|  |
| --- |
| CRED Salesforce Payment module |
| User Guide Version 22.1.0  Wednesday, July 20, 2022 |

Overview:  
The Jifiti Payment Cartridge - B2C Commerce and Quick Launch is a full payment solution for your Salesforce Commerce Cloud ecommerce site. The merchant can present an additional payment method in the checkout page that will allow the customer to pre-qualify for financing with the lender and then complete their purchase.

Introduction:  
The purpose of this manual is to guide the user through the configuration settings and the general use of the Payment Module Extension for the Salesforce ecommerce platform. Then you will be able to review/modify the related configuration in the Custom preferences and start using the new payment in the checkout page - storefront.

[Compatibility](#_heading=h.tyjcwt):

Our custom payment cartridge can be used with Storefront Reference Architecture (SFRA) version 5.3.0 or later.  
The compatibility mode (21.2).

[Locales](#_heading=h.tyjcwt):  
 list of all locales this cartridge is available: for now en\_US.  
  
Installation steps:  
Here we will guide you through the installation process as well as the configuration of our extension as following:  
Install the cartridge:

1. get “int\_custompayment\_sfra” cartridge from the Marketplace and add it to your project in cartridges folder.
2. Add the changes that applied in webbackconfig.js and stylelintrc.json, package.json, .eslintignore, .eslintrc.json files from the payment cartridge, also add custompaymenttest  
   folder as well
3. Install Node modules

From your project's root directory, install Node modules using your command line: npm install

1. In BM from Administration > Sites > Manage Sites choose the required site and  
    add the payment cartridge “int\_custompayment\_sfra” to cartridge path.
2. Now navigate to the custom payment cartridge folder specially in “metadata” folder and do the following:  
   To add new configuration items, import the predefined metadata:

- From the cartridge/metadata/metadata.zip: Open the metadata folder.

- Rename the yourSiteId folder to the ID of your site in the Business Manager.

- Edit jobs file to the ID of your site in the Business Manager.

<context site-id="yourSiteId"/>

- Zip the site\_import folder.

- In the Business Manager, go to Administration > Site Development > Site Import & Export and import the zipped file.

- After the import, attributes and groups named JifitiPayment are added to:

a- Administration > Site Development > System Object Types > Site Preferences > Attribute Definitions

b- Administration > Site Development > System Object Types > Order > Attribute Definitions

c- Administration > Site Development > System Object Types > OrderPaymentInstrument > Attribute Definitions

d- Administration > Site Development > System Object Types > PaymentMethod > Attribute Definitions

e- The following services will be added to Administration > Operations > Services:

credPayment.rest, credPaymentEU.rest

f- The custom object type “lenderDetails” is added to Administration > Site Development > Custom Object Types

g- The job “GetLenderDetails” is added to Administration > Operations > Jobs

Configurations:  
**Custom Preferences configuration:**

Navigate to Merchant Tools > Site Preferences > Custom Site Preference Groups and click on the following groups and fill the below configurations:

P.S. for any update to take effect, you must click save -you can find it on the top right of the page-.

1. **JifitiPayment group:**

a – credPaymentToken  
b- LenderId  
c- MerchantId  
d- storeId  
e- callBackURL: this URL should be the URL of call back method “Callback” on “CredPaymentController” controller as example : yoursiteBaseURL/CredPaymentController-Callback  
f- Completion Method: Indication if the flow should end in displaying the card to the customer, returning the card (or other payment token) back to the hosting website (default if not sent), completing the purchase internally, or just creating the credit account.

You can set the Completion Method to be one of the following values:

* 1. None
  2. DisplayCard
  3. TransferCard (Selected By Default)
  4. CompletePurchase
  5. CreateAccount

Please note the cartridge logic only implemented to handle the value of “TransferCard” in “Completion Method” at system configuration. Choosing another value will make the flow not work as expected.

g- Allowed Product Types: Determine the financial product type, you can select multi values of the following:CO**,** LC, NaN, ZeroPing, Funded, RevolvingInstore, RevolvingOnline, OTLInstore, OTLOnline, \_3xCB

h- Window Behavior: Determines how the new flow opens, and you can choose one of these values: NewTab, SameTab or Lightbox  
**Note**: New Tab is not supported for now.

i - sourcePageType: it should be a “Checkout” value   
j- SourceChannel: should be “Online” value  
k- paymentTransactionType: indicates to status of payment (authorize, capture) this is the logic that we applied for each status:

Authorize: in SFCC if the status set as authorize then we will call the purchase API with the InstantCommit flag as false. And the capture call of the API should be called from OMS or any other third party.

Capture: in SFCC if the status set as capture, then we will call the purchase API with InstantCommit flag as true

l- checkAccountStatusApi: applications/v2/CheckAccountStatus

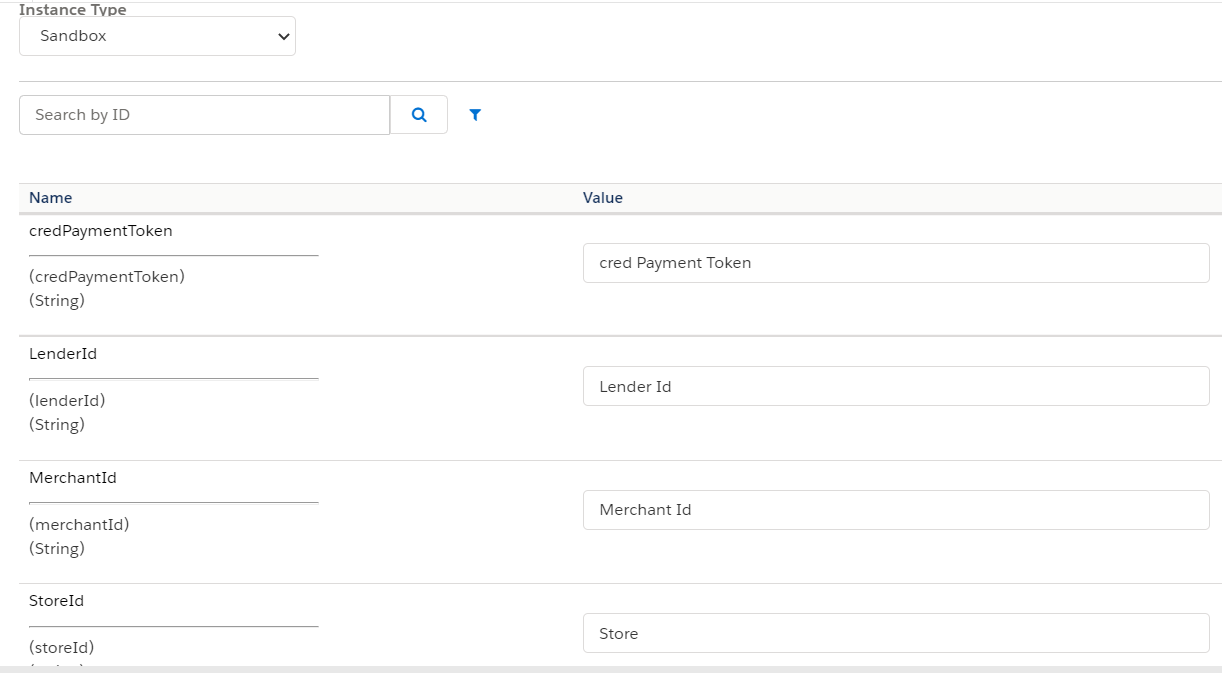
m- PurchaseApi: purchases/v2/Authorize

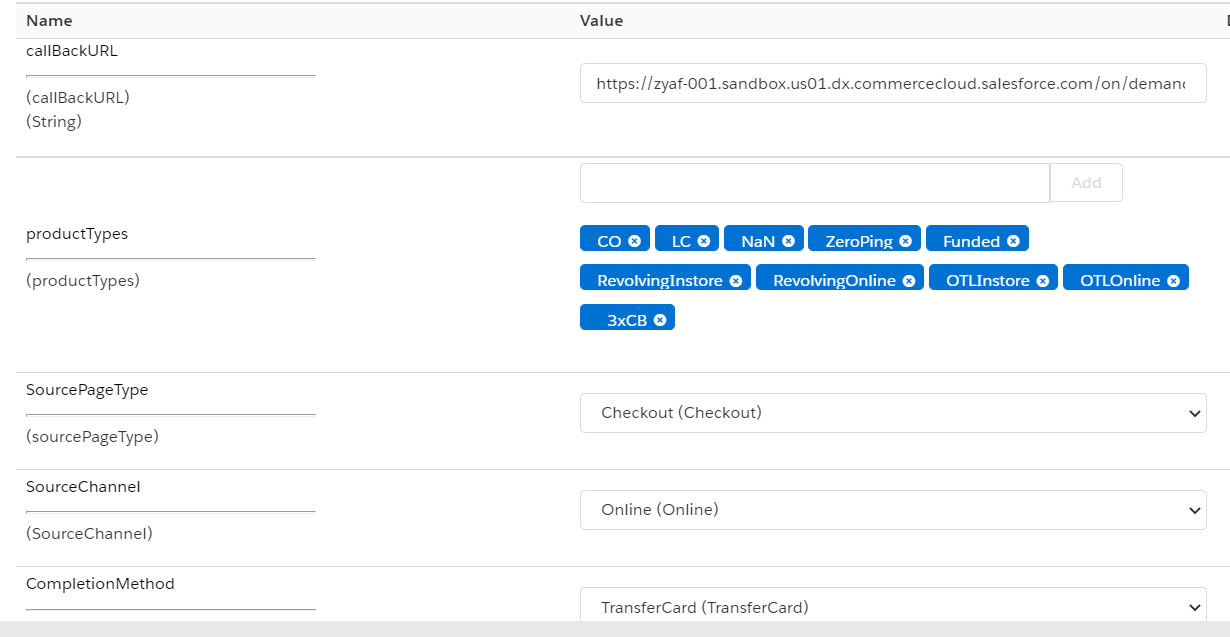
n- initiateFlowApi: applications/v2/InitiateFlow

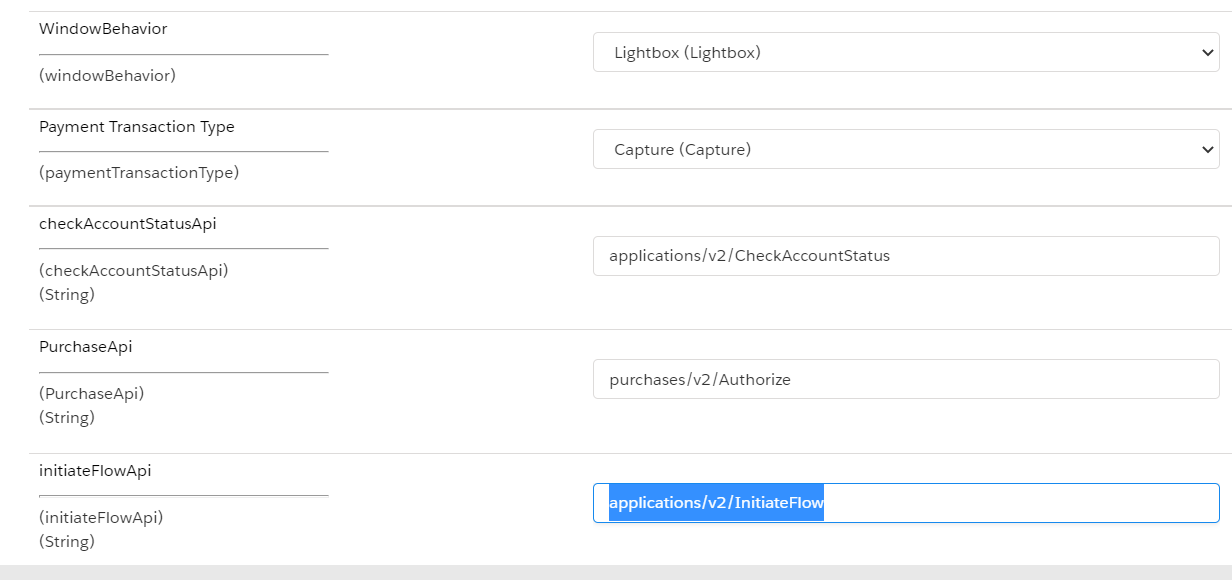
o- isJifitiEnabled: To enable or disable Jifiti payment

p- JifitiServiceLocation: This custom Attribute used to decide which is the service that will be   
 used either US API URL's or the EU ones.

q- getLenderDetailsApi: applications/v2/LenderBranding/GetLenderDetails

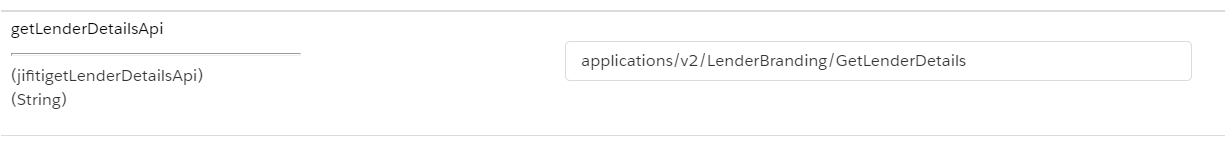
**Note**: Please don’t change the values for checkAccountStatusApi, initiateFlowApi, PurchaseApi, getLenderDetailsApi unless you communicated with the payment provider.  






Graphical user interface, application

Description automatically generated

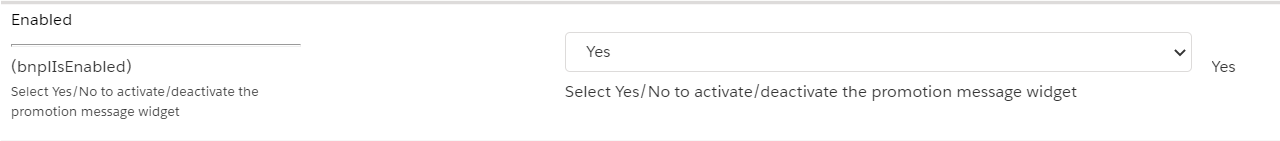


1. **BNPL Promotional widget group:**

a- To enable or disable promotion features provided by Jifiti:

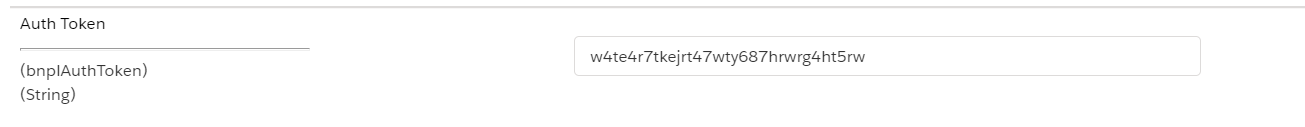
*Set to yes to enable Jifiti features, false to disable. Default is Yes*

*Docs: Select Yes/No to activate/deactivate the promotion message widget*



b- To set the auth token:

*Required*



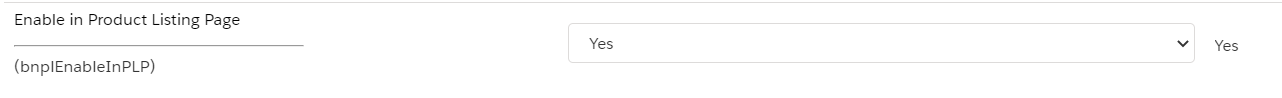
c- To enable promo component in PDP:

*Default is Yes*



d- To enable promo component in PLP:

*Default is Yes*



e- To enable promo component in Cart:

*Default is Yes*



*The following 5 configurations configure the look of the promo component*

f- To change how the promo component looks:

*Default is {}*

*Example: {"border":"1px solid red"}*

*Docs: Customize outer container style it should be a javaScript object containing the css attributes, ex: {backgroundColor: '#D0E5EE',textAlign: 'center',lineHeight: 'normal',padding: '0px'}*

*Graphical user interface, text, application

Description automatically generated*

g- To change the text of the promo component:

*Default is Only {pmt}/mo for {mons} months*

*Docs: Text to be display, should include the following placeholders: {pmt} - payment amount, {mons} - number of months*

*PS: will only take effect if template name is `standard`*

Graphical user interface, text, application

Description automatically generated

h- To change the style of the promo component:

*Default is {}*

*Example: {"color":"yellow"}*

*Docs: Customize link style it should be a javaScript object contains css attributes PS: will only take effect if template name is `standard`*

Graphical user interface, text, application

Description automatically generated

i- To change the text of the link of the promo component:

*Default is Check Eligibility*

*Docs: Determine the behavior of clicking on the link*

*PS: will only take effect if template name is `standard`*

Graphical user interface, application

Description automatically generated

j- To change the style of the link of the promo component:

*Default is {}*

*Example: {"color":"yellow"}*

*Docs: Customize link style it should be a javaScript object contains css attributes PS: will only take effect if template name is `standard`*

*Graphical user interface, text, application

Description automatically generated*

k-To change the behavior of when clicking on the link:

*Default is Popup*

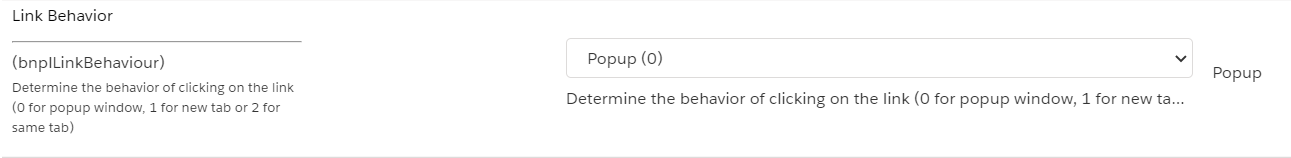
*Docs: Determine the behavior of clicking on the link*

*Popup => open in a popup*

*New Tab => open in a new tab*

*Same Tab => open a same tab*

*PS: if `flow` is set to to `Continue Shopping` or `None` , link behavior will be picked up by Jifiti, otherwise it will not*



l- To change the url of the link:

*Docs: Widget link URL for the specified selector. This is the link that will open the application flow PS: if `flow` is set to `Continue Shopping` or `None` , link behavior will be picked up by Jifiti, otherwise it will not*

Chart

Description automatically generated with low confidence

m- To change the currency symbol:

*Default is $*

*Docs: The currency symbol to be used in the promo widget*

Graphical user interface, application

Description automatically generated

n- Offer Category:

*Example: ZERO3*

*Docs: (optional) Represents the eligible offer category, if not provided, displays the default offer* Table

Description automatically generated

o- To change the template:

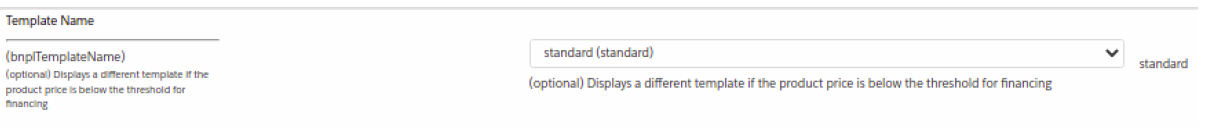
*Default is standard*

*Docs: (optional) Displays a different template if the product price is below the threshold for financing*

***Standard*** *=> pick up the configuration entered for text and link*

***Min Financing*** *=> template with no logo*

***Min Financing With Logo*** *=> template with logo*



p- To configure the flow after clicking on the link:

*(Optional)*

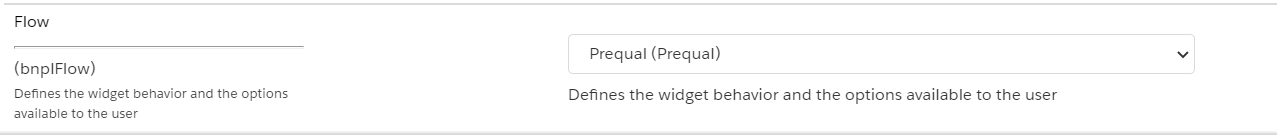
*Docs: what happens when the link is clicked.*

*it has three options:*

***Prequal*** *=> Starts the soft credit check with the InitiateFlow API request, so that the referenceId can be used later on at checkout for a shorter application process.*

***ContinueShopping*** *=> Displays consumer financing information page followed by call to action button to continue shopping.*

***CreateAccount*** *=> Full application flow, ending in creating an account.*



**Service configuration:**Go to Administration > Operations > Services  
and open the credential for credPayment.rest service and add below url in URl field for testing and save it:  
US UAT URL: <https://apiuat.jifiti.com>  
If you are going to use the EU APi urls, please open the credential for credPaymentEU.rest and add below URL:  
EU UAT URL: https://apiuat-eu.jifiti.com

**Jobs configuration:**

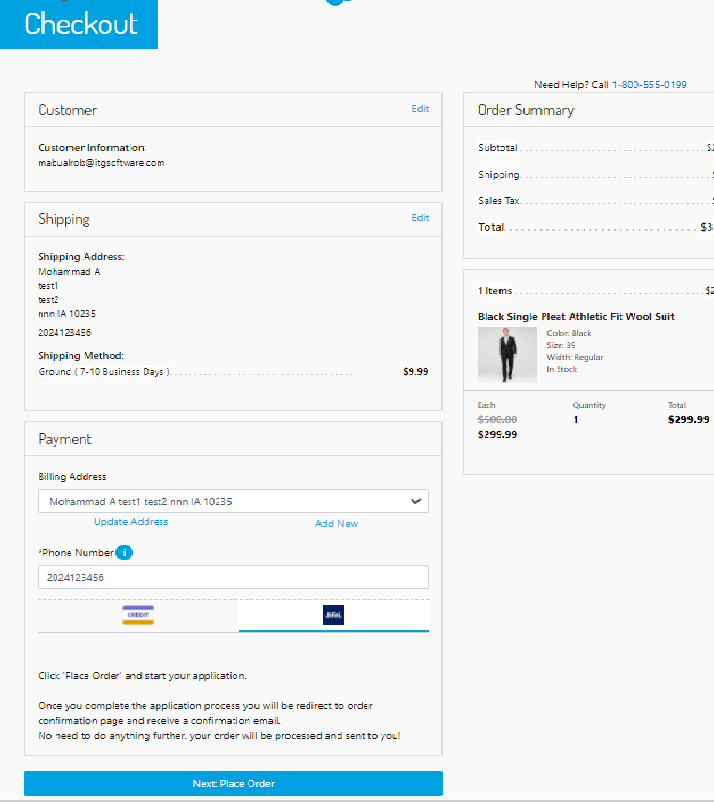
Navigate to Administration > Operations > Jobs and click on GetLenderDetails and click on Run now to get lender details from jifiti

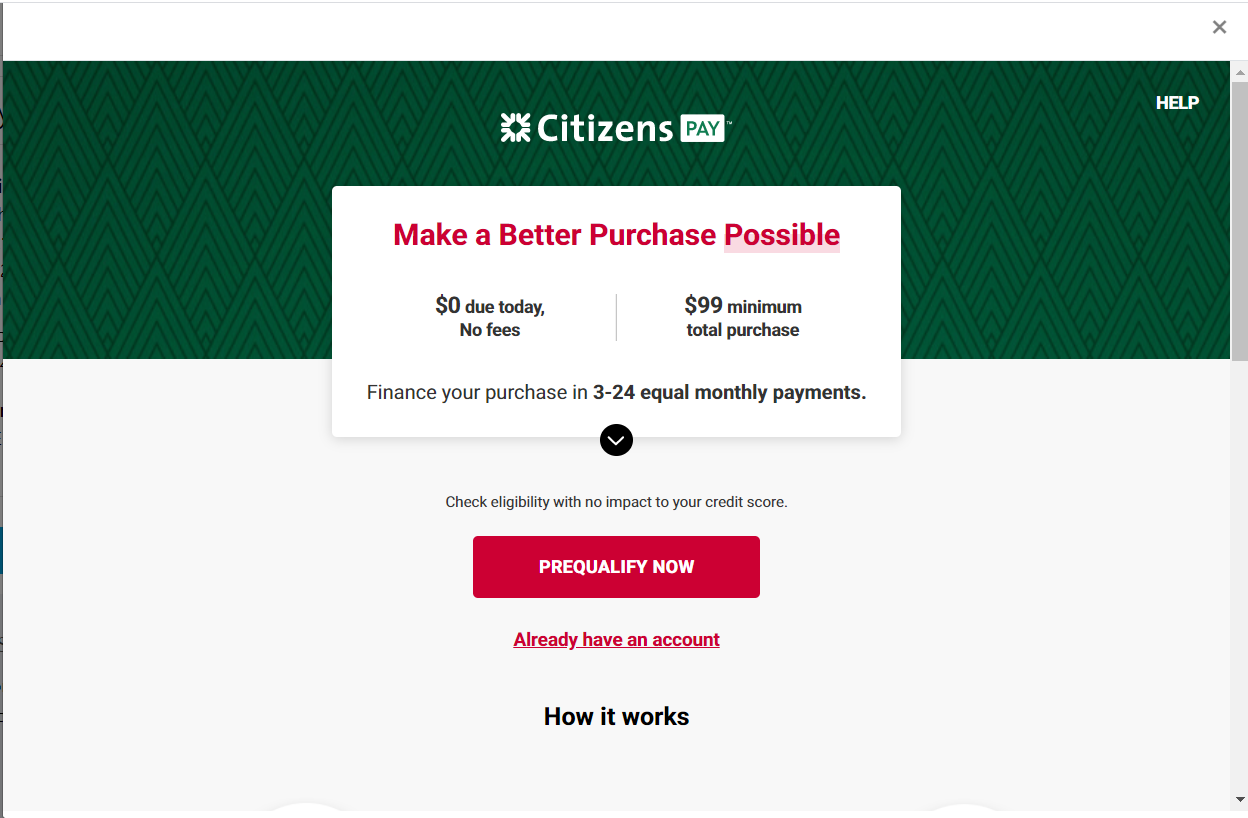
Uploading The code :  
1- please refer below link to upload the cartridges properly:  
[https://documentation.b2c.commercecloud.salesforce.com/DOC2/topic/com.demandware.dochelp/content/b2c\_commerce/topics/sfra/b2c\_uploading\_code.html?resultof=%22%64%77%2e%6a%73%6f%6e%22%20](https://documentation.b2c.commercecloud.salesforce.com/DOC2/topic/com.demandware.dochelp/content/b2c_commerce/topics/sfra/b2c_uploading_code.html?resultof=%22%64%77%2e%6a%73%6f%6e%22%20%20)   
2- compile JS and scss files using below commands  
npm run comopile:js, npm run compile:scss

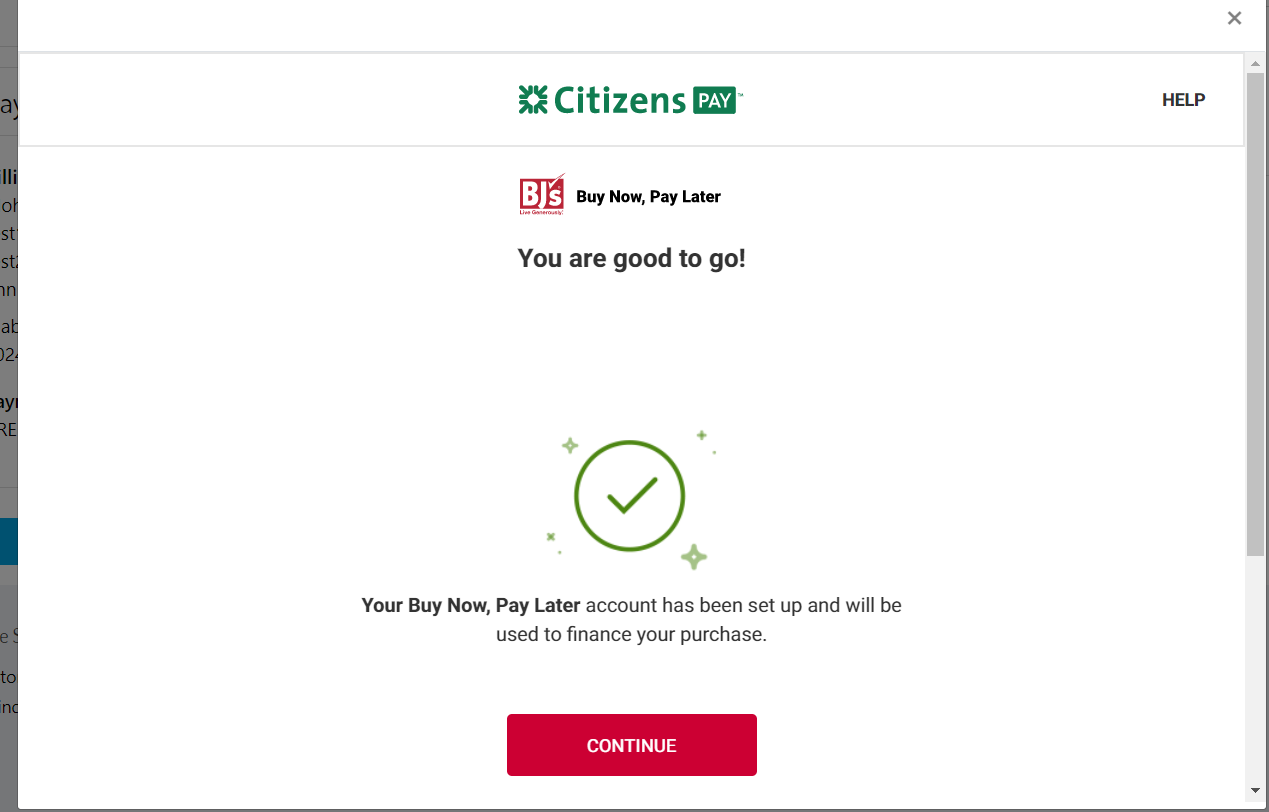
Storefront:  
At the front end, customers can add the items into the shopping cart by clicking the Add to Cart button, then, going to the checkout page, Customer fills in their shipping address and selects the payment method (as shown below in the screenshot) the icon depends on lender.

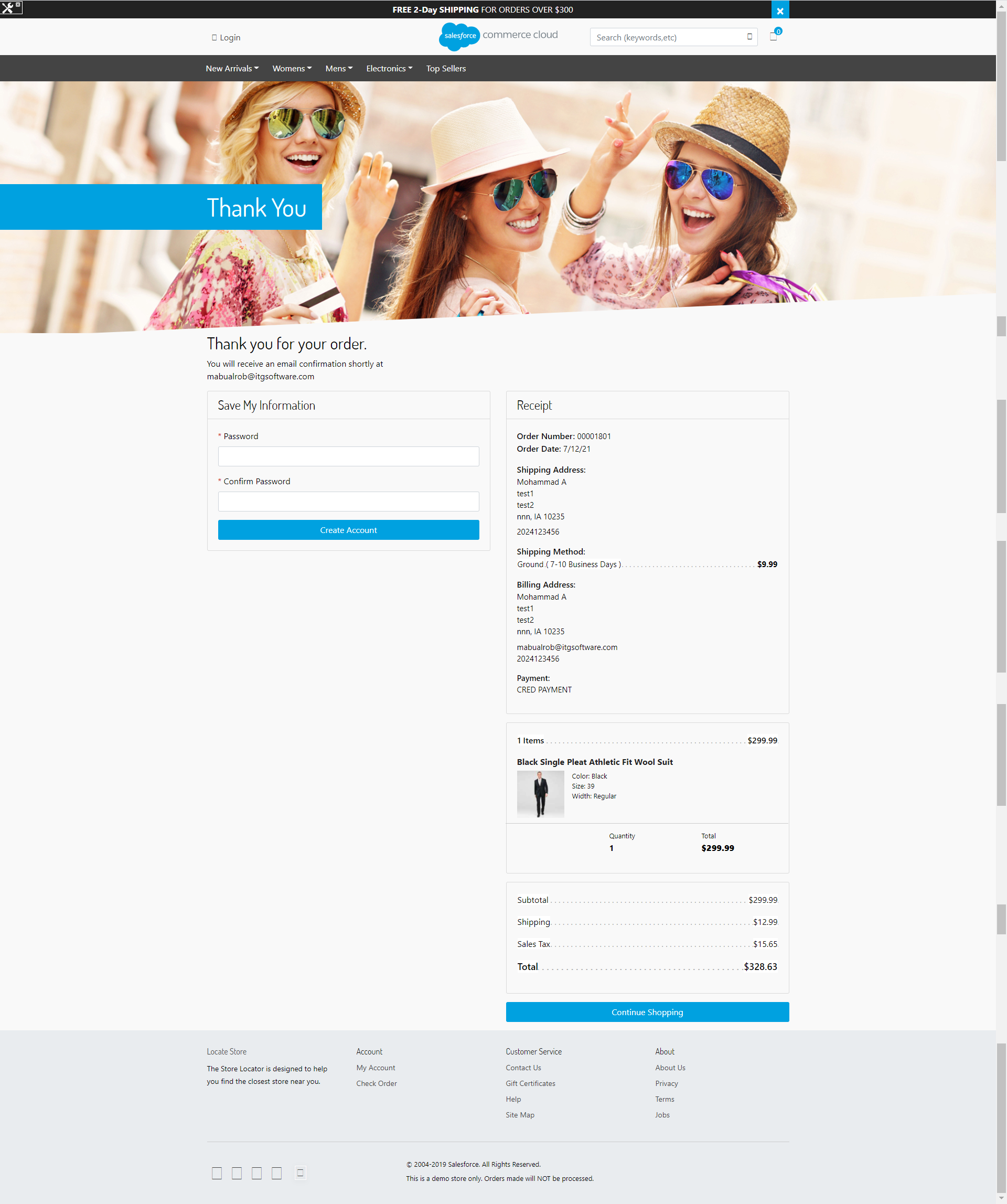
Application Flow will be started, and after the customer completes the flow, they will be redirected to the confirmation page in case all good, or the customer will be redirected back to the checkout page with displaying a failure popup for lightbox case or out of the box error in case same tabs.

Jifiti Messaging feature could be enabled for PDP (product details page), PLP (products listing page), and cart, and it will be added to each product tile and card individually. For example, if we enable Jifiti in PDP, then head to a PDP page for a product, we see Jifiti component mounted on the PDP page.

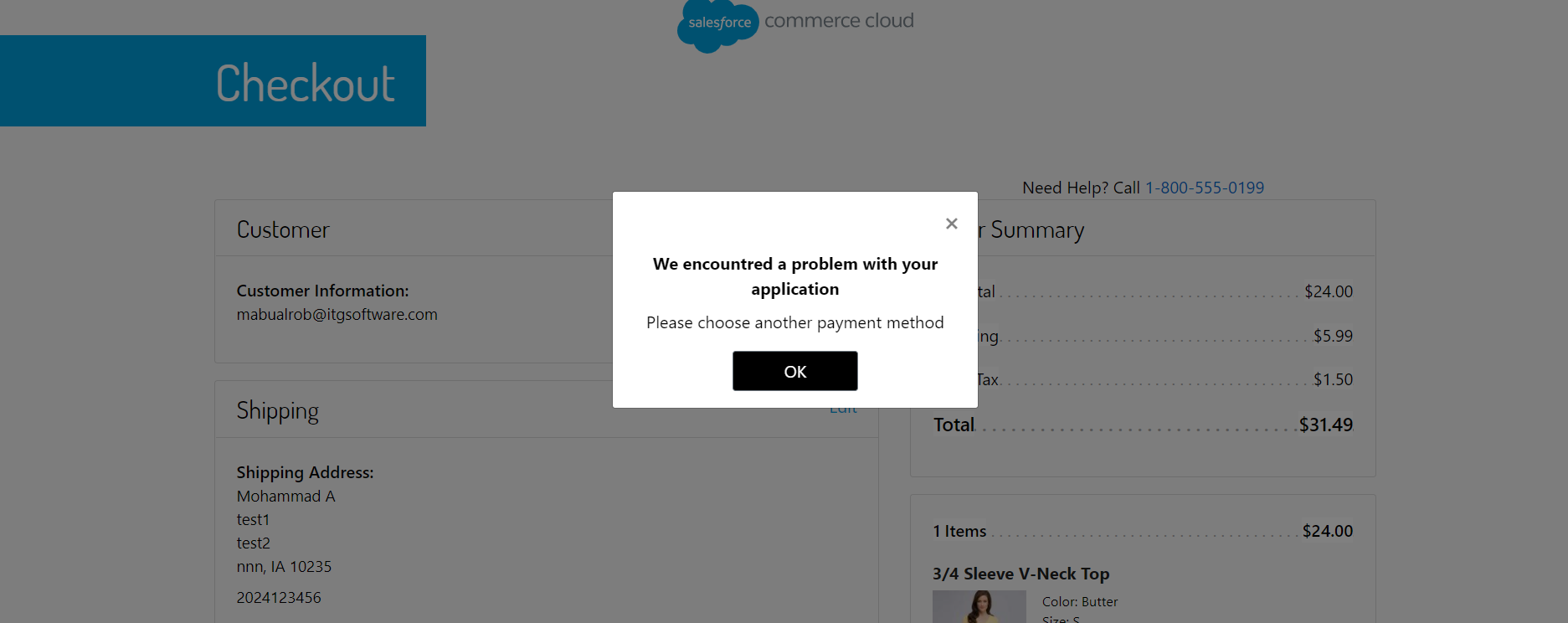








Example of the UX for an unsuccessful application here:



Promotional Widget screenshots:

* In PLP

A picture containing handwear

Description automatically generated

* In PDP

A picture containing text, handwear

Description automatically generated

* In Cart

Graphical user interface, application

Description automatically generated

* After clicking on the link

Graphical user interface

Description automatically generated

Order Status  
After the payment is approved, a new order will be created with a “NEW” status.

[Failover/RecoveryProcess](#_heading=h.8dzugef2k69c):

In the storefront - the order will be failed and the customer will be redirected to checkout page to select new payment method.  
contact support for any issue: ops@jifiti.com .