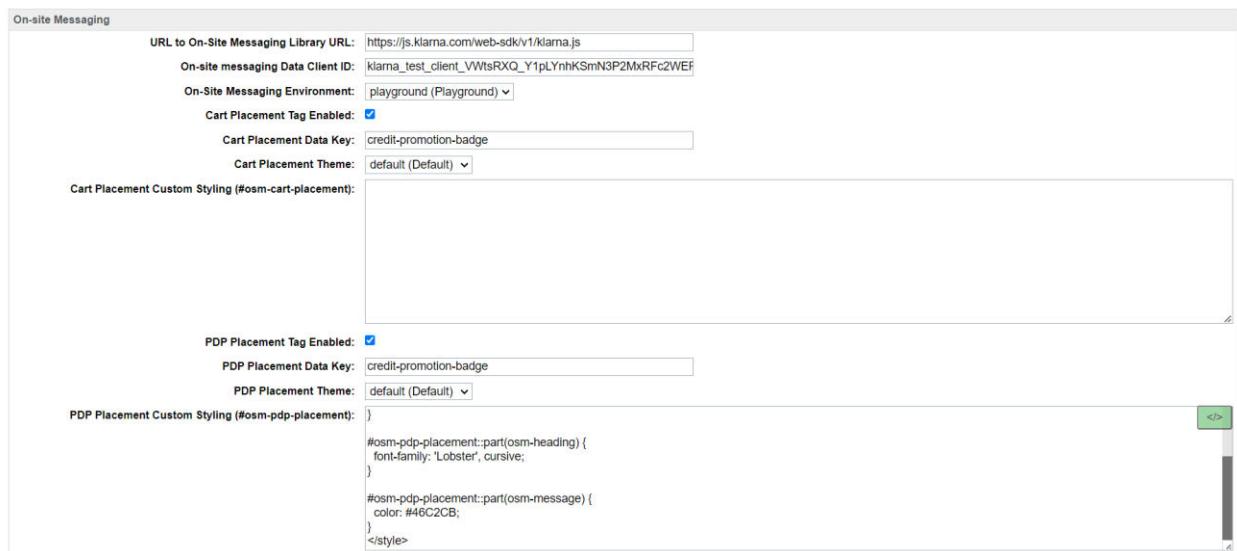


Figure 20 OSM Settings in BM

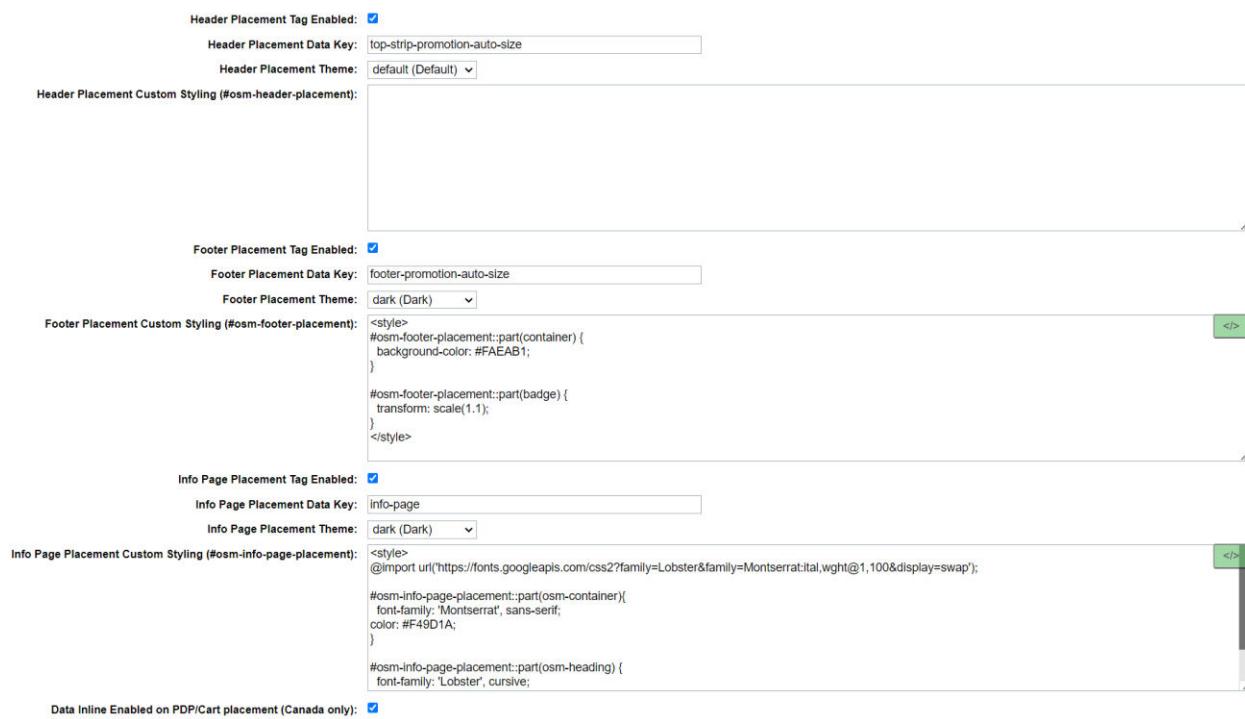


Figure 21OSM Settings in BM (cont.)

To enable Placement tag for the Cart Page, the “Cart Placement Data Key” must be filled with Data Key value and the “Cart Placement Tag Enabled” must be checked. To customize the Cart Placement – theme could be selected and/or custom CSS entered in “Cart Placement Custom Styling” attribute using the placement id - **#osm-cart-placement**.

Note: Cart placements amount must be [updated](#) and latest Klarna credit offering placement (where required) displayed to customer when order line quantity is updated on cart page.
Refer to [3.6.2 app_storefront_base\cartridge\client\default\js\cart\cart.js](#) (required)

To enable Placement tag for the PDP Page, the “PDP Placement Data Key” must be filled with Data Key value and the “PDP Placement Tag Enabled” must be checked. To customize the PDP Placement – theme could be selected and/or custom CSS entered in “PDP Placement Custom Styling” attribute using the placement id - **#osm-pdp-placement**.

To enable Placement tag for header, the “Header Placement Data Key” must be filled with Data Key value and the “Header Placement Tag Enabled” must be checked. To customize the Header Placement – theme could be selected and/or custom CSS entered in “Header Placement Custom Styling” attribute using the placement id - **#osm-header-placement**.

To enable Placement tag for footer, the “Footer Placement Data Key” must be filled with Data Key value and the “Footer Placement Tag Enabled” must be checked. To customize the Footer

Placement Tag – theme could be selected and/or custom CSS entered in “Footer Placement Custom Styling” attribute using the placement id - **#osm-footer-placement**.

To enable Placement tag for the Info Page, the “Info Page Placement Data Key” must be filled with Data Key value and the “Info Page Placement Tag Enabled” must be checked. To customize the Info Page Placement Tag – theme could be selected and/or custom CSS entered in “Info Page Placement Custom Styling” attribute using the placement id - **#osm-info-page-placement**.

In Library URL, please input the full URL to the On-Site Messaging JavaScript Library.

In On-Site Messaging Environment please select the corresponding environment playground for test or production.

In On-site messaging Data Client ID, please input the On-Site Messaging Data client ID value.

On-site messaging Data locale, please input the valid On-Site Messaging Data locale.

All the placements can be customized with custom CSS using the placement component id – in the brackets of attribute name. The customization value should be wrapped in <style> element:

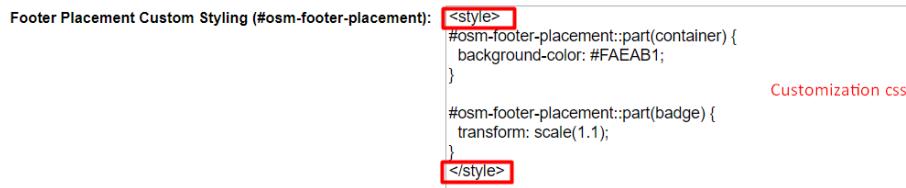


Figure 22 OSM Placements CSS customization

Note: Current configuration of using attributes from Custom Object is deprecated with version 24.3. In the new implementation, settings are retrieved from Site Preferences as explained below

The screenshot shows the "Site Preferences" configuration page in the Klarna Payments for SFRA v25.4.0 interface. It displays several configuration fields:

Name	Value	Default Value	Action
(osm_enable)	Yes	Yes	Edit Across Sites
Theme*	Light (recommended) (Light (recommended))		Edit Across Sites
Placement*	Product Page (productPage) Cart Page (cartPage) Site Wise Banners (siteBanners) FAQ (faq) Footer (footer)		Edit Across Sites
Placement			Edit Across Sites
(osm_custom_styling)			

Fig 23 OSM configuration in BM

Current configuration under BM is placed in **Merchant Tools – Site Preferences-Custom Preference under Klarna_OSM group**

To enable Klarna OSM implementation in storefront ,**osm_enable** must be set to true.

osm_theme dropdown is used to apply theme to OSM .Options include default,dark and custom.

To enable placement in pdp, cart, header, footer and info page **osm_placement** is enabled in the mutiselect dropdown. Placement tag ID is hardcoded from code .

Placement	TagId
Product Page	credit-promotion-auto-size
Cart Page	credit-promotion-badge
Site Wide Banner	top-strip-promotion-badge
FAQ	info-page-auto-size
Footer	footer-promotion-auto-size

Table: Placement tag id for different pages

Placement styling is taken from **osm_placement_styling** which is a JSON.

osmLibraryURL is hardcoded and retrieved from constants.

osmEnvironment is used from KlarnaPayments.

CSS customizations are available only for the new OSM library version.

Please follow Klarna guidance for placement styling - <https://docs.klarna.com/on-site-messaging/in-depth-knowledge/styling-on-site-messaging-with-css/>.

For Canada only, please update “Data Inline Enabled on PDP/Cart placement (Canada only)” value as follows:

- For PayBright enabled payment methods – set to true.
- For Klarna enabled payment methods – set to false.

Note : As per new implementation, osmDataInlineEnabled is deprecated

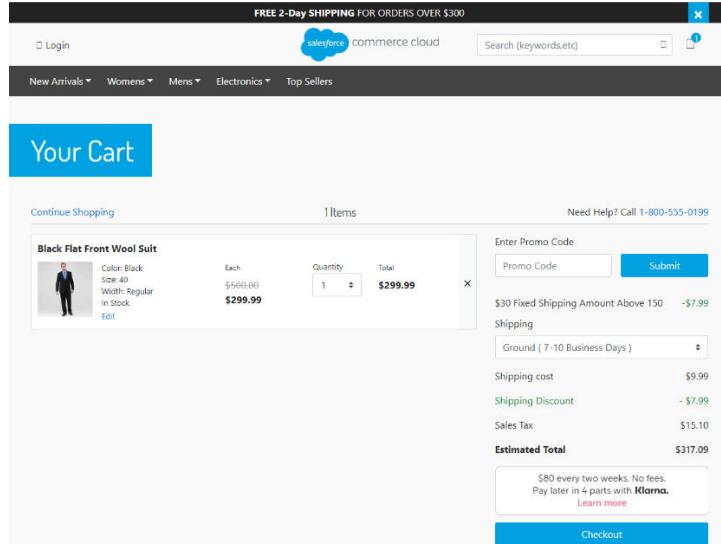


Figure 23 On-Site Messaging Enabled on Cart Page

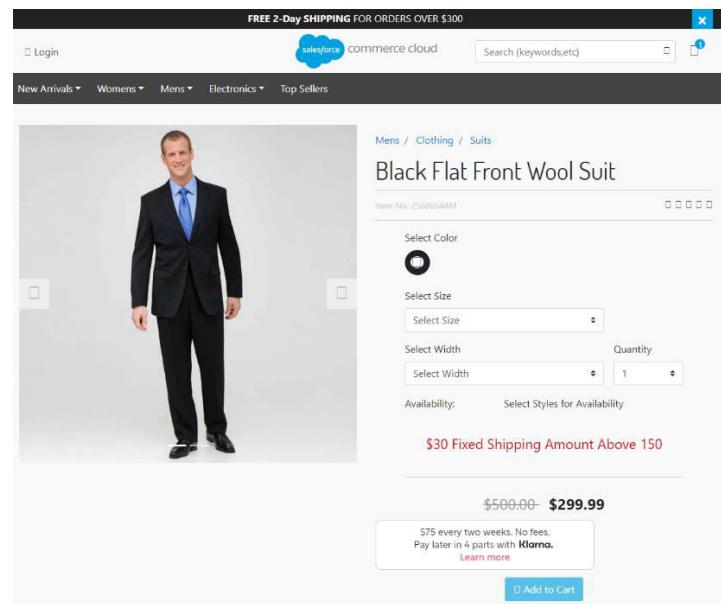


Figure 24 On-Site Messaging Enabled on PDP Page

In addition to the above, if you wish to display the dedicated (custom) Klarna info OSM page you can use the following controller endpoint “**KlarnaPayments-InfoPage**”. For example, you should update the “**footer-about**” content asset to include this line of code as shown on the screenshot.

```
<li><a href="$url('KlarnaPayments-InfoPage')$" title="Go to Klarna Info">Klarna  
Info</a></li>
```



Figure 25 Footer Asset Update

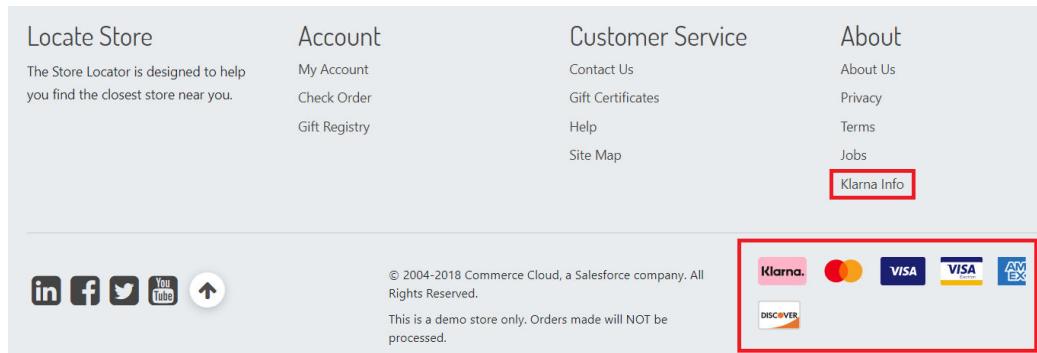


Figure 26 On-Site Messaging Enabled on Footer and link to Klarna Info Page

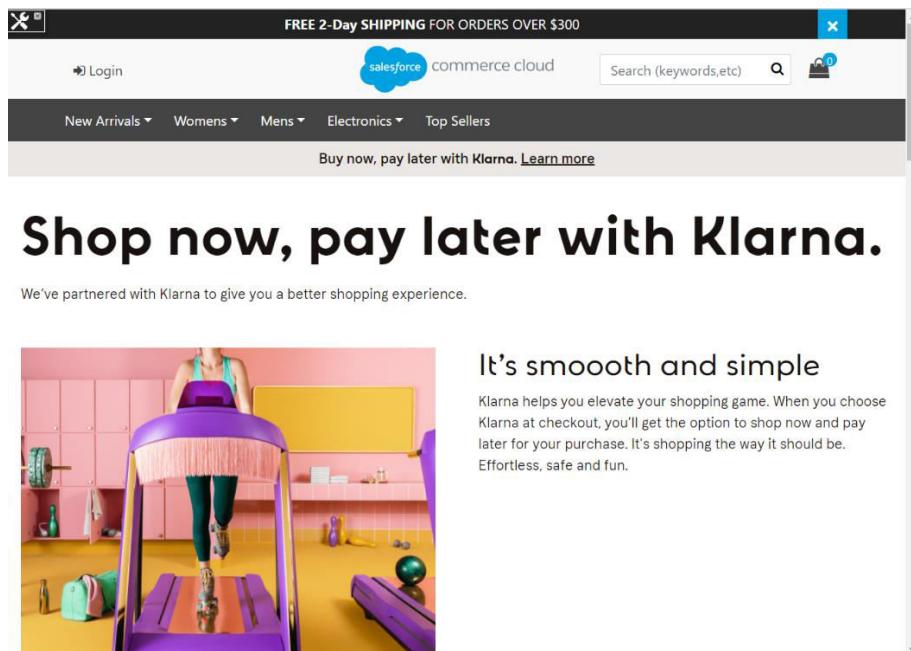


Figure 27 On-Site Messaging Enabled on Header & dedicated Klarna Info Page

For more information regarding OSM customizations and best practices, please refer to the Klarna [Developer/ Docs](#)

Integration Best practice + information about Klarna Branding and Co-marketing options here.

Note: It is the merchant's responsibility to ensure that user content is collected when required for OSM placements for your local market to abide by the legal requirement. e.g., [EU cookie-guide](#) in Klarna Merchant Portal

2.3.14. Klarna Express Button

The cartridge supports the **Klarna Express button (KEB)** on the standard Cart page and mini-Cart.

By enabling Klarna's Express button on the cart page of your website, shoppers can choose to log into their (*or sign up for a*) Klarna account and have their personal details pre-filled in the checkout. Klarna's payment method will be pre-selected for the shopper. In [supported](#) markets, Klarna's network of shoppers benefit from a pre-filled checkout experience, which also includes first time shoppers on a merchant storefront.

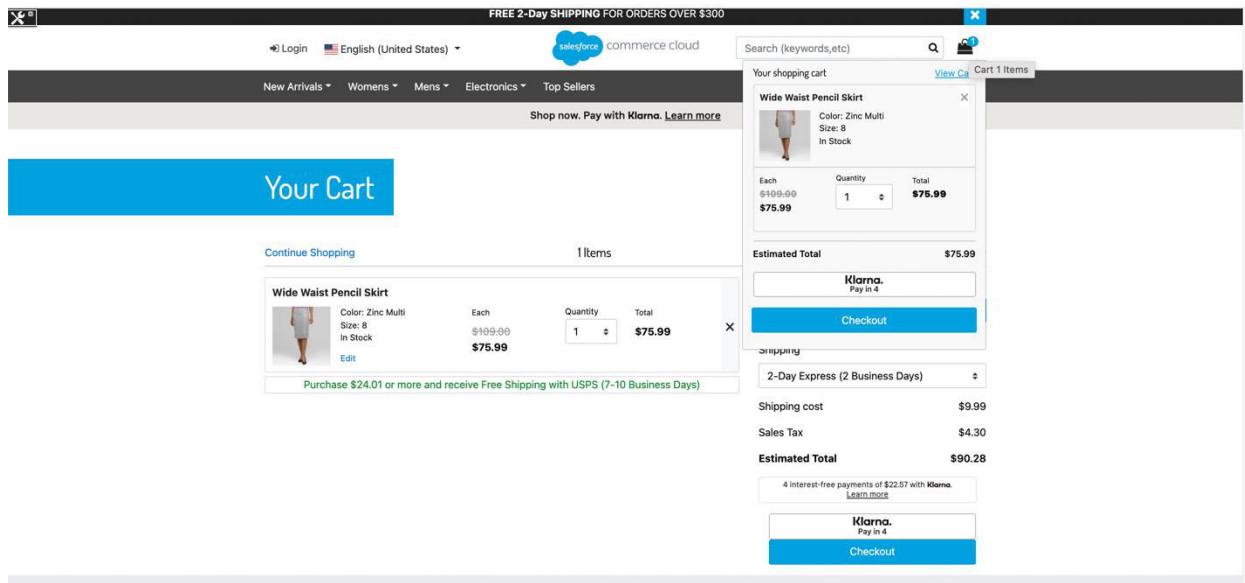


Figure 28 Klarna Express Button on Cart Page

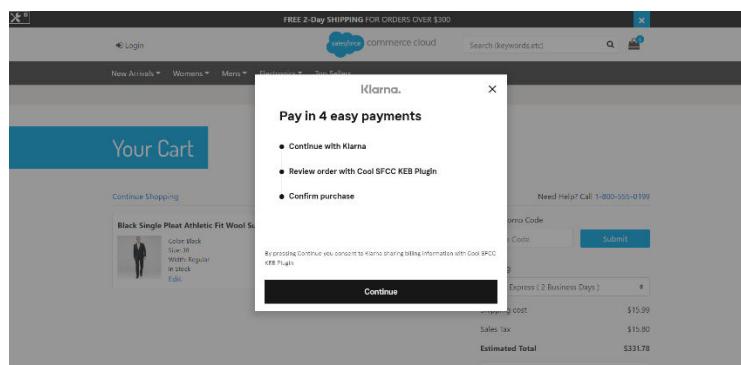


Figure 29 Klarna Express Button clicked Cart Page

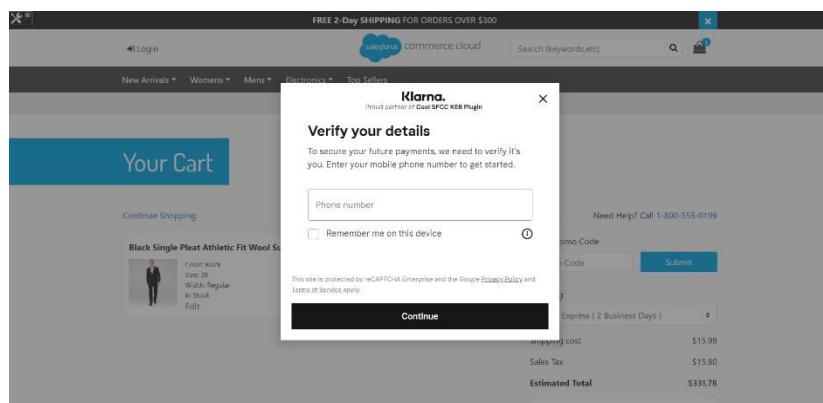


Figure 30 Klarna Express Button user credentials input actions

Once the credentials have been provided and the shopper has successfully authenticated with their Klarna account, they are redirected to the checkout page with relevant details pre-populated and Klarna Payment (e.g.: “Pay in 4”) method pre-selected on the billing page.

The screenshot shows a Klarna checkout interface. On the left, there's a 'Shipping' section with an address for John Doe at Lombard St 10, Beverly Hills, CA 90210, and a note about a 2-day express delivery option costing \$15.99. Below that is a 'Payment' section where the billing address is set to the same address. It includes fields for email (someone@example.com) and phone number (+16173279035). A 'Klarna' button is visible, indicating the selected payment method. To the right is an 'Order Summary' table:

Order Summary	
Subtotal	\$289.99
Shipping	\$15.99
Sales Tax	\$15.30
Total	\$321.28

Below the summary is a product detail section for a 'Black Single Pleat Athletic Fit Wool Suit' in size 38, regular fit, with a price of \$289.99. At the bottom, there's a timeline for 4 interest-free payments of \$80.32, with options for payment today or in 2 weeks, 4 weeks, or 6 weeks. A note states the product is trusted by over 15 million Americans. At the very bottom is a 'Next: Place Order' button.

Figure 31 Checkout details prefilled and Klarna Pay in 4 selected

For storefronts with custom checkout design or address field validations, please ensure that the authenticated users address data is populated without modification to ensure the Klarna Payment method authorization is completed successfully.

Note: On re-direct, in the standard checkout, any existing email ID and phone number are updated with the latest provided by the shopper

2.3.15. Klarna Payment Method Based Promotions

As of B2C 20.7 release, merchants can include payment methods as qualifier for product, order & shipping promotions.

OOTB when such promotion is set up to use a payment method as qualifier, the total order amount will be visible to the customer once they reach the review page. This means that the Klarna authorization call will be made for larger amount than the final one.

To address this issue, once the customer clicks on any of the payment options in the billing section – a call will be made to the backend. This will re-calculate the basket totals if any promotions are applicable and will update the Klarna session details. As a result, the Klarna iframe widgets and the mini summary section on the storefront will update to show the final order details.

When the selected payment method is non-Klarna one, this logic should be customized by the merchant to handle any 3rd party payment integrations.

2.3.16. Price Adjustment Taxation Handling

OOTB the Klarna API calls will send the product / shipping method details and the relevant discounts as separate lines items as shown below:

```
"order_lines": [
  {
    "type": "discount",
    "name": "5 Off Ties Promotion",
    "reference": "682875540326M_5_off_ties_promotion",
    "quantity": 1,
    "merchant_data": "5ties",
    "unit_price": -500,
    "tax_rate": 500,
    "total_amount": -500,
    "total_tax_amount": 0,
    "total_discount_amount": 0,
    "product_url": null,
    "image_url": null
  },
  {
    "type": "physical",
    "name": "Checked Silk Tie",
    "reference": "682875540326M",
    "quantity": 1,
    "merchant_data": "",
    "unit_price": 1919,
    "tax_rate": 500,
    "total_amount": 1919,
    "total_tax_amount": 68,
    "total_discount_amount": 0,
  }
]
```

Figure 32 Line Items with Default Taxation Setting

In some cases, merchants using gross taxation might enable the “Tax Products and Shipping Only Based on Adjusted Price” preference under “Merchant Tools > Site Preferences > Pricing and Promotion” where the price adjustments are not taxed.

The setting, called “**kpPromoTaxation**” has been introduced, where merchants should update this to match the promotion setting below:

- price (Based on Price) – The product, shipping and their discounts will be sent as separate lines. This is the default setting.

- adjustment (Based on Adjusted Price) – When this is selected, the product or shipping method line item will be sent with attribute “total_amount” matching the prorated price and attribute “total_discount_amount” – matching the total sum of all discounts for this item.

```

"order_lines": [
  {
    "type": "physical",
    "name": "Checked Silk Tie",
    "reference": "682875540326M",
    "quantity": 1,
    "merchant_data": "",
    "unit_price": 1919,
    "tax_rate": 2200,
    "total_amount": 1419,
    "total_tax_amount": 256,
    "total_discount_amount": 500,
    ...
  }
]
  
```

Figure 33 Line Items with "Based on Adj." Taxation Setting

Note: Enabling this setting is not required for storefronts with net taxation as the tax is not included in the products base price. The total order sales tax is sent as a separate line item to Klarna and not on product/shipping line-item level.

2.3.17. Buy Online, Pickup in Store (BOPIS)

When store pickup has been enabled on the storefront, the integration will send stores details to Klarna in the authorization request and when placing the Klarna order. Store information is not sent prior the interaction of the customer with the Klarna payment method widgets.

The store address(es) is always included in the EMD attachment “other_delivery_address” when applicable.

The address included on the shipping address in the Klarna order with store pick-up, is as below:

- Orders that have 1 or more store pickup shipments (no home delivery address), the first store shipment details will be set as the shipping address
- Order with store pick-up(s) and home delivery shipment, home delivery address will be used as the shipping address in the Klarna calls
- If the order contains no store pickups, no information is sent in “other_delivery_addresses” attribute

{

```

"attachment": {
    "content_type": "application/vnd.klarna.internal.emd-v2+json",
    "body": {
        "other_delivery_address": [
            {
                "shipping_method": "store pick-up",
                "shipping_type": "normal",
                "first_name": "Test",
                "last_name": "Customer",
                "street_address": "1487 Bay St",
                "street_number": "",
                "postal_code": "01109",
                "city": "Springfield",
                "country": "US"
            }
        ]
    }
}

```

For more information on the options refer [here](#)

2.3.18. Configuration Support for Service Rate Limits

Klarna Payment API sets rate limits by operation (session creation, order creation, etc) to maintain a high quality of service for all its customers. A merchant has the flexibility to request higher rate limits for a specific duration. The duration of such events may last the period of high traffic events (e.g.: Flash sales or Holiday shopping). To enable these agreed rate limits, the service rate limits feature can be utilized.

The Service ID (e.g.: klarna.http.createSession) and Service Profile (e.g.: klarna.http.createSession) for default operations are included in the meta file.

The screenshot shows the Salesforce Administration interface with the path: Administration > Operations > Services. Under the Services tab, there is a table listing various services. The columns are Select All, Name, Type, and Profile. The table contains 14 entries, mostly HTTP services for Braintree, PayPal, and Klarna.

Select All	Name	Type	Profile
<input type="checkbox"/>	int_braintree.http.xml.payment.Braintree	HTTP	Braintree Default Profile
<input type="checkbox"/>	int_paypal.http.nvp.payment.PayPal_RefArch	HTTP	PayPal Default Profile
<input type="checkbox"/>	klarna.http.cancelAuthorization	HTTP	klarna.http.cancelAuthorization
<input type="checkbox"/>	klarna.http.cancelOrder	HTTP	klarna.http.cancelOrder
<input type="checkbox"/>	klarna.http.createCapture	HTTP	klarna.http.createCapture
<input type="checkbox"/>	klarna.http.createOrder	HTTP	klarna.http.createOrder
<input type="checkbox"/>	klarna.http.createSession	HTTP	klarna.http.createSession
<input type="checkbox"/>	klarna.http.defaultEndpoint	HTTP	klarna.http.service
<input type="checkbox"/>	klarna.http.getOrder	HTTP	klarna.http.getOrder
<input type="checkbox"/>	klarna.http.getSession	HTTP	klarna.http.getSession
<input type="checkbox"/>	klarna.http.updateSession	HTTP	klarna.http.updateSession
<input type="checkbox"/>	klarna.http.vcnSettlement	HTTP	klarna.http.vcnSettlement

Default Klarna Service Operations:

- Create Session
- Update Session
- Create Order
- Cancel Order
- Capture Order (Full Capture)
- New Order Settlement (Virtual Card)

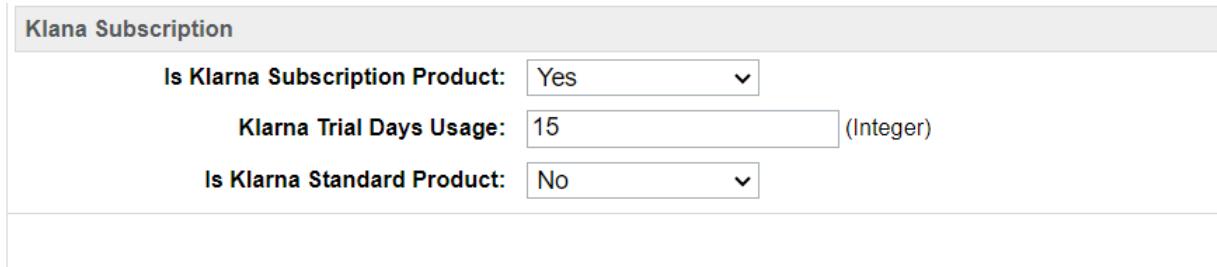
For more information on the options refer [here](#)

2.3.19. Klarna Subscriptions

The cartridge supports subscription handling.

2.3.19.1. Configuration

Subscription details are configured on product level. Products can be subscription, standard or both. The trial period should be an integer value.



The screenshot shows a configuration interface for Klarna Subscription. It has a header 'Klarna Subscription' and three input fields:

- Is Klarna Subscription Product:** A dropdown menu set to 'Yes'.
- Klarna Trial Days Usage:** An input field containing '15' with the note '(Integer)'.
- Is Klarna Standard Product:** A dropdown menu set to 'No'.

Figure 34 Product subscription configuration

2.3.19.2. Cart page

- Subscription products

Subscription-only products are automatically added to the shopping cart as subscription line items. For products that can be both subscription and standard, users can select their preference on the cart page. of standard and subscription products with different trial periods, or if some products

have a trial period while others do not. It's important to note that the checkout process won't proceed if the cart contains a mix

[Continue Shopping](#)
3 Items

3/4 Sleeve Boat Neck Striped Knit



Color: Navy Multi	Each	Quantity	Total
Size: L	\$79.00	1	\$79.00
In Stock	\$71.10		\$71.10
Edit	10% off Summer Flats standard and subscription product without trial period		

[Subscribe](#) This product can be purchased as a Subscription with Klarna payments. product without trial period

3/4 Sleeve V-Neck Top



Color: Butter	Each	Quantity	Total
Size: L	\$24.00	1	\$24.00
In Stock	\$21.60		\$21.60
Edit	10% off Summer Flats standard and subscription product with trial period		

[Subscribe](#) This product can be purchased as a Subscription with Klarna payments. product with trial period

Subscription Trial period: 12 days

3/4 Sleeve Solid Drape Neck Sweater



Color: Carnation	Each	Quantity	Total
Size: M	\$46.99	1	\$46.99
In Stock	\$42.29		\$42.29
Edit	10% off Summer Flats subscription only product with trial period		

[Subscribe](#) This product can be purchased as a Subscription with Klarna payments. subscription only product with trial period

Subscription Trial period: 12 days

Figure 35 Subscription products on cart page

- Subscription details

Dropdown menus with predefined values appear on the cart page when there's at least one subscription product in the cart. The configuration of these values can be managed in the Administration panel under "Site Development" > "System Object Types" > "Basket - Attribute Definitions." The attributes available for configuration are:

- kpSubscriptionFrequency
- kpSubscriptionPeriod

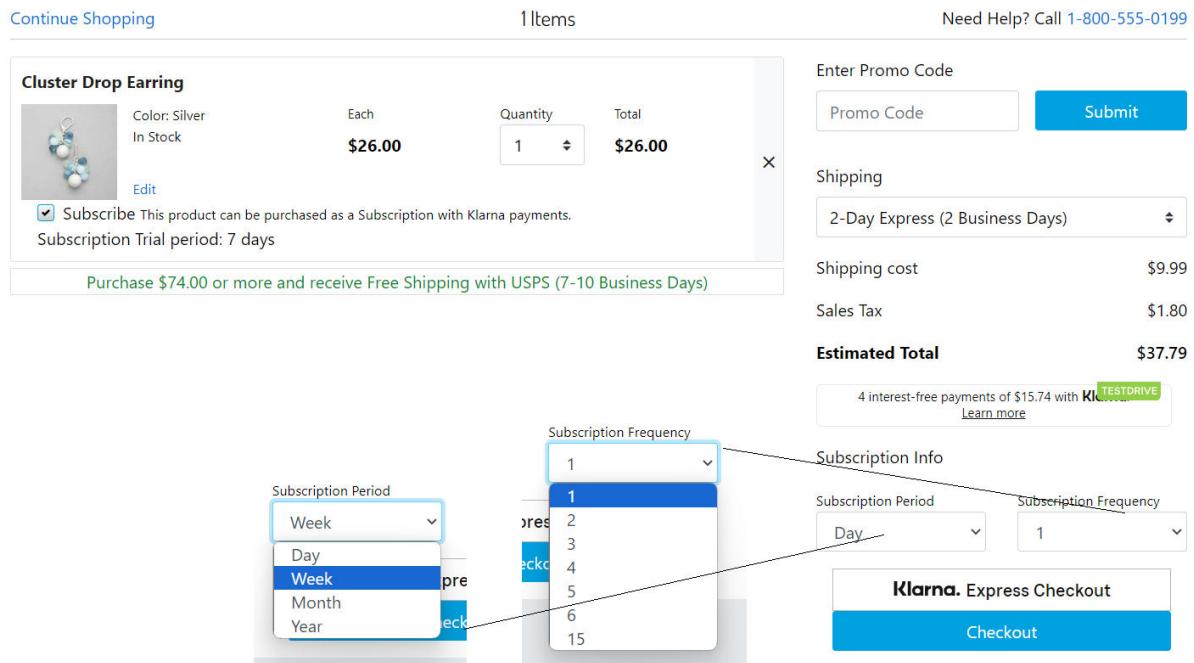


Figure 36 Subscriptions details in cart page

2.3.19.3. *Checkout*

Only Logged in user is allowed to do a subscription check out. Session intent is defined based on the basket content:

- “tokenize” – basket contains products with trial period, user is not charged on order creation
- “buy_and_tokenize” – subscription products without trial period
- “buy” – only standard products, no subscription products

For intents “tokenize” and “buy_and_tokenize” Klarna customer token for recurring payments is created. This is stored in customer profile for future usage.

Once the order is created the user profile is updated with subscription data:

- token
- enabled: Status of subscription
- nextChargeDate: A calculated date for the next subscription charge

- subscriptionPeriod: An enumerated value representing the subscription period ('week', 'month', or 'year').
- subscriptionFrequency: The frequency of subscription with will be numeric 1,2,3,4,5,6,15
- subscriptionProductID: The corresponding subscription product ID.
- lastOrderID: The ID of the last order for the subscription.

2.3.19.4. Account Subscription dashboard

There is a full list of subscriptions per user in my account section. The user will be able to cancel subscriptions. Cancelled subscriptions are displayed with **Inactive** status. Cancel subscription button will deactivate the customer token and no further charges will be made with it.

[Home](#) / [My Account](#)

The screenshot shows a list of subscriptions under the heading "Subscriptions". There are two items listed:

- Subscription ID: 69c0aeb2-e34e-4d11-847a-6a217dbce435** (Inactive)
- Subscription ID: 9404e832-2c91-4e8b-8535-6e26c881d9bf** (Active)

Each item includes a small image of a beaded keychain, the subscription ID, a "Cancel" button, and detailed information:

- Subscription ID: 69c0aeb2-e34e-4d11-847a-6a217dbce435 (Inactive)**
- Last Order ID: 00050402
Next Charge Date: 10/5/23
Status: **Inactive**
Total subscription amount: \$38.15
- Subscription ID: 9404e832-2c91-4e8b-8535-6e26c881d9bf (Active)**
- Last Order ID: 00050403
Next Charge Date: 10/5/23
Status: **Active**
Total subscription amount: \$38.15

Figure 37 My Account Subscription dashboard

2.3.19.5. Recurring Subscription Order Creation

A back-end job to process recurring subscription against each user in SFCC if subscription exists. The job iterates through customers and checks for subscriptions due for payment on the same day.

- Create a new order (with data from lastOrderID) and make a charge call to Klarna using the token and price from the lastOrderID.
- On success - update nextChargeDatebased on subscriptionPeriod and set lastOrderID to the ID of the newly placed order. Update the same in the Subscription Dashboard against the subscription.
- On failure:
 - o Retry

Merchants can configure a retry mechanism where they can set a retry after specific number of days. If the retry fails, then the subscription can be Deactivated. This is generic configuration for retry:

- Retry: Boolean field with value Yes/No
- Number of Retry: 1,2 If Retry field is set as Yes, then merchant can configure this field.
- Retry Frequency: 1, 2 If Retry field is set as Yes, then merchant can configure this field.

- o Cancel subscription – if retry is disabled the subscription is cancelled.

Orders with trial period subscriptions are paid after the trial period is over. On the next charge date, a new order will be created.

New orders are with channel type = SUBSCRIPTIONS.

2.3.20. Klarna Express Checkout

Klarna Express Checkout is a new feature introduced in Storefront where Express Checkout Button will be displayed in PDP, Cart and Mini Cart and user will be redirected to Klarna after clicking Express Checkout button. Klarna Express Checkout enabled quick and easy checkout where Shipping Address, Billing Address and Payment details will be preselected so that Checkout can be completed in fewer clicks. Klarna Documentation - <https://docs.klarna.com/express-checkout/>. Multi step checkout is implemented with finalize call at place order stage.

2.3.20.1. PDP

- Button is not visible if the product is subscription only.

- Product is added to cart and the checkout is started with new basket only for this product. Once the checkout is completed the old basket is restored based on a session attribute.

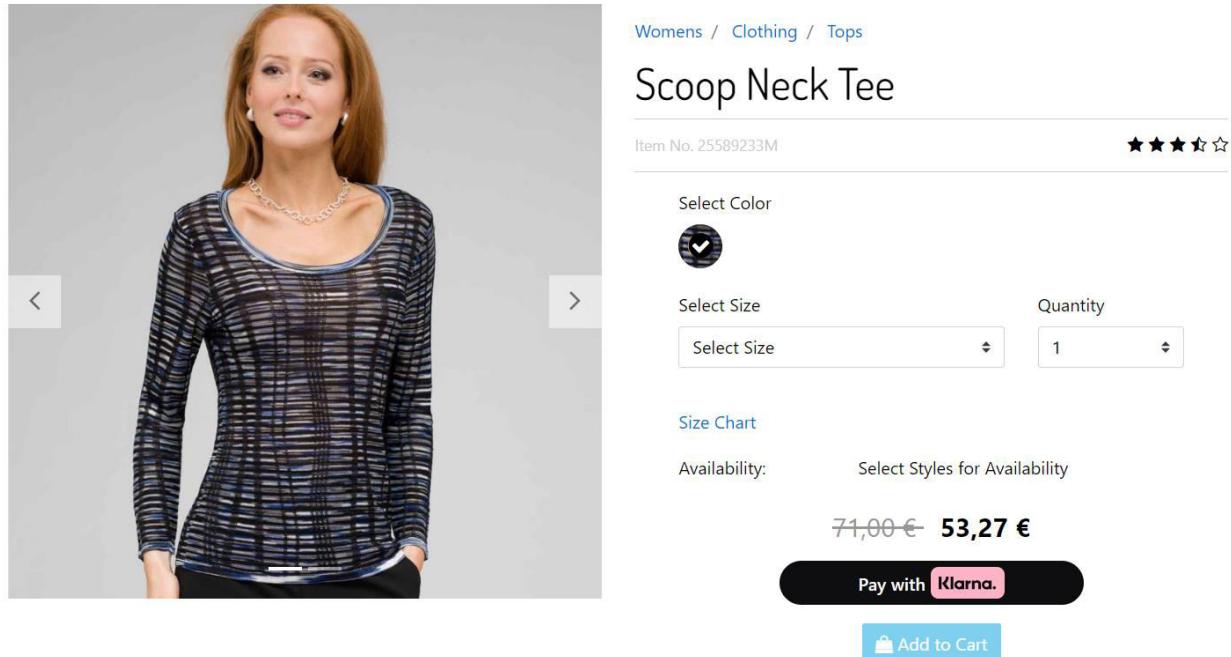


Figure 38 Klarna Express Checkout on PDP

2.3.20.2. Minicart and Cart buttons

- Express checkout button is not displayed in minicart if there are only subscription only products in the cart.

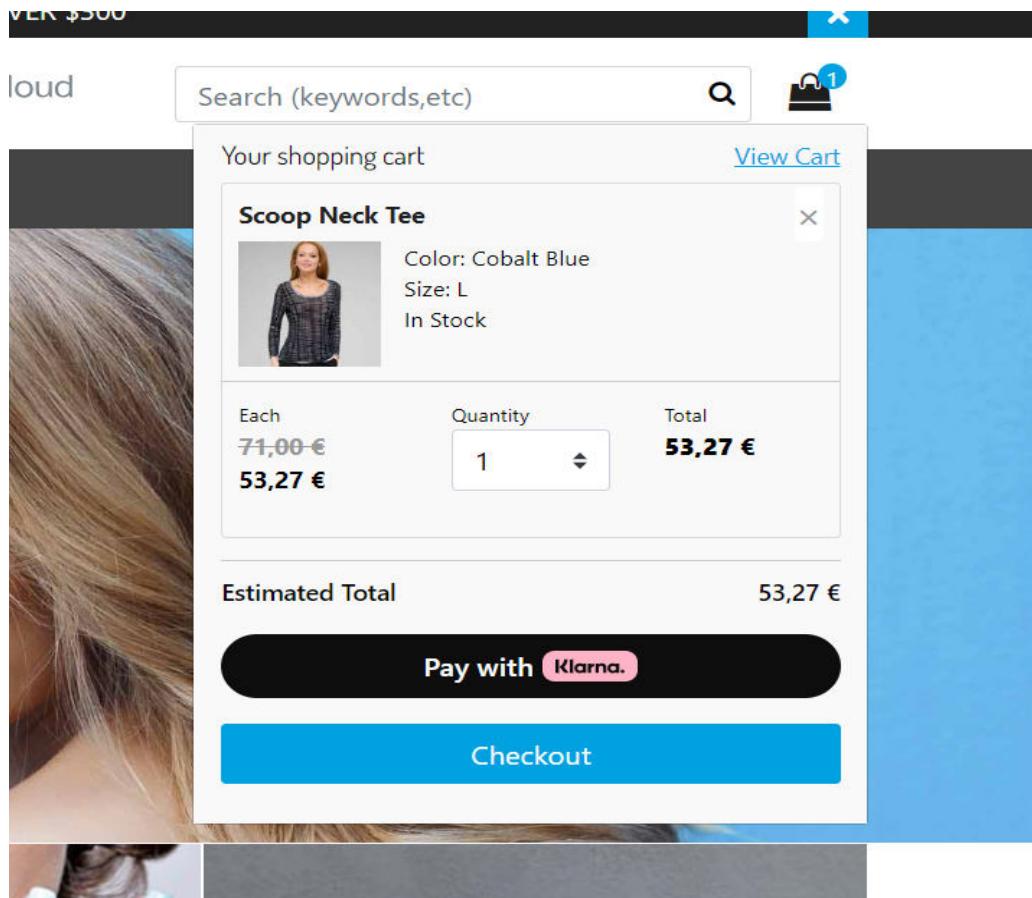


Figure 39 Klarna Express Checkout on minicart

This screenshot shows the Klarna Express Checkout integrated into a standard shopping cart page. At the top, there are links for 'Continue Shopping', '1 Items', and 'Need Help? Call 1-800-555-0199'. The cart item is a 'Scoop Neck Tee' in Cobalt Blue, Size L, with a quantity of 1 and a total of '53,27 €'. To the right, there's a section for entering a promo code with a 'Submit' button. Below that is a 'Shipping' section showing 'Ground (7-10 Business Days)' as the selected option. Shipping costs are listed as '5,99 €' and 'Sales Tax' as '2,83 €'. The 'Estimated Total' is '59,26 €'. At the bottom, there are the same 'Pay with Klarna.' and 'Checkout' buttons as in Figure 39.

Figure 40 Klarna Express Checkout on cart page

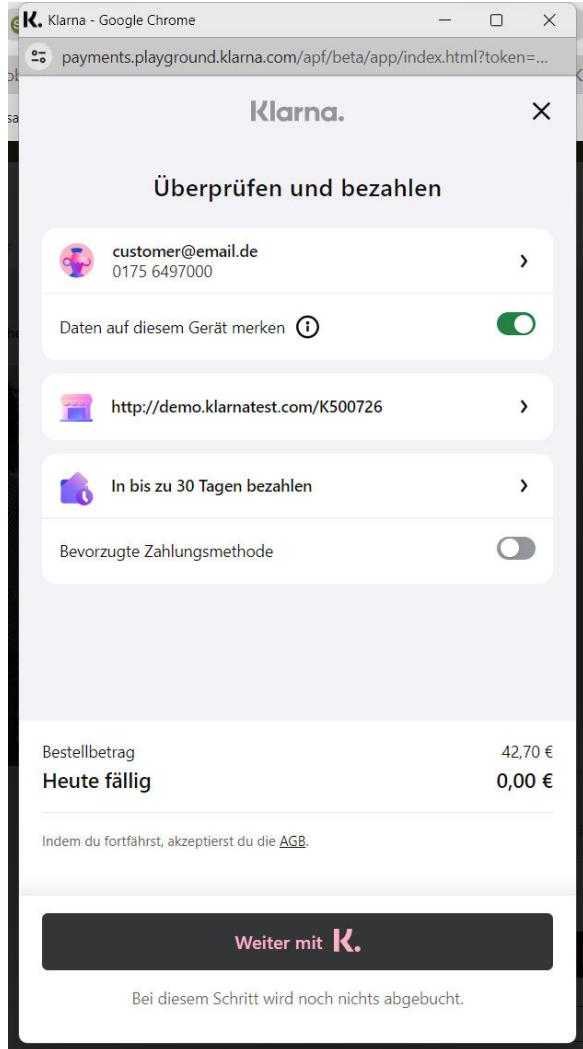


Figure 41 Klarna Express Checkout widget

2.3.20.3. Configuration (Deprecated)

Note: This configuration has been deprecated with release 24.4.0 and will be removed in the future.

- Basket has custom attribute “kpIsExpressCheckout” set as true for Express Checkout process.
- Merchant Tools > Site Preferences > Custom Site Preference Groups > Klarna Express Checkout

In Site Preferences we can enable/disable the feature and configure the buttons style. By default we have “Klarna” as payment method category which is displayed in checkout payment methods.

Name	Value	Default Value	
Klarna Express Checkout Enabled (kpECEnabled)	Yes	No	Edit Across Sites
Klarna Express Checkout Library URL (kpECLibraryURL) (String)	https://x.klamacdn.net/kp/lib/v1/api.js	https://x.klamacdn.net/kp/lib/v1/api.js	Edit Across Sites
Klarna Express Checkout Collect Shipping Address (kpECCollectShippingAddress)	Yes	No	Edit Across Sites
Klarna Express Checkout Button Theme (kpECButtonTheme)	Default (default)	Default	Edit Across Sites
Klarna Express Checkout Button Shape (kpECButtonShape)	Default (default)	Default	Edit Across Sites
Payment Category Content (kpECPaymentCategoryContent)	[{ "asset_urls": { "descriptive": "https://x.klamacdn.net/payment-method/assets/badges/generic/klarna.svg", "standard": "https://x.klamacdn.net/payment-method/assets/badges/generic/klarna.svg" }, "identifier": "klarna", "name": "klarna.payment.method" }]		Edit Across Sites
Placement (kec_placement)	None Cart (cart) PDF (pdf) Minicart (minicart)		Edit Across Sites

Figure 42 Klarna Express Checkout configuration

- Merchant Tools > Custom Objects > Manage Custom Objects > KlarnaCountries

Client key configuration by region in “**Klarna Express Checkout Client Key:**”

Minicart Express Button Shape:	- None -
Express Checkout	
Klarna Express Checkout Client Key:	klarna_test_client_VIU3KUNJMW53JDBVRUxucG1lZFlxdmNsc
<< Back to List	

Figure 43 KlarnaCountries Express Checkout Client Key

2.3.20.4. Configuration

- Basket has custom attribute “kpIsExpressCheckout” set as true for Express Checkout process.
- Merchant Tools > Site Preferences > Custom Site Preference Groups > Klarna Express Checkout

The screenshot shows the 'Klarna Express Checkout' configuration page within the Site Preferences section of Merchant Tools. At the top, there are buttons for 'Cancel', 'Apply to Other Sites', and 'Save'. Below this, the 'Instance Type' is set to 'Sandbox'. A search bar labeled 'Search by ID' is followed by a magnifying glass icon and a dropdown arrow. On the right, a progress indicator shows '14 of 4' items. The main area contains four configuration items:

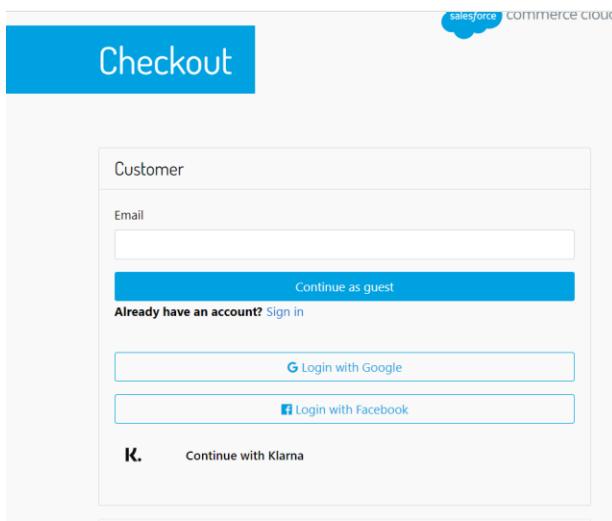
Name	Value	Default Value	Action
Enable Express Checkout (kec_enable)	None	Yes	Edit Across Sites
Theme (kec_theme)	None	Dark	Edit Across Sites
Button Shape (kec_shape)	None	Rounded corners	Edit Across Sites
Placement (kec_placement)	Cart (cart) PDP (pdः) Mini cart (minicart)		Edit Across Sites

Figure 44 Klarna Express Checkout configuration

2.3.21. Klarna SIWK (Sign in with Klarna)

Klarna SIWK is new feature introduced in where sign in with Klarna button will be displayed in login and checkout pages. User will get signed in through Klarna after clicking the siwk button. This will help users to easily checkout without entering the address as the user details will already be saved in the Klarna account.

2.3.21.1. Checkout page



After clicking on 'Continue with Klarna' button, a popup will appear asking for country and contact information. User will get logged in and address fields will be populated so that user directly place order.

2.3.21.2. Login Page

The screenshot shows a standard login form with fields for Email and Password, a Remember me checkbox, and links for forgot password? and Continue with Klarna. To the right, there is a 'Check order' section with fields for Order number, Order Email, and Billing ZIP code, along with a Check status button.

After clicking on Continue with Klarna button in login page, user will be redirected to account page .

2.3.21.3. Configuration

Klarna SIWK feature can be enabled in Site preferences under Klarna_swik group preferences.

The screenshot shows the 'Klarna Sign-in With Klarna' configuration page. It includes fields for Instance Type (Sandbox), a search bar, and a table for enabling sign-in with Klarna. The table has columns for Name, Value, and Default Value. One row shows 'Enable Sign-in with Klarna*' with Value 'Yes' and Default Value 'Yes'. Another row shows 'Scope*' with a dropdown menu containing options like 'profile.name', 'profile.phone', 'profile.date_of_birth', etc. A note below the enable field says: 'Reduce fraud rates through verified identities and consented shopper data. Improve personalized shopping experiences.' The bottom of the page includes a copyright notice and a RefArchGlobal Time Zone: Coordinated Universal Time/Instance Time Zone: Eastern Daylight Time (Version: 24.6 Li).

Client id is configured in Merchant Tools > Custom Object > Custom Object Editor.

The screenshot shows the 'Express Button' configuration in the Custom Object Editor. It includes fields for Express Button Merchant ID, Environment (playground), Theme (light), Shape (None), and Cart Enabled (checked). Below this is the 'MiniCart Express Button' section with Enabled checked, Theme set to default (Default), and Shape set to None. At the bottom is the 'Express Checkout' section, which includes a Klarna Client Identifier field containing 'klarna_test_client_VIU3KUNjMW53JDBVRUxucG1IZFixdmN'. A green arrow points to the Klarna Client Identifier field. The bottom left of the page has a 'Sign In Client ID' field with 'klarna_test_client_cUE3SEiLaVRLWGlJSz9od3dMZWxEcUY' and a 'Sign In Client ID' field with 'klarna_test_client_cUE3SEiLaVRLWGlJSz9od3dMZWxEcUY'.

Button styles can be changed in siwk_alignment, siwk_shape and siwk_theme in Site preferences. Button placement can be configured in login or checkout page as per site preference.

Note : Existing users will always be merged with Klarna account once signed in.

2.4. Compatibility

This cartridge has been tested against API Version: 22.6 (Compatibility Mode: 21.10) & SFRA version 6.0.0

2.5. Privacy, Payment

2.5.1. GDPR Compliance

The cartridge is compliant with GDPR recommendation and follows the best practice mentioned here and implementation transmits only required (PII) data to authorize payment method.

2.5.2. EMD (Extra Merchant Data)

The cartridge supports sending additional information on the customer's past purchase history, as well as "Buy Online, Pickup in Store" (BOPIS) store addresses when turned on in custom preferences: "Attachments" (**kpAttachments (Deprecated with v24.4.0) kpEMD**). The type of data that can be send as an attachment is mentioned here. EMD is required for certain types of merchant orders and the inclusion of EMD is (e.g.customer_account_info: past interaction with merchant store) generally beneficial to improve acceptance rates.

EMD is included as part of Authorization step in Commerce Cloud checkout. The data send to Klarna is customizable and can be seen in "**int_klarna_payments/scripts/payments/additionalCustomerInfo.js**". This script should return a JSON string to be used as a value for the body sub-field of the attachment field as [described here](#) .

If the example additionalCustomerInfo.js file is used unchanged the data send to Klarna is by the following schema:

```
{  
  "$schema": "http://json-schema.org/draft-03/schema#",  
  "id": "http://klarna.com/v2/emd#",  
  "description": "Extended Merchant Data Payload Schema",  
  "type": "object",  
  "properties": {  
    "customer_account_info": {  
      "type": "array",  
      "items": {  
        "type": "object",  
        "properties": {  
          "key": "string",  
          "value": "string"  
        }  
      }  
    }  
  }  
}
```

```

    "items": {
        "type": "object",
        "properties": {
            "unique_account_identifier": {
                "type": "string",
                "maxLength": 24
            },
            "account_registration_date": {
                "description": "ISO 8601 e.g. 2012-11-24T15:00",
                "type": "string",
                "format": "date-time",
                "pattern": "^[0-9][0-9][0-9][0-9]-[0-1][0-9]-[0-3][0-9]T[0-2][0-9]:[0-5][0-9](:[0-5][0-9])\\{0,1\\}Z\\{0,1\\}\\$"
            },
            "account_last_modified": {
                "description": "ISO 8601 e.g. 2012-11-24T15:00",
                "type": "string",
                "format": "date-time",
                "pattern": "^[0-9][0-9][0-9][0-9]-[0-1][0-9]-[0-3][0-9]T[0-2][0-9]:[0-5][0-9](:[0-5][0-9])\\{0,1\\}Z\\{0,1\\}\\$"
            }
        }
    },
    "payment_history_full": {
        "type": "array",
        "items": {
            "type": "object",
            "additionalProperties": false,
            "properties": {

```

```

    "unique_account_identifier": {
        "type": "string"
    },
    "payment_option": {
        "type": "string",
        "enum": ["card", "direct banking", "non klarna credit", "sms", "other"]
    },
    "number_paid_purchases": {
        "type": "integer"
    },
    "total_amount_paid_purchases": {
        "type": "number"
    },
    "date_of_last_paid_purchase": {
        "description": "ISO 8601 e.g. 2012-11-24T15:00",
        "type": "string",
        "format": "date-time",
        "pattern": "^[0-9][0-9][0-9][0-9]-[0-1][0-9]-[0-3][0-9]T[0-2][0-9]:[0-5][0-9](:[0-5][0-9])\\{0,1\\}Z\\{0,1\\}$"
    },
    "date_of_first_paid_purchase": {
        "description": "ISO 8601 e.g. 2012-11-24T15:00",
        "type": "string",
        "format": "date-time",
        "pattern": "^[0-9][0-9][0-9][0-9]-[0-1][0-9]-[0-3][0-9]T[0-2][0-9]:[0-5][0-9](:[0-5][0-9])\\{0,1\\}Z\\{0,1\\}$"
    }
}
}
}
},

```

```

"other_delivery_address": {
    "type": "array",
    "items": {
        "type": "object",
        "additionalProperties": false,
        "properties": {
            "shipping_method": {
                "type": "string",
                "enum": ["store pick-up", "pick-up point", "registered box", "unregistered box"]
            },
            "shipping_type": {
                "type": "string",
                "enum": ["normal", "express"]
            },
            "first_name": {
                "type": "string"
            },
            "last_name": {
                "type": "string"
            },
            "street_address": {
                "type": "string"
            },
            "street_number": {
                "type": "string"
            },
            "postal_code": {
                "type": "string"
            }
        }
    }
}

```

```

        },
        "city": {
            "type": "string"
        },
        "country": {
            "type": "string"
        }
    }
}
}
}
}
```

Example data:

```
{
    "attachment": {
        "content_type": "application/vnd.klarna.internal.emd-v2+json",
        "body": {
            "customer_account_info": [
                {
                    "unique_account_identifier": "5509d9f7c8720c0e4575154b",
                    "account_registration_date": "2015-03-18T20:03:03Z",
                    "account_last_modified": "2015-03-18T20:03:03Z"
                }
            ],
            "payment_history_full": [
                {
                    "unique_account_identifier": "5509d9f7c8720c0e4575154b",
                    "payment_option": "card",
                    "number_paid_purchases": "23",
                    "total_amount_paid_purchases": "140023",
                    "date_of_last_paid_purchase": "2015-03-18T20:03:03Z",

```

```

        "date_of_first_paid_purchase": "2015-03-18T20:03:03Z"
    }],
    "other_delivery_address": [
        "shipping_method": "store pick-up",
        "shipping_type": "normal",
        "first_name": "Test",
        "last_name": "Customer",
        "street_address": "1487 Bay St",
        "street_number": "",
        "postal_code": "01109",
        "city": "Springfield",
        "country": "US"
    ],
    }
}
}
```

Please, note that in cases when the customer uses Guest Checkout, the EMD sent includes **payment_history_full[0].unique_account_identifier** (cqid value set by SFCC), and all other fields are empty!!

2.5.3. PCI-DSS Compliance

Important Note: DO NOT SAVE UNENCRYPTED PCI DATA ON THE SERVER!

The virtual card (MCSv3) solution enables settlements using individual virtual card issued against a Klarna order. To be compliant with PCI-DSS requirements, merchant must ensure the data is securely maintained and transmitted as part of their operation in their live store environment. The required steps to ensure this, must be done in consultation with your payment service provider/acquirer and completed prior to go-live. 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 VCN validity date.

3. Implementation Guide

3.1. Setup of Business Manager

The Klarna Payments LINK Cartridge contains 2 cartridges that are required for full functionality. Additionally, Controller and SFRA support is broken out into two separate cartridges, thereby facilitating the installation and use of one or the other models.

`int_klarna_payments` – Implements the core storefront functionality.

`int_klarna_payments_sfra` - Implements the storefront functionality with SFRA code.

3.1.1. Cartridge Upload & Assignment

Import the two cartridges into UX studio and associate them with a Server Connection.

- Import the “`int_klarna_payments`” cartridge into the SCC Studio Workspace:
 - Open SCC Studio
 - Click File -> Import -> General -> Existing Projects into Workspace
 - Browse to the directory where you saved the “`int_klarna_payments`” cartridge.
 - Click Finish.
 - Click OK when prompted to link the cartridge to the sandbox.
- Import the “`int_klarna_payments_sfra`” cartridge into the SCC Studio Workspace:
 - Open SCC Studio
 - Click File -> Import -> General -> Existing Projects into Workspace
 - Browse to the directory where you saved the “`int_klarna_payments_sfra`” cartridge.
 - Click Finish.
 - Click OK when prompted to link the cartridge to the sandbox.
- Prepend the Klarna cartridges to the effective site cartridge path:
 - Log into the SCC Business Manager.
 - Click Administration -> Sites -> Manage Sites.
- Select the desired site.

- Click on the Settings tab.
- Prepend “**int_klarna_payments_sfra:int_klarna_payments**” to the “**Cartridges**” field.
- Click Apply

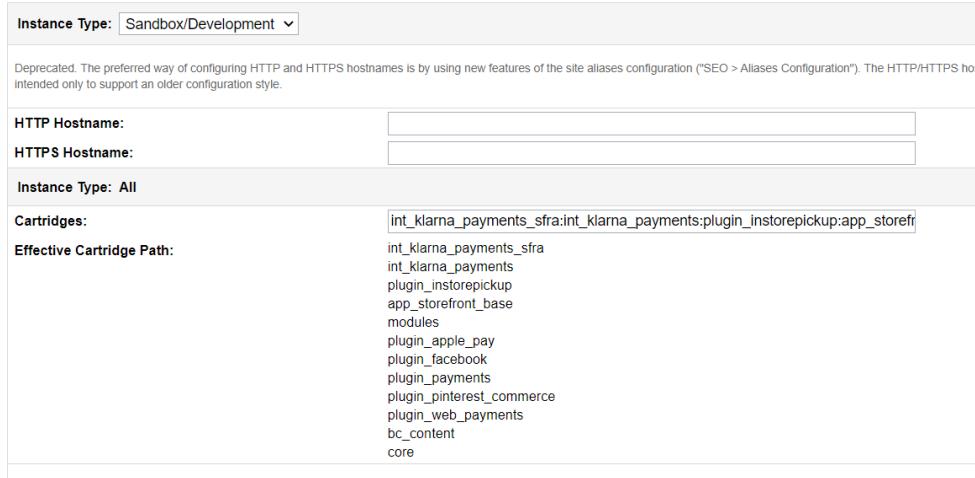


Figure 45 Effective Cartridge Path

3.1.2. *Metadata Import*

- Go to main directory “**metadata**” folder, review the site-template content, and edit if needed. (Site template is prepared to setup “**SiteGenesis**” and “**RefArch**” sites - you may want to change that to your actual sites and delete the ones that are not needed). Zip the directory and you will have “**site-template.zip**” installation package.
- Log into the SCC Business Manager.
- Click Administration -> Sites Development -> Site Import & Export
- Browse to the directory where you saved the “**site-template.zip**”.
- Click “**Upload**”
- Select the uploaded site zip and click “**Import**”.

Note: Please, review the default **service.xml** file in the **site-template.zip** and update the configuration for Playground and Production accordingly before importing.

3.1.3. *Build Klarna JS & CSS*

If necessary, update the path to your base SFRA installation in **package.json** file from the same root folder.

Normally you would have a top-level project folder, into which the repositories of SFRA base cartridge and all required plugins, libraries and any other LINK cartridges will be cloned. In case you have cloned the Klarna cartridge into that folder as well, the below change will not be required. Otherwise, update **paths.base** property in the **package.json** to contain a relative path to the local directory containing the Storefront Reference Architecture repository. Its default value will be as follows:

```
"paths": {  
    "base": "../storefront-reference-architecture/cartridges/app_storefront_base/"  
}
```

Once you are certain the correct path to SFRA cartridge is configured, run “**npm run compile:scss**” command from the root folder of Klarna repository.

Run “**npm run compile:js**” command from the root folder of Klarna repository to build the Klarna specific JavaScript files.

3.2. Configuration

- Add your account settings to the KlarnaCountries Custom Objects.
 - Log into the SCC Business Manager.
 - Select the desired site from the tabs across the top of the page.
 - Click Custom Objects -> Custom Object Editor
 - Change the Object Type dropdown to “**KlarnaCountries**”.
 - Click the “**Find**” button.
 - Click the desired country you wish to edit (See screenshot below).
 - Update the required fields as mentioned in “KlarnaCountries” section
 - Repeat for the other countries.

Merchant Tools > Custom Objects > Custom Objects > US - General

General

Manage 'US' (KlarnaCountries)

Fields with a red asterisk (*) are mandatory. You can view and edit the name and description in other languages, if required. Click **Apply** to save the details.

custom	Country Code: * <input type="text" value="US"/> On-site Messaging Data Default Locale: <input type="text" value="en-US"/> Service Credential ID: <input type="text" value="klarna.http.uscredentials"/> On-site messaging Data Client ID: <input type="text" value="60a22a39-c2fd-5d09-bfe1-771459318a4d"/> <input checked="" type="checkbox"/> Cart Placement Tag Enabled: <input type="text" value="info-page-standard"/> <input checked="" type="checkbox"/> PDP Placement Tag Enabled: <input type="text" value="credit-promotion-small"/> <input checked="" type="checkbox"/> Header Placement Tag Enabled: <input type="text" value="top-strip-promotion-standard"/> <input checked="" type="checkbox"/> Footer Placement Tag Enabled: <input type="text" value="footer-promotion-auto-size"/> <input checked="" type="checkbox"/> Info Page Placement Tag Enabled: <input type="text" value="info-page"/> Library URL: <input type="text" value="https://na-library.playground.klarnaservices.com/lib.js"/>
--------	--

Figure 46 KlarnaCountries Settings

- Configure Klarna Payment Custom Preferences using the SCC Business Manager
 - Log into the SCC Business Manager
 - Select the desired site from the tabs across the top of the page.
 - Click Site Preferences -> Custom Preferences -> KlarnaPayment
 - Fill out the settings as desired. Descriptions of the site preferences are in the Site Preferences section.
- Configure Klarna Payment Service using the SCC Business Manager
 - Log into the SCC Business Manager
 - Click Administration > Operations > Services.
 - Click the Credentials tab.
 - Each Klarna credential correspond to one of the KlarnaCountries custom objects. Click on the one you want to edit.
 - Enter the MID API username and API password you received from Klarna.

- Edit URL field if Production environment. Klarna API URLs information - <https://developers.klarna.com/api/#api-urls>.

[Administration](#) > [Operations](#) > [Services](#) > [Service Credentials](#) > klarna.http.gbcredentials - Details

klarna.http.gbcredentials

Fields with a red asterisk (*) are mandatory. Click **Apply** to save the details. Click **Reset** to revert to the last saved state.

These credentials are used by 0 services.

Name:*	klarna.http.gbcredentials
URL:	https://api.playground.klarna.com/
User:	your Merchant ID
Password:	*****

Figure 47 Service Settings

- Configure Klarna Rate Limited Service Profile using the SCC Business Manager (Optional)
 - Log into the SCC Business Manager
 - Navigation : Merchant Tools > Site preferences > Custom Site Preferences group > Klarna Payments
 - Enable Service Limit configuration > Select “Rate Limit By Operation” to Yes. If it is selected to NO, the default service will control the rate limit.
- Configure Custom Rate limits in SCC Business Manager (Optional)
 - Log into the SCC Business Manager
 - Navigation: Administration > Operations > Services
 - Select a required profile (e.g : Klarna.http.createSession)
 - Enable Rate Limit: Check box
 - Max Rate Limit Calls: 50 (e.g.: Higher rate limit of 50 requests/sec agreed with Klarna)
 - Rate Limit Internal (millisecond): 1000
 - Repeat for all Service operation with respective agreed rate limits

The screenshot shows the Salesforce Admin UI with the following navigation path: Administration > Operations > Services > Service Profiles > klarna.http.createSession - Details.

klarna.http.createSession

Fields with a red asterisk (*) are mandatory. Click **Apply** to save the details. Click **Reset** to revert to the last saved state.

This profile is used by 1 service.

Name: [*]	klarna.http.createSession
Connection Timeout (ms):	30,000
Enable Circuit Breaker:	<input type="checkbox"/>
Max Circuit Breaker Calls:	0
Circuit Breaker Interval (ms):	0
Enable Rate Limit:	<input checked="" type="checkbox"/>
Max Rate Limit Calls:	50
Rate Limit Interval (ms):	1000

<< Back to List

3.3. Extended Controllers

Controller	Start Node	Remarks
Checkout.js	Begin	Extended to call Klarna session manager
CheckoutServices.js	Get, SubmitPayment, PlaceOrder	Klarna payment method/category and totals are being stored
CheckoutShippingServices.js	SubmitShipping, ToggleMultiShipping	Calling the Klarna session manager
Order.js	Confirm	Extending Klarna order data to view data

Table 1 Extended Controllers List

3.4. Template Updates

Templates have been updated to support On-site messaging and Addresses forms for Klarna. To be used as reference but feel free to customize the templates to match your specific needs. Final review and sign-off as per project requirements and contract agreements.

3.5. Jobs

3.5.1. Job “*OrderCleanUp*” (Optional)

This 1-time clean-up job is only applicable to merchants integrated with Klarna Payments cartridge version (< 19.1.6), utilizing (or previously used) virtual card-based settlement (VCN) and stored decrypted card details within Business Manager.

The job iterates over orders with status “**Exported**” and attribute “**custom.kpIsVCN=true**” to remove the sensitive details saved in fields kpVCNPAN, kpVCNCSC, kpVCNExpirationMonth, kpVCNExpirationYear as part of the previous releases. There are no parameters passed to the script.

Upon successful run, the job will log the result of processed orders in the custom debug log located in “**webdav/Sites/Logs**”. Depending on the setup, you will receive a message for the processed orders count for each storefront or message that there are no orders needing update.

```
Wed, 09 Sep 2020 09:45:38 GMT DEBUG CustomJobThread[1740004063]OrderCleanUp|executeRefArch custom [] Job [OrderCleanUp] - [RefArch]: No orders require processing
Wed, 09 Sep 2020 09:45:38 GMT DEBUG CustomJobThread[1740004063]OrderCleanUp|executeRefArchGlobal custom [] Job [OrderCleanUp] - [RefArchGlobal]: No orders require processing
Wed, 09 Sep 2020 09:45:38 GMT DEBUG CustomJobThread[1740004063]OrderCleanUp|executeSiteGen custom [] Job [OrderCleanUp] - [SiteGenesis]: Orders processed: 2
Wed, 09 Sep 2020 09:45:38 GMT DEBUG CustomJobThread[1740004063]OrderCleanUp|executeSiteGenGlobal custom [] Job [OrderCleanUp] - [SiteGenesisGlobal]: No orders require processing
```

Figure 48 Job Logs

Upon error, the cause of the failure (message and stack) will be logged in the standard error log.

```
Wed, 09 Sep 2020 09:05:22 GMT ERROR CustomJobThread[6100040111]OrderCleanUp|execute com.demandware.beehive.core.internal.domain.SystemObjectQueryMgrImpl Sites-RefArch-Site 208 43e0d91e96 e67f0cb0ade56d99ae0c4bc12b 16486241b1284573184 - Exception while parsing system object query.
System
RequestID: e67f0cb0ade56d99ae0c4bc12b
SessionType: JOB
Truncated SessionID: 43e0d91e96
User Profile UUID: 03e37ff77529fda91ae75408ab
```

Figure 49 Job Logs

To setup the job, go to **Administration > Operations > Import & Export** and import file “**jobs.xml**”. Out of the box, the XML file includes only the RefArch scope, but it can be configured with multiple flows if you have more than one site using this functionality as seen bellow. Each site should be added as a separate flow.

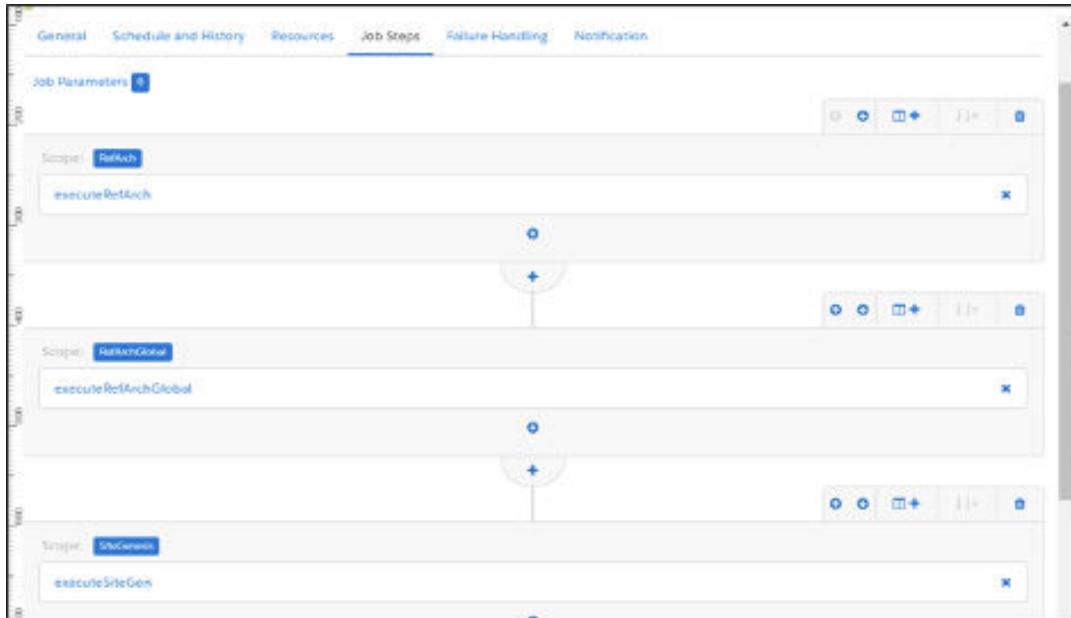


Figure 50 Job Steps

To set up the required job parameters and add new flow, follow the bellow steps. If you only have one storefront and need to change the scope to the correct one – proceed to steps 4-5 directly.

1. Click on the “Add a sequential flow” button at the bottom of the current flow.



Figure 51 Add New Job Step

- Once done, click on “Configure a step” button within the flyout that has just opened search for “script” and select “ExecuteScriptModule”.

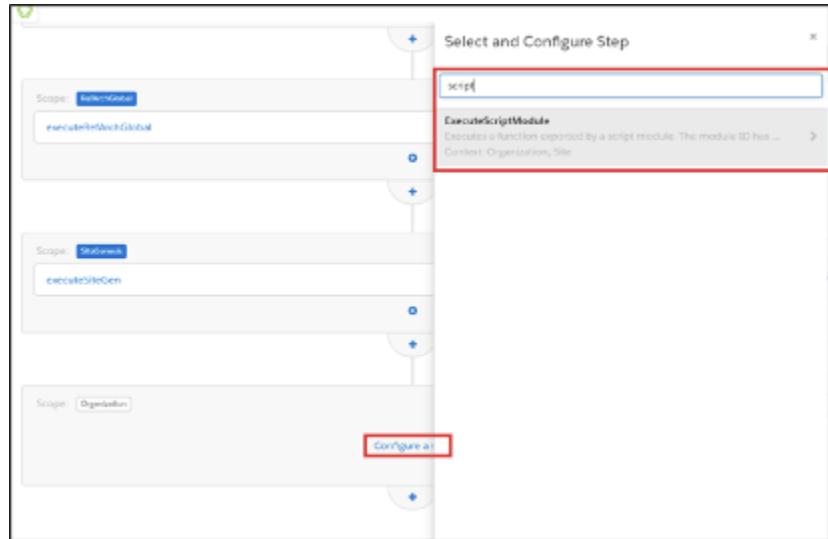


Figure 52 Configure Step

- In the flyout populate these fields and click the “Assign” button.
 - ID – Enter any meaningful name in the field. In case you have multiple flows, this should not be a duplicate one. If you enter a duplicate name, the details won’t be saved and you will see an error message.
 - ExecuteScriptModule.Module – Enter the location of the “OrderCleanUpJob.js” file. Out of the box it should be

“int _klarna_payments/cartridge/scripts/job/OrderCleanUpJob.js” or the location where you have placed the script file.

- c. ExecuteScriptModule.FunctionName – Leave the field value to “execute”

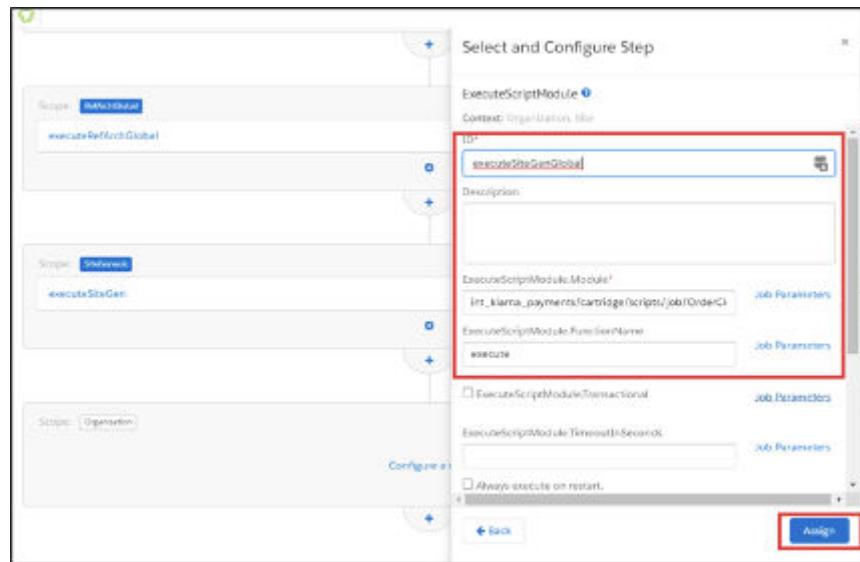


Figure 53 Configure Step (cont.)

4. Once the step has been added, you should make sure to assign it to the correct site scope. Click on “Organization” and select “Specific Sites” from the drop-down.

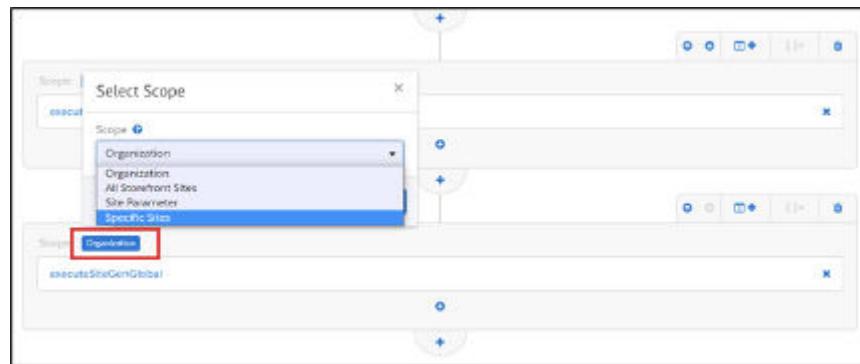


Figure 54 Job Scope

5. From the list of sites, select the respective site ID (i.e. SiteGenesisGlobal) and click on “Assign”.

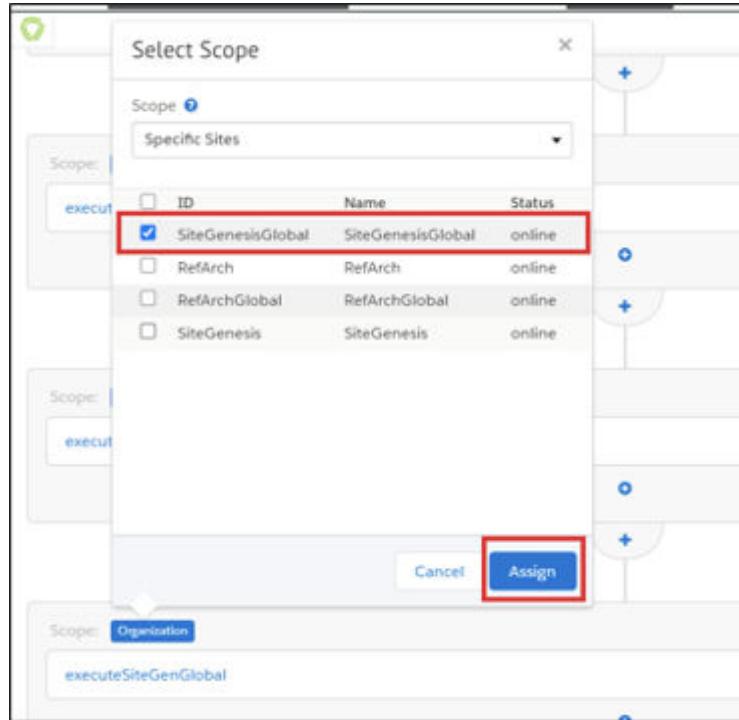


Figure 55 Job Scope (cont.)

6. Repeat steps 1-5 for each site/storefront that you have using Klarna VCN and need additional configuration.

3.5.2. Job “RecurringOrders”

The job iterates over all customers checking for subscriptions entries. Process all subscription entries eligible for charge – subscription should be enabled and nextChargeDate or nextRetryDate should match the current date. New SFCC orders are created and the old ones are replaced. Orders with expiring trial period, are only charged.

By default in jobs.xml the job is configured to run on RefArch site but this can be changed either in file or in storefront.

The job has one job step – createOrder with following configuration:

- ExecuteScriptModule.Module -
int_klarna_payments/cartridge/scripts/job/RecurringOrdersJob.js
- ExecuteScriptModule.FunctionName – **execute**

The job is executed on site level.

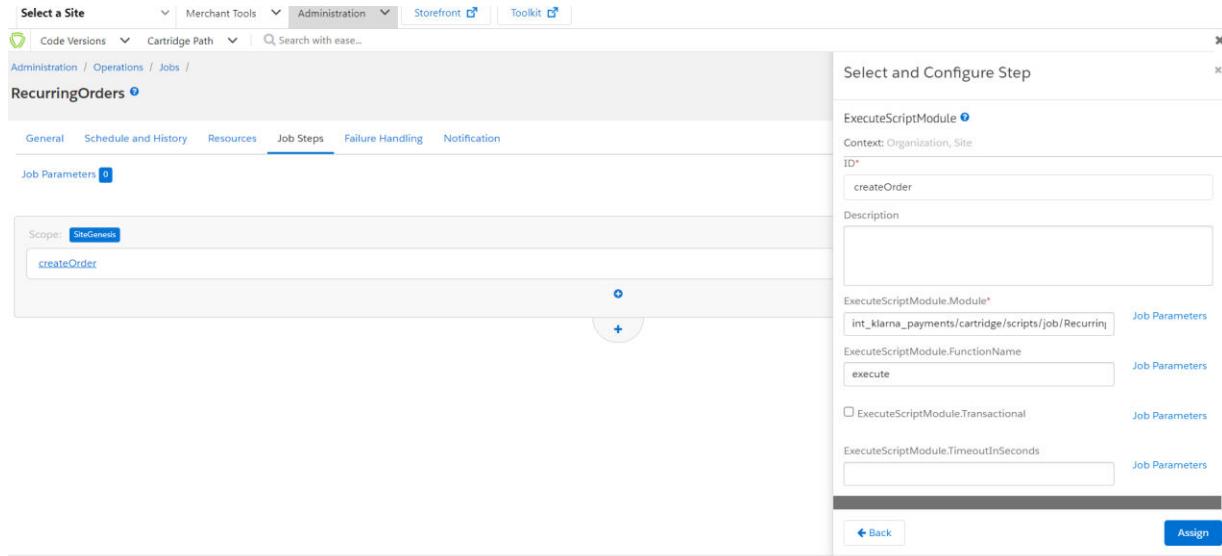


Figure 56 RecurringOrdersJobs

3.6. Custom Code

Store Front Reference Architecture (SFRA) does not imply modification of core cartridge to enable any of LINK integration cartridges. The "int_klarna_payments_sfra" cartridge follows best practices recommended by Salesforce. Refer to below 2 cases where changes may be applicable:

3.6.1. app_storefront_base\cartridge\client\default\js\checkout\checkout.js (optional)

After placing an order every customer is redirected to Klarna and then sent back to the site with order confirmation page. To prevent sending of any additional URL params to Klarna please do the following:

```

310 else if (stage === 'placeOrder') {
311   // disable the placeOrder button here
312   $('body').trigger('checkout:disableButton', '.next-step-button button');
313   $.ajax({
314     url: $('.place-order').data('action'),
315     method: 'POST',
316     success: function (data) {
317       // enable the placeOrder button here
318       $('body').trigger('checkout:enableButton', '.next-step-button button');
319       if (data.error) {
320         if (data.cartError) {
321           window.location.href = data.redirectUrl;
322           defer.reject();
323         } else {
324           // go to appropriate stage and display error message
325           defer.reject(data);
326         }
327       } else {
328         var continueUrl = data.continueUrl;
329         var urlParams = {
330           ID: data.orderID,
331           token: data.orderToken
332       };
333       continueUrl += (continueUrl.indexOf('?') !== -1 ? '&' : '?') +
334         Object.keys(urlParams).map(function (key) {
335           return key + '=' + encodeURIComponent(urlParams[key]);
336         }).join('&');
337     };
338   });
339 }

```

```

310 else if (stage === 'placeOrder') {
311   // disable the placeOrder button here
312   $('body').trigger('checkout:disableButton', '.next-step-button button');
313   $.ajax({
314     url: $('.place-order').data('action'),
315     method: 'POST',
316     success: function (data) {
317       // enable the placeOrder button here
318       $('body').trigger('checkout:enableButton', '.next-step-button button');
319       if (data.error) {
320         if (data.cartError) {
321           window.location.href = data.redirectUrl;
322           defer.reject();
323         } else {
324           // go to appropriate stage and display error message
325           defer.reject(data);
326         }
327       } else {
328         var continueUrl = data.continueUrl;
329         var urlParams = {};
330
331         if (data.orderID && data.orderToken) {
332           urlParams.ID = data.orderID;
333           urlParams.token = data.orderToken;
334         };
335       }
336       continueUrl += (continueUrl.indexOf('?') !== -1 ? '&' : '?') +
337         Object.keys(urlParams).map(function (key) {
338           return key + '=' + encodeURIComponent(urlParams[key]);
339         }).join('&');
340     };
341   });
342 }

```

Figure 57 Changes in checkout.js

3.6.2. *app_storefront_base\cartridge\client\default\js\cart\cart.js (required)*

In order get updates of the cart OSM widget please do the following in cart.js or in your app cartridge.

In function updateCartTotals(data) {} add the following code at the bottom:

```

if (data.totals.klarnaTotal) {
  $('#klarna-placement').attr('data-purchase-amount', data.totals.klarnaTotal);
  if (window.Klarna && window.Klarna.OnsiteMessaging) {
    window.Klarna.OnsiteMessaging.refresh();
  }
}

```

```

 63  function updateCartTotals(data) {
 64    $('.number-of-items').empty().append(data.resources.numberOfItems);
 65    $('.shipping-cost').empty().append(data.totals.totalShippingCost);
 66    $('.tax-total').empty().append(data.totals.totalTax);
 67    $('.grand-total').empty().append(data.totals.grandTotal);
 68    $('.sub-total').empty().append(data.totals.subTotal);
 69    $('.minicart-quantity').empty().append(data.numItems);
 70    $('.minicart-link').attr({
 71      'aria-label': data.resources.minicartCountOfItems,
 72      title: data.resources.minicartCountOfItems
 73    });
 74    if (data.totals.orderLevelDiscountTotal.value > 0) {
 75      $('.order-discount').removeClass('hide-order-discount');
 76      $('.order-discount-total').empty()
 77        .append('- ' + data.totals.orderLevelDiscountTotal.formatted);
 78    } else {
 79      $('.order-discount').addClass('hide-order-discount');
 80    }
 81    if (data.totals.shippingLevelDiscountTotal.value > 0) {
 82      $('.shipping-discount').removeClass('hide-shipping-discount');
 83      $('.shipping-discount-total').empty().append('- ' +
 84        data.totals.shippingLevelDiscountTotal.formatted);
 85    } else {
 86      $('.shipping-discount').addClass('hide-shipping-discount');
 87    }
 88    data.items.forEach(function (item) {
 89      if (item.renderedPromotions) {
 90        $('.item-' + item.UUID).empty().append(item.renderedPromotions);
 91      }
 92      if (item.priceTotal && item.priceTotal.renderedPrice) {
 93        $('.item-total-' + item.UUID).empty().append(item.priceTotal.renderedPrice);
 94      }
 95    });
 96  });
 97 }
 98 }

 99  /*
100   * Function updateCartTotals(data) {
101   *   $('.number-of-items').empty().append(data.resources.numberOfItems);
102   *   $('.shipping-cost').empty().append(data.totals.totalShippingCost);
103   *   $('.tax-total').empty().append(data.totals.totalTax);
104   *   $('.grand-total').empty().append(data.totals.grandTotal);
105   *   $('.sub-total').empty().append(data.totals.subTotal);
106   *   $('.minicart-quantity').empty().append(data.totals.numItems);
107   *   $('.minicart-link').attr{
108   *     'aria-label': data.resources.minicartCountOfItems,
109   *     title: data.resources.minicartCountOfItems
110   *   };
111   *   if (data.totals.orderLevelDiscountTotal.value > 0) {
112   *     $('.order-discount').removeClass('hide-order-discount');
113   *     $('.order-discount-total').empty()
114   *       .append('- ' + data.totals.orderLevelDiscountTotal.formatted);
115   *   } else {
116   *     $('.order-discount').addClass('hide-order-discount');
117   *   }
118   *   if (data.totals.shippingLevelDiscountTotal.value > 0) {
119   *     $('.shipping-discount').removeClass('hide-shipping-discount');
120   *     $('.shipping-discount-total').empty().append('- ' +
121   *       data.totals.shippingLevelDiscountTotal.formatted);
122   *   } else {
123   *     $('.shipping-discount').addClass('hide-shipping-discount');
124   *   }
125   *   data.items.forEach(function (item) {
126     if (item.renderedPromotions) {
127       $('.item-' + item.UUID).empty().append(item.renderedPromotions);
128     }
129     if (item.priceTotal && item.priceTotal.renderedPrice) {
130       $('.item-total-' + item.UUID).empty().append(item.priceTotal.renderedPrice);
131     }
132   });
133 }

134 if (data.totals.klarnaTotal) {
135   $('.klarna-placement').attr('data-purchase-amount', data.totals.klarnaTotal);
136   window.KlarnaOnsiteService = window.KlarnaOnsiteService || [];
137   window.KlarnaOnsiteService.push({ eventName: 'refresh-placements' });
138 }

```

Figure 58 Changes in cart.js

3.7. Integration with Other Payment Cartridges

The contents of this section are only applicable if there are other payment cartridge integrations within the cartridge path besides **Klarna Payments SFRA**. No matter of the order of those cartridges in the cartridge path, there are certain templates that need to be overwritten by adding a new if-condition and including the right sub-template:

- \templates\default\checkout\billing\paymentOptions\paymentOptionsContent.isml
- \templates\default\checkout\billing\paymentOptions\paymentOptionsSummary.isml
- \templates\default\checkout\billing\paymentOptions\paymentOptionsTabs.isml

Suppose the website owner requires PayPal as well as Klarna. Each of these templates mentioned above must be copied to a new custom cartridge. The example below shows the new code of the *paymentOptionsContent.isml* template:

```

<isloop items="${pdict.order.billing.payment.applicablePaymentMethods}"
var="paymentOption" status="loopSate">

  <isif condition="${paymentOption.ID === 'CREDIT_CARD'}">
    <isinclu
template="checkout/billing/paymentOptions/creditCardContent" />
  </isif>

  <isif condition="${paymentOption.ID === 'KLARNA_PAYMENTS'}">

```

```

        <isinclude
template="checkout/billing/paymentOptions/klarnaPaymentsContent" />
</isif>
...
<isif condition="${paymentOption.ID === 'PayPal'}">
    <isinclude template="paypal/checkout/paypalContent" />
</isif>
</isloop>
```

Same goes for the other two templates.

3.8. External Interfaces

All requests are done through Klarna's REST API and encrypted using SHA-256 with the shared secret provided by Klarna. Only HTTPS is allowed. JSON is used across all communications.

The full reference guide, along with the resource structure for requests & responses, can be found in the developer portal - <https://docs.klarna.com/klarna-payments/api/>

4. Testing

Klarna has a set of testing credentials and triggers that can be used.

Please, refer to the following URL: <https://docs.klarna.com/resources/test-environment/>

5. Operations, Maintenance

5.1. Data Storage

5.1.1. System Object Extensions

5.1.1.1. Basket

Parameter Name	Attribute ID	Description
Klarna Session ID	kpSessionId	The Klarna session ID returned after “Create Session” API endpoint is called (Applicable since version 21.2.0)
Klarna Client Token	kpClientToken	Client token returned by “Create Session” API endpoint and used to initialize the JS SDK (Applicable since version 21.2.0)
Klarna Client Token	kpClientToken	Client token returned by “Create Session” API endpoint and used to initialize the JS SDK.
Klarna Subscription Frequency	kpSubscriptionFrequency	Subscription frequency values (day, month, etc.)
Klarna Subscription Period	kpSubscriptionPeriod	Predefined subscription period in numbers
Klarna Is Express Checkout	kpIsExpressCheckout	Determines if the basket is an express checkout

Table 2 Basket Attributes

5.1.1.2. Order

Parameter Name	Attribute ID	Description
Klarna Payments Order ID	kpOrderID	The Klarna payments Order ID for Klarna payment method selected by customer

VCN Brand	kpVCNBrand	Klarna Payments virtual card scheme name
VCN Holder	kpVCNHolder	Klarna Payments virtual card holder name
VCN Card ID	kpVCNCARDID	Klarna Payments Virtual Card - Card ID
VCN PCI Data	kpVCNPcidata	Klarna Payments Virtual Card PCI Data in encrypted format
VCN Initialization Vector	kpVCNIV	Klarna Payments Virtual Card Initialization Vector
VCN AES Key	kpVCNAESKey	Klarna Payments Virtual Card AES Key
Is VCN Used	kplsvCN	True if virtual card is enabled & used for payment of the order, otherwise false
Klarna Session ID	kpSessionId	The Klarna session ID returned after “Create Session” API endpoint is called (Applicable since version 21.2.0)
Klarna Client Token	kpClientToken	Client token returned by “Create Session” API endpoint and used to initialize the JS SDK (Applicable since version 21.2.0)
Klarna Subscription Frequency	kpSubscriptionFrequency	Subscription frequency values (day, month, etc.)
Klarna Subscription Period	kpSubscriptionPeriod	Predefined subscription period in numbers

Table 3 Order Attributes

5.1.1.3. *Order Payment Instrument*

Parameter Name	Attribute ID	Description
Klarna Payment Category ID	klarnaPaymentCategoryID	ID of Klarna payment category
Klarna Payment Category Name	klarnaPaymentCategoryName	Name of Klarna payment category

Table 4 Order Payment Instrument Attributes

5.1.1.4. *Payment Transaction*

Parameter Name	Attribute ID	Description
Fraud Status	kpFraudStatus	Klarna Payments order fraud status
Klarna Authorization Token	kpAuthorizationToken	This attribute stores the Klarna authorization token, which is a string value used to authenticate and finalize the order creation process with Klarna's payment system. (Applicable since version 23.2.0)
Klarna Redirect URL	kpRedirectURL	This attribute holds the URL to which the customer is redirected after the payment authorization is successfully completed by Klarna's system. (Applicable since version 23.2.0)

Table 5 Payment Transaction Attributes

5.1.1.5. *Site Preferences*

The site custom preferences have been extended with a new group called “*Klarna_Payments*”. The table below describes the preferences within that group:

Parameter Name	Attribute ID	Description
Auto-capture	kpAutoCapture	When enabled “Yes”, a full order capture will be attempted automatically. The standalone

		order management API capture request will include total order amount value for “captured_amount”. Default value is “No”
Klarna Payments Service Name	kpServiceName	The service name used for the current site
Send product_url and image_url	sendProductAndImageURLs	If set to true, product_url and image_url fields will be included in the Klarna session and order API calls. This enhances shopper experience post purchase. Default value is “Yes”
Merchant Reference 2 Mapping	merchant_reference2_mapping Deprecated and renamed to merchant_reference2 under Klarna_KP	The field from SCC order (basket) object that is mapped to merchant_reference2 field from Klarna API request. Has to be one of the class attributes of SCC LineItemCtnr. Note that for complex data structures result may vary. Note: Merchant Reference 1 value is always set to the SCC order ID
Border Color Preference	kpColorBorder Deprecated and used in kpColorCustomization under site preference group Klarna_KP	CSS (hex value) color set for Border in Klarna Payments iFrame
Border Selected Color Preference	kpColorBorderSelected Deprecated and used in kpColorCustomization under site preference group Klarna_KP	CSS (hex value) color set for selected element Border in Klarna Payments iFrame

Button Color Preference	<code>kpColorButton</code> Deprecated and used in kpColorCustomization under site preference group Klarna_KP	CSS (hex value) color set for Button in Klarna Payments iFrame
Button Text Color Preference	<code>kpColorButtonText</code> Deprecated and used in kpColorCustomization under site preference group Klarna_KP	CSS (hex value) color set for Button text in Klarna Payments iFrame
Checkbox Color Preference	<code>kpColorCheckbox</code> Deprecated and used in kpColorCustomization under site preference group Klarna_KP	CSS (hex value) color set for Checkbox in Klarna Payments iFrame
Checkbox Checkmark Color Preference	<code>kpColorCheckboxCheckmark</code> Deprecated and used in kpColorCustomization under site preference group Klarna_KP	CSS (hex value) color set for checkbox checked(selected) in Klarna Payments iFrame
Details Color Preference	<code>kpColorDetails</code> Deprecated and used in kpColorCustomization under site preference group Klarna_KP	CSS (hex value) color set for details in Klarna Payments iFrame
Header Color Preference	<code>kpColorHeader</code> Deprecated and used in kpColorCustomization under site preference group Klarna_KP	CSS (hex value) color set for Header in Klarna Payments iFrame
Rate limit By Operation	<code>kpRateLimitByOperation</code>	Select “Rate Limit By Operation” to Yes. If it is selected to NO, the default service profile will be utilized. The standard API rate limit for the Klarna service is as mentioned listed on docs.klarna.com . The default service id is <code>klarna.http.defaultendpoint</code> .

Klarna Payment Create New Session When Expires	<code>kpCreateNewSessionWhenExpires</code>	If set to Yes, then a new Klarna session will be created if Klarna session expires before SFCC basket expires
Link Color Preference	<code>kpColorLink</code> Deprecated and used in <code>kpColorCustomization</code> under site preference group <code>Klarna_KP</code>	CSS (hex value) color set for link in Klarna Payments iFrame
Text Color Preference	<code>kpColorText</code> Deprecated and used in <code>kpColorCustomization</code> under site preference group <code>Klarna_KP</code>	CSS (hex value) color set for text in Klarna Payments iFrame
Secondary Text Color Preference	<code>kpColorTextSecondary</code> Deprecated and used in <code>kpColorCustomization</code> under site preference group <code>Klarna_KP</code>	CSS (hex value) color set for secondary text in Klarna Payments iFrame
Border Radius Preference	<code>kpRadiusBorder</code> Deprecated and used in <code>kpColorCustomization</code> under site preference group <code>Klarna_KP</code>	Value (in pixels) of the border radius to be used in Klarna Payments iFrame
Attachments	<code>kpAttachments</code> Deprecated and renamed to <code>kpEMD</code> under <code>Klarna_KP</code>	Toggle (Yes/No) for the inclusion of attachments when creating an order. Specific to inclusion of EMD (customer_account_info, other_delivery_address) when applicable. Default is “No”.
Not available message on billing page	<code>kpNotAvailableMessage</code>	The Klarna Payment not available message on billing page. JSON string holding country code and corresponding message string. For example:

		<pre>{ "GB": "Klarna Payment not available", "default": "Klarna Payment not available" }</pre> <p>Note: This is deprecated and will be removed in next releases!</p>
Virtual Card Number Enabled	kpVCEnabled	If this option is set to “Yes”, Klarna settlement request will generate a Virtual Card Number for every Klarna order. Note: the option will only work if VCN private/public keys are configured properly as mentioned below and public key shared in advance with Klarna
VCN Public Key ID	kpVCNkeyId	UUIDv4 value corresponding to the key pair. Shared with Klarna representative for Production & Playground (test) env configuration
VCN Private Key	vcnPrivateKey	SSL private key used only to decode Virtual Card information (used with kpVCEnabled). Refer to section 9.3 Decrypt VCN Card Details
VCN Public Key	vcnPublicKey	SSL public key used with Virtual Card integration (used with kpVCEnabled). Shared with Klarna and stored here for reference.
VCN Settlement Retry Enabled	kpVCNRetryEnabled	If set to “Yes”, SFCC will retry the VCN settlement once again in

	Deprecated and Renamed to kpVCNRetry under group Klarna_KP	case of service error. Default is "No"
Promotion Price Taxation	kpPromoTaxation	Only use "Based on Adjusted Price" value if you have enabled the corresponding value in "Merchant Tools > Site Preferences > Promotions > Discount Taxation" and use gross taxation. Default: Based on Price
Hide Payment Methods on Deny	kpRejectedMethodDisplay	If set to value other than "No", the Klarna payment method options on the checkout will be greyed out or not displayed to customer in the current view when Klarna authorization request is rejected in the response (.i.e hard reject - "show_form" and "approved" values are both "false")
Alternative Klarna Payment Flow	kpUseAlternativePaymentFlow	If set to "Yes", Klarna Authorization and Order creation steps will be triggered on Checkout Review page when customer clicks CTA/Place Order Button. Default: No
Enable OMS	kpOMSEnabled	If set to "Yes", it updates the order information ingested in SFOMS (SalesForce Order Management System) in the format that Klarna expects. It is required when there is an

		integration between SFCC and SFOMS. Default: No
Klarna Payments Additional Logging	kpAdditionalLogging Deprecated and Renamed to kp_log_data under Klarna_KP	If set to "Yes", it writes additional logging info to get more order details when issue occurs. Default: No
Agent User Name	kpAgentUserName Deprecated and renamed to kpSubsUsername under Klarna_KP	User Agent Name for orders on behalf
Agent User Password	kpAgentUserPassword Deprecated and renamed to kpSubsPassword under Klarna_KP	User Agent Password for orders on behalf
Enable Retry for Recurring Orders	kpEnableRecurringOrderRetry Deprecated and renamed to kpSubsRetryEnable under Klarna_KP	Enable retry for failed orders
Number Of Retries	kpRecurringNumberOfRetry Deprecated and renamed to kpSubsRetryNumber under Klarna_KP	Recurring number of retries
Recurring Retry Frequency	kpRecurringRetryFrequency Deprecated and renamed to kpSubsRetryFrequency under Klarna_KP	Retry frequency for recurring orders
Klarna Create Order Token	kpCreateOrderToken	Token used to verify the caller for recurring orders

Use Bank Transfer callback	<code>kpBankTransferCallback</code>	If set to "Yes", SFCC will wait for Klarna callback to place an order. Default is "No"
Klarna Express Checkout (Klarna_ExpressCheckout) deprecated from 24.4.0		
Klarna Express Checkout Enabled	<code>kpeCEEnabled</code>	Boolean flag to enable/disable Klarna Express Checkout
Klarna Express Checkout Button Theme	<code>kpeCButtonTheme</code>	The theme of the button. Options include default, light & dark
Klarna Express Checkout Button Shape	<code>kpeCButtonShape</code>	The shape of the button. Options include default, rect & pill
Placement	<code>kec_placement</code>	Multiselect attribute to choose where to display express checkout buttons. Values: cart, pdp, minicart
Klarna Express Checkout (Klarna_KEC)		
Enable Express Checkout	<code>kec_enable</code>	Boolean flag to enable/disable Klarna Express Checkout
Theme	<code>kec_theme</code>	The theme of the button. Options include default, light & dark
Button Shape	<code>kec_shape</code>	The shape of the button. Options include default, rect & pill
Placement	<code>kec_placement</code>	Multiselect attribute to choose where to display express checkout buttons. Values: cart, pdp, minicart
Klarna_SignIn(Klarna Sign In)		
Enable/Disable Klarna SignIn	<code>isKpSigninEnabled</code> – Deprecated with version 24.4.0 Renamed to <code>siwk_enable</code>	Enable /Disable Klarna sign in

Klarna Button Logo Alignment	kpSignInButtonLogoAlignment - Deprecated with version 24.4.0 Renamed to siwk_alignment	Alignment of Klarna Button.Options include Badge (Recommended), Left, Center. Default value is Badge(Recommended)
Klarna Button Shape	kpSignInButtonShape - Deprecated with version 24.4.0 Renamed to siwk_shape	Shape of Klarna Button.Options include Rounded corners (Recommended), Rectangular, Pill with default value Rounded corners(Recommended)s
Klarna Button Theme	kpSignInButtonTheme - Deprecated with version 24.4.0 Renamed to siwk_theme	Theme of Klarna Button.Options include Light (Recommended), Dark, Outlined with default value Light(Recommended)
Customer Unique Identifier Login Method	kpSignInCustomerUniqueId - Deprecated with version 24.4.0	Options include Bank ID and OTP
Klarna SignIn Environment	kpSignInEnvironment - Deprecated with version 24.4.0 Removed from preference and handled in code with value playground	Preference for configuring environment.Options include Playground and Production
Klarna Sign in web flow options	kpSignInInteractionMode - Deprecated with version 24.4.0 Removed from preferences and used default value as DEVICE_BEST	Interaction mode when Klarna button is clicked .Options include redirect url,popup and device best.Default value is device best
Auto Login to Klarna payments	kpSignInIsAutoLogin - Deprecated with version 24.4.0	Boolean to automatically loginto Klarna payments
Link Existing user	kpSignInLinkExistingUser - Deprecated with version 24.4.0	To link existing users.Options include Merge accounts

	Always merge accounts without any prompt	automatically and prompt to merge accounts
Redirect Uri	kpSignInRedirectUri - Deprecated with version 24.4.0 Renamed to siwk_redirect_url	Redirect url after sign-in
Script URL	kpSignInScriptURL - Deprecated with version 24.4.0 Removed preferences and handled in code .Used the script url in constants	Klarna Sign in Script URL
Sign In scope	kpSignInScope - Deprecated with version 24.4.0 Renamed to siwk_scope	Sign in scope.Uses a multiselect list of all scopes available
Sign In Placement	siwk_placement	Placement of sign in button.Multiselect List of strings which include loginpage and checkout page.
Klarna OSM(On site Messaging)		
Enable On-Site Messaging	osm_enable	Enable/Disable OSM functionality
Theme	osm_theme	Theme for on site messaging .Options include default,dark and custom
Placement	osm_placement	Multiselect dropdown which includes cart,pdp,header,footer and info. Placement Tag IDs are hardcoded in code
Custom Styling	osm_custom_styling	JSON to store custom styling
Klarna_KP(Klarna Payments)		

Enable Klarna Payments	kp_enable	Enable/Disable Klarna Payments.Boolean attribute with Default value as true
Color Customization	kpColorCustomization	JSON attribute to customize colour
Enable Extra Merchant Data	kpEMD	Enable this option to include customer_account_info and other_delivery_address as attachments when creating an order.Boolean with default value false
VCN - Enable settlement retry	kpVCNRetry	When set to "Yes", SFCC will automatically retry the VCN settlement in the event of a service error. Boolean with default value false
Log Extra debug data	kpLogExtraData	Attribute to log debug data.Boolean with default value true
Merchant Reference 2	merchant_reference2	Enter the attribute from the SCC order (basket) object that you would like to forward as an additional merchant reference to Klarna. This attribute must be a class attribute of the SCC LineItemCtnr. String attribute
Agent username for subscriptions	kpSubsUsername	Agent username used to login on behalf of customers to create subscription orders.String attribute
Agent password for subscriptions	kpSubsPassword	Agent password to login on behalf of customers to create

		subscription orders.Password Attribute
Enable retry for subscriptions	kpSubsRetryEnable	Boolean with default value false
Number of Retries	kpSubsRetryNumber	Number attribute
Recurring retry frequency	kpSubsRetryFrequency	Number attribute

Table 6 Site Preferences

5.1.1.6. *Product*

Parameter Name	Attribute ID	Description
Is Klarna Standard Product	kplIsStandardProduct	Boolean attribute to define if the product is standard. (Applicable since version)
Is Klarna Subscription Product	kplIsSubscriptionProduct	Boolean attribute to define if the product is eligible for subscription.
Klarna Trial Days Usage	kpTrialDaysUsage	Numeric value used for free trial definition.

5.1.1.7. *ProductLineItem*

Parameter Name	Attribute ID	Description
Is Selected for Subscription product	kpSubscription	Boolean attribute to define if the product is selected for subscription in the basket.
Enriched data for product line item.	klarna_oms__lineItemJSON	String data for product line item.

5.1.1.8. *Profile*

Parameter Name	Attribute ID	Description

Klarna Subscriptions	kpSubscriptions	Text attribute to store customer subscriptions to Klarna
----------------------	-----------------	--

5.1.2. Custom Objects

5.1.2.1. Klarna Express Button

The KEB button can be configured and adapted to the needs of your storefront. Steps to enable KEB button:

- To enable the button, your Klarna Merchant ID (MID) and domains must be white-listed. Please reach out to your Klarna delivery manager for support with this activity prior to testing and production go-live (as the domain names white-listed defer). Please ensure that the MID matches that which is configured in the BM service credentials for your storefront.
- To configure the KEB settings for a given site, you must visit “***Merchant Tools – Custom Object Editor***” and search for **KlarnaCountries** custom object.
- Choose the respective country key (e.g.: “US”) and provide the below details:
 - Express Button Merchant ID: *Allow-listed Merchant ID*
 - Express Button environment: *Klarna environment – Production/Playground*
 - Express Button Cart Enabled: *Check box to display on cart page*
 - MiniCart Express Button Enabled: *Check box to display on minicart page*
 - MiniCart Express Button Theme/shape: *theme/shape to display on minicart page*
 - Express Button Library URL: [Available here](#)
 - Express Button Category: *The Klarna category to be pre-selected on Checkout, e.g., “pay_over_time”*
- Please review Figure 45 for reference.

More information to customize the button can be found [here](#).

Note: Klarna Express Button is currently available in [these](#) markets with more to follow

PDP Placement Data Key: credit-promotion-badge

Header Placement Tag Enabled:

Header Placement Data Key: top-strip-promotion-standard

Footer Placement Tag Enabled:

Footer Placement Data Key: footer-promotion-auto-size

Info Page Placement Tag Enabled:

Info Page Placement Data Key: info-page

Data Inline Enabled on PDP/Cart placement (Canada only):

URL to On-Site Messaging Library URL: <https://na-library.playground.klarnaservices.com/lib.js>

Express Button

Express Button Merchant ID: PNXXXXX

Express Button Environment: playground (Playground)

Express Button Theme: light (Light)

Express Button Shape: rect (Rectangle)

Express Button Cart Enabled:

Express Button Library URL: <https://x.klarnacd.net/express-button/v1/lib.js>

Express Button Category: pay_over_time

MiniCart Express Button

MiniCart Express Button Enabled:

MiniCart Express Button Theme: default (Default)

Minicart Express Button Shape: pill (Pill)

Figure 59 Klarna Express Button completed configuration

Shoppers experience on the storefront cart is as below:

1. Lands on cart page where KEB is displayed
2. Clicks on KEB button
3. Shopper prompted to share their Klarna registered credentials to complete authentication
4. Re-direct to checkout with Klarna Payment Method pre-selected

5.1.2.2. *KlarnaCountries*

The respective object is dynamically selected based on the request locale country, e.g., SFCC site with locale “**de_DE**” or “**en_DE**” will use the “**DE**” custom object. In cases when the request locale country can’t be dynamically resolved (i.e. with “default” SFCC locale) – attribute “**klarnaLocale**” can be utilized to pass the proper locale to Klarna. For all other cases, this field can be left blank and will not be taken into consideration.

Even if you have locales that are not supported by Klarna Payments, we recommend creating a corresponding entry in the custom object for that locale. Thus, on the billing page of the unsupported locale you will have the Klarna Payments widget showing an appropriate message.

The custom objects store data such as Klarna default locale, service credential IDs and Klarna Payments placement data keys to ensure that Klarna Payments integration is correctly configured.

Note: The same custom object is used by Klarna Checkout cartridge integration!

ID or Name	Type	Attribute Settings	Values
Select All			
UUID	String	*	Edit
country	String	*	Edit
creationDate	Date+Time	*	Edit
credentialID	String		Edit
internalLocale	String		Edit
lastModified	Date+Time	*	Edit
osmCartEnabled	Boolean		Edit
osmCartTagId	String		Edit
osmLibraryUrl	String		Edit
osmPDPEnabled	Boolean		Edit
osmPDPTagId	String		Edit
osmUCI	String		Edit

Figure 60 KlarnaCountries Attributes

The table below describes attributes of the **KlarnaCountries** custom object:

Attribute Name	Attribute ID	Description
Country Code	country	Two-letter country code
On-site Messaging Data Default Locale	klarnaLocale	Fallback, if the request locale can't be dynamically resolved, i.e., when using "default" SFCC locale
Service Credential ID	credentialID	The ID of service credentials for this locale.
On-site messaging Data Client ID	esmUCI	The Klarna On-site Messaging "data-client-id" applicable for a given country
Cart Placement Tag Enabled	esmCartEnabled	To enable Cart Placement for a given locale.
Cart Placement Tag ID	esmCartTagId	The Klarna On-site Messaging "data-key" of placement applicable for Cart Page for a given locale.
Cart Placement Theme	esmCartTheme	The theme for cart placement tag. It could be default or dark.
Cart Placement Custom Styling (#osm-cart-placement)	esmCartCustomStyling	Custom styling for cart placement. The css should be wrapped in <style>...customCss...</style> element.
PDP Placement Tag Enabled	esmPDPEnabled	To enable PDP Placement for a given locale.
PDP Placement Tag ID	esmPDPTagId	The Klarna On-site Messaging "data-key" of placement applicable for Product Display page for a given locale.
PDP Placement Theme	esmPDPTheme	The theme for pdp placement tag. It could be default or dark.
PDP Placement Custom Styling (#osm-pdp-placement)	esmPDPCustomStyling	Custom styling for pdp placement. The css should be wrapped in <style>...customCss...</style> element.

Attribute Name	Attribute ID	Description
Header Placement Tag Enabled	<code>osmHeaderEnabled</code>	To enable Klarna Header Placement in a given storefront
Header Placement Data Key	<code>osmHeaderTagId</code>	The Klarna On-site Messaging “data-key” of placement applicable for the Header
Header Placement Theme	<code>osmHeaderTheme</code>	The theme for header placement tag. It could be default or dark.
Header Placement Custom Styling (#osm-header-placement)	<code>osmHeaderCustomStyling</code>	Custom styling for header placement. The css should be wrapped in <code><style>...customCss...</style></code> element.
Footer Placement Tag Enabled	<code>osmFooterEnabled</code>	To enable Klarna footer Placement in a given storefront
Footer Placement Data Key	<code>osmFooterTagId</code>	The Klarna On-site Messaging “data-key” of placement applicable for the footer
Footer Placement Theme	<code>osmFooterTheme</code>	The theme for footer placement tag. It could be default or dark.
Footer Placement Custom Styling (#osm-footer-placement)	<code>osmFooterCustomStyling</code>	Custom styling for footer placement. The css should be wrapped in <code><style>...customCss...</style></code> element.
Info Page Placement Tag Enabled	<code>osmInfoPageEnabled</code>	To enable Klarna Info Page Placement in a given storefront
Info Page Placement Data Key	<code>osmInfoPageTagId</code>	The Klarna On-site Messaging “data-key” of placement applicable for the Info Page
Info Page Placement Theme	<code>osmInfoPageTheme</code>	The theme for info page placement tag. It could be default or dark.

Attribute Name	Attribute ID	Description
Info Page Placement Custom Styling (#osm-info-page-placement)	esmInfoPageCustomStyling	Custom styling for info page placement. The css should be wrapped in <style>...customCss...</style> element.
Data Inline Enabled on PDP/Cart placement (Canada only)	esmDataInlineEnabled	Enable this when using PayBright payment method in Canada
URL to On-Site Messaging Library URL	esmLibraryUrl	<p>URL for On-Site Messaging library, applicable for testing or production must be saved.</p> <p>Please use only Klarna production URL in live storefront. Verify test environment URL which includes "playground" in URL For production or live environment, ensure URL includes "production". E.g: Old library Test URL></p> <p>https://na-library.playground.klarnaservices.com/lib.js</p> <p>, e.g: Live URL>https://na-library.production.klarnaservices.com/lib.js</p> <p>New library - https://js.klarna.com/web-sdk/v1/klarna.js</p>
On-Site Messaging Environment	esmEnvironment	Environment on which the OSM is used. It could be playground or production.
Express Button Environment	kebEnvironment	The express button environment. Default is playground
Express Button Merchant ID	kebMerchantID	The merchant ID used to display the button

Attribute Name	Attribute ID	Description
Express Button Library URL	kebLibraryUrl	URL of the express button library
Express Button Cart Enabled	kebCartEnabled	To enable the Klarna Express Button on cart page
Express Button Theme	kebTheme	The theme of the button. Options include default, light & dark
MiniCart Express Button Enabled	kebMCEnabled	To enable the Klarna Express Button on minicart
MiniCart Express Button Theme	kebMCTheme	The theme of the button. Options include default, light & dark
Express Button Category	kebCategory	The Klarna category to be pre-selected on Checkout Page, e.g., "pay_over_time"
Klarna Express Checkout Client Key	expressCheckoutClientKey	The Klarna Express client key used for express checkout

Table 7 KlarnaCountries Attributes

The data-client-id and data-key values used in the OSM placements are available in the Klarna Merchant Portal (Europe/US (CA included)/Oceania) within the On-site Messaging App. When selecting the data-key values, ensure that the filter is set to the right country and language.

5.1.2.3. *KlarnaSigningKey*

The respective object is populated when we subscribe to webhooks. For webhook subscription, a signing key gets generated, and this key will get saved to this custom object. The signing key will be later used in code for validating the webhook notification that is received.

The table below describes attributes of the **KlarnaSigningKey** custom object:

Attribute Name	Attribute ID	Description
Signing Key Id	signingKeyId	The signing key id which will be used for subscribing to webhooks.
Signing Key	signingKey	The signing key which will be used for HMAC validation of webhook notification received.

Attribute Name	Attribute ID	Description
Created Date	createdDate	The date when the signing key was created.

Please note that the value in this custom object gets created when ‘CreateWebhook’ job is run. The values in this custom object should not be altered manually.

Custom Objects > Custom Objects > krn:partner:global:notification:signing-key:7a39f909-cd45-4420-943e-76e2d119afc1 - General

General

Manage 'krn:partner:global:notification:signing-key:7a39f909-cd45-4420-943e-76e2d119afc1' (KlarnaSigningKey)

Fields with a red asterisk (*) are mandatory. You can view and edit the name and description in other languages, if required. Click **Apply** to save the details.

Signing Key	
Signing Key Id: *	krn:partner:global:notification:signing-key:7a39f909-cd45
Signing Key:	FLBwJCT0V1I1+N9pVXXoJPwAk/yN61ms9DLiO+O1fEggk
Created Date:	2025-10-06T06:40:17.748Z

Figure 61 KlarnaSigningKey sample record

5.1.2.4. *KlarnaWebhookNotification*

The respective object is populated when webhook notifications are received. The notification status will be ‘PROCESS’ initially and once its processed, its state gets changed accordingly. The notification log will contain the notification request and this will be used later while processing the notification.

The table below describes attributes of the **KlarnaWebhookNotification** custom object:

Attribute Name	Attribute ID	Description
Payment Request ID	paymentRequestId	The payment request id will be used to map to the customer who has initiated the payment request.
Notification Log	notificationLog	The notification log will contain the notification request, and this will be used later while processing the notification.

Attribute Name	Attribute ID	Description
Notification Status	notificationStatus	The notification status will be ‘PROCESS’ initially and once its processed, its state gets changed accordingly.

Please note that the value in this custom object gets created when notifications are received for the payments initiated in the site. The values in this custom object should not be altered manually.

Custom Objects > Custom Objects > krn:payment:us1:request:ddb2fedd-4a82-6e16-b131-3e911e3215f1 - General

General

Manage 'krn:payment:us1:request:ddb2fedd-4a82-6e16-b131-3e911e3215f1' (KlarnaWebhookNotification)

Fields with a red asterisk (*) are mandatory. You can view and edit the name and description in other languages, if required. Click **Apply** to save the details.

Klarna Webhook Notification

Payment Request ID: * krn:payment:us1:request:ddb2fedd-4a82-6e16-b131-3e911e3215f1

Notification Log: {"metadata": {"event_id": "af7cb161-e385-4b44-85f5-558d"}}

Notification Status: PROCESS (PROCESS)

Figure 62 KlarnaWebhookNotificaiton sample record

5.1.3. Session Attributes & Cookies

The following session custom attributes are saved in “**session.privacy**” storage and accessible in checkout. The attributes are retained for the session lifetime & cleared when the customer logs out of their profile.

Attribute	Description
KlarnaLocale	The Klarna locale in use

Attribute	Description
KlarnaPaymentsSessionID (Not applicable since version 21.2.0)	The Klarna session ID returned after “Create Session” API endpoint is called
KlarnaPaymentsClientToken (Not included as session attributes since version 21.2.0)	Client token returned by “Create Session” API endpoint and used to initialize the JS SDK
KlarnaPaymentMethods	The available payment method categories for the respective Klarna session; Saved in JSON format
KlarnaPaymentsAuthorizationToken	The authorization token returned by JS SDK “Authorize” call
KPAuthInfo	Whether finalization is required for the payment method; Returned by JS SDK “Authorize” call; Saved in JSON format
KlarnaExpressCategory	The KEB payment category; Currently applicable for US and defaults to “pay_over_time”
KlarnaPaymentsRedirectURL	The URL to redirect the customer to after placing the order; Returned by the “Create Order” API call

Table 8 Klarna Session Attributes

The following cookies are being set by Klarna integration:

Cookie Name	Description
selectedKlarnaPaymentCategory	The selected payment method on checkout (e.g. “pay_later”)

Table 9 Klarna Cookies

5.1.4. Library

In addition to the configurations, the following 2 library assets will be added:

- “**footer-about**” – Updated OOTB asset including link to Klarna OSM dedicated page in the footer.

- “**klarna-email-info**” – Asset containing links to review the Klarna Payment information. Used in the confirmation email sent to the customers.

5.1.5. Services

An HTTP service “**klarna.http.defaultendpoint**” has been added with “**klarna.http.service**” profile and service credentials for each country (described in **KlarnaCountries** custom object).

Please, note that up until version **21.2.0** of the cartridge, the **KlarnaCountries** custom object was **replicable**. To avoid issues with service credentials during replication, merchants should use the same service credential name in staging, development, and production environments!

Please, review section **9.4** Update KlarnaCountries Definition on possible ways to update the definition in your instances.

An HTTP service ‘**klarna.http.webhook**’ has been added for subscribing to webhooks. This service will be used when ‘CreateWebhook’ job is run.

5.1.6. Jobs

5.1.6.1. CreateWebhook

This job is used for subscribing to webhooks. The payment types to which we need to subscribe should be entered in the event types section before executing the job. When the job is run,

- It checks whether a signing key is already present for the storefront site for which it is run. If it is not present, then the signing key is created and saved in the custom object -KlarnaSigningKey. If present, then the signing key id from the custom object will be used.
- Using the signing key id and then event types configured, webhook API is called for subscribing to webhooks.

Once webhook is subscribed, for the subscribed payment types we will start receiving webhook notification which gets saved to the custom object **KlarnaWebhookNotification**.

5.1.6.2. DeleteWebhookNotification

This job is used for deleting the webhook notifications which are no longer needed. A custom parameter, expiry days, is present which needs to be configured.

Any notification which is older than the expiry days configured gets deleted by this job.

5.2. Logs

The integration includes the following logs:

- Service communication logs – starts with “service-klarna-***”. These logs contain every request and response to the Klarna endpoints. Personal information, i.e. emails & names required for the Klarna API calls are masked in the logs.
- Custom errors and debug info are logged under “customerror-***”, “custodebug-***” & “custominfo-***” files depending on the case.

5.3. Availability

Cartridge functionality will be dependent on the availability of the Klarna API service. Current Klarna operational status can be viewed here - <http://status.klarna.com/>

5.4. Failover/Recovery Process

If Klarna API is not available, Klarna is not presented as a payment option. In case of any failure within the Klarna API, contact Klarna for support.

5.5. Support

A customer service workshop can be conducted during the implementation process before going live to align the operational processes and ensure customer satisfaction. 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 (Android/iOS). The customers can contact support, view their statements, pay for their purchase, track delivery updates, and prolong the due dates if they have chosen to pay after delivery.

5.5.1. Merchant Support

Reporting core SFCC functionality issues in the Klarna cartridge technical integration – please contact commercecloud@klarna.com

For production issue related to Klarna API availability, merchant representative should reach their Klarna Account manager after reviewing the current operational status at <http://status.klarna.com/>. Report the problem in Production (Post Go-live) if you have a suspicion about degraded performances or issues with Klarna's service. The Klarna contact would then be able to report this internally to the incident management team who have established routines to handle and resolve reported incidents. The Klarna contact may request additional information from the individual reporting the problem to help internal team ascertain and identify the issue. The KAM may also advise the merchant to follow the updates on the status page if it is a known 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 MID or market
- Impact and examples of customer orders (order_id or Klarna session_id if available)
- Screenshots, timeframe, additional information as required

6. User Guide

6.1. Cartridge upgrade

Regular updates to our cartridge code include bug fixes, performance enhancements, security patches, and new features. It's essential to stay up to date with these changes to ensure compatibility, security, and optimal functionality of your integration.

6.2. Upgrade Process

Follow these steps to upgrade your custom code with the latest version of our cartridge:

6.2.1. *Review Release Notes*

Start by reviewing the release notes for the latest version of the cartridge. The release notes outline the changes, improvements, and any potential compatibility considerations associated with the upgrade.

6.2.2. *Assess Custom Code Changes*

Identify any customizations or modifications you have made to the SFCC cartridge code in your custom files. These may include storefront customizations, controller adjustments, or custom business logic built on top of our cartridge.

6.2.3. *Backup Custom Files*

Before proceeding with the upgrade, ensure that you have a backup of your custom files, including any modifications made to the SFCC cartridge code. This backup will serve as a safety net in case any issues arise during the upgrade process.

6.2.4. *Compare Code Differences*

Use a version control system or a file comparison tool to compare the differences between your custom code and the latest version of the SFCC cartridge code. Pay close attention to areas where changes have been made to ensure compatibility and maintain functionality.

6.2.5. *Update Integration Code*

Integrate the latest version of the SFCC cartridge code into your custom files, replacing any outdated or deprecated code with the new implementations. Follow the migration guides and best practices provided to ensure a smooth transition.

6.2.6. Test and Validate

After updating your custom code, thoroughly test the integration to ensure that all functionality works as expected. Test various scenarios, including user interactions, data processing, and third-party integrations, to identify any potential issues or regressions.

6.2.7. Address Compatibility Issues

If you encounter any compatibility issues or conflicts with your existing custom code, troubleshoot and resolve them accordingly. Consult our support resources or reach out to our team for assistance in addressing compatibility concerns.

6.2.8. Deploy Changes

Once you are satisfied with the upgrade and have validated its functionality, deploy the changes to your production environment. Monitor the integration closely following deployment to ensure ongoing stability and performance.

6.3. Roles, Responsibilities

There are no recurring tasks required by the merchant. Once configurations are set up, the functionality runs on demand.

6.4. Storefront Functionality

When Klarna has been setup, the Klarna Payments options and iframe widgets will be shown on the billing step. All the SFCC OOTB checkout functionality remains in place, such as but not limited to cart updates during checkout, checkout with applied coupon(s) code(s), checkout with applied product level promotion, checkout with applied order level promotion, checkout with applied shipping level promotion, checkout with applied order level promotion with bonus product.

Select one of Klarna's payment options as the payment method on billing step of checkout process and click the "Next: Place Order" button:

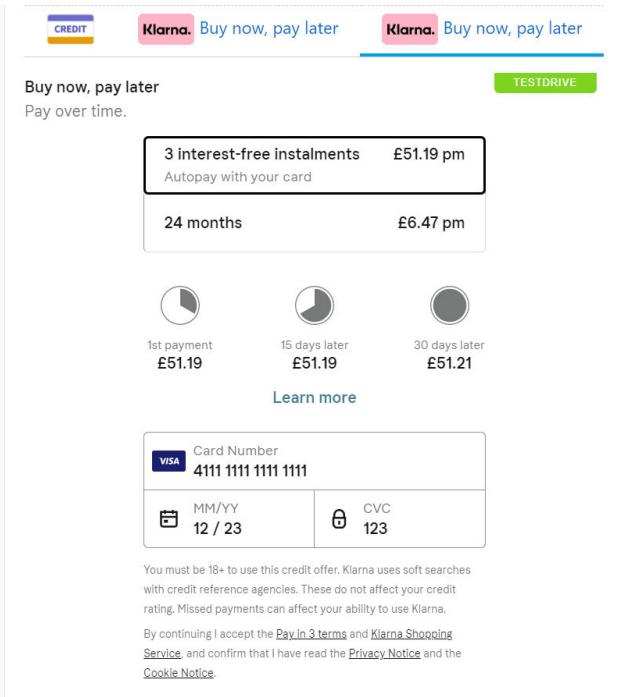


Figure 63 Payment Options on Checkout

Depending on the payment method selected and the region, you will see one of Klarna's popup windows to provide the details. Follow the steps on the screen:

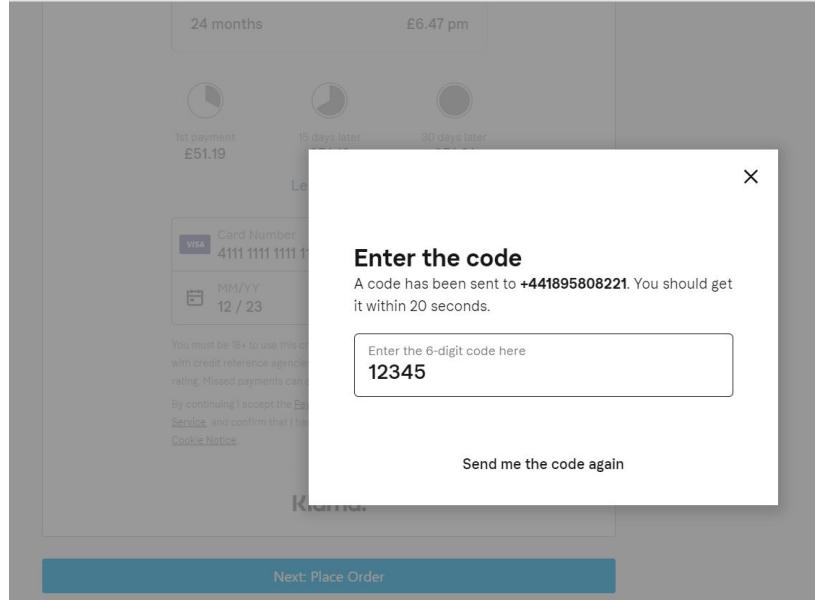


Figure 64 Klarna Popup Screen

On the Review step click on “Place Order” button:

The image shows a payment review screen. On the left, there are sections for 'Shipping Address' (John Doe, 13 New Burlington St, London W13 3BG, 01895808221) and 'Shipping Method' (2-Day Express (2 Business Days), £11.99). On the right, there is a summary table:

Subtotal	£153.59
Shipping	£11.99
Shipping Discount	- £11.99
Total	£153.59

Below the summary, there is a section for '1 Items' (Charcoal Flat Front Athletic Fit Shadow Striped Wo...). It shows a small image of a person wearing a striped shirt, details like 'Colour: Charcoal', 'Size: 36', 'Width: Short', and 'In Stock', and a breakdown of costs: 'Each £320.00', 'Quantity 1', and 'Total £194.99' (which appears to be a mistake as it matches the subtotal). At the bottom, there is a blue 'Place Order' button.

Figure 65 Payment Review Screen

The customer's browser is sent to the “**redirect_url**” and immediately thereafter shown the Commerce Cloud Order Confirmation page.

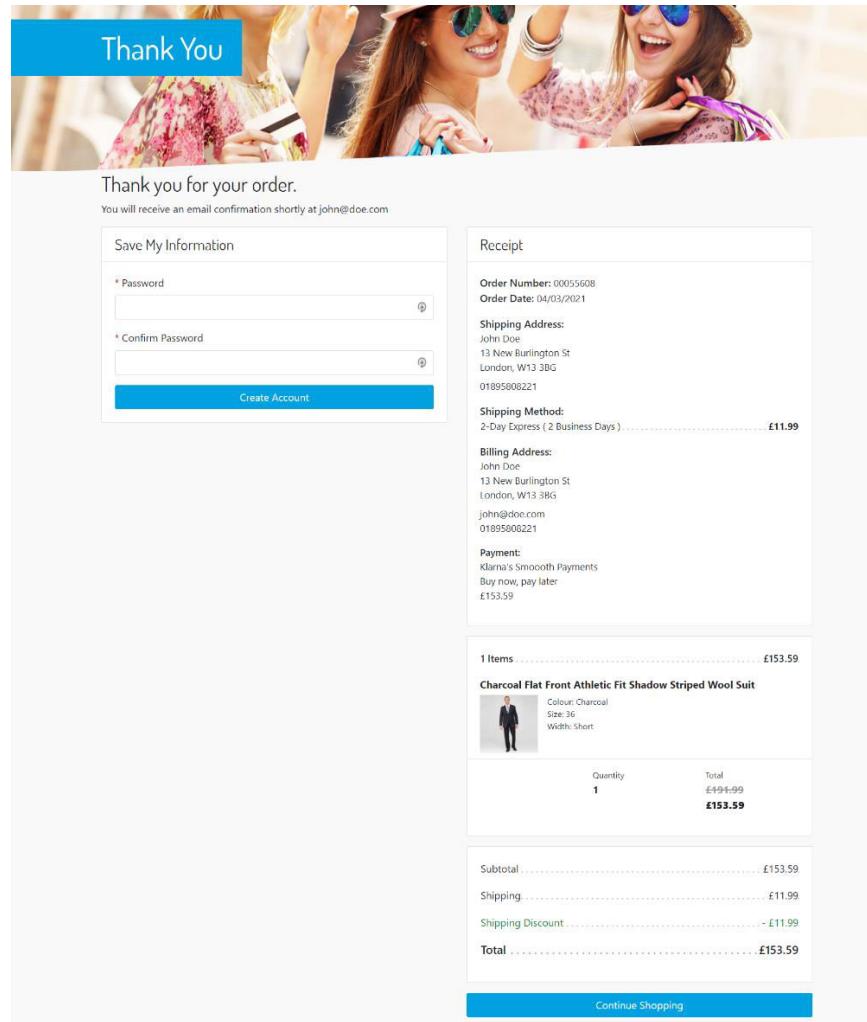


Figure 66 Order Confirmation Page

The newly created order can be inspected in Business Manager:

Orders				
You're using the new Search service.				
This page allows you to search for orders by order number. Select Advanced to use more search options. Select By Number to search by or newline. Entered text is treated as case-sensitive; substring matching isn't supported.				
Order Search				<input type="button" value="Find"/>
Number	Order Date	Site	Created By	Registration Status
00055608	3/4/21 4:38:34 pm Etc/UTC	RefArchGlobal	Customer	Unregistered
00055909	3/3/21 8:13:04 pm Etc/UTC	RefArchGlobal	Customer	Registered
00055409	3/2/21 10:07:53 am Etc/UTC	RefArchGlobal	Customer	Unregistered
00055311	3/1/21 10:07:01 pm Etc/UTC	RefArchGlobal	Customer	Registered
00055309	3/1/21 9:55:21 pm Etc/UTC	RefArchGlobal	Customer	Registered
00055307	3/1/21 9:47:04 pm Etc/UTC	RefArchGlobal	Customer	Registered
00055306	3/1/21 6:57:04 pm Etc/UTC	RefArchGlobal	Customer	Registered
00054903	2/19/21 10:10:07 am Etc/UTC	RefArchGlobal	Customer	Registered
00054802	2/18/21 9:15:16 am Etc/UTC	RefArchGlobal	Customer	Registered
00054606	2/10/21 12:33:17 pm Etc/UTC	RefArchGlobal	Customer	Registered

Figure 67 Orders List in BM

Klarna Payments order id can be inspected in the Attributes tab of the order:

Merchant Tools > Ordering > Orders > Order: 00055608(RefArchGlobal)

General	Attributes	Payment	Notes	History								
Attributes for Order '00055608'												
On this page you can edit the attributes of the order. Fields with a red asterisk (*) are mandatory. Click Apply to save changes. Click Reset to revert your changes.												
<table border="1"> <tr> <td colspan="2">Klarna Payments</td> </tr> <tr> <td>Klarna Payments Order ID:</td> <td>6fa5c3ad-411d-2629-b543-4f5ea672ea9e</td> </tr> <tr> <td>Is VCN Used:</td> <td><input type="checkbox"/></td> </tr> <tr> <td>VCN Card ID:</td> <td><input type="text"/></td> </tr> </table>					Klarna Payments		Klarna Payments Order ID:	6fa5c3ad-411d-2629-b543-4f5ea672ea9e	Is VCN Used:	<input type="checkbox"/>	VCN Card ID:	<input type="text"/>
Klarna Payments												
Klarna Payments Order ID:	6fa5c3ad-411d-2629-b543-4f5ea672ea9e											
Is VCN Used:	<input type="checkbox"/>											
VCN Card ID:	<input type="text"/>											
<< Back to List												

Figure 68 Order Attributes

Payment method details can be inspected on the Payment tab of the order, and it should be Klarna:

Merchant Tools > Ordering > Orders > Order: 00055608(RefArchGlobal)

General Attributes **Payment** Notes History

Payment Information for Order '00055608'

Order Total:	£153.59
Amount Paid:	£0.00
Balance Due:	£153.59
Invoice Number:	00242008
Payment Status:	Paid
Payment Method:	KLARNA_PAYMENTS Processor: KLARNA_PAYMENTS Transaction: 6fa5c3ad-411d-2629-b543-4f5ea672ea9e Amount: £153.59 Klarna Payment Category ID: pay_over_time Klarna Payment Category Name: Buy now, pay later Fraud Status: ACCEPTED

<< Back to List

Figure 69 Order Payment Detail

Order can be further inspected in Klarna Merchant Portal:

- EU: eu.portal.klarna.com
- US: us.portal.klarna.com
- OC: us.portal.klarna.com

#7BRVXW4H Captured

£153.59

Merchant reference 1 00055608 Edit	Merchant reference 2 cfbfbf55f8f98bc392030178105 Edit	Created Mar 4, 2021, 6:38 PM	Expires Apr 1, 2021, 3:00 AM	Merchant ID K500726
---	--	--	--	-------------------------------

Customer

Shipping address

John Doe
13 New Burlington St
London
W13 3BG
GB
Tel
01895808221
Email
john@doe.com
Edit shipping address

Billing address

Additional Info

Order lines (2)

[Refund](#) [Print packing slip](#)

	Item / Reference	Qty	Unit price	Discount	Tax	Amount
<input type="checkbox"/>	Charcoal Flat Front Athletic Fit Shadow Striped Wool Suit 640188017003M	1	191.99	38.40	5% 7.31	£153.59 Captured
<input type="checkbox"/>	2-Day Express GBP002	1	11.99	11.99	5% 0.00	0.00 Captured

PAYMENT DETAILS

Initial Payment Method
Pay later in parts
VISA 411111*****1111
Resend statement

ORDER TOTAL

£153.59

Captured	£153.59
Refunded	£0.00
Not Captured	£0.00

CUSTOMER BILLED

£153.59

Activity Log

- Mar 4, 2021 6:38 PM **Captured: £153.59** Via API
- 6:38 PM **Order placed: £153.59** By Klarna

Klarna. Copyright © 2005-2021 Klarna Bank AB (publ). All rights reserved

[Terms & Conditions](#)

Figure 70 Klarna Portal Order View

7. Known Issues

The LINK Cartridge has no known issues.

8. Release History

Version	Date	Changes
18.1.0		Initial release of Klarna Payments SFRA.
19.1.0		Added SFRA version
19.1.1		Updated VCN to use the newest API version
19.1.2		Fix auto capture for the pipelines cartridge
19.1.4		New country locales added. Minor bug fixes. Cartridge templates and forms updated for latest SFRA.
19.1.5		Added additional verification for all notifications. Minor fixes around the configuration objects. Added Canadian support. Documentation updates.
19.1.6		New country locales added. Updated VCN to store encrypted card details
21.1.0		Fixes around discounts taxation & VCN error handling. Added VCN improvements, additional OSM placements, BOPIS support. New IT, CA, FR & NZ country locales. Removed acknowledge call. Documentation updates.
21.1.1		New On-Site Messaging attribute for Canada. Remove not required locale templates for SG Spain & Belgium. Documentation updates.

Version	Date	Changes
21.1.2		<p>Fixed core file naming convention issues in 21.1.0 and 21.1.1. Please upgrade to the latest version if you are currently using 21.1.0 or 21.1.1.</p> <p>Removed deprecated “scripts/util/Builder.js” file.</p>
21.2.0		<p>Added Klarna Express Button.</p> <p>Moved Klarna session ID & client token from SFCC session privacy to Basket attributes.</p> <p>Changed KlarnaCountries definition to not replicable.</p> <p>Code cleanup.</p> <p>Documentation updates.</p>
21.3.0		<p>Improvements for create_session errors</p> <p>Expired user session issues related to empty shipment.shippingMethod</p> <p>Additional locale (PL) included in config files</p>
21.3.1		Documentation updates.
22.1.0		<p>Improvements for create and update session errors</p> <p>Added Klarna Express Button in minicart.</p> <p>Support for long running basket</p> <p>Rate-limits by operations</p>
22.2.0		<p>One Klarna Optimisation</p> <p>Mexico locale support</p>
22.2.1		Rollback of One Klarna Optimisation
22.3.0		<p>SFRA ver. 6.0.0 support</p> <p>Rollback hide VAT from Checkout functionality</p>

Version	Date	Changes
22.3.1		Fix User-agent version sent to Klarna services
22.4.0		Intent field addition in Klarna Payment session creation Combine Klarna Authorization and Create Order in Checkout Review Step
22.5.0		OMS support
23.1.0		Fix issue with incorrect values for EMD Improvement Klarna Auto Capture and error handling Logging information for troubleshooting bugs Add Auto_finalise=True to the review checkout flow
23.1.1		Fixed an issue where sessions with negative order_tax_amount occurred due to SFCC session expiration Compatibility mode 21.2 support Replace deprecated window.KlarnaOnsiteService.push with window.Klarna.OnsiteMessaging.refresh.
23.2.0		Subscription Payments support: recurring payments and subscription handling directly within the SFCC environment. This update includes configuration options, subscription management in the cart and checkout pages, and a customer dashboard for subscription oversight. Klarna Bank Transfer payments: added a new server-side authorization callback feature for Klarna Bank Transfer payments, enhancing reliability across EU markets and supporting all existing KP cartridge functionalities.
24.1.0		Klarna Express Checkout: a new feature introduced in Storefront where Express Checkout Button will be displayed in PDP, Cart and Mini Cart and user will be redirected to Klarna after clicking Express Checkout button. Klarna Express Checkout enabled quick and easy checkout where Shipping Address, Billing Address and Payment details will

Version	Date	Changes
		<p>be preselected so that Checkout can be completed in fewer clicks.</p> <p>Fix for create order service calls with 500 status response: now orders are not created and error is thrown for these cases.</p>
24.2.0		<p>Update of OSM functionality that is more aligned with current web standards, provides a consistent identifier system as used in KEC, and offers enhanced customization options to our merchants. This will not only improve the user experience but also reinforce Klarna's commitment to providing versatile and state-of-the-art e-commerce solutions. CSS customizations are available only in the new library version.</p> <p>Documentation update - new section added for cartridge upgrade process.</p>
24.3.0		<p>Update of Klarna Express Checkout configuration. The display of Klarna Express Checkout buttons is now customizable by selecting preferred placements. By default, none is selected.</p> <p>Documentation update – new section added “Klarna Display Conditions and Authorization Handling”.</p>
25.4.0		<p>Added webhook subscription for KEC one-step checkout via PSP. A storefront site can now subscribe to webhook and receive notifications for subscribed payment events. These notifications are stored in a custom object for further processing.</p> <p>Documentation update – Updated new custom objects ‘KlarnaSigningKey’ and ‘KlarnaWebhookNotification’. Updated services section in which API related to webhook is added. Added a new section for jobs, CreateWebhook and DeleteWebhookNotificaiton.</p>

9. Additional Information

9.1. Klarna API Information

The Klarna Payments API is accessible through different endpoint based on the context of the webstore. There are separate endpoints for testing and live and the Klarna merchant identifier (MID) is configured for respective markets in regions (EU, NA, OC) by endpoint.

9.1.1. Live Environment

The API for the European production environment can be found at

- <https://api.klarna.com/>

The API for the North America production environment can be found at

- <https://api-na.klarna.com/>

The API for the Oceania production environment can be found at

- <https://api-oc.klarna.com/>

9.1.2. Testing Environment

The API for the European Playground/testing environment can be found at

- <https://api.playground.klarna.com/>

The API for the North America Playground/testing environment can be found at

- <https://api-na.playground.klarna.com/>

The API for the Oceania Playground/testing environment can be found at

- <https://api-oc.playground.klarna.com/>

9.2. Generate Key Pair and Key Id for Virtual Card Settlements (VCN)

The recommend RSA keypair size of 4096 bits. This key pair must be associated with a key_id (UUIDv4). The public key must be shared in JWK format with Klarna contact. Note that for production and playground, the key_id and keypair combination shared are different and must be configured prior to testing/go-live of the virtual card product.

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-----  
MIICIJANBhkqkG9w0BAQEFAOCAg8AMIIICCgKCAgEAoNYG7l2G8nZa+22oBYZktV228lw3UE9WO4oxfknJtKEdHn84x55ULt8KQTh9NVtdeKC8nTfTgyvMt/GNCa18xuZV/lGYDftKt85hbV5EjOum+StAIufEXvlBX7nMOMc1KyWm9kp2kbqd88mFIX63KV94OoNEXcNatRDFYR+qz53+ifadDQtQ1sLVNSTdroCZDJ1+LxtBy9V+BdmsBK1ERLsKh/JLXyWE24FJKV+z00s7TQkdWW/5ET12OGQYZsWo1yqgi9HplNvrisve8vWPxaL4m8iZ3I/9yYdg7yANQbTxSJccbRCgaaagPo30CNxeqU6qafY5g8vY3E52CoXHDdO4UsIX1qcuYIDhqaDzey6W+b8m755xLi+rqQyM4PBWL0J0dM3FVid8+4YKILex3AKBFciqRCMHSGaEeyrXKTjlAsghr9RS8PifvQRrl440cHzqw2vX0DvpjSWcmUJtW4wUq5RNSsobrxnVmoV6fj1z67Q/1P+l5Ie+oowdahR5ztVqJlO+2PNoX4I5VDs/Pkz3f8wWVc3Mp2oNT244o+/NIiyRfPFaJJx7JAgrcvZt2nFAmY4QApXLFJCpgEMwYucE4AH4gJKsh3KZbxRERrrO72bL2rxvWqBp/0h7DcMsV9sQs4BvxxIl6CF506FThzmclaKLBAyd5LALiXiPfkCAwEAAQ==-----END PUBLIC KEY-----
```

private_key.pem

```
-----BEGIN PRIVATE KEY-----  
MIIJQQIBADANBhkqkG9w0BAQEFAASCCSswggknAgEAAoICAQCg1gbuXYbydlrbagFhmS1XbbyXDdQT1Y7ijF+Scm0oR0efzjHnlQu3wpBOH01W114oLydN9ODK8y38Y0JrXzG5lX+UZgN+0q3zmFtXkSM66b5K0Ai58Re+UFfucw4xzUrJab2SnaRup3zyYUhfrfpX3g6g0Rdw1q1EMVhH6rPnf6J9p0NC1DWyVU1K12ugJkMnX4vG0HL1X4F
```

2awErUREuwqH8ktfJYTbgUkpX7PTSztNCR1Zb/kRPXY4ZBhmxajXKqCL0emU2+uK
y97y9Y/FovibyJncj/3Jh2DvIA1BtPFIlxttEKBppqA+jfQI3F6pTqpp9jmDy9jc
TnYKhccN07hSyVfWpy5ggOGpoPN7Lpb5vybvnnEuL6upDIzg8FYvQnR0zcVWJ3z7
hgogt7HcAoEVyKpEIwdl4ZoR7KtcpOOUCyCGv1FLw+J+9BGsvjjRwfOrDa9fQO+m
NJZyZQm1bjBSrlE1KyhuvGdWahXp+PXPrD/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/nIcgQdqbIbmFO2zEs0TOe6y4uBMndRsbQprTNSMUDBkrA
lNaYVSTQ1Z0Y/8DZDpGcyS1OnJv74F15uDjKN6/ov991mZ1JrZ+V2sdS3EDUlmvP
6thQKwI7Ln6h+ApHtWUG1NmVQe5gJE0qAeJ9b45clUzIRUwhVmEp8NoIJh0kAjaN
d4lk7xy9ZRDUY5yekPeYrJPShjsHAyEoktJIjRufI2UUq3uxNjjICoQcOVGfNDIS
YTTPwpu1pmC0C+rh2fgBAoIBAQDRultRArvtc2JKhVOUyZk88zd9kvrI6fNiyKmi
HgiWf7qkTPD9xhOQWDw3iwRFQAD+YkgV5MCBO8wp8oO8GEsOCI+XZWEExOcPT0
Vfj

PZHiQrTFnlfG/+fAO14xLf3j3ED4YQXdHOKI3xoLknQx/EydLoctxgkkpgWLrsA7
DwdSAg1/0sBvaHY27ogAfdimHdaKZ5OAe4a9k1qP3xVZBuOe8Sd65unBavUJLDuv
ikeNmkSVgW1sm55/729Jlr63USHF76It+vE1cdZ+vKg5vYotsQgPzvNBmUO/E8Gj
zMXQRfqfvEDINXEX0rCupTkw1G6AGTwQc/NPzyr/LTpLe6UBAoIBAQDEUjTiG11V
hf7WjdG3gctRlr+mYapQHgXdVLx2QSaqUYid+0QXK11YfJlsRB6nwa+OED83RfP0
IIFqxpzudSLPmoDuIBT7Dl5c/aleyKs/siUusP8QVDXk6OAR84XSytC35sIRV7pE
VMuBL91jfklQ0Lf/PreslK/kI6Yvwwp4qrHK6/f9TgciHcIYtf+/oti4ky6GJgfmP
fmuCqjxmUKbXXFPd5RbL2THGOowlb8zDLjf3RlbjlQFqogAk6H9hp2V0VZLiJH

```

UWM3z3zxDWeDaqJ08sHuk/rA9QpsVTu8IGTQsxdj8JwluN1Q+YZiOuPiSENBqPzT
V3exexzo3sD5AoIBAGU3qEyPojz1+9D1SaI8LW2CABzlq4z9g84ABAZOslxX5q7W
x1PinZyDSQRXg1B13jt29ZdIR79ygnQlg1YOBjcvtgVQHPuafk3RIBQbbCh+vaI
9dn/tUxMGqhnunKaby1rovJHfdqnPpKwzNAjYUqaGkJ822xhmmke/fEyAanIPa4
stDRvIPEWPTLx5xcOCdx13khkPSnkgRvaLEfpwkVX7Vr7hK/2OSFaYTNmrzXYBQ7
c6D/9d3Oo4nLb/mu+Tq67S19t53Qg/GEgTfkpuRoVPi0KyhUnKKCGWlBMZLTwyIG
S9eTFDKoJ0cSTGipjW7bPua93wZ8eEbRABpf4QECggEANNhQBeEJ0aCdBVHtdrEI
crDaa8X0W1aJi5dol4hYCRajaKsfHAF/QfdgMQVxHwUC5YG4En/Q+DAVWhGWYpXD
RhC3zeFy5FVszyk0sx/fAOIKGvRn5BRW4YRR9GMRzbjsT+RcruBnckdE9ERXGpX9
c/JB3rxZBIt+oIiFM8yfWKtMwsrmNKtFuDftv.Jeok4KejycFF4eWDqsf828xjPT+
xA/FP4CQD1UqkcpmuFSIwAwXo6LXVY7NTS0nKMiUnTLkL1TIHtLnO9+9jmNapWRP
Tc+hZUuHKlpI8DHFmX2j87LgkFD05eD5lynY4RgZtU1W1C1RdVYwoA72WB7knEaB
uQKCAQAH9s67P/7fFX9dfEans3PHU4nGjD8dJ8eoNQ6DhBMydZpGW15ZUeEBZDRk
0cBOeRs5BOcS43Em9kETpzawyCwxmnwzl+CzoPzMQcTw9tXomF9HG6RJ9XBdJfGA
ALAwCd4bASxmFM6guSP5GKnZ9aY3tR3tWWDFr7f9z8wOewzzpPclwRh009fPe4TC
NXoEm1MELJVeUieDSLKZgjgCw8WHGqlItONpA0/fwSM2gIcxETVV7qx3aPuJzCVh
LQZoBLQk3UMKsWDdpzeBdiERe66NAgV92Xe7SY9EY2vymaq761i1x1vlprT27qp
240LDJawqM0IraKmdCvWjofWSaOU

```

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

9.3. Decrypt VCN Card Details

To decrypt the virtual card details stored on order level and authorize the credit card processor you can use the following code snippet. You can find more information about the decryption process [here](#).

```

var OrderMgr = require( 'dw/order/OrderMgr' );
var Cipher = require( 'dw/crypto/Cipher' );
var Encoding = require( 'dw/crypto/Encoding' );
var Site = require( 'dw/system/Site' );

```

```

var Order = OrderMgr.getOrder( "order_id" );
var VCNPrivateKey = Site.getCurrent().getCustomPreferenceValue( 'vcnPrivateKey' );
var cipher = new Cipher();

var keyEncryptedBase64 = Order.custom.kpVCNAESKey;
var keyEncryptedBytes = Encoding.fromBase64( keyEncryptedBase64 );
var keyDecrypted = cipher.decryptBytes( keyEncryptedBytes, VCNPrivateKey,
"RSA/ECB/PKCS1PADDING", null, 0 );
var keyDecryptedBase64 = Encoding.toBase64( keyDecrypted );
var cardDataEncryptedBase64 = Order.custom.kpVCNPCIData;
var cardDataEncryptedBytes = Encoding.fromBase64( cardDataEncryptedBase64 );
var cardDecrypted = cipher.decryptBytes( cardDataEncryptedBytes, keyDecryptedBase64,
"AES/CTR/NoPadding", Order.custom.kpVCNIV, 0 );

var cardDecryptedUtf8 = decodeURIComponent( cardDecrypted );
var cardObj = JSON.parse( cardDecryptedUtf8 );
var expiryDateArr = cardObj.expiry_date.split( "/" );

// Retrieve encrypted card details
var cardPAN = cardObj.pan, cardCVV = cardObj.cvv,
cardExpiryMonth = expiryDateArr[0], cardExpiryYear = expiryDateArr[1];

```

9.4. Update KlarnaCountries Definition

With version 21.2.0 of the cartridge, the KlarnaCountries custom object definition changed to non-replicable. If you are using an earlier version, to mitigate any issues that may be present in your environments, please follow these steps to update the definitions:

1. Back-up KlarnaCountries configurations per country for each site as changing the definition will remove all configurations on your environment!

- a. For merchant with one site - Go to "**Merchant Tools > Custom Objects > Import & Export**", select KlarnaCountries & export the data.
 - b. For merchant with multiple sites using Klarna – You can export the configurations via "**Admin > Site Development > Site Import & Export**" and select each site that uses Klarna to be included in the export zip file. This section requires Account Manager access for users and will export all custom objects, not just Klarna ones.
2. Export custom object definitions from "**Admin > Site Development > Import & Export**". This action will export all custom object definitions that you have.
 3. Update KlarnaCountries.xml definitions in file exported in step 2 and set "`<staging-mode>source-to-target</staging-mode>`" to "`<staging-mode>no-staging</staging-mode>`"
 4. Import the updated KlarnaCountries.xml definition in "**Admin > Site Development > Import & Export**".
 5. Import the KlarnaCountries configs exported in step 1 (manually or via bulk import)
 - a. If you've followed step 1.b with multiple sites, you may want to edit the zip file and remove everything else apart from KlarnaCountries.xml