Klarna Payments for SFRA

Version 19.1.6

Table of Contents

1. Overview

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 based on site configurations.

The cartridge makes use of the Klarna Payments JSON REST API and a JavaScript SDK.

The integration can be tested using Klarna playground environment. For production (live) deployment contraction agreements and sign-off are required. Klarna provides dedicated support for integration and go-live, please contact the Key Account Manager to further details

Key features:

- Integrate Klarna Payments using best practices on international sites based on SFRA (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
- 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
- SFRA Multi Shipping Address support
- Enable Onsite Messaging placements on PDP and Cart

1.1 Enable Klarna Payments across international sites (NA, EU, OC)

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

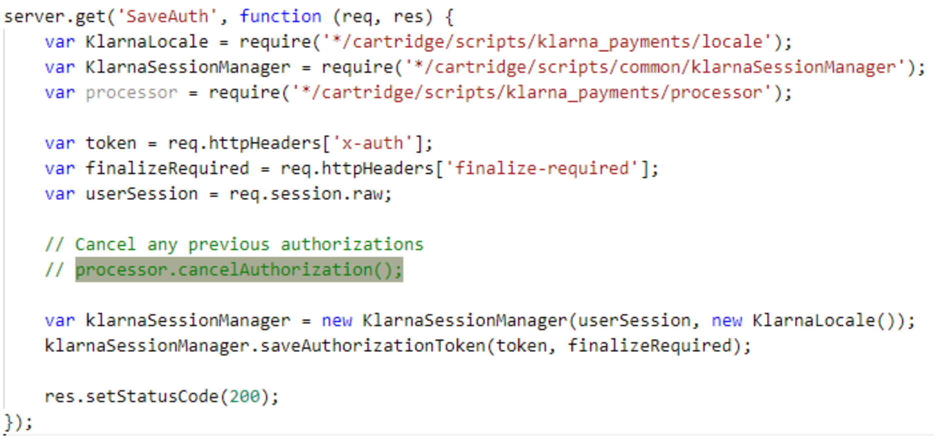
1.2 Authorize/Place Klarna order in SFRA single-page checkout

Successful payment authorizations (Klarna Payments status: APPROVED) followed by placement of order lead to a 'Paid' order payment status and ‘Ready for Export’ export status. Refused and pending payment authorizations (Klarna Payments statuses REJECTED and PENDING) lead to a 'Not Paid' order payment status, and ‘Not Exported’ export status.

Cartridge provides best practice implementation and options to include Extra merchant data (EMD) to optimize acceptance rate for Klarna payment products. The EMD data sent should be reviewed case by case and optimized and validated based on merchant data and privacy requirements prior to go-live

Cancelled orders to a 'Not Paid' order payment status (even if the status was 'Paid' before), and ‘Not Exported’ export status.

Merchants have the option to utilize the cancelAuthorization(disabled by default) to delete prior authorization when required for specific use-cases. If enabled the checkout flow should be tested thoroughly as part of integration, considering the valid checkout scenarios (relevant Klarna order flow, Klarna Payment method changes, order amount updates or checkout with external payment method)

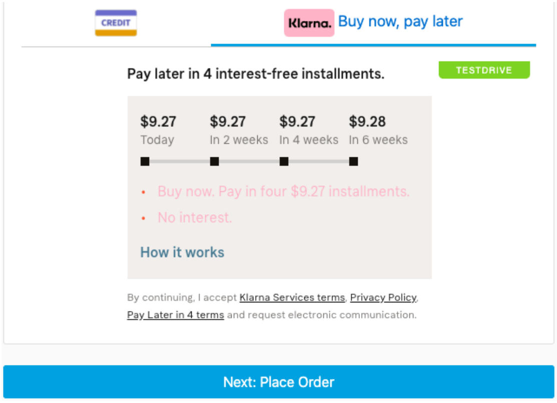


```
server.get('SaveAuth', function (req, res) {
    var KlarnaLocale = require('*/cartridge/scripts/klarna_payments/locale');
    var KlarnaSessionManager = require('*/cartridge/scripts/common/klarnaSessionManager');
    var processor = require('*/cartridge/scripts/klarna_payments/processor');

    var token = req.httpHeaders['x-auth'];
    var finalizeRequired = req.httpHeaders['finalize-required'];
    var userSession = req.session.raw;

    // Cancel any previous authorizations
    // processor.cancelAuthorization();

    var klarnaSessionManager = new KlarnaSessionManager(userSession, new KlarnaLocale());
    klarnaSessionManager.saveAuthorizationToken(token, finalizeRequired);

    res.setStatusCode(200);
});
```

1.3 Multitude of payment options for Customers

On the checkout billing page, configured Klarna payment options are dynamically loaded based on customer cart information. The customer information (personally identifiable information) is sent once customer clicks on a Klarna payment method to display the widget. The additional information facilitates assessment and verification of customer data to display payments method options available for the customer

The screen below shows an example of the payment options displayed in the Checkout billing page – Pay Later, Pay over time:

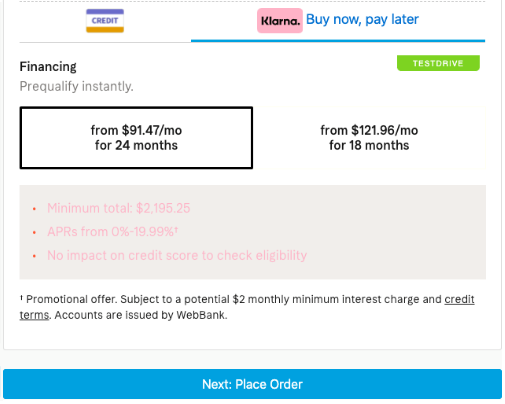


Pay later in 4 interest-free installments.

$9.27 Today
$9.27 In 2 weeks
$9.27 In 4 weeks
$9.28 In 6 weeks

- Buy now. Pay in four $9.27 installments.
- No interest.

By continuing, I accept Klarna Services terms, Privacy Policy, Pay Later in 4 terms and request electronic communication.



Financing
Prequalify instantly.

from $91.47/mo for 24 months
from $121.96/mo for 18 months

- Minimum total: $2,195.25
- APRs from 0%-19.99%†
- No impact on credit score to check eligibility

† Promotional offer. Subject to a potential $2 monthly minimum interest charge and credit terms. Accounts are issued by WebBank.

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

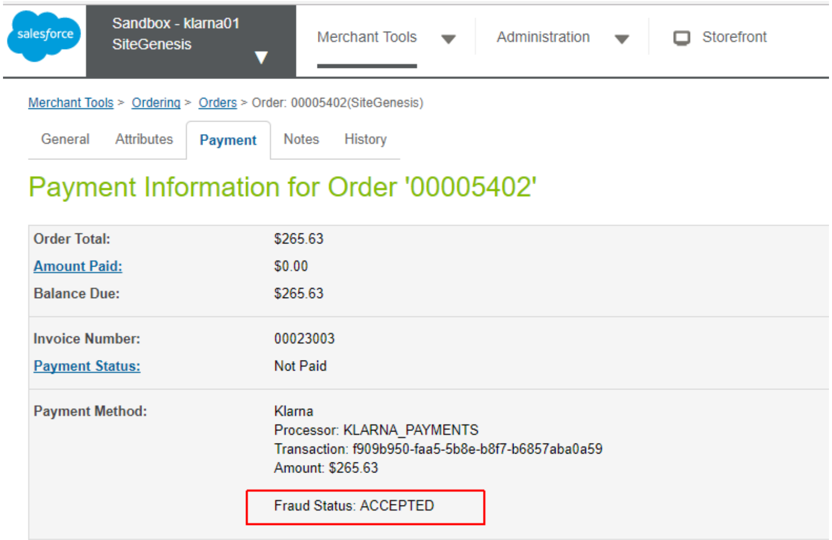
Note: The payment methods displayed are based on market and contractual agreement with Klarna

1.4 Handling notifications

In scenarios were the Klarna Order has been created but the order is placed on hold if suspected of fraud. This results in Commerce Cloud order to stay in created status with Fraud Status: PENDING

After verification, Klarna may send a notification to SFCC to update risk status with one of following event types: FRAUD_RISK_ACCEPTED, FRAUD_RISK_REJECTED, FRAUD_RISK_STOPPED

The notification updates are 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:



Payment Information for Order '00005402'

Order Total:	$265.63
Amount Paid:	$0.00
Balance Due:	$265.63
Invoice Number:	00023003
Payment Status:	Not Paid
Payment Method:	Klarna Processor: KLARNA_PAYMENTS Transaction: f909b950-faa5-5b8e-b8f7-b6857aba0a59 Amount: $265.63 Fraud Status: ACCEPTED

If the order is ACCEPTED upon notification, the order will be placed in SFCC (order status changes to OPEN).

If the order is REJECTED or STOPPED upon notification, the order is failed (FAIL) in SFCC.

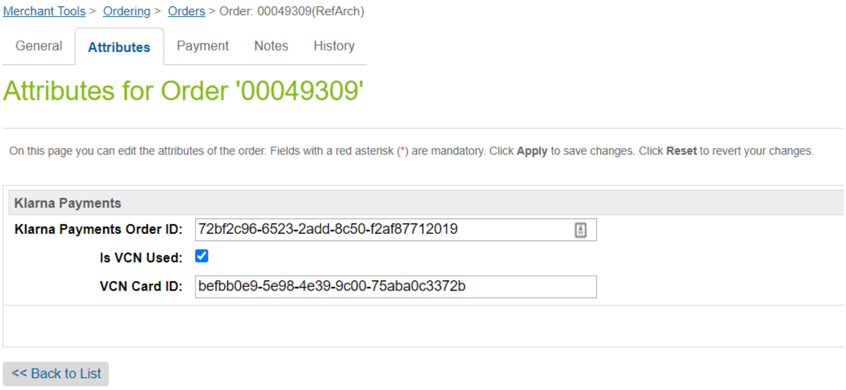
1.5 Creation of VCN Settlements for each order

If standard order management is not a reasonable option for a Klarna integration, then Klarna's Merchant Card Service based virtual card solution may be utilized. When a customer places an order, the order is first booked in SFCC. If the 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 ACCEPTED. Once an order has been accepted by Klarna, the merchant platform creates a VCN settlement, per the merchant card services (MCS) API.

Once a settlement has been created, the merchant platform can authorize the virtual card until the Klarna order is valid. Then, once the order has been fulfilled, the card funds should be captured. (For delays in capture, or other special use cases, please speak with the Klarna key account manager in advance) While Klarna is the original payment method of the order, the order will be settled with a credit card instead of direct bank account transfer. You can find more information here around other use cases.

Enable virtual card settlement for each order using custom preference within the Business Manager. A predefined Key ID corresponding to the relevant private key needs to be configured. This option is disabled by default. If enabled, virtual card settlement request is made. A virtual card settlement request triggered for New order or those orders with pending status resolved to FRAUD_RISK_ACCEPTED

An order placed with the VCN settlement turned on is going to have custom attributes related to the VCN settlement as show below:


Merchant Tools > Ordering > Orders > Order: 00049309(RefArch)
General
Attributes
Payment
Notes
History
<< Back to List

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.

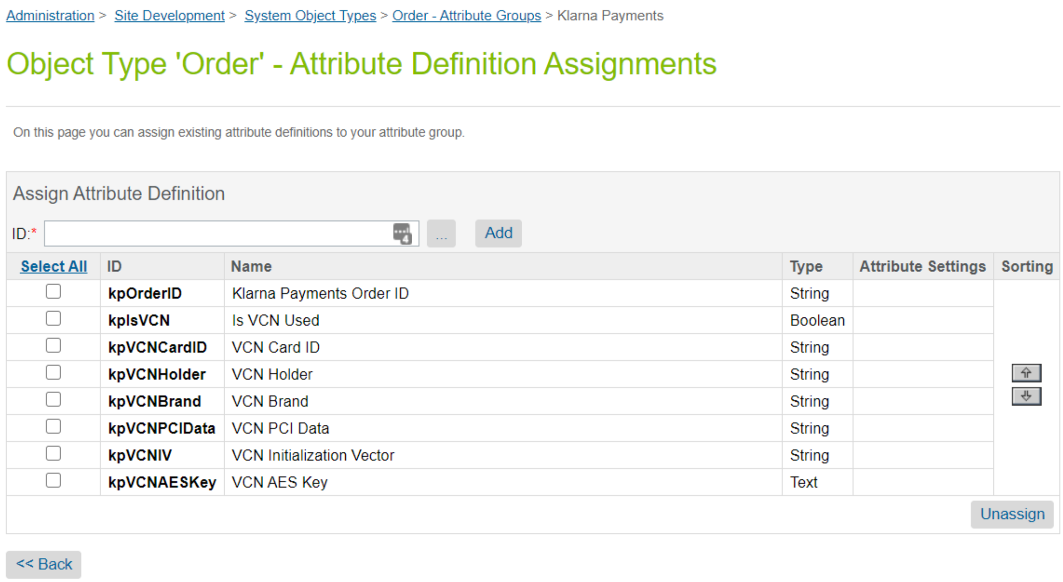
Klarna Payments

Klarna Payments Order ID: 72bf2c96-6523-2add-8c50-f2af87712019

Is VCN Used: ☑

VCN Card ID: befbb0e9-5e98-4e39-9c00-75aba0c3372b

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:*

Select All	ID	Name	Type	Attribute Settings	Sorting
☐	**kpOrderID**	Klarna Payments Order ID	String		
☐	**kpIsVCN**	Is VCN Used	Boolean		
☐	**kpVCNCardID**	VCN Card ID	String		
☐	**kpVCNHolder**	VCN Holder	String		
☐	**kpVCNBrand**	VCN Brand	String		
☐	**kpVCNPCIData**	VCN PCI Data	String		
☐	**kpVCNIV**	VCN Initialization Vector	String		
☐	**kpVCNAESKey**	VCN AES Key	Text		

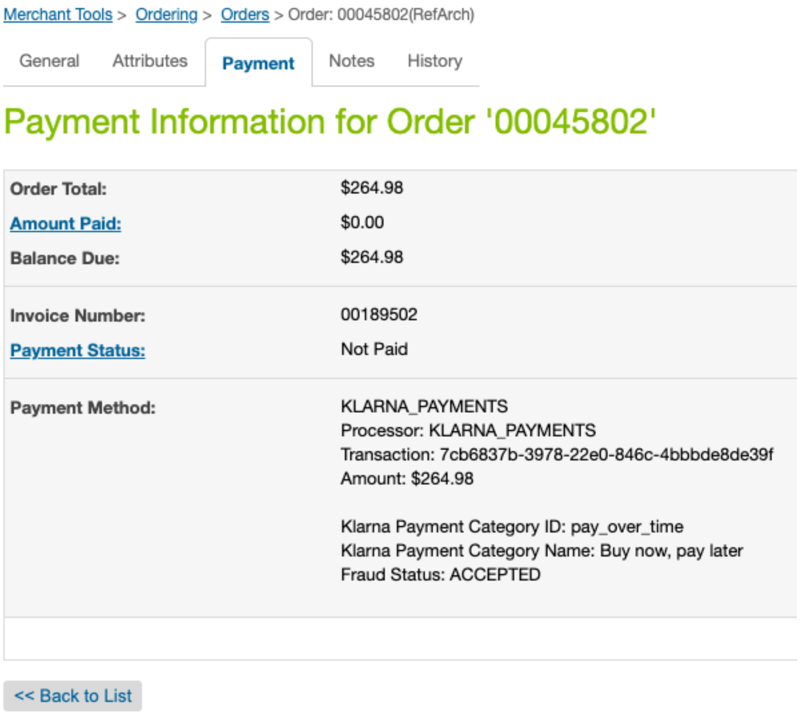
Please work with Klarna Account Manager and Delivery contact to select the appropriate virtual card product based on your business requirements and use-cases.

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.

1.6 Auto-capture

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

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

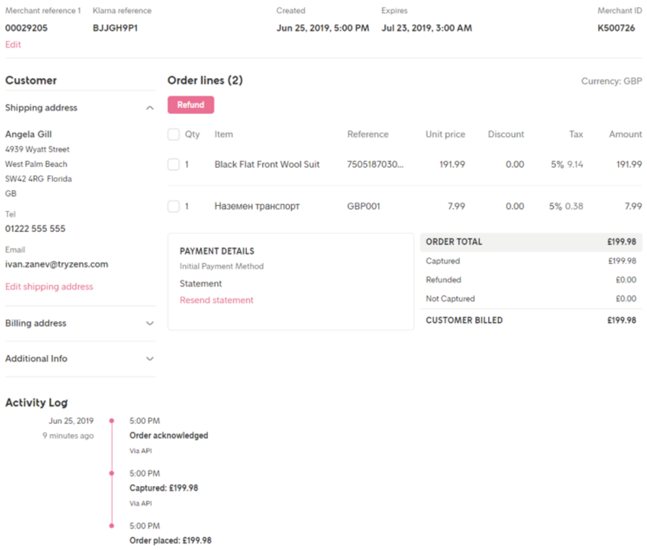


Merchant Tools > Ordering > Orders > Order: 00045802(RefArch)
General
Attributes
Payment
Notes
History

Payment Information for Order '00045802'

Order Total:	$264.98
Amount Paid:	$0.00
Balance Due:	$264.98
Invoice Number:	00189502
Payment Status:	Not Paid
Payment Method:	KLARNA_PAYMENTS Processor: KLARNA_PAYMENTS Transaction: 7cb6837b-3978-22e0-846c-4bbbde8de39f Amount: $264.98 Klarna Payment Category ID: pay_over_time Klarna Payment Category Name: Buy now, pay later Fraud Status: ACCEPTED

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



Merchant reference 1	Klarna reference	Created	Expires	Merchant ID
00029205	BJJGH9P1	Jun 25, 2019, 5:00 PM	Jul 23, 2019, 3:00 AM	K500726

Angela Gill
4939 Wyatt Street
West Palm Beach
SW42 4RG Florida
GB

Tel
01222 555 555

Email
ivan.zanev@tryzens.com

Order lines (2)

Currency: GBP

Qty	Item	Reference	Unit price	Discount	Tax	Amount
□ 1	Black Flat Front Wool Suit	7505187030...	191.99	0.00	5% 9.14	191.99
□ 1	Наземен транспорт	GBP001	7.99	0.00	5% 0.38	7.99

PAYMENT DETAILS
Initial Payment Method
Statement

ORDER TOTAL	£199.98
Captured	£199.98
Refunded	£0.00
Not Captured	£0.00
CUSTOMER BILLED	£199.98

Activity Log

Jun 25, 2019
9 minutes ago

5:00 PM
Order acknowledged
Via API

5:00 PM
Captured: £199.98
Via API

5:00 PM
Order placed: £199.98

If the capture is unsuccessful, an error will be logged in the custom error log.

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

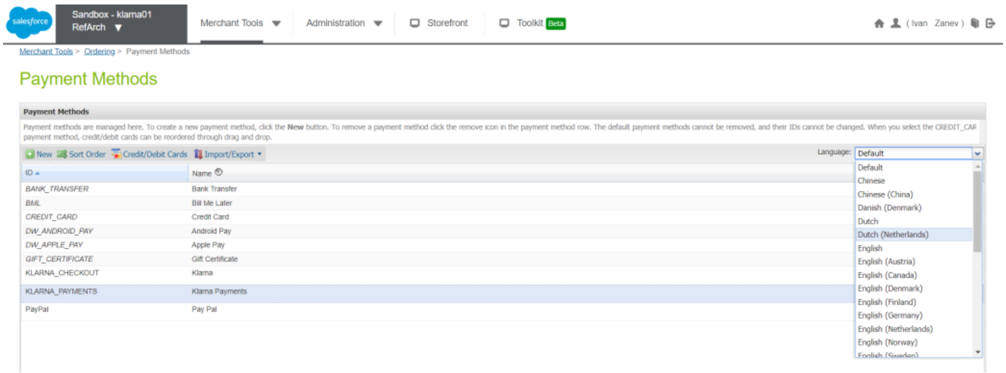
1.7 Widget customizations

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"

1.8 Customizing Payment method name

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



Merchant Tools > Ordering > Payment Methods

Payment Methods

Payment Methods

Payment methods are managed here. To create a new payment method, click the **New** button. To remove a payment method click the remove icon in the payment method row. The default payment methods cannot be removed, and their IDs cannot be changed. When you select the CREDIT_CAR payment method, credit/debit cards can be reordered through drag and drop.

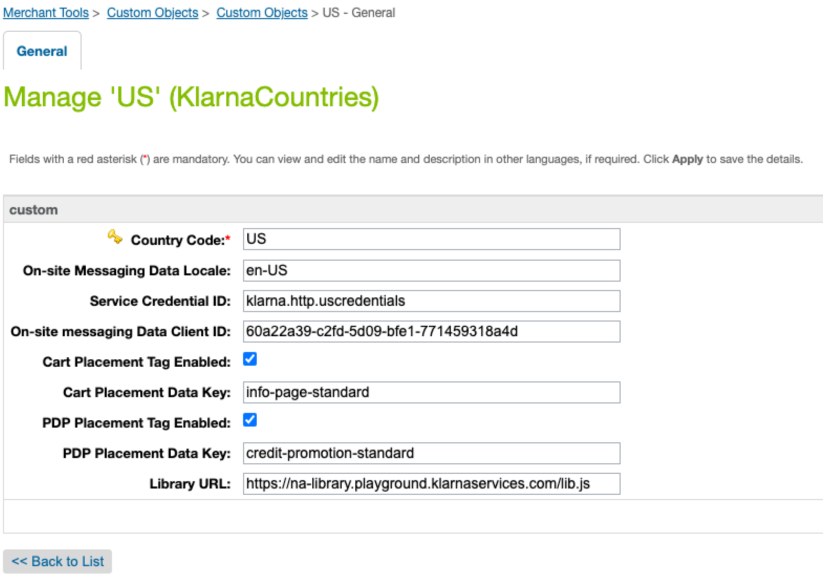
Language: Default

ID	Name
BANK_TRANSFER	Bank Transfer
BML	Bill Me Later
CREDIT_CARD	Credit Card
DW_ANDROID_PAY	Android Pay
DW_APPLE_PAY	Apple Pay
GIFT_CERTIFICATE	Gift Certificate
KLARNA_CHECKOUT	Klarna
KLARNA_PAYMENTS	Klarna Payments
PayPal	Pay Pal

The screenshot below shows the "*KLARNA_PAYMENTS*" method selected and the administrator choosing a language from the drop-down.

1.9 Configure Klarna On-Site Messaging

Klarna On-Site Messaging (OSM) is configured per-site per-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. 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 – On-site Messaging



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:*	US
On-site Messaging Data Locale:	en-US
Service Credential ID:	klarna.http.uscredentials
On-site messaging Data Client ID:	60a22a39-c2fd-5d09-bfe1-771459318a4d
Cart Placement Tag Enabled:	☑
Cart Placement Data Key:	info-page-standard
PDP Placement Tag Enabled:	☑
PDP Placement Data Key:	credit-promotion-standard
Library URL:	https://na-library.playground.klarnaservices.com/lib.js

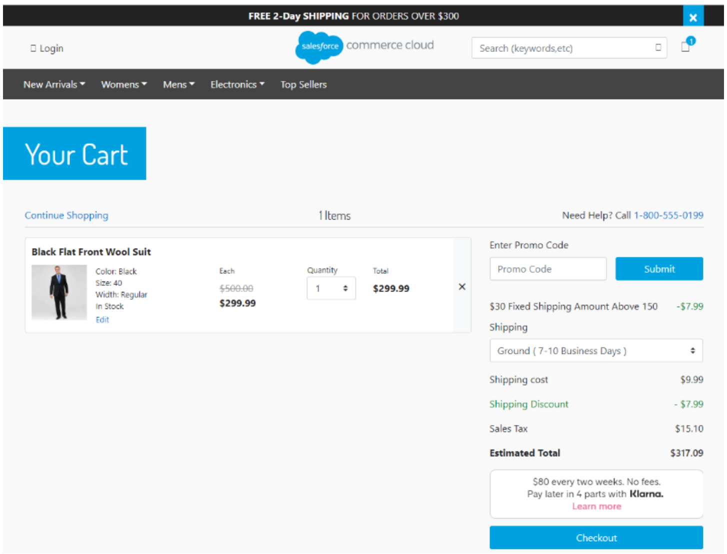
To enable Placement tag for the Cart Page, the "*Cart Placement Data Key*" must be filled with Data Key value and the "*Cart Placement Tag Enabled*" must be checked. Note: Cart placements amount must be updated and latest Klarna credit offering (where required) displayed to customer when order line quantity is updated on cart page. Refer: 3.8: cart.js

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

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

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



Your Cart

1 Items

Need Help? Call 1-800-555-0199

Black Flat Front Wool Suit

Color: Black
Size: 40
Width: Regular
In Stock

Each	Quantity	Total
~~$500.00~~ $299.99	1	$299.99

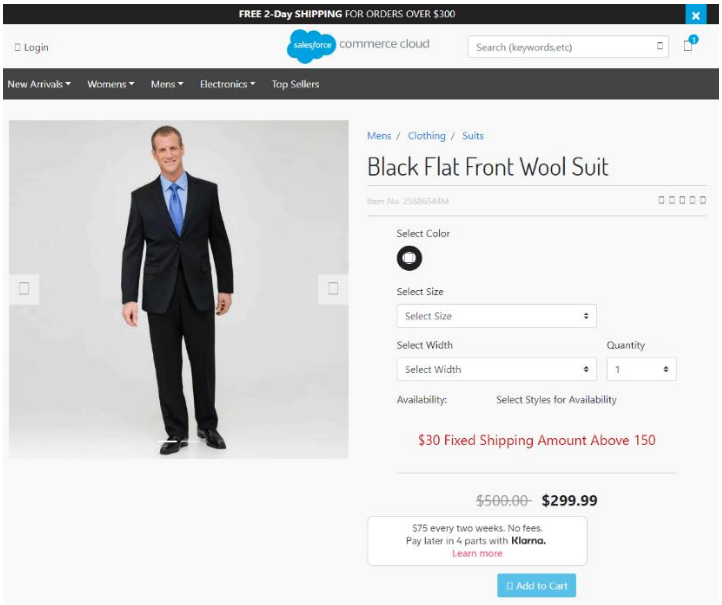
Enter Promo Code

Promo Code

$30 Fixed Shipping Amount Above 150	-$7.99
Shipping	
Ground (7-10 Business Days)	
Shipping cost	$9.99
Shipping Discount	- $7.99
Sales Tax	$15.10
Estimated Total	$317.09

$80 every two weeks. No fees.
Pay later in 4 parts with **Klarna.**

On-Site Messaging Enabled on Cart Page



Black Flat Front Wool Suit

Item No. 25686544M

Select Color

Select Size: Select Size

Select Width: Select Width

Quantity: 1

Availability: Select Styles for Availability

$30 Fixed Shipping Amount Above 150

~~$500.00~~ $299.99

$75 every two weeks. No fees.
Pay later in 4 parts with **Klarna.**

On-Site Messaging Enabled on PDP Page

For more information regarding customizations and best practices, please refer to the Klarna Developer: https://developers.klarna.com/documentation/on-site-messaging/customization/

2. System Extensions

This section describes all cartridge extensions (custom attributes to system objects, site preferences). Once you have installed the cartridge ("Installation" section), you can check and confirm the extensions have been applied to your system.

2.1 Compatibility

API Version: 20.7 (Compatibility Mode: 19.10)

SFRA version. 5.0.1

2.2 Extended Controllers

Controller	Start Node	Remarks
Checkout.js	Begin	Extended to call Klarna session manager
CheckoutServices.js	SubmitPayment	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

2.3 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

2.4 System object extensions

2.4.1 *Order*

Order system object has been extended with the following custom attributes:

Parameter Name	Attribute ID	Description
Klarna Payments Order ID	kpOrderID	The Klarna payments Order ID when Klarna payment method was selected by customer
VCN Brand	kpVCNBrand	Klarna Payments virtual card scheme name
VCN Holder	kpVCNHolder	Klarna Payments virtual card expiration year
VCN Card ID	kpVCNCardID	Holding the Klarna Payments Virtual Card Number Card ID
VCN PCI Data	kpVCNPCIData	Holding the Klarna Payments Virtual Card Number Encrypted Card Data
VCN Initialization Vector	kpVCNIV	Holding the Klarna Payments Virtual Card Number Initialization Vector
VCN AES Key	kpVCNAESKey	Holding the Klarna Payments Virtual Card Number AES Key
Is VCN Used	kpIsVCN	True if virtual card is used for payment of the order, otherwise false

Table 1. Order system object attributes

2.4.2 *PaymentTransaction*

PaymentTransaction system object has been extended with the following custom attributes:

Parameter Name	Attribute ID	Description
Fraud Status	kpFraudStatus	Klarna Payments order fraud status

Table 2. PaymentTransaction system object attributes.

2.4.3 *OrderPaymentInstrument*

OrderPaymentInstrument system object has been extended with the following custom attributes:

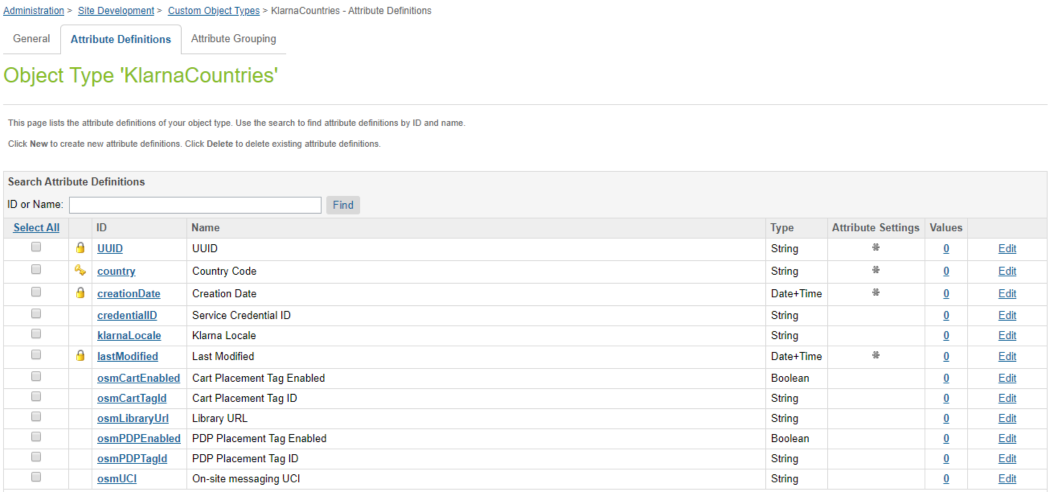
Parameter Name	Attribute ID	Description

Klarna Payment Category ID	klarnaPaymentCategoryID	Holds ID of Klarna payment category.
Klarna Payment Category Name	klarnaPaymentCategoryName	Holds name of Klarna payment category.

Table 3. OrderPaymentInstrument system object attributes.

2.5 Custom objects

2.5.1 *KlarnaCountries*



Administration > Site Development > Custom Object Types > KlarnaCountries - Attribute Definitions

General | **Attribute Definitions** | Attribute Grouping

Object Type 'KlarnaCountries'

This page lists the attribute definitions of your object type. Use the search to find attribute definitions by ID and name.

Click **New** to create new attribute definitions. Click **Delete** to delete existing attribute definitions.

Search Attribute Definitions

ID or Name: [ ] Find

Select All		ID	Name	Type	Attribute Settings	Values	
□	🔒	UUID	UUID	String	*	0	Edit
□	🔑	country	Country Code	String	*	0	Edit
□	🔒	creationDate	Creation Date	Date+Time	*	0	Edit
□		credentialID	Service Credential ID	String		0	Edit
□		klarnaLocale	Klarna Locale	String		0	Edit
□	🔒	lastModified	Last Modified	Date+Time	*	0	Edit
□		osmCartEnabled	Cart Placement Tag Enabled	Boolean		0	Edit
□		osmCartTagId	Cart Placement Tag ID	String		0	Edit
□		osmLibraryUrl	Library URL	String		0	Edit
□		osmPDPEnabled	PDP Placement Tag Enabled	Boolean		0	Edit
□		osmPDPTagId	PDP Placement Tag ID	String		0	Edit
□		osmUCI	On-site messaging UCI	String		0	Edit

The cartridge is selecting the object based on the request locale. E.g. site with locale “de_DE” will use the “DE” custom object. Even if you have locales that are not supported by Klarna Payments we recommend to make a corresponding entry in the custom object for that locale (you can use one of the other endpoints). Thus, on the billing page of the unsupported locale you will have the Klarna Payments widget showing an appropriate message.

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

Attribute Name	Attribute ID	Description
Country Code	country	Two-letter country code.
Klarna Locale	klarnaLocale	Fallback, if the site locale cant be used.
Service Credential ID	credentialID	The ID of service credentials for this locale.
On-Site Marketing UCI	osmUCI	UCI for Klarna on-site marketing for this locale.

Attribute Name	Attribute ID	Description
Cart Placement Tag Enabled	osmCartEnabled	Whether Cart Placement Tag is Enabled for this locale.
Cart Placement Tag ID	osmCartTagId	ID of On-site messaging placement tag for Cart Page for this locale.
PDP Placement Tag Enabled	osmPDPEnabled	Whether PDP Placement Tag is Enabled for this locale.
PDP Placement Tag ID	osmPDPTagId	ID of On-site messaging placement tag for PDP Page for this locale.
Library URL	osmLibraryUrl	URL to On-Site Messaging Library URL

Table 4. KlarnaCountries custom object attributes.

2.6 Site preferences

2.6.1 *Klarna_Payments group*

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	If enabled, a full capture will be attempted automatically prior to order acknowledge. If disabled, the merchant needs to manually capture the order amount from the KP dashboard.
Send product_url and image_url	sendProductAndImageURLs	If set to true, product_url and image_url fields in Klarna Payments create session call will be included in API call, otherwise product_url and image_url fields will not be populated.
Merchant Reference 2 Mapping	merchant_reference2_mapping	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 not always be as expected.
Border Color Preference	kpColorBorder	CSS hex color to be used in Klarna Payments iFrame
Border Selected Color Preference	kpColorBorderSelected	CSS hex color to be used in Klarna Payments iFrame
Button Color Preference	kpColorButton	CSS hex color to be used in Klarna Payments iFrame

Parameter Name	**Attribute ID**	**Description**
Button Text Color Preference	kpColorButtonText	CSS hex color to be used in Klarna Payments iFrame
Checkbox Color Preference	kpColorCheckbox	CSS hex color to be used in Klarna Payments iFrame
Checkbox Checkmark Color Preference	kpColorCheckboxCheckmark	CSS hex color to be used in Klarna Payments iFrame
Details Color Preference	kpColorDetails	CSS hex color to be used in Klarna Payments iFrame
Header Color Preference	kpColorHeader	CSS hex color to be used in Klarna Payments iFrame
Link Color Preference	kpColorLink	CSS hex color to be used in Klarna Payments iFrame
Text Color Preference	kpColorText	CSS hex color to be used in Klarna Payments iFrame
Secondary Text Color Preference	kpColorTextSecondary	CSS hex color to be used in Klarna Payments iFrame
Border Radius Preference	kpRadiusBorder	Size (in pixels) of the border radius to be used in Klarna Payments iFrame
Attachments	kpAttachments	Flag to swicth on/off the using of attachments when creating a session. Default is OFF.

Not available message on billing page	kpNotAvailableMessage	The Klarna Payment not available message on billing page. JSON string holding country code and corresponding message string.
		For example:
		`{`
		`   "GB":"Klarna Payment not available",`
		`   "default":"Klarna Payment not available"`
		`}`

Virtual Card Number Enabled	kpVCNEnabled	If this option is set to TRUE, SFCC will create a Virtual Card Number settlement for every Klarna order. Note: the option will only work if VCN private/public keys are configured

Parameter Name	Attribute ID	Description
		properly (see vcnPrivateKey and vcnPublicKey below).
VCN Public Key ID	kpVCNkeyId	UUIDv4 value corresponding to the key pair. Shared with Klarna.
VCN Private Key	vcnPrivateKey	SSL private key is used by SFCC to decode Virtual Card information (used with kpVCNEnabled).
VCN Public Key	vcnPublicKey	Shared with Klarna and stored here for reference.

2.7 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).

3. Installation

Installation of this cartridge is a process consisting of the following steps:

1. Verify package contents.
2. Upload the cartridge to the currently active code version on your environment.
3. Import metadata.
4. Configure Klarna integration settings (preferences).
5. Configure Klarna service and credentials.
6. Add the cartridge to the cartridge path.
7. Test the SFRA front-end and make sure Klarna payment options appear correctly.

Note that this section describes the minimum installation process without going in specifics on how to customize the cartridge. By the end of the section, you should have a working Klarna Payments SFRA cartridge.

3.1 Verify Package Contents

Klarna Payments SFRA package contains the following items:

- Cartridge called "*Klarna Payments SFRA*" (or simply "*the cartridge*").
- A *site-template* archive containing attributes and settings required for the cartridge to work properly.
- This document.

3.2 Cartridge upload

Make sure the "*int_klarna_payments_sfra*" cartridge is uploaded to the active code version of your environment.

3.3 Metadata import

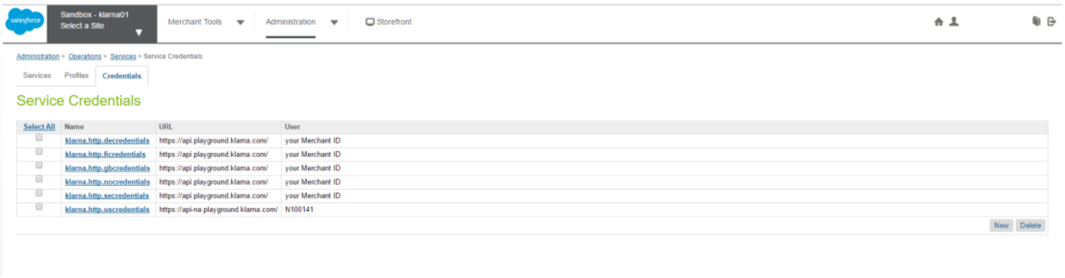
Go to ‘metadata’ folder, review and edit, if needed, the site-template contents. (Site template is prepared to setup SiteGenesis and RefArch sites - you may want to change that to your actual sites and delete the ones that are not needed). Zip the directory and you’ll have ‘site-template.zip’ installation package. Import it through BM Administration > Site Development > Site Import & Export section..

Right after metadata import, you should verify all extensions are available within the business manager as described in Section 2. “System Extensions”.

3.4 Service configuration

You need to configure Klarna Payments services credentials.

Go to Credentials tab as shown below:

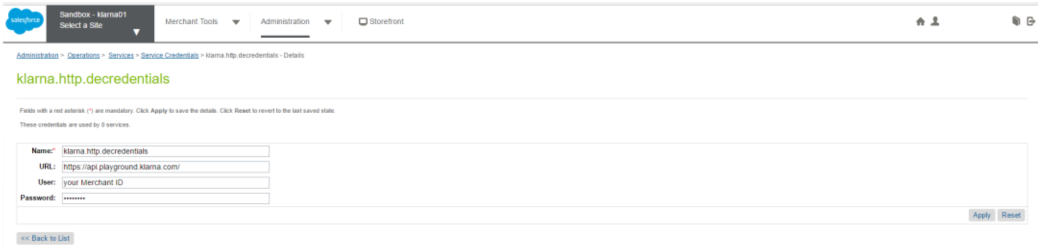


Administration > Operations > Services > Service Credentials

Service Credentials

Select All	Name	URL	User
□	klarna.http.decredentials	https://api.playground.klarna.com/	your Merchant ID
□	klarna.http.ficredentials	https://api.playground.klarna.com/	your Merchant ID
□	klarna.http.gbcredentials	https://api.playground.klarna.com/	your Merchant ID
□	klarna.http.nocredentials	https://api.playground.klarna.com/	your Merchant ID
□	klarna.http.secredentials	https://api.playground.klarna.com/	your Merchant ID
□	klarna.http.uscredentials	https://api-na.playground.klarna.com/	N100141

You need to edit the existing credentials or create new ones using the credentials provided to you by Klarna and the API URL for the specific location (see “Klarna API Information” for the API URLs) as shown below:



Administration > Operations > Services > Service Credentials > klarna.http.decredentials - Details

klarna.http.decredentials

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.decredentials
URL: https://api.playground.klarna.com/
User: your Merchant ID
Password: ••••••••

3.5 Add cartridge to cartridge path

To activate the cartridge, make sure you prepend “*int_klarna_payments_sfra*” to your cartridge path. This cartridge should be prepended to the “*app_storefront_base*” cartridge so it can override base templates and controllers.

3.6 (optional) Enable Virtual Card Settlements

In order to enable Virtual Credit settlements (VC settlements, in short), you need to do the following:

1. Generate an SSL keypair (see “**Error! Reference source not found.**”).
2. Input the keys from the generated keypair in the site preferences, *vcnPrivateKey* and *vcnPublicKey* respectively.
3. Send the generated public key/Key_ID(UUIDv4) to Klarna in JWK format for testing/golive. It will be used to encrypt the virtual card pan and csc on Klarna side. Wait for confirmation from Klarna that the key has been successfully added to merchant (MID/Market) profile.(Different for playground & Production)
4. Enable the VCN site preference “*kpVCNEnabled*”.
5. Enter the public/Private keys & key_ID provided from step 3 in “*kpVCNkeyId/* vcnPrivateKey / *vcnPublicKey*” site preference.
6. Use a credit-card processor of your choice. By default, this cartridge uses basic_credit to authorize credit-card settlements.

In order 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.

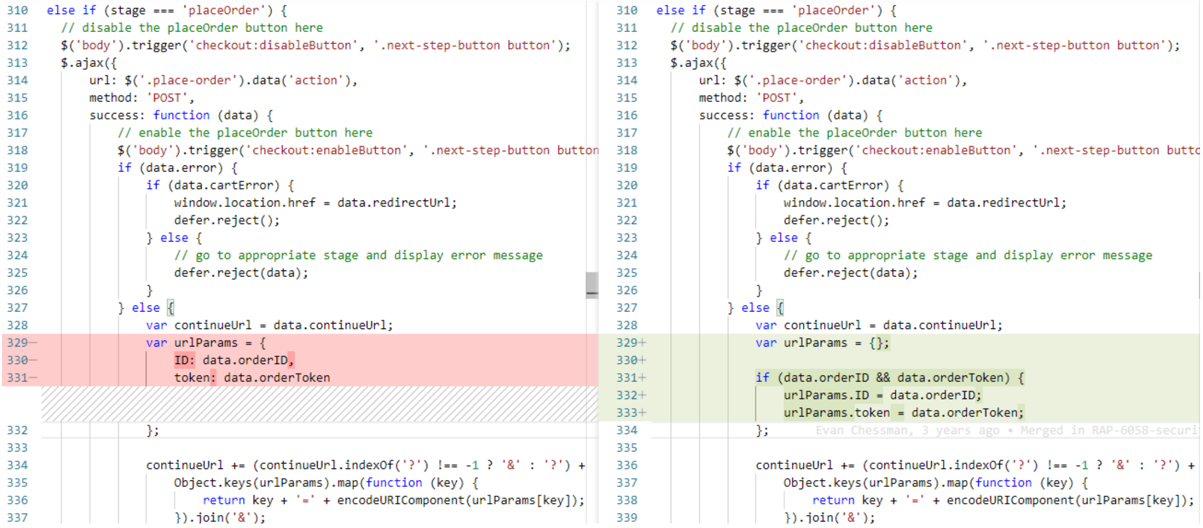
```
var OrderMgr = require( 'dw/order/OrderMgr' );
var Cipher = require( 'dw/crypto/Cipher' );
var Encoding = require( 'dw/crypto/Encoding' );
var Site = require( 'dw/system/Site' );

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 ecnrypted card details
var cardPAN = cardObj.pan, cardCVV = cardObj.cvv,
    cardExpiryMonth = expiryDateArr[0], cardExpiryYear =
expiryDateArr[1];
```

3.7 (optional) checkout.js

1. app_storefront_base\cartridge\client\default\js\checkout\checkout.js

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



```
else if (stage === 'placeOrder') {
  // disable the placeOrder button here
  $('body').trigger('checkout:disableButton', '.next-step-button button');
  $.ajax({
      url: $('.place-order').data('action'),
      method: 'POST',
      success: function (data) {
          // enable the placeOrder button here
          $('body').trigger('checkout:enableButton', '.next-step-button button
          if (data.error) {
              if (data.cartError) {
                  window.location.href = data.redirectUrl;
                  defer.reject();
              } else {
                  // go to appropriate stage and display error message
                  defer.reject(data);
              }
          } else {
              var continueUrl = data.continueUrl;
              var urlParams = {
                  ID: data.orderID,
                  token: data.orderToken
              };

              continueUrl += (continueUrl.indexOf('?') !== -1 ? '&' : '?') +
                  Object.keys(urlParams).map(function (key) {
                      return key + '=' + encodeURIComponent(urlParams[key]);
                  }).join('&');
```

```
else if (stage === 'placeOrder') {
  // disable the placeOrder button here
  $('body').trigger('checkout:disableButton', '.next-step-button button');
  $.ajax({
      url: $('.place-order').data('action'),
      method: 'POST',
      success: function (data) {
          // enable the placeOrder button here
          $('body').trigger('checkout:enableButton', '.next-step-button butto
          if (data.error) {
              if (data.cartError) {
                  window.location.href = data.redirectUrl;
                  defer.reject();
              } else {
                  // go to appropriate stage and display error message
                  defer.reject(data);
              }
          } else {
              var continueUrl = data.continueUrl;
              var urlParams = {};

              if (data.orderID && data.orderToken) {
                  urlParams.ID = data.orderID;
                  urlParams.token = data.orderToken;
              };      Evan Chessman, 3 years ago • Merged in RAP-6058-securi

              continueUrl += (continueUrl.indexOf('?') !== -1 ? '&' : '?') +
                  Object.keys(urlParams).map(function (key) {
                      return key + '=' + encodeURIComponent(urlParams[key]);
                  }).join('&');
```

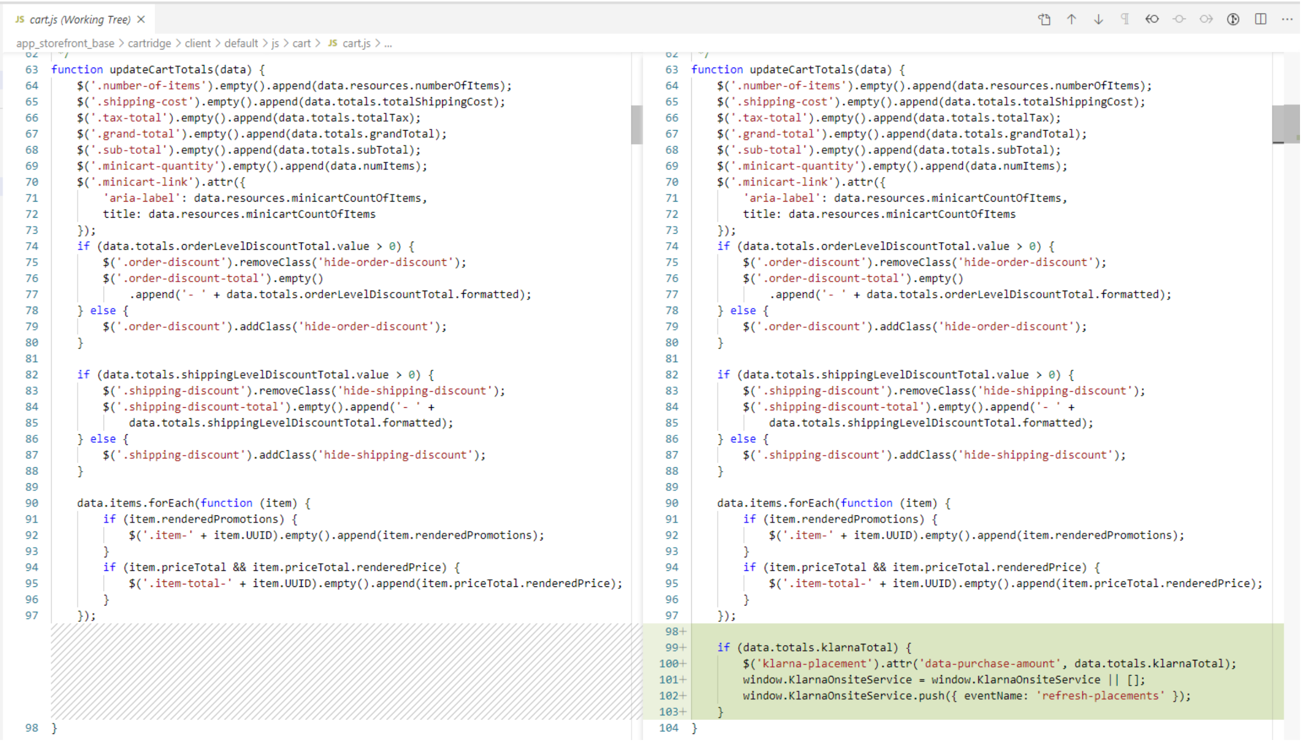
3.8 (required) cart.js

1. app_storefront_base\cartridge\client\default\js\cart\cart.js

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);
    window.KlarnaOnsiteService = window.KlarnaOnsiteService || [];
    window.KlarnaOnsiteService.push({ eventName: 'refresh-placements' });
}
```


cart.js (Working Tree)
app_storefront_base > cartridge > client > default > js > cart > cart.js > ...

```
function updateCartTotals(data) {
    $('.number-of-items').empty().append(data.resources.numberOfItems);
    $('.shipping-cost').empty().append(data.totals.totalShippingCost);
    $('.tax-total').empty().append(data.totals.totalTax);
    $('.grand-total').empty().append(data.totals.grandTotal);
    $('.sub-total').empty().append(data.totals.subTotal);
    $('.minicart-quantity').empty().append(data.numItems);
    $('.minicart-link').attr({
        'aria-label': data.resources.minicartCountOfItems,
        title: data.resources.minicartCountOfItems
    });
    if (data.totals.orderLevelDiscountTotal.value > 0) {
        $('.order-discount').removeClass('hide-order-discount');
        $('.order-discount-total').empty()
            .append('- ' + data.totals.orderLevelDiscountTotal.formatted);
    } else {
        $('.order-discount').addClass('hide-order-discount');
    }

    if (data.totals.shippingLevelDiscountTotal.value > 0) {
        $('.shipping-discount').removeClass('hide-shipping-discount');
        $('.shipping-discount-total').empty().append('- ' +
            data.totals.shippingLevelDiscountTotal.formatted);
    } else {
        $('.shipping-discount').addClass('hide-shipping-discount');
    }

    data.items.forEach(function (item) {
        if (item.renderedPromotions) {
            $('.item-' + item.UUID).empty().append(item.renderedPromotions);
        }
        if (item.priceTotal && item.priceTotal.renderedPrice) {
            $('.item-total-' + item.UUID).empty().append(item.priceTotal.renderedPrice);
        }
    });
}
```

```
function updateCartTotals(data) {
    $('.number-of-items').empty().append(data.resources.numberOfItems);
    $('.shipping-cost').empty().append(data.totals.totalShippingCost);
    $('.tax-total').empty().append(data.totals.totalTax);
    $('.grand-total').empty().append(data.totals.grandTotal);
    $('.sub-total').empty().append(data.totals.subTotal);
    $('.minicart-quantity').empty().append(data.numItems);
    $('.minicart-link').attr({
        'aria-label': data.resources.minicartCountOfItems,
        title: data.resources.minicartCountOfItems
    });
    if (data.totals.orderLevelDiscountTotal.value > 0) {
        $('.order-discount').removeClass('hide-order-discount');
        $('.order-discount-total').empty()
            .append('- ' + data.totals.orderLevelDiscountTotal.formatted);
    } else {
        $('.order-discount').addClass('hide-order-discount');
    }

    if (data.totals.shippingLevelDiscountTotal.value > 0) {
        $('.shipping-discount').removeClass('hide-shipping-discount');
        $('.shipping-discount-total').empty().append('- ' +
            data.totals.shippingLevelDiscountTotal.formatted);
    } else {
        $('.shipping-discount').addClass('hide-shipping-discount');
    }

    data.items.forEach(function (item) {
        if (item.renderedPromotions) {
            $('.item-' + item.UUID).empty().append(item.renderedPromotions);
        }
        if (item.priceTotal && item.priceTotal.renderedPrice) {
            $('.item-total-' + item.UUID).empty().append(item.priceTotal.renderedPrice);
        }
    });

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

4. 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'}">

        <isinclude 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.

5. Uninstallation

Uninstallation of the cartridge is:

1. Remove cartridge “*int_klarna_payments_sfra*” from the cartridge path.
2. Remove cartridge code from active code version.
3. Remove cartridge-related metadata from the website.

6. Known Issues

None

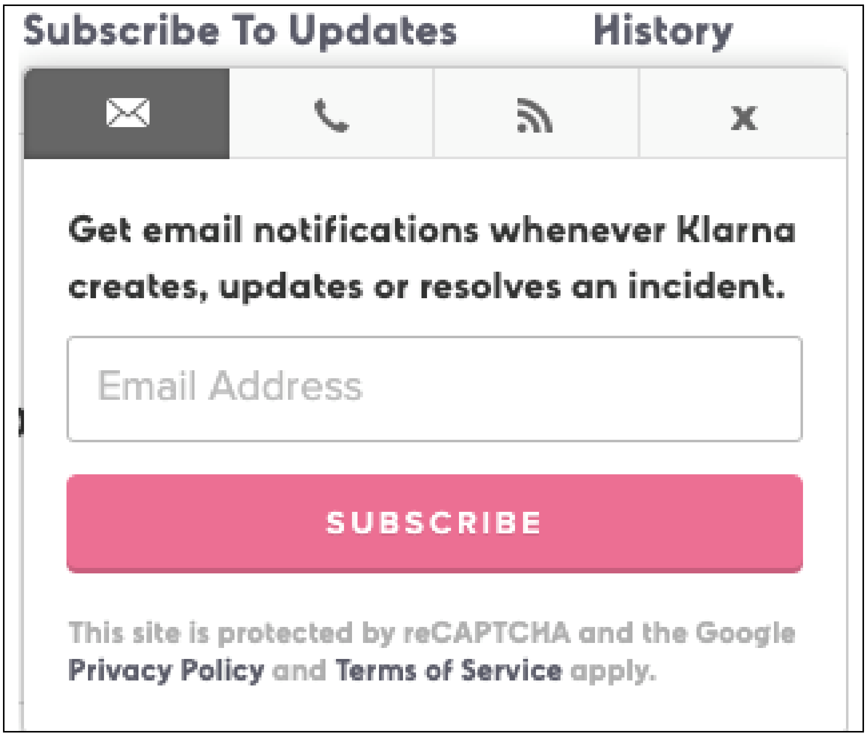
7. Support from Klarna

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

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

8. Klarna Status Page

Klarna provides an external facing page, https://status.klarna.com/ ,where merchant can refer the health of the Klarna system. The page also provides the opportunity to subscribe to specific ongoing incident by clicking on the issue subscribe button (via SMS/emails)



Get email notifications whenever Klarna creates, updates or resolves an incident.

Email Address

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

9. Reporting an Incident

Merchant representative should reach their Klarna Account manager to report a Production incident (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 advice 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

10.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

11. Additional Information

11.1 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.

11.1.1 *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/

11.1.2 *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/

11.2 Generate Key pair and Key_id for virtual card settlements

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

In order 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

In order to extract the public key from an RSA keypair, you can use the following *openssl* command:

openssl rsa -pubout -in private_key.pem -out public_key.pem

In the folder where you have executed the above commands two new files will be created - public_key.pem and private_key.pem.

The contents of the files should look something like:

public_key.pem

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

MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAoNYG7l2G8nZa+22oBYZk

tV228lw3UE9WO4oxfknJtKEdHn84x55ULt8KQTh9NVtdeKC8nTfTgyvMt/GNCa18

xuZV/lGYDftKt85hbV5EjOum+StAIufEXvlBX7nMOMc1KyWm9kp2kbqd88mFIX63

KV94OoNEXcNatRDFYR+qz53+ifadDQtQ1slVNStdroCZDJ1+LxtBy9V+BdmsBK1E

RLsKh/JLXyWE24FJKV+z00s7TQkdWW/5ET12OGQYZsWo1yqgi9HplNvrisve8vWP

xaL4m8iZ3I/9yYdg7yANQbTxSJcbbRCgaaagPo30CNxeqU6qafY5g8vY3E52CoXH

DdO4UslX1qcuYIDhqaDzey6W+b8m755xLi+rqQyM4PBWL0J0dM3FVid8+4YKILex

3AKBFciqRCMHSOGaEeyrXKTjlAsghr9RS8PifvQRrL440cHzqw2vX0DvpjSWcmUJ

tW4wUq5RNSsobrxnVmoV6fj1z67Q/1P+l5Ie+oowdahR5ztVqJlO+2PNoX4I5VDs

/Pkz3f8wWVc3Mp2oNT244o+/NIiyRfPFaJJx7JAgrcvZt2nFAmY4QApXLFJCpgEM

wYucE4AH4gJKsh3KZbxRERrrO72bL2rxvWqBp/0h7DcMsV9sQs4BvxxIl6CF506F

ThzmclaKLBAyd5LALiXiPfkCAwEAAQ==

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

private_key.pem

```
-----BEGIN PRIVATE KEY-----
MIIJQQIBADANBgkqhkiG9w0BAQEFAASCCSswggknAgEAAoICAQCg1gbuXYbydlr7
bagFhmS1XbbyXDdQT1Y7ijF+Scm0oR0efzjHnlQu3wpBOH01W114oLydN9ODK8y3
8Y0JrXzG5lX+UZgN+0q3zmFtXkSM66b5K0Ai58Re+UFfucw4xzUrJab2SnaRup3z
yYUhfrcpX3g6g0Rdw1q1EMVhH6rPnf6J9p0NC1DWyVU1K12ugJkMnX4vG0HL1X4F
2awErUREuwqH8ktfJYTbgUkpX7PTSztNCR1Zb/kRPXY4ZBhmxajXKqCL0emU2+uK
y97y9Y/FovibyJncj/3Jh2DvIA1BtPFIlxttEKBppqA+jfQI3F6pTqpp9jmDy9jc
TnYKhccN07hSyVfWpy5ggOGpoPN7Lpb5vybvnnEuL6upDIzg8FYvQnR0zcVWJ3z7
hgogt7HcAoEVyKpEIwdI4ZoR7KtcpOOUCyCGv1FLw+J+9BGsvjjRwfOrDa9fQO+m
NJZyZQm1bjBSrlE1KyhuvGdWahXp+PXPrtD/U/6Xkh76ijB1qFHnO1WomU77Y82h
fgjlUOz8+TPd/zBZVzcynag1Pbjij780iLJF88VoknHskCCty9m3acUCZjhAClcs
UkKmAQzBi5wTgAfiAkqyHcplvFERGus7vZsvavG9aoGn/SHsNwyxX2xCzgG/HEiX
oIXnToVOHOZyVoosEDJ3ksAuJeI9+QIDAQABAoICACRkaUsUNI22RB3yEPu3DiCP
pO6v+QAeA4gTW+GUdqR9dCZLaSCZ7bhxVVOuoX4qPzslO6hjUmOyzG6upFgVPk+P
HNQfyEUZoC148Eib9OziAXUN2URMpv1KbwVm+BO814X8zguai7uru0PHTG1oy677
4Ct1OknxAxxHQDIaxT6XJFo5SA4EinUfNz2Bo3/xry/QjxW/mCK0GwDd4PNp9TGM
FPTv2SgdSDOWzGQlOH5N3owuzMpI8NV6z74wv+i5Ptv41Dzu8WhyXpiYSsk00SRK
HPC68j2bAzTPghp5aSZ9976SGm2SPonJXyboXdiHbI/osdyqDxeIT3iB9GmrHX/i
kHPGJCh7fRZvqj39Hc+IxYjabwW3rDeDIPB7ab9z1KLF4z1D6AZOKCPyTaDRdQ1Q
eDi7LwDmk7NHEPrmF/nIcguQdqbIbmFO2zEs0TOe6y4uBMndRsbQprTNSMUdBkrA
lNaYVSTQ1Z0Y/8DZDpGcyS1OnJv74F15uDjKN6/ov991mZ1JrZ+V2sdS3EDUlmvP
6thQKwI7Ln6h+ApHtWUG1NmvQe5gJE0qAeJ9b45clUzIRUwhVmEp8NoIJh0kAjaN
d4lk7xy9ZRDUY5yekPeYrJPShjsHAyEoktJIjRufI2UUq3uxNjjICoQcOVGfNDIS
YTTPwpu1pmC0C+rh2fgBAoIBAQDRultRArvtc2JKhVOUyZk88zd9kvrI6fNiyKmi
HgiWf7qkTPD9xhOQWDw3iwRFQAD+YkgV5MCBO8wp8oO8GEsOCI+XZWExOcPT0Vfj
```

```
PZHiQrTFnlfG/+fAO14xLf3j3ED4YQXdHOKI3xoLknQx/EydLoctxgkkpgWLrsA7
DwdSAg1/0sBvaHY27ogAfdimHdaKZ5OAe4a9k1qP3xVZBuOe8Sd65unBavUJLDuv
ikeNmkSVgW1sm55/729JIr63USHF76It+vE1cdZ+vKg5vYotsQgPzvNBmUO/E8Gj
zMXQRfqfvEDlNXEX0rCupTkw1G6AGTwQc/NPzyr/LTpLe6UBAoIBAQDEUjTiG11V
hf7WjdG3gctRlr+mYapQHgXdVLx2QSaqUYid+0QXK11YfJlsRB6nwa+OED83RfP0
lIFqxpzudSLPmoDuIBT7Dl5c/aleyKs/siUusP8QVDXk6OAR84XSytC35sIRV7pE
VMuBL91jfkQ0Lf/PreslK/kI6Yvwwp4qrHK6/f9TgciHclYtf+/oti4ky6GJgfmP
fmuCqjxmUKbXXFPd5RbL2THGOowilb8zDLjf3RlbjlQFqogAk6H9hp2V0VZLiJHp
UWM3z3zxDWeDaqJ08sHuk/rA9QpsVTu8IGTQsxdj8JwluN1Q+YZiOuPiSENBqPzT
V3exexzo3sD5AoIBAGU3qEyPojz1+9D1SaI8LW2CABzlq4z9g84ABAZOslxX5q7W
x1PinZyDSQSRXg1B13jt29ZdIR79ygnQlg1YOBjcvtgVQHPuafk3RlBQbbCh+vaI
9dn/tUxMGqhnhunKaby1rovJHfdqnPpKwzNAjYUqaGkJ822xhmmke/fEyAanIPa4
stDRvIPEWPTLx5xcOCdx13khpKSnkgRvaLEfpwkVX7Vr7hK/2OSFaYTNmrzXYBQ7
c6D/9d3Oo4nLb/mu+Tq67S19t53Qg/GEgTfkpuRoVPi0KyhUnKKCGWlBMZLTwyIG
S9eTFDKoJ0cSTGipjW7bPua93wZ8eEbRABpf4QECggEANNhQBeEJ0aCdBVHtdrEI
crDaa8X0W1aJi5dol4hYCRajaKsfHAF/QfdgMQVxHwUC5YG4En/Q+DAVWhGWYpXD
RhC3zeFy5FVszyk0sx/fAOlKGvRn5BRW4YRR9GMRzbjsT+RcruBnckdE9ERXGpX9
c/JB3rxZBIt+oIiFM8yfWKtMwsrmNKtFuDftvJeok4KejycFF4eWDqsf828xjPT+
xA/FP4CQD1UqkcpmuFSIwAwXo6LXVY7NTS0nKMiUnTLkLlTIHtLnO9+9jmNapWRP
Tc+hZUuHKlpI8DHFmX2j87LgkFD05eD5lynY4RgZtU1W1C1RdVYwoA72WB7knEaB
uQKCAQAH9s67P/7fFX9dfEans3PHU4nGjD8dJ8eoNQ6DhBMydZpGWI5ZUeEBZDRk
0cBOeRs5BOcS43Em9kETpzawyCwxmnwzl+CzoPzMQcTw9tXomF9HG6RJ9XBdJfGA
ALAwCd4bASxmFM6guSP5GKnZ9aY3tR3tWWDfr7f9z8wOewzzpPclwRh009fPe4TC
NXoEm1MELJVeUieDSLKZgjgCw8WHGqLItONpA0/fwSM2gIcxETVV7qx3aPuJzCVh
LQZoBLQk3UMKsWDdpzeBdiERe66NAgVk92Xe7SY9EY2vymaq761i1x1vlprT27qp
240LDJawqM0IraKmdCvWjofWSaOU
-----END PRIVATE KEY-----
```