**Humm Integration**

Version 20.1.0



Table of Contents

[1. Overview 2-3](#_Toc23857143)

[1.1 Compatibility 2-3](#_Toc23857144)

[1.2 Limitations, Constraints 2-3](#_Toc23857145)

[1.3 Privacy, Payment 2-3](#_Toc23857146)

[2. Integration Components 2-3](#_Toc23857147)

[2.1 Components 2-3](#_Toc23857148)

[2.1.1 Signature Generation 2-3](#_Toc23857149)

[2.1.2 Prepare Redirect 2-4](#_Toc23857150)

[2.2 Promotional widgets 2-4](#_Toc23857151)

[2.3 Adding the cartridge in SFCC UX Studio 2-4](#_Toc23857152)

[2.4 Business Manager Configurations 2-4](#_Toc23857153)

[2.4.1 Activating the cartridge in Business Manager 2-4](#_Toc23857154)

[2.4.2 Enable Cartridge Module 2-5](#_Toc23857155)

[2.4.3 Importing Metadata 2-5](#_Toc23857156)

[2.4.4 Type Extensions 2-6](#_Toc23857157)

[2.4.5 Payment Methods 2-7](#_Toc23857158)

[2.4.6 Payment Processors 2-10](#_Toc23857159)

[2.4.7 HTTP Service 2-10](#_Toc23857160)

[2.4.8 Modifying Content 2-10](#_Toc23857161)

[3. SiteGenesis Application Implementation 3-11](#_Toc23857162)

[3.1 Humm Payment 3-11](#_Toc23857163)

[3.2 Custom Code 3-11](#_Toc23857164)

[3.2.1 JavaScript Controllers 3-11](#_Toc23857165)

[3.2.2 CSS 3-14](#_Toc23857166)

[3.2.3 Templates 3-14](#_Toc23857167)

[4. Testing 4-19](#_Toc23857168)

[4.1 Checkout data flow 4-20](#_Toc23857169)

[4.1.1 Humm checkout 4-20](#_Toc23857170)

[4.2 Sample Widgets 4-23](#_Toc23857171)

[4.2.1 Home Page Banner 4-23](#_Toc23857172)

[4.2.2 Cart Page Widget 4-24](#_Toc23857173)

[4.2.3 Product Details Page Widget 4-24](#_Toc23857174)

[4.3 Backend Transaction 4-25](#_Toc23857175)

[5. Known Issues 5-27](#_Toc23857176)

[6. Release History 6-28](#_Toc23857177)

# Overview

This document describes the technical integration of Humm link cartridge. The Humm link cartridge will allow merchants to offer Humm, the buy now pay later payment option for your Commerce Cloud shop solution.

Supported schemes/payment types are:

* Debit
* Visa
* MasterCard

Outages/failures relating to 3rd party services used in Humm are raised with the appropriate 3rd party, who would resolve it as per agreed SLAs.

However, failures relating to Humm are treated as Priority 1, Priority 2 or Priority 3 with P1 including complete system outages.

P1 and P2 are typically assigned an Incident Manager within IT who will:

* Advise Humm Buyer and Seller facing teams of resolution progress.
* Liase with Seller Account Managers if such managers are appointed.
* Liase with sellers' IT Teams if additional custom development was performed.
* Once resolved, an Incident Report is sent to relevant Account Managers.

P3 issues are treated in a similar fashion except that:

* No Incident Reports are prepared.
* Can be deprioritised in favour of P1, P3 depending on issue's impact/severity.

## Compatibility

Designed for Commerce Cloud API version 16.2 or higher, based on SiteGenesis 2.

## Limitations, Constraints

## Privacy, Payment

This integration requires access to the following customer data elements: Billing Address, Shipping Address, Order Details and Customer Profile.

No credit card details are stored within SFCC using this integration.

# Integration Components

## Components

Humm Integration has the following cartridges.

* **int\_humm\_core -** This cartridge has scripts necessary for integration on SiteGenesis site. The files can be changed to meet customer's requirements.
* **int\_humm\_controllers -** This cartridge has controllers necessary for integration on SiteGenesis site. The files can be changed to meet customer's requirements.
* **bm\_humm -** Business Manager extension cartridge for transaction management, for both SiteGenesis.

### Signature Generation

To begin the checkout of a customer that chooses Humm payment method on your website, Humm implements a signing mechanism based on HMAC-SHA256 to prevent against malicious attacks and hijacking of browser session.

### Prepare Redirect

Once signature is generated, a post request will send to the Humm gateway with transaction information in application/x-www-form-url encoded format for redirection to Humm page.

## Promotional widgets

* Homepage Humm banner indicates that Humm payment method is available for the website.
* In footer, "About Humm " link provides more information about humm payment methods.
* In Product details page, payment widget is used to display a price breakdown.
* Cart page widget is used to remind the customers about humm payment method.

## Adding the cartridge in SFCC UX Studio

To upload the cartridges into the SFCC server, you first need to add the cartridges in SFCC UX Studio. In order to do this, follow these instructions.

1. In UX Studio, select in the menu File > Import.
2. In the import dialog, select General > Existing Projects in the workspace and click on 'Next'.
3. Ensure 'Select archive file' is selected and select the compressed cartridge file by clicking on the 'Browse' button.
4. Click on 'Finish' button to import the cartridges.
5. Studio will now ask you if you want to link the cartridge to your active DigitalServer connection. Click on yes or manually link the cartridge to your server by checking the project under project references in the server connection properties.

## Business Manager Configurations

For the Humm integration to work, the following needs to be configured in the Business manager.

### Activating the cartridge in Business Manager

Before the Humm functionality can become available to Site, the cartridges have to be added to the cartridge path of the Site. In order to do this, follow the following instructions:

1. Log into Business Manager.
2. Navigate to Administration > Sites > Manage Sites.
3. Click on the site name and on the next page go to the 'Settings' tab.
4. In the 'Cartridges' input field append 'int\_humm\_controllers:int\_humm\_core' to storefront cartridges path.
5. Click on 'Apply' button.
6. To activate the cartridge for the Sandbox/Development/Production instances repeat steps 4 and 5 after selecting the appropriate instance from the 'Instance Type' dropdown menu.
7. Repeat steps 3 to 6 for each site that is to use Humm.
8. Go to 'Manage the Business Manager site'.
9. In the Cartridges input field add 'bm\_humm:int\_humm\_core'.

### Enable Cartridge Module

To enable Cartridge module,

1. Go to Administration > Organization > Roles & Permissions.
2. Click on "Administrator" and click on "Business Manager Modules" tab.
3. In Select Context dialog, check your site ID checkbox and click on "Apply" button.
4. Find ‘Flexigroup' and check the checkbox to enable it.

### Importing Metadata

For the Humm integration to work, the following object structures (metadata) needs to be imported and configured in the Business manager. Follow the below steps:

1. In the cartridge bundle find **metadata/** **humm-meta-data** folder.
2. Find the **sites** folder and inside it find **RefArch** folder.
3. Rename this RefArch folder to the ID of your site. If you have multiple site to which these meta has to be imported, then copy and paste this RefArch folder to replicate for other sites. Renames those folders with ID of corresponding sites.
4. Open the specific site folder and find **library** folder. Inside library folder find library.xml.
5. Edit the library.xml file to change the library-id mentioned in it. If your site uses private library, then remove the library-id attribute and its value. If your site uses shared library, then replace the existing library-id **RefArchSharedLibrary** in xml by your actual shared library id. Save the library.xml file.
6. In the cartridge bundle, inside **metadata** folder compress humm-meta-data folder to generate **humm-meta-data.zip** file.
7. Go to Business Manager Menu > Administration > Site Development > Site Import & Export
8. Under Import: Upload Archive : Ensure that the radio button with label **Local** is enabled (Else click on the radio button to enable it)
9. Click on Choose File input field, select the humm-meta-import.zip file from open dialog box and click on upload button
10. After finishing the upload, from the Archives list click the radio button corresponding to **humm-meta-data.zip** and click on Import button
11. Click on OK button of the confirmation box asking “Are you sure that you want to import the selected archive?”

### Type Extensions

* **Custom site preference**

**After the successful metadata import following attributes should have been created :**

In Business Manager, navigate to the SiteGenesis Site > Site Preferences > Custom Preferences. Custom site preference groups with the ID Integration\_Humm is available. Please select it and edit the attributes according to your Humm account data.

For each site, set the values for your site preferences.

**Group ID: Integration\_Humm**

1. **Is Humm Enabled**(ID - isHummEnabled, Type - Boolean) - To enable or disable humm and functionality in a single point.
2. **Merchant ID** (ID - hummMerchantID, Type - String) - This is a unique Merchant ID that is assigned by Humm to individual merchants.
3. **Gateway Key** (ID - hummGatewayKey, Type - String) - This is a unique key that is assigned by Humm to individual merchants.
4. **Gateway URL** (ID - hummGatewayURL, Type - String) - This is the Humm gateway to communicate with Humm in order to process a transaction.
5. **Refund Endpoint** (ID - hummRefundEndpoint, Type - String) - This is the API to programmatically process refunds or adjustments to an Humm purchase.
6. **Minimum Order Total** (ID - hummMinOrderTotal, Type - Number) - This is used to set minimum ordertotal for carts which are eligible to transact against Humm. Default value is 20.
7. **Maximum Order Total** (ID - hummMaxOrderTotal, Type - Number) - This is used to set maximum ordertotal for carts which are eligible to transact against Humm.
8. **Widget URL** (ID - hummWidgetURL, Type - String) - This is used to set the widget URL.

* **OrderPaymentInstrument custom attributes :**

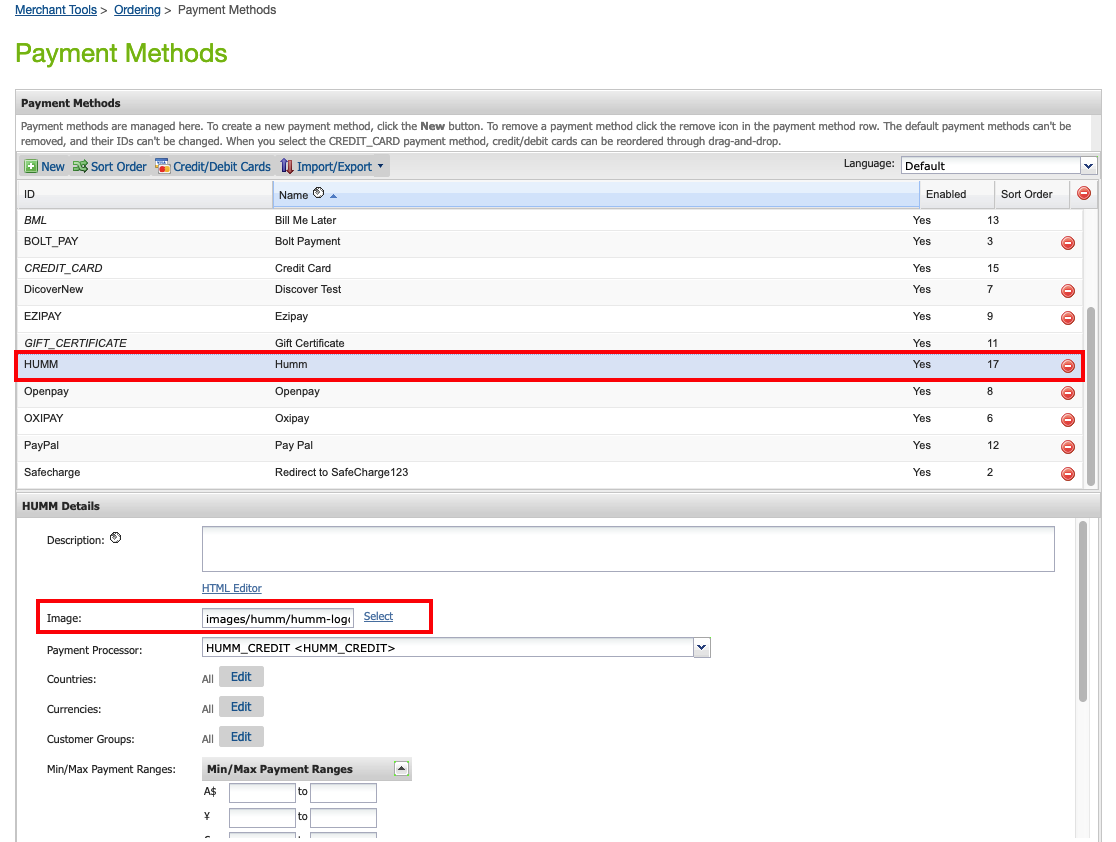
1. hummPaidAmount (String)
2. hummSettleHistory (Text)
3. hummTransactionDate (String)
4. hummTransactionID (String)
5. hummTransactionStatus (String)

### Payment Methods

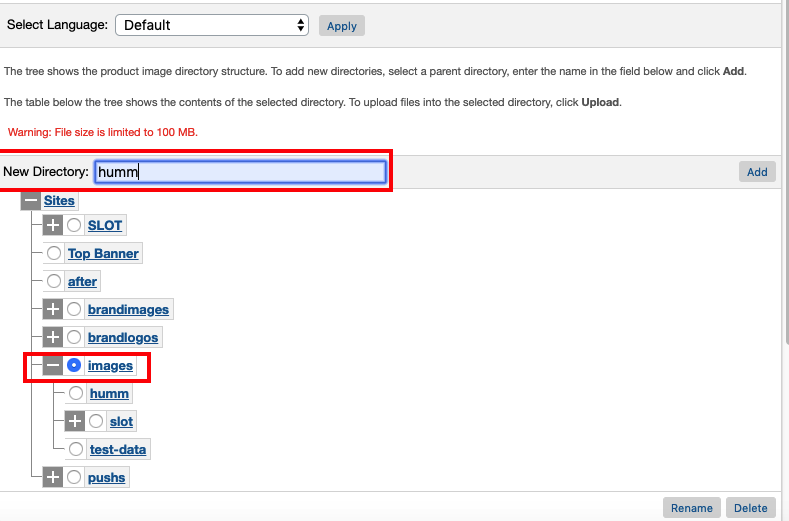
* **Payment methods import**
* After the successful import check that payment method with ID “HUMM” is created.
* **Add the image to Payment Method**

In order to display the logo of humm in payment method section in the site, the image needs to be added to your site.

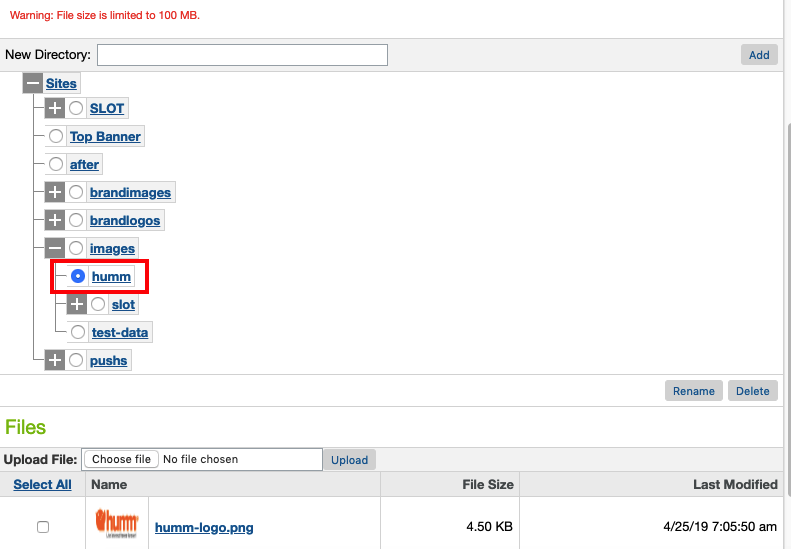
1. Go to Site > Ordering > Payment Methods.
2. Select the payment method with ID “HUMM” and locate the image attribute.



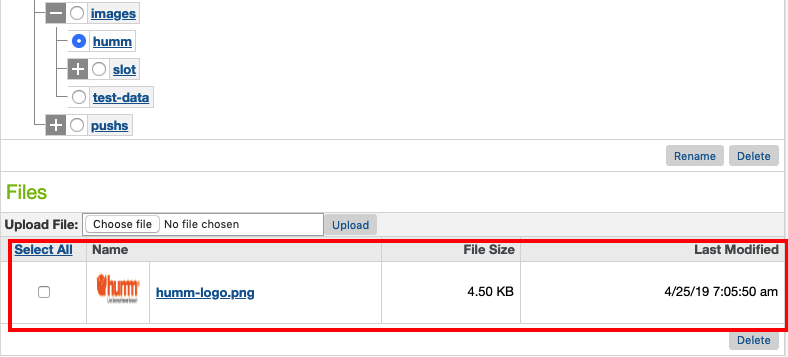
1. Click on 'Select' link and a dialog will open.
2. Expand the 'Sites' link and Select 'images' radio button.
3. Create a new directory with name 'humm' under 'images.

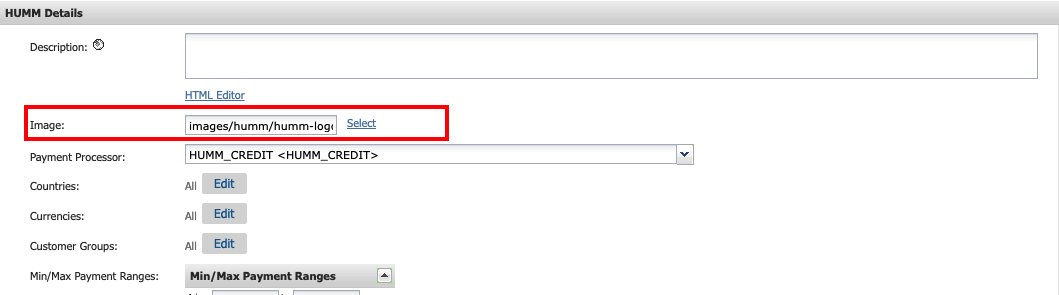


1. Select 'humm' radio button and upload the image included in the cartridge (int\_humm\_core/cartridge/static/default/images/humm-logo.png).



1. Click on the uploaded image link and click on 'Apply'.





### Payment Processors

* After the successful import check that payment processor with ID “HUMM\_CREDIT” is created.

### HTTP Service

* After the successful import check that service “humm.http” is created.

### Modifying Content

There are content assets and content slot those are used to display humm widgets. These content assets and content slot are used in the checkout page and are displayed when selecting Humm as payment method. The content assets can be modified as required to suit your site.

Below content assets are available after the successful metadata import.

* footer-humm-badge
* humm-banner
* humm-cart-page-widget
* humm-checkout
* humm-landing

Below content slot is available after the successful metadata import.

* humm-banner

**Note**: In the slot configuration, for the field “Template” the value must be according to your storefront cartridge. For ex: “slots/content/contentassetbody.isml”

# SiteGenesis Application Implementation

## Humm Payment

Humm will be offered as another payment method for the shopper to be selected on the Billing Page.

When the shopper selects to Continue with Humm payment methods, they will be redirected to the Humm gateway. If the shopper selects to Cancel the transaction, they will be returned back to the Home page. If transaction is success in Humm gateway, they will return to Commerce Cloud. The shopper will be presented with the Order confirmation page with their order number. If transaction is failed in Humm gateway, they will return to Commerce Cloud. The shopper will be presented with the page where transaction status will be available.

## Custom Code

To integrate humm into SiteGenesis a few changes have to be made on JavaScript controllers.

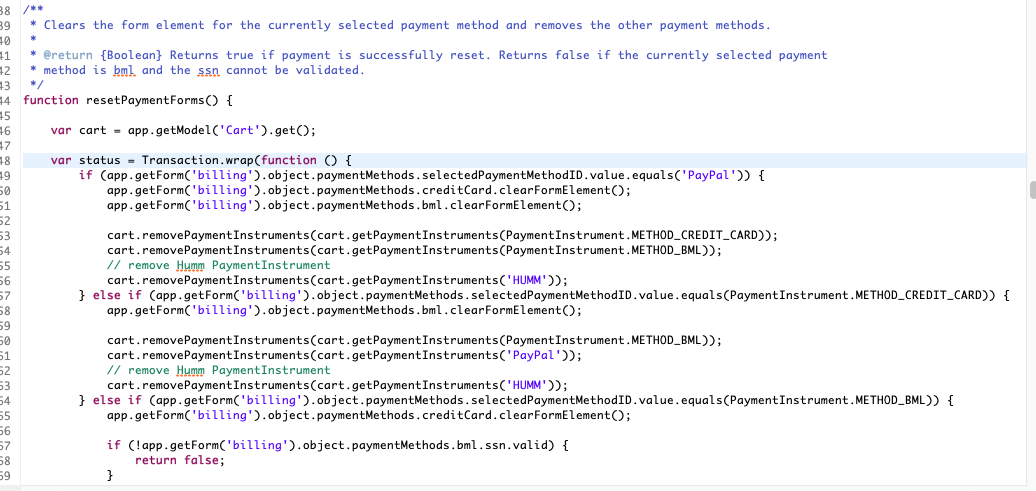
### JavaScript Controllers

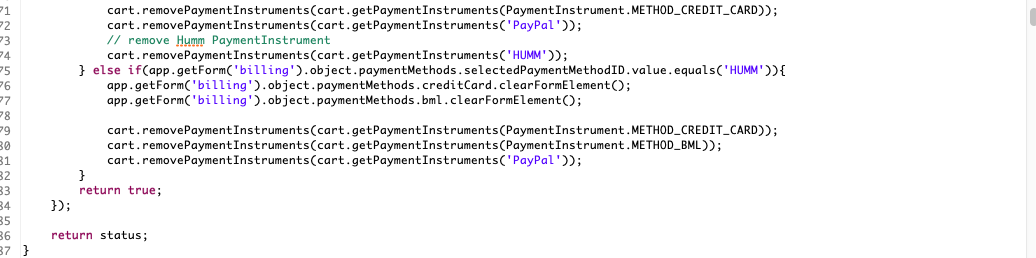
* In SiteGenesis controller **COBilling.js** in function **resetPaymentForms()**, add below line for the the if conditions of payment methods other than “HUMM”.

|  |
| --- |
| // remove Humm PaymentInstrument  cart.removePaymentInstruments(cart.getPaymentInstruments('HUMM')); |

* In the same function ,add below mentioned condition.

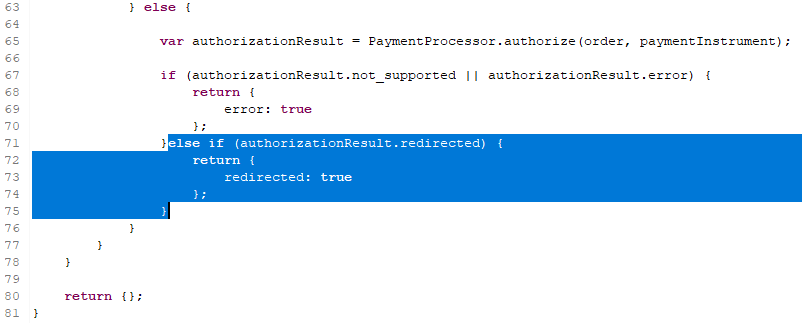
|  |
| --- |
| **else if**(app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value.equals(' HUMM ')){  app.getForm('billing').object.paymentMethods.creditCard.clearFormElement();  app.getForm('billing').object.paymentMethods.bml.clearFormElement();  cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD\_CREDIT\_CARD); cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD\_BML));  cart.removePaymentInstruments(cart.getPaymentInstruments('PayPal'));  } |





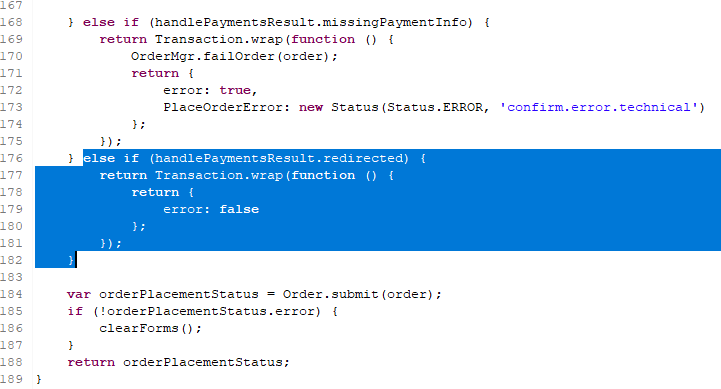
* In SiteGenesis controller **COPlaceOrder.js** in function handlePayments(order) ,add another “else if” branch to return a response indicating a redirect to humm.

|  |
| --- |
| **else** **if** (authorizationResult.redirected) {  **return** {  redirected: **true**  };  } |



* In SiteGenesis controller **COPlaceOrder.js** in function start(), after the call of function handlePayments(order) add another “else if” branch to the response handling.

|  |
| --- |
| **else** **if** (handlePaymentsResult.redirected) {  **return** Transaction.wrap(**function** () {  **return** {  error: **false**  };  });  } |



### *CSS*

* In SiteGenesis default **app\_storefront\_core/cartridge/scss/default/style.scss**,add below line at end.

|  |
| --- |
| @import "int\_humm\_core/cartridge/scss/default/humm"; |

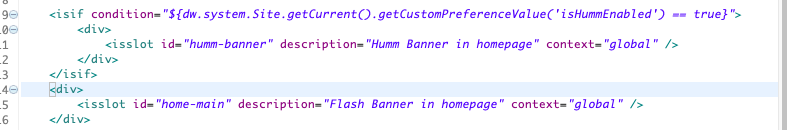


* Please compile scss file into css file to reflect styling in storefront.

### Templates

* In the file **app\_storefront\_core/cartridge/templates/default/content/home/homepage.isml** add below lines.

|  |
| --- |
| <isif condition=*"${dw.system.Site.getCurrent().getCustomPreferenceValue('* *isHummEnabled ') == true}"*>  <div>  <isslot id=*"humm-banner"* description=*"Humm Banner in homepage"* context=*"global"* />  </div>  </isif> |



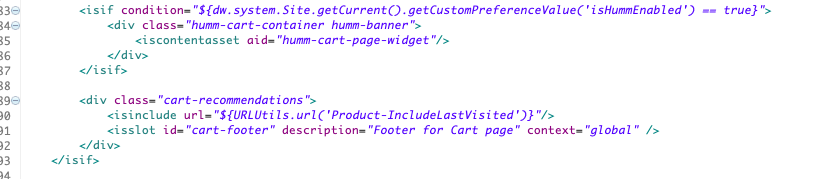
* In the file, **app\_storefront\_core/cartridge/templates/default/components/footer/footer.isml** add below lines.

|  |
| --- |
| <div class=*"footer-container footer-badge-container"*>  <div class=*"footer-item"*>  <isif condition=*"${dw.system.Site.getCurrent().getCustomPreferenceValue('isHummEnabled') == true}"*>  <iscontentasset aid=*"footer-humm-badge"*/>  </isif>  </div>  </div> |



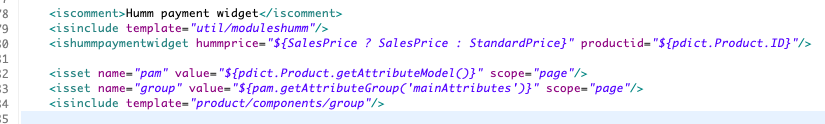
* In the file, **app\_storefront\_core/cartridge/templates/default/checkout/cart/cart.isml** add below lines.

|  |
| --- |
| <isif condition=*"${dw.system.Site.getCurrent().getCustomPreferenceValue('isHummEnabled') == true}"*>  <div class=*"humm-cart-container humm-banner"*>  <iscontentasset aid=*"humm-cart-page-widget"*/>  </div>  </isif> |



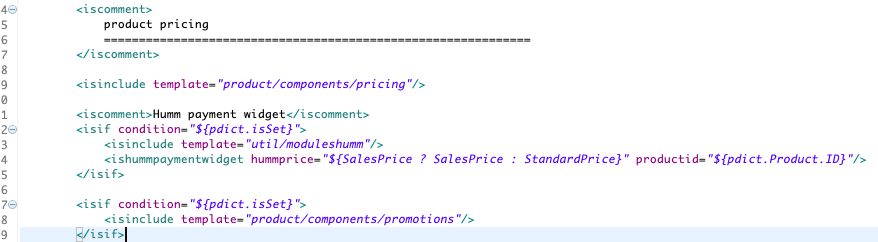
* In the file, **app\_storefront\_core/cartridge/templates/default/product/productcontent.isml** add below lines.

|  |
| --- |
| <iscomment>Humm payment widget</iscomment>  <isinclude template=*"util/moduleshumm"*/>  <ishummpaymentwidget hummprice=*"${SalesPrice ? SalesPrice : StandardPrice}"* productid=*"${pdict.Product.ID}"*/> |



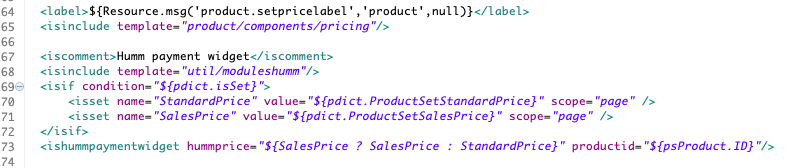
* In the file, **app\_storefront\_core/cartridge/templates/default/product/components/productsetproduct.isml** add below lines.

|  |
| --- |
| <iscomment>Humm payment widget</iscomment>  <isif condition=*"${pdict.isSet}"*>  <isinclude template=*"util/moduleshumm"*/>  <ishummpaymentwidget hummprice=*"${SalesPrice ? SalesPrice : StandardPrice}"* productid=*"${pdict.Product.ID}"*/>  </isif> |



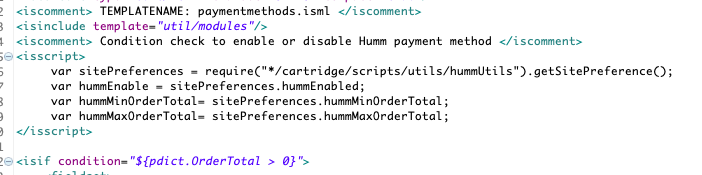
* In the file, **app\_storefront\_core/cartridge/templates/default/product/producttopcontentPS.isml** add below lines.

|  |
| --- |
| <iscomment>Humm payment widget</iscomment>  <isinclude template=*"util/moduleshumm"*/>  <isif condition=*"${pdict.isSet}"*>  <isset name=*"StandardPrice"* value=*"${pdict.ProductSetStandardPrice}"* scope=*"page"* />  <isset name=*"SalesPrice"* value=*"${pdict.ProductSetSalesPrice}"* scope=*"page"* />  </isif>  <ishummpaymentwidget hummprice=*"${SalesPrice ? SalesPrice : StandardPrice}"* productid=*"${psProduct.ID}"*/> |



* In the file, **app\_storefront\_core/cartridge/templates/default/checkout/billing/paymentmethods.isml** add below lines.

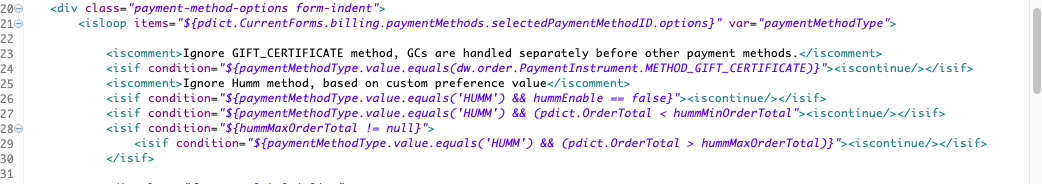
|  |
| --- |
| <isscript>  var sitePreferences = require("\*/cartridge/scripts/utils/hummUtils").getSitePreference();  var hummEnable = sitePreferences.hummEnabled;  var hummMinOrderTotal= sitePreferences.hummMinOrderTotal;  var hummMaxOrderTotal= sitePreferences.hummMaxOrderTotal;  </isscript> |



* In the file, **app\_storefront\_core/cartridge/templates/default/checkout/billing/paymentmethods.isml** add below linesafter the line :

<isif condition=*"${paymentMethodType.value.equals(dw.order.PaymentInstrument.METHOD\_GIFT\_CERTIFICATE)}"*><iscontinue/></isif>, add the below mentioned code :

|  |
| --- |
| <iscomment>Ignore Humm method, based on custom preference value</iscomment>  <isif condition=*"${paymentMethodType.value.equals('HUMM') && hummEnable == false}"*><iscontinue/></isif>  <isif condition=*"${paymentMethodType.value.equals('HUMM') && (pdict.OrderTotal < hummMinOrderTotal"*><iscontinue/></isif>  <isif condition=*"${hummMaxOrderTotal != null}"*>  <isif condition=*"${paymentMethodType.value.equals('HUMM') && (pdict.OrderTotal > hummMaxOrderTotal)}"*>  <iscontinue/>  </isif>  </isif> |

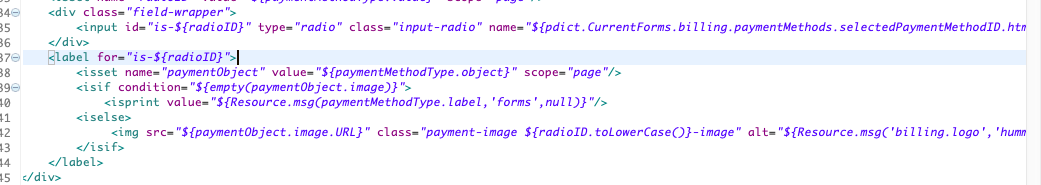


* In the file, **app\_storefront\_core/cartridge/templates/default/checkout/billing/paymentmethods.isml** add below lines, locate the below line.

|  |
| --- |
| <label for=*"is-${radioID}"*><isprint value=*"${Resource.msg(paymentMethodType.label,'forms',null)}"*/></label> |

and replace with below lines.

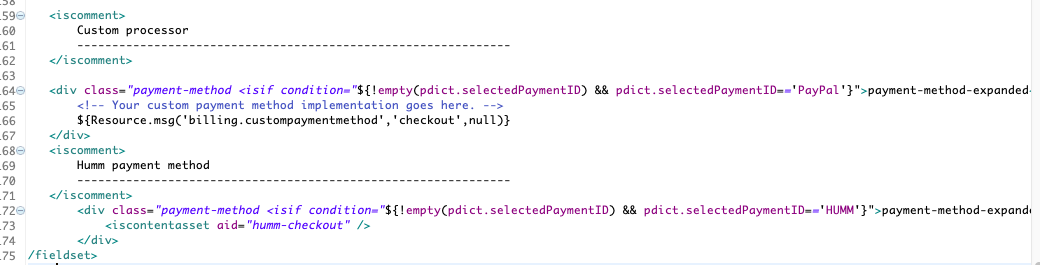
|  |
| --- |
| <label for=*"is-${radioID}"*>  <isset name=*"paymentObject"* value=*"${paymentMethodType.object}"* scope=*"page"*/>  <isif condition=*"${empty(paymentObject.image)}"*>  <isprint value=*"${Resource.msg(paymentMethodType.label,'forms',null)}"*/>  <iselse>  <img src=*"${paymentObject.image.URL}"* class=*"payment-image ${radioID.toLowerCase()}-image"* alt=*"${Resource.msg('billing.logo','humm',null)}"*/>  </isif>  </label> |



* In the file, **app\_storefront\_core/cartridge/templates/default/checkout/billing/paymentmethods.isml**

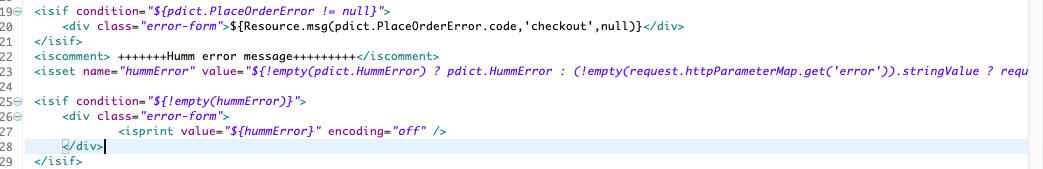
add the below mentioned div.

|  |
| --- |
| <iscomment>  Humm payment method  --------------------------------------------------------------  </iscomment>  <div class=*"payment-method <isif condition="*${!empty(pdict.selectedPaymentID) && pdict.selectedPaymentID=*=*'HUMM'}">payment-method-expanded</isif>" data-method="HUMM">  <iscontentasset aid=*"humm-checkout"* />  </div> |



* In the file, **app\_storefront\_core/cartridge/templates/default/checkout/summary/summary.isml** add below lines.

|  |
| --- |
| <iscomment> +++++++Humm error message+++++++++</iscomment>  <isset name=*"hummError"* value=*"${!empty(pdict.HummError) ? pdict.HummError : (!empty(request.httpParameterMap.get('error')).stringValue ? request.httpParameterMap.get('error').stringValue : null)}"* scope=*"page"*/>  <isif condition=*"${!empty(hummError)}"*>  <div class=*"error-form"*>  <isprint value=*"${hummError}"* encoding=*"off"* />  </div>  </isif> |

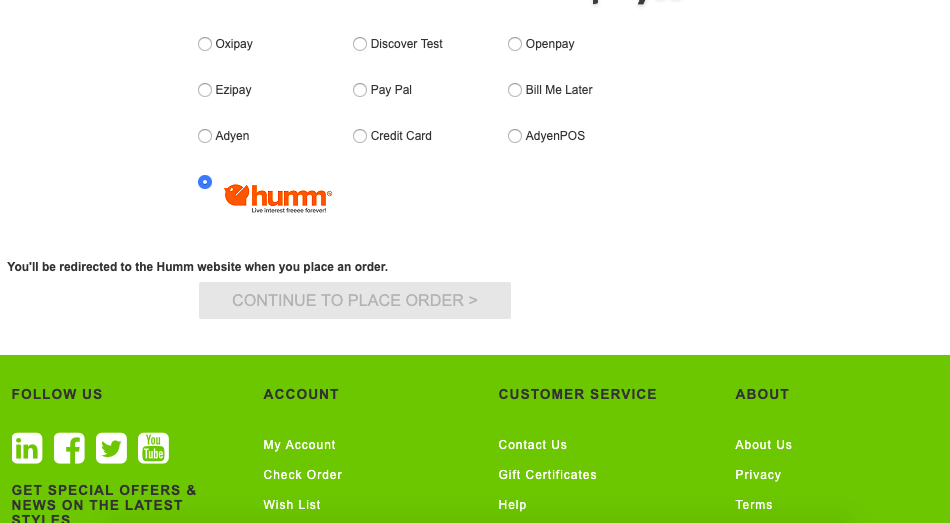


# Testing

Once the cartridge is installed and integrated based on instructions, please try to place an order on your sandbox to test the functionality.

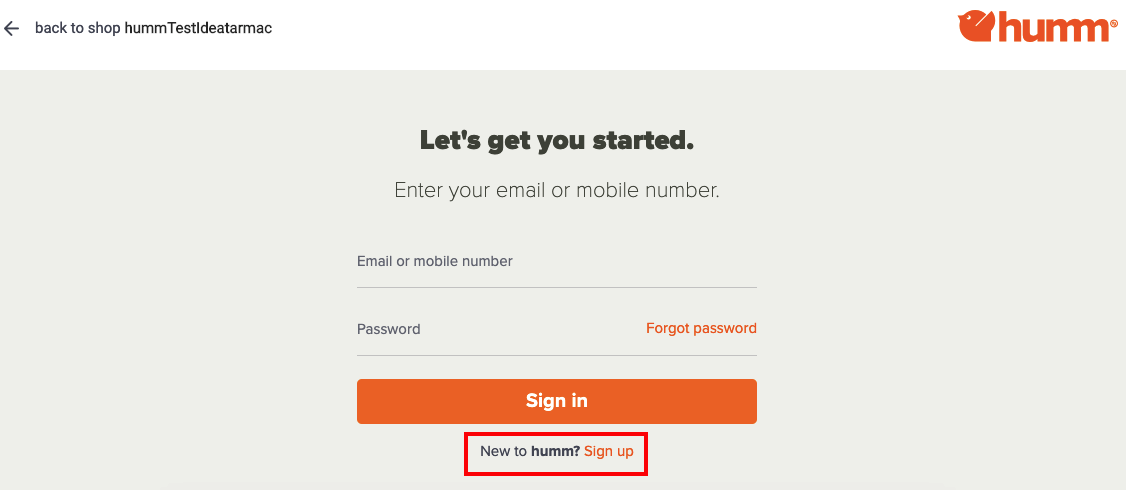
## Checkout data flow

1. If Humm integration is enabled and the order total is equal or greater than "Minimum Order Total" custom preference value and the order total is equal or less than "Maximum Order Total" custom preference value, then "Humm" payment method will visible in billing page.

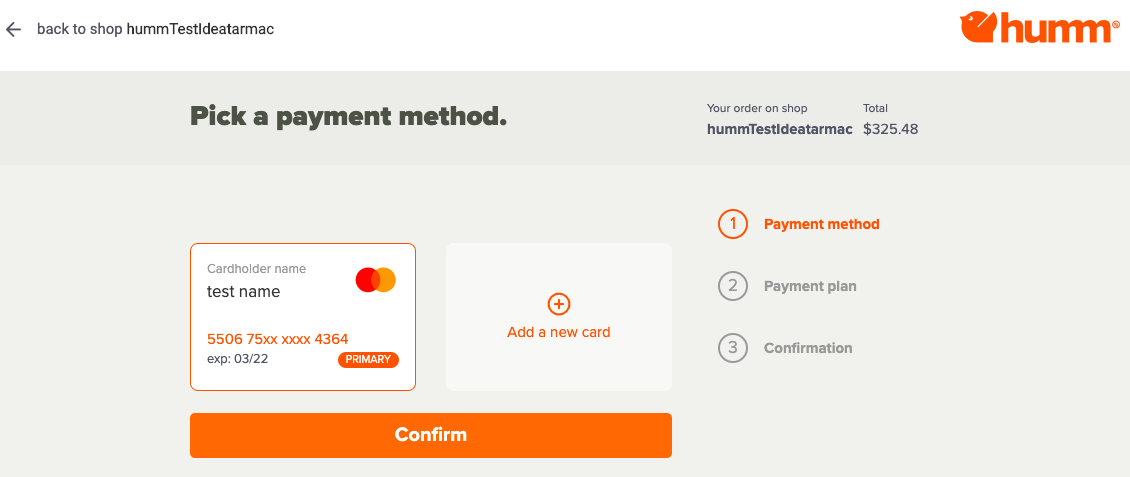


### Humm checkout

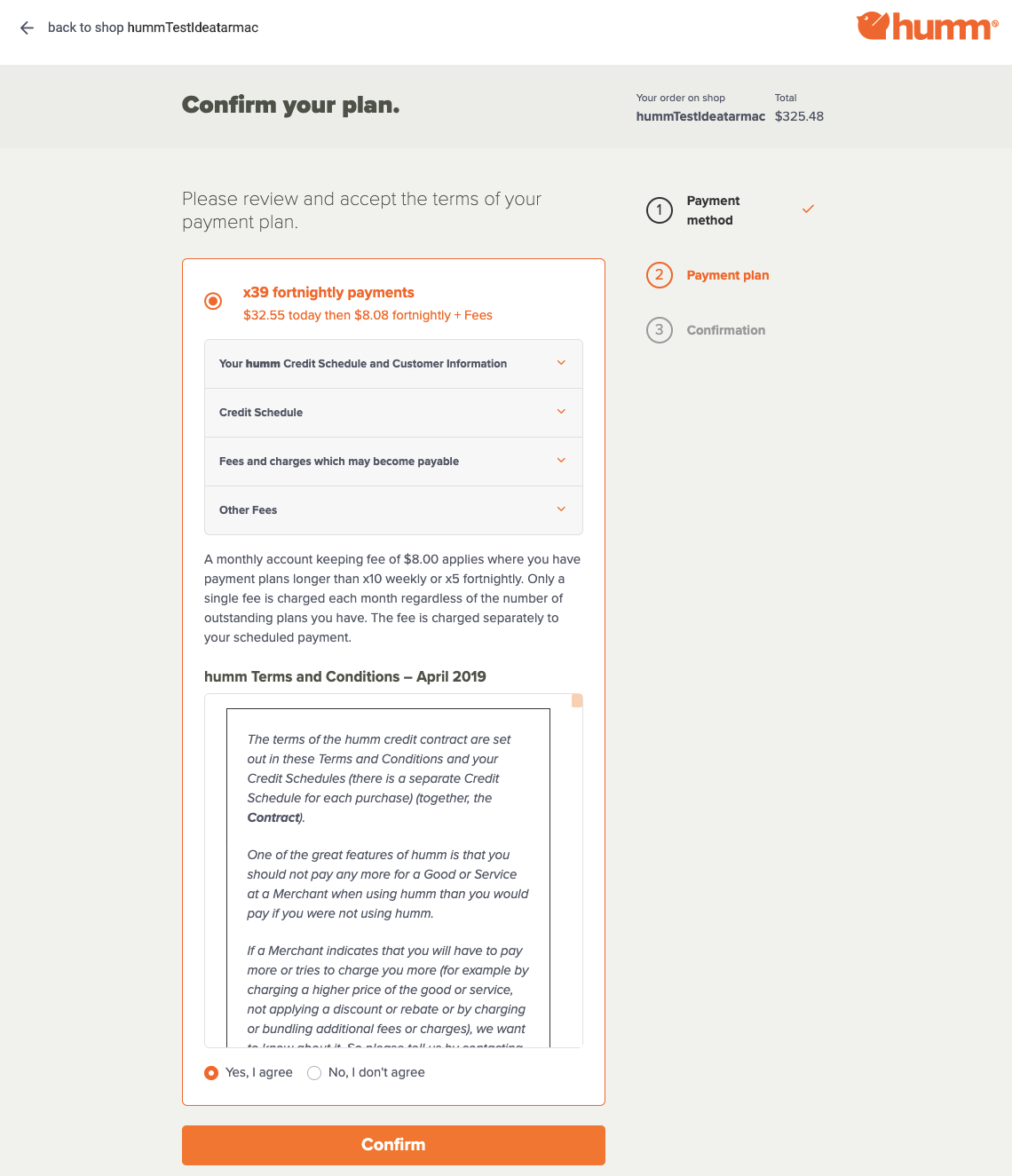
* After selecting payment method as "Humm" , click on "CONTINUE TO PLACE ORDER>" button on billing page.
* After you click on "PLACE ORDER" button on order summary page then you should be redirected to Humm website for choosing the right installment.
* If you have account then login into your account. Otherwise fill the register form to create an account.



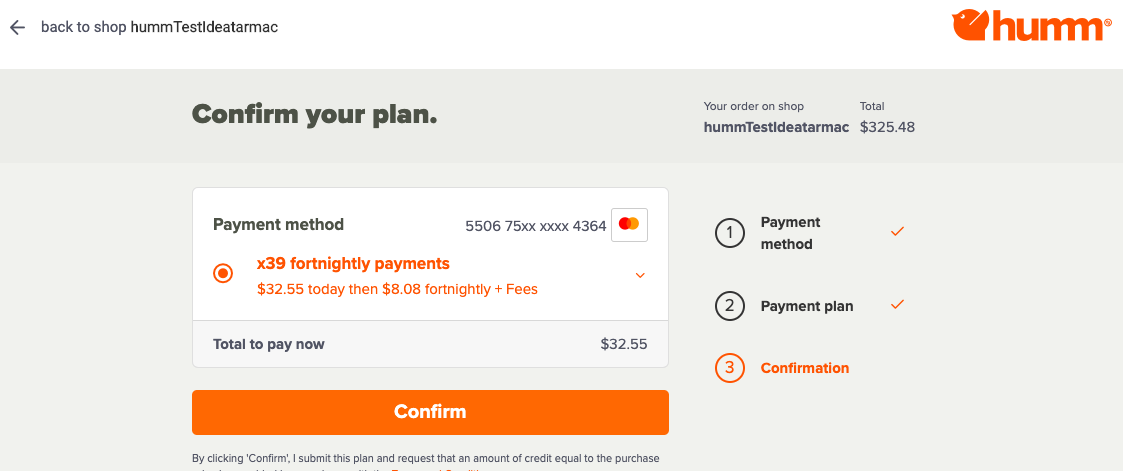
* After login into humm account, select the payment option and click on "Confirm" button.



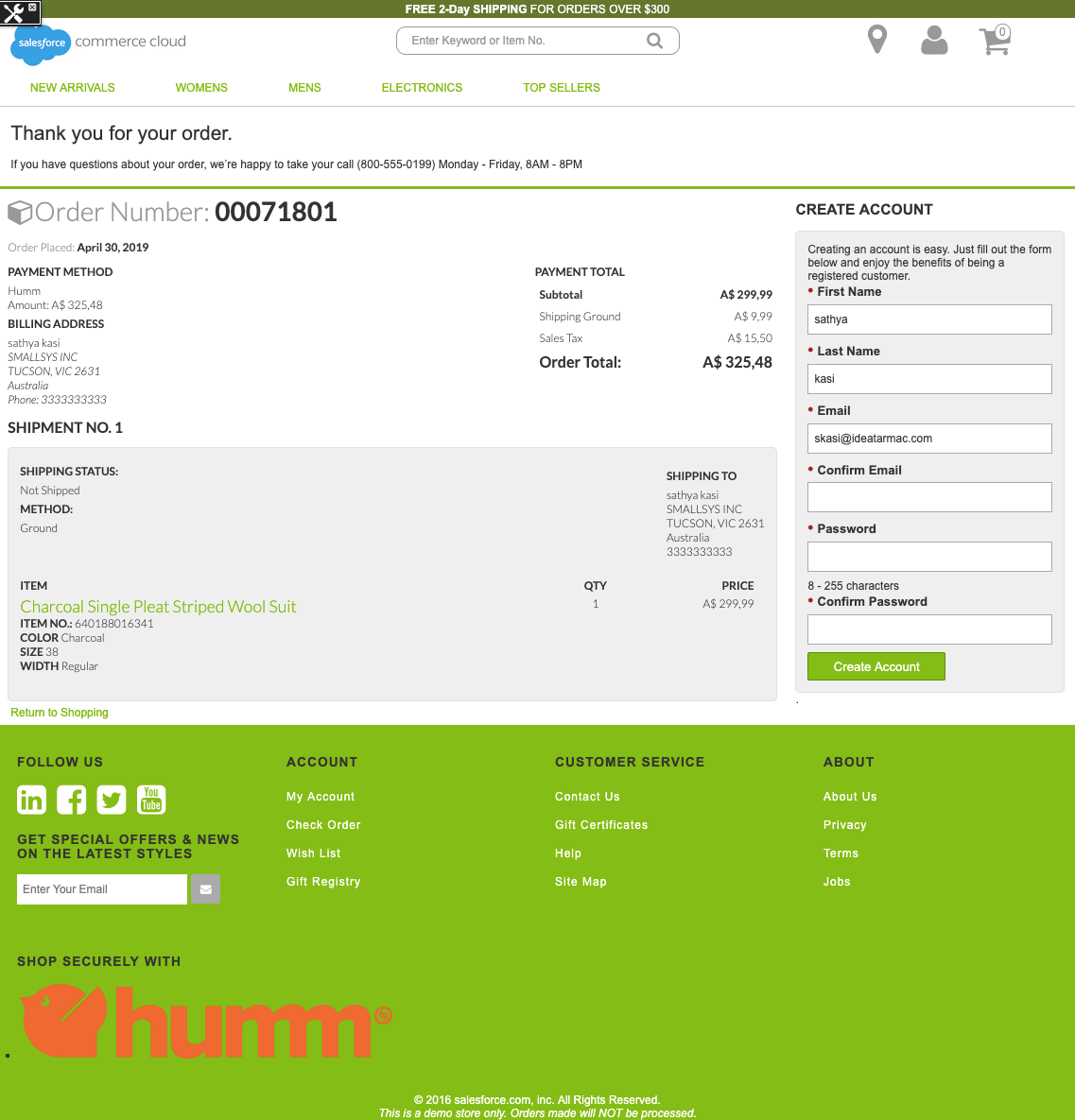
* Select the plan to pay, check ‘Yes I agree’ and click on confirm.



* Confirm the plan



* If payment is success, then you will be automatically redirected to Commerce Cloud storefront confirmation page like one below.



* After redirecting to Humm page, if you cancel the payment in middle, then you will be redirect to Commerce Cloud home page.

**Failover case:**

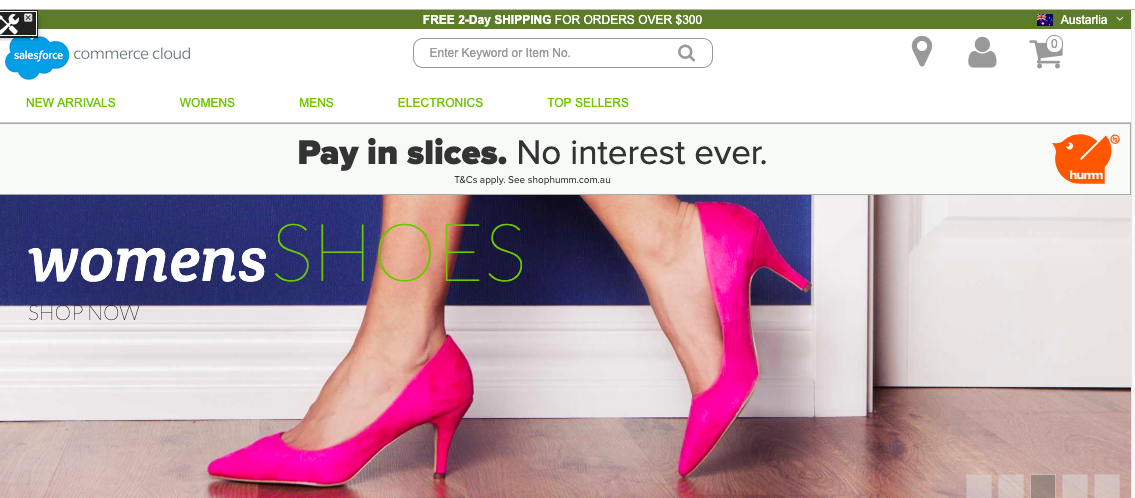
* When a transaction is declined then a declined message will be displayed and redirected back to the storefront.

**A screenshot of a cell phone

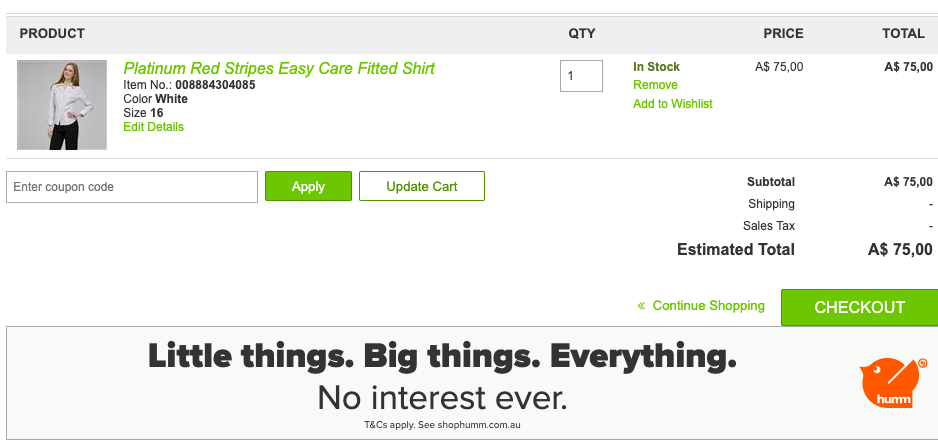
Description automatically generated**

## Sample Widgets

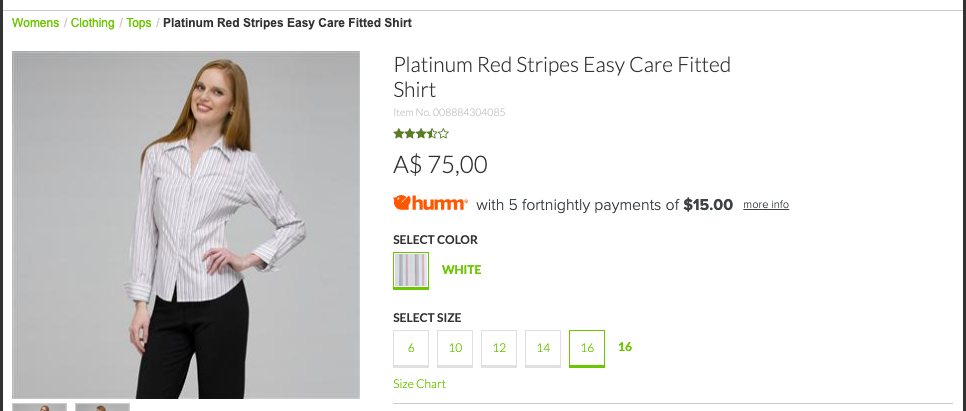
### Home Page Banner



### Cart Page Widget

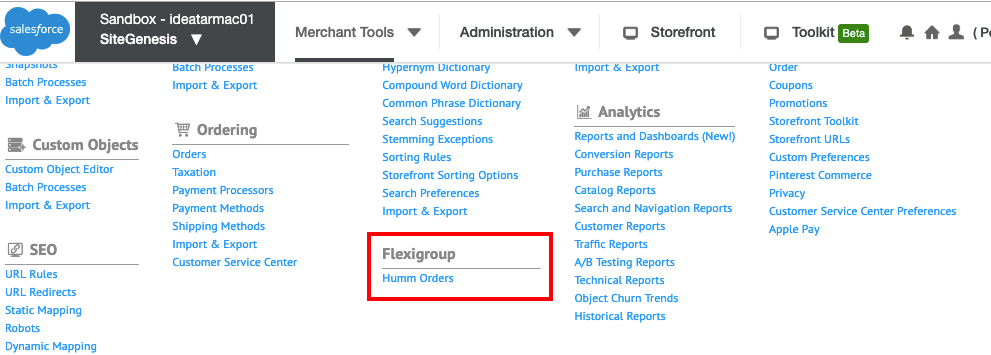


### Product Details Page Widget

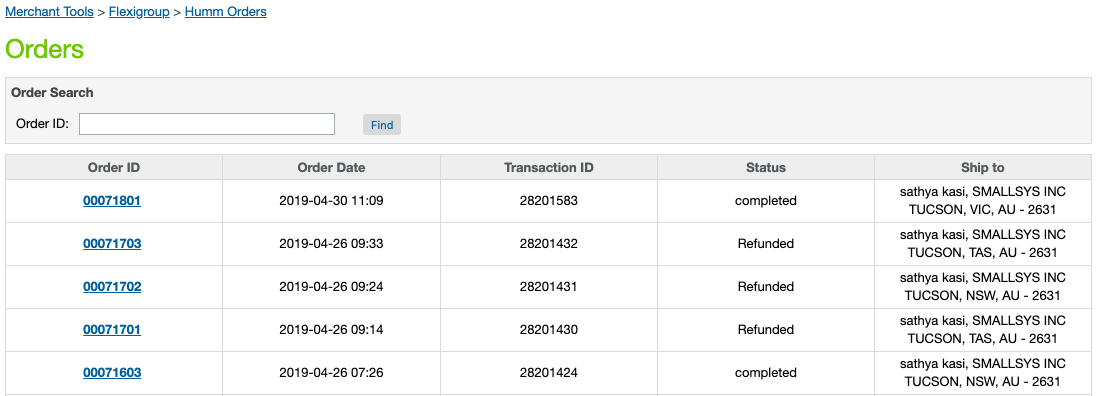


## Backend Transaction

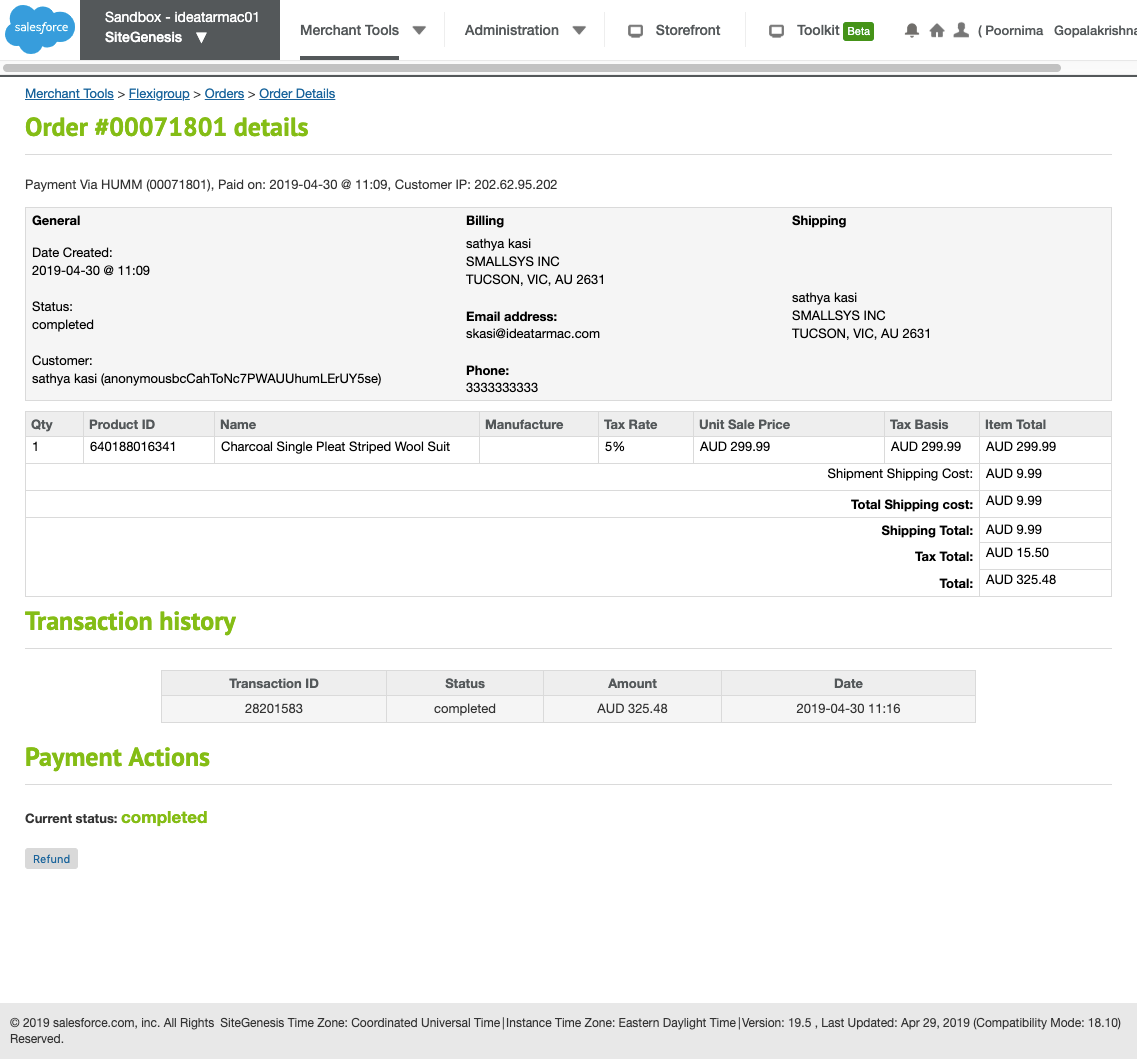
* Go to Merchant Tools and check if you see Flexigroup and its links as shown below:



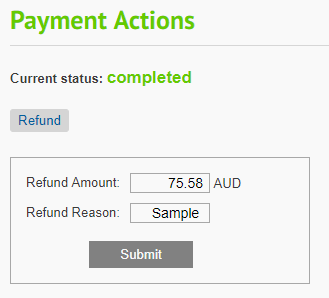
* On click of "Humm Orders", it will redirect to order list page.



* Click on "completed" order ID which you want to refund the transaction.



* Click on "Refund” button and enter the full amount or partial amount and refund reason. Click on "Submit" button.



# Known Issues

None

# Release History

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Changes** |
| 20.1.0 | 16-Jan-2020 | Initial release |