|  |
| --- |
| CRED Salesforce Payment module |
| User Guide Version 1.0.0  Wednesday, August 25, 2021 |

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).  
  
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 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.

- 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 credPayment are added to:

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

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

Also, the following service are added to Administration > Operations > Services:

credPayment.rest

Configurations:  
**Custom Preferences configuration:**

1. Navigate to Merchant Tools > Site Preferences > Custom Site Preference Groups and click on CredPayment group and fill the below configurations:  
   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 :

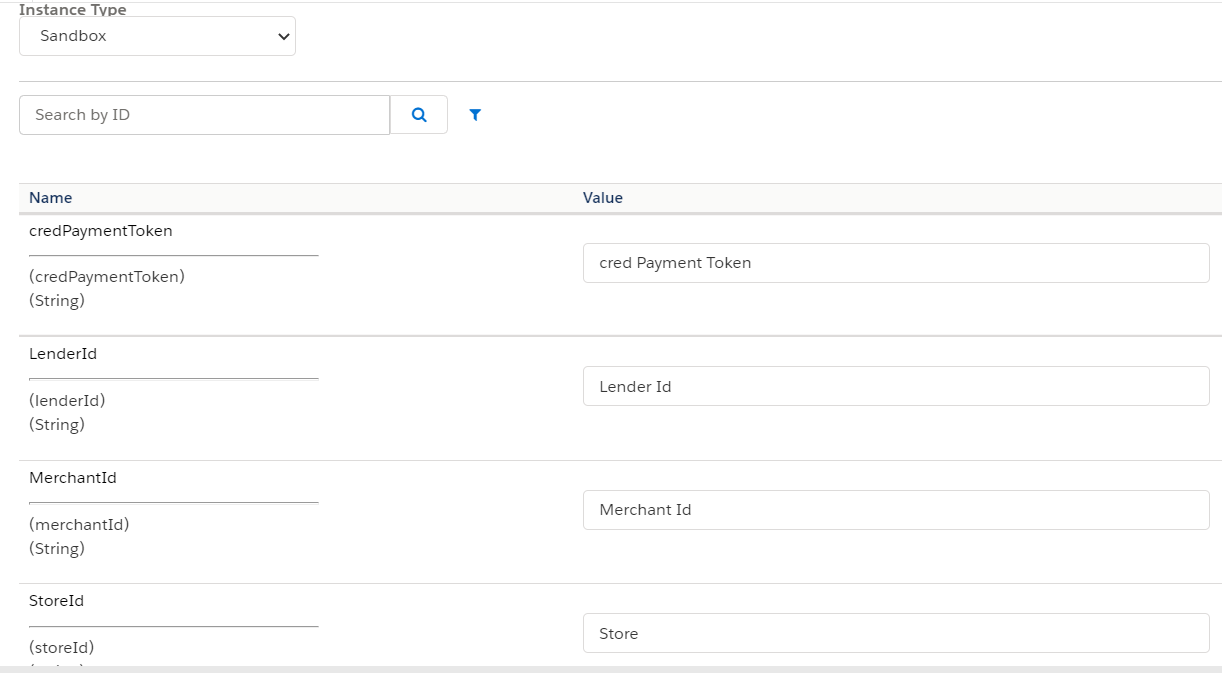
Authorize : in SFCC if the status set as authorize then we will call the purchase API with the InstantCommit flag as false . And the capture call of the API should be called from OMS or any other third party.

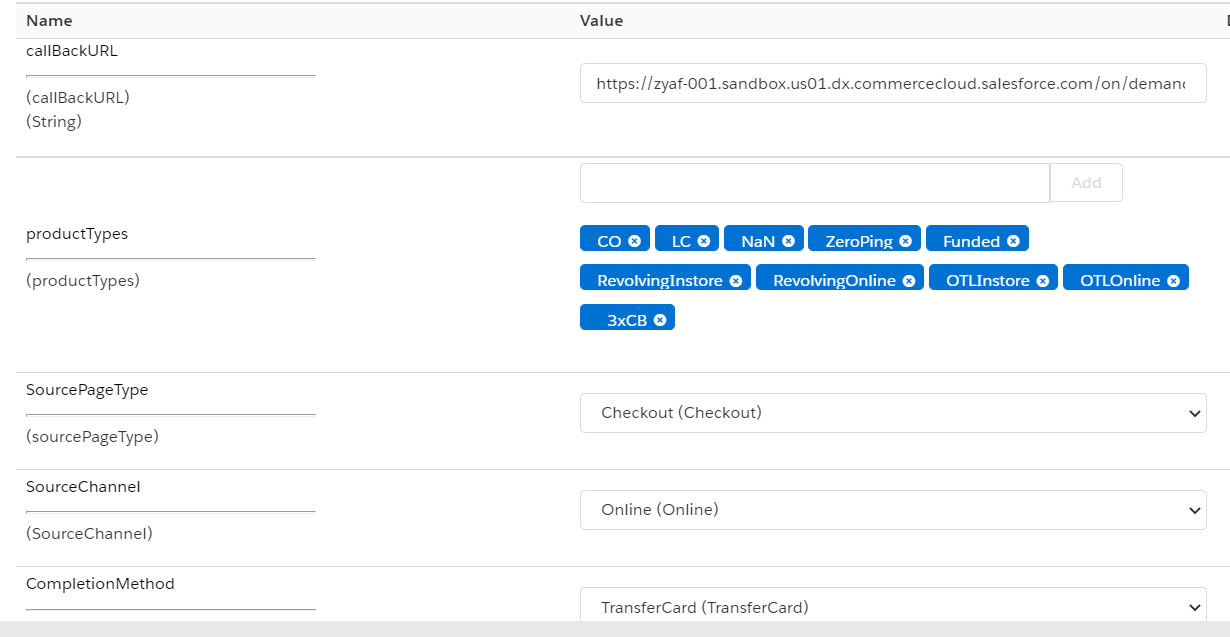
Capture : in SFCC if the status set as capture then we will call the purchase API with InstantCommit flag as true

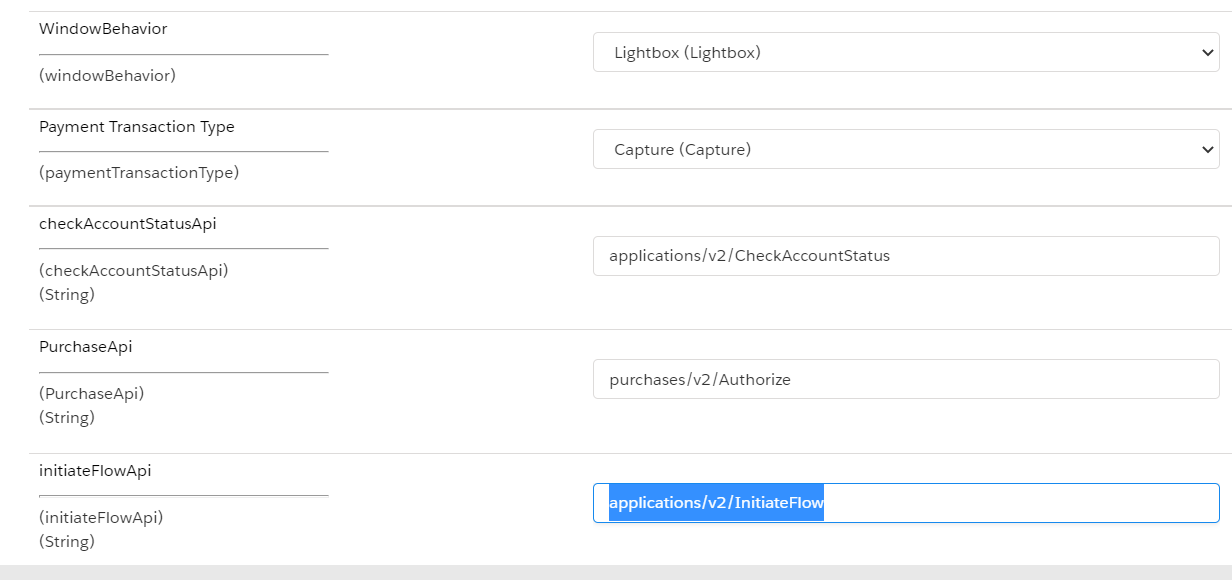
l- checkAccountStatusApi: applications/v2/CheckAccountStatus

m- initiateFlowApi: applications/v2/InitiateFlow

n- PurchaseApi: purchases/v2/Authorize

**Note**: please don’t touch the values for checkAccountStatusApi, initiateFlowApi, PurchaseApi unless you communicated with the payment provider  




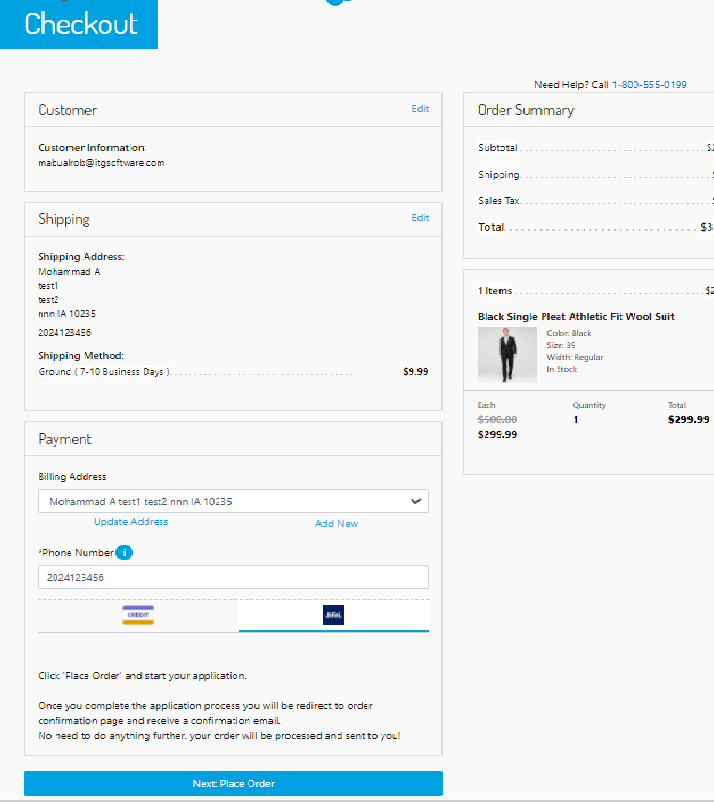


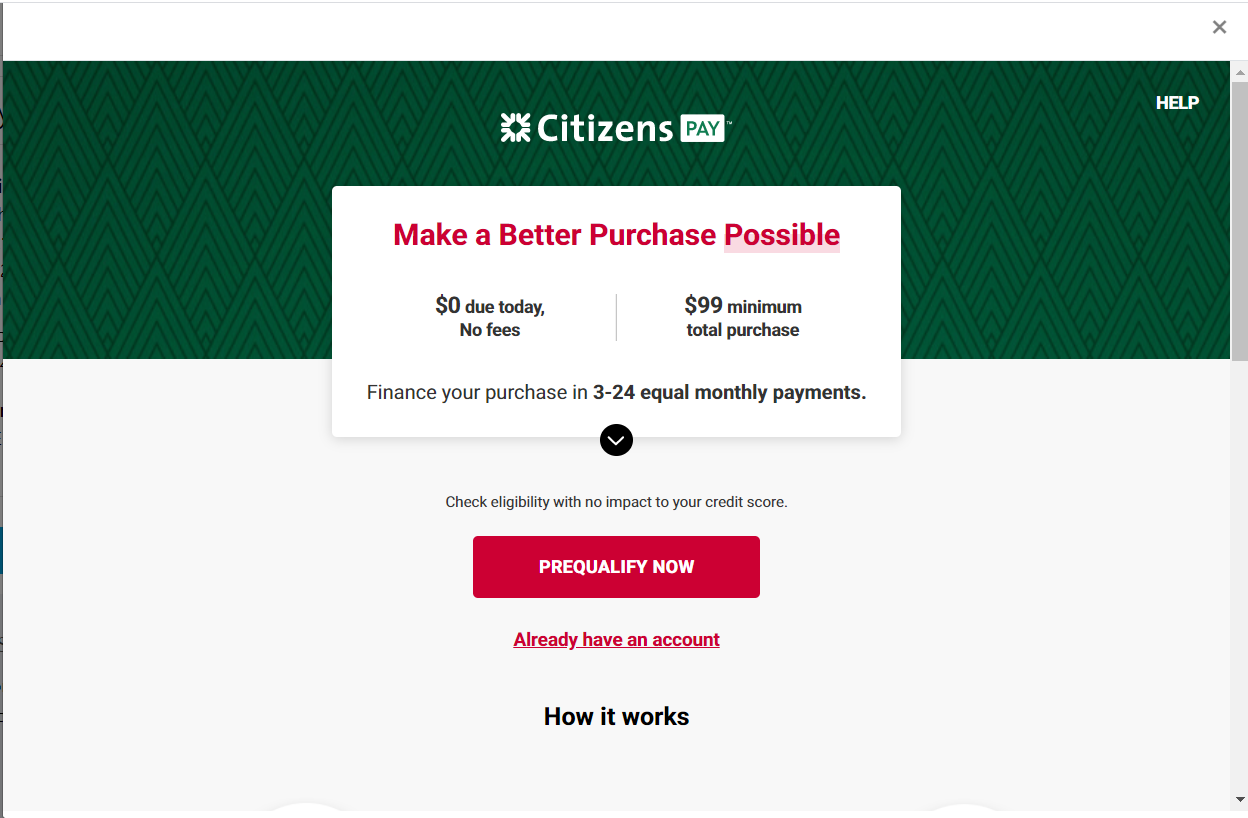
**Service configuration:**Go to Administration > Operations > Services  
and open the credential for credPayment.rest service and add below url in URl field for testing and save it:  
URL: https://apiuat2.jifiti.com

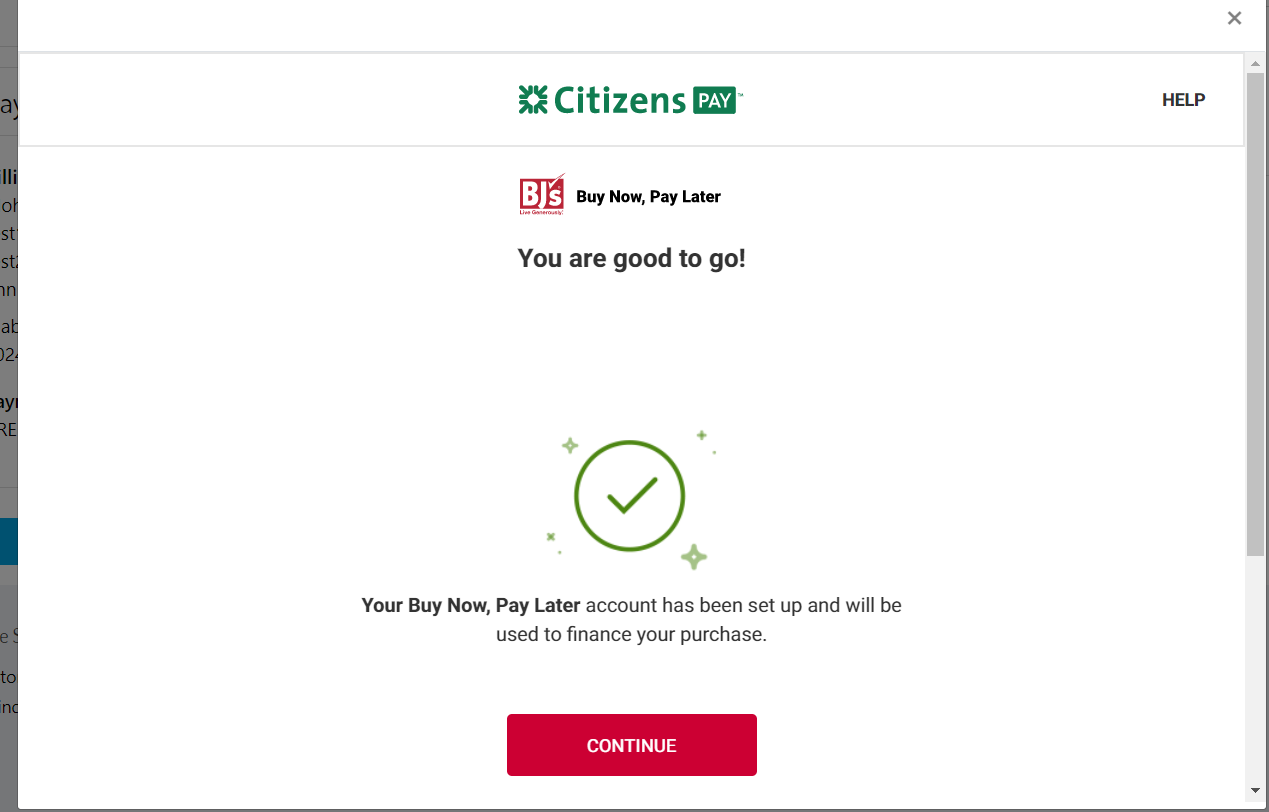
Uploading The code :  
1- please refer below link to upload the cartridges properly:  
[https://documentation.b2c.commercecloud.salesforce.com/DOC2/topic/com.demandware.dochelp/content/b2c\_commerce/topics/sfra/b2c\_uploading\_code.html?resultof=%22%64%77%2e%6a%73%6f%6e%22%20](https://documentation.b2c.commercecloud.salesforce.com/DOC2/topic/com.demandware.dochelp/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

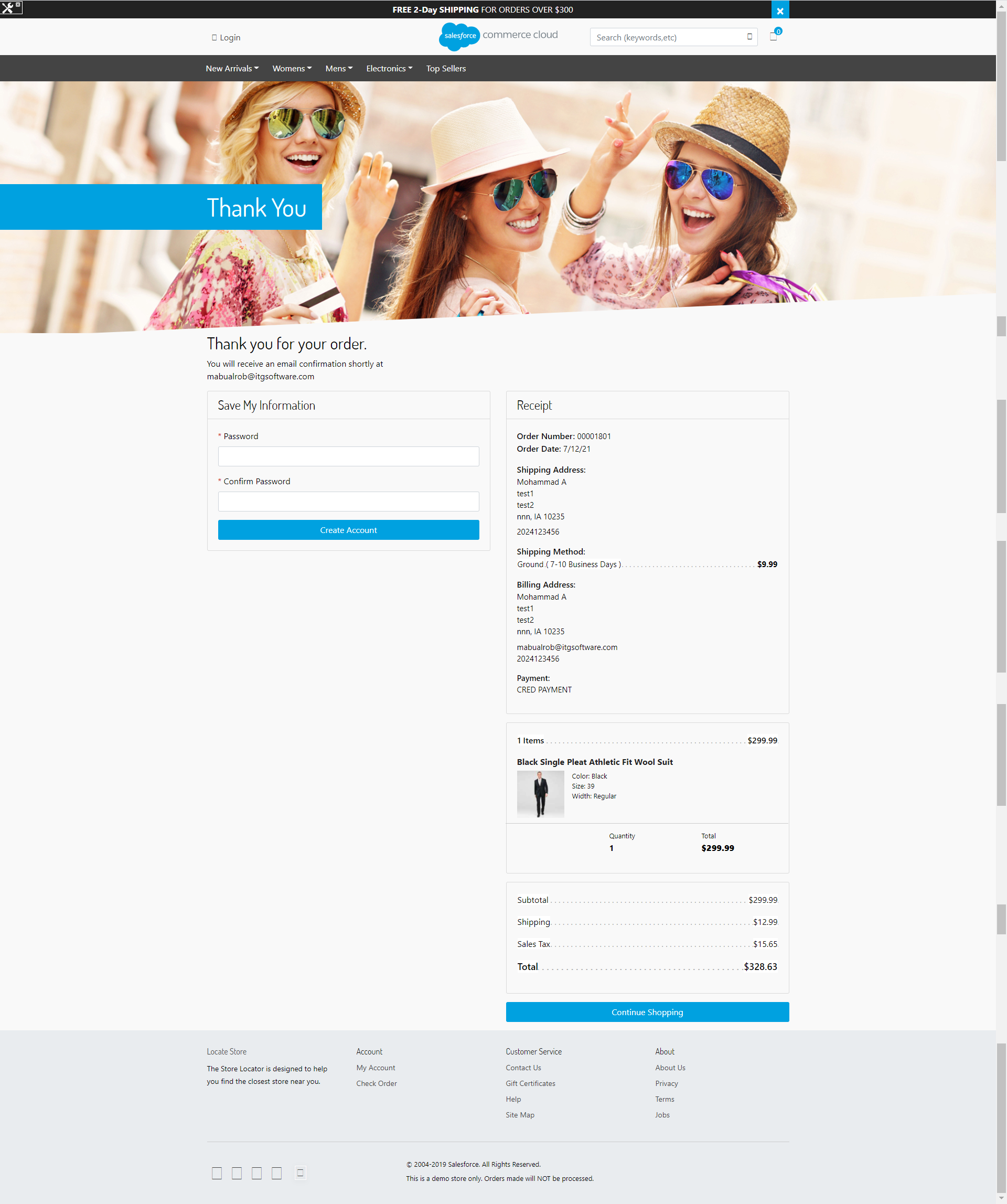
Storefront:  
At the front end, customers can add the items into the shopping cart by clicking the Add to Cart button, then, going to the checkout page, Customer fills in their shipping address and selects the payment method (as shown below in the screenshot).

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.

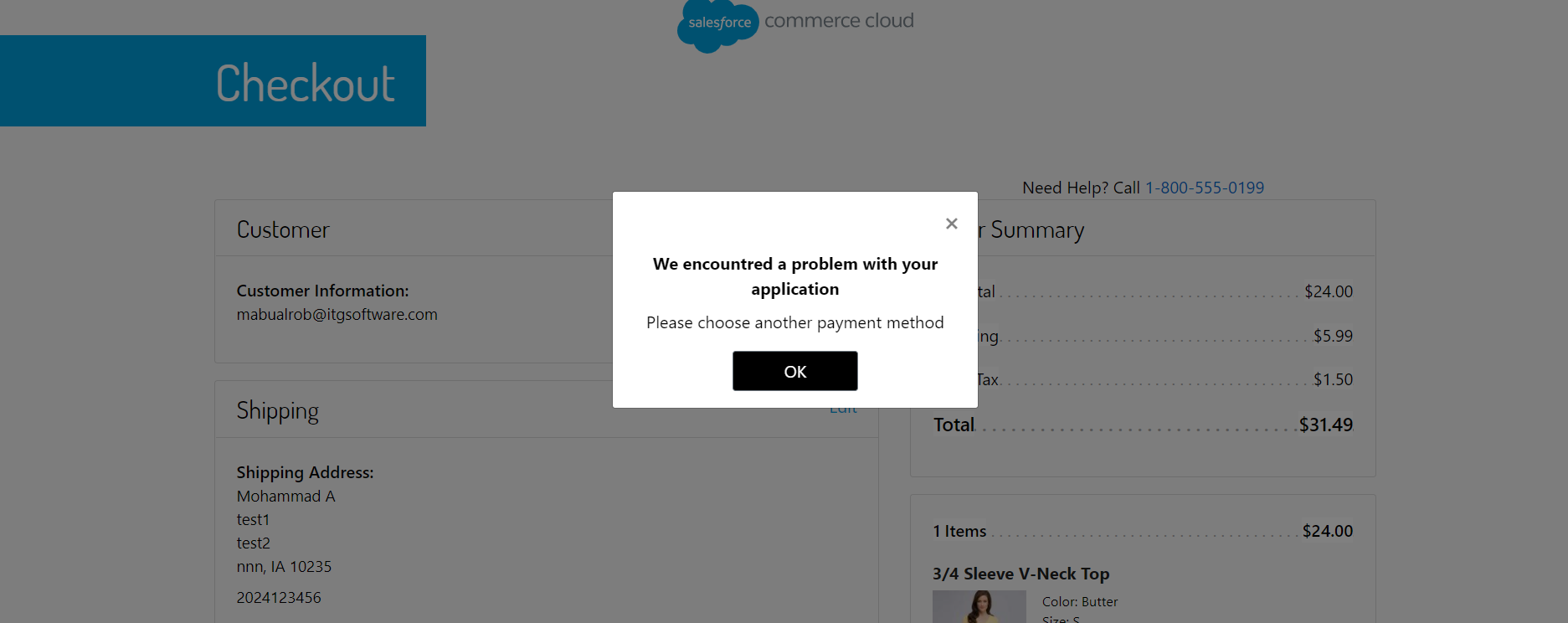








Example of the UX for an unsuccessful application here:



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 .