



KOMOJU Payment Gateway Integration Guide

FOR SALESFORCE COMMERCE CLOUD

Prepared By:

Cyntexa Labs Private Limited

✉ hello@cyntexa.com

🌐 www.cyntexa.com

Table of Contents

| | |
|--|-----------|
| 2. Implementation Guide | 3 |
| 2.1 Overview | 3 |
| 2.2 Compilation | 3 |
| 2.2.1 SFRA Compilation | 3 |
| 2.3 Installation | 4 |
| 2.3.1 Add cartridge to cartridge path - SFRA and SiteGenesis | 4 |
| 2.3.2 Import metadata archive | 5 |
| 2.3.3 BM Module installation | 6 |
| 2.4 Webhook Setup | 10 |
| 3. Operations and Maintenance | 11 |
| 3.1 Data Storage | 11 |
| 3.1.1 Custom attributes for Order object: | 11 |
| 3.1.2 Custom attributes for Basket object | 12 |
| 3.1.3 Custom attributes for OrderPaymentInstrument object | 12 |
| 3.1.4 Custom Objects | 13 |
| 3.2 HTTP Server Availability | 14 |
| 4. User Guide | 14 |
| 4.1 BM Module | 14 |
| 4.1.1 Schedule Jobs | 14 |
| 4.2 Storefront functionality | 17 |
| 4.2.1 Login | 17 |
| 4.2.2 Add Product to Cart and Proceed to Checkout | 18 |
| 4.2.3 Checkout Flow | 19 |
| 5. Locales and Currency support | 27 |
| 6. Version and SFRA support info | 29 |
| 7. Service Errors Handling | 30 |
| 7.1 Service Timeout handling | 30 |
| 7.2 Handle Error response | 31 |
| 8. Known Issues | 31 |
| 8.1 Test Cases | |
| In progress | 31 |
| 9. Failover/ Recovery process | 31 |

2. Implementation Guide

2.1 Overview

Cartridge Support integration:

SFRA

- int_komoju_common
- int_komoju_sfra
- bm_komoju
- app_storefront_base

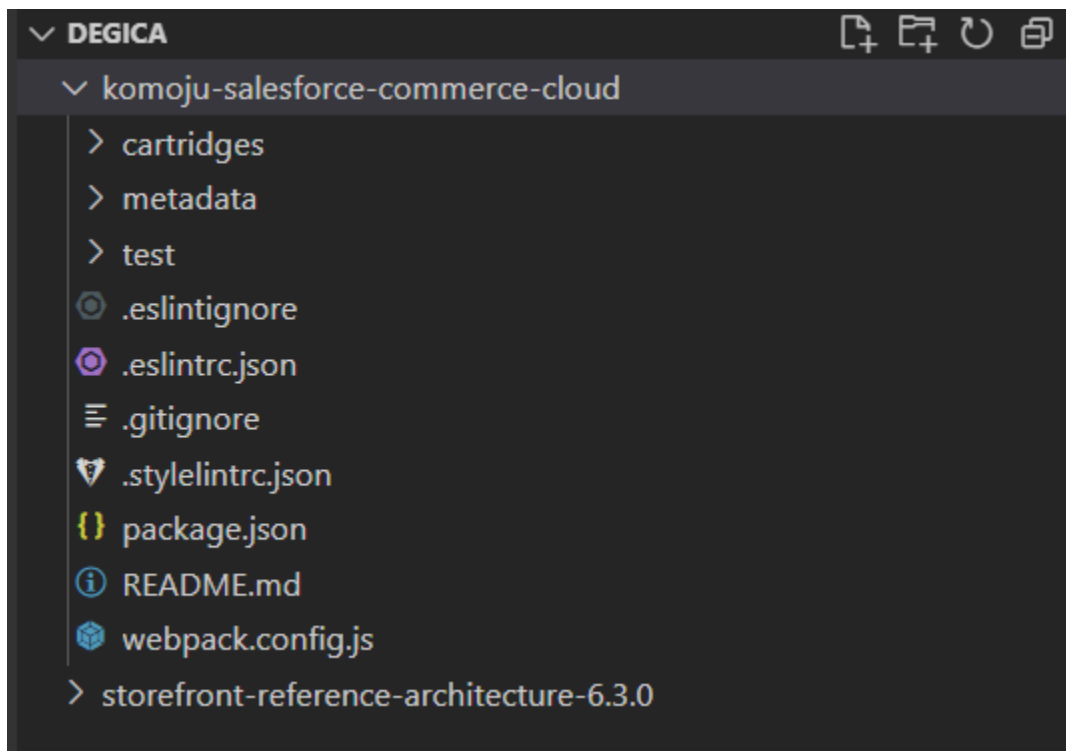
SiteGenesis

- bm_komoju
- int_komoju_common
- app_komoju_controllers
- app_komoju_core
- app_storefront_controllers
- app_storefront_core

2.2 Compilation

2.2.1 SFRA Compilation

- Clone the repository using the command prompt.
- Put Latest [Storefront Reference Architecture\(SFRA\)](https://github.com/orgs/SalesforceCommerceCloud/sso) parallel to the cloned directory.
(<https://github.com/orgs/SalesforceCommerceCloud/sso>)
- The cartridge structure should be as below:



- Make sure you have the node version 14.16.0
- cd to SFRA (storefront-reference-architecture)
- npm install
- Update `base` of `paths` in package.json if need be
- Go back to the root directory and run the following commands
- npm install
- npm run compile:js
- npm run compile:scss

2.2.2 SiteGenesis Compilation

- Clone the SiteGenesis Base Cartridge from <https://github.com/SalesforceCommerceCloud/sitegenesis>

- Copy `app_komoju_controllers` and `app_komoju_core` from KOMOJU cloned repository.
- Put them In the Parallel of `app_storefront_core` cartridge.
- Now Put this path in the `package.json` file.

```
• {  
  
  "src": "app_komoju_core/cartridge/scss/default/",  
    "dest": "app_komoju_core/cartridge/static/default/css/"  
  
}
```

```
• "ecstatic": "4.1.4",
```

```
• "grunt": "^1.4.1",
```

```
•
```

```
},  
  "devDependencies": {  
    "@tridnguyen/config": "^2.3.1",  
    "@tridnguyen/version": "^1.0.0",  
    "ampersand-model": "^6.0.2",  
    "ampersand-view": "^9.0.0",  
    "async": "^2.0.1",  
    "babel-core": "^6.1.2",  
    "babel-preset-es2015": "^6.1.2",  
    "chai": "^3.4.0",  
    "chalk": "^1.1.3",  
    "cldr-data": "^32.0.1",  
    "dwupload": "^3.8.0",  
    "ecstatic": "4.1.4",  
    "eslint": "^1.10.3",  
    "event-stream": "^3.3.4",  
    "exorcist": "^0.4.0",  
    "gaze": "^1.1.1",  
    "glob": "^7.0.6",  
    "globalize": "^1.1.1",  
    "grunt": "^1.4.1",  
    "handlebars": "^4.0.4",  
    "hbsfy": "^2.4.1",  
    "http-server": "^0.9.0",  
  }  
}
```

```
"css": [  
  {  
    "src": "app_storefront_core/cartridge/scss/default/",  
    "dest": "app_storefront_core/cartridge/static/default/css/"  
  },  
  {  
    "src": "app_storefront_core/cartridge/scss/fr_FR/",  
    "dest": "app_storefront_core/cartridge/static/fr_FR/css/"  
  },  
  {  
    "src": "app_storefront_core/cartridge/scss/it_IT/",  
    "dest": "app_storefront_core/cartridge/static/it_IT/css/"  
  },  
  {  
    "src": "app_storefront_core/cartridge/scss/ja_JP/",  
    "dest": "app_storefront_core/cartridge/static/ja_JP/css/"  
  },  
  {  
    "src": "app_storefront_core/cartridge/scss/zh_CN/",  
    "dest": "app_storefront_core/cartridge/static/zh_CN/css/"  
  },  
  {  
    "src": "app_komoju_core/cartridge/scss/default/",  
    "dest": "app_komoju_core/cartridge/static/default/css/"  
  }  
],
```

- Run the following commands with the recommended node version 12.21.0

NOTE: For SiteGenesis the app_storefront_controllers base cartridge also needs to be modified. For this navigate to app_storefront_controllers>cartridge>scripts>app.js and replace the '~' symbol with '*'.

NOTE: To compile other cartridges together with KOMOJU cartridge please refer to <https://github.com/SalesforceCommerceCloud/sfra-webpack-builder>

- npm i
- npm run build

2.3 Installation

2.3.1 Add cartridge to cartridge path - SFRA and SiteGenesis

Add `bm_komoju`, `int_komoju_sfra`, `int_komoju_common`, and `app_storefront_base` for SFRA and `app_komoju_controllers`, `app_komoju_core`, `int_komoju_common`, `bm_komoju`, `app_storefront_core`, `app_storefront_controllers` for SiteGenesis into the cartridge path of a target site.

- To do so, go to **Business Manager > Administration > Sites > Manage Sites > Your Site > Settings**
- Add cartridges in the following order for the **SFRA** site

`bm_komoju:int_komoju_sfra:int_komoju_common:app_storefront_base`

- Add cartridges in the following order for **SiteGenesis** site

`bm_komoju:int_komoju_common:app_komoju_controllers:app_komoju_core:app_storefront_controllers:app_storefront_core`

2.3.2 Import metadata archive

- Upload and import `metadata.zip` from the metadata folder (extract it first) of the `link_cartridge` (note-change the folder name (`metadata` → `sites` → `RefArch`) with (`metadata` → `sites` → `<yoursite>`)).
- Make the zip folder again and upload it to the BMs site import and export.

- To do so, go to **Business Manager > Administration > Site Development > Site Import & Export**.
- Upload the archive using the Local option in the Upload Archive section.
- After uploading, choose <metadata file name>.zip in the list and click on the import button.

Site Import & Export

This page allows you to export the current configuration of your organization including all of its sites. To download an archive, just click its file name.

Import

You can delete a file only if it's a local file or if you're logged into the instance where the file was created.

Upload Archive:

☒ Local ☐ Remote

No file chosen

| Select | Name ▲ | Location | Delete Allowed? | File Size | Last Modified |
|----------------------------------|---|----------|-----------------|-----------|---------------------|
| <input checked="" type="radio"/> | metadata.zip | local | Yes | 255.08 KB | 11/2/22 6:10:12 am |
| <input type="radio"/> | instance/komoujuseiteexport.zip | local | Yes | 1.04 MB | 11/2/22 5:59:11 am |
| <input type="radio"/> | instance/metadata.zip | local | Yes | 1.41 KB | 10/18/22 2:28:23 am |
| <input type="radio"/> | instance/refarchSite.zip | local | Yes | 50.92 KB | 10/20/22 8:35:04 am |
| <input type="radio"/> | instance/stripe_site_template.zip | local | Yes | 7.37 KB | 10/18/22 5:23:27 am |
| <input type="radio"/> | SiteGenesis Demo Site | | | | |
| <input type="radio"/> | Storefront Reference Architecture Demo Sites | | | | |

Figure: 2

- **Administration > Site Development >> Code Deployment.**

Reinitialize the active code version in which the cartridge resides.

To do so, activate any other code version and then reactivate your code version in which the KOMOJU cartridges reside.

- Upload cartridge via Prophet Debugger on VSCode

2.3.3 BM Module installation

Upload the cartridge bm_komoju to the current code version and add bm_komoju to the cartridge path of the business manager and your site.

- To do so, go to **Business Manager > Administration > Sites > Manage Sites > Manage the Business Manager site > Settings**
- Insert bm_komoju before your cartridge's record, as shown below.

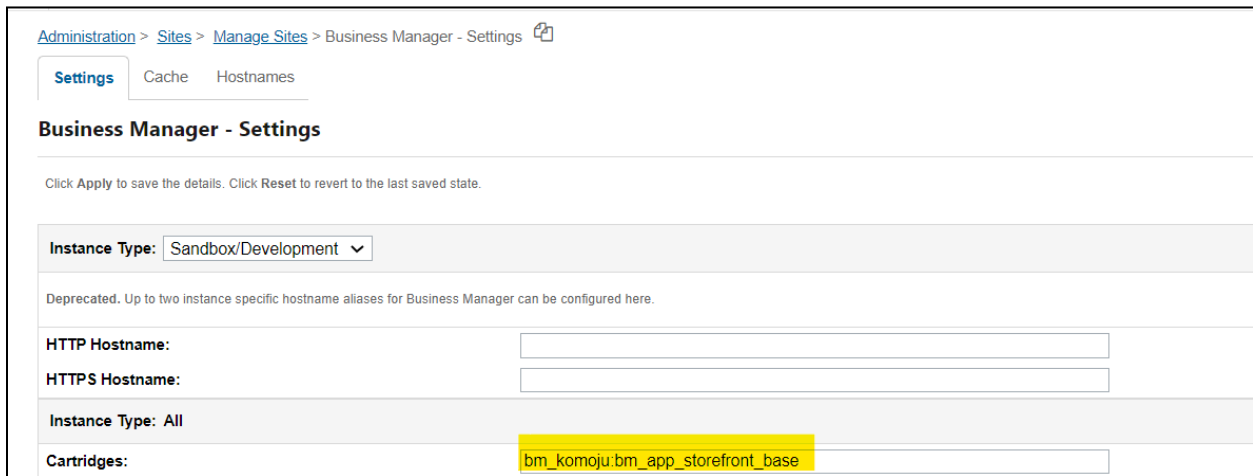


Figure:3

- For the site, go to **Business Manager > Administration > Sites > Manage Sites > Your Site > Settings**
- Insert bm_komoju before your cartridge's record, as shown below

General **Settings** Cache Site Status Page Meta Tag Rules

RefArch - Settings

Click **Apply** to save the details. Click **Reset** to revert to the last saved state.

Instance Type:

Deprecated. The preferred way of configuring HTTP and HTTPS hostnames is by using new features of the site aliases configuration ("SEO > Aliases Configuration"). The HTTP/HTTPS hostna configuration style.

HTTP Hostname:

HTTPS Hostname:

Instance Type: All

Cartridges:

Figure: 4

- Now go to **Administration > Organization > Roles & permissions > Select Role > Business Manager Modules > Select Site > enable Komoju Payment Settings > update**

| | | | | |
|--------------------------------|--|--|-------------------------------------|-------------------------------------|
| Ordering | | | | |
| Orders | Manage the orders of this site. Please note that for read-only access across the entire "Orders" module the functional permission 'Delete_Order_Note' is required in the site context. | | | |
| KOMOJU Payment Settings | This extension allows a merchant to enable or disable komoju payment methods. | | | <input checked="" type="checkbox"/> |
| Taxation | Manage taxation settings of this site. | | | <input checked="" type="checkbox"/> |
| Payment Processors | Manage the payment processors of this site. | | | <input checked="" type="checkbox"/> |
| Payment Methods | Manage the payment methods of this site. | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Shipping Methods | Manage the shipping methods of this site. | | | <input checked="" type="checkbox"/> |
| Import & Export | Import and export ordering data of this site. | | | <input checked="" type="checkbox"/> |
| Customer Service Center | Capture orders on behalf of a customer. | | | <input checked="" type="checkbox"/> |

Figure: 5

- Now you will see the module under **Merchant Tools>Ordering>KOMOJU Payment Settings**
- Click on KOMOJU Payment Settings and you will see all the KOMOJU-related configurations here
- KOMOJU payment methods might not be visible here so first, you need to update the KOMOJU secret key given here and then run the update payment method job
- To do so
Administration>Operations>Jobs>updatePaymentMethod>

- The scope of the job step can also be changed to one or more specified sites. This can be done by navigating to the Job Steps section (But remember, whatever the sites we select in the job scope, should contain the metadata and the secret key value)

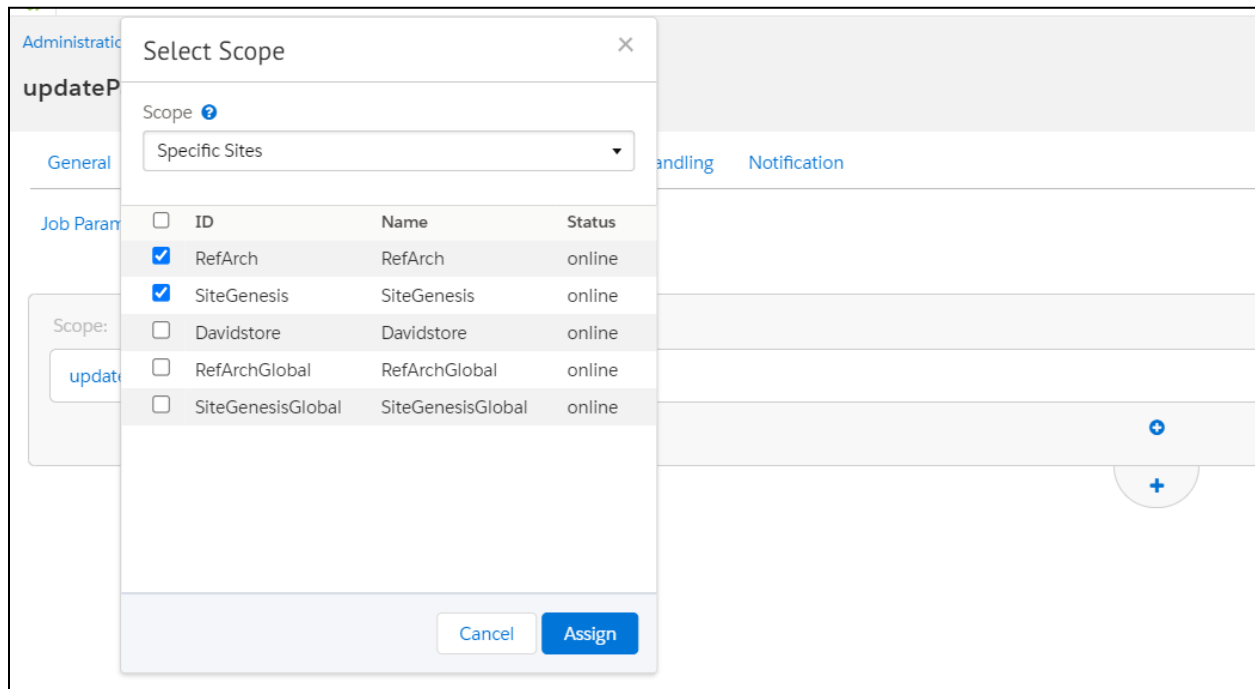


Figure: 6

- Now run the job by clicking on **Run Now**

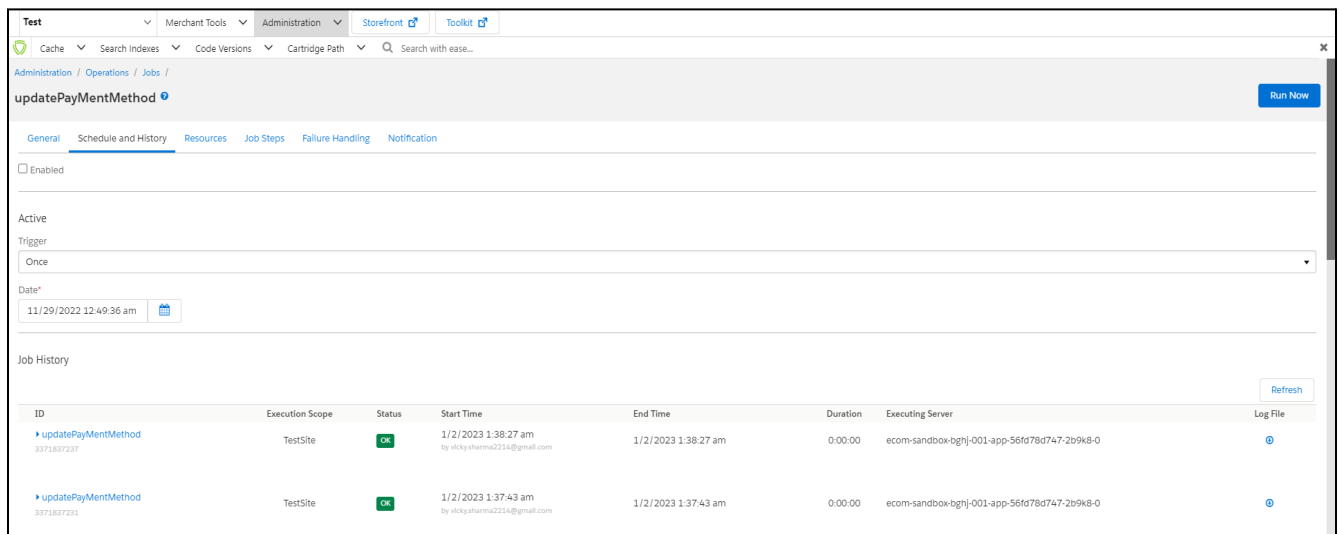







Figure: 7


- Now all the payment methods are visible to you, you can enable and disable the payment methods with the help of given toggle button
- You have access to the email functionality as well, which is used for sending emails to the registered email ID, whenever orders records fail to update during the refund or cancel process.
- You can enable or disable this functionality as well
- Now you have the webhook authentication code as well which is to be updated with webhooks registered with the KOMOJU platform.
- This code should be matched with all the authentication codes of all webhooks in order to successfully update the order record after a webhook call.
- You have the Webhook URL also visible here, these need to be created on the KOMOJU platform.
- After all the configurations the BM module should look as shown below:

Ordering > KOMOJU Payment Settings

Payment Methods

*Choose the payment methods you want to display on check out page

| | | | |
|---|---------------|-----|-------------------------------------|
|  | Konbini | JPY | <input checked="" type="checkbox"/> |
|  | Credit Card | JPY | <input checked="" type="checkbox"/> |
|  | Pay Easy | JPY | <input checked="" type="checkbox"/> |
|  | Bank Transfer | JPY | <input checked="" type="checkbox"/> |
|  | WebMoney | JPY | <input checked="" type="checkbox"/> |

 PayPay

JPY

☒

KOMOJU Secret Key

*Find your secret key in [KOMOJU admin page](#)

Secret Key :

Update

Job failure alert

*This email address is used to send alert for the failure of the job to synchronize cancel order to KOMOJU for payment cancel or refund was failed.

Email :

Update

Enable/Disable job failure alert

☒

KOMOJU webhook authentication

*Setup the same secret code on KOMOJU webhook setting page

Secret Token :

Update

KOMOJU webhook setting

*Setup following webhooks on [KOMOJU admin page](#)

| | | |
|--------------------|---|------|
| payment.refunded | https://zyhe-001.dx.commercecloud.salesforce.com/on/demandware.store/Sites-RefArch-Site/default/KomojuController-HandleWebHooksRefund | Copy |
| payment.captured | https://zyhe-001.dx.commercecloud.salesforce.com/on/demandware.store/Sites-RefArch-Site/default/KomojuController-HandleWebHooksCaptureComplete | Copy |
| payment.cancelled | https://zyhe-001.dx.commercecloud.salesforce.com/on/demandware.store/Sites-RefArch-Site/default/KomojuController-HandleWebHooksCancelled | Copy |
| payment.expired | https://zyhe-001.dx.commercecloud.salesforce.com/on/demandware.store/Sites-RefArch-Site/default/KomojuController-HandleWebHooksExpired | Copy |
| payment.authorized | https://zyhe-001.dx.commercecloud.salesforce.com/on/demandware.store/Sites-RefArch-Site/default/KomojuController-HandleWebHooksPaymentAuthorized | Copy |

Figure: 8

2.3.4 Uninstallation

To Uninstall the KOMOJU Link Cartridge use the following Steps:

- Remove All Cartridges from the Storefront site
Administration > Sites > Manage Site > click on your site name > settings > cartridges
- Remove the bm_komoju cartridge from the business manager site
Administration > Sites > Manage Site > Business Manager Site > Settings > cartridges
- Remove Custom Object instance
Merchant Tools > Custom Objects > Import & Export > Upload metadata only for the custom object (komojuPaymentMethodsObjectType.xml) > Now go to Import > select the recent uploaded file > next > now import as delete.
- Remove Payment Method
Merchant Tools > Ordering > Import & Export > Upload metadata only for Payment Method (payment-methods.xml) > Now go to Import Payment Methods > select the recently uploaded file > next > now import as delete.
- Remove Jobs
Administration > Operations > Import & Export > Upload > Upload meta Data only for Jobs (jobs.xml) > Jobs > Import > select the recently uploaded file > next > now import as delete.
- Remove Services
Administration > Operations > Services > Select KOMOJUService, KOMOJUCancelAndRefundService and KOMOJUGetPaymentMethods > Delete
- Remove Payment Processor
Merchant Tools > Ordering > Payment Processors > Select KOMOJU >

Delete

- Remove Custom Object Type
Administration > Site Development > Custom Object Types > Select komojuPaymentMethodsObjectType > Delete
- Remove Custom Attributes on system object type one by one
Administration > Site Development > System Object Types >
 - Basket > Attribute Definitions > Search KOMOJU > Select All > Delete.
 - Order > Attribute Definitions > Search KOMOJU > Select All > Delete.
 - Order Payment Instrument > Attribute Definitions > Search KOMOJU > Select All > Delete.

2.4 Webhook Setup

To create a webhook go to the KOMOJU platform and click on **Webhooks> New Webhook** and give the Webhook URL, Secret (i.e. authentication code), Event (Refund, Cancel, etc), and activate this.

Take all this information from the BM module **KOMOJU payment** settings, configured in the Business Manager.

* Webhook URL

https://bghj-001.dx.commercecloud.salesforce.com/on/demandware.store/Sites-TestSite-Site/default/KomojuController-HandleWebHooksRefu

* Secret

cyntexa

Please only use letters, numbers and any of !, #, \$, @.

☐ ping
 ☐ customer.created
 ☐ customer.updated
 ☐ customer.deleted
 ☐ payment.authorized
 ☐ payment.captured
 ☐ payment.updated
 ☐ payment.expired
 ☐ payment.cancelled
 ☐ payment.refund.created
 ☒ payment.refunded
 ☐ payment.failed
 ☐ payout.created
 ☐ subscription.created
 ☐ subscription.captured
 ☐ subscription.failed
 ☐ subscription.suspended
 ☐ subscription.deleted
 ☐ settlement.created
 ☐ disbursement.created
 ☐ disbursement.cancelled
 ☐ disbursement.failed
 ☐ disbursement.paid

Event list

☒ Active
 Deliver event when the selected events are triggered.

Figure: 9

Update the messages

To update messages in SFRA, go to
 cartridges\int_komoju_sfra\cartridge\templates\resources\komojuPayment.properties in the code and Update.

```

cartridges > int_komoju_sfra > cartridge > templates > resources > komojuPayment.properties
1  label.description=Description:
2  label.close=Close
3  label.multiship=multi-ship
4  method.name=KOMOJU
5  method.not.supported=Your selected payment method or the shipping method is not supported. Please select another
6  error.connecting.komoju=There was some error in connecting to KOMOJU
7  returned.back.komoju=You returned back from KOMOJU servers, please go again to complete the payment.
8  cancelled.order.komoju=You cancelled the order
9  order.created.komoju=You order is pending, Please complete the payment on KOMOJU
10 order.captured.komoju=Thanks, Your Order is placed and payment captured successfully
11 amount.refunded.komoju.message=This Amount has been successfully refunded on KOMOJU Servers
12 amount.refunded.komoju=Refunded
13 amount.cancelled.komoju=Cancelled
14 amount.cancelled.komoju.message=This order has been successfully cancelled on KOMOJU Servers
15 no.payment.method=Please run "update payment method job" to get payment method from KOMOJU
16 authentication.code.mismatch=Error occurred in {0} event due to authentication code mismatch and the order {1} W
17
    
```


To update messages in SiteGenesis, go to `app_komoju_core\cartridge\templates\resources\komojuPayment.properties` in the code and Update.

```
app_komoju_core > cartridge > templates > resources > * komojuPayment.properties
1 method.not.supported=Your selected payment method or the shipping method is not supported. Please select an
2 returned.back.komoju=You returned back from KOMOJU servers, please go again to complete the payment.
3 order.created.komoju=You order is pending, Please complete the payment on KOMOJU
4 order.captured.komoju=Thanks, Your Order is placed and payment captured successfully
5 cancelled.order.komoju=You cancelled the order
6 amount.refunded.komoju.message=This Amount has been successfully refunded on KOMOJU Servers
7 amount.refunded.komoju=Refunded
8 amount.cancelled.komoju=Cancelled
9 amount.cancelled.komoju.message=This order has been successfully cancelled on KOMOJU Servers
10 no.payment.method=Please run "update payment method job" to get payment method from KOMOJU
11 authentication.code.mismatch=Error occurred in {0} event due to authentication code mismatch and the order {
12 error.connecting.komoju=There was some error in connecting to komoju You, 38 minutes ago • Uncommitted
```

To update messages in KOMOJU Payment Settings page, go to `cartridges\bm_komoju\cartridge\templates\resources\komojuPaymentMethods.properties` in the code and Update.

```
cartridges > bm_komoju > cartridge > templates > resources > * komojuPaymentMethods.properties
1 komoju.payment.methods=Payment Methods
2 toggle.komoju.methods=*Choose the payment methods you want to display on check out page
3 komoju.secret.head=KOMOJU Secret Key
4 komoju.secret.title=Secret Key :
5 komoju.secret.descFirst=*Find your secret key in
6 komoju.secret.descSecond= KOMOJU admin page
7 komoju.secret.note=Please notice the secret key under Live model and Test model will be different
8 komoju.secret.update=Update
9 komoju.email.head=Job failure alert
10 komoju.email.title=Email :
11 komoju.email.desc=*This email address is used to send alert for the failure of the job to synchronize cancel on
12 komoju.email.update=Update
13 komoju.email.enable=Enable/Disable job failure alert
14 komoju.authentication.head=KOMOJU webhook authentication
15 komoju.authentication.title=Secret Token :
16 komoju.authentication.desc=*Setup the same secret code on KOMOJU webhook setting page
17 komoju.authentication.update=Update
18 please.update.secret.run.job=Please update the KOMOJU secret key and then run the update payment method job in
19 komoju.webhook.list=KOMOJU webhook setting
20 komoju.webhook.descFirst=*Setup following webhooks on
21 komoju.webhook.descSecond=KOMOJU admin page
22 komoju.copy.webhook=Copy
23 komoju.copy.message=WebHook URL was copied to the clipboard.
24 komoju.webhook.url=https://{0}/on/demandware.store/Sites-{1}-Site/default/KomojuController-{2}
25 please.import.metadata=Please import the metadata to access the KOMOJU Payment Settings
26 komoju.menuaction.name=KOMOJU Payment Settings
```

3. Operations and Maintenance

3.1 Data Storage

The integration requires System Objects Extension to store partner-related information.

3.1.1 Custom attributes for Order object:

- **KOMOJU Session URL:** This custom attribute stores the session URL of the hosted page once a customer submits for payment
- **UUID:** This custom attribute stores the UUID of the hosted page once a customer submits for payment
- **KOMOJU Refund Status:** Enum of strings storing the status of the refund.
- **KOMOJU Cancel Status:** Enum of strings that stores the cancellation status of an order.
- **KOMOJU Refund Response:** This attribute contains the response of the refund API call in that order.
- **KOMOJU Cancel Response:** This attribute contains the response of the cancel API call in that order.
- **KOMOJU Order Processed:** This attribute specifies that the order has been forwarded to KOMOJU to create the transaction record.
- **KOMOJU Payment Id:** This custom attribute on an order contains the payment UUID associated with the KOMOJU session id on a particular order.

- **KOMOJU Payment Method Type:** It stores the payment method selected by the user, for example, Konbini, Bank transfer, etc.
- **KOMOJU Exchange Rate:** It stores the exchange rate of the currency of the payment method selected with respect to the storefront currency.

3.1.2 Custom attributes for Basket object

- **KOMOJU Payment Method Type:** It stores the payment method selected by the user, for example - Konbini, Bank transfer, etc.
- **KOMOJU Session URL:** It holds the KOMOJU session URL in the basket.
- **UUID:** It holds the KOMOJU session UUID in the basket.

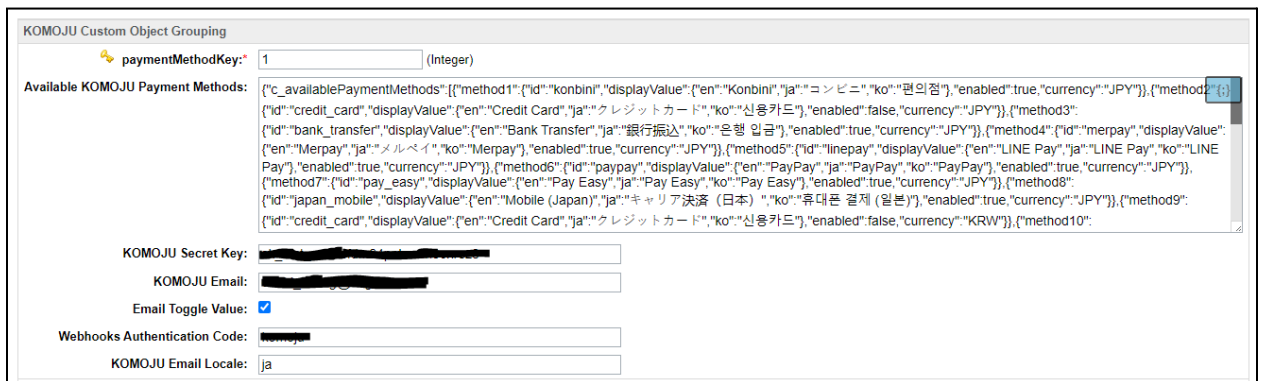
3.1.3 Custom attributes for OrderPaymentInstrument object

- **KOMOJU Payment Method Type:** It stores the particular method selected by the customer among all komoju methods (i.e. Konbini, Paypay, Merpay, etc).
- **Prepaid Card Last Digits:** It stores the prepaid card's last 4 digits numbers.
- **Brand:** It stores brands associated with a credit card.
- **Store:** It contains the store selected by the customer in case of the konbini payment method.
- **Transaction Status:** It stores the transaction status of the payment.

- **KOMOJU Payment Id:** It stores the payment id received from komoju.
- **KOMOJU Exchange Rate:** It stores the exchange rate of the currency of the payment method selected with respect to the storefront currency.
- **KOMOJU Processing Currency:** It stores the currency in which the customer makes a payment on KOMOJU HOSTED PAGE.
- **KOMOJU Exchange Amount:** It stores the amount converted from storefront currency to the KOMOJU payment method selected currency.
- **KOMOJU Processing Fee:** It stores the processing fee of the payment method selected.

3.1.4 Custom Objects

- **KOMOJU Payment Methods Object Type:** This type is used to store the KOMOJU payment methods, KOMOJU secret key, the email to send an email, webhook authentication code, and KOMOJU Email Locale to store the current BM Locale. All these values get updated from the BMmodule configured.



KOMOJU Custom Object Grouping

paymentMethodKey: 1 (Integer)

Available KOMOJU Payment Methods:

```
{
  "c_availablePaymentMethods": [
    {
      "method1": {
        "id": "konbin",
        "displayValue": {
          "en": "Konbin",
          "ja": "コンビニ",
          "ko": "편의점",
          "enabled": true,
          "currency": "JPY"
        },
        "method2": {
          "id": "credit_card",
          "displayValue": {
            "en": "Credit Card",
            "ja": "クレジットカード",
            "ko": "신용카드",
            "enabled": false,
            "currency": "JPY"
          },
          "method3": {
            "id": "bank_transfer",
            "displayValue": {
              "en": "Bank Transfer",
              "ja": "銀行振込",
              "ko": "은행 입금",
              "enabled": true,
              "currency": "JPY"
            },
            "method4": {
              "id": "merpay",
              "displayValue": {
                "en": "Merpay",
                "ja": "メルペイ",
                "ko": "Merpay",
                "enabled": true,
                "currency": "JPY"
              },
              "method5": {
                "id": "linepay",
              "displayValue": {
                "en": "LINE Pay",
                "ja": "LINE Pay",
                "ko": "LINE Pay",
                "enabled": true,
                "currency": "JPY"
              },
              "method6": {
                "id": "paypay",
                "displayValue": {
                  "en": "PayPay",
                  "ja": "PayPay",
                  "ko": "PayPay",
                  "enabled": true,
                  "currency": "JPY"
                },
                "method7": {
                  "id": "pay_easy",
                  "displayValue": {
                    "en": "Pay Easy",
                    "ja": "Pay Easy",
                    "ko": "Pay Easy",
                    "enabled": true,
                    "currency": "JPY"
                  },
                  "method8": {
                    "id": "japan_mobile",
                    "displayValue": {
                      "en": "Mobile (Japan)",
                      "ja": "キャリア決済 (日本)",
                      "ko": "휴대폰 결제 (일본)",
                      "enabled": true,
                      "currency": "JPY"
                    },
                    "method9": {
                      "id": "credit_card",
                      "displayValue": {
                        "en": "Credit Card",
                        "ja": "クレジットカード",
                        "ko": "신용카드",
                        "enabled": false,
                        "currency": "KRW"
                      },
                      "method10": {

```

KOMOJU Secret Key: [Redacted]

KOMOJU Email: [Redacted]

Email Toggle Value: ☒

Webhooks Authentication Code: [Redacted]

KOMOJU Email Locale: ja

Figure: 10

3.2 HTTP Server Availability

- You can track availability and downtime by service status in the Commerce Cloud Business Manager.
- Go to **Administration > Operations > Services > service_name**.
- **KOMOJUService**: This service is used to create a hosted page session, get the details of the session at any point of the checkout flow and cancel the session as well.
- **KOMOJUCancelAndRefundService**: This service is used to request the refund and cancel on the KOMOJU's end.
- **KOMOJUGetPaymentMethods**: This service is used to get the latest payment methods from komoju servers.

4. User Guide

4.1 BM Module

4.1.1 Schedule Jobs

UpdatePaymentMethod: It will call the KOMOJU API to fetch available payment methods and update the custom object.

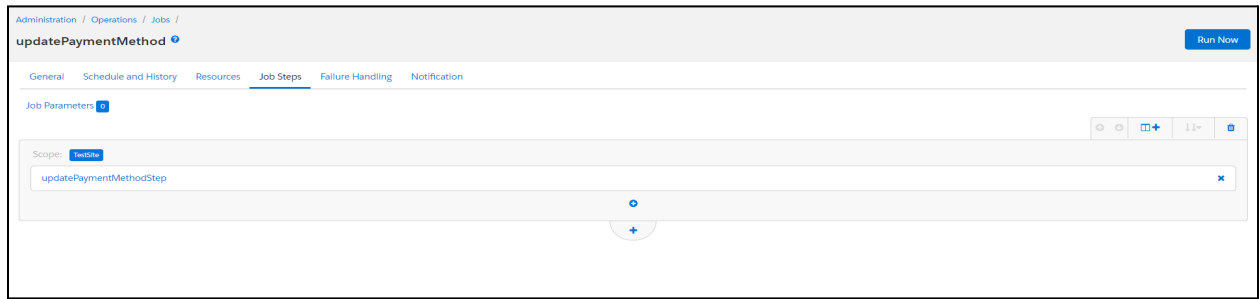


Figure: 11

The custom object will be updated according to the API response and the attribute 'Available Komoju Payment Methods' will be updated with the stringified text containing payment methods and their details.

NOTE: If there are already payment methods stored in the custom attribute then this job won't change the status of those payment methods. For ex- if konbini was disabled earlier, this job won't enable konbini instead it will just store the extra payment methods received from the servers and the rest will be stored as earlier.

komojuCancelProcess: It will call the KOMOJU API to cancel the orders requested.

- For the cancellation flow, we have to run the job named as komojuCancelProcess.
- Navigate to **Administration>jobs>komojuCancelProcess>Run Now**

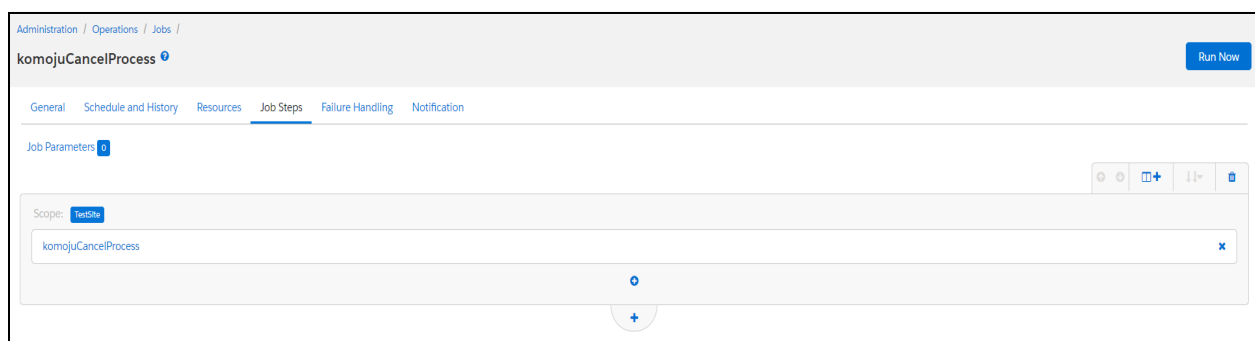
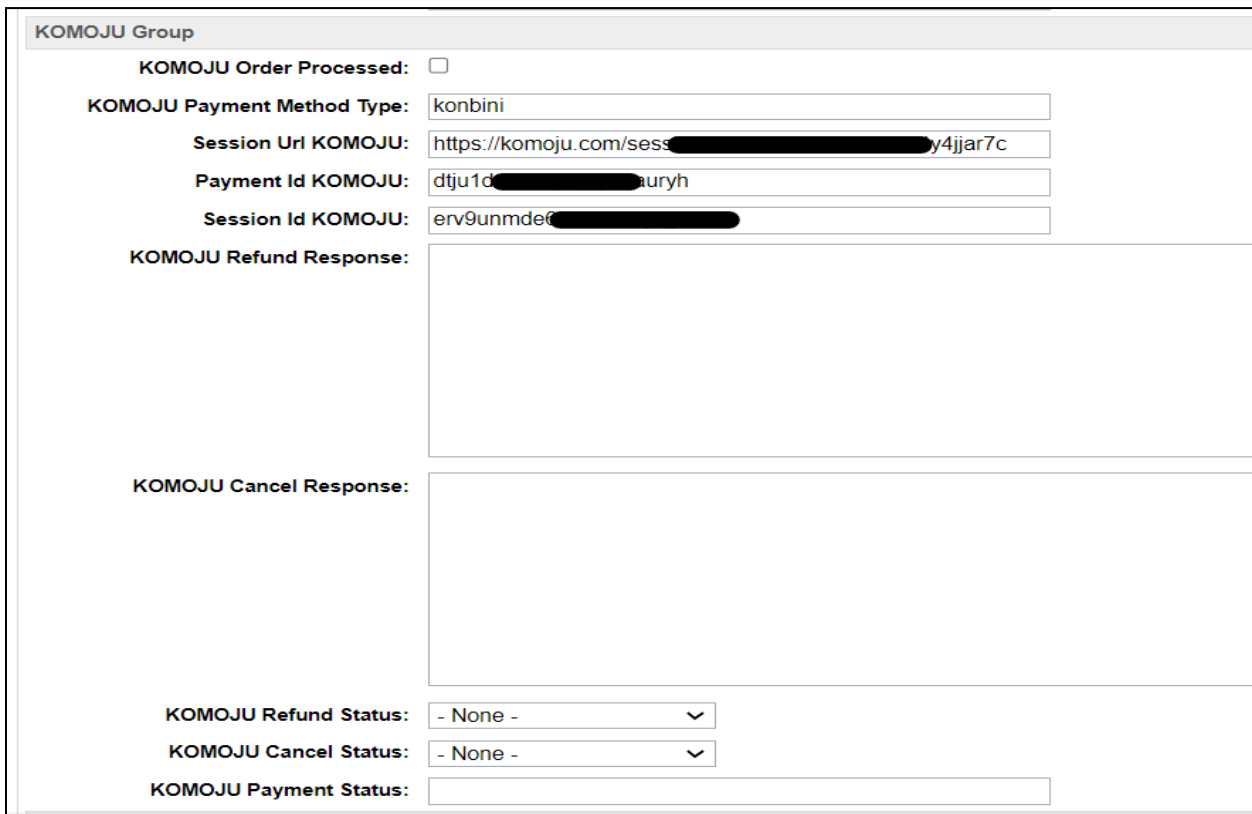


Figure: 12

- This job processes the orders which are in order status Cancelled, payment status as Not Paid and the komojuCancelStatus dropdown should neither be Cancelled nor Rejected.
- If the API response is “error” then the komojuCancelStatus drop-down is updated to **Failed** and the reason is stored in komojuCancelResponse
- If the API response returns a status code as “not_cancellable” then the dropdown is updated to **Rejected**.
- If the API response is ‘OK’ then the komojuCancelStatus dropdown is updated to **Cancelled**.



The screenshot shows a web form titled "KOMOJU Group". It contains several input fields and dropdown menus. The "KOMOJU Order Processed" field is a checkbox. The "KOMOJU Payment Method Type" field is a text input with the value "konbini". The "Session Url KOMOJU" field is a text input with the value "https://komoju.com/sess-...y4jjar7c". The "Payment Id KOMOJU" field is a text input with the value "dtju1d...auryh". The "Session Id KOMOJU" field is a text input with the value "erv9unmde...". The "KOMOJU Refund Response" and "KOMOJU Cancel Response" fields are large text areas. The "KOMOJU Refund Status" and "KOMOJU Cancel Status" fields are dropdown menus, both showing "- None -". The "KOMOJU Payment Status" field is a text input.

| | |
|-----------------------------|-------------------------------------|
| KOMOJU Group | |
| KOMOJU Order Processed: | <input type="checkbox"/> |
| KOMOJU Payment Method Type: | konbini |
| Session Url KOMOJU: | https://komoju.com/sess-...y4jjar7c |
| Payment Id KOMOJU: | dtju1d...auryh |
| Session Id KOMOJU: | erv9unmde... |
| KOMOJU Refund Response: | |
| KOMOJU Cancel Response: | |
| KOMOJU Refund Status: | - None - |
| KOMOJU Cancel Status: | - None - |
| KOMOJU Payment Status: | |

Figure: 13

And if there are some orders which are not able to be Cancelled then we will store their Ids and an email will be sent to the merchant email Id whichever is stored in the attribute **KOMOJU**

Email under BM module KOMOJU Payment Settings, containing the order numbers and the reason for failure.

komojuRefundProcess: It will call the KOMOJU API to refund the orders requested.

- For the Refund flow, we have to run the job named as komojuRefundProcess.
- Navigate to **Administration>jobs>komojuRefundProcess>Run Now**

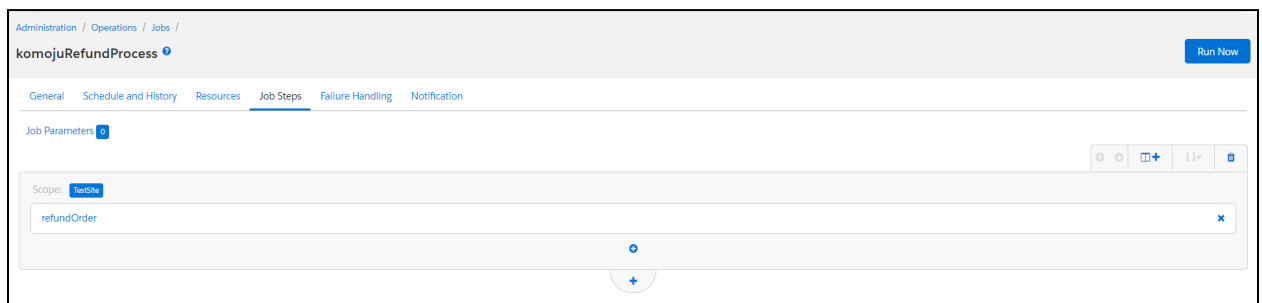


Figure: 14

- This job processes the orders which are in order status Cancelled, payment status as Paid and the komojuRefundStatus dropdown should neither be Refunded nor Rejected.
- If the API response is “error” then the komojuRefundStatus drop-down is updated to **Failed** and the reason is stored in komojuRefundResponse
- If the API response returns a status code as “not_refundable” then the dropdown is updated to **Rejected**.
- If the API response is ‘OK’ then the komojuRefundStatus dropdown is updated to **Refunded**.

And if there are some orders which are not able to be refunded then we will store their Ids and an email will be sent to the merchant email Id whichever is stored in the attribute **KOMOJU Email** under BM module KOMOJU Payment Settings, containing the order numbers and the reason for failure.

4.2 Storefront functionality

4.2.1 Login

- Customers can either log in or shop as guest customers. The UI will look something like this. You can enhance UI by using content slots for the menu and through ISML change if you want.

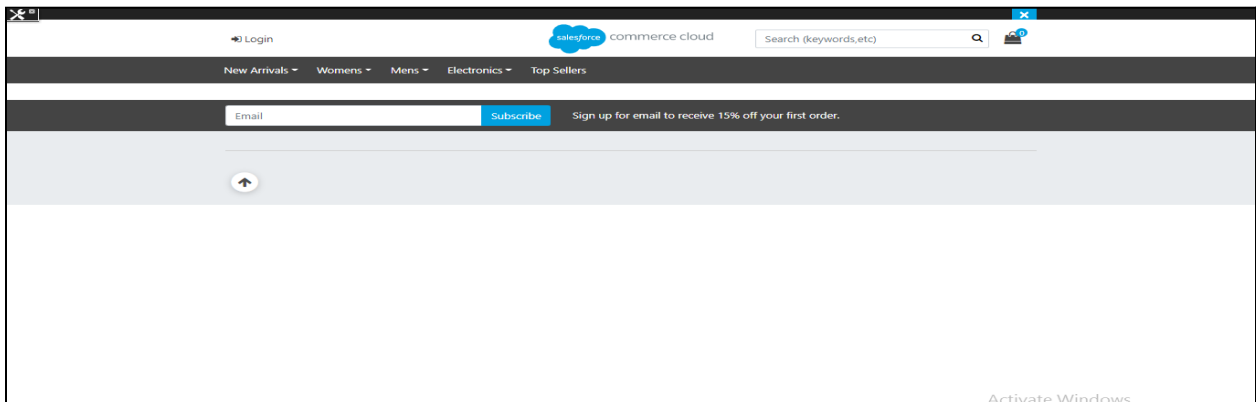


Figure: 15

4.2.2 Add Product to Cart and Proceed to Checkout

- After that customer will select a product and add the product to the cart and click on the cart icon in the top right corner, it will be redirected to the (cart-show) route and you will be able to view the cart as shown below.

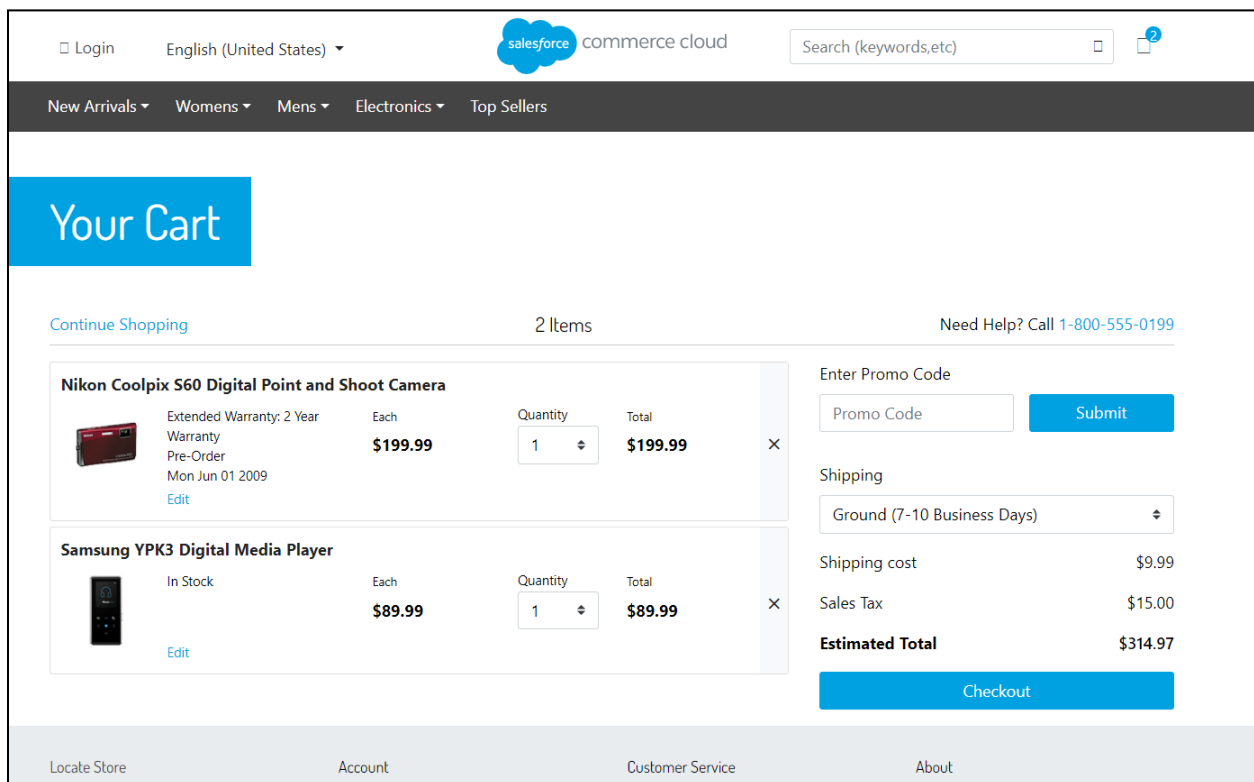
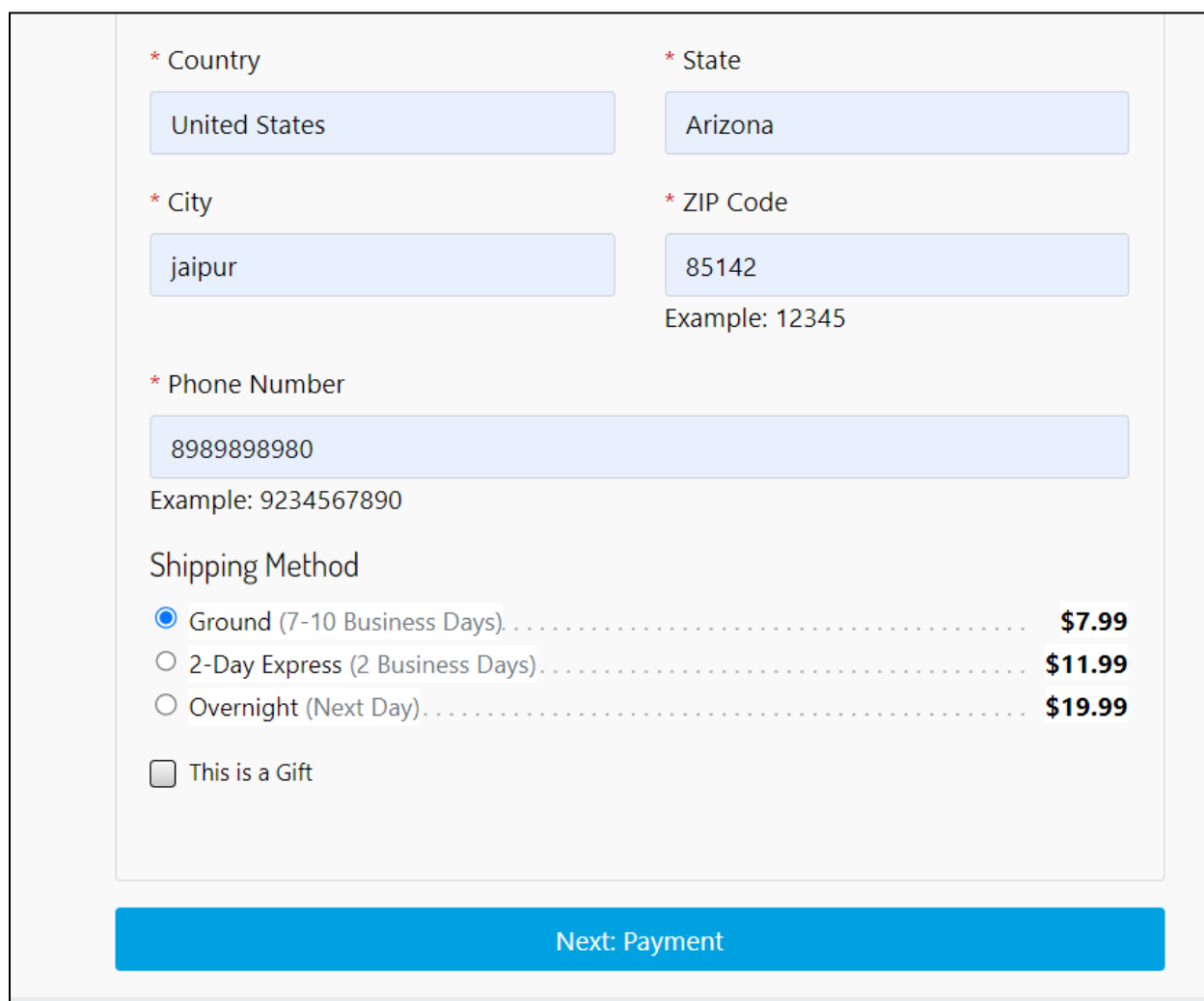


Figure: 16

4.2.3 Checkout Flow

- Now enter Email and continue as a guest or log in to your account, the checkout stage will be updated to the shipping, enter shipping details, and click on the next payment.



A screenshot of a shipping information form. The form is divided into several sections. The first section contains two rows of input fields: 'Country' with 'United States' and 'State' with 'Arizona'. The second section contains 'City' with 'jaipur' and 'ZIP Code' with '85142'. Below the ZIP code is an example: 'Example: 12345'. The third section is 'Phone Number' with '8989898980' and an example: 'Example: 9234567890'. The fourth section is 'Shipping Method' with three radio button options: 'Ground (7-10 Business Days)' selected at \$7.99, '2-Day Express (2 Business Days)' at \$11.99, and 'Overnight (Next Day)' at \$19.99. Below these is a checkbox labeled 'This is a Gift'. At the bottom is a large blue button labeled 'Next: Payment'.

| Field | Value |
|--|---------------|
| * Country | United States |
| * State | Arizona |
| * City | jaipur |
| * ZIP Code | 85142 |
| Example: 12345 | |
| * Phone Number | 8989898980 |
| Example: 9234567890 | |
| Shipping Method | |
| <input checked="" type="radio"/> Ground (7-10 Business Days) | \$7.99 |
| <input type="radio"/> 2-Day Express (2 Business Days) | \$11.99 |
| <input type="radio"/> Overnight (Next Day) | \$19.99 |
| <input type="checkbox"/> This is a Gift | |
| Next: Payment | |


Figure: 17

- Now we get all applicable payment methods from the custom object of the current site and these methods will be visible on the payment page. The Stage will be updated to payment and now you can select the payment method, which the merchant has enabled from the business manager module KOMOJU Payment Settings. The payment page looks like this.

test demo new market Jaipur 北海道 123-1234


住所の更新


新しく追加


*電話番号 


0112341234


支払方式


☒ ペイディ (Paidy) 


☐ PayPay 


☐ クレジットカード 


☐ メルペイ 


☐ 楽天ペイ 


☐ LINE Pay 

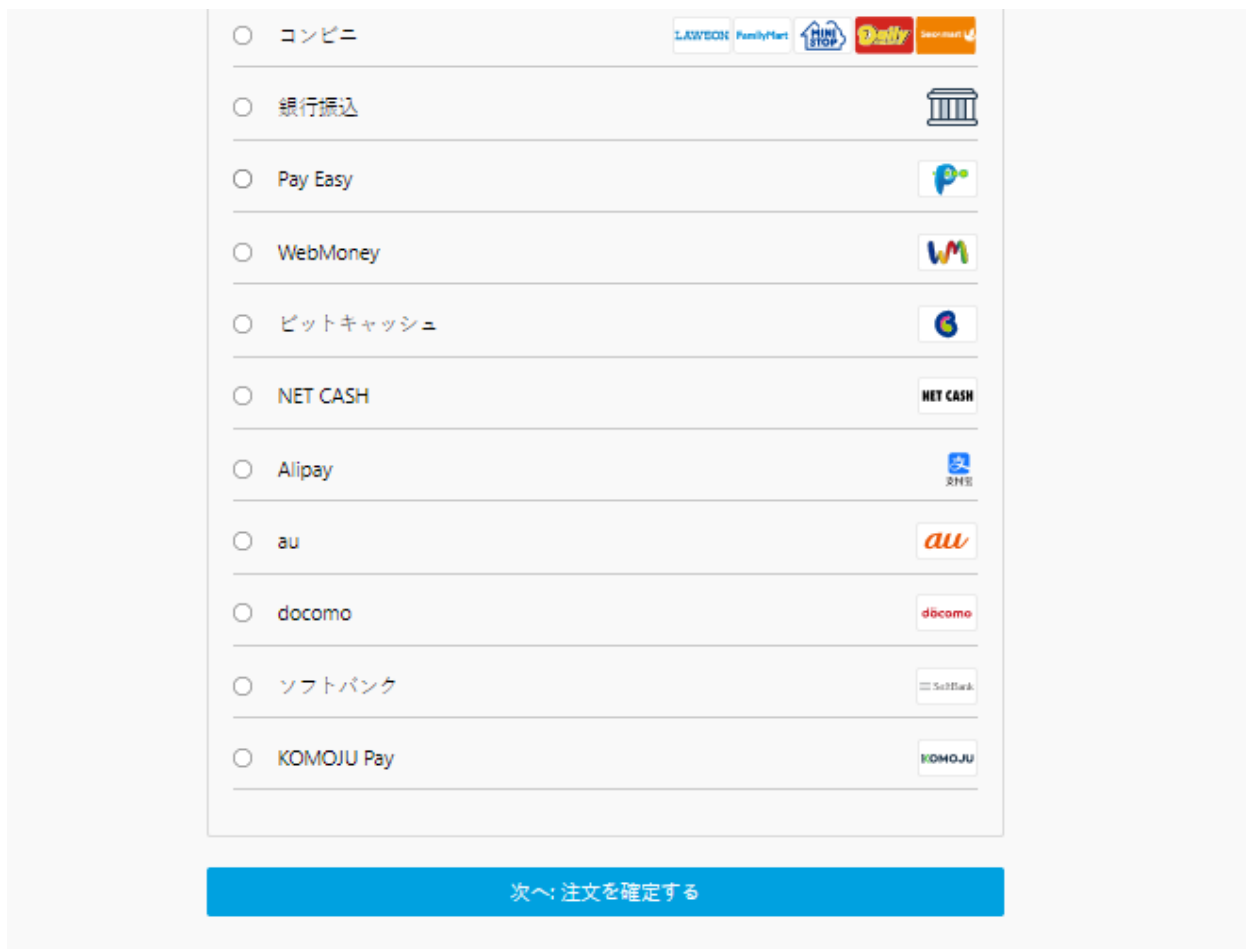
☐ コンビニ 






☐ 銀行振込 

☐ Pay Easy 

☐ WebMoney 

☐ ビットキャッシュ 



| | |
|----------------------------------|---|
| <input type="radio"/> コンビニ | LAWSON FamilyMart MINISTOP Daily SEVEN |
| <input type="radio"/> 銀行振込 |  |
| <input type="radio"/> Pay Easy |  |
| <input type="radio"/> WebMoney |  |
| <input type="radio"/> ビットキャッシュ |  |
| <input type="radio"/> NET CASH | NET CASH |
| <input type="radio"/> Alipay |  |
| <input type="radio"/> au | au |
| <input type="radio"/> docomo | docomo |
| <input type="radio"/> ソフトバンク | SoftBank |
| <input type="radio"/> KOMOJU Pay | KOMOJU |

次へ: 注文を確定する

Figure: 18

- Click on Next: Place Order (CheckoutServices-SubmitPayment) route will be called in the int_komoju_sfra cartridge controller CheckoutServices.
- Whichever payment method is selected by the user, will be sent to this route and it will set this payment method in the Basket custom attribute **KOMOJU Payment Method Type**

UI will be updated and the checkout stage will be updated to place order as below:

(Might need to invalidate cache to get latest changes reflected on your store and checkout page)

支払

編集

Billing Address:
123-1234
秋田県 test city
demo demo
virat kohli
cyntexa@jaipur.com
09012341234

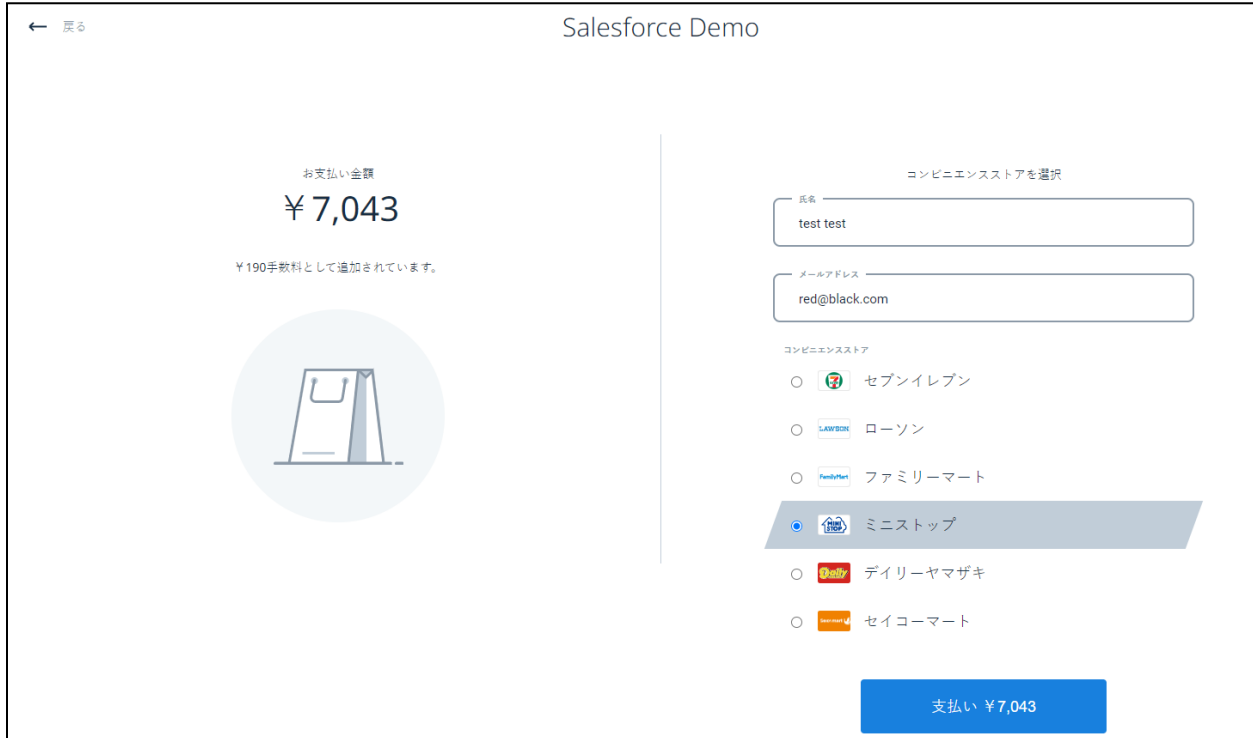
Payment:
コンビニ

注文を確定する

Figure: 19

- Clicking on the place order will call KomojuController-getUrl.
- This controller will create a new session for the hosted page on KOMOJU with the selected payment method by the customer and store it in the current basket custom attribute **KOMOJU Session URL**.
- This route will create an order from the basket and return the return URL to the client-side ajax call in the client-side checkout.js file.
- From the client side, using this **KOMOJU Session URL** customer will be redirected to the hosted page and after paying from the selected payment method flow will be redirected to the KomojuController-KomojuOrder route where the order will be processed, and accordingly, the order status will be updated.

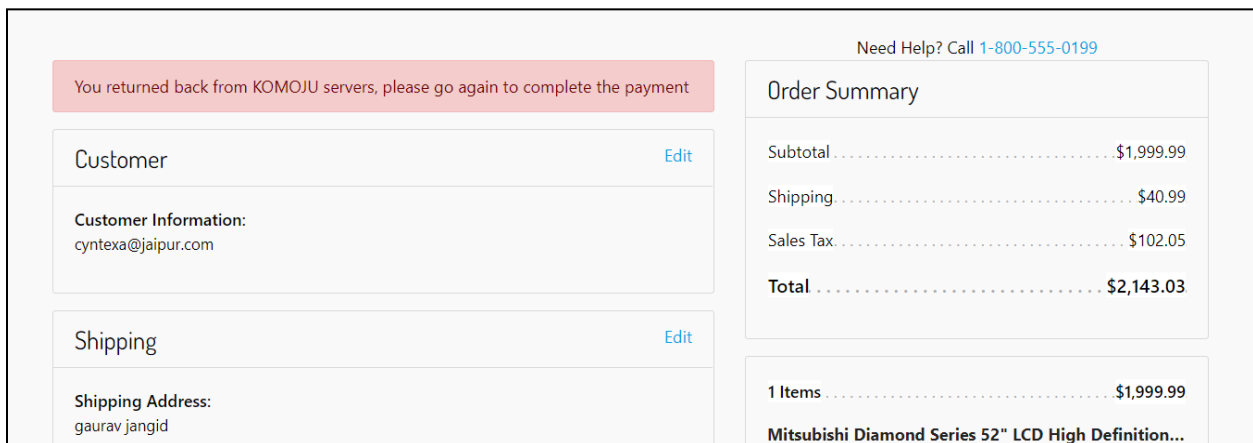
Note: The return URL is sent with the session creation API request to the KOMOJU platform.



The image shows a checkout page titled "Salesforce Demo". On the left, it displays the payment amount "お支払い金額 ¥7,043" and a note "¥190手数料として追加されています。" (Added as a ¥190 handling fee). Below this is a shopping bag icon. On the right, there are input fields for "氏名" (Name) with "test test" and "メールアドレス" (Email Address) with "red@black.com". Below these is a section "コンビニエンスストアを選択" (Select convenience store) with radio button options: セブンイレブン (Seven-Eleven), ローソン (Lawson), ファミリーマート (FamilyMart), ミニストップ (Ministop - selected), デイリーヤマザキ (Daily Yamazaki), and セイコーマート (Seicomart). A blue button at the bottom right says "支払い ¥7,043" (Payment ¥7,043).

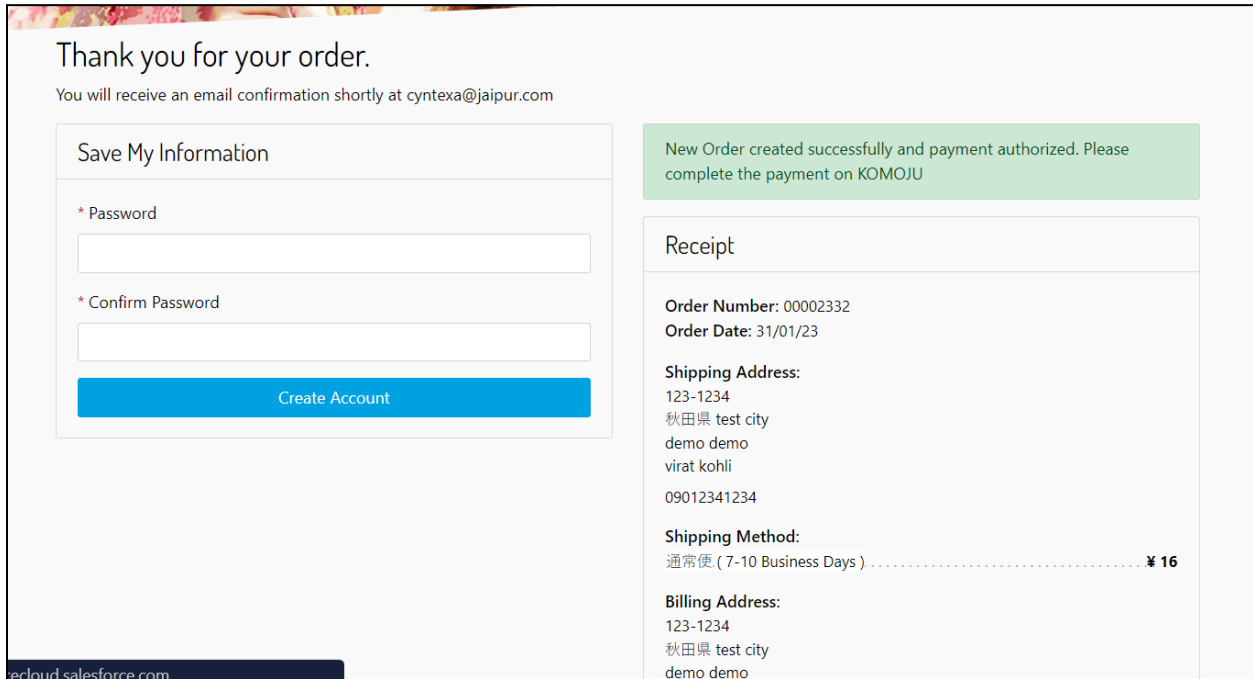
Figure: 20

- If the customer returns back without paying from the hosted page an error is shown on the place-order page.



The image shows an "Order Summary" page. At the top, a red banner message reads: "You returned back from KOMOJU servers, please go again to complete the payment". Below this, on the left, are sections for "Customer" (with "Edit" link) and "Shipping" (with "Edit" link). The "Customer Information" shows "cyntexa@jaipur.com". The "Shipping Address" shows "gaurav jangid". On the right, the "Order Summary" table lists: Subtotal (\$1,999.99), Shipping (\$40.99), Sales Tax (\$102.05), and Total (\$2,143.03). Below this, it shows "1 Items" for "\$1,999.99" and lists "Mitsubishi Diamond Series 52" LCD High Definition...". At the top right, a link says "Need Help? Call 1-800-555-0199".

- Finally, a confirmation page is shown containing the details of the transaction from the KOMOJU. The static messages can be updated through the komojuPayment.properties file.



Thank you for your order.

You will receive an email confirmation shortly at cyntexa@jaipur.com

Save My Information

* Password

* Confirm Password

Create Account

New Order created successfully and payment authorized. Please complete the payment on KOMOJU

Receipt

Order Number: 00002332
Order Date: 31/01/23

Shipping Address:
123-1234
秋田県 test city
demo demo
virat kohli
09012341234

Shipping Method:
通常便 (7-10 Business Days) ¥ 16

Billing Address:
123-1234
秋田県 test city
demo demo

tecloud.salesforce.com...

- The merchant can view all payment-related data in the order details and the payment invoice as shown below.

General

Attributes

Payment

Notes

History

Payment Information for Order '00002529'

| | |
|------------------------------|----------|
| Order Total: | \$646.78 |
| Amount Paid: | \$0.00 |
| Balance Due: | \$646.78 |

Invoice Number:

00009057

[Payment Status:](#)

Not Paid

Payment Method:

KOMOJU_HOSTED_PAGE
Processor: KOMOJU
Transaction: 00002529
Amount: \$646.78

Transaction Status: authorized
KOMOJU Payment ID: dx89wfodz5rm40ivt8izl7n91
KOMOJU Payment Method Type: konbini
KOMOJU Payment Method Sub Type: family-mart
KOMOJU Processing Currency: JPY
KOMOJU Processing Fee: 190 ¥
KOMOJU Exchange Rate: 132.2706
KOMOJU Exchange Amount: 85549.98 ¥

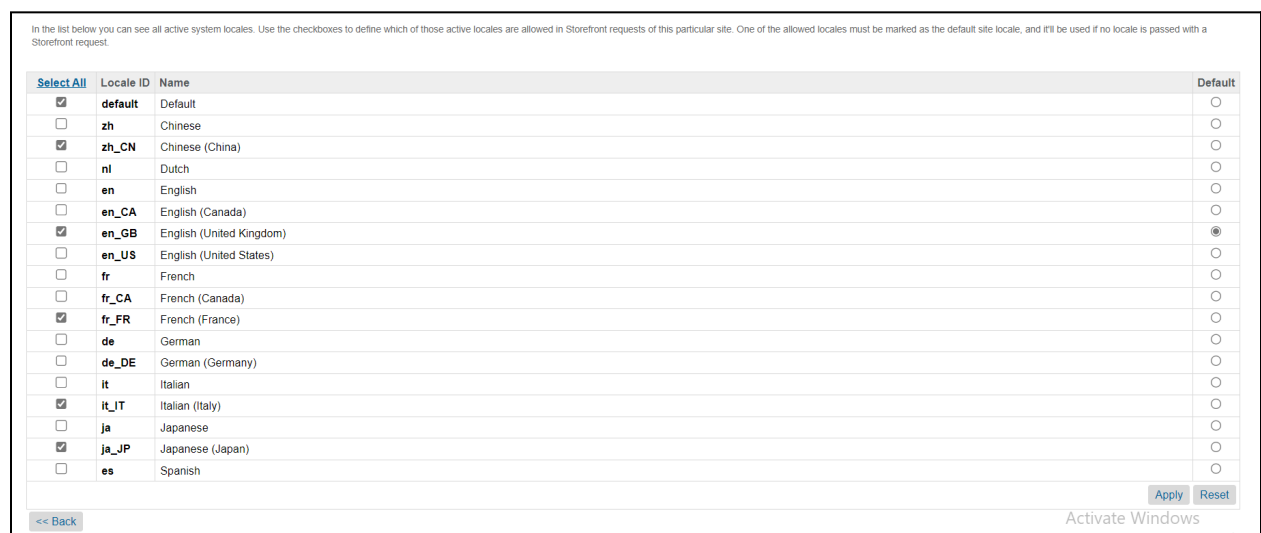
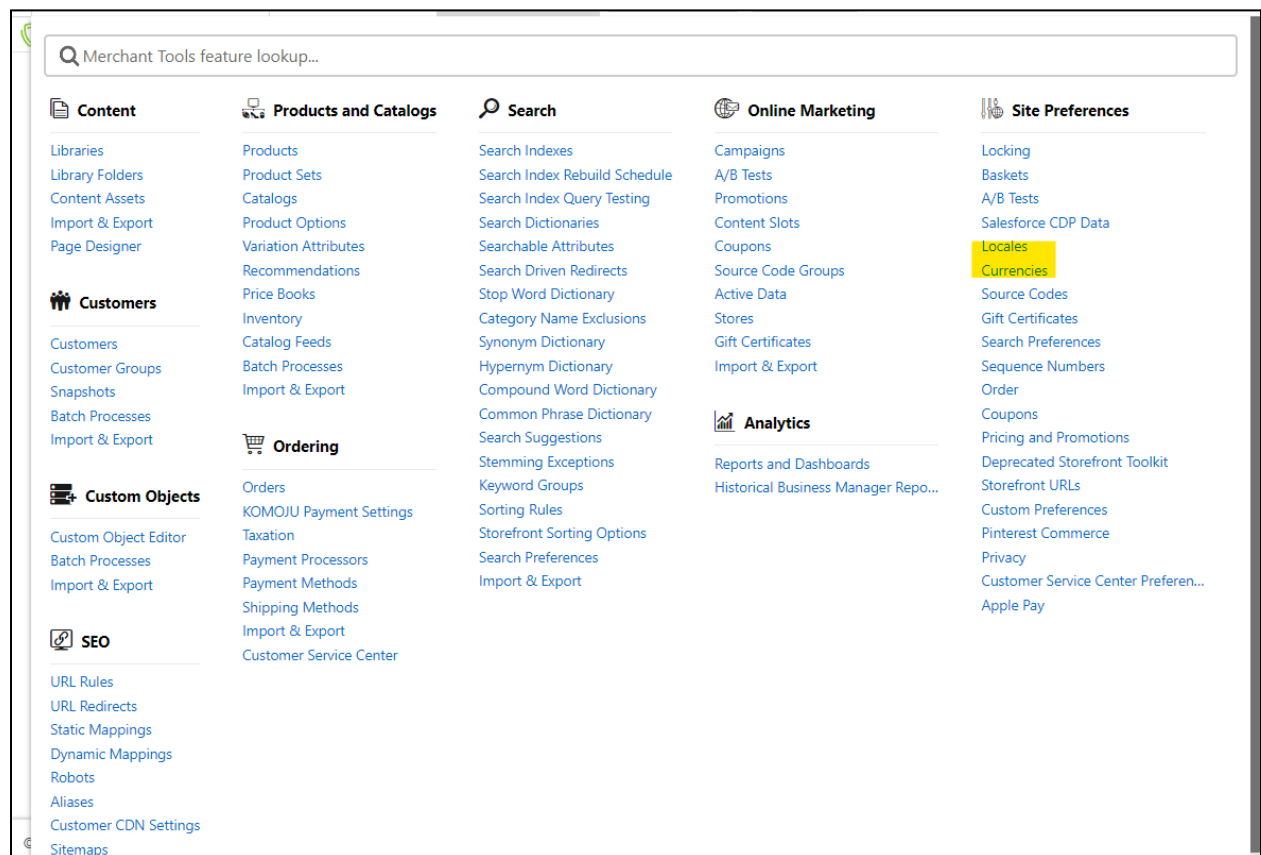
Invoice:**00009057**

| | | |
|-----------------|---|--|
| | SFRA 5 Wall Street Burlington, MA 01803 USA | Order: 00002529 Date: 2/7/23 Total: \$646.78 |
| Bill To: | test test test test city AZ 12345 United States | Phone: 9234567890 Email: cyntexa@jaipur.com |
| Payment Method: | KOMOJU_HOSTED_PAGE Transaction Status: authorized KOMOJU Payment ID: dx89wfodz5rm40ivt8izl7n91 KOMOJU Payment Method Type: konbini KOMOJU Payment Method Sub Type: family-mart KOMOJU Processing Currency: JPY KOMOJU Processing Fee: 190 ¥ KOMOJU Exchange Rate: 132.2706 KOMOJU Exchange Amount: 85549.98 ¥ | |
| Shipment 1 | | |
| | Product Details | Qty. Total Shipping Details |
| | TomTom Go 720 Portable GPS Unit | |
| | Item Number:tomtom-go-720M | 1 \$599.99 test test test |
| | Option Product - Extended Warranty: 2 Year Warranty | 1 \$0.00 test city, AZ 12345 Ground - \$15.99 |

5. Locales and Currency support

- The locale of the Komoju site will be according to the site configuration. The currency will be the same as the locale, which will apply to Revenue and Insights.

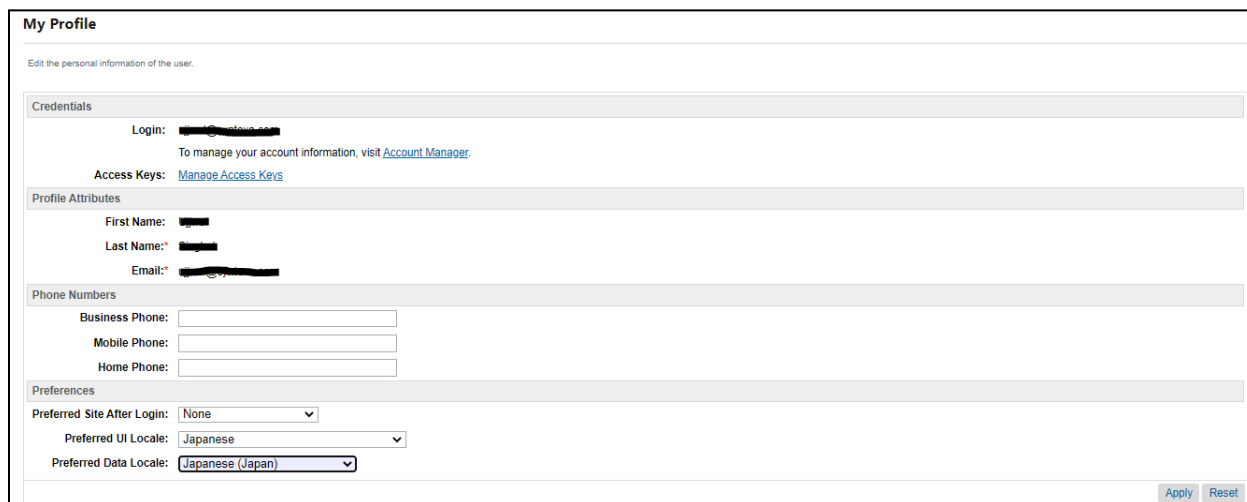
- The Current integration supports 3 Locales: English, Japanese, and South Korean.
- The payment method will be shown according to the configured site locale among **ja**, **en**, **ko**.



● Change the Locale of BM

The current integration supports two locales English and Japanese.

To change the locale of the current login user into Japanese go to user profile from Profile Icon > In Preferred UI Locale choose locale Japanese > In Preferred Data Locale choose locale Japanese (Japan) > Apply



My Profile

Edit the personal information of the user.

Credentials

Login: [redacted]
To manage your account information, visit [Account Manager](#).

Access Keys: [Manage Access Keys](#)

Profile Attributes

First Name: [redacted]
Last Name: [redacted]
Email: [redacted]

Phone Numbers

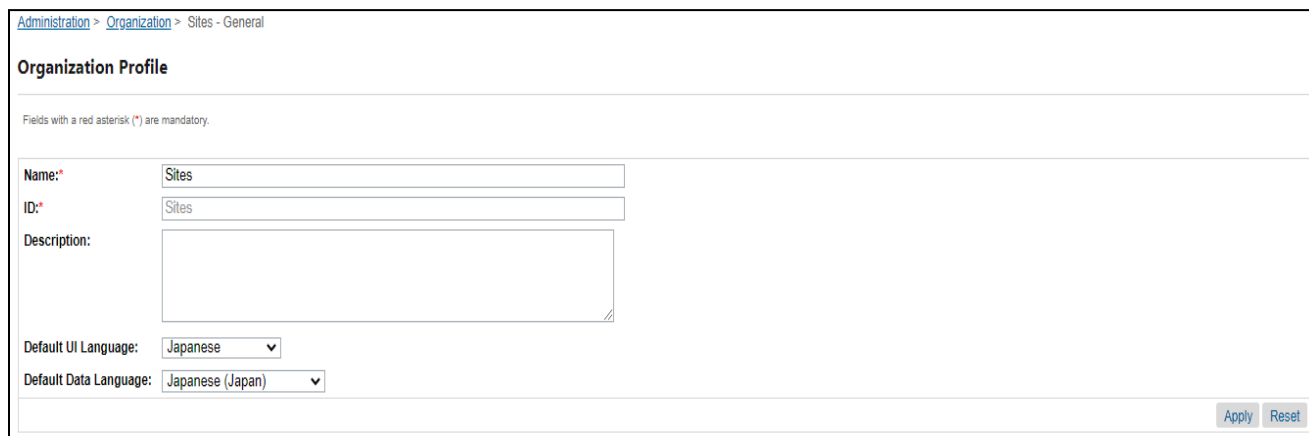
Business Phone:
Mobile Phone:
Home Phone:

Preferences

Preferred Site After Login:
Preferred UI Locale:
Preferred Data Locale:

[Apply](#) [Reset](#)

To change locale of the organization into Japanese go to Administration > Organization > Organization Profile > In Default UI Language choose Japanese > In Default Data Language choose locale Japanese to choose Japanese (Japan) > Apply



[Administration](#) > [Organization](#) > Sites - General

Organization Profile

Fields with a red asterisk (*) are mandatory.

Name:
ID:
Description:

Default UI Language:
Default Data Language:

[Apply](#) [Reset](#)

6. Version and SFRA support info

- The JavaScript Files are located at:
Cartridges\int_komoju_sfra\cartridge\client\default\js
- The file with the current version of the int_komoju_sfra cartridge tool cartridge is located at:
int_komoju_sfra\cartridge\int_komoju_sfra.properties
- **SFRA Version Support**

Komoju cartridge version ----- supports SFRA 6.0.0 and above.

- **SiteGenesis Version Support**

Komoju cartridge version ----- supports SiteGenesis 105.2.1 and above.

- **Compatibility Mode**

Cartridge tested with Compatibility Mode 22.4.

7. Service Errors Handling

7.1 Service Timeout handling

To Configure Logs go to Administration > Operations > Custom Log Settings.

In Log Category put customKomojuErrors, Choose Log level error and click on add. Now Add another category customKomojuSourceLogger, choose log level Info, and click on add.

Custom Log Filters

Custom log category names can't include 'custom' as a prefix.

| Active | Log Category | Log Level | |
|-------------------------------------|--|-----------|---------------------------------------|
| | <input type="text" value="Enter a log category..."/> | INFO | <input type="button" value="Add"/> |
| | root | WARN | |
| <input checked="" type="checkbox"/> | customKomojuErrors | ERROR | <input type="button" value="Remove"/> |
| <input checked="" type="checkbox"/> | customKomojuSourceLogger | INFO | <input type="button" value="Remove"/> |

Custom Log Targets

Email: Messages with log level FATAL can be sent to email recipients.

Log Files: Select Log Levels Written to Files:

- ☒ FATAL
- ☒ ERROR
- ☒ WARN
- ☒ INFO
- ☒ DEBUG

Log Center: Messages that pass the filters are sent to the log search application.

If Komoju's service becomes unavailable, error details can be found in the following logs:

- Error details can be found in the log with the **custom.customKomojuErrors** prefix.
- Service requests can be found in the log with the **custom.customKomojuSourceLogger** prefix.

For more details about logs, see the Logs section.

7.2 Handle Error response

Error handling in the integration based on SFRA build in **custom.customKomojuErrors**.


- Detailed errors from Komoju can be found in custom-search errors.
- All error text is located at:

Administration > Site Development > Development Setup > Log files folder icon. Now Search for **customKomojuErrors** errors.

8. Known Issues

No Reported Issues so far. Please Report any issue at deshna@cyntexa.com

8.1 Test Cases

 Test Case Sheet Komoju

9. Failover/ Recovery process

In case of any technical failures or issues in processing communication with the servers please contact us at deshna@cyntexa.com