DigitalRiver SFRA LINK Cartridge Developer Guide

Version 22.3.0
SFRA 6.0.0

Table of Contents

The *DigitalRiver* LINK cartridge for Salesforce B2C provides a global payments and risk mitigation solution. Digital River enables localized payment methods, tax calculation and secure localized payment processing to minimize declines and foreign transaction fees. Beyond payments, Digital River works behind the scenes to fight fraud, minimize chargebacks, optimize billing to increase authorization rates, and manage global taxes and regulatory compliance.
You can find Digital River documentation on the Digital River website:

The *DigitalRiver* LINK cartridge is essential for brands looking to enter new global markets with ease while delivering the right local payment methods, currencies, and tax calculations. This is the only extension brands need for payments, fraud, tax, financial reconciliation, and compliance.

Features:

- Payment gateway (cards and alternative payment methods)
- PCI compliance
- Tokenization
- Acquirer redundancy
- Billing optimization
- Multi-currency support and conversion
- Tax calculation
- Tax management (registration, collection, filing, and remittance)
- Fraud screening
- Advanced fraud management
- Fraud liability guarantee
- Chargeback mitigation and dispute management
- Global regulatory compliance (e.g., GDPR and PSD2)

1. Component Overview

Functional Overview

The DigitalRiver package contains three cartridges with the DropIn functionality for SFRA architecture:

- *bm_digitalriver*
- *int_digitalriver*
- *int_digitalriver_sfra*
- *int_digitalriver_webhooks*
- *Int_digitalriver_customercredit*

Implementation follows the DigitalRiver DropIn on each step.

bm_digitalriver
A Business Manager extension cartridge which will:

communicate with Digital River services and report if they are up or down. These include static requests that use the REST API to send a payload to the Digital River services. The response is then checked to see if the service returns an expected successful response.

give permission to execute job *sheduledDeltaSKUs*. The button for interacting with the job will have the corresponding text indicating that the job can be launched or that the job is being executed.

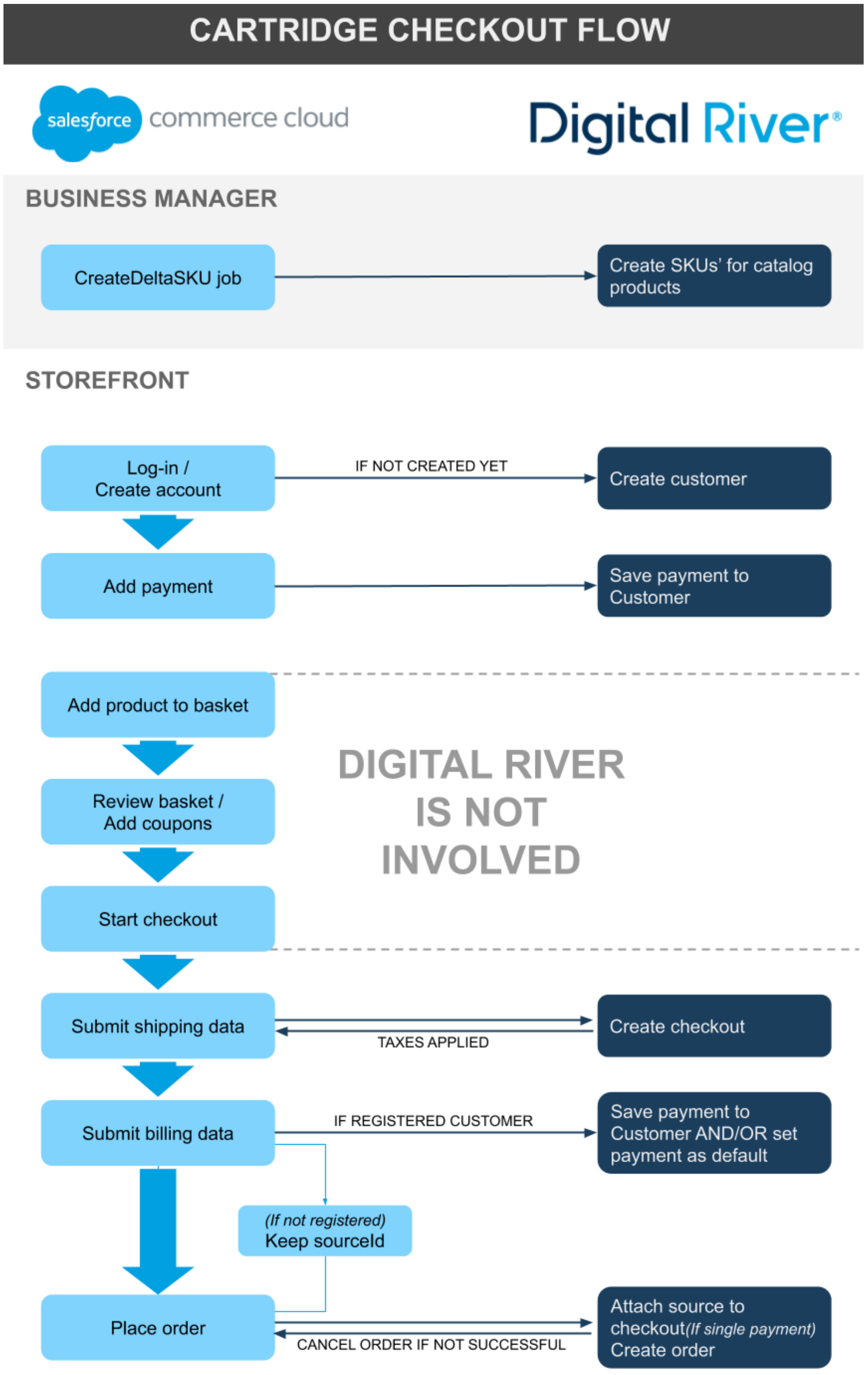
See section *Business Manager Roles & Permissions* on adding the context menu item to a role.

int_digitalriver
Integration cartridge that contains API integrations that form requests, parse responses, and updates objects.

int_digitalriver_sfra
This cartridge delivers adjustments and extensions to existing storefront functionality.

int_digitalriver_webhooks
Integration cartridge containing API integrations that implement DigitalRiver webhooks processing.

int_digitalriver_customercredit
Integration cartridge containing API integrations that implement DigitalRiver customer credit functionality.



NOTE1: This diagram shows general checkout flow, one arrow may mean multiple calls
NOTE2: After shipping submit basket state is tracked on B2C Commerce side. Each time basket is re-calculated, basket state compared with previous one. New create checkout call is made each time basket line items or their base price changes

Use Cases

Drop-In uses the client-side *DigitalRiver.js* library and data collected during the checkout session to present customers a payment method selection based on the configuration and allow them to complete the payment from within the widget.

Limitations, Constraints

Only SFRA architecture is supported.
Use of the *DigitalRiver* cartridge requires credentials and keys from Digital River. Please contact Digital River customer service. The cartridge is designed for the US locale, but can work with other locales.

The following SFCC features are not supported:

- Multiple Shipping Addresses
- Tax Exempt is currently under development

***Please note:** When a customer stores a credit card in the Account, card data is actually stored on the DigitalRiver service, hence if you turn off DigitalRiver, the customer will not be able to pay with stored cards and will have to add them to Account again.*

Compatibility

This package was implemented against SF B2C Version 22.5. Tested against Compatibility mode 21.7. The cartridge only supports SFRA. This cartridge is not supported with SiteGenesis. The cartridge was developed against SFRA version 6.0.0.

Privacy, Payment

This cartridge does access customer profile data to send the SFCC Customer ID and email to DigitalRiver.

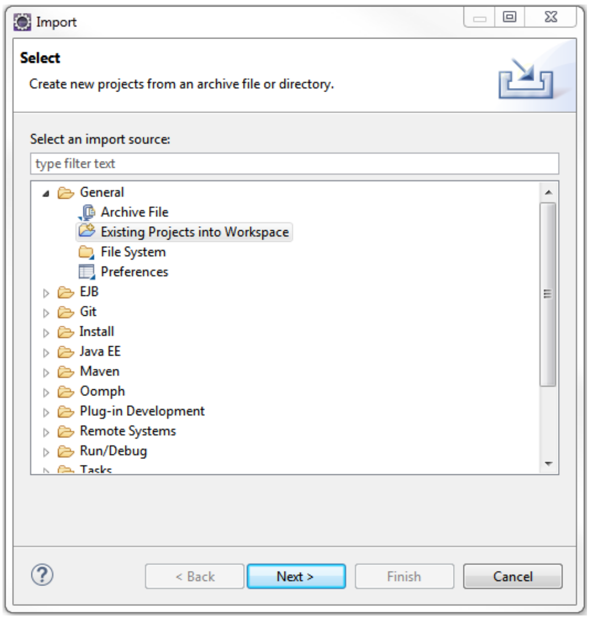
Digital River provides PCI compliance, tokenization, and fraud screening through an API reference. When the shopper enters payment information at checkout, they interact with secure payment fields that are hosted by Digital River within a Salesforce-hosted page. The shopper is never redirected to another page. Those secure fields are served up by *DigitalRiver.js*, a developer-friendly library that allows brands to securely process transactions without having to take on the responsibilities of PCI compliance. When brands use the *DigitalRiver.js* API reference on their Salesforce store, Digital River serves up payment methods, validates card data, tokenizes customer information, performs fraud screening, collects sensitive card data, and processes transactions in a manner that is fully-compliant with the Payment Card Industry Data Security Standards (PCI DSS). Digital River is a Level 1 PCI compliant service provider. Customer payment data is stored on Digital River’s secure servers, never on Salesforce servers. The service does not allow PCI data to hit the Salesforce server.
For additional privacy information, please contact your DigitalRiver Account Manager.

2. Implementation Guide

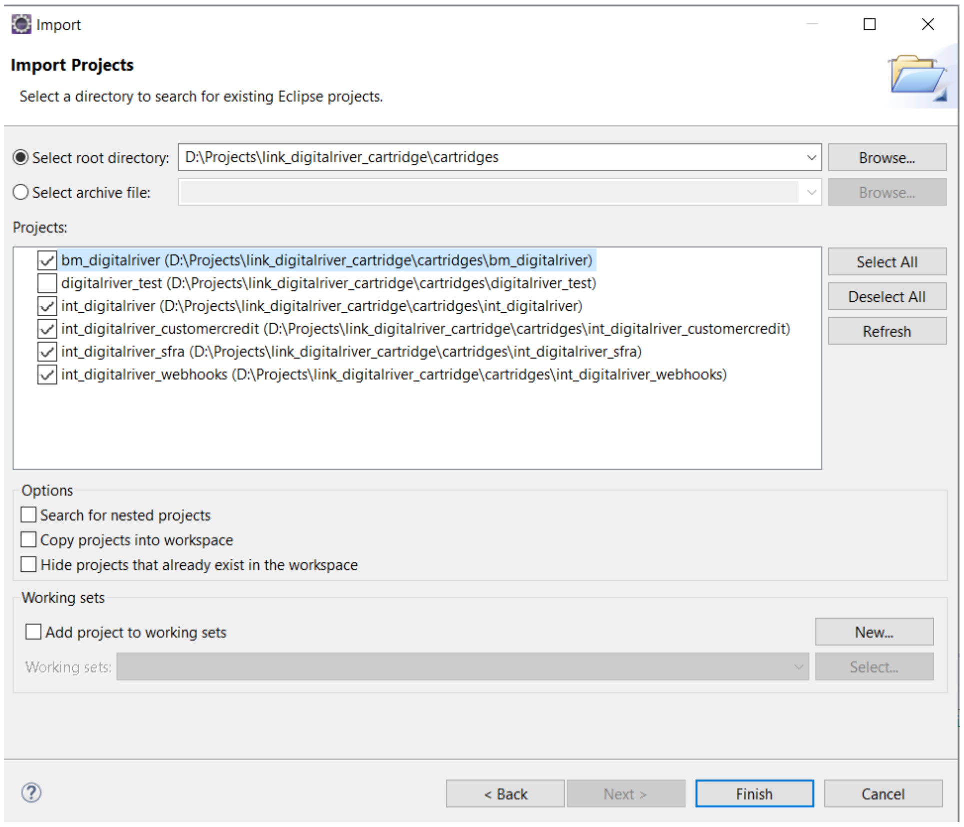
Setup of Business Manager

Importing cartridges

1. Download the cartridge source code.
2. Establish a new digital server connection with your SFCC Instance
3. Import cartridges to a workspace in Salesforce UX Studio.



4. Add cartridges to Project Reference of Server Connection.


Import

Import Projects

Select a directory to search for existing Eclipse projects.

(●) Select root directory: D:\Projects\link_digitalriver_cartridge\cartridges

( ) Select archive file:

Projects:

- [✓] bm_digitalriver (D:\Projects\link_digitalriver_cartridge\cartridges\bm_digitalriver)
- [ ] digitalriver_test (D:\Projects\link_digitalriver_cartridge\cartridges\digitalriver_test)
- [✓] int_digitalriver (D:\Projects\link_digitalriver_cartridge\cartridges\int_digitalriver)
- [✓] int_digitalriver_customercredit (D:\Projects\link_digitalriver_cartridge\cartridges\int_digitalriver_customercredit)
- [✓] int_digitalriver_sfra (D:\Projects\link_digitalriver_cartridge\cartridges\int_digitalriver_sfra)
- [✓] int_digitalriver_webhooks (D:\Projects\link_digitalriver_cartridge\cartridges\int_digitalriver_webhooks)

Options
- [ ] Search for nested projects
- [ ] Copy projects into workspace
- [ ] Hide projects that already exist in the workspace

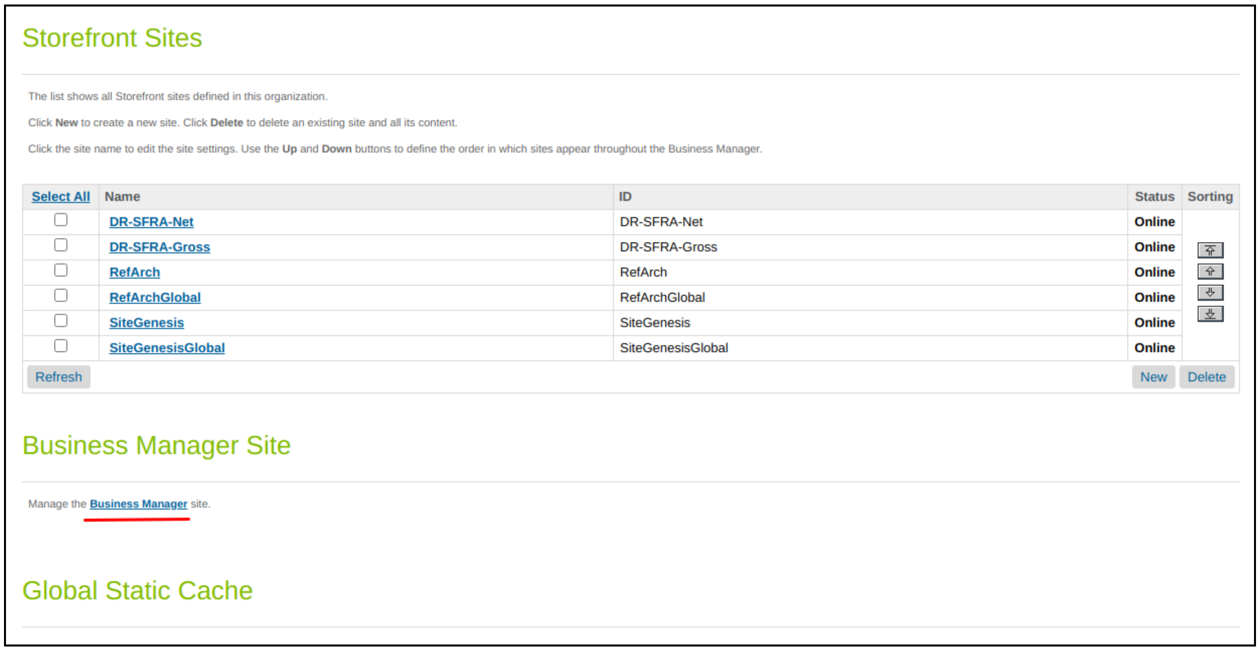
Working sets
- [ ] Add project to working sets

Working sets:

5. Wait until Studio completes workspace build and uploads source codes to the sandbox.

Assigning cartridges to the site

1. Go to *Administration > Sites > Manage Sites* . Follow link *Business Manager*.


Storefront Sites

The list shows all Storefront sites defined in this organization.

Click **New** to create a new site. Click **Delete** to delete an existing site and all its content.

Click the site name to edit the site settings. Use the **Up** and **Down** buttons to define the order in which sites appear throughout the Business Manager.

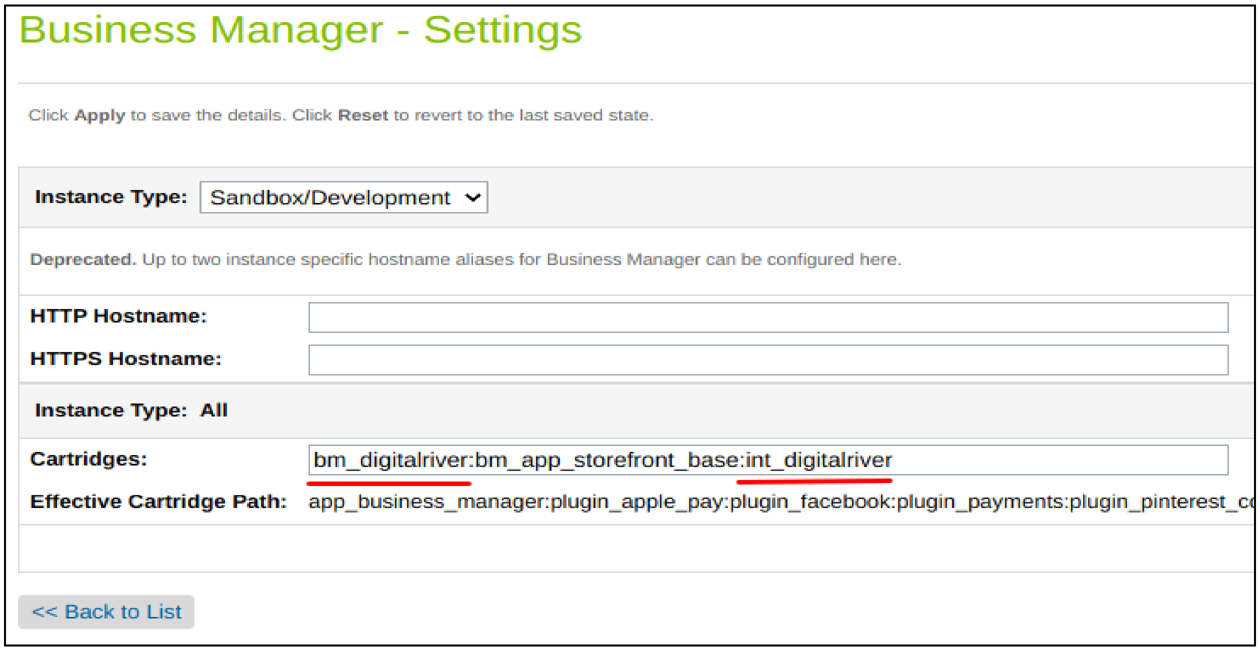
Select All	Name	ID	Status	Sorting
☐	DR-SFRA-Net	DR-SFRA-Net	Online	
☐	DR-SFRA-Gross	DR-SFRA-Gross	Online	
☐	RefArch	RefArch	Online	
☐	RefArchGlobal	RefArchGlobal	Online	
☐	SiteGenesis	SiteGenesis	Online	
☐	SiteGenesisGlobal	SiteGenesisGlobal	Online	

Business Manager Site

Manage the Business Manager site.

Global Static Cache

2. In the beginning of the cartridge path add the following:
   bm_digitalriver: and *:int_digitalriver*
   Press the Apply button.



Business Manager - Settings

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

Instance Type: Sandbox/Development

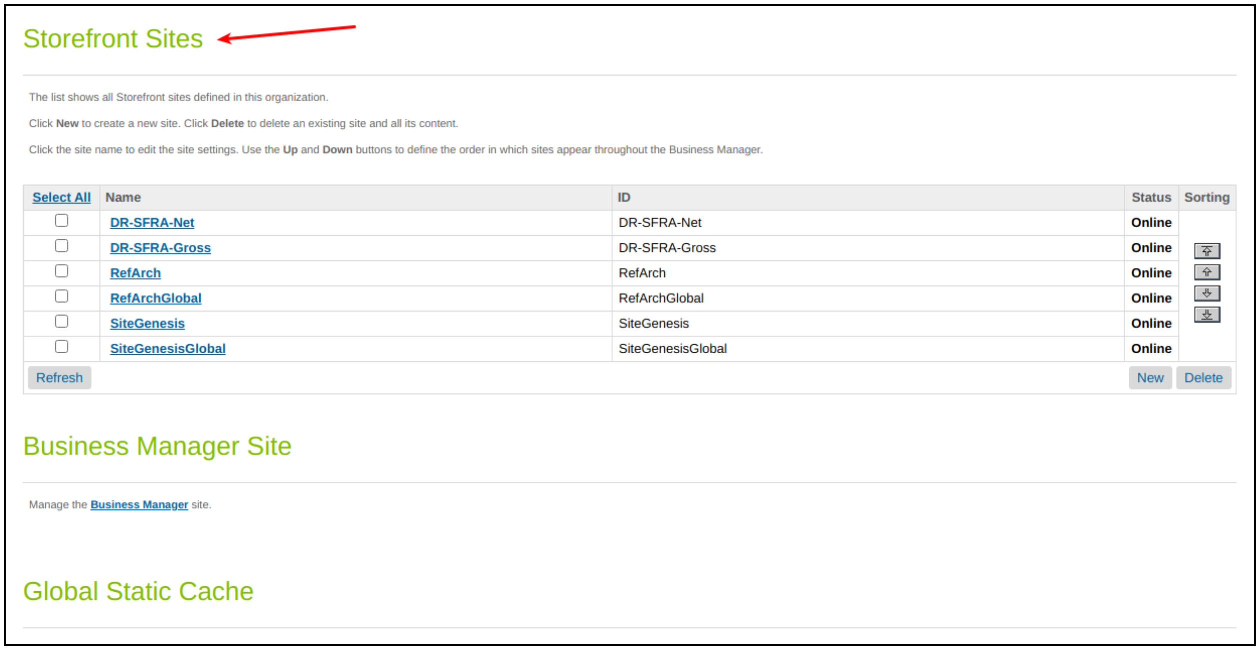
Deprecated. Up to two instance specific hostname aliases for Business Manager can be configured here.

HTTP Hostname:	
HTTPS Hostname:	

Instance Type: All

Cartridges:	bm_digitalriver:bm_app_storefront_base:int_digitalriver
Effective Cartridge Path:	app_business_manager:plugin_apple_pay:plugin_facebook:plugin_payments:plugin_pinterest_c

3. Go to *Administration* > *Sites* > *Manage Sites* . Select your site from the list *Storefront Sites*.



Storefront Sites

The list shows all Storefront sites defined in this organization.

Click **New** to create a new site. Click **Delete** to delete an existing site and all its content.

Click the site name to edit the site settings. Use the **Up** and **Down** buttons to define the order in which sites appear throughout the Business Manager.

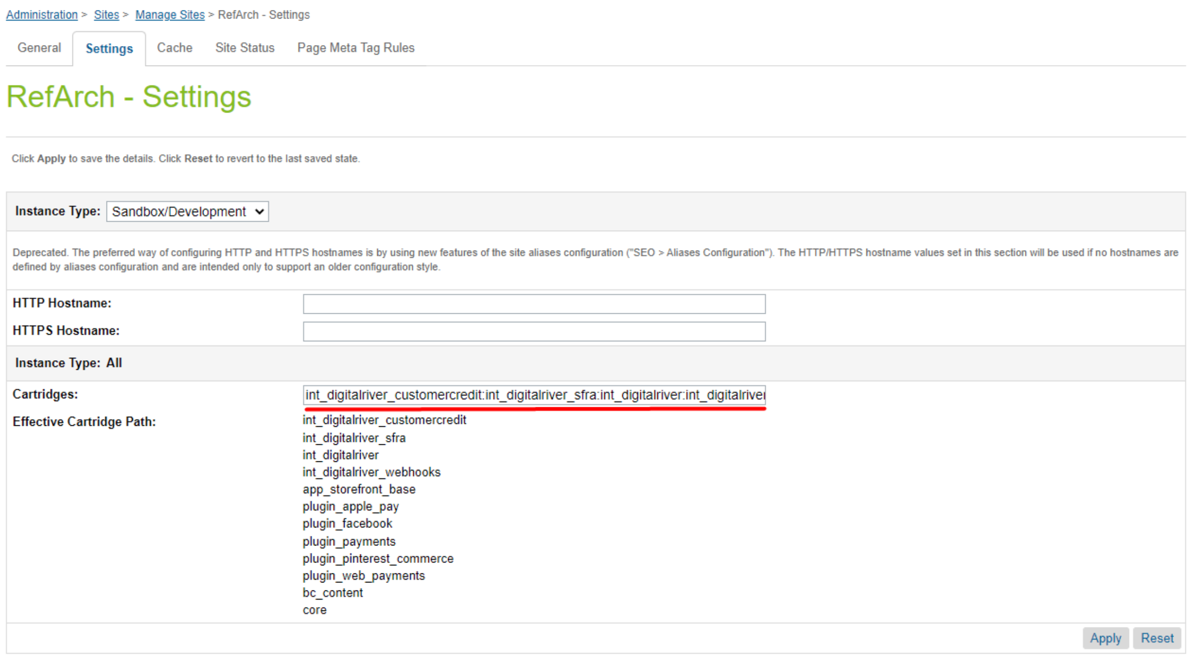
Select All	Name	ID	Status	Sorting
☐	DR-SFRA-Net	DR-SFRA-Net	Online	
☐	DR-SFRA-Gross	DR-SFRA-Gross	Online	
☐	RefArch	RefArch	Online	
☐	RefArchGlobal	RefArchGlobal	Online	
☐	SiteGenesis	SiteGenesis	Online	
☐	SiteGenesisGlobal	SiteGenesisGlobal	Online	

Business Manager Site

Manage the Business Manager site.

Global Static Cache

4. Select *Settings* tab. In the beginning of the cartridge path add the following:
   `int_digitalriver_customercredit:int_digitalriver_sfra:int_digitalriver:int_digitalriver_webhooks:`
   Press the *Apply* button.


Administration > Sites > Manage Sites > RefArch - Settings

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: Sandbox/Development

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 hostname values set in this section will be used if no hostnames are defined by aliases configuration and are intended only to support an older configuration style.

HTTP Hostname:

HTTPS Hostname:

Instance Type: All

Cartridges: int_digitalriver_customercredit:int_digitalriver_sfra:int_digitalriver:int_digitalrive

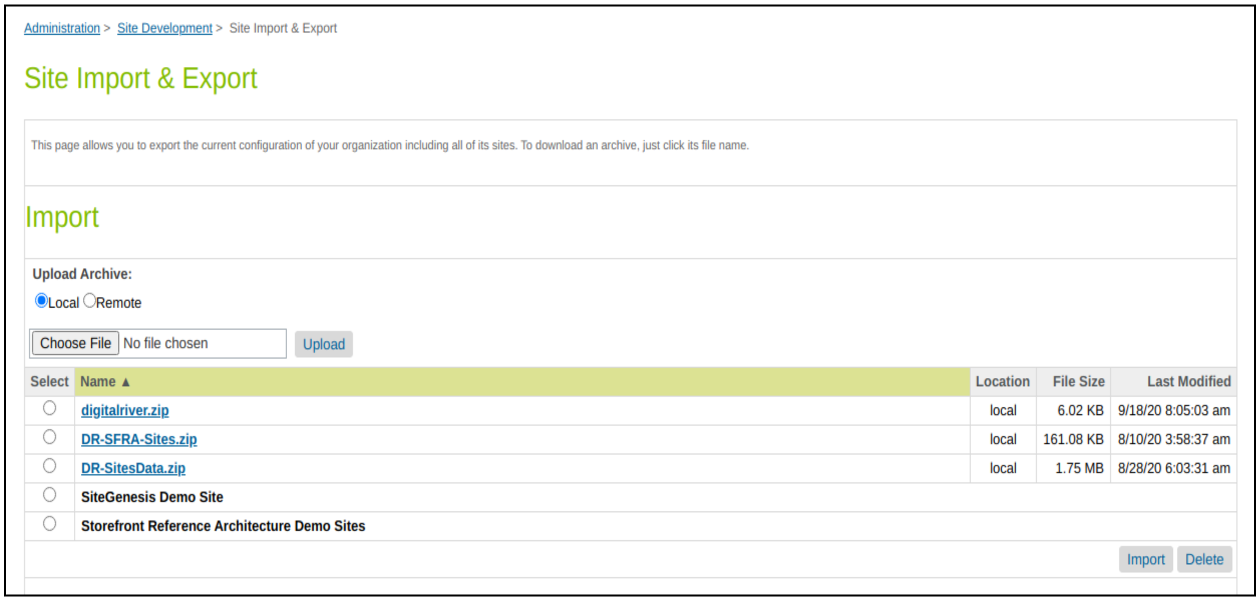
Effective Cartridge Path:
int_digitalriver_customercredit
int_digitalriver_sfra
int_digitalriver
int_digitalriver_webhooks
app_storefront_base
plugin_apple_pay
plugin_facebook
plugin_payments
plugin_pinterest_commerce
plugin_web_payments
bc_content
core

Apply | Reset

Import of metadata

In order to get these onto your instance, navigate into the *metadata* folder in cartridge distribution. Inside, find the folder *zip_this_up_after_site_change* and enter that folder. In there, go into the *sites* folder. In the *sites* folder, there will be a folder called *ChangeThisToSiteID* – Modify this folder name to be the Site ID of your current site. Once that is done, create a ZIP file of the main folder (*zip_this_up_after_site_change* with a name like *digitalriver* as an example).

Go to *Administration > Site Development > Site Import & Export*. Upload *digitalriver.zip* from the *metadata* folder.



Administration > Site Development > Site Import & Export

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

Upload Archive:

[✓ Local] Remote

Select	Name ▲	Location	File Size	Last Modified
○	digitalriver.zip	local	6.02 KB	9/18/20 8:05:03 am
○	DR-SFRA-Sites.zip	local	161.08 KB	8/10/20 3:58:37 am
○	DR-SitesData.zip	local	1.75 MB	8/28/20 6:03:31 am
○	SiteGenesis Demo Site			
○	Storefront Reference Architecture Demo Sites			

Select *digitalriver.zip* and finish the import process.

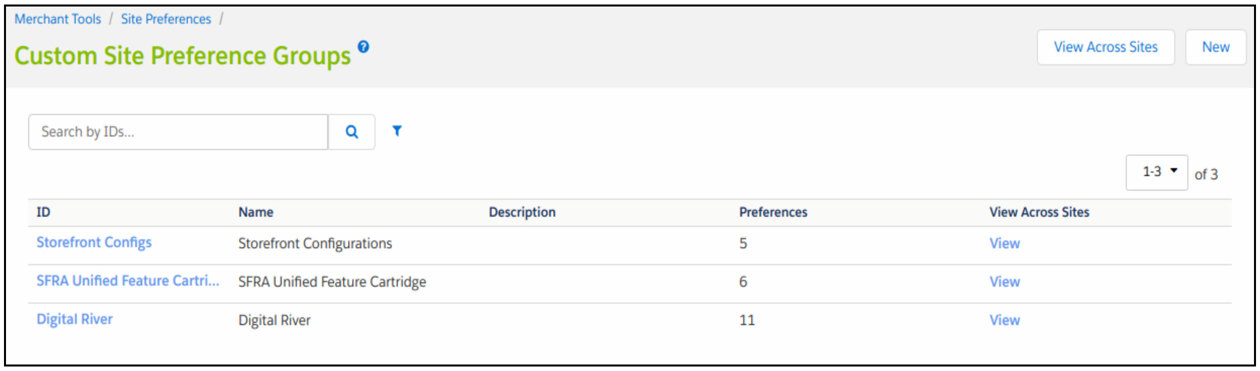
***CreateSKUs, createDeltaSKUs, scheduledDeltaSKUs* jobs**

The zip will create the initial configuration for the jobs used to send SKU data to Digital River. Once the file is imported, be sure to go to *Administration > Operations > Jobs* the *Job Steps* configuration tab and change the site *Scope* to your site ID.

Configuration

Custom Site Preferences

After import is successfully done, you should see *Custom Site Preferences Groups* in BM *Merchant Tools > Site Preferences > Custom Preferences*.

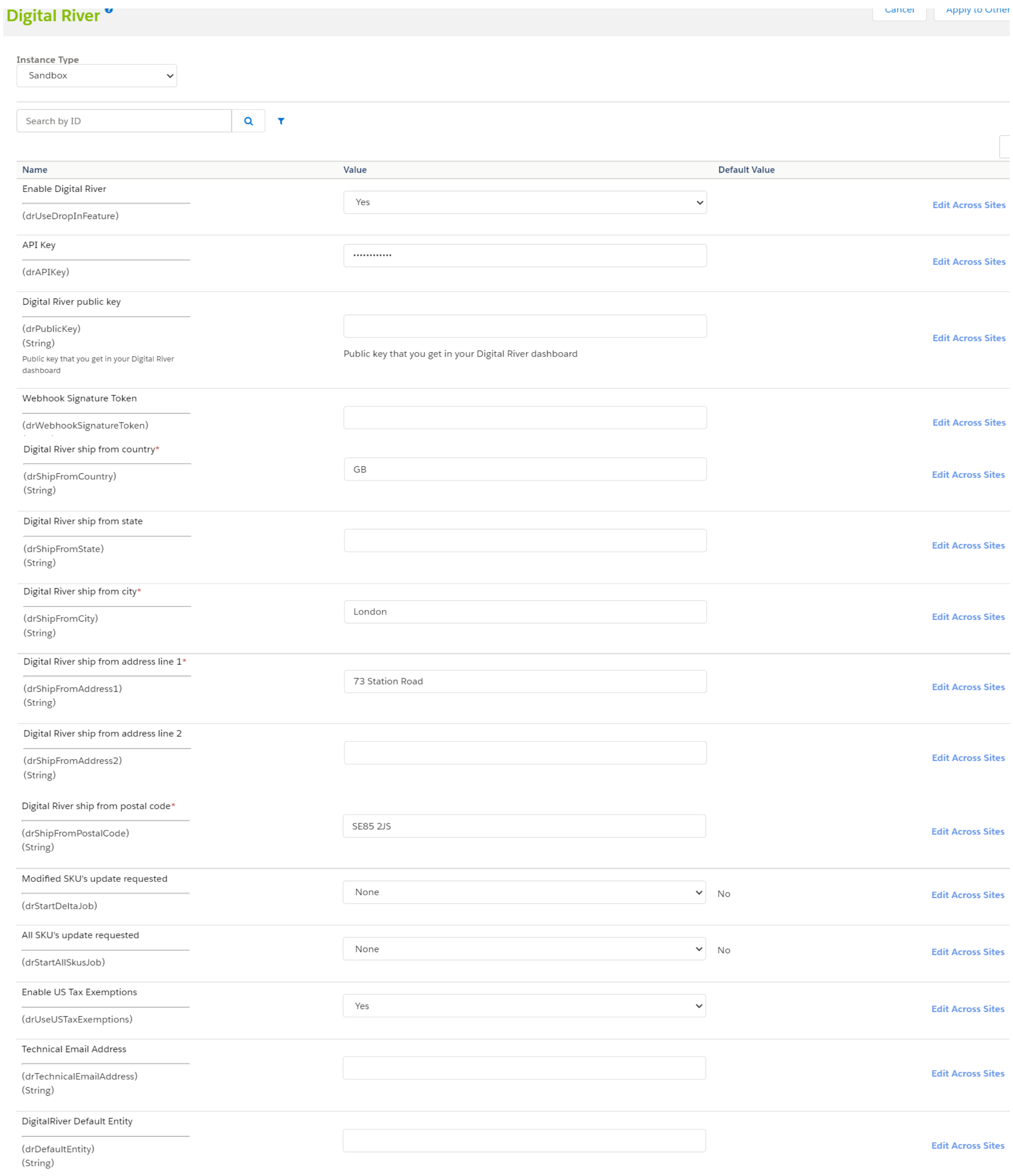


Merchant Tools / Site Preferences /

Custom Site Preference Groups

ID	Name	Description	Preferences	View Across Sites
Storefront Configs	Storefront Configurations		5	View
SFRA Unified Feature Cartri...	SFRA Unified Feature Cartridge		6	View
Digital River	Digital River		11	View

Select *Digital River* and you should see fields for cartridge setup.

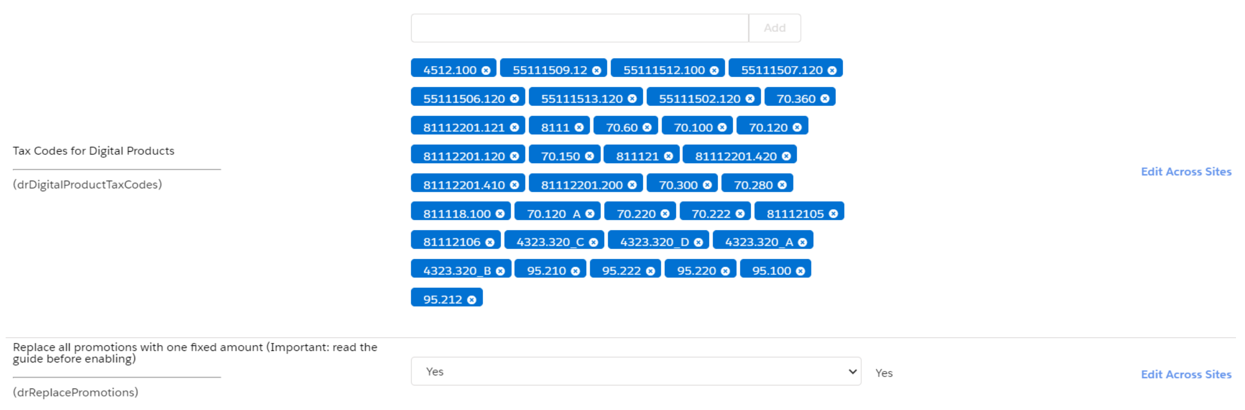

Digital River

Cancel | Apply to Other

Instance Type: Sandbox

Search by ID

Name	Value	Default Value	
Enable Digital River (drUseDropInFeature)	Yes		Edit Across Sites
API Key (drAPIKey)	·············		Edit Across Sites
Digital River public key (drPublicKey) (String) Public key that you get in your Digital River dashboard	Public key that you get in your Digital River dashboard		Edit Across Sites
Webhook Signature Token (drWebhookSignatureToken)			Edit Across Sites
Digital River ship from country* (drShipFromCountry) (String)	GB		Edit Across Sites
Digital River ship from state (drShipFromState) (String)			Edit Across Sites
Digital River ship from city* (drShipFromCity) (String)	London		Edit Across Sites
Digital River ship from address line 1* (drShipFromAddress1) (String)	73 Station Road		Edit Across Sites
Digital River ship from address line 2 (drShipFromAddress2) (String)			Edit Across Sites
Digital River ship from postal code* (drShipFromPostalCode) (String)	SE85 2JS		Edit Across Sites
Modified SKU's update requested (drStartDeltaJob)	None	No	Edit Across Sites
All SKU's update requested (drStartAllSkusJob)	None	No	Edit Across Sites
Enable US Tax Exemptions (drUseUSTaxExemptions)	Yes		Edit Across Sites
Technical Email Address (drTechnicalEmailAddress) (String)			Edit Across Sites
DigitalRiver Default Entity (drDefaultEntity) (String)			Edit Across Sites



Tax Codes for Digital Products

(drDigitalProductTaxCodes)

4512.100, 55111509.12, 55111512.100, 55111507.120, 55111506.120, 55111513.120, 55111502.120, 70.360, 81112201.121, 8111, 70.60, 70.100, 70.120, 81112201.120, 70.150, 811121, 81112201.420, 81112201.410, 81112201.200, 70.300, 70.280, 811118.100, 70.120_A, 70.220, 70.222, 81112105, 81112106, 4323.320_C, 4323.320_D, 4323.320_A, 4323.320_B, 95.210, 95.222, 95.220, 95.100, 95.212

Replace all promotions with one fixed amount (Important: read the guide before enabling)

(drReplacePromotions)

Yes

Yes

#	Field Title	Description
1	Enable Digital River	Turns on/off DigitalRiver integration
2	* API Key	Digital River publishable API Key
3	* DigitalRiver public key	Your public API key
4	* Webhook Signature Token	Webhook Signature Token
5	** Digital River ship from country	The ship-from country for physical products
6	Digital River ship from state	The ship-from state for physical products
7	** Digital River ship from city	The ship-from city for physical products
8	** Digital River ship from address line 1	The ship-from address line 1 for physical products
9	Digital River ship from address line 2	The ship-from address line 2 for physical products
10	** Digital River ship from postal code	The ship-from postal code of address for physical products
11	Modified SKU's update requested	The custom attribute (flag) updated from the Business Manager extension.
12	All SKU's update requested	The custom attribute (flag) updated from the Business Manager extension.
13	Enable US Tax Exemptions	Enable processing of tax certificates.
14	Technical Email Address	The email address where the webhook notifications will be sent.

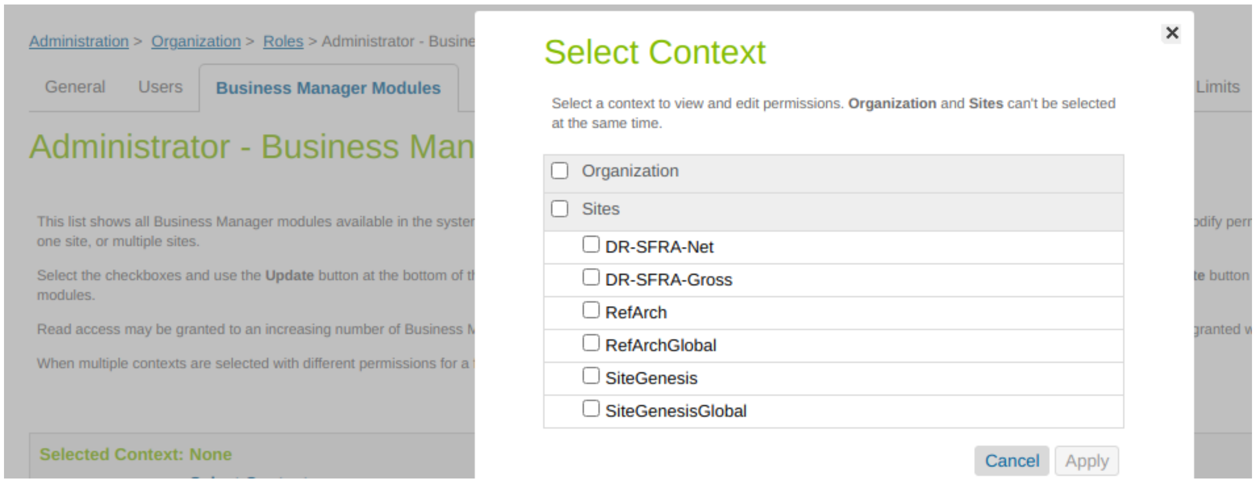
15	DigitalRiver Default Entity	The default selling entity used for compliance. Note that this value must match the [selling entity](#) format expected by the Digital River API and will be provided by Digital River.
16	Tax Codes for Digital Products	
17	Replace all promotions with one fixed amount	

* These values are provided by Digital River.
** These values are required

Business Manager Roles & Permissions

To enable the *DigitalRiver* BM extensions, the permission needs to be added to the targeted access role. This is optional to enable the *Digital River Service Checker* and *Trigger Delta Job Run* modules.

In Business Manager, go to *Administration > Roles & Permissions*. Click on the role to modify (or create a new one) to add the *DigitalRiver* modules to that role. Click on Business Manager Modules within the role.


Administration > Organization > Roles > Administrator - Busine

General Users **Business Manager Modules** Limits

Administrator - Business Man

This list shows all Business Manager modules available in the syster … one site, or multiple sites.

Select the checkboxes and use the **Update** button at the bottom of t … modules.

Read access may be granted to an increasing number of Business M …

When multiple contexts are selected with different permissions for a …

Selected Context: None

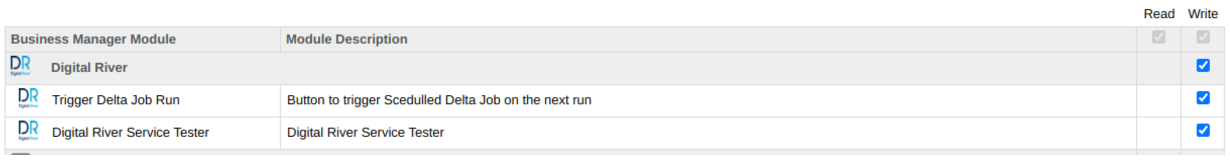
Select Context

Select a context to view and edit permissions. **Organization** and **Sites** can't be selected at the same time.

- ☐ Organization
- ☐ Sites
  - ☐ DR-SFRA-Net
  - ☐ DR-SFRA-Gross
  - ☐ RefArch
  - ☐ RefArchGlobal
  - ☐ SiteGenesis
  - ☐ SiteGenesisGlobal

Cancel Apply

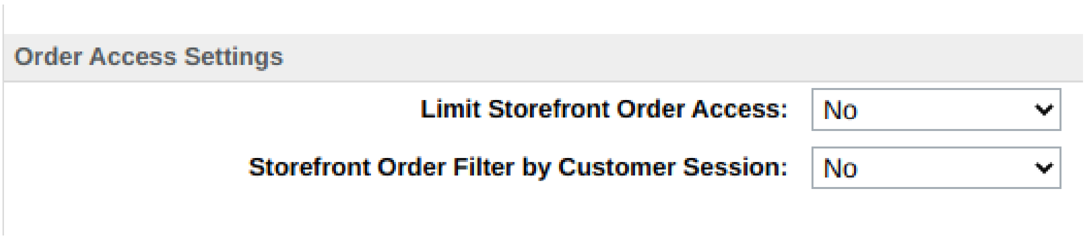
In the next menu, add the Write permission to the *Digital River* group as indicated below.



Business Manager Module	Module Description	Read	Write
Digital River			✓
Trigger Delta Job Run	Button to trigger Scedulled Delta Job on the next run		✓
Digital River Service Tester	Digital River Service Tester		✓

Lastly, the order status webhook will need to be able to access the order to update status on the order properly. To enable this, settings on the site must be changed.

Merchant Tools > Site Preferences > Order
Storefront Order Filter by Customer Session: No
Limit Storefront Order Access: No


Order Access Settings

Limit Storefront Order Access: No

Storefront Order Filter by Customer Session: No

Custom Code

NOTE: *Cartridge int_digitalriver_sfra already contains all templates and scripts changes described in this section. All you need to do is to compile client side scripts and styles, afterwards upload the cartridge and add it to your storefront cartridge path. This guide does not include the changes that have been added to the digitalriver_test and customercredit_ui_mockup cartridges needed to test the functionality of the DigitalRiver cartridge.*

TEMPLATES:

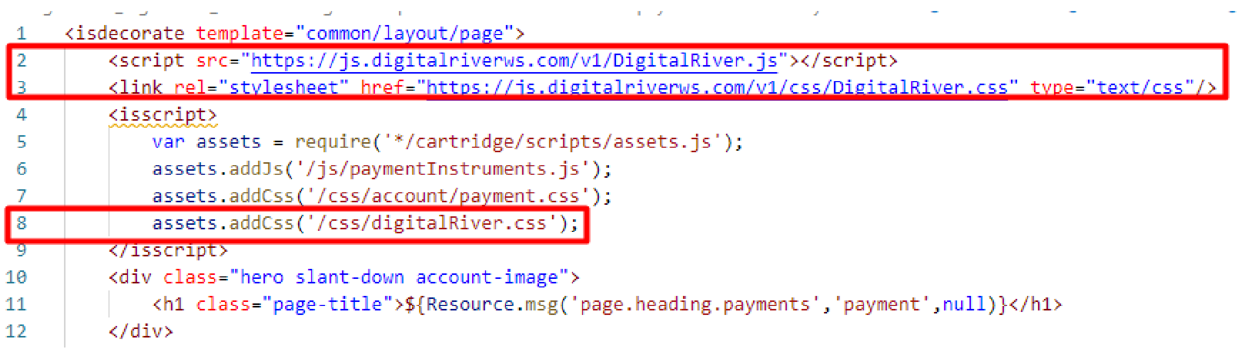
1) Template **cartridge\templates\default\account\payment\addPayment.isml**

Add DigitalRiver script and styles to init drop in functionality on the page

```
<script src="https://js.digitalriverws.com/v1/DigitalRiver.js"></script>
<link rel="stylesheet" href="https://js.digitalriverws.com/v1/css/DigitalRiver.css"
type="text/css"/>
```

As well as drop in styles

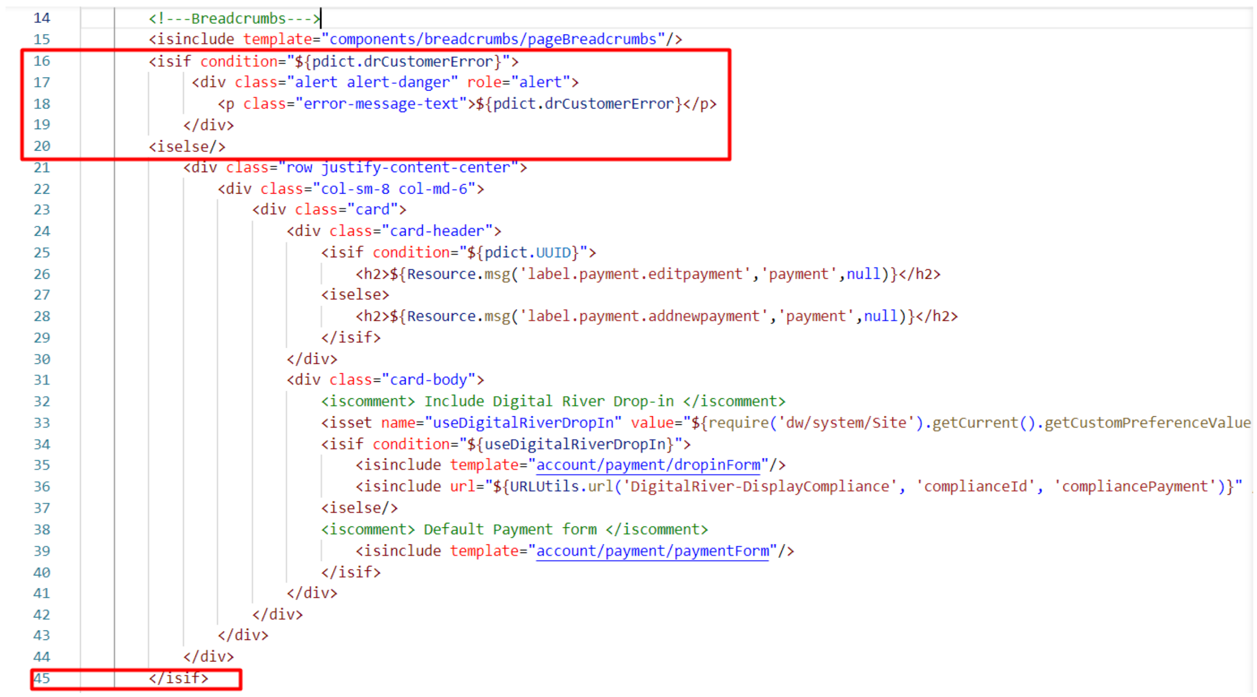
```
assets.addCss('/css/digitalRiver.css');
```



```
<isdecorate template="common/layout/page">
    <script src="https://js.digitalriverws.com/v1/DigitalRiver.js"></script>
    <link rel="stylesheet" href="https://js.digitalriverws.com/v1/css/DigitalRiver.css" type="text/css"/>
    <isscript>
        var assets = require('*/cartridge/scripts/assets.js');
        assets.addJs('/js/paymentInstruments.js');
        assets.addCss('/css/account/payment.css');
        assets.addCss('/css/digitalRiver.css');
    </isscript>
    <div class="hero slant-down account-image">
        <h1 class="page-title">${Resource.msg('page.heading.payments','payment',null)}</h1>
    </div>
```

Add the following condition

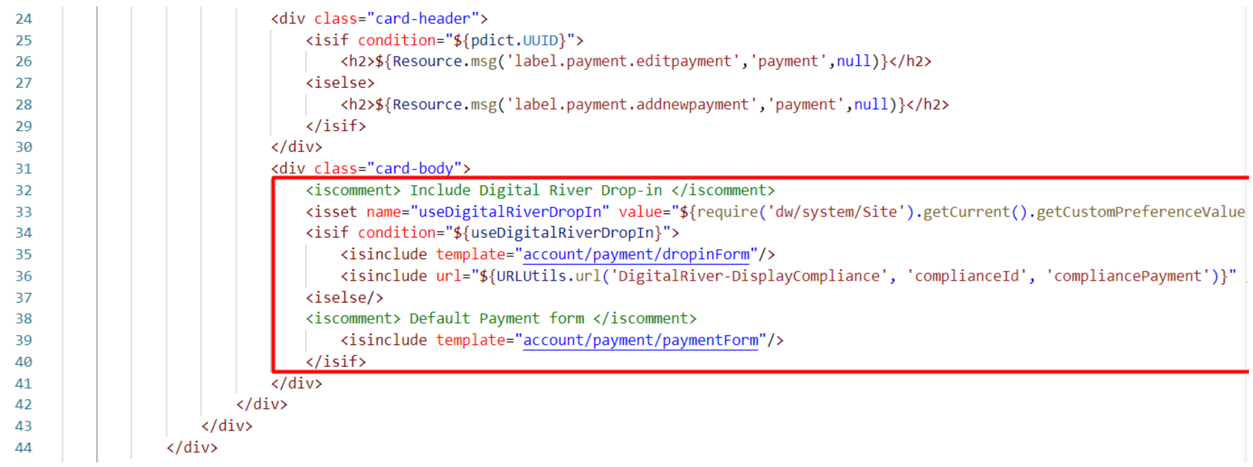
```
<isif condition="${pdict.drCustomerError}">
    <div class="alert alert-danger" role="alert">
        <p class="error-message-text">${pdict.drCustomerError}</p>
    </div>
<iselse/>
…
</isif>
```



```
<!---Breadcrumbs--->
<isinclude template="components/breadcrumbs/pageBreadcrumbs"/>
<isif condition="${pdict.drCustomerError}">
     <div class="alert alert-danger" role="alert">
        <p class="error-message-text">${pdict.drCustomerError}</p>
    </div>
<iselse/>
    <div class="row justify-content-center">
        <div class="col-sm-8 col-md-6">
            <div class="card">
                <div class="card-header">
                    <isif condition="${pdict.UUID}">
                        <h2>${Resource.msg('label.payment.editpayment','payment',null)}</h2>
                    <iselse>
                        <h2>${Resource.msg('label.payment.addnewpayment','payment',null)}</h2>
                    </isif>
                </div>
                <div class="card-body">
                    <iscomment> Include Digital River Drop-in </iscomment>
                    <isset name="useDigitalRiverDropIn" value="${require('dw/system/Site').getCurrent().getCustomPreferenceValue
                    <isif condition="${useDigitalRiverDropIn}">
                        <isinclude template="account/payment/dropinForm"/>
                        <isinclude url="${URLUtils.url('DigitalRiver-DisplayCompliance', 'complianceId', 'compliancePayment')}"
                    <iselse/>
                    <iscomment> Default Payment form </iscomment>
                        <isinclude template="account/payment/paymentForm"/>
                    </isif>
                </div>
            </div>
        </div>
    </div>
</isif>
```

Put following condition inside card-body div

```
<iscomment> Include Digital River Drop-in </iscomment>
<isset name="useDigitalRiverDropIn"
value="${require('dw/system/Site').getCurrent().getCustomPreferenceValue('drUseDropIn
Feature')}" scope="page" />
<isif condition="${useDigitalRiverDropIn}">
    <isinclude template="account/payment/dropinForm"/>
    <isinclude url="${URLUtils.url('DigitalRiver-DisplayCompliance', 'complianceId',
'compliancePayment')}" />
<iselse/>
    <iscomment> Default Payment form </iscomment>
    <isinclude template="account/payment/paymentForm"/>
</isif>
```

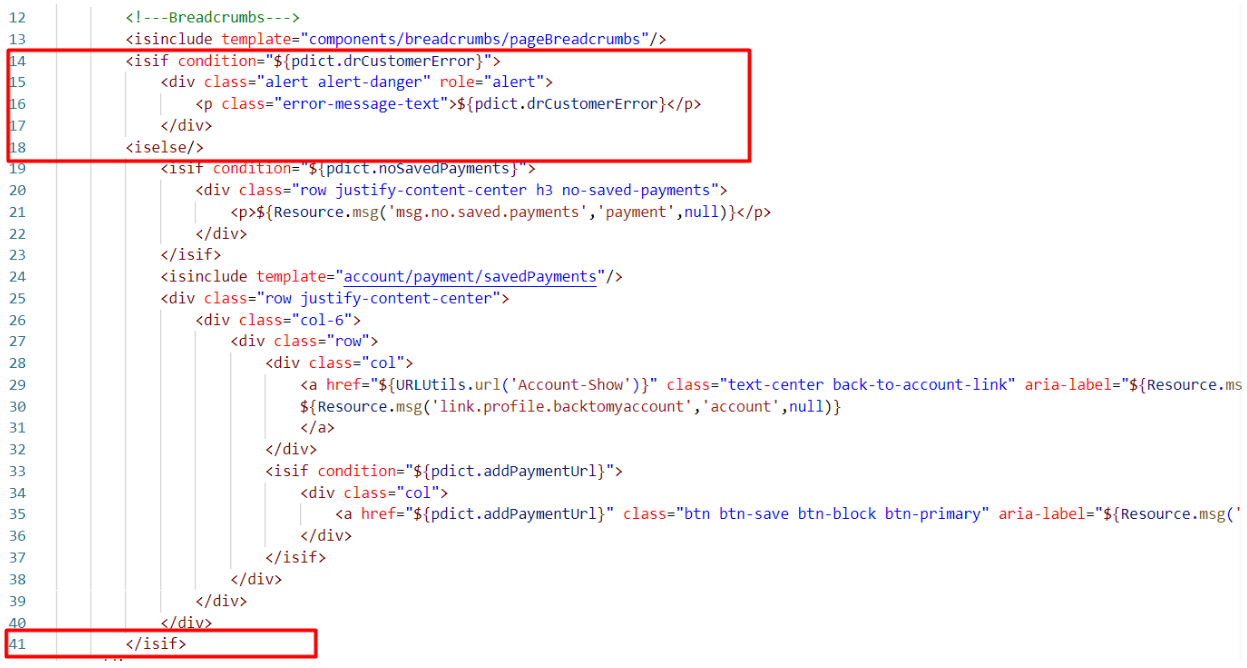


```
<div class="card-header">
    <isif condition="${pdict.UUID}">
        <h2>${Resource.msg('label.payment.editpayment','payment',null)}</h2>
    <iselse>
        <h2>${Resource.msg('label.payment.addnewpayment','payment',null)}</h2>
    </isif>
</div>
<div class="card-body">
    <iscomment> Include Digital River Drop-in </iscomment>
    <isset name="useDigitalRiverDropIn" value="${require('dw/system/Site').getCurrent().getCustomPreferenceValue
    <isif condition="${useDigitalRiverDropIn}">
        <isinclude template="account/payment/dropinForm"/>
        <isinclude url="${URLUtils.url('DigitalRiver-DisplayCompliance', 'complianceId', 'compliancePayment')}"
    <iselse/>
    <iscomment> Default Payment form </iscomment>
        <isinclude template="account/payment/paymentForm"/>
    </isif>
</div>
        </div>
    </div>
</div>
```

2) Template **templates\default\account\payment\payment.isml**

Add the following condition:

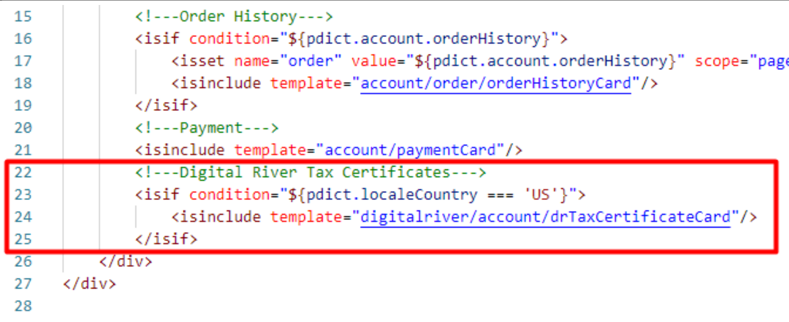
```
<isif condition="${pdict.drCustomerError}">
    <div class="alert alert-danger" role="alert">
        <p class="error-message-text">${pdict.drCustomerError}</p>
    </div>
<iselse/>
…
</isif>
```



```
        <!---Breadcrumbs--->
        <isinclude template="components/breadcrumbs/pageBreadcrumbs"/>
        <isif condition="${pdict.drCustomerError}">
            <div class="alert alert-danger" role="alert">
                <p class="error-message-text">${pdict.drCustomerError}</p>
            </div>
        <iselse/>
            <isif condition="${pdict.noSavedPayments}">
                <div class="row justify-content-center h3 no-saved-payments">
                    <p>${Resource.msg('msg.no.saved.payments','payment',null)}</p>
                </div>
            </isif>
            <isinclude template="account/payment/savedPayments"/>
            <div class="row justify-content-center">
                <div class="col-6">
                    <div class="row">
                        <div class="col">
                            <a href="${URLUtils.url('Account-Show')}" class="text-center back-to-account-link" aria-label="${Resource.ms
                            ${Resource.msg('link.profile.backtomyaccount','account',null)}
                            </a>
                        </div>
                        <isif condition="${pdict.addPaymentUrl}">
                            <div class="col">
                                <a href="${pdict.addPaymentUrl}" class="btn btn-save btn-block btn-primary" aria-label="${Resource.msg('
                            </div>
                        </isif>
                    </div>
                </div>
            </div>
        </isif>
```

3) Template **templates\default\account\dashboardProfileCards.isml**
Next change allows you to display tax certificates in My account only for the US locale.
Add the Digital River tax certificate section on the page (this section will available for US locale only):

```
<!---Digital River Tax Certificates--->
<isif condition="${pdict.localeCountry === 'US'}">
    <isinclude template="digitalriver/account/drTaxCertificateCard"/>
</isif>
```



```
        <!---Order History--->
        <isif condition="${pdict.account.orderHistory}">
            <isset name="order" value="${pdict.account.orderHistory}" scope="page"/>
            <isinclude template="account/order/orderHistoryCard"/>
        </isif>
        <!---Payment--->
        <isinclude template="account/paymentCard"/>
        <!---Digital River Tax Certificates--->
        <isif condition="${pdict.localeCountry === 'US'}">
            <isinclude template="digitalriver/account/drTaxCertificateCard"/>
        </isif>
    </div>
</div>
```

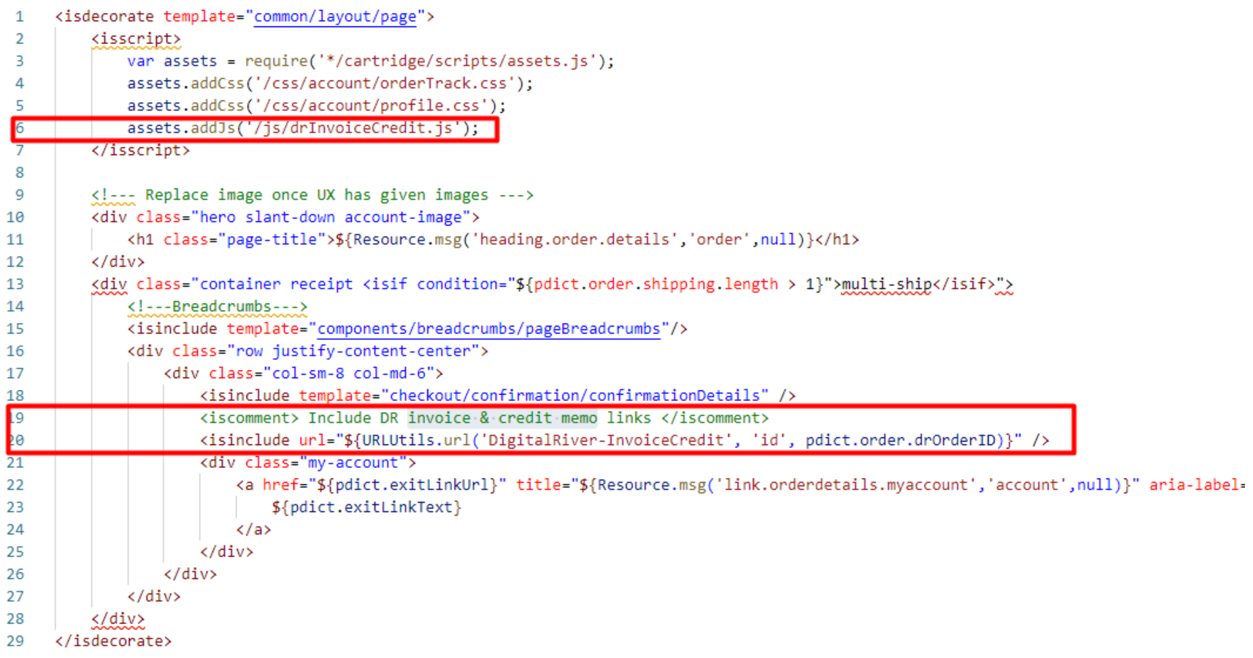
4) Template **cartridge\templates\default\account\orderDetail.isml**

As well as drop in styles

```
assets.addJs('/js/drInvoiceCredit.js');
```

Put following condition to the line 19

```
<iscomment> Include DR invoice & credit memo links </iscomment>
<isinclude url="${URLUtils.url('DigitalRiver-InvoiceCredit', 'id',
pdict.order.drOrderID)}" />
```



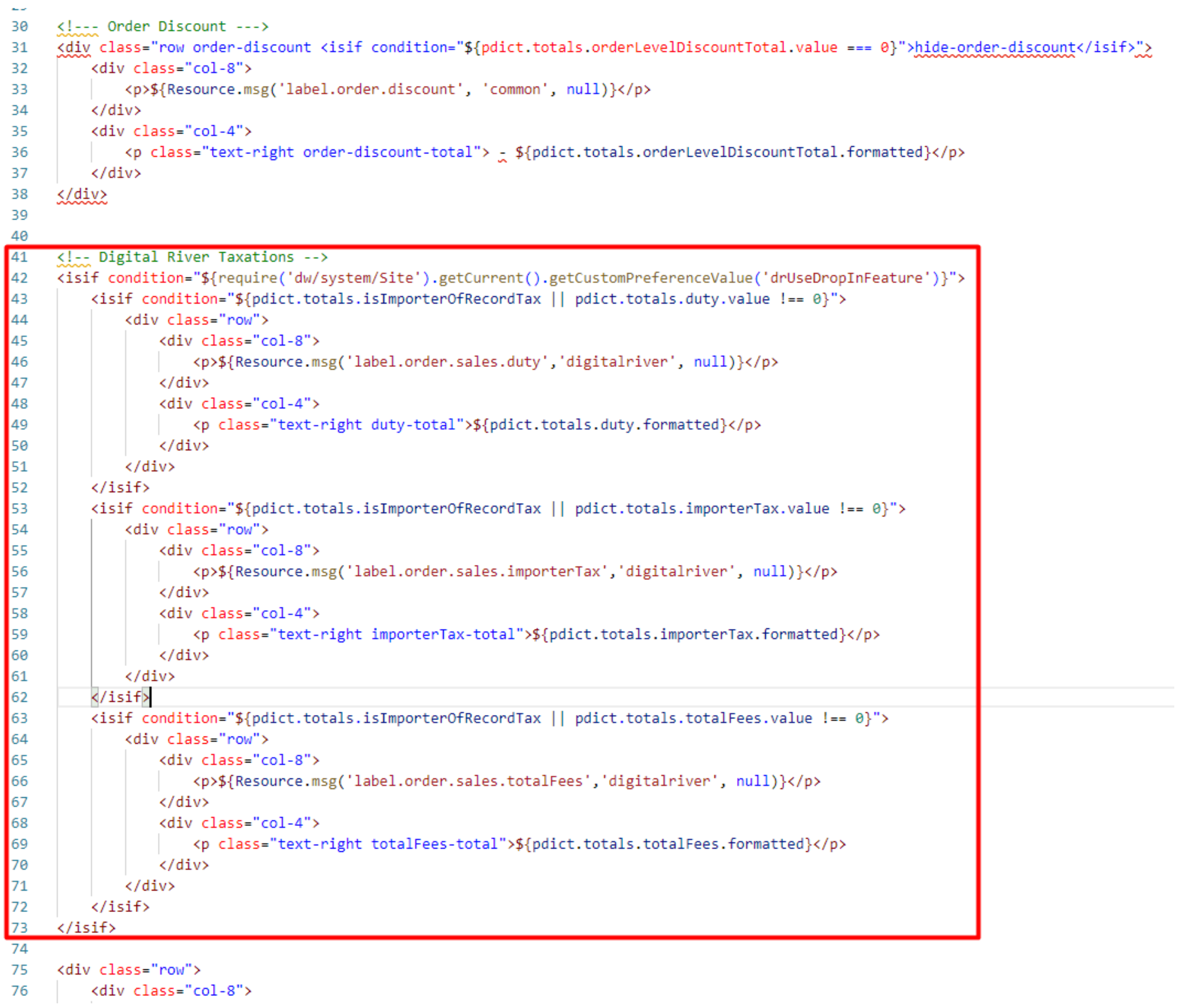
```
1   <isdecorate template="common/layout/page">
2       <isscript>
3           var assets = require('*/cartridge/scripts/assets.js');
4           assets.addCss('/css/account/orderTrack.css');
5           assets.addCss('/css/account/profile.css');
6           assets.addJs('/js/drInvoiceCredit.js');
7       </isscript>
8
9       <!--- Replace image once UX has given images --->
10      <div class="hero slant-down account-image">
11          <h1 class="page-title">${Resource.msg('heading.order.details','order',null)}</h1>
12      </div>
13      <div class="container receipt <isif condition="${pdict.order.shipping.length > 1}">multi-ship</isif>">
14          <!---Breadcrumbs--->
15          <isinclude template="components/breadcrumbs/pageBreadcrumbs"/>
16          <div class="row justify-content-center">
17              <div class="col-sm-8 col-md-6">
18                  <isinclude template="checkout/confirmation/confirmationDetails" />
19                  <iscomment> Include DR invoice & credit memo links </iscomment>
20                  <isinclude url="${URLUtils.url('DigitalRiver-InvoiceCredit', 'id', pdict.order.drOrderID)}" />
21                  <div class="my-account">
22                      <a href="${pdict.exitLinkUrl}" title="${Resource.msg('link.orderdetails.myaccount','account',null)}" aria-label=
23                          ${pdict.exitLinkText}
24                      </a>
25                  </div>
26              </div>
27          </div>
28      </div>
29  </isdecorate>
```

5) Template **cartridge\templates\default\cart\cartTotals.isml**

Include Digital River Taxations section in cart totals template after discount section

```
<!-- Digital River Taxations -->
<isif
condition="${require('dw/system/Site').getCurrent().getCustomPreferenceValue('drUseDr
opInFeature')}">
    <isif condition="${pdict.totals.isImporterOfRecordTax || pdict.totals.duty.value
!== 0}">
        <div class="row">
            <div class="col-8">
                <p>${Resource.msg('label.order.sales.duty','digitalriver', null)}</p>
            </div>
            <div class="col-4">
                <p class="text-right duty-total">${pdict.totals.duty.formatted}</p>
            </div>
        </div>
    </isif>
    <isif condition="${pdict.totals.isImporterOfRecordTax ||
pdict.totals.importerTax.value !== 0}">
        <div class="row">
            <div class="col-8">
                <p>${Resource.msg('label.order.sales.importerTax','digitalriver',
null)}</p>
            </div>
            <div class="col-4">
```

```
                <p class="text-right
importerTax-total">${pdict.totals.importerTax.formatted}</p>
            </div>
        </div>
    </isif>
    <isif condition="${pdict.totals.isImporterOfRecordTax ||
pdict.totals.totalFees.value !== 0}">
        <div class="row">
            <div class="col-8">
                <p>${Resource.msg('label.order.sales.totalFees','digitalriver',
null)}</p>
            </div>
            <div class="col-4">
                <p class="text-right
totalFees-total">${pdict.totals.totalFees.formatted}</p>
            </div>
        </div>
    </isif>
</isif>
```

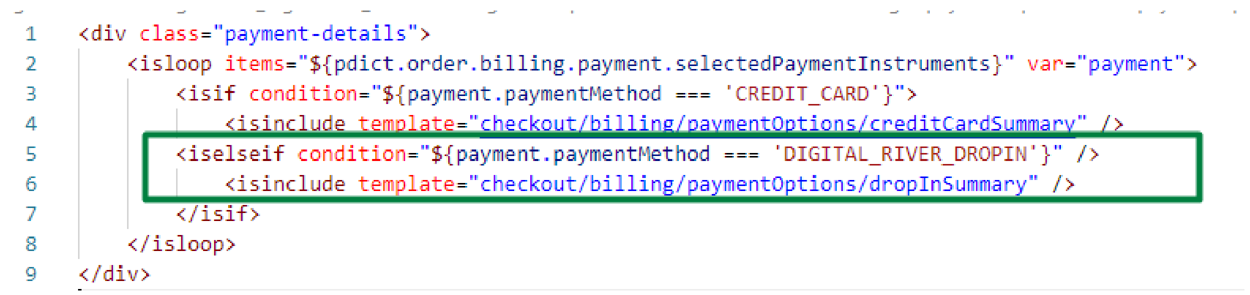


```
<!--- Order Discount --->
<div class="row order-discount <isif condition="${pdict.totals.orderLevelDiscountTotal.value === 0}">hide-order-discount</isif>">
    <div class="col-8">
        <p>${Resource.msg('label.order.discount', 'common', null)}</p>
    </div>
    <div class="col-4">
        <p class="text-right order-discount-total"> - ${pdict.totals.orderLevelDiscountTotal.formatted}</p>
    </div>
</div>


<!-- Digital River Taxations -->
<isif condition="${require('dw/system/Site').getCurrent().getCustomPreferenceValue('drUseDropInFeature')}">
    <isif condition="${pdict.totals.isImporterOfRecordTax || pdict.totals.duty.value !== 0}">
        <div class="row">
            <div class="col-8">
                <p>${Resource.msg('label.order.sales.duty','digitalriver', null)}</p>
            </div>
            <div class="col-4">
                <p class="text-right duty-total">${pdict.totals.duty.formatted}</p>
            </div>
        </div>
    </isif>
    <isif condition="${pdict.totals.isImporterOfRecordTax || pdict.totals.importerTax.value !== 0}">
        <div class="row">
            <div class="col-8">
                <p>${Resource.msg('label.order.sales.importerTax','digitalriver', null)}</p>
            </div>
            <div class="col-4">
                <p class="text-right importerTax-total">${pdict.totals.importerTax.formatted}</p>
            </div>
        </div>
    </isif>
    <isif condition="${pdict.totals.isImporterOfRecordTax || pdict.totals.totalFees.value !== 0}">
        <div class="row">
            <div class="col-8">
                <p>${Resource.msg('label.order.sales.totalFees','digitalriver', null)}</p>
            </div>
            <div class="col-4">
                <p class="text-right totalFees-total">${pdict.totals.totalFees.formatted}</p>
            </div>
        </div>
    </isif>
</isif>

<div class="row">
    <div class="col-8">
```

6) Template
cartridge\templates\default\checkout\billing\paymentOptions\paymentOptionsSummary.isml

Extend condition with ‘DIGITAL_RIVER_DROPIN’ payment method

```
<iselseif condition="${payment.paymentMethod === 'DIGITAL_RIVER_DROPIN'}" />
    <isinclude template="checkout/billing/paymentOptions/dropInSummary" />
```



```
<div class="payment-details">
    <isloop items="${pdict.order.billing.payment.selectedPaymentInstruments}" var="payment">
        <isif condition="${payment.paymentMethod === 'CREDIT_CARD'}">
            <isinclude template="checkout/billing/paymentOptions/creditCardSummary" />
        <iselseif condition="${payment.paymentMethod === 'DIGITAL_RIVER_DROPIN'}" />
            <isinclude template="checkout/billing/paymentOptions/dropInSummary" />
        </isif>
    </isloop>
</div>
```

7) Template **cartridge\templates\default\checkout\billing\billing.isml**

Add at the beginning of the file the line

```
<isset name="isDRDropInEnabled"
value="${require('dw/system/Site').getCurrent().getCustomPreferenceValue('drUseDropIn
Feature')}" scope="page" />
```

Include accordion components to the billing page

```
<iscomment> DigitalRiver accordion section </iscomment>
<isif condition="${pdict.digitalRiverUseDropInFeature}">
    <div class="accordion" id="accordionBilling"
data-digital-cart="${pdict.isDigitalCart}">
        <div class="dr-accordion-card">
            <div class="dr-accordion-card-header" id="headingBilling">
                <h2 class="mb-0">
                    <span class="btn btn-link dr-accordion-btn"
data-toggle="collapse" data-target="#collapse-billing" aria-expanded="true"
aria-controls="collapse-billing">
                                    ${Resource.msg('button.billing', 'digitalriver',
null)}
                     </span>
                 </h2>
             </div>
             <div id="collapse-billing" class="collapse show"
aria-labelledby="headingBilling" data-parent="#accordionBilling">
                  <div class="dr-accordion-card-body">
</isif>
```

```
            <isinclude template="digitalriver/checkout/billing/drPurchaseTypeSection"
/>

            <iscomment> DigitalRiver accordion section </iscomment>
            <isif condition="${pdict.digitalRiverUseDropInFeature}">
                </div></div></div>

                <iscomment>DigitalRiver tax identifier</iscomment>
                <div id="tax-id-accordion" class="dr-accordion-card
${pdict.digitalRiverUseTaxIdentifier ? '' : 'hidden'}">
```

```
                    <div class="dr-accordion-card-header" id="headingTaxIdentifier">
                        <h2 class="mb-0">
                            <span class="btn btn-link dr-accordion-btn"
data-toggle="collapse" data-target="#collapse-taxidentifier" aria-expanded="false"
aria-controls="collapse-taxidentifier">
                                ${Resource.msg('button.taxidentifier',
'digitalriver', null)}
                            </span>
                        </h2>
                    </div>
                    <div id="collapse-taxidentifier" class="collapse"
aria-labelledby="headingTaxIdentifier" data-parent="#accordionBilling">
                        <div class="dr-accordion-card-body">
                            <div id="tax-id"
data-config-url="${URLUtils.url('DigitalRiver-TaxIdentifierConfig')}"></div>
                            <button class="btn btn-block btn-primary
dr-btn-taxidentifier-submit" type="button"
data-apply-url="${URLUtils.url('DigitalRiver-TaxIdentifierApply')}">
                                ${Resource.msg('button.submit.taxidentifier',
'digitalriver', null)}
                            </button>
                            <div id="dr-list-of-applied-identifiers"
data-delete-url="${URLUtils.url('DigitalRiver-TaxIdentifierDelete')}"></div>
                        </div>
                    </div>
                </div>
                <iscomment>End of DigitalRiver tax identifier</iscomment>

                <div class="dr-accordion-card">
                    <div class="dr-accordion-card-header
${pdict.order.totals.isZeroTotal ? 'digitalriver-hide' : ''}" id="headingPayment">
                        <h2 class="mb-0">
                            <span class="btn btn-link dr-accordion-btn"
data-toggle="collapse" data-target="#collapse-payment" aria-expanded="false"
aria-controls="collapse-payment">
                                ${Resource.msg('button.payment', 'digitalriver',
null)}
                            </span>
                        </h2>
                    </div>
                    <div id="collapse-payment" class="collapse"
aria-labelledby="headingPayment" data-parent="#accordionBilling">
                        <div class="dr-accordion-card-body">
            </isif>
```

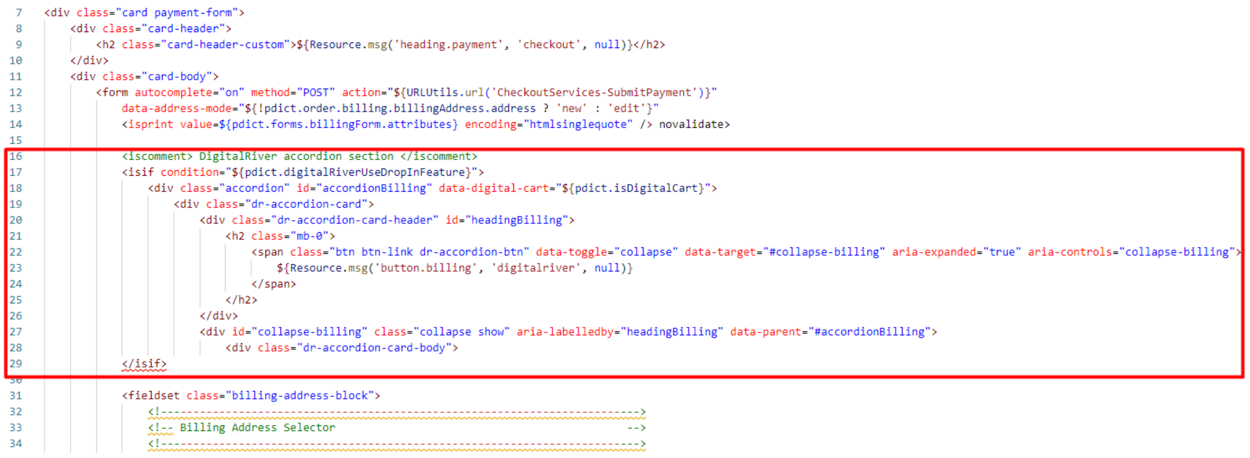
```
<iscomment> DigitalRiver accordion section </iscomment>
<isif condition="${pdict.digitalRiverUseDropInFeature}">
    </div></div></div></div>
</isif>
```

Include Digital River payment options template

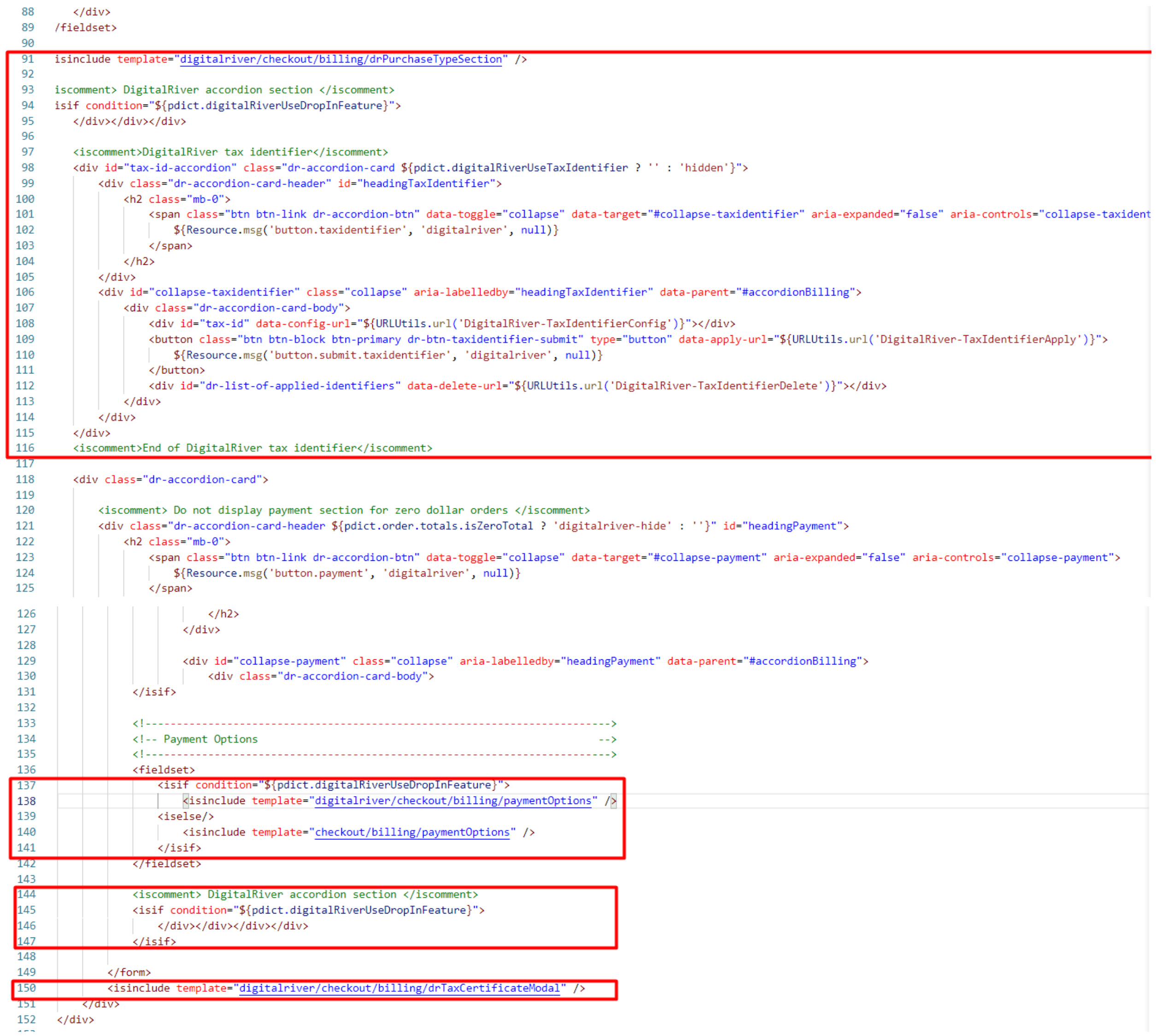
```
<isif condition="${pdict.digitalRiverUseDropInFeature}">
    <isinclude template="digitalriver/checkout/billing/paymentOptions" />
<iselse/>
    <isinclude template="checkout/billing/paymentOptions" />
</isif>
```

Include Digital River tax cetrificate template

```
<isinclude template="digitalriver/checkout/billing/drTaxCertificateModal" />
```



```
<div class="card payment-form">
    <div class="card-header">
        <h2 class="card-header-custom">${Resource.msg('heading.payment', 'checkout', null)}</h2>
    </div>
    <div class="card-body">
        <form autocomplete="on" method="POST" action="${URLUtils.url('CheckoutServices-SubmitPayment')}"
            data-address-mode="${!pdict.order.billing.billingAddress.address ? 'new' : 'edit'}"
            <isprint value=${pdict.forms.billingForm.attributes} encoding="htmlsinglequote" /> novalidate>

            <iscomment> DigitalRiver accordion section </iscomment>
            <isif condition="${pdict.digitalRiverUseDropInFeature}">
                <div class="accordion" id="accordionBilling" data-digital-cart="${pdict.isDigitalCart}">
                    <div class="dr-accordion-card">
                        <div class="dr-accordion-card-header" id="headingBilling">
                            <h2 class="mb-0">
                                <span class="btn btn-link dr-accordion-btn" data-toggle="collapse" data-target="#collapse-billing" aria-expanded="true" aria-controls="collapse-billing">
                                    ${Resource.msg('button.billing', 'digitalriver', null)}
                                </span>
                            </h2>
                        </div>
                        <div id="collapse-billing" class="collapse show" aria-labelledby="headingBilling" data-parent="#accordionBilling">
                            <div class="dr-accordion-card-body">
            </isif>

            <fieldset class="billing-address-block">
                <!------------------------------------------------------------------------>
                <!-- Billing Address Selector                                            -->
                <!------------------------------------------------------------------------>
```



```
    </div>
/fieldset>

isinclude template="digitalriver/checkout/billing/drPurchaseTypeSection" />

iscomment> DigitalRiver accordion section </iscomment>
isif condition="${pdict.digitalRiverUseDropInFeature}">
    </div></div></div>

    <iscomment>DigitalRiver tax identifier</iscomment>
    <div id="tax-id-accordion" class="dr-accordion-card ${pdict.digitalRiverUseTaxIdentifier ? '' : 'hidden'}">
        <div class="dr-accordion-card-header" id="headingTaxIdentifier">
            <h2 class="mb-0">
                <span class="btn btn-link dr-accordion-btn" data-toggle="collapse" data-target="#collapse-taxidentifier" aria-expanded="false" aria-controls="collapse-taxident
                    ${Resource.msg('button.taxidentifier', 'digitalriver', null)}
                </span>
            </h2>
        </div>
        <div id="collapse-taxidentifier" class="collapse" aria-labelledby="headingTaxIdentifier" data-parent="#accordionBilling">
            <div class="dr-accordion-card-body">
                <div id="tax-id" data-config-url="${URLUtils.url('DigitalRiver-TaxIdentifierConfig')}"></div>
                <button class="btn btn-block btn-primary dr-btn-taxidentifier-submit" type="button" data-apply-url="${URLUtils.url('DigitalRiver-TaxIdentifierApply')}">
                    ${Resource.msg('button.submit.taxidentifier', 'digitalriver', null)}
                </button>
                <div id="dr-list-of-applied-identifiers" data-delete-url="${URLUtils.url('DigitalRiver-TaxIdentifierDelete')}"></div>
            </div>
        </div>
    </div>
    <iscomment>End of DigitalRiver tax identifier</iscomment>

    <div class="dr-accordion-card">

        <iscomment> Do not display payment section for zero dollar orders </iscomment>
        <div class="dr-accordion-card-header ${pdict.order.totals.isZeroTotal ? 'digitalriver-hide' : ''}" id="headingPayment">
            <h2 class="mb-0">
                <span class="btn btn-link dr-accordion-btn" data-toggle="collapse" data-target="#collapse-payment" aria-expanded="false" aria-controls="collapse-payment">
                    ${Resource.msg('button.payment', 'digitalriver', null)}
                </span>
                            </h2>
                        </div>

                        <div id="collapse-payment" class="collapse" aria-labelledby="headingPayment" data-parent="#accordionBilling">
                            <div class="dr-accordion-card-body">
            </isif>

            <!------------------------------------------------------------------------>
            <!-- Payment Options                                                     -->
            <!------------------------------------------------------------------------>
            <fieldset>
                <isif condition="${pdict.digitalRiverUseDropInFeature}">
                    <isinclude template="digitalriver/checkout/billing/paymentOptions" />
                <iselse/>
                    <isinclude template="checkout/billing/paymentOptions" />
                </isif>
            </fieldset>

            <iscomment> DigitalRiver accordion section </iscomment>
            <isif condition="${pdict.digitalRiverUseDropInFeature}">
                </div></div></div></div>
            </isif>

        </form>
        <isinclude template="digitalriver/checkout/billing/drTaxCertificateModal" />
    </div>
</div>
```

8) Template **cartridge\templates\default\checkout\billing\storedPaymentInstruments.isml**

Add code to the start of the file:

```
<isset name="isDRDropInEnabled"
value="${require('dw/system/Site').getCurrent().getCustomPreferenceValue('drUseDropIn
Feature')}" scope="page" />
<div class="invalid-feedback"
id="savedPaymentNotSelectedMessage">${Resource.msg('error.select.stored.card',
'digitalriver', null)}</div>
```

Modify <div> tag with class="row saved-payment-instrument as following:

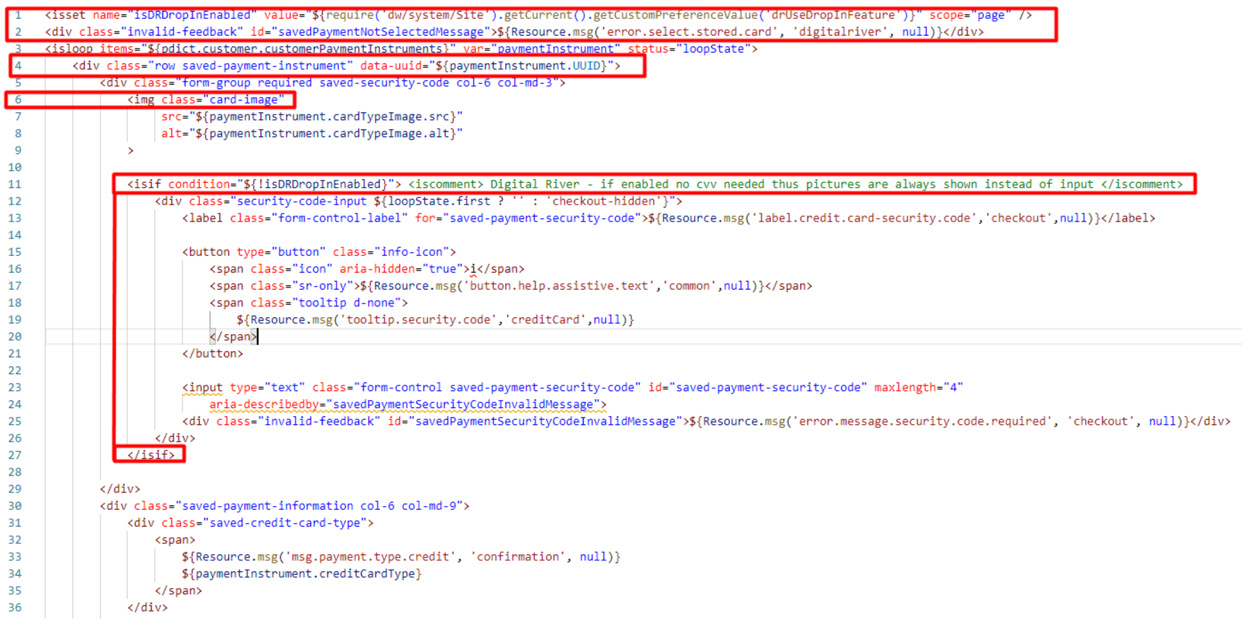
```
<div class="row saved-payment-instrument" data-uuid="${paymentInstrument.UUID}">
```

Modify card-image class condition as following:

```
<img class="card-image
```

Also wrap security-code-input div with condition:

```
<isif condition="${!pdict.digitalRiverUseDropInFeature}"> <iscomment> Didital River -
if enabled no cvv needed thus pictures are always shown instead of input </iscomment>

… security-code-unput goes here

</isif>
```



```
<isset name="isDRDropInEnabled" value="${require('dw/system/Site').getCurrent().getCustomPreferenceValue('drUseDropInFeature')}" scope="page" />
<div class="invalid-feedback" id="savedPaymentNotSelectedMessage">${Resource.msg('error.select.stored.card', 'digitalriver', null)}</div>
<isloop items="${pdict.customer.customerPaymentInstruments}" var="paymentInstrument" status="loopState">
    <div class="row saved-payment-instrument" data-uuid="${paymentInstrument.UUID}">
        <div class="form-group required saved-security-code col-6 col-md-3">
            <img class="card-image"
                 src="${paymentInstrument.cardTypeImage.src}"
                 alt="${paymentInstrument.cardTypeImage.alt}"
            >

            <isif condition="${!isDRDropInEnabled}"> <iscomment> Digital River - if enabled no cvv needed thus pictures are always shown instead of input </iscomment>
                <div class="security-code-input ${loopState.first ? '' : 'checkout-hidden'}">
                    <label class="form-control-label" for="saved-payment-security-code">${Resource.msg('label.credit.card-security.code','checkout',null)}</label>

                    <button type="button" class="info-icon">
                        <span class="icon" aria-hidden="true">i</span>
                        <span class="sr-only">${Resource.msg('button.help.assistive.text','common',null)}</span>
                        <span class="tooltip d-none">
                            ${Resource.msg('tooltip.security.code','creditCard',null)}
                        </span>
                    </button>

                    <input type="text" class="form-control saved-payment-security-code" id="saved-payment-security-code" maxlength="4"
                        aria-describedby="savedPaymentSecurityCodeInvalidMessage">
                    <div class="invalid-feedback" id="savedPaymentSecurityCodeInvalidMessage">${Resource.msg('error.message.security.code.required', 'checkout', null)}</div>
                </div>
            </isif>

        </div>
        <div class="saved-payment-information col-6 col-md-9">
            <div class="saved-credit-card-type">
                <span>
                    ${Resource.msg('msg.payment.type.credit', 'confirmation', null)}
                    ${paymentInstrument.creditCardType}
                </span>
            </div>
```

9) Template **cartridge\templates\default\checkout\confirmation\confirmation.isml**

As well as drop in styles

```
assets.addCss('/css/digitalRiver.css');
```

Add Digital River payment instructions template to the line 31

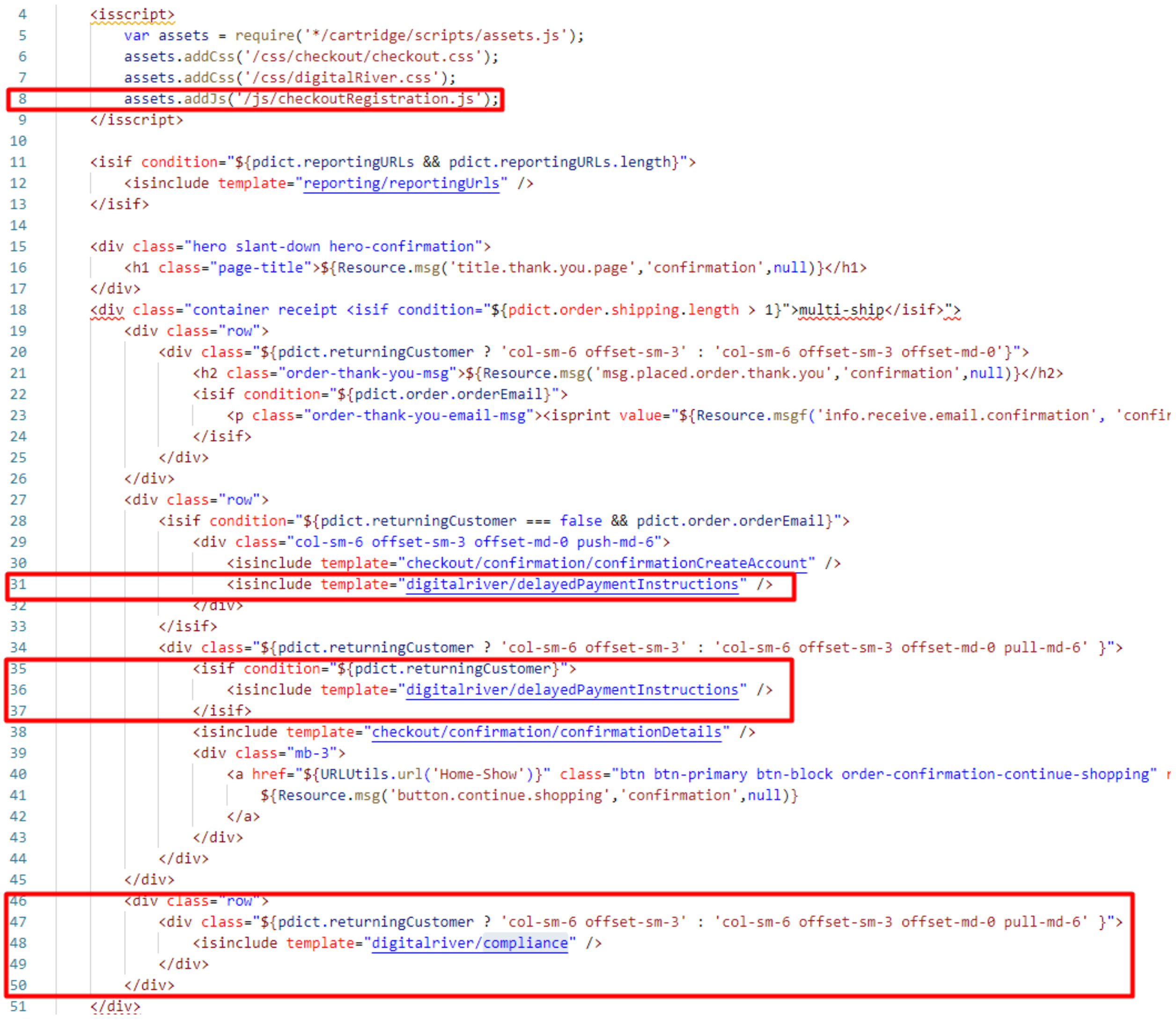
```
<isinclude template="digitalriver/delayedPaymentInstructions" />
```

Add Digital River payment instructions template to the line 35

```
<isif condition="${pdict.returningCustomer}">
    <isinclude template="digitalriver/delayedPaymentInstructions" />
</isif>
```

Add Digital River compliance template to the line 47

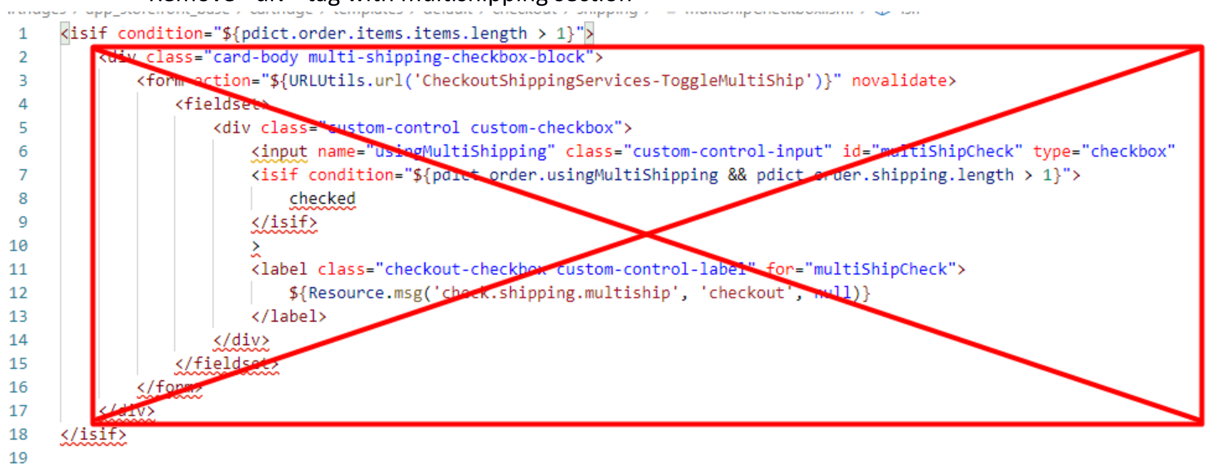
```
<div class="row">
    <div class="${pdict.returningCustomer ? 'col-sm-6 offset-sm-3' : 'col-sm-6 
offset-sm-3 offset-md-0 pull-md-6' }">
        <isinclude template="digitalriver/compliance" />
    </div>
</div>
```



```
4   <isscript>
5       var assets = require('*/cartridge/scripts/assets.js');
6       assets.addCss('/css/checkout/checkout.css');
7       assets.addCss('/css/digitalRiver.css');
8       assets.addJs('/js/checkoutRegistration.js');
9   </isscript>
10
11  <isif condition="${pdict.reportingURLs && pdict.reportingURLs.length}">
12      <isinclude template="reporting/reportingUrls" />
13  </isif>
14
15  <div class="hero slant-down hero-confirmation">
16      <h1 class="page-title">${Resource.msg('title.thank.you.page','confirmation',null)}</h1>
17  </div>
18  <div class="container receipt <isif condition="${pdict.order.shipping.length > 1}">multi-ship</isif>">
19      <div class="row">
20          <div class="${pdict.returningCustomer ? 'col-sm-6 offset-sm-3' : 'col-sm-6 offset-sm-3 offset-md-0'}">
21              <h2 class="order-thank-you-msg">${Resource.msg('msg.placed.order.thank.you','confirmation',null)}</h2>
22              <isif condition="${pdict.order.orderEmail}">
23                  <p class="order-thank-you-email-msg"><isprint value="${Resource.msgf('info.receive.email.confirmation', 'confir
24              </isif>
25          </div>
26      </div>
27      <div class="row">
28          <isif condition="${pdict.returningCustomer === false && pdict.order.orderEmail}">
29              <div class="col-sm-6 offset-sm-3 offset-md-0 push-md-6">
30                  <isinclude template="checkout/confirmation/confirmationCreateAccount" />
31                  <isinclude template="digitalriver/delayedPaymentInstructions" />
32              </div>
33          </isif>
34          <div class="${pdict.returningCustomer ? 'col-sm-6 offset-sm-3' : 'col-sm-6 offset-sm-3 offset-md-0 pull-md-6' }">
35              <isif condition="${pdict.returningCustomer}">
36                  <isinclude template="digitalriver/delayedPaymentInstructions" />
37              </isif>
38              <isinclude template="checkout/confirmation/confirmationDetails" />
39              <div class="mb-3">
40                  <a href="${URLUtils.url('Home-Show')}" class="btn btn-primary btn-block order-confirmation-continue-shopping" r
41                      ${Resource.msg('button.continue.shopping','confirmation',null)}
42                  </a>
43              </div>
44          </div>
45      </div>
46      <div class="row">
47          <div class="${pdict.returningCustomer ? 'col-sm-6 offset-sm-3' : 'col-sm-6 offset-sm-3 offset-md-0 pull-md-6' }">
48              <isinclude template="digitalriver/compliance" />
49          </div>
50      </div>
51  </div>
```

10) Template **cartridge\templates\default\checkout\shipping\multiShippingCheckbox.isml**

Remove <div> tag with multishipping section



```
<isif condition="${pdict.order.items.items.length > 1}">
    <div class="card-body multi-shipping-checkbox-block">
        <form action="${URLUtils.url('CheckoutShippingServices-ToggleMultiShip')}" novalidate>
            <fieldset>
                <div class="custom-control custom-checkbox">
                    <input name="usingMultiShipping" class="custom-control-input" id="multiShipCheck" type="checkbox"
                    <isif condition="${pdict.order.usingMultiShipping && pdict.order.shipping.length > 1}">
                        checked
                    </isif>
                    >
                    <label class="checkout-checkbox custom-control-label" for="multiShipCheck">
                        ${Resource.msg('check.shipping.multiship', 'checkout', null)}
                    </label>
                </div>
            </fieldset>
        </form>
    </div>
</isif>
```

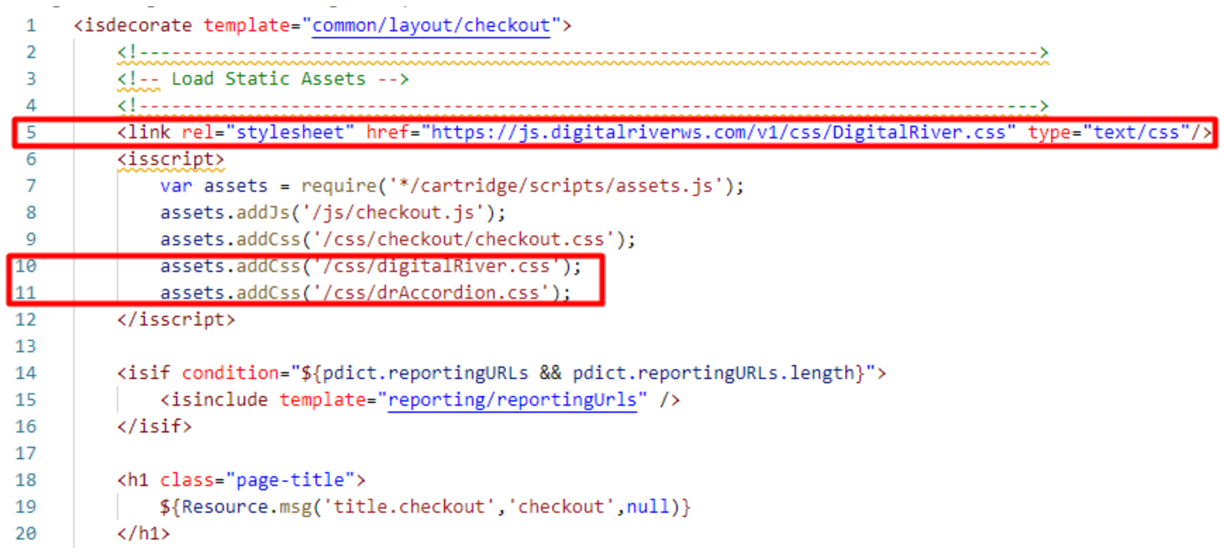
11) Template **cartridge\templates\default\checkout\checkout.isml**

Include remote DigitalRiver styles to the line 5

```
<link rel="stylesheet" href="https://js.digitalriverws.com/v1/css/DigitalRiver.css"
type="text/css"/>
```

Include locale DigitalRiver styles to the line 10

```
assets.addCss('/css/digitalRiver.css');
assets.addCss('/css/drAccordion.css');
```

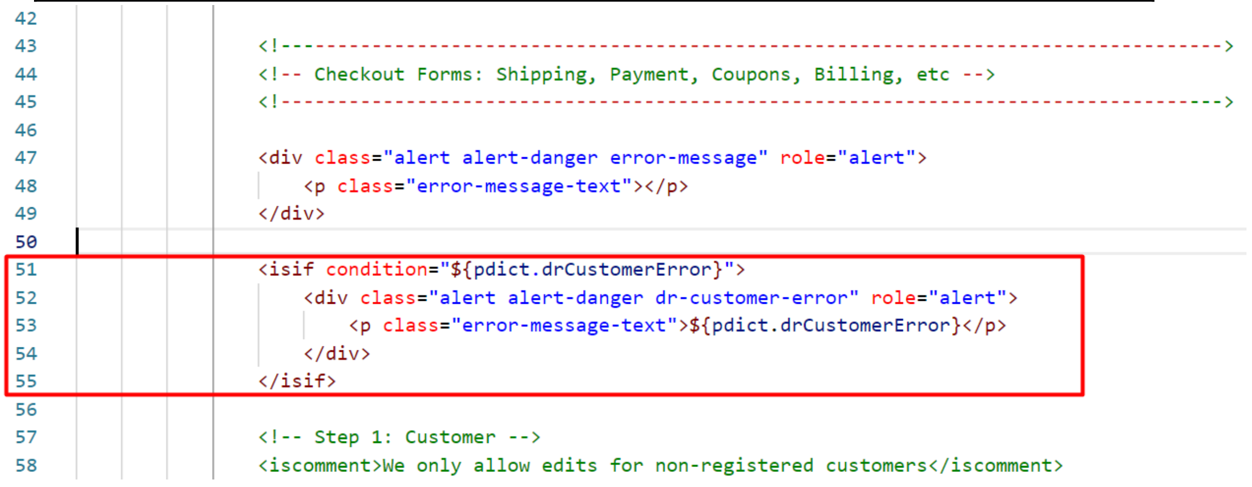


```
1   <isdecorate template="common/layout/checkout">
2       <!----------------------------------------------------------------------------------->
3       <!-- Load Static Assets -->
4       <!----------------------------------------------------------------------------------->
5       <link rel="stylesheet" href="https://js.digitalriverws.com/v1/css/DigitalRiver.css" type="text/css"/>
6       <isscript>
7           var assets = require('*/cartridge/scripts/assets.js');
8           assets.addJs('/js/checkout.js');
9           assets.addCss('/css/checkout/checkout.css');
10          assets.addCss('/css/digitalRiver.css');
11          assets.addCss('/css/drAccordion.css');
12      </isscript>
13
14      <isif condition="${pdict.reportingURLs && pdict.reportingURLs.length}">
15          <isinclude template="reporting/reportingUrls" />
16      </isif>
17
18      <h1 class="page-title">
19          ${Resource.msg('title.checkout','checkout',null)}
20      </h1>
```

Add DigitalRiver template to the Payment and Billing section to the line 51

```
<isif condition="${pdict.drCustomerError}">
    <div class="alert alert-danger dr-customer-error" role="alert">
        <p class="error-message-text">${pdict.drCustomerError}</p>
    </div>
```

```
</isif>
```



```
<!------------------------------------------------------------------------------------->
<!-- Checkout Forms: Shipping, Payment, Coupons, Billing, etc -->
<!------------------------------------------------------------------------------------->

<div class="alert alert-danger error-message" role="alert">
    <p class="error-message-text"></p>
</div>

<isif condition="${pdict.drCustomerError}">
    <div class="alert alert-danger dr-customer-error" role="alert">
        <p class="error-message-text">${pdict.drCustomerError}</p>
    </div>
</isif>

<!-- Step 1: Customer -->
<iscomment>We only allow edits for non-registered customers</iscomment>
```

Add DigitalRiver template to the Payment and Billing section to the line 96

```
<isinclude template="digitalriver/confirmDisclosure" />
```

Modify condition in the line 106

```
<button class="btn btn-primary btn-block submit-shipping" type="submit" name="submit"
value="submit-shipping" <isif condition="${(pdict.order.usingMultiShipping &&
!pdict.order.shippable) || pdict.drCustomerError}">disabled</isif>>
```

Add condition that will control submit payment button display to the line 110

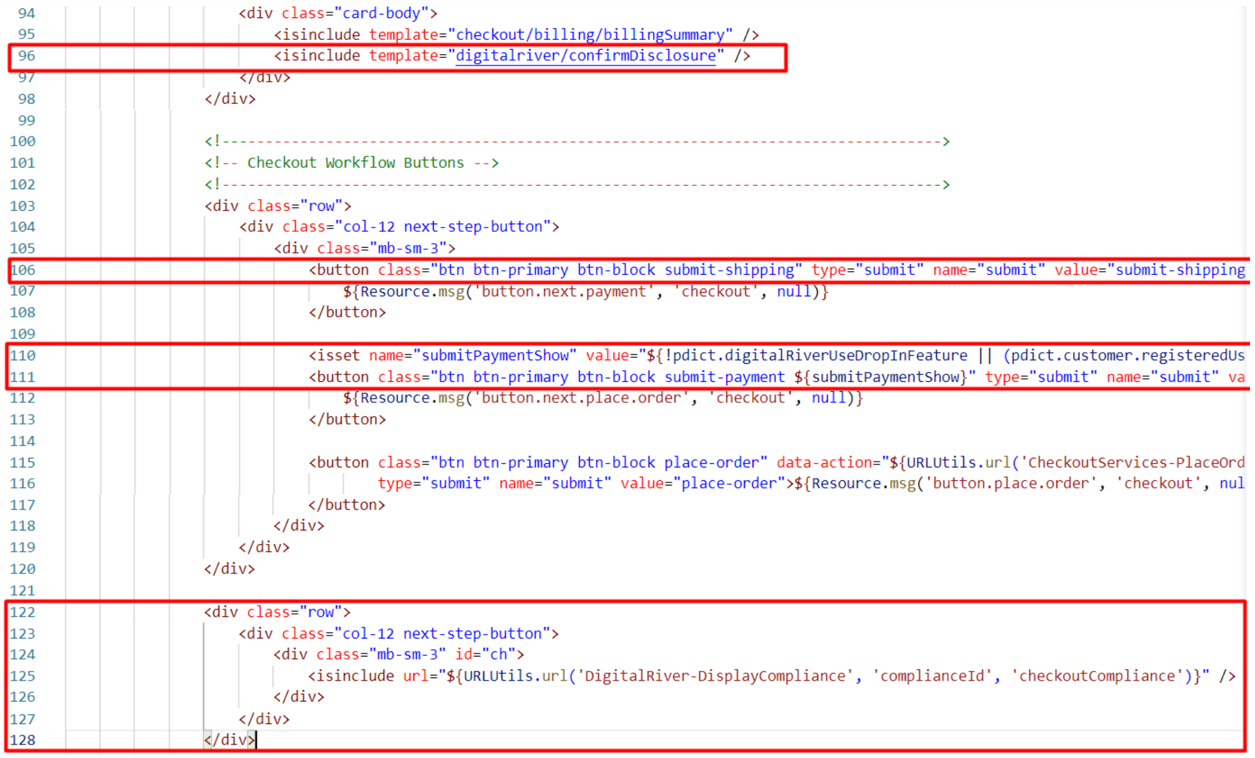
```
<isset name="submitPaymentShow" value="${!pdict.digitalRiverUseDropInFeature
|| (pdict.customer.registeredUser &&
pdict.customer.customerPaymentInstruments.length) ||
pdict.order.totals.isZeroTotal  ? '' : 'digitalriver-hide'}" scope="page" />
```

Modify button in the line 111

```
<button class="btn btn-primary btn-block submit-payment ${submitPaymentShow}"
type="submit" name="submit" value="submit-payment"
data-saved-payments="${pdict.customer.registeredUser &&
pdict.customer.customerPaymentInstruments.length}">
```

Add DigitalRiver Compliance page after Checkout Workflow Buttons to the line 122

```
<div class="row">
    <div class="col-12 next-step-button">
        <div class="mb-sm-3" id="ch">
            <isinclude url="${URLUtils.url('DigitalRiver-DisplayCompliance',
'complianceId', 'checkoutCompliance')}" />
        </div>
    </div>
</div>
```



```
            <div class="card-body">
                <isinclude template="checkout/billing/billingSummary" />
                <isinclude template="digitalriver/confirmDisclosure" />
            </div>
        </div>

        <!-------------------------------------------------------------------------->
        <!-- Checkout Workflow Buttons -->
        <!-------------------------------------------------------------------------->
        <div class="row">
            <div class="col-12 next-step-button">
                <div class="mb-sm-3">
                    <button class="btn btn-primary btn-block submit-shipping" type="submit" name="submit" value="submit-shipping
                        ${Resource.msg('button.next.payment', 'checkout', null)}
                    </button>

                    <isset name="submitPaymentShow" value="${!pdict.digitalRiverUseDropInFeature || (pdict.customer.registeredUs
                    <button class="btn btn-primary btn-block submit-payment ${submitPaymentShow}" type="submit" name="submit" va
                        ${Resource.msg('button.next.place.order', 'checkout', null)}
                    </button>

                    <button class="btn btn-primary btn-block place-order" data-action="${URLUtils.url('CheckoutServices-PlaceOrd
                            type="submit" name="submit" value="place-order">${Resource.msg('button.place.order', 'checkout', nul
                    </button>
                </div>
            </div>
        </div>

        <div class="row">
            <div class="col-12 next-step-button">
                <div class="mb-sm-3" id="ch">
                    <isinclude url="${URLUtils.url('DigitalRiver-DisplayCompliance', 'complianceId', 'checkoutCompliance')}" />
                </div>
            </div>
        </div>
```

12) Template **cartridge\templates\default\checkout\orderTotalSummary.isml**

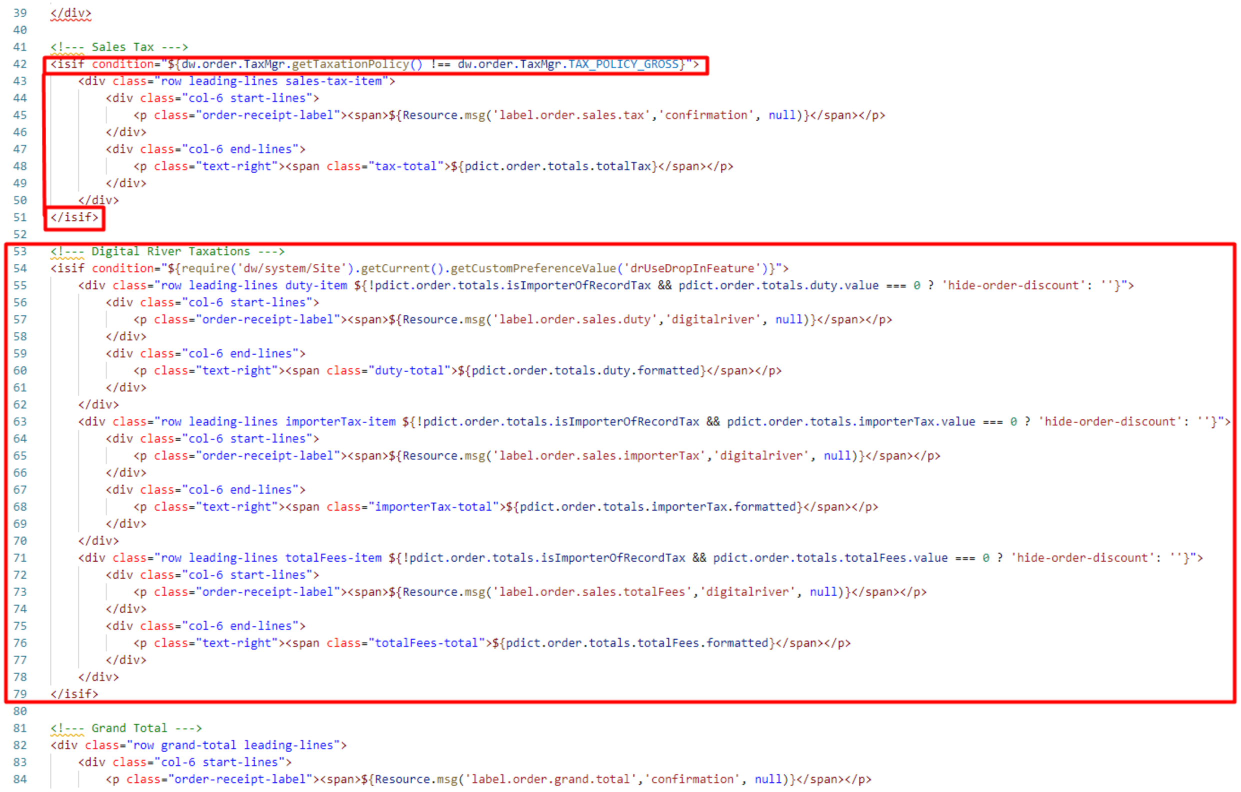
Wrap sales tax section with condition at line 42:

```
<isif condition="${dw.order.TaxMgr.getTaxationPolicy() !==
dw.order.TaxMgr.TAX_POLICY_GROSS}">

… sales tax section

</isif>
```

Add Digital River Taxations section to the line 54

```
!--- Digital River Taxations --->
<isif
condition="${require('dw/system/Site').getCurrent().getCustomPreferenceValue('drUseDropInFea
ture')}">
    <div class="row leading-lines duty-item ${!pdict.order.totals.isImporterOfRecordTax &&
pdict.order.totals.duty.value === 0 ? 'hide-order-discount': ''}">
        <div class="col-6 start-lines">
            <p
class="order-receipt-label"><span>${Resource.msg('label.order.sales.duty','digitalriver',
null)}</span></p>
        </div>
        <div class="col-6 end-lines">
            <p class="text-right"><span
class="duty-total">${pdict.order.totals.duty.formatted}</span></p>
        </div>
    </div>
    <div class="row leading-lines importerTax-item
${!pdict.order.totals.isImporterOfRecordTax && pdict.order.totals.importerTax.value === 0 ?
'hide-order-discount': ''}">
        <div class="col-6 start-lines">
            <p
```

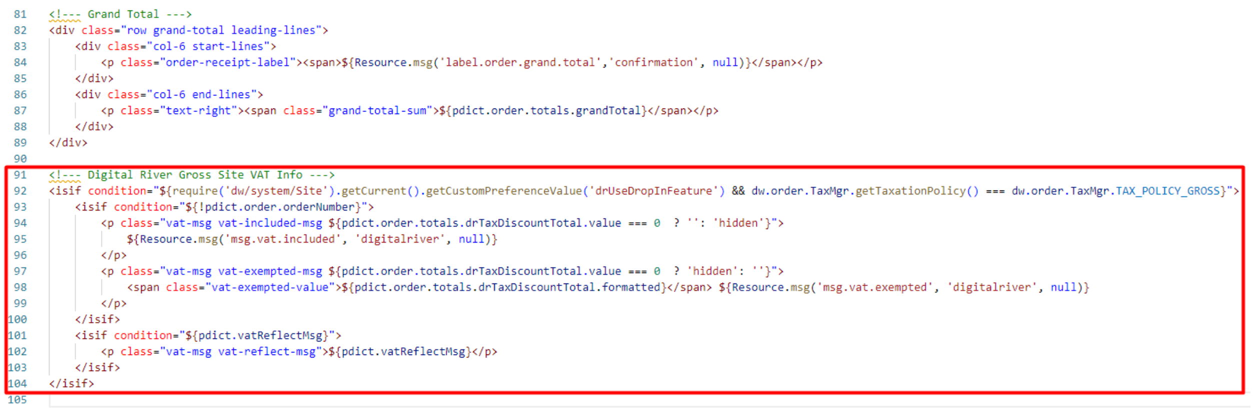
```
class="order-receipt-label"><span>${Resource.msg('label.order.sales.importerTax','digitalriv
er', null)}</span></p>
        </div>
        <div class="col-6 end-lines">
            <p class="text-right"><span
class="importerTax-total">${pdict.order.totals.importerTax.formatted}</span></p>
        </div>
    </div>
    <div class="row leading-lines totalFees-item ${!pdict.order.totals.isImporterOfRecordTax
&& pdict.order.totals.totalFees.value === 0 ? 'hide-order-discount': ''}">
        <div class="col-6 start-lines">
            <p
class="order-receipt-label"><span>${Resource.msg('label.order.sales.totalFees','digitalriver
', null)}</span></p>
        </div>
        <div class="col-6 end-lines">
            <p class="text-right"><span
class="totalFees-total">${pdict.order.totals.totalFees.formatted}</span></p>
        </div>
    </div>
</isif>
```



```
</div>

<!--- Sales Tax --->
<isif condition="${dw.order.TaxMgr.getTaxationPolicy() !== dw.order.TaxMgr.TAX_POLICY_GROSS}">
    <div class="row leading-lines sales-tax-item">
        <div class="col-6 start-lines">
            <p class="order-receipt-label"><span>${Resource.msg('label.order.sales.tax','confirmation', null)}</span></p>
        </div>
        <div class="col-6 end-lines">
            <p class="text-right"><span class="tax-total">${pdict.order.totals.totalTax}</span></p>
        </div>
    </div>
</isif>

<!--- Digital River Taxations --->
<isif condition="${require('dw/system/Site').getCurrent().getCustomPreferenceValue('drUseDropInFeature')}">
    <div class="row leading-lines duty-item ${!pdict.order.totals.isImporterOfRecordTax && pdict.order.totals.duty.value === 0 ? 'hide-order-discount': ''}">
        <div class="col-6 start-lines">
            <p class="order-receipt-label"><span>${Resource.msg('label.order.sales.duty','digitalriver', null)}</span></p>
        </div>
        <div class="col-6 end-lines">
            <p class="text-right"><span class="duty-total">${pdict.order.totals.duty.formatted}</span></p>
        </div>
    </div>
    <div class="row leading-lines importerTax-item ${!pdict.order.totals.isImporterOfRecordTax && pdict.order.totals.importerTax.value === 0 ? 'hide-order-discount': ''}">
        <div class="col-6 start-lines">
            <p class="order-receipt-label"><span>${Resource.msg('label.order.sales.importerTax','digitalriver', null)}</span></p>
        </div>
        <div class="col-6 end-lines">
            <p class="text-right"><span class="importerTax-total">${pdict.order.totals.importerTax.formatted}</span></p>
        </div>
    </div>
    <div class="row leading-lines totalFees-item ${!pdict.order.totals.isImporterOfRecordTax && pdict.order.totals.totalFees.value === 0 ? 'hide-order-discount': ''}">
        <div class="col-6 start-lines">
            <p class="order-receipt-label"><span>${Resource.msg('label.order.sales.totalFees','digitalriver', null)}</span></p>
        </div>
        <div class="col-6 end-lines">
            <p class="text-right"><span class="totalFees-total">${pdict.order.totals.totalFees.formatted}</span></p>
        </div>
    </div>
</isif>

<!--- Grand Total --->
<div class="row grand-total leading-lines">
    <div class="col-6 start-lines">
        <p class="order-receipt-label"><span>${Resource.msg('label.order.grand.total','confirmation', null)}</span></p>
```

Add DigitalRiver gross site VAT Info page to the end of file

```
<!--- Digital River Gross Site VAT Info --->
<isif
condition="${require('dw/system/Site').getCurrent().getCustomPreferenceValue('drUseDropInFea
ture') && dw.order.TaxMgr.getTaxationPolicy() === dw.order.TaxMgr.TAX_POLICY_GROSS}">
    <isif condition="${!pdict.order.orderNumber}">
        <p class="vat-msg vat-included-msg ${pdict.order.totals.drTaxDiscountTotal.value ===
0  ? '': 'hidden'}">
            ${Resource.msg('msg.vat.included', 'digitalriver', null)}
        </p>
        <p class="vat-msg vat-exempted-msg ${pdict.order.totals.drTaxDiscountTotal.value ===
0  ? 'hidden': ''}">
            <span
class="vat-exempted-value">${pdict.order.totals.drTaxDiscountTotal.formatted}</span>
${Resource.msg('msg.vat.exempted', 'digitalriver', null)}
        </p>
    </isif>
    <isif condition="${pdict.vatReflectMsg}">
        <p class="vat-msg vat-reflect-msg">${pdict.vatReflectMsg}</p>
    </isif>
</isif>
```



```
<!--- Grand Total --->
<div class="row grand-total leading-lines">
    <div class="col-6 start-lines">
        <p class="order-receipt-label"><span>${Resource.msg('label.order.grand.total','confirmation', null)}</span></p>
    </div>
    <div class="col-6 end-lines">
        <p class="text-right"><span class="grand-total-sum">${pdict.order.totals.grandTotal}</span></p>
    </div>
</div>

<!--- Digital River Gross Site VAT Info --->
<isif condition="${require('dw/system/Site').getCurrent().getCustomPreferenceValue('drUseDropInFeature') && dw.order.TaxMgr.getTaxationPolicy() === dw.order.TaxMgr.TAX_POLICY_GROSS}">
    <isif condition="${!pdict.order.orderNumber}">
        <p class="vat-msg vat-included-msg ${pdict.order.totals.drTaxDiscountTotal.value === 0  ? '': 'hidden'}">
            ${Resource.msg('msg.vat.included', 'digitalriver', null)}
        </p>
        <p class="vat-msg vat-exempted-msg ${pdict.order.totals.drTaxDiscountTotal.value === 0  ? 'hidden': ''}">
            <span class="vat-exempted-value">${pdict.order.totals.drTaxDiscountTotal.formatted}</span> ${Resource.msg('msg.vat.exempted', 'digitalriver', null)}
        </p>
    </isif>
    <isif condition="${pdict.vatReflectMsg}">
        <p class="vat-msg vat-reflect-msg">${pdict.vatReflectMsg}</p>
    </isif>
</isif>
```

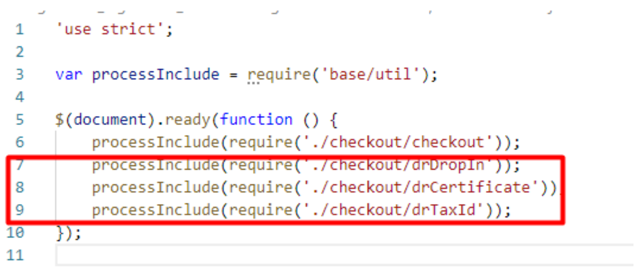
CLIENT SCRIPTS:

***NOTE:** Don't forget to compile client side scripts after implementing changes in source code*

1) Script **cartridge\client\default\js\checkout.js**

Include additional scripts on checkout page

```
processInclude(require('./checkout/drDropIn'));
processInclude(require('./checkout/drCertificate'));
processInclude(require('./checkout/drTaxId'));
```

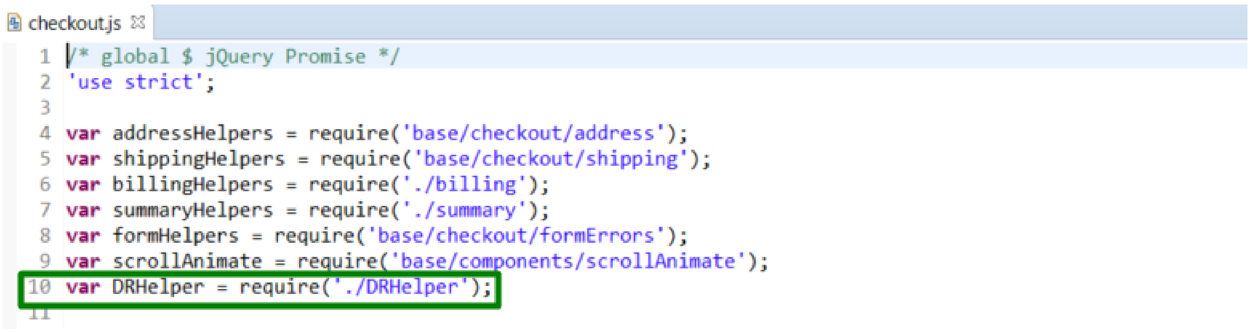


```
'use strict';

var processInclude = require('base/util');

$(document).ready(function () {
    processInclude(require('./checkout/checkout'));
    processInclude(require('./checkout/drDropIn'));
    processInclude(require('./checkout/drCertificate'))
    processInclude(require('./checkout/drTaxId'));
});
```

2) Script **cartridge\client\default\js\checkout\checkout.js**

Add at the beginning of the file the line

```
var DRHelper = require('./DRHelper');
```

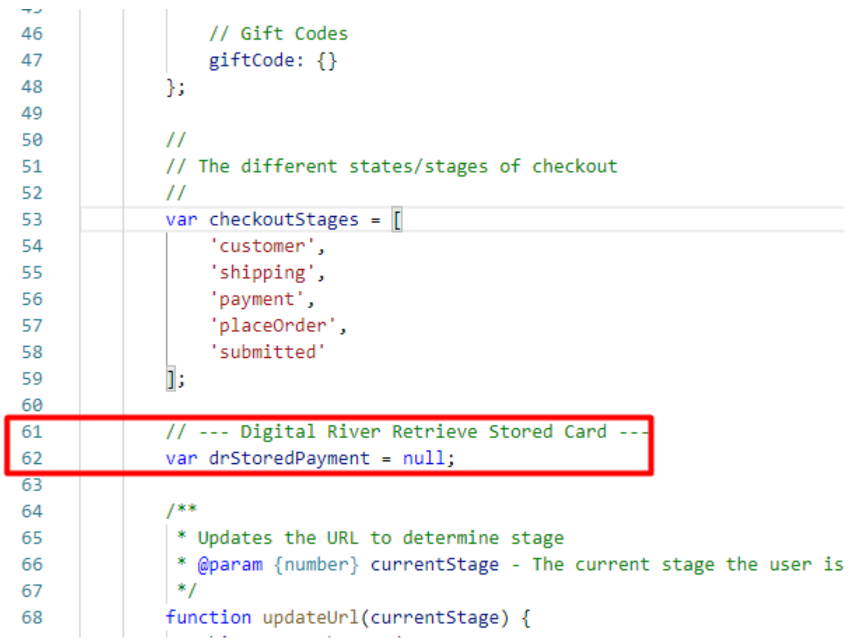


checkout.js

```
/* global $ jQuery Promise */
'use strict';

var addressHelpers = require('base/checkout/address');
var shippingHelpers = require('base/checkout/shipping');
var billingHelpers = require('./billing');
var summaryHelpers = require('./summary');
var formHelpers = require('base/checkout/formErrors');
var scrollAnimate = require('base/components/scrollAnimate');
var DRHelper = require('./DRHelper');
```

Add global variable near the line 61

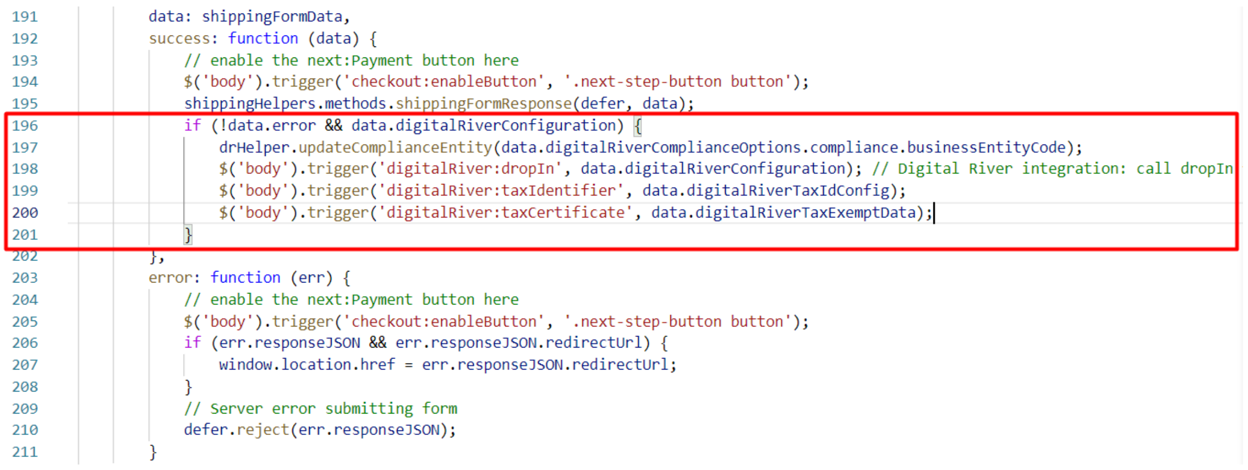
```
// --- Digital River Retrieve Stored Card ---
var drStoredPayment = null;
```



```
46            // Gift Codes
47            giftCode: {}
48        };
49
50        //
51        // The different states/stages of checkout
52        //
53        var checkoutStages = [
54            'customer',
55            'shipping',
56            'payment',
57            'placeOrder',
58            'submitted'
59        ];
60
61        // --- Digital River Retrieve Stored Card ---
62        var drStoredPayment = null;
63
64        /**
65         * Updates the URL to determine stage
66         * @param {number} currentStage - The current stage the user is
67         */
68        function updateUrl(currentStage) {
```

Add following code to shipping submit success handler near the line 196

```
if (!data.error && data.digitalRiverConfiguration) {

drHelper.updateComplianceEntity(data.digitalRiverComplianceOptions.compliance.businessEntityC
ode);
    $('body').trigger('digitalRiver:dropIn', data.digitalRiverConfiguration); // Digital
River integration: call dropIn feature after checkoutCreate
    $('body').trigger('digitalRiver:taxIdentifier', data.digitalRiverTaxIdConfig);
    $('body').trigger('digitalRiver:taxCertificate', data.digitalRiverTaxExemptData);
}
```



```
        data: shippingFormData,
        success: function (data) {
            // enable the next:Payment button here
            $('body').trigger('checkout:enableButton', '.next-step-button button');
            shippingHelpers.methods.shippingFormResponse(defer, data);
            if (!data.error && data.digitalRiverConfiguration) {
                drHelper.updateComplianceEntity(data.digitalRiverComplianceOptions.compliance.businessEntityCode);
                $('body').trigger('digitalRiver:dropIn', data.digitalRiverConfiguration); // Digital River integration: call dropIn
                $('body').trigger('digitalRiver:taxIdentifier', data.digitalRiverTaxIdConfig);
                $('body').trigger('digitalRiver:taxCertificate', data.digitalRiverTaxExemptData);
            }
        },
        error: function (err) {
            // enable the next:Payment button here
            $('body').trigger('checkout:enableButton', '.next-step-button button');
            if (err.responseJSON && err.responseJSON.redirectUrl) {
                window.location.href = err.responseJSON.redirectUrl;
            }
            // Server error submitting form
            defer.reject(err.responseJSON);
        }
```

Add following code near the line 261

```
drStoredPayment = null; // --- Digital River Retrieve Stored Card ---
```

Extend code as following on line 265

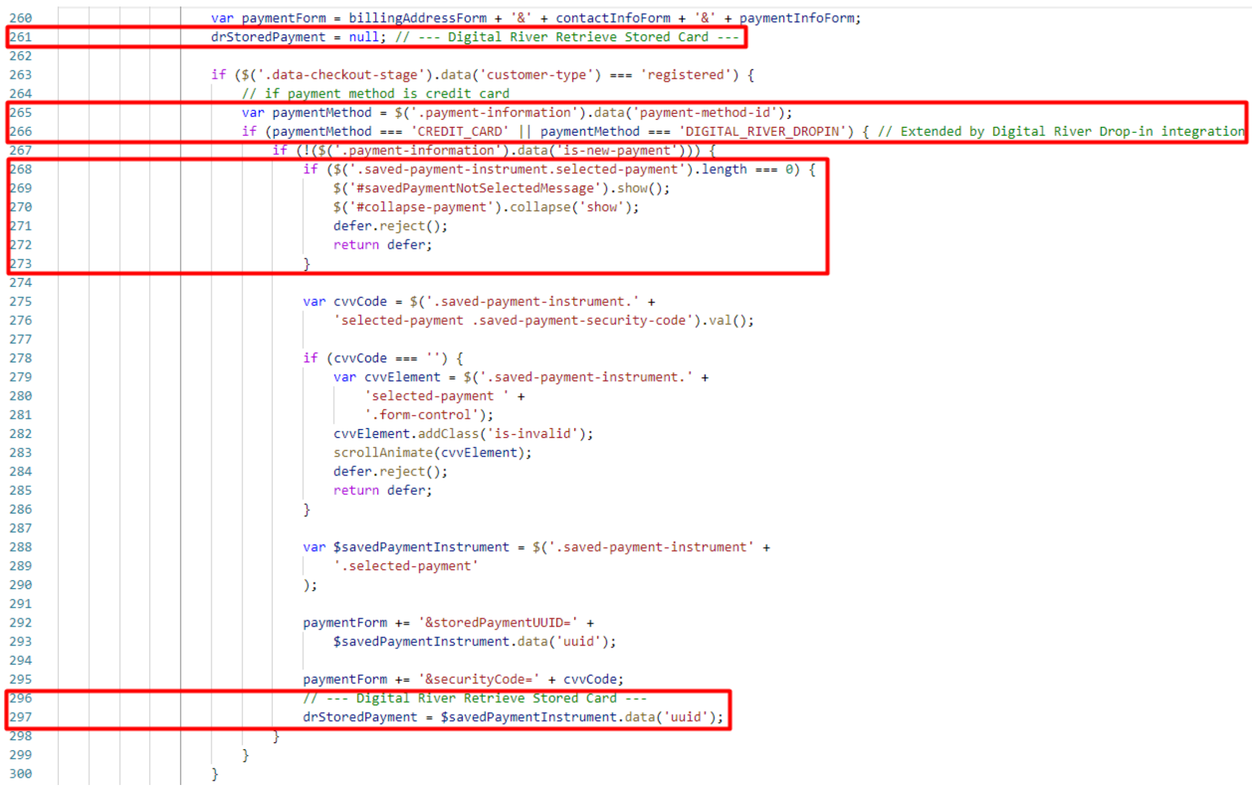
```
var paymentMethod = $('.payment-information').data('payment-method-id');
if (paymentMethod === 'CREDIT_CARD' || paymentMethod === 'DIGITAL_RIVER_DROPIN') { //
Extended by Digital River Drop-in integration
```

Add following code near the line 268

```
if ($('.saved-payment-instrument.selected-payment').length === 0) {
    $('#savedPaymentNotSelectedMessage').show();
    $('#collapse-payment').collapse('show');
    defer.reject();
    return defer;
}
```

Add following code near the line 296

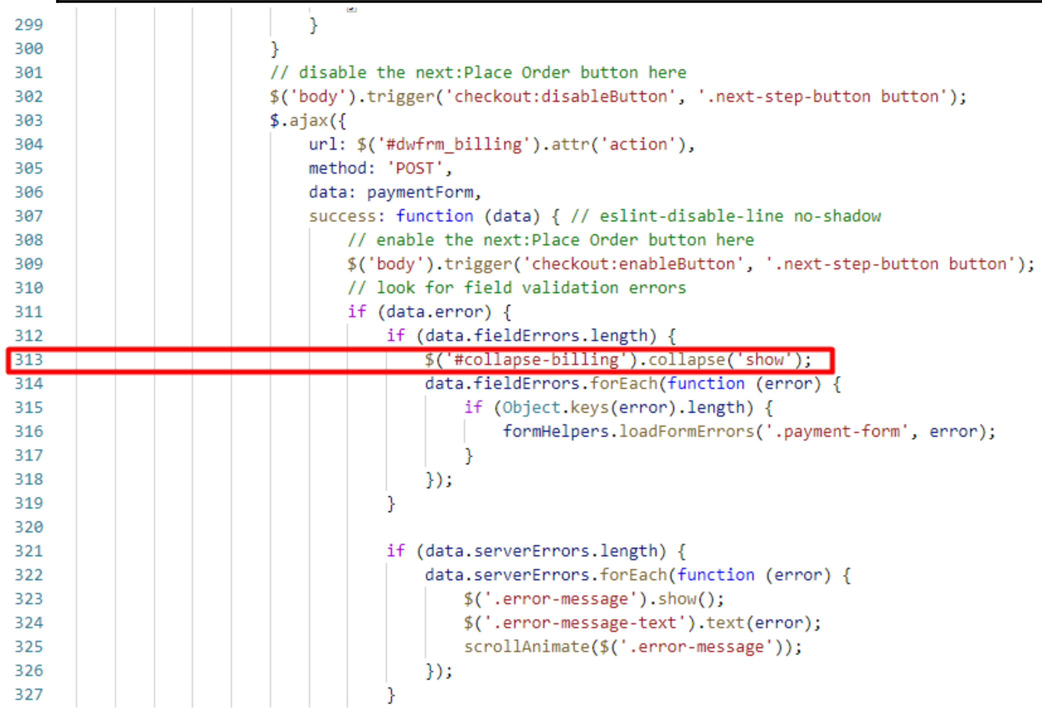
```
// --- Digital River Retrieve Stored Card ---
drStoredPayment = $savedPaymentInstrument.data('uuid');
```



```
260         var paymentForm = billingAddressForm + '&' + contactInfoForm + '&' + paymentInfoForm;
261         drStoredPayment = null; // --- Digital River Retrieve Stored Card ---
262
263         if ($('.data-checkout-stage').data('customer-type') === 'registered') {
264             // if payment method is credit card
265             var paymentMethod = $('.payment-information').data('payment-method-id');
266             if (paymentMethod === 'CREDIT_CARD' || paymentMethod === 'DIGITAL_RIVER_DROPIN') { // Extended by Digital River Drop-in integration
267                 if (!($('.payment-information').data('is-new-payment'))) {
268                     if ($('.saved-payment-instrument.selected-payment').length === 0) {
269                         $('#savedPaymentNotSelectedMessage').show();
270                         $('#collapse-payment').collapse('show');
271                         defer.reject();
272                         return defer;
273                     }
274
275                     var cvvCode = $('.saved-payment-instrument.' +
276                         'selected-payment .saved-payment-security-code').val();
277
278                     if (cvvCode === '') {
279                         var cvvElement = $('.saved-payment-instrument.' +
280                             'selected-payment ' +
281                             '.form-control');
282                         cvvElement.addClass('is-invalid');
283                         scrollAnimate(cvvElement);
284                         defer.reject();
285                         return defer;
286                     }
287
288                     var $savedPaymentInstrument = $('.saved-payment-instrument' +
289                         '.selected-payment'
290                     );
291
292                     paymentForm += '&storedPaymentUUID=' +
293                         $savedPaymentInstrument.data('uuid');
294
295                     paymentForm += '&securityCode=' + cvvCode;
296                     // --- Digital River Retrieve Stored Card ---
297                     drStoredPayment = $savedPaymentInstrument.data('uuid');
298                 }
299             }
300         }
```

Add following code near the line 313

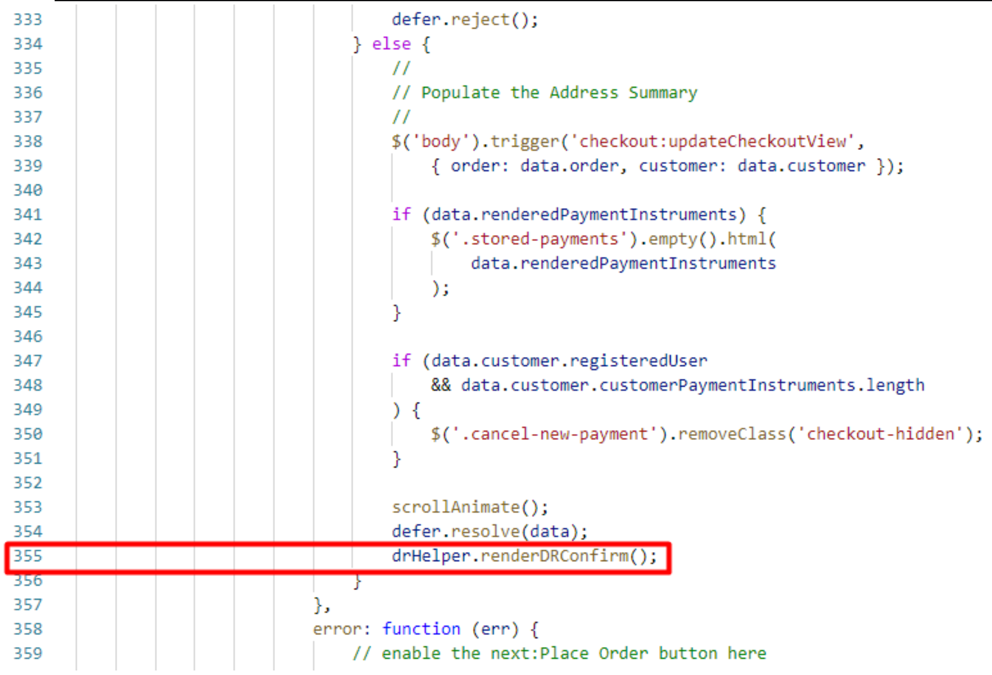
```
$('#collapse-billing').collapse('show');
```



```
299                 }
300             }
301             // disable the next:Place Order button here
302             $('body').trigger('checkout:disableButton', '.next-step-button button');
303             $.ajax({
304                 url: $('#dwfrm_billing').attr('action'),
305                 method: 'POST',
306                 data: paymentForm,
307                 success: function (data) { // eslint-disable-line no-shadow
308                     // enable the next:Place Order button here
309                     $('body').trigger('checkout:enableButton', '.next-step-button button');
310                     // look for field validation errors
311                     if (data.error) {
312                         if (data.fieldErrors.length) {
313                             $('#collapse-billing').collapse('show');
314                             data.fieldErrors.forEach(function (error) {
315                                 if (Object.keys(error).length) {
316                                     formHelpers.loadFormErrors('.payment-form', error);
317                                 }
318                             });
319                         }
320
321                         if (data.serverErrors.length) {
322                             data.serverErrors.forEach(function (error) {
323                                 $('.error-message').show();
324                                 $('.error-message-text').text(error);
325                                 scrollAnimate($('.error-message'));
326                             });
327                         }
```

Extend code as following on line 355

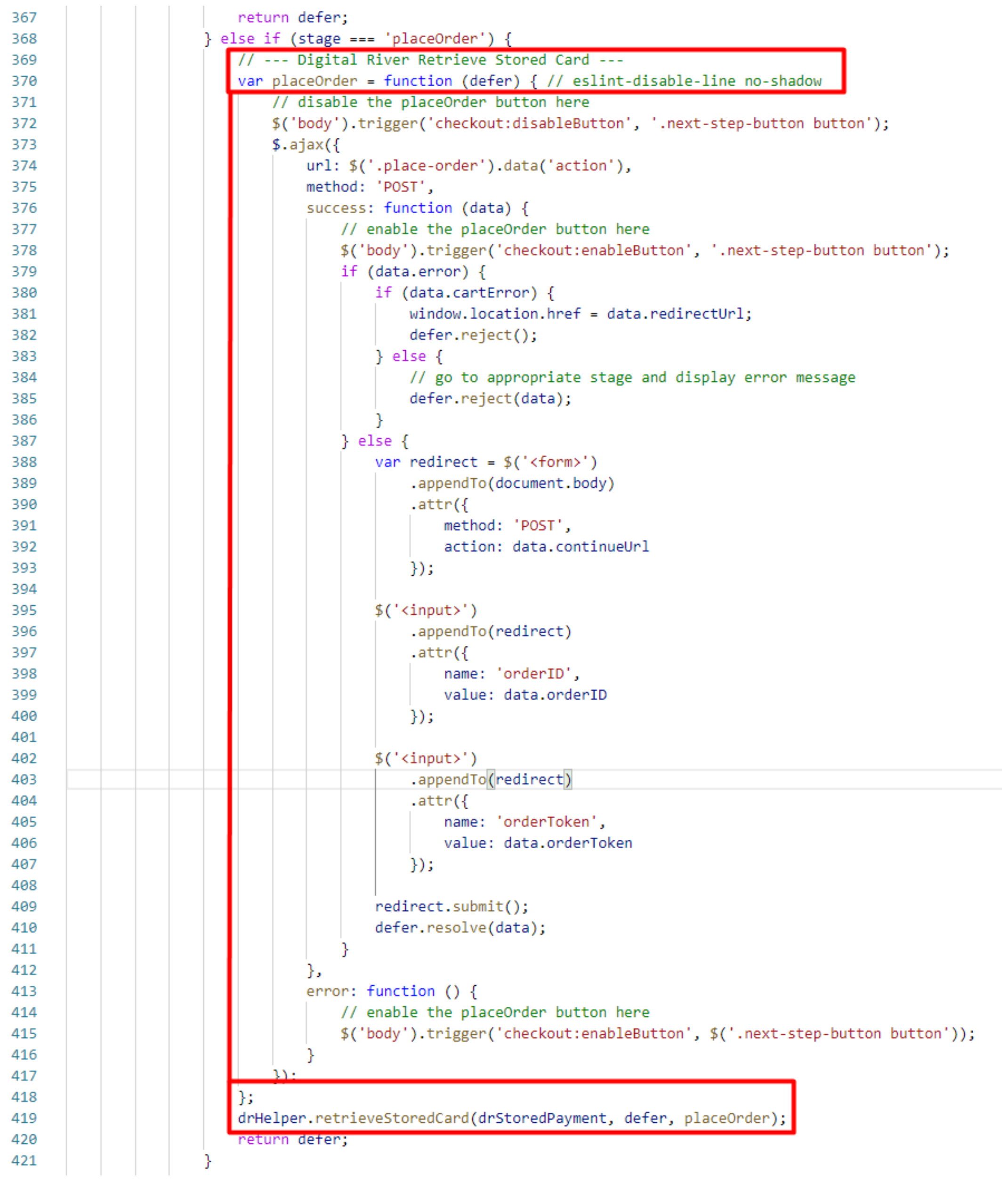
```
DRHelper.renderDRConfirm();
```



```
                    defer.reject();
                } else {
                    //
                    // Populate the Address Summary
                    //
                    $('body').trigger('checkout:updateCheckoutView',
                        { order: data.order, customer: data.customer });

                    if (data.renderedPaymentInstruments) {
                        $('.stored-payments').empty().html(
                            data.renderedPaymentInstruments
                        );
                    }

                    if (data.customer.registeredUser
                        && data.customer.customerPaymentInstruments.length
                    ) {
                        $('.cancel-new-payment').removeClass('checkout-hidden');
                    }

                    scrollAnimate();
                    defer.resolve(data);
                    drHelper.renderDRConfirm();
                }
            },
            error: function (err) {
                // enable the next:Place Order button here
```

Wrap placeOrder stage code section with function at line 369:

```
 // --- Digital River Retrieve Stored Card ---
var placeOrder = function (defer) { // eslint-disable-line no-shadow

… placeOrder stage code section

}
drHelper.retrieveStoredCard(drStoredPayment, defer, placeOrder);
```

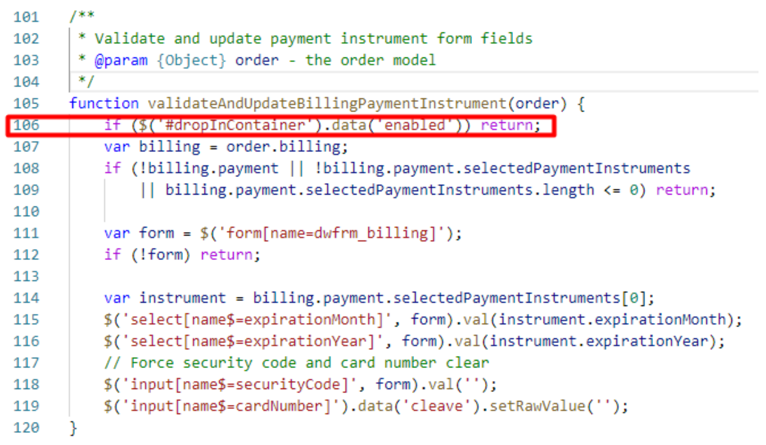


```
    return defer;
} else if (stage === 'placeOrder') {
    // --- Digital River Retrieve Stored Card ---
    var placeOrder = function (defer) { // eslint-disable-line no-shadow
        // disable the placeOrder button here
        $('body').trigger('checkout:disableButton', '.next-step-button button');
        $.ajax({
            url: $('.place-order').data('action'),
            method: 'POST',
            success: function (data) {
                // enable the placeOrder button here
                $('body').trigger('checkout:enableButton', '.next-step-button button');
                if (data.error) {
                    if (data.cartError) {
                        window.location.href = data.redirectUrl;
                        defer.reject();
                    } else {
                        // go to appropriate stage and display error message
                        defer.reject(data);
                    }
                } else {
                    var redirect = $('<form>')
                        .appendTo(document.body)
                        .attr({
                            method: 'POST',
                            action: data.continueUrl
                        });

                    $('<input>')
                        .appendTo(redirect)
                        .attr({
                            name: 'orderID',
                            value: data.orderID
                        });

                    $('<input>')
                        .appendTo(redirect)
                        .attr({
                            name: 'orderToken',
                            value: data.orderToken
                        });

                    redirect.submit();
                    defer.resolve(data);
                }
            },
            error: function () {
                // enable the placeOrder button here
                $('body').trigger('checkout:enableButton', $('.next-step-button button'));
            }
        });
    };
    drHelper.retrieveStoredCard(drStoredPayment, defer, placeOrder);
    return defer;
}
```

3) Script **cartridge\client\default\js\checkout\billing.js**

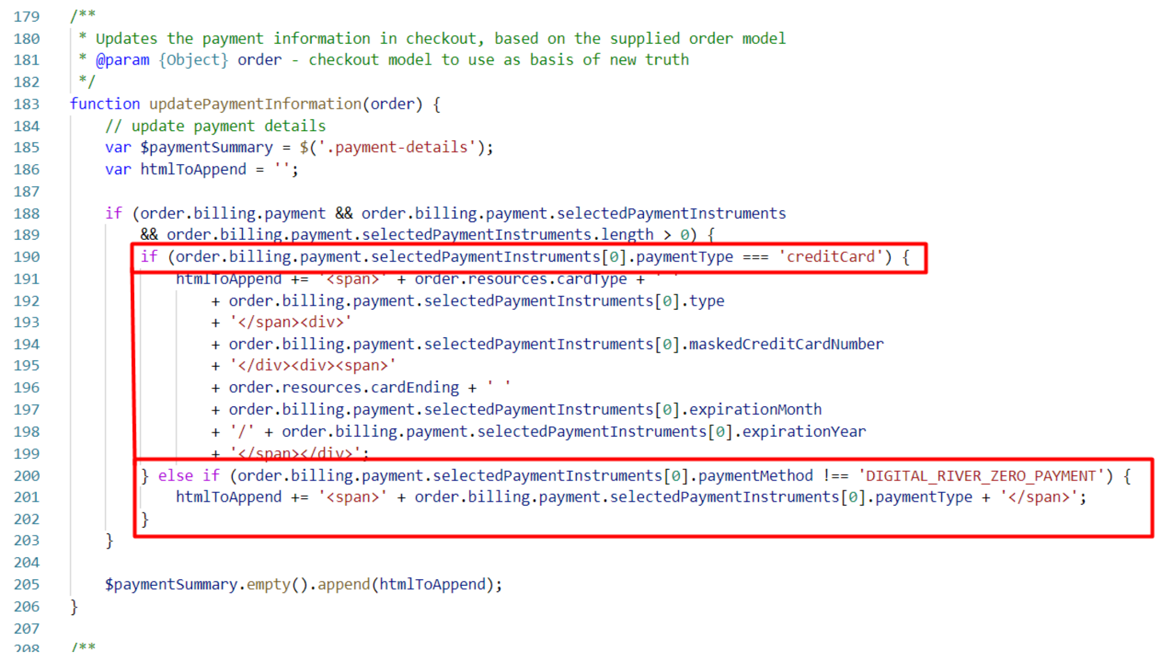
Add a condition to the beginning of the *validateAndUpdateBillingPaymentInstrument* function

```
if ($('#dropInContainer').data('enabled')) return;
```


```
/**
 * Validate and update payment instrument form fields
 * @param {Object} order - the order model
 */
function validateAndUpdateBillingPaymentInstrument(order) {
    if ($('#dropInContainer').data('enabled')) return;
    var billing = order.billing;
    if (!billing.payment || !billing.payment.selectedPaymentInstruments
        || billing.payment.selectedPaymentInstruments.length <= 0) return;

    var form = $('form[name=dwfrm_billing]');
    if (!form) return;

    var instrument = billing.payment.selectedPaymentInstruments[0];
    $('select[name$=expirationMonth]', form).val(instrument.expirationMonth);
    $('select[name$=expirationYear]', form).val(instrument.expirationYear);
    // Force security code and card number clear
    $('input[name$=securityCode]', form).val('');
    $('input[name$=cardNumber]').data('cleave').setRawValue('');
}
```

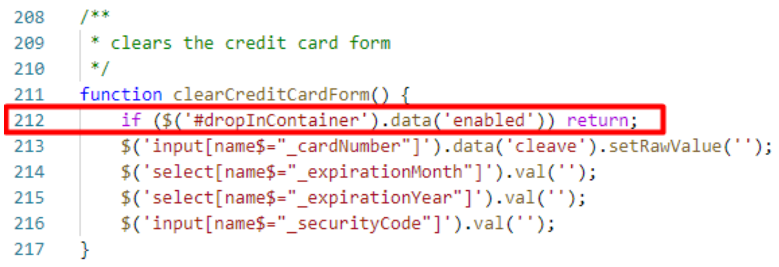
Wrap condition htmlToAppend variable inside *updatePaymentInformation* function

```
if (order.billing.payment.selectedPaymentInstruments[0].paymentType ===
'creditCard') {

… htmlToAppend = …

} else if (order.billing.payment.selectedPaymentInstruments[0].paymentMethod !==
'DIGITAL_RIVER_ZERO_PAYMENT')
 {
    htmlToAppend += '<span>' +
order.billing.payment.selectedPaymentInstruments[0].paymentType + '</span>';
}
```


```
/**
 * Updates the payment information in checkout, based on the supplied order model
 * @param {Object} order - checkout model to use as basis of new truth
 */
function updatePaymentInformation(order) {
    // update payment details
    var $paymentSummary = $('.payment-details');
    var htmlToAppend = '';

    if (order.billing.payment && order.billing.payment.selectedPaymentInstruments
        && order.billing.payment.selectedPaymentInstruments.length > 0) {
        if (order.billing.payment.selectedPaymentInstruments[0].paymentType === 'creditCard') {
            htmlToAppend += '<span>' + order.resources.cardType + ' '
                + order.billing.payment.selectedPaymentInstruments[0].type
                + '</span><div>'
                + order.billing.payment.selectedPaymentInstruments[0].maskedCreditCardNumber
                + '</div><div><span>'
                + order.resources.cardEnding + ' '
                + order.billing.payment.selectedPaymentInstruments[0].expirationMonth
                + '/' + order.billing.payment.selectedPaymentInstruments[0].expirationYear
                + '</span></div>';
        } else if (order.billing.payment.selectedPaymentInstruments[0].paymentMethod !== 'DIGITAL_RIVER_ZERO_PAYMENT') {
            htmlToAppend += '<span>' + order.billing.payment.selectedPaymentInstruments[0].paymentType + '</span>';
        }
    }

    $paymentSummary.empty().append(htmlToAppend);
}

/**
```

Add a condition to the beginning of the *clearCreditCardForm* function

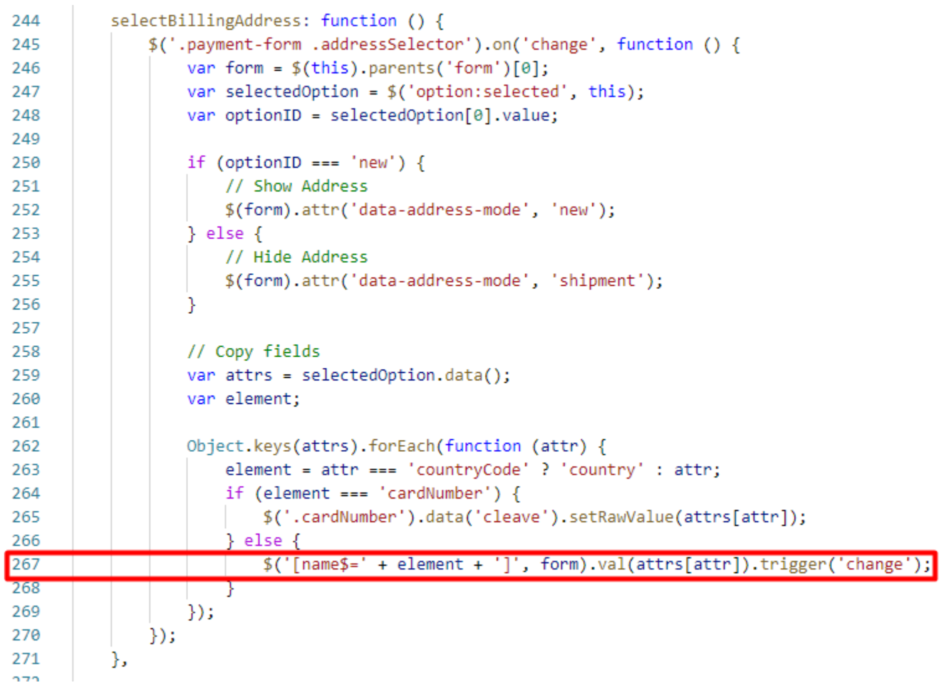
```
if ($('#dropInContainer').data('enabled')) return;
```



```
/**
 * clears the credit card form
 */
function clearCreditCardForm() {
    if ($('#dropInContainer').data('enabled')) return;
    $('input[name$="_cardNumber"]').data('cleave').setRawValue('');
    $('select[name$="_expirationMonth"]').val('');
    $('select[name$="_expirationYear"]').val('');
    $('input[name$="_securityCode"]').val('');
}
```

Extend expression inside *selectBillingAddress* method

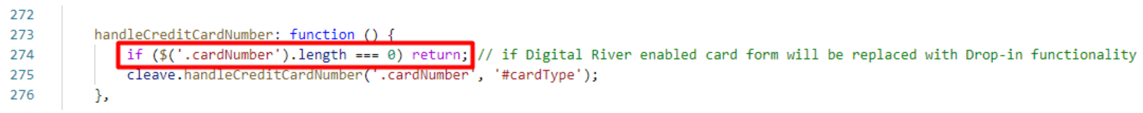
```
$('[name$=' + element + ']', form).val(attrs[attr]).trigger('change');
```



```
    selectBillingAddress: function () {
        $('.payment-form .addressSelector').on('change', function () {
            var form = $(this).parents('form')[0];
            var selectedOption = $('option:selected', this);
            var optionID = selectedOption[0].value;

            if (optionID === 'new') {
                // Show Address
                $(form).attr('data-address-mode', 'new');
            } else {
                // Hide Address
                $(form).attr('data-address-mode', 'shipment');
            }

            // Copy fields
            var attrs = selectedOption.data();
            var element;

            Object.keys(attrs).forEach(function (attr) {
                element = attr === 'countryCode' ? 'country' : attr;
                if (element === 'cardNumber') {
                    $('.cardNumber').data('cleave').setRawValue(attrs[attr]);
                } else {
                    $('[name$=' + element + ']', form).val(attrs[attr]).trigger('change');
                }
            });
        });
    },
```

Add a condition to the beginning of the *handleCreditCardNumber* method

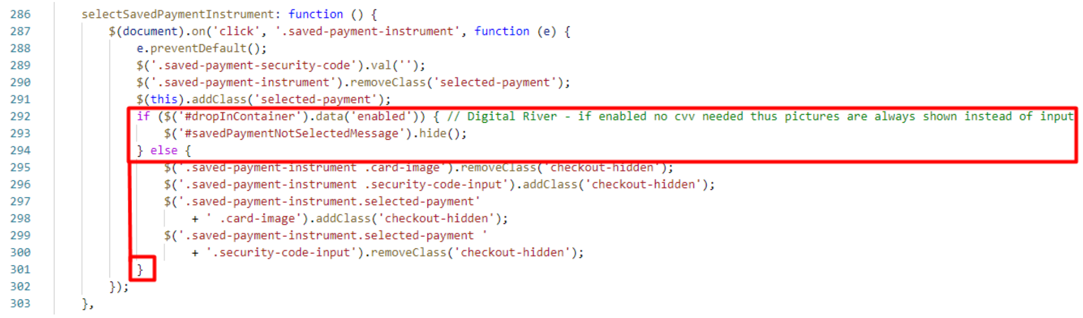
```
if ($('.cardNumber').length === 0) return; // if Digital River enabled card form
will be replaced with Drop-in functionality
```



```

    handleCreditCardNumber: function () {
        if ($('.cardNumber').length === 0) return; // if Digital River enabled card form will be replaced with Drop-in functionality
        cleave.handleCreditCardNumber('.cardNumber', '#cardType');
    },
```

Wrap condition expressions inside *selectSavedPaymentInstrument* method

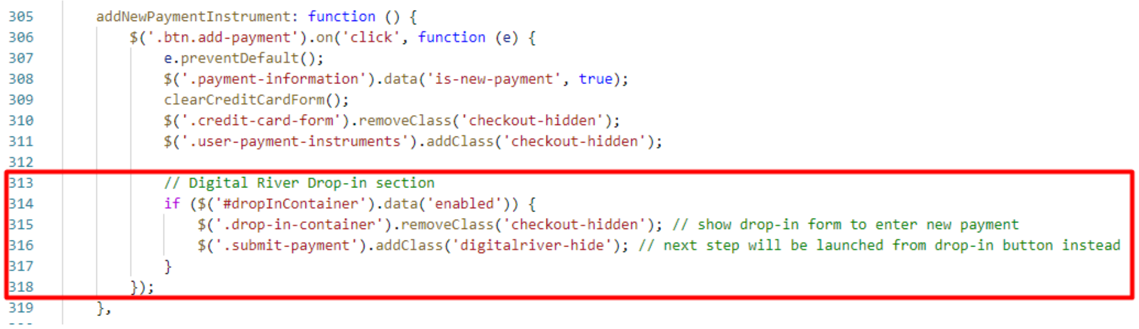
```
if ($('#dropInContainer').data('enabled')) { // Digital River - if enabled no cvv
needed thus pictures are always shown instead of input
    $('#savedPaymentNotSelectedMessage').hide();
} else {

… expressions …

}
```



```
selectSavedPaymentInstrument: function () {
    $(document).on('click', '.saved-payment-instrument', function (e) {
        e.preventDefault();
        $('.saved-payment-security-code').val('');
        $('.saved-payment-instrument').removeClass('selected-payment');
        $(this).addClass('selected-payment');
        if ($('#dropInContainer').data('enabled')) { // Digital River - if enabled no cvv needed thus pictures are always shown instead of input
            $('#savedPaymentNotSelectedMessage').hide();
        } else {
            $('.saved-payment-instrument .card-image').removeClass('checkout-hidden');
            $('.saved-payment-instrument .security-code-input').addClass('checkout-hidden');
            $('.saved-payment-instrument.selected-payment'
                + ' .card-image').addClass('checkout-hidden');
            $('.saved-payment-instrument.selected-payment '
                + '.security-code-input').removeClass('checkout-hidden');
        }
    });
},
```

Add code to the *addNewPaymentInstrument* function

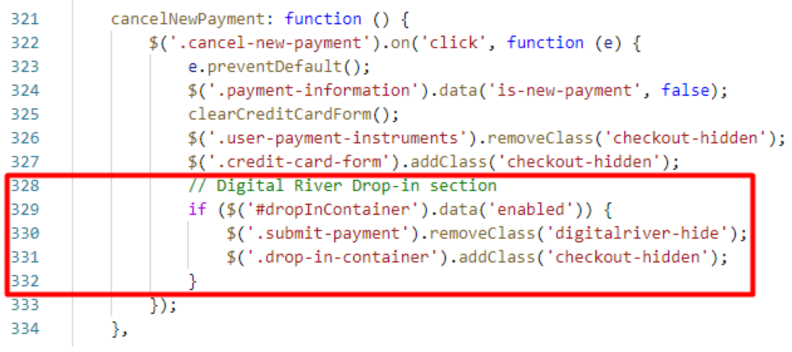
```
 // Digital River Drop-in section
if ($('#dropInContainer').data('enabled')) {
    $('.drop-in-container').removeClass('checkout-hidden'); // show drop-in form to
enter new payment
    $('.submit-payment').addClass('digitalriver-hide'); // next step will be
launched from drop-in button instead
}
```



```
addNewPaymentInstrument: function () {
    $('.btn.add-payment').on('click', function (e) {
        e.preventDefault();
        $('.payment-information').data('is-new-payment', true);
        clearCreditCardForm();
        $('.credit-card-form').removeClass('checkout-hidden');
        $('.user-payment-instruments').addClass('checkout-hidden');

        // Digital River Drop-in section
        if ($('#dropInContainer').data('enabled')) {
            $('.drop-in-container').removeClass('checkout-hidden'); // show drop-in form to enter new payment
            $('.submit-payment').addClass('digitalriver-hide'); // next step will be launched from drop-in button instead
        }
    });
},
```

Add code to the *cancelNewPayment* function

```
// Digital River Drop-in section
if ($('#dropInContainer').data('enabled')) {
    $('.submit-payment').removeClass('digitalriver-hide');
    $('.drop-in-container').addClass('checkout-hidden');
}
```

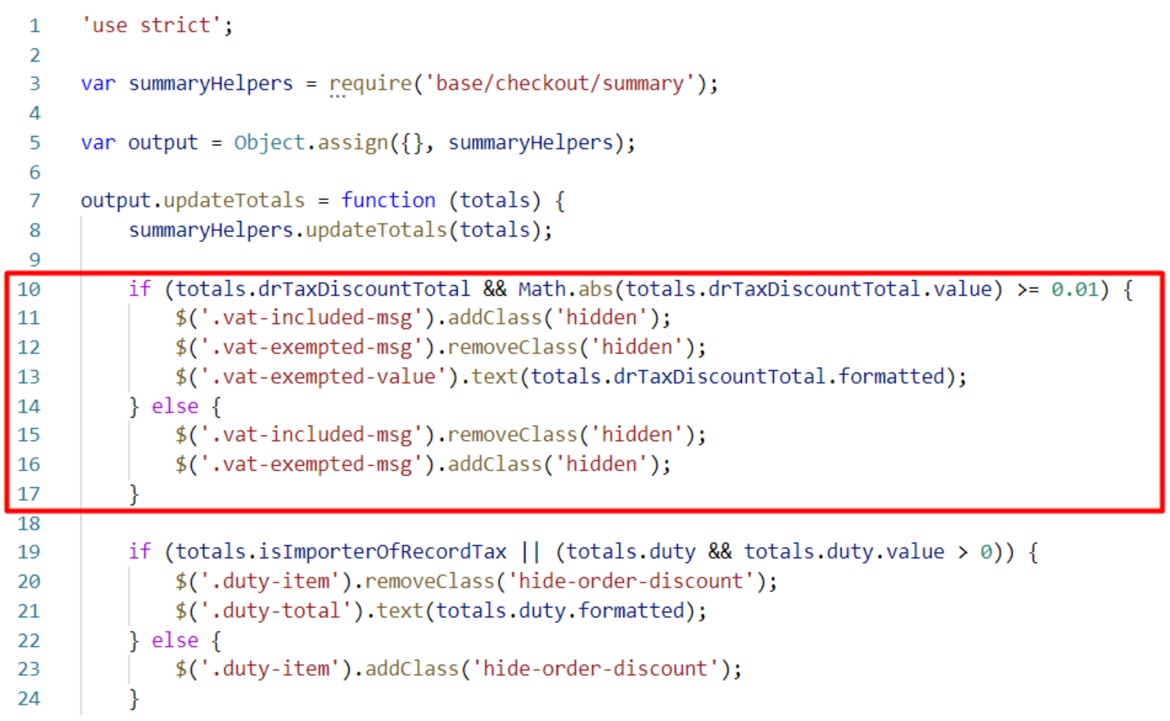


```
cancelNewPayment: function () {
    $('.cancel-new-payment').on('click', function (e) {
        e.preventDefault();
        $('.payment-information').data('is-new-payment', false);
        clearCreditCardForm();
        $('.user-payment-instruments').removeClass('checkout-hidden');
        $('.credit-card-form').addClass('checkout-hidden');
        // Digital River Drop-in section
        if ($('#dropInContainer').data('enabled')) {
            $('.submit-payment').removeClass('digitalriver-hide');
            $('.drop-in-container').addClass('checkout-hidden');
        }
    });
},
```

4) Script **cartridge\client\default\js\checkout\summary.js**

Add rows in line 10:

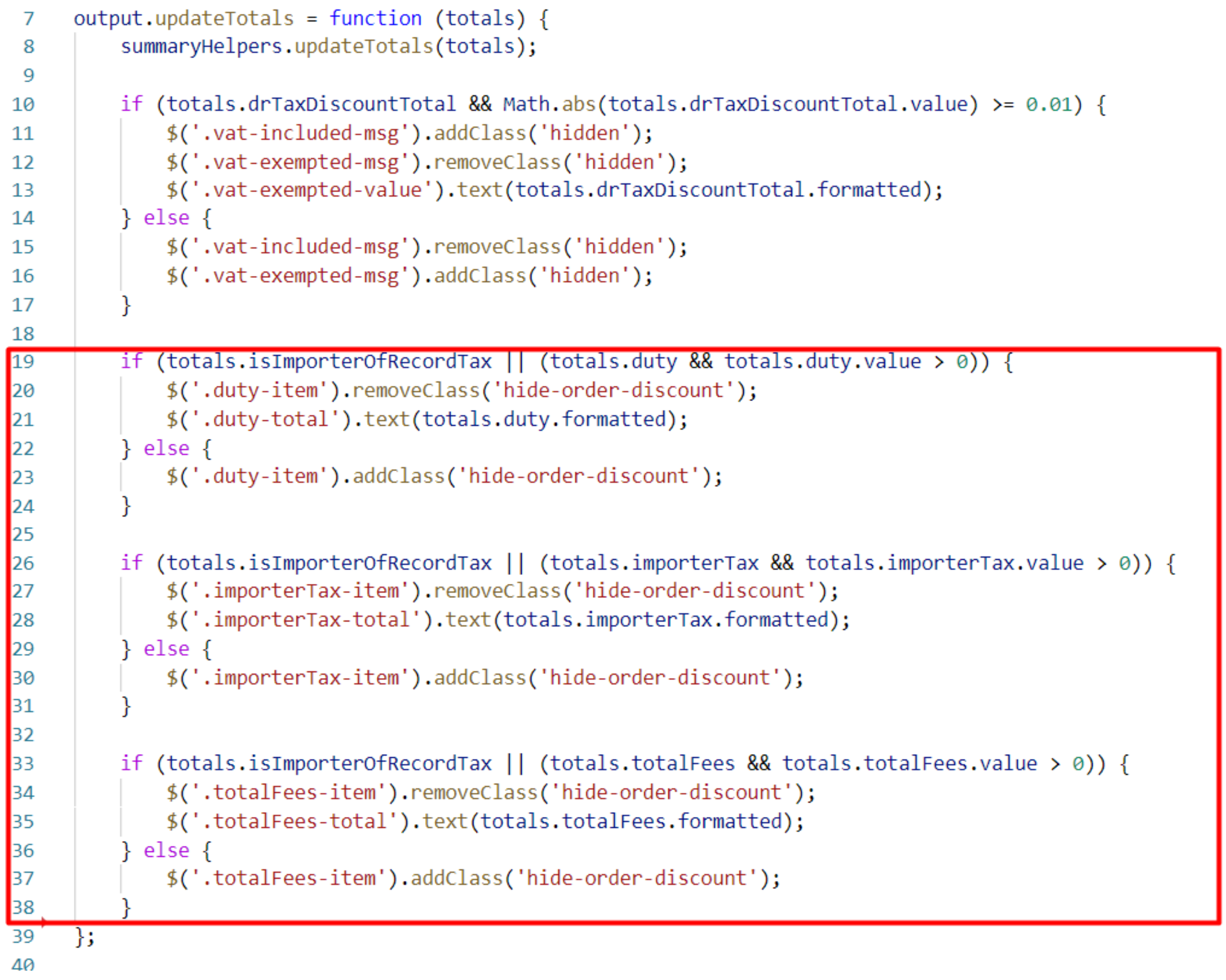
```
if (totals.drTaxDiscountTotal && Math.abs(totals.drTaxDiscountTotal.value) >= 0.01) {
    $('.vat-included-msg').addClass('hidden');
    $('.vat-exempted-msg').removeClass('hidden');
    $('.vat-exempted-value').text(totals.drTaxDiscountTotal.formatted);
} else {
    $('.vat-included-msg').removeClass('hidden');
    $('.vat-exempted-msg').addClass('hidden');
}
```



```
1   'use strict';
2
3   var summaryHelpers = require('base/checkout/summary');
4
5   var output = Object.assign({}, summaryHelpers);
6
7   output.updateTotals = function (totals) {
8       summaryHelpers.updateTotals(totals);
9
10      if (totals.drTaxDiscountTotal && Math.abs(totals.drTaxDiscountTotal.value) >= 0.01) {
11          $('.vat-included-msg').addClass('hidden');
12          $('.vat-exempted-msg').removeClass('hidden');
13          $('.vat-exempted-value').text(totals.drTaxDiscountTotal.formatted);
14      } else {
15          $('.vat-included-msg').removeClass('hidden');
16          $('.vat-exempted-msg').addClass('hidden');
17      }
18
19      if (totals.isImporterOfRecordTax || (totals.duty && totals.duty.value > 0)) {
20          $('.duty-item').removeClass('hide-order-discount');
21          $('.duty-total').text(totals.duty.formatted);
22      } else {
23          $('.duty-item').addClass('hide-order-discount');
24      }
```

Add rows in line 19:

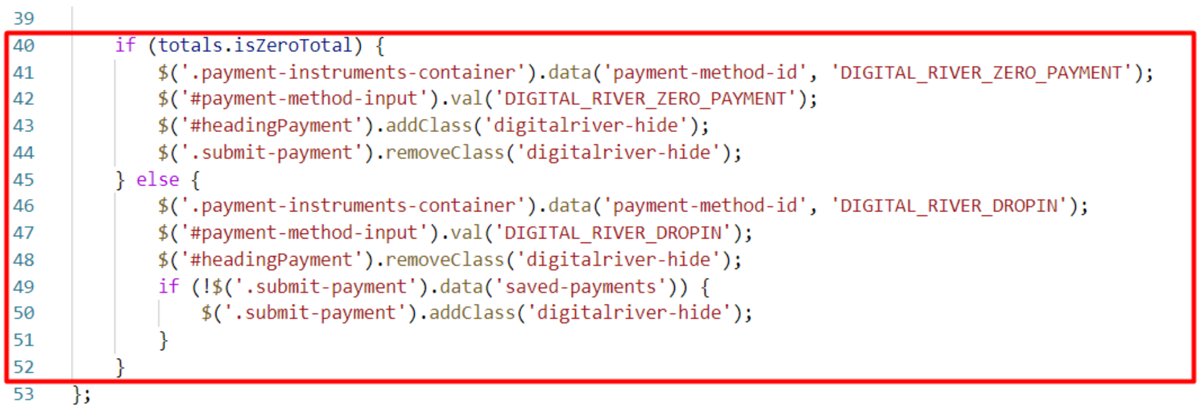
```
if (totals.isImporterOfRecordTax || (totals.duty && totals.duty.value > 0)) {
        $('.duty-item').removeClass('hide-order-discount');
        $('.duty-total').text(totals.duty.formatted);
    } else {
        $('.duty-item').addClass('hide-order-discount');
    }

    if (totals.isImporterOfRecordTax || (totals.importerTax && totals.importerTax.value
> 0)) {
        $('.importerTax-item').removeClass('hide-order-discount');
        $('.importerTax-total').text(totals.importerTax.formatted);
    } else {
        $('.importerTax-item').addClass('hide-order-discount');
    }

    if (totals.isImporterOfRecordTax || (totals.totalFees && totals.totalFees.value >
0)) {
        $('.totalFees-item').removeClass('hide-order-discount');
        $('.totalFees-total').text(totals.totalFees.formatted);
    } else {
        $('.totalFees-item').addClass('hide-order-discount');
    }
```



```
7   output.updateTotals = function (totals) {
8       summaryHelpers.updateTotals(totals);
9
10      if (totals.drTaxDiscountTotal && Math.abs(totals.drTaxDiscountTotal.value) >= 0.01) {
11          $('.vat-included-msg').addClass('hidden');
12          $('.vat-exempted-msg').removeClass('hidden');
13          $('.vat-exempted-value').text(totals.drTaxDiscountTotal.formatted);
14      } else {
15          $('.vat-included-msg').removeClass('hidden');
16          $('.vat-exempted-msg').addClass('hidden');
17      }
18
19      if (totals.isImporterOfRecordTax || (totals.duty && totals.duty.value > 0)) {
20          $('.duty-item').removeClass('hide-order-discount');
21          $('.duty-total').text(totals.duty.formatted);
22      } else {
23          $('.duty-item').addClass('hide-order-discount');
24      }
25
26      if (totals.isImporterOfRecordTax || (totals.importerTax && totals.importerTax.value > 0)) {
27          $('.importerTax-item').removeClass('hide-order-discount');
28          $('.importerTax-total').text(totals.importerTax.formatted);
29      } else {
30          $('.importerTax-item').addClass('hide-order-discount');
31      }
32
33      if (totals.isImporterOfRecordTax || (totals.totalFees && totals.totalFees.value > 0)) {
34          $('.totalFees-item').removeClass('hide-order-discount');
35          $('.totalFees-total').text(totals.totalFees.formatted);
36      } else {
37          $('.totalFees-item').addClass('hide-order-discount');
38      }
39  };
40
```

Add rows in line 40:

```
if (totals.isZeroTotal) {
    $('.payment-instruments-container').data('payment-method-id',
'DIGITAL_RIVER_ZERO_PAYMENT');
    $('#payment-method-input').val('DIGITAL_RIVER_ZERO_PAYMENT');
    $('#headingPayment').addClass('digitalriver-hide');
    $('.submit-payment').removeClass('digitalriver-hide');
} else {
    $('.payment-instruments-container').data('payment-method-id',
'DIGITAL_RIVER_DROPIN');
    $('#payment-method-input').val('DIGITAL_RIVER_DROPIN');
    $('#headingPayment').removeClass('digitalriver-hide');
    if (!$('.submit-payment').data('saved-payments')) {
        $('.submit-payment').addClass('digitalriver-hide');
    }
}
```

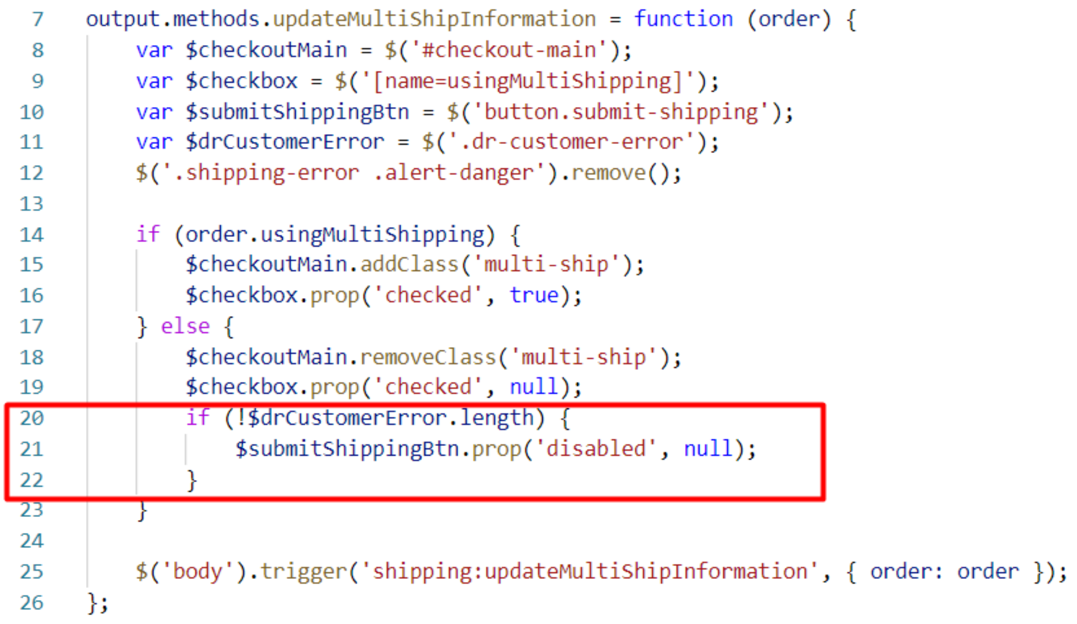


```
    if (totals.isZeroTotal) {
        $('.payment-instruments-container').data('payment-method-id', 'DIGITAL_RIVER_ZERO_PAYMENT');
        $('#payment-method-input').val('DIGITAL_RIVER_ZERO_PAYMENT');
        $('#headingPayment').addClass('digitalriver-hide');
        $('.submit-payment').removeClass('digitalriver-hide');
    } else {
        $('.payment-instruments-container').data('payment-method-id', 'DIGITAL_RIVER_DROPIN');
        $('#payment-method-input').val('DIGITAL_RIVER_DROPIN');
        $('#headingPayment').removeClass('digitalriver-hide');
        if (!$('.submit-payment').data('saved-payments')) {
            $('.submit-payment').addClass('digitalriver-hide');
        }
    }
};
```

5) Script **cartridge\client\default\js\checkput\shipping.js**

Modify the updateMultiShipInformation function with the following condition

```
if (!$drCustomerError.length) {
    $submitShippingBtn.prop('disabled', null);
}
```

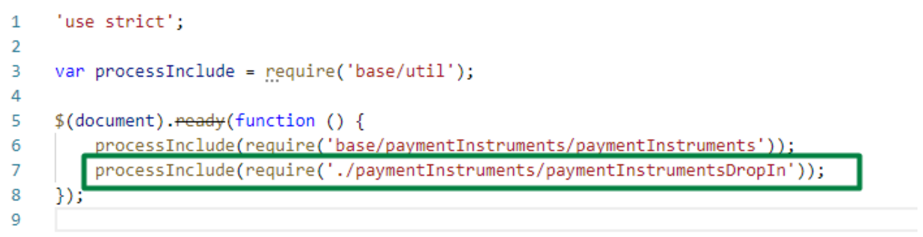


```
output.methods.updateMultiShipInformation = function (order) {
    var $checkoutMain = $('#checkout-main');
    var $checkbox = $('[name=usingMultiShipping]');
    var $submitShippingBtn = $('button.submit-shipping');
    var $drCustomerError = $('.dr-customer-error');
    $('.shipping-error .alert-danger').remove();

    if (order.usingMultiShipping) {
        $checkoutMain.addClass('multi-ship');
        $checkbox.prop('checked', true);
    } else {
        $checkoutMain.removeClass('multi-ship');
        $checkbox.prop('checked', null);
        if (!$drCustomerError.length) {
            $submitShippingBtn.prop('disabled', null);
        }
    }

    $('body').trigger('shipping:updateMultiShipInformation', { order: order });
};
```

6) Script **cartridge\client\default\js\paymentInstruments.js**

Add drop in script to page

```
processInclude(require('./paymentInstruments/paymentInstrumentsDropIn'));
```



```
'use strict';

var processInclude = require('base/util');

$(document).~~ready~~(function () {
    processInclude(require('base/paymentInstruments/paymentInstruments'));
    processInclude(require('./paymentInstruments/paymentInstrumentsDropIn'));
});
```

`

External Interfaces

The cartridge uses the Digital River API endpoints. You can find API service documentation on the Digital River website

There is a service sharing 1 profile and 1 credential.

- *DigitalRiver.http.service*

Also this cartridge uses external script to handle client payments.
IMPORTANT: DIGITAL_RIVER_DROPIN payment method and payment processor in fact represent not one but multiple payment types such as credit card, PayPal, Wire Transfer, etc. Selection of payment types is provided by Drop-in widget and handled by client side/backend scripts. This cartridge is designed mainly to handle creditCard payment type, though any other payment type provided by Drop-in widget will also be successfully processed. If you want to add any business logic to specific payment type handling or provide shoppers with better customer experience you can extend drop-in data handlers that are mentioned below:

int_digitalriver\cartridge\scripts\digitalRiver\dropinHelper.js

switch(source.type) in function ***saveDropInPaymentToWallet*** - to save any specific payment data from drop-in to the customer wallet which will be used in further checkouts.
switch(paymentType) in function ***getPaymentInformationFromPaymentInstrument*** - to handle any payment type specific data which was saved in customer profile
switch(source.type) in function ***getPaymentInformationFromDropIn*** - to handle any payment type specific data which was provided by drop-in

int_digitalriver_sfra\cartridge\models\payment.js

function ***extendDigitalRiverInfo*** - extract all information you want to be available in templates or be provided to the client side as json.

int_digitalriver_sfra\cartridge\client\default\js\checkout\billing.js

switch(selectedPaymentInstrument.paymentType) in function ***updatePaymentInformation*** - add payment information to html. Make sure you have necessary data provided by Payment model

Firewall Requirements

No adjustments to the SFCC Firewall settings are needed.

3. Implementing the Tax Identifiers for Gross taxation: workaround for promotions issue

Description

Applying a tax identifier during a checkout may result in a tax exemption. In this case, the gross price should be reduced by the value of the tax (referenced as Tax Discount) to synchronize the position price on both platforms. On the SFCC platform for the gross taxation sites, the following approach is implemented to calculate the price Net price = Gross price - Tax. When the tax changes, the Net price is recalculated, but the Gross price remains constant. This led to the incorrect promotional prices calculation.

This section describes the approach used to account for this issue and correctly exempt tax when applicable while not modifying the value of the applied promotions.

To synchronize the prices on both platforms, the following approach is used:
total tax discount is handled by the cartridge code, and stored as a non-basket item. Whenever we need to display the total to the customer we subtract it from the order total.

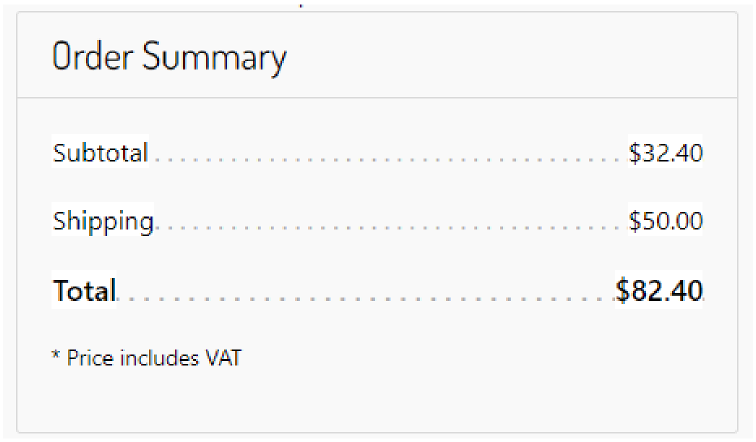
On the last step of the checkout, just before the order is created from the basket, absolute discount values are calculated, then all price adjustments that were added by the SfCC, Promotion Engine are removed and replaced with one custom priceAdjustment with type amountOff on the both Order and Line Item Level. Then line item level price adjustment with the tax discount amount is added to each line item.

General steps:

1. Replace product line item price adjustments and shipment line item price adjustments with one amountOff adjustment. Create an amountOff price adjustment and set the amount to the sum of all discounts that were applied to the item during checkout. Then remove all system price adjustments generated by the Promotion engine during the checkout (store the info in the order notes see s.3.).
2. Create a custom tax discount adjustment for each product line item and for shipping line items, with the amount of the line item tax discount.
3. Add to the order notes descriptions of all removed discounts. The description includes the product ID, the name of the promotion and the amount of the promotion discount.
4. Keep the Tax Discount as a non-basket item. Subtract it from the SfCC basket total to display the correct value to the customer, and send the correct value to the DropIn.

Storefront changes for the gross site taxation

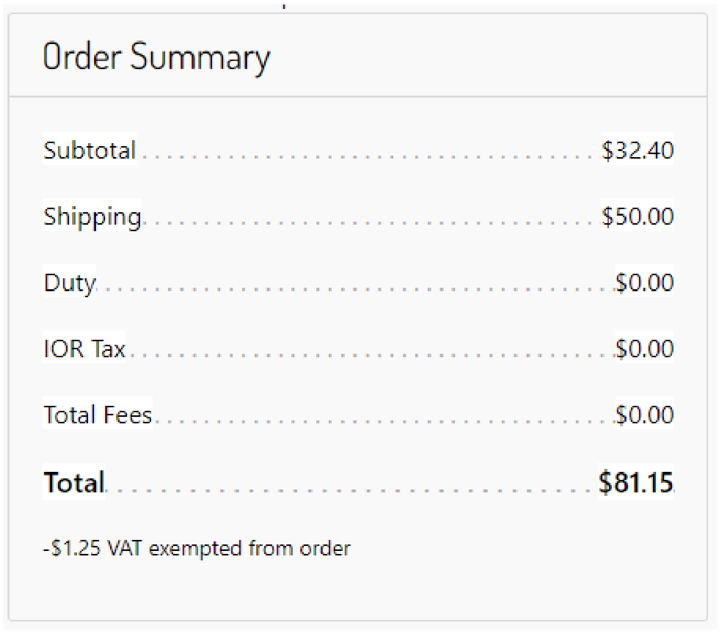
Removed the "Total Tax" line from the summary. Instead, the following message was added underneath the total price - "Price includes VAT".


Order Summary

Subtotal	$32.40
Shipping	$50.00
Total	**$82.40**

* Price includes VAT

When Tax Identifier applied causing the Tax Discount, the “Price includes VAT” line is removed and replaced with the “X.XX VAT exempted from order”.


Order Summary

Subtotal	$32.40
Shipping	$50.00
Duty	$0.00
IOR Tax	$0.00
Total Fees	$0.00
Total	**$81.15**

-$1.25 VAT exempted from order

Please note that Shipping Discount on the picture above reflects the SfCC Shipping Promotion applied, not the Shipping Tax Discount.

NOTE: Shipping discount line
This is a standard SFRA message for displaying any types of price adjustments applied to shipping line items. In the case when both DR Tax Discount and SfCC Shipping promotion are applied their amount will be summed up and displayed as a single line.

We agree not to change the message and the way the amount is calculated because it will take more effort which seems to be redundant due to future customization of the Order Summary on real projects.

However the requirements are not to display the Shipping Discount if there are no Shipping Promotions and the Tax Identifier is not applied.

It could appear on the Order Summary when it reflects the SfCC Promotion applied to the Shipping not the Tax Discount.

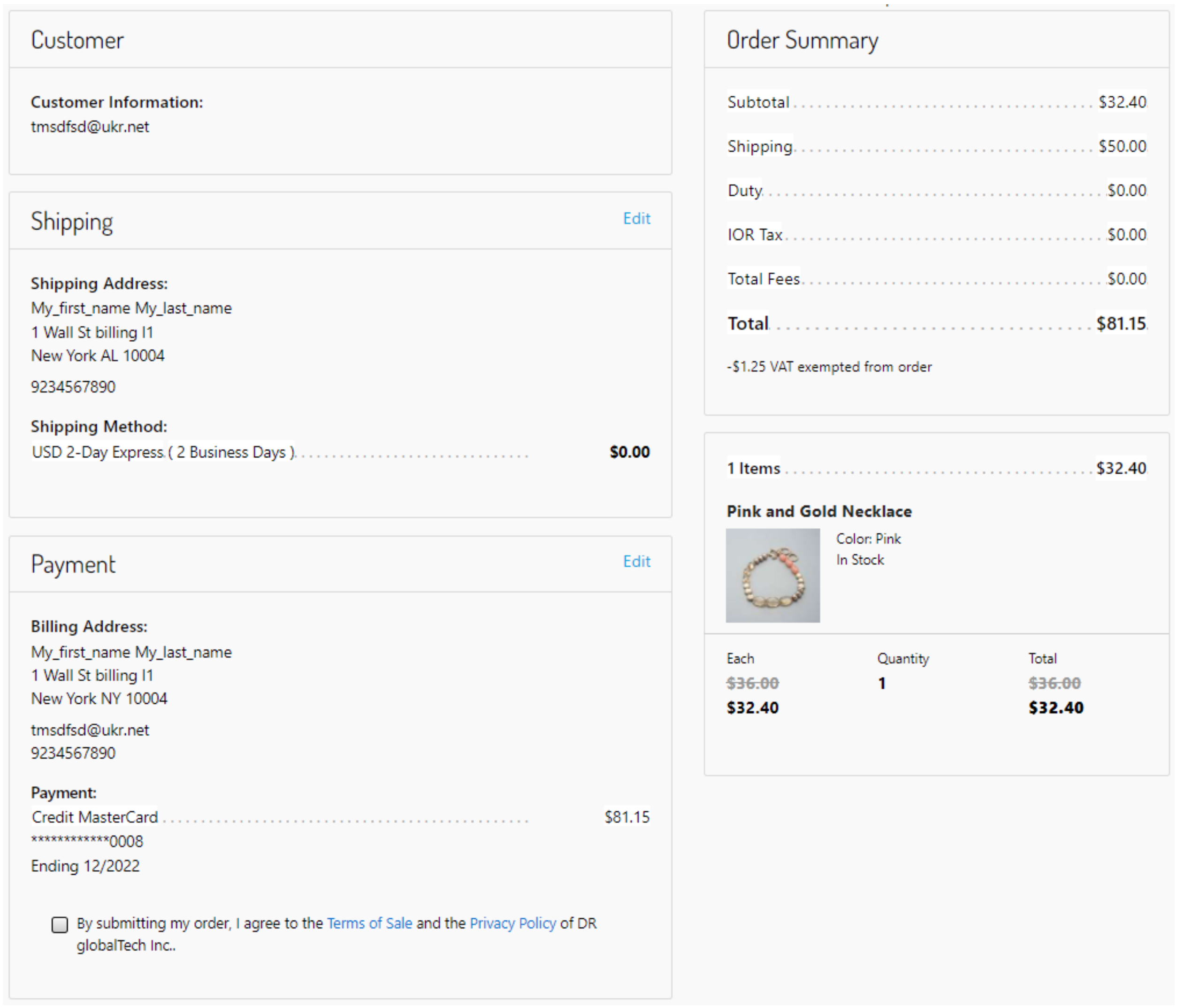
Basket to Order transition UI changes

The following screens displays the visual difference between basket and order after the promotions were replaced and tax discounts added.

“Order review” page

At this stage the Tax Discount isn’t applied to the basket and the SfCC system Line Item level promotions are in place.

Sum of Subtotal and Shipping is greater than the Total, this is explained by the text beneath the Total.



Customer

Customer Information:
tmsdfsd@ukr.net

Shipping Edit

Shipping Address:
My_first_name My_last_name
1 Wall St billing l1
New York AL 10004
9234567890

Shipping Method:
USD 2-Day Express (2 Business Days) **$0.00**

Payment Edit

Billing Address:
My_first_name My_last_name
1 Wall St billing l1
New York NY 10004
tmsdfsd@ukr.net
9234567890

Payment:
Credit MasterCard $81.15
************0008
Ending 12/2022

☐ By submitting my order, I agree to the Terms of Sale and the Privacy Policy of DR globalTech Inc..

Order Summary

Subtotal	$32.40
Shipping	$50.00
Duty	$0.00
IOR Tax	$0.00
Total Fees	$0.00
Total	**$81.15**

-$1.25 VAT exempted from order

1 Items $32.40

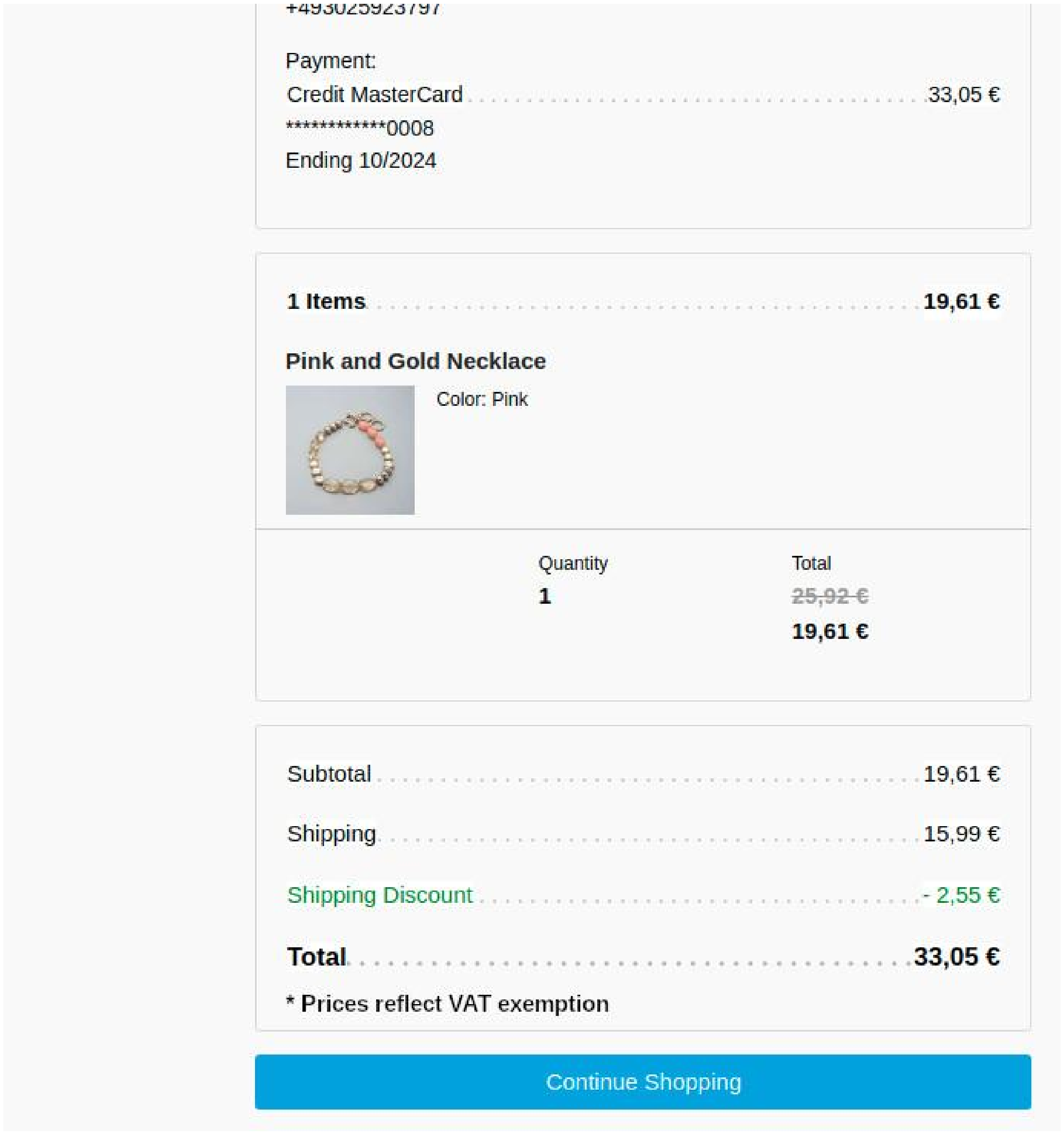
Pink and Gold Necklace
Color: Pink
In Stock

Each	Quantity	Total
~~$36.00~~ **$32.40**	**1**	~~$36.00~~ **$32.40**

“Thank you” page

At this stage Line item tax discount is applied to the product causing the decrease of price. Also line Shipping Discount appears. This is SFRA standard to reflect any type of price adjustment applied to the Shipping Line item.

Sum of Subtotal, Shipping and Discount is equal to the Total which is expected. To notify customers about the reason the prices were changed add the line “Prices reflect VAT exemption” below order total.


+493025923797

Payment:
Credit MasterCard .... 33,05 €
************0008
Ending 10/2024

1 Items .... **19,61 €**

Pink and Gold Necklace

Color: Pink

Quantity	Total
1	~~25,92 €~~ **19,61 €**

Subtotal .... 19,61 €

Shipping .... 15,99 €

Shipping Discount .... - 2,55 €

Total .... **33,05 €**

*** Prices reflect VAT exemption**

Business manager updates

Configuration page

Add the following Site Preference to the Digital River section

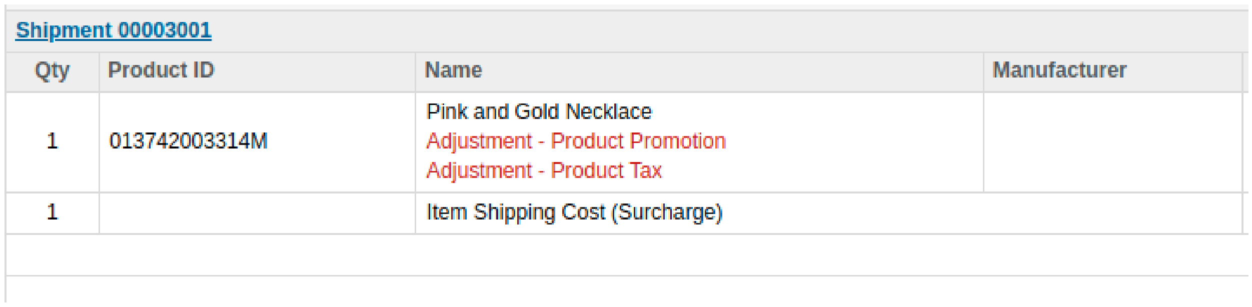
id	Label	type	default

drReplacePromotions	Replace all promotions with one fixed amount (Important: read the guide before enabling)	Boolean (yes/no)	No

Order display page

When applicable, anorder note “Applied promotions” is added with the data of the promotions applied by the Promotion engine.

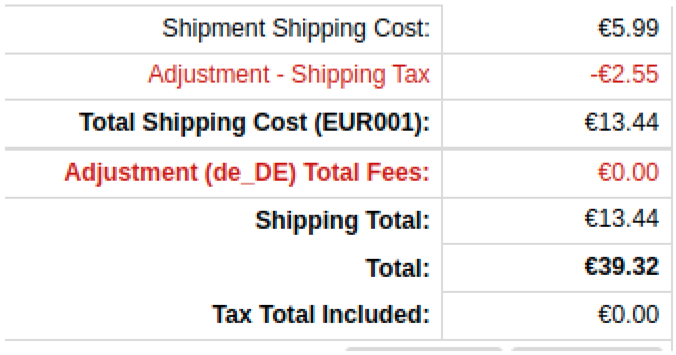
When applicable, the General tab displaying the order details will display the amount of replaced promotions as a “Adjustment - Product Promotions”
Additionally, the amount of the Tax Discount applied is as a “Adjustment - Product Tax”



Shipment 00003001

Qty	Product ID	Name	Manufacturer
1	013742003314M	Pink and Gold Necklace Adjustment - Product Promotion Adjustment - Product Tax	
1		Item Shipping Cost (Surcharge)	

Finally, theShipping Tax Discount is displayed as ““Adjustment - Shipping Tax”



Shipment Shipping Cost:	€5.99
Adjustment - Shipping Tax	-€2.55
Total Shipping Cost (EUR001):	€13.44
Adjustment (de_DE) Total Fees:	€0.00
Shipping Total:	€13.44
Total:	**€39.32**
Tax Total Included:	€0.00

Total Tax Included value should be zero when tax identifier applied.

4. Customer Credit extension

Digital River's customer credit feature can be used to enable a number of different storefront functionalities such as gift certificates and rewards points.
SFCC does not natively support gift cards or store credit functionality. Instead this functionality is provided by 3rd party integrations. Rather than directly integrating with any of these 3rd parties, the Digital River cartridge provides several functions which make the necessary calls to Digital River to support the customer credit. It is the responsibility of the SI at the time of implementation to code the necessary changes to call the functions provided by the DR cartridge to connect to the 3rd party integration.

This section describes how the cartridge supports this feature and the steps required by the implementer to integrate the customer credit feature to the specific use case.

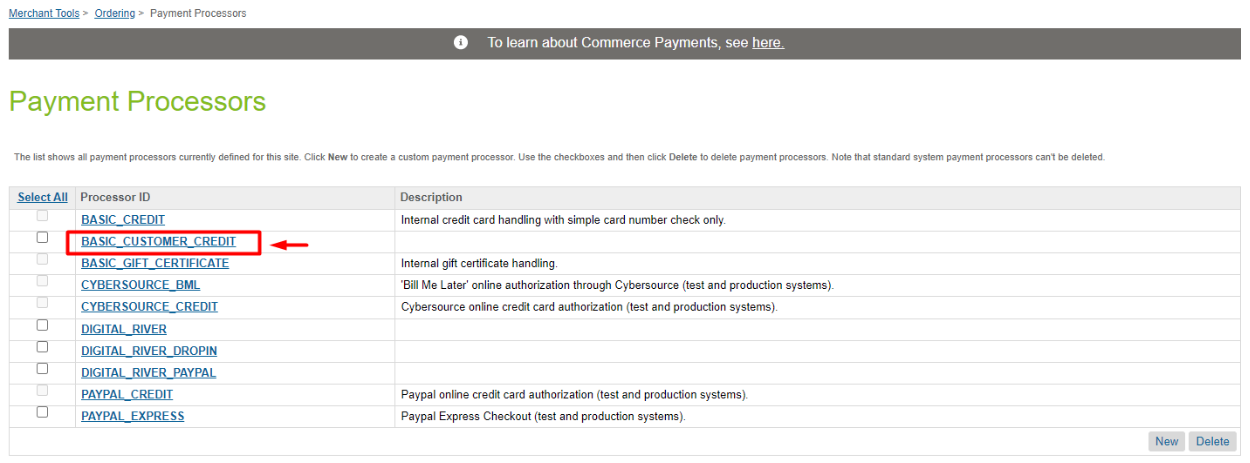
The back-end implementation of the customer credit functionality is in the int_digitalriver_customercredit cartridge.

Business Manager settings

The example implementation of the customer credit functionality that is integrated into the cartridge uses the GIFT_CERTIFICATE payment method and the BASIC_CUSTOMER_CREDIT payment processor on the SFCC side.
If using the cartridge as shown in this example, make sure that the specified payment method and payment processor are set for the customer's site.
If using the cartridge for integration with other use cases, perform the appropriate method/processor configuration to meet the use case. This documentation indicates the steps required to complete the integration of the customer credit feature with the example payment method.

Example Payment Processor and Payment Method

In the Business Manager go to Merchant Tools > Ordering > Payment Processors and make sure the BASIC_CUSTOMER_CREDIT payment processor is present. If using a different payment processor, complete this step according to the use case.



Merchant Tools > Ordering > Payment Processors

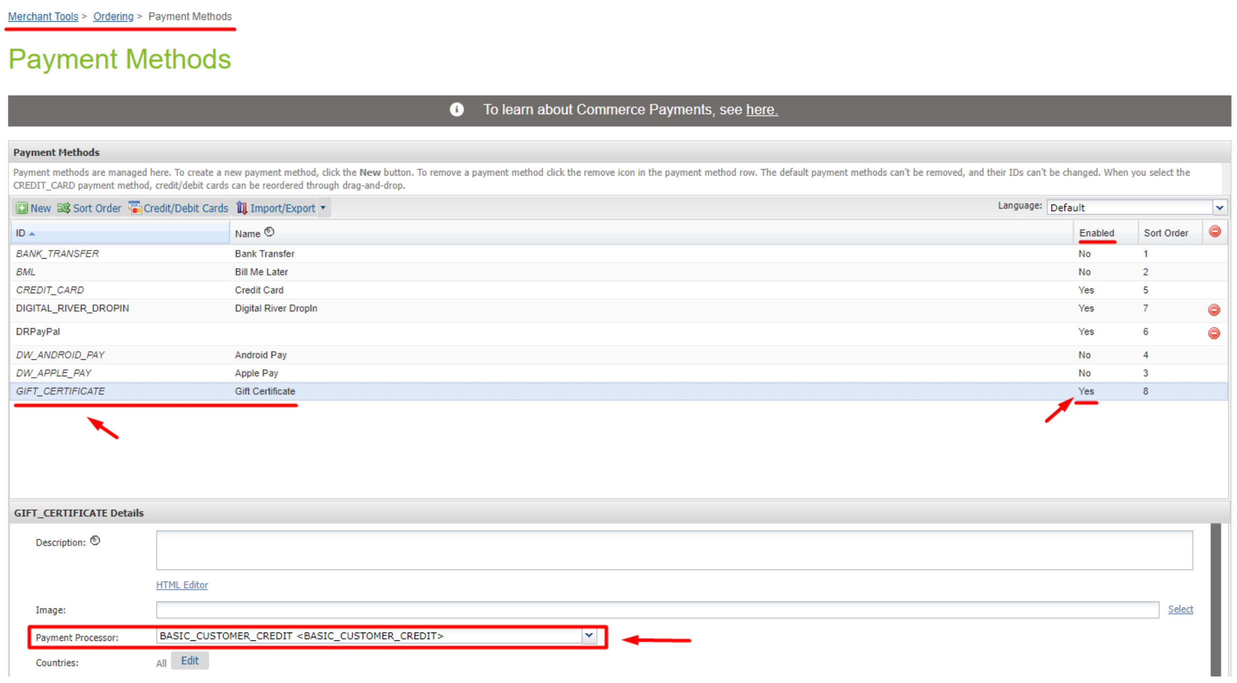
To learn about Commerce Payments, see here.

Payment Processors

The list shows all payment processors currently defined for this site. Click **New** to create a custom payment processor. Use the checkboxes and then click **Delete** to delete payment processors. Note that standard system payment processors can't be deleted.

Select All	Processor ID	Description
☐	BASIC_CREDIT	Internal credit card handling with simple card number check only.
☐	BASIC_CUSTOMER_CREDIT	
☐	BASIC_GIFT_CERTIFICATE	Internal gift certificate handling.
☐	CYBERSOURCE_BML	'Bill Me Later' online authorization through Cybersource (test and production systems).
☐	CYBERSOURCE_CREDIT	Cybersource online credit card authorization (test and production systems).
☐	DIGITAL_RIVER	
☐	DIGITAL_RIVER_DROPIN	
☐	DIGITAL_RIVER_PAYPAL	
☐	PAYPAL_CREDIT	Paypal online credit card authorization (test and production systems).
☐	PAYPAL_EXPRESS	Paypal Express Checkout (test and production systems).

In the Business Manager go to Merchant Tools > Ordering > Payment Methods and ensure the GIFT_CERTIFICATE payment method is present. and turn on. If using a different payment method, complete this step according to the use case. The BASIC_CUSTOMER_CREDIT payment processor is selected for this payment method. Select the payment processor from the previous step.


Merchant Tools > Ordering > Payment Methods

Payment Methods

To learn about Commerce Payments, see here.

Payment Methods

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

New | Sort Order | Credit/Debit Cards | Import/Export

Language: Default

ID	Name	Enabled	Sort Order
BANK_TRANSFER	Bank Transfer	No	1
BML	Bill Me Later	No	2
CREDIT_CARD	Credit Card	Yes	5
DIGITAL_RIVER_DROPIN	Digital River DropIn	Yes	7
DRPayPal		Yes	6
DW_ANDROID_PAY	Android Pay	No	4
DW_APPLE_PAY	Apple Pay	No	3
GIFT_CERTIFICATE	Gift Certificate	Yes	8

GIFT_CERTIFICATE Details

Description:	
	HTML Editor
Image:	Select
Payment Processor:	BASIC_CUSTOMER_CREDIT <BASIC_CUSTOMER_CREDIT>
Countries:	All Edit

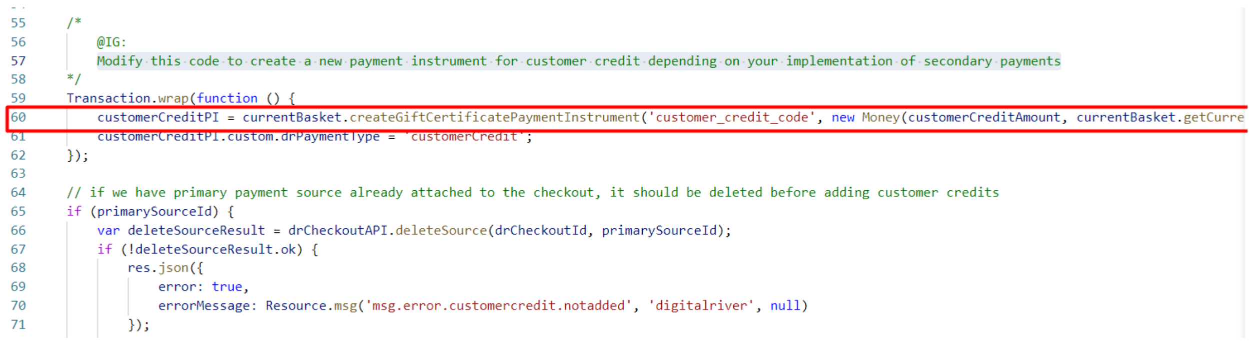
Controllers endpoint

Three controllers ***DigitalRiver-AddCustomerCredit, DigitalRiver-RemoveCustomerCredit and DigitalRiver-CustomerCredits***, are provided in the cartridge to integrate the backend part of the customer credit. Use these controllers to integrate customer credit functionality into your storefront.

1. **DigitalRiver-AddCustomerCredit controller.**

The controller makes an API call to Digital River to create a customer credit source, attaches the source to the current checkout and adds a customer credit to the current basket as a GIFT_CERTIFICATE payment method. Modify the code at line 60 to create a new payment instrument for customer credit depending on your implementation of secondary payments.
int_digitalriver_customercredit/cartridge/controllers/DigitalRiver.js

```
customerCreditPI =
currentBasket.createGiftCertificatePaymentInstrument('customer_credit_code', new
Money(customerCreditAmount, currentBasket.getCurrencyCode()));
```



```
55      /*
56          @IG:
57          Modify this code to create a new payment instrument for customer credit depending on your implementation of secondary payments
58      */
59      Transaction.wrap(function () {
60          customerCreditPI = currentBasket.createGiftCertificatePaymentInstrument('customer_credit_code', new Money(customerCreditAmount, currentBasket.getCurre
61          customerCreditPI.custom.drPaymentType = 'customerCredit';
62      });
63
64      // if we have primary payment source already attached to the checkout, it should be deleted before adding customer credits
65      if (primarySourceId) {
66          var deleteSourceResult = drCheckoutAPI.deleteSource(drCheckoutId, primarySourceId);
67          if (!deleteSourceResult.ok) {
68              res.json({
69                  error: true,
70                  errorMessage: Resource.msg('msg.error.customercredit.notadded', 'digitalriver', null)
71              });
```

Request type: POST

REQUEST BODY SCHEMA: application/json

PARAMETERS	TYPE	DESCRIPTION
customerCreditAmount	string (required)	Customer Credit value
primarySourceId	string (optional)	Primary source ID

Example of request
```
{
    "customerCreditAmount ": 10,
    "primarySourceId": "3cfd3746d-9b3a-4cb2-bcaa-g1476348h76a"
}
```

It is expected that customerCreditAmount parameter should be less than or equal to the amountRemainingToBeContributed value on the checkout. The value of customerCreditAmount should be checked to ensure that the amount matches the criteria before making a request. In case an amount greater is sent, a response with the error message will be received.

The primarySourceId parameter should be passed in case the primary source is already attached to the checkout.

Prior to calling the DigitalRiver-AddCustomerCredit controller, make a call to the DigitalRiver-CustomerCredits controller and obtain amountRemainingToBeContributed and primarySourceId from the response.

Example of error response:
```
{
  "action": "DigitalRiver-AddCustomerCredit",
  "queryString": "",
  "locale": "en_US",
  "error": true,
  "errorMessage": "Invalid customer credit amount"
}
```

RESPONSE BODY SCHEMA: application/json
Customer Credit Source created successfully

PARAMETERS	TYPE	DESCRIPTION
success: true	boolean	Customer Credit Source created successfully
customerCreditSources	Array	Array of Customer Credit sources that are applied to the checkout
adjustedGrandTotal	Object	
value	number	Basket total value minus customer credit value

formatted	string	Basket total value minus customer credit value in the format of the current site currency
customerCreditTotal	Object	
value	number	The sum of all Customer Credit values
formatted	string	The sum of all Customer Credit values in the format of the current site currency
amountRemainingToBeContributed	number	Represents the amount needed to fully fund the transaction

Example of response

```
{
  "action": "DigitalRiver-AddCustomerCredit",
  "queryString": "",
  "locale": "en_US",
  "success": true,
  "customerCreditSources": [
    {
      "id": "9ced946c-9c5a-4bb7-bddb-e1976368b71e",
      "type": "customerCredit",
      "amount": 10,
      "owner": {
        "firstName": "My_first_name",
        "lastName": "My_last_name",
        "email": "test@test.net",
        "address": {
          "line1": "1 Wall St billing l1",
          "city": "New York",
          "postalCode": "10004",
          "state": "NY",
          "country": "US"
        }
      },
      "customerCredit": {},
      "formatted": "$10.00"
    },
    {
      "id": "3ed7a503-8f13-4a78-81f2-6ad749dd99a5",
      "type": "customerCredit",
      "amount": 5,
      "owner": {
        "firstName": "My_first_name",
        "lastName": "My_last_name",
        "email": "test@test.net",
        "address": {
          "line1": "1 Wall St billing l1",
          "city": "New York",
          "postalCode": "10004",
          "state": "NY",
          "country": "US"
        }
      },
      "customerCredit": {},
      "formatted": "$5.00"
    }

  ],
  "adjustedGrandTotal": {
    "value": 74.11,
    "formatted": "$74.11",
    "customerCreditTotal": {
      "value": 15,
      "formatted": "$15.00"
    }
  },
  "amountRemainingToBeContributed": 74.11
}
```

RESPONSE BODY SCHEMA: application/json
Error creating Customer Credit Source

PARAMETERS	TYPE	DESCRIPTION
error: true	boolean	Customer Credit Source created successfully
errorMessage	string	Error message

Example of error response

```
{
  "action": "DigitalRiver-AddCustomerCredit",
  "queryString": "",
  "locale": "en_US",
  "error": true,
  "errorMessage": “There was a problem adding customer credit source”
}
```

NOTE: The error message returned is meant for display to the shopper. A more detailed technical error message can be located in the cartridge logs in the business manager.

2. DigitalRiver-RemoveCustomerCredit controller.

Deletes the customer credit source firm the current checkout and removes the customer credit from the current basket.

Request type: POST

REQUEST BODY SCHEMA: application/json

PARAMETERS	TYPE	DESCRIPTION
sourceId - string	string (required)	Source ID of the customer credit source to be removed

Example of request
```
{
    “sourceId”: “9ced946c-9c5a-4bb7-bddb-e1976368b71e”
}
```

RESPONSE BODY SCHEMA: application/json
Customer Credit Source remover successfully

PARAMETERS	TYPE	DESCRIPTION
success: true	boolean	Customer Credit Source created successfully
customerCreditSources	Array	Array of Customer Credit sources that are applied to the checkout after the sourceId from the request is removed. If no customer credit sources remain on the checkout, this array will be empty.
adjustedGrandTotal	Object	
value	number	Basket total value minus customer credit value
formatted	string	Basket total value minus customer credit value in the format of the current site currency
customerCreditTotal	Object	
value	number	The sum of all Customer Credit values
formatted	string	The sum of all Customer Credit values in the format of the current site currency
amountRemainingToBeContributed	number	Represents the amount needed to fully fund the transaction

Example of response
```
{
  "action": "DigitalRiver-RemoveCustomerCredit",
  "queryString": "",
  "locale": "en_US",
  "success": true,
  "customerCreditSources": [],
  "adjustedGrandTotal": {
    "value": 89.11,
    "formatted": "$89.11",
    "customerCreditTotal": {
      "value": 0,
      "formatted": "$0.00"
    }
  },
  "amountRemainingToBeContributed": 89.11
}
```

RESPONSE BODY SCHEMA: application/json
Error removed Customer Credit Source

PARAMETERS	TYPE	DESCRIPTION
error: true	boolean	Customer Credit Source created successfully
errorMessage	string	Error message

Example of error response

```
{
  "action": "DigitalRiver-RemoveCustomerCredit",
  "queryString": "",
  "locale": "en_US",
  "error": true,
  "errorMessage": “There was a problem removing customer credit
source”
}
```

NOTE: The error message returned is meant for display to the shopper. A more detailed technical error message can be located in the cartridge logs in the business manager.

3. DigitalRiver-CustomerCredits controller.

This controller can be used to retrieve actual information about already applied customer credit sources, primary source, and some total amounts. It is expected that this controller will be called to get information before adding new customer credit sources.

Request type: **GET**
No request parameters.

RESPONSE BODY SCHEMA: application/json
Customer Credit Source get successfully

PARAMETERS	TYPE	DESCRIPTION
success: true	boolean	Customer Credit Source created successfully
customerCreditSources	Array	Array of Customer Credit sources that are applied to the checkout
adjustedGrandTotal	Object	
value	number	Basket total value minus customer credit value

formatted	string	Basket total value minus customer credit value in the format of the current site currency
customerCreditTotal	Object	
value	number	The sum of all Customer Credit values
formatted	string	The sum of all Customer Credit values in the format of the current site currency
primarySourceId	string	Unique identifier for the primary source (non-customer credit) that is already applied to the checkout. If no primary source has been applied to the checkout this value will be empty.
amountRemainingToBeContributed	number	Represents the amount needed to fully fund the transaction

Example of response

```
{
  "action": "DigitalRiver-CustomerCredits",
  "queryString": "",
  "locale": "en_US",
  "success": true,
  "customerCreditSources": [
    {
      "id": "81aa2c77-4108-466d-aa0f-11bf382b861f",
      "type": "customerCredit",
      "amount": 10,
      "owner": {
        "firstName": "My_first_name",
        "lastName": "My_last_name",
        "email": "tmsdfsd@ukr.net",
        "address": {
          "line1": "1 Wall St billing l1",
          "city": "New York",
          "postalCode": "10004",
          "state": "NY",
          "country": "US"
        }
      },
      "customerCredit": {},
      "formatted": "$10.00"
    },
    {
      "id": "3ed7a503-8f13-4a78-81f2-6ad749dd99a5",
      "type": "customerCredit",
      "amount": 5,
      "owner": {
        "firstName": "My_first_name",
        "lastName": "My_last_name",
        "email": "tmsdfsd@ukr.net",
        "address": {
          "line1": "1 Wall St billing l1",
          "city": "New York",
          "postalCode": "10004",
          "state": "NY",
          "country": "US"
        }
      },
      "customerCredit": {},
      "formatted": "$5.00"
    }

  ],
  "adjustedGrandTotal": {
    "value": 54.11,
    "formatted": "$54.11",
    "customerCreditTotal": {
      "value": 15,
      "formatted": "$15.00"
    }
  },
  "primarySourceId": "83hh7c67-4207-455d-aa3a-94fo349k136s"
  "amountRemainingToBeContributed": 54.11
}
```

RESPONSE BODY SCHEMA: application/json
Error creating Customer Credit Source

PARAMETERS	TYPE	DESCRIPTION
error: true	boolean	Customer Credit Source created successfully
errorMessage	string	Error message

Example of error response

```
{
  "action": "DigitalRiver-CustomerCredits",
  "queryString": "",
  "locale": "en_US",
  "error": true,
  "errorMessage": “Failed to load customer credit sources”
}
```

Custom Code

This section details changes included in the cartridge to support the implementation of the customer credit feature. It is expected that during implementation of this feature that changes will need to be made to the code to account for the specific client use case. This section also highlights where changes may be needed.

Note: all changes are already made in the int_digitalriver_customercredit cartridge.

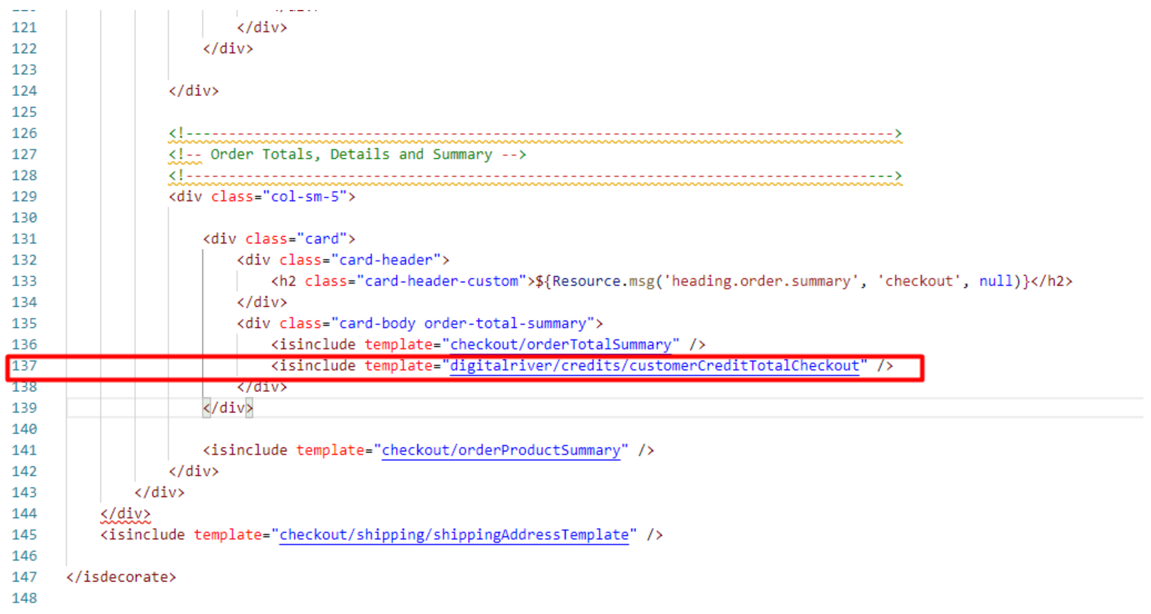
Perform changes in templates

1. cartridge\templates\default\checkout\checkout.isml

Next change allows you to display customer credit lines in order of Totals, Details and Summary.

Add the Digital River customer credit template tle checkout page to the line 137:

```
<isinclude template="digitalriver/credits/customerCreditTotalCheckout" />
```



```
121                 </div>
122             </div>
123
124         </div>
125
126         <!----------------------------------------------------------------------------->
127         <!-- Order Totals, Details and Summary -->
128         <!----------------------------------------------------------------------------->
129         <div class="col-sm-5">
130
131             <div class="card">
132                 <div class="card-header">
133                     <h2 class="card-header-custom">${Resource.msg('heading.order.summary', 'checkout', null)}</h2>
134                 </div>
135                 <div class="card-body order-total-summary">
136                     <isinclude template="checkout/orderTotalSummary" />
137                     <isinclude template="digitalriver/credits/customerCreditTotalCheckout" />
138                 </div>
139             </div>
140
141             <isinclude template="checkout/orderProductSummary" />
142         </div>
143     </div>
144 </div>
145 <isinclude template="checkout/shipping/shippingAddressTemplate" />
146
147 </isdecorate>
148
```

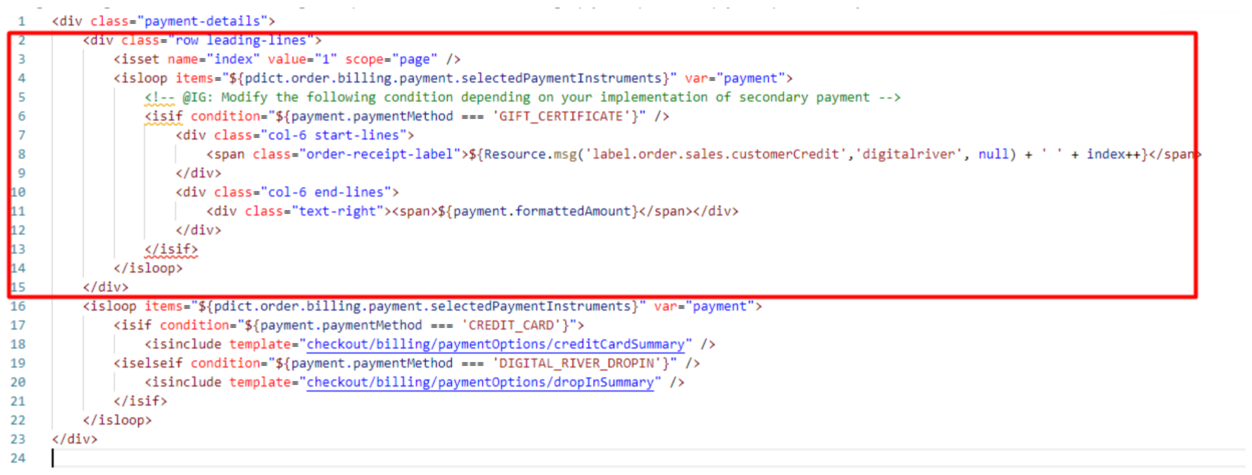
2. cartridge\templates\default\checkout\billing\paymentOptions\paymentOptionsSummary.isml

Next changes add the Customer Credits section to the payment info section at the order review page.

Add rows in line 2:

```
<div class="row leading-lines">
    <isset name="index" value="1" scope="page" />
    <isloop items="${pdict.order.billing.payment.selectedPaymentInstruments}" var="payment">
        <!-- @IG: Modify the following condition depending on your implementation of secondary
payment -->
        <isif condition="${payment.paymentMethod === 'GIFT_CERTIFICATE'}" />
            <div class="col-6 start-lines">
                <span
class="order-receipt-label">${Resource.msg('label.order.sales.customerCredit','digitalriver',
null) + ' ' + index++}</span>
            </div>
            <div class="col-6 end-lines">
```

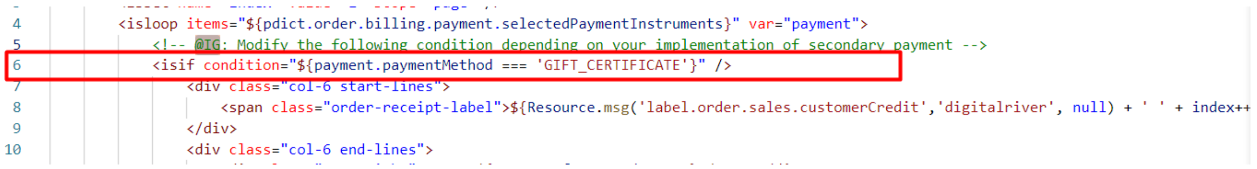
```
                <div class="text-right"><span>${payment.formattedAmount}</span></div>
            </div>
        </isif>
    </isloop>
</div>
```



```
1  <div class="payment-details">
2      <div class="row leading-lines">
3          <isset name="index" value="1" scope="page" />
4          <isloop items="${pdict.order.billing.payment.selectedPaymentInstruments}" var="payment">
5              <!-- @IG: Modify the following condition depending on your implementation of secondary payment -->
6              <isif condition="${payment.paymentMethod === 'GIFT_CERTIFICATE'}" />
7                  <div class="col-6 start-lines">
8                      <span class="order-receipt-label">${Resource.msg('label.order.sales.customerCredit','digitalriver', null) + ' ' + index++}</span>
9                  </div>
10                 <div class="col-6 end-lines">
11                     <div class="text-right"><span>${payment.formattedAmount}</span></div>
12                 </div>
13             </isif>
14         </isloop>
15     </div>
16     <isloop items="${pdict.order.billing.payment.selectedPaymentInstruments}" var="payment">
17         <isif condition="${payment.paymentMethod === 'CREDIT_CARD'}">
18             <isinclude template="checkout/billing/paymentOptions/creditCardSummary" />
19         <iselseif condition="${payment.paymentMethod === 'DIGITAL_RIVER_DROPIN'}" />
20             <isinclude template="checkout/billing/paymentOptions/dropInSummary" />
21         </isif>
22     </isloop>
23 </div>
24
```

Modify the code at line 6 depending on the payment method that is used for Customer Credit.

```
<isif condition="${payment.paymentMethod === 'GIFT_CERTIFICATE'}" />
```



```
4          <isloop items="${pdict.order.billing.payment.selectedPaymentInstruments}" var="payment">
5              <!-- @IG: Modify the following condition depending on your implementation of secondary payment -->
6              <isif condition="${payment.paymentMethod === 'GIFT_CERTIFICATE'}" />
7                  <div class="col-6 start-lines">
8                      <span class="order-receipt-label">${Resource.msg('label.order.sales.customerCredit','digitalriver', null) + ' ' + index++
9                  </div>
10                 <div class="col-6 end-lines">
```

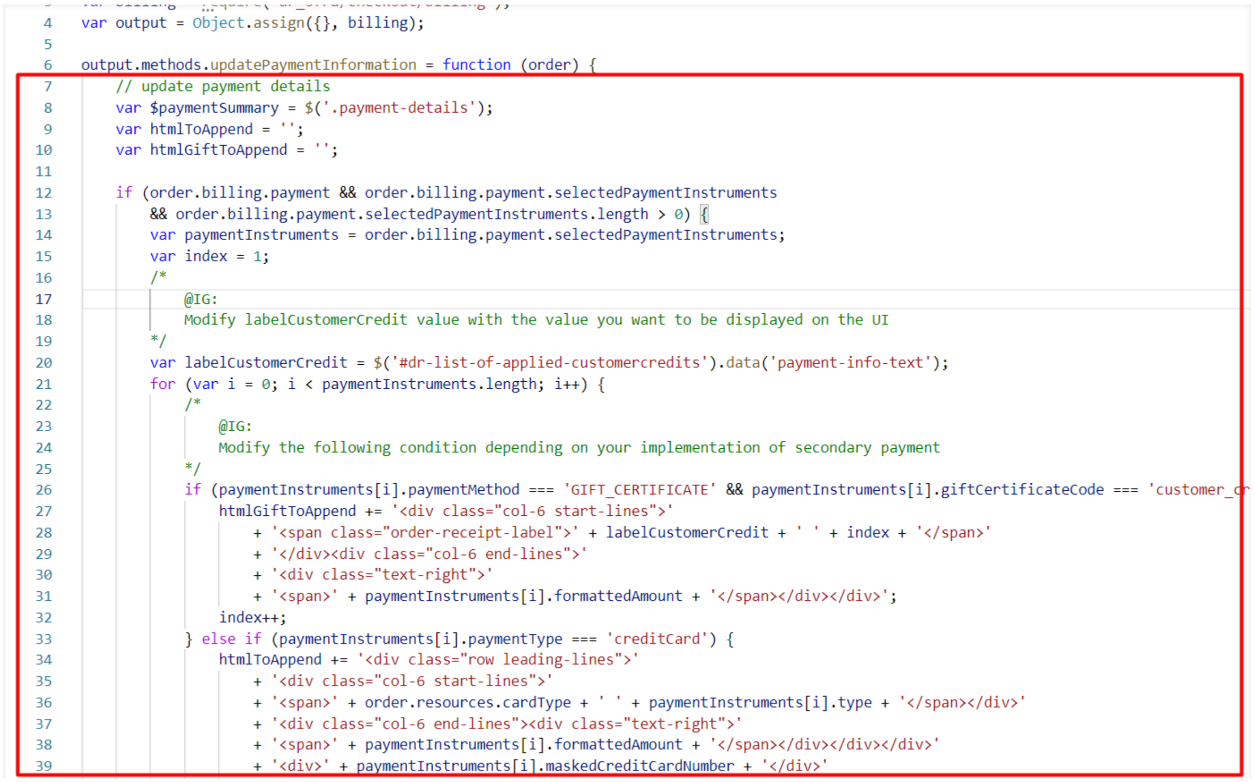
Perform changes in client script

Note: Don't forget to compile client side scripts after making the changes in the source code.

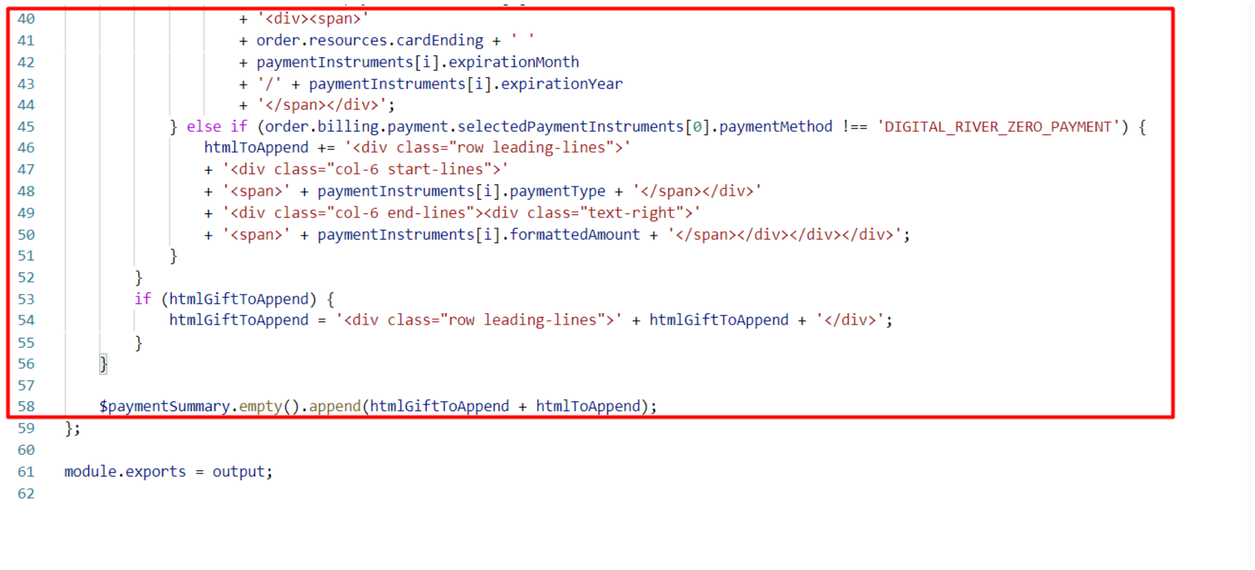
1. cartridge/client/default/js/checkout/billing.js

Change *updatePaymentInformation* function for updating the customer credit in the payment section on the checkout page in line 6:

```
output.methods.updatePaymentInformation = function (order) {
    // update payment details
    var $paymentSummary = $('.payment-details');
    var htmlToAppend = '';
    var htmlGiftToAppend = '';

    if (order.billing.payment && order.billing.payment.selectedPaymentInstruments
        && order.billing.payment.selectedPaymentInstruments.length > 0) {
        var paymentInstruments = order.billing.payment.selectedPaymentInstruments;
        var index = 1;
        /*
            @IG:
            Modify labelCustomerCredit value with the value you want to be displayed on the UI
        */
        var labelCustomerCredit =
$('#dr-list-of-applied-customercredits').data('payment-info-text');
        for (var i = 0; i < paymentInstruments.length; i++) {
            /*
                @IG:
                Modify the following condition depending on your implementation of secondary
payment
            */
            if (paymentInstruments[i].paymentMethod === 'GIFT_CERTIFICATE' &&
paymentInstruments[i].giftCertificateCode === 'customer_credit_code') {
                htmlGiftToAppend += '<div class="col-6 start-lines">'
                    + '<span class="order-receipt-label">' + labelCustomerCredit + ' ' + index
+ '</span>'
                    + '</div><div class="col-6 end-lines">'
                    + '<div class="text-right">'
                    + '<span>' + paymentInstruments[i].formattedAmount +
'</span></div></div>';
                index++;
            } else if (paymentInstruments[i].paymentType === 'creditCard') {
                htmlToAppend += '<div class="row leading-lines">'
                    + '<div class="col-6 start-lines">'
                    + '<span>' + order.resources.cardType + ' ' + paymentInstruments[i].type +
'</span></div>'
                    + '<div class="col-6 end-lines"><div class="text-right">'
                    + '<span>' + paymentInstruments[i].formattedAmount +
'</span></div></div></div>'
                    + '<div>' + paymentInstruments[i].maskedCreditCardNumber + '</div>'
                    + '<div><span>'
                    + order.resources.cardEnding + ' '
                    + paymentInstruments[i].expirationMonth
                    + '/' + paymentInstruments[i].expirationYear
                    + '</span></div>';
            } else if (order.billing.payment.selectedPaymentInstruments[0].paymentMethod !==
'DIGITAL_RIVER_ZERO_PAYMENT') {
                htmlToAppend += '<div class="row leading-lines">'
                + '<div class="col-6 start-lines">'
                + '<span>' + paymentInstruments[i].paymentType + '</span></div>'
                + '<div class="col-6 end-lines"><div class="text-right">'
                + '<span>' + paymentInstruments[i].formattedAmount +
'</span></div></div></div>';
            }
        }
        if (htmlGiftToAppend) {
            htmlGiftToAppend = '<div class="row leading-lines">' + htmlGiftToAppend +
'</div>';
        }
    }

    $paymentSummary.empty().append(htmlGiftToAppend + htmlToAppend);
}
```



```
 4  var output = Object.assign({}, billing);
 5
 6  output.methods.updatePaymentInformation = function (order) {
 7      // update payment details
 8      var $paymentSummary = $('.payment-details');
 9      var htmlToAppend = '';
10      var htmlGiftToAppend = '';
11
12      if (order.billing.payment && order.billing.payment.selectedPaymentInstruments
13          && order.billing.payment.selectedPaymentInstruments.length > 0) {
14          var paymentInstruments = order.billing.payment.selectedPaymentInstruments;
15          var index = 1;
16          /*
17              @IG:
18              Modify labelCustomerCredit value with the value you want to be displayed on the UI
19          */
20          var labelCustomerCredit = $('#dr-list-of-applied-customercredits').data('payment-info-text');
21          for (var i = 0; i < paymentInstruments.length; i++) {
22              /*
23                  @IG:
24                  Modify the following condition depending on your implementation of secondary payment
25              */
26              if (paymentInstruments[i].paymentMethod === 'GIFT_CERTIFICATE' && paymentInstruments[i].giftCertificateCode === 'customer_cr
27                  htmlGiftToAppend += '<div class="col-6 start-lines">'
28                      + '<span class="order-receipt-label">' + labelCustomerCredit + ' ' + index + '</span>'
29                      + '</div><div class="col-6 end-lines">'
30                      + '<div class="text-right">'
31                      + '<span>' + paymentInstruments[i].formattedAmount + '</span></div></div>';
32                  index++;
33              } else if (paymentInstruments[i].paymentType === 'creditCard') {
34                  htmlToAppend += '<div class="row leading-lines">'
35                      + '<div class="col-6 start-lines">'
36                      + '<span>' + order.resources.cardType + ' ' + paymentInstruments[i].type + '</span></div>'
37                      + '<div class="col-6 end-lines"><div class="text-right">'
38                      + '<span>' + paymentInstruments[i].formattedAmount + '</span></div></div></div>'
39                      + '<div>' + paymentInstruments[i].maskedCreditCardNumber + '</div>'
```



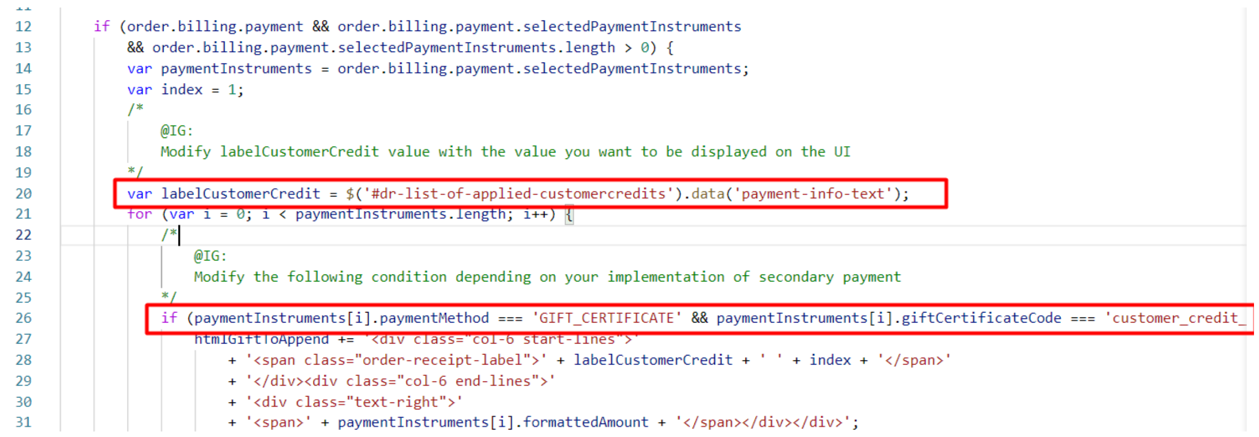
```
40                      + '<div><span>'
41                      + order.resources.cardEnding + ' '
42                      + paymentInstruments[i].expirationMonth
43                      + '/' + paymentInstruments[i].expirationYear
44                      + '</span></div>';
45              } else if (order.billing.payment.selectedPaymentInstruments[0].paymentMethod !== 'DIGITAL_RIVER_ZERO_PAYMENT') {
46                  htmlToAppend += '<div class="row leading-lines">'
47                  + '<div class="col-6 start-lines">'
48                  + '<span>' + paymentInstruments[i].paymentType + '</span></div>'
49                  + '<div class="col-6 end-lines"><div class="text-right">'
50                  + '<span>' + paymentInstruments[i].formattedAmount + '</span></div></div></div>';
51              }
52          }
53          if (htmlGiftToAppend) {
54              htmlGiftToAppend = '<div class="row leading-lines">' + htmlGiftToAppend + '</div>';
55          }
56      }
57
58      $paymentSummary.empty().append(htmlGiftToAppend + htmlToAppend);
59  };
60
61  module.exports = output;
62
```

Modify labelCustomerCredit value at line 20 with the value you want to be displayed on the UI.

```
var labelCustomerCredit = $('#dr-list-of-applied-customercredits').data('payment-info-text');
```

Modify the code at line 26 depending on the payment method that is used for Customer Credit.

```
if (paymentInstruments[i].paymentMethod === 'GIFT_CERTIFICATE' &&
paymentInstruments[i].giftCertificateCode === 'customer_credit_code') {
```



```
    if (order.billing.payment && order.billing.payment.selectedPaymentInstruments
        && order.billing.payment.selectedPaymentInstruments.length > 0) {
        var paymentInstruments = order.billing.payment.selectedPaymentInstruments;
        var index = 1;
        /*
            @IG:
            Modify labelCustomerCredit value with the value you want to be displayed on the UI
        */
        var labelCustomerCredit = $('#dr-list-of-applied-customercredits').data('payment-info-text');
        for (var i = 0; i < paymentInstruments.length; i++) {
            /*
                @IG:
                Modify the following condition depending on your implementation of secondary payment
            */
            if (paymentInstruments[i].paymentMethod === 'GIFT_CERTIFICATE' && paymentInstruments[i].giftCertificateCode === 'customer_credit_
                htmlGiftToAppend += '<div class="col-6 start-lines">'
                    + '<span class="order-receipt-label">' + labelCustomerCredit + ' ' + index + '</span>'
                    + '</div><div class="col-6 end-lines">'
                    + '<div class="text-right">'
                    + '<span>' + paymentInstruments[i].formattedAmount + '</span></div></div>';
```

2. cartridge/client/default/js/checkout/summary.js

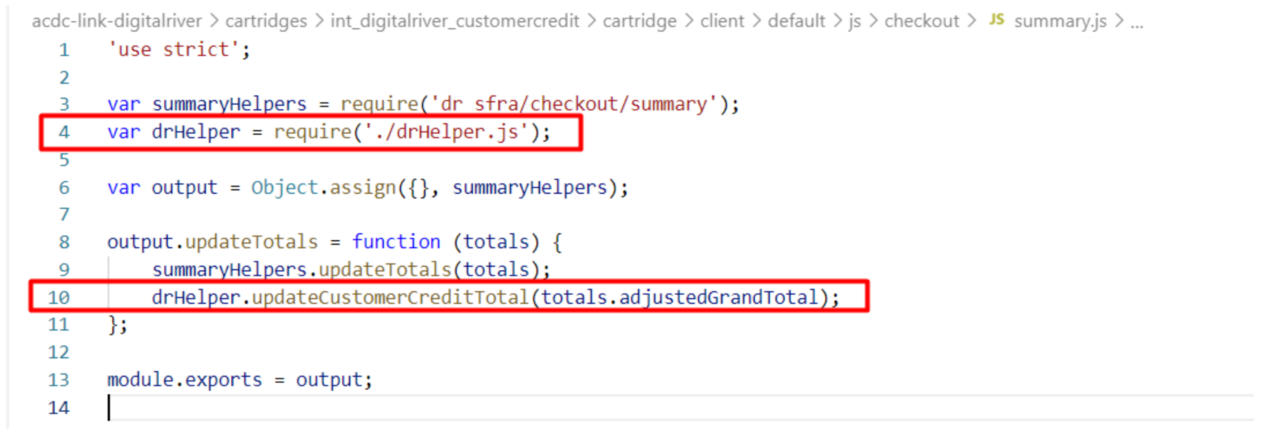
Change *updateTotals* function for updating the customer credit lines in order of Totals, Details and Summary.

Add row in line 4:

```
var drHelper = require('./drHelper.js');
```

Add row to the end of the function *updateTotals* in line 10:

```
drHelper.updateCustomerCreditTotal(totals.adjustedGrandTotal);
```



acdc-link-digitalriver > cartridges > int_digitalriver_customercredit > cartridge > client > default > js > checkout > JS summary.js > ...

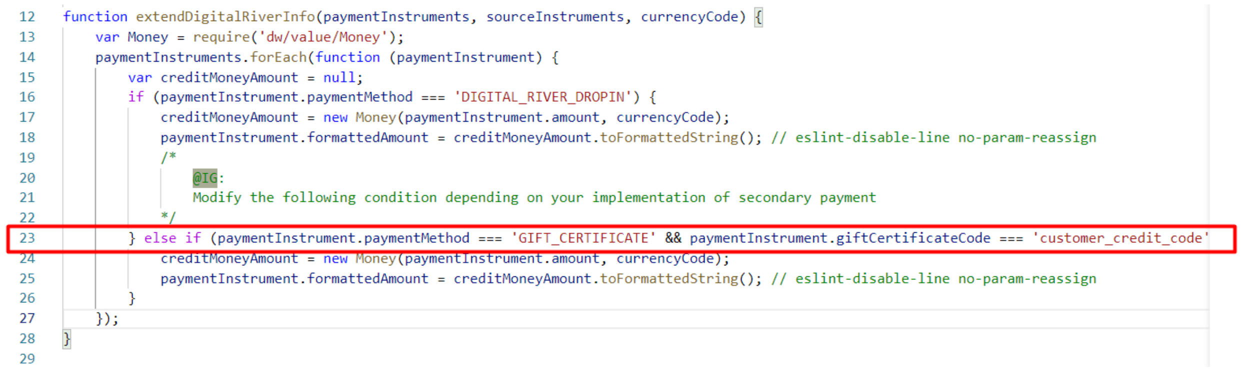
```
1   'use strict';
2
3   var summaryHelpers = require('dr_sfra/checkout/summary');
4   var drHelper = require('./drHelper.js');
5
6   var output = Object.assign({}, summaryHelpers);
7
8   output.updateTotals = function (totals) {
9       summaryHelpers.updateTotals(totals);
10      drHelper.updateCustomerCreditTotal(totals.adjustedGrandTotal);
11  };
12
13  module.exports = output;
14
```

Perform changes in server scripts

1. cartridge/models/payment.js

Modify the code at line 23 depending on the payment method that is used for Customer Credit.

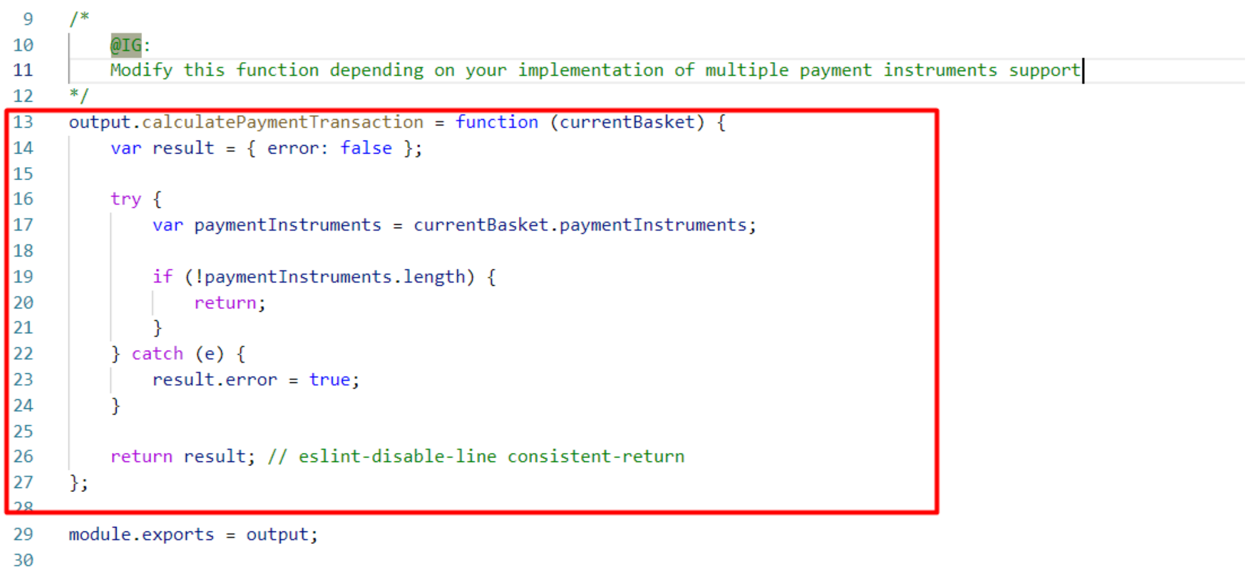
```
} else if (paymentInstrument.paymentMethod === 'GIFT_CERTIFICATE' &&
paymentInstrument.giftCertificateCode === 'customer_credit_code') {
```



```
function extendDigitalRiverInfo(paymentInstruments, sourceInstruments, currencyCode) {
    var Money = require('dw/value/Money');
    paymentInstruments.forEach(function (paymentInstrument) {
        var creditMoneyAmount = null;
        if (paymentInstrument.paymentMethod === 'DIGITAL_RIVER_DROPIN') {
            creditMoneyAmount = new Money(paymentInstrument.amount, currencyCode);
            paymentInstrument.formattedAmount = creditMoneyAmount.toFormattedString(); // eslint-disable-line no-param-reassign
            /*
                @IG:
                Modify the following condition depending on your implementation of secondary payment
            */
        } else if (paymentInstrument.paymentMethod === 'GIFT_CERTIFICATE' && paymentInstrument.giftCertificateCode === 'customer_credit_code'
            creditMoneyAmount = new Money(paymentInstrument.amount, currencyCode);
            paymentInstrument.formattedAmount = creditMoneyAmount.toFormattedString(); // eslint-disable-line no-param-reassign
        }
    });
}
```

2. cartridge/scripts/checkout/checkoutHelpers.js

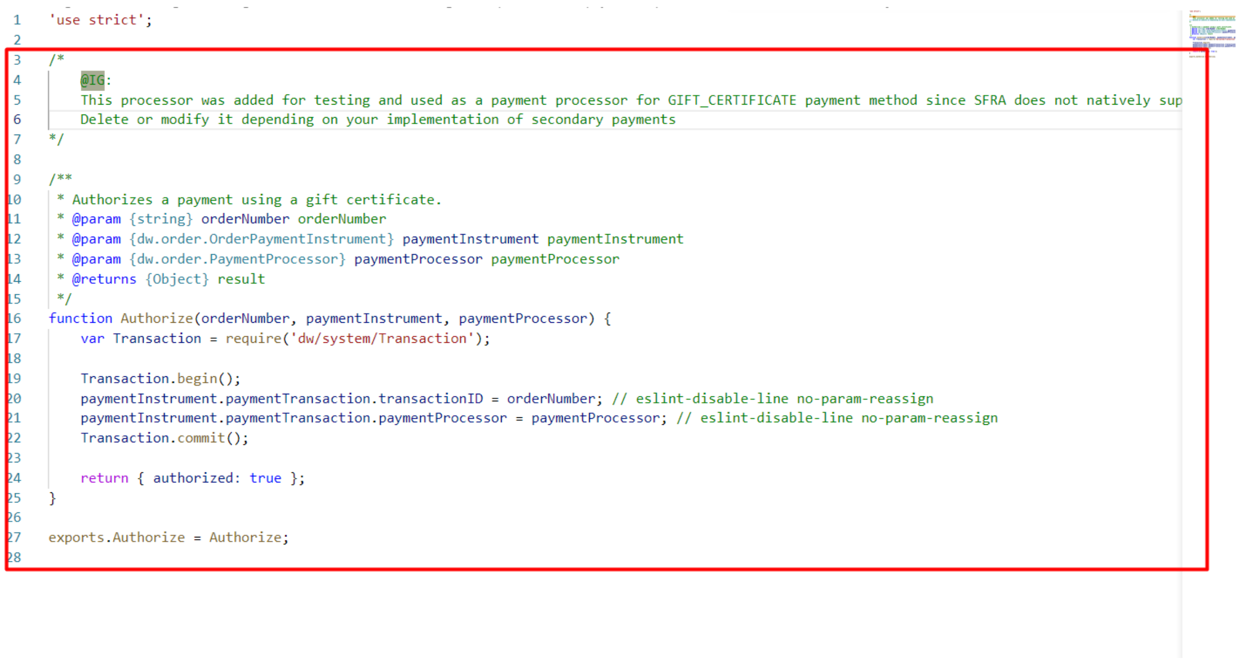
Modify **calculatePaymentTransaction** function at line 13 depending on your implementation of multiple payment instruments support.



```
9   /*
10      @IG:
11      Modify this function depending on your implementation of multiple payment instruments support
12  */
13  output.calculatePaymentTransaction = function (currentBasket) {
14      var result = { error: false };
15
16      try {
17          var paymentInstruments = currentBasket.paymentInstruments;
18
19          if (!paymentInstruments.length) {
20              return;
21          }
22      } catch (e) {
23          result.error = true;
24      }
25
26      return result; // eslint-disable-line consistent-return
27  };
28
29  module.exports = output;
30
```

3. cartridge/scripts/hooks/payment/processor/basic_customer_credit.js

This processor was added for testing and used as a payment processor for GIFT_CERTIFICATE payment method since SFRA does not natively support gift certificates. Delete or modify it depending on the payment method that is used for Customer Credit.



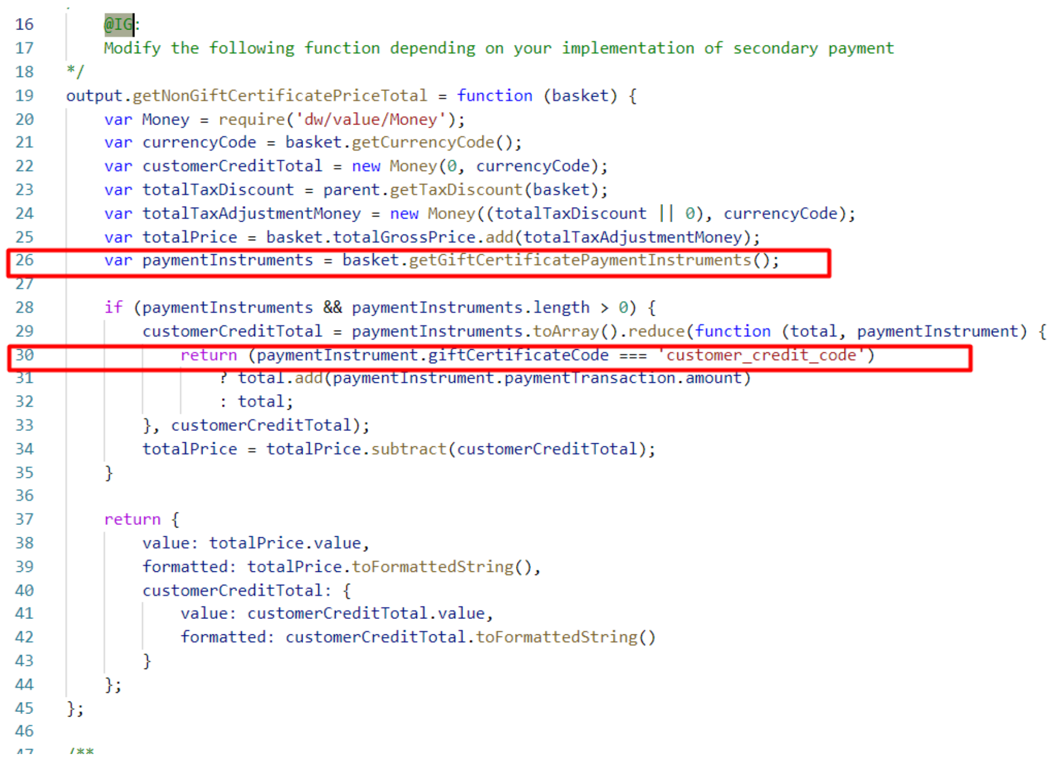
```
'use strict';

/*
    @IG:
    This processor was added for testing and used as a payment processor for GIFT_CERTIFICATE payment method since SFRA does not natively sup
    Delete or modify it depending on your implementation of secondary payments
*/

/**
 * Authorizes a payment using a gift certificate.
 * @param {string} orderNumber orderNumber
 * @param {dw.order.OrderPaymentInstrument} paymentInstrument paymentInstrument
 * @param {dw.order.PaymentProcessor} paymentProcessor paymentProcessor
 * @returns {Object} result
 */
function Authorize(orderNumber, paymentInstrument, paymentProcessor) {
    var Transaction = require('dw/system/Transaction');

    Transaction.begin();
    paymentInstrument.paymentTransaction.transactionID = orderNumber; // eslint-disable-line no-param-reassign
    paymentInstrument.paymentTransaction.paymentProcessor = paymentProcessor; // eslint-disable-line no-param-reassign
    Transaction.commit();

    return { authorized: true };
}

exports.Authorize = Authorize;
```

4. cartridge/scripts/digitalRiver/drTaxHelper.js

Modify the getNonGiftCertificatePriceTotal function at lines 26 and 30 depending on the payment method that is used for Customer Credit.

```
var paymentInstruments = basket.getGiftCertificatePaymentInstruments();
```

```
return (paymentInstrument.giftCertificateCode === 'customer_credit_code')
```

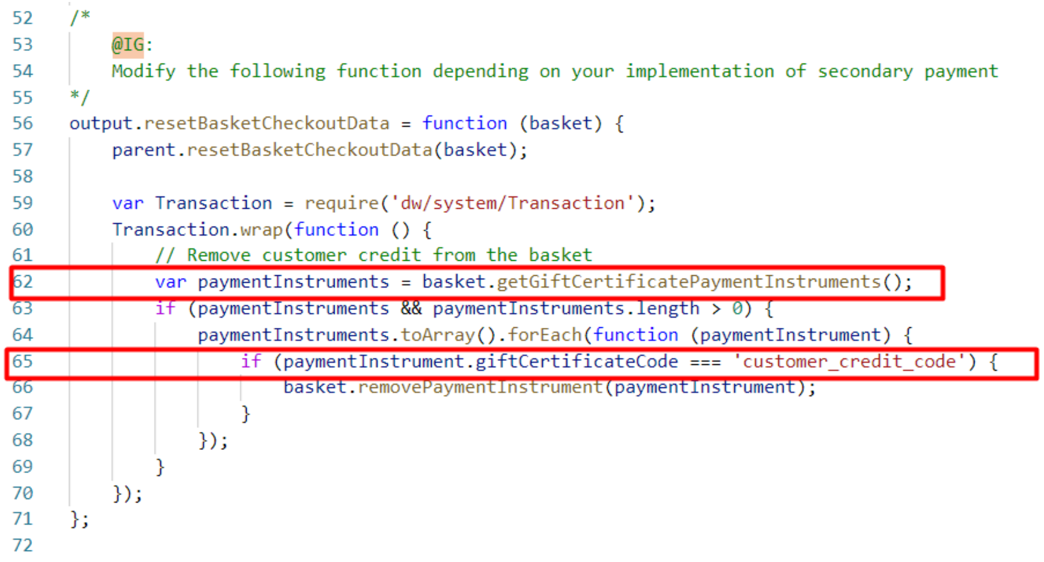


```
    @IC:
    Modify the following function depending on your implementation of secondary payment
*/
output.getNonGiftCertificatePriceTotal = function (basket) {
    var Money = require('dw/value/Money');
    var currencyCode = basket.getCurrencyCode();
    var customerCreditTotal = new Money(0, currencyCode);
    var totalTaxDiscount = parent.getTaxDiscount(basket);
    var totalTaxAdjustmentMoney = new Money((totalTaxDiscount || 0), currencyCode);
    var totalPrice = basket.totalGrossPrice.add(totalTaxAdjustmentMoney);
    var paymentInstruments = basket.getGiftCertificatePaymentInstruments();

    if (paymentInstruments && paymentInstruments.length > 0) {
        customerCreditTotal = paymentInstruments.toArray().reduce(function (total, paymentInstrument) {
            return (paymentInstrument.giftCertificateCode === 'customer_credit_code')
                ? total.add(paymentInstrument.paymentTransaction.amount)
                : total;
        }, customerCreditTotal);
        totalPrice = totalPrice.subtract(customerCreditTotal);
    }

    return {
        value: totalPrice.value,
        formatted: totalPrice.toFormattedString(),
        customerCreditTotal: {
            value: customerCreditTotal.value,
            formatted: customerCreditTotal.toFormattedString()
        }
    };
};

/**
```

Modify the resetBasketCheckoutData function at lines 62 and 65 depending on the payment method that is used for Customer Credit.

```
var paymentInstruments = basket.getGiftCertificatePaymentInstruments();
```

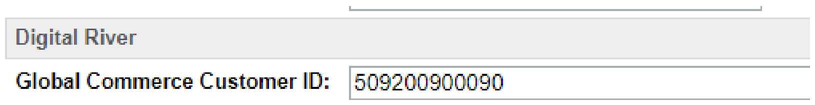
```
if (paymentInstrument.giftCertificateCode === 'customer_credit_code') {
```



```
/*
    @IG:
    Modify the following function depending on your implementation of secondary payment
*/
output.resetBasketCheckoutData = function (basket) {
    parent.resetBasketCheckoutData(basket);

    var Transaction = require('dw/system/Transaction');
    Transaction.wrap(function () {
        // Remove customer credit from the basket
        var paymentInstruments = basket.getGiftCertificatePaymentInstruments();
        if (paymentInstruments && paymentInstruments.length > 0) {
            paymentInstruments.toArray().forEach(function (paymentInstrument) {
                if (paymentInstrument.giftCertificateCode === 'customer_credit_code') {
                    basket.removePaymentInstrument(paymentInstrument);
                }
            });
        }
    });
};
```

5. Testing

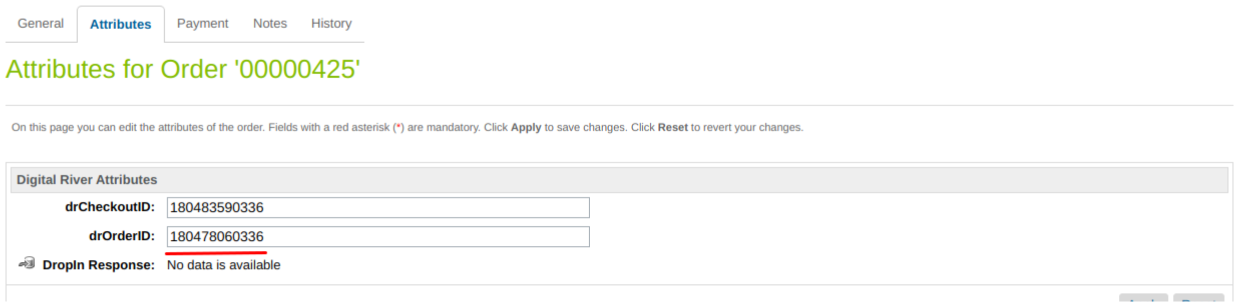
Create Customer Service – Create an account in the Storefront. In Business Manager, find that account. Look for the Digital River Attribute Grouping. You should see the returned value from Digital River now assigned to that profile.



Digital River

Global Commerce Customer ID:	509200900090

Placing an order – Once an order is placed in the storefront, it will have received an external order ID from Digital River. This number is how Digital River will know the order.



Attributes for Order '00000425'

On this page you can edit the attributes of the order. Fields with a red asterisk (*) are mandatory. Click **Apply** to save changes. Click **Reset** to revert your changes.

Digital River Attributes

drCheckoutID:	180483590336
drOrderID:	180478060336
DropIn Response:	No data is available

The following test cases were verified by Digital River during testing:

Test Scenarios/Test cases

Test cases for checking cartridge performance are described in the *DigitalRiver test cases* document.

6. Operations, Maintenance

Data Storage

Outside of Commerce Cloud, Digital River API stores Customer, Order, and SKU data across three areas: CPG (payments which is SOC and PCI-compliant), GC (Global Commerce commerce engine which is SOC and PCI-compliant), and PS (Payments Service which is PCI-compliant).

The location is Elastic Search, Oracle, and Cassandra, and the duration is a maximum of 10 years for Customer Data and Order Data with the exception of the EU “Right to be forgotten” GDPR requirements which we support.

Availability

When the cartridge is enabled, there is a possibility to use the Digital River payment method on the checkout process. Digital River payment method is available on storefront based on special settings (restrictions) saved in custom site preferences.

Services should always be available.

In the event that services are not available, orders and payments cannot be processed.

A Business Manager extension cartridge is provided that will give real-time status on available services by sending a static request and evaluating the response code sent back to SFCC.

Failover/Recovery Process

In the case of service disruption, the ecommerce experience will be impacted in the following ways:

1. Payment Methods will not render in the cart during checkout
2. Tax calculations will not return to the cart
3. Overall customers will be unable to transact on the ecommerce store which is leveraging Digital River for tax, payments, and risk.

Contact your Digital River representative as soon as possible.
In order to mitigate the consequences of service disruption, disable the cartridge in custom properties and set up other payment methods in Merchant Tools > Ordering > Payment methods. Please note that all payment instruments saved to customer accounts while the cartridge was disabled will be displayed but won’t be applicable after Digital River is back on. When the cartridge is disabled taxes will be calculated by SalesForce B2C tools.

Support

In the case of any errors, issues, or defects with your implementation of the *DigitalRiver* LINK Cartridge please contact the installing System Integrator or Partner.

If you believe there is a defect with the core functionality offered by the *DigitalRiver* LINK Cartridge please email with your Salesforce install URL, your System Integrator/Partner’s information, a description of the issue you are experiencing (with a transaction id if possible), and detailed reproduction steps.

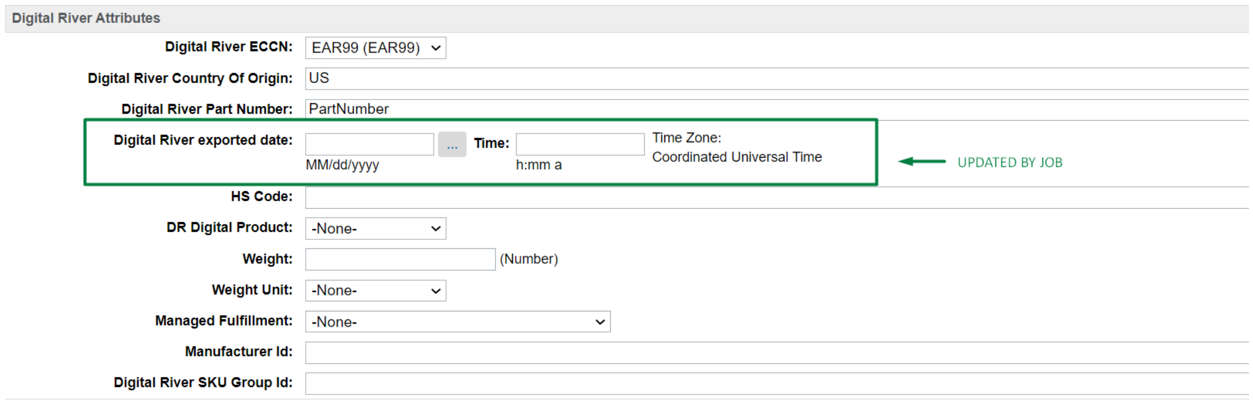
For all other questions or inquiries please contact your Salesforce representative.

7. User Guide

Roles, Responsibilities

SKU Creation job needs to be run any time new products are added to the catalog.

IMPORTANT NOTE:
In order SKU's to be processed on Digital River side all of products must have filled Digital River custom attributes

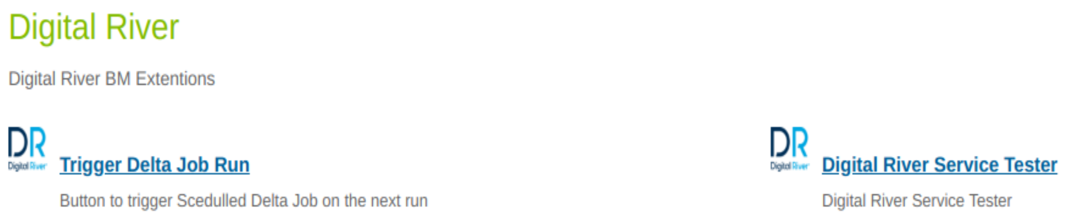


Digital River Attributes

Digital River ECCN:	EAR99 (EAR99)
Digital River Country Of Origin:	US
Digital River Part Number:	PartNumber
Digital River exported date:	MM/dd/yyyy Time: h:mm a Time Zone: Coordinated Universal Time
HS Code:	
DR Digital Product:	-None-
Weight:	(Number)
Weight Unit:	-None-
Managed Fulfillment:	-None-
Manufacturer Id:	
Digital River SKU Group Id:	

Business Manager

A new Business Manager menu options has been added under Merchant Tools



Digital River

Digital River BM Extentions

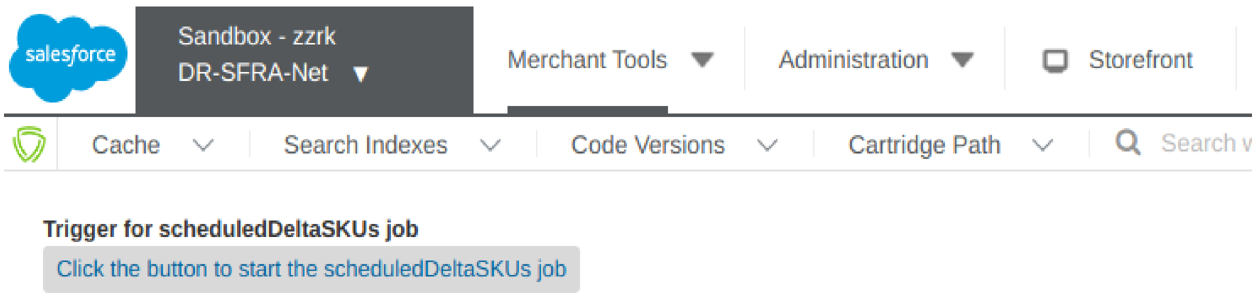
Trigger Delta Job Run

Button to trigger Scedulled Delta Job on the next run

Digital River Service Tester

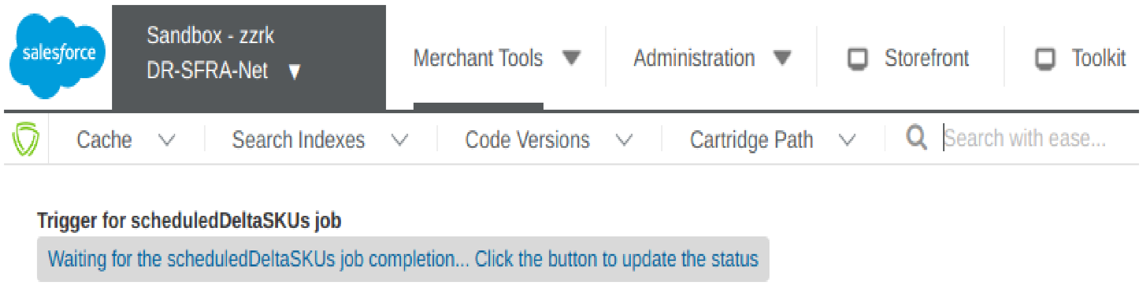
Digital River Service Tester

Trigger Delta Job Run - Button in BM to trigger Delta Job Run



Trigger for scheduledDeltaSKUs job

Click the button to start the scheduledDeltaSKUs job

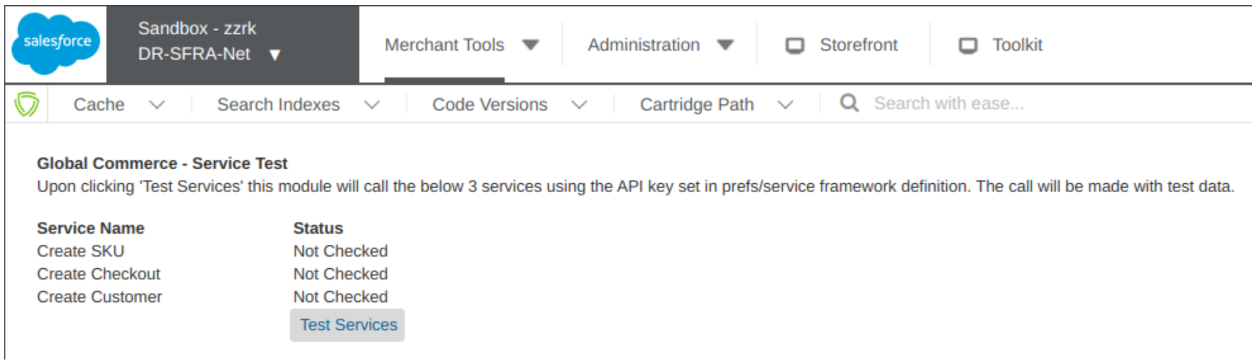


Trigger for scheduledDeltaSKUs job

Waiting for the scheduledDeltaSKUs job completion... Click the button to update the status

Clicking the button gives permission to execute job *sheduledDeltaSKUs*. Job will start according to the schedule set for it.
The button will have the corresponding text indicating that the job can be launched or that the job is being executed.

Digital River Service Tester - Button in BM to test Digital River service availability



Global Commerce - Service Test

Upon clicking 'Test Services' this module will call the below 3 services using the API key set in prefs/service framework definition. The call will be made with test data.

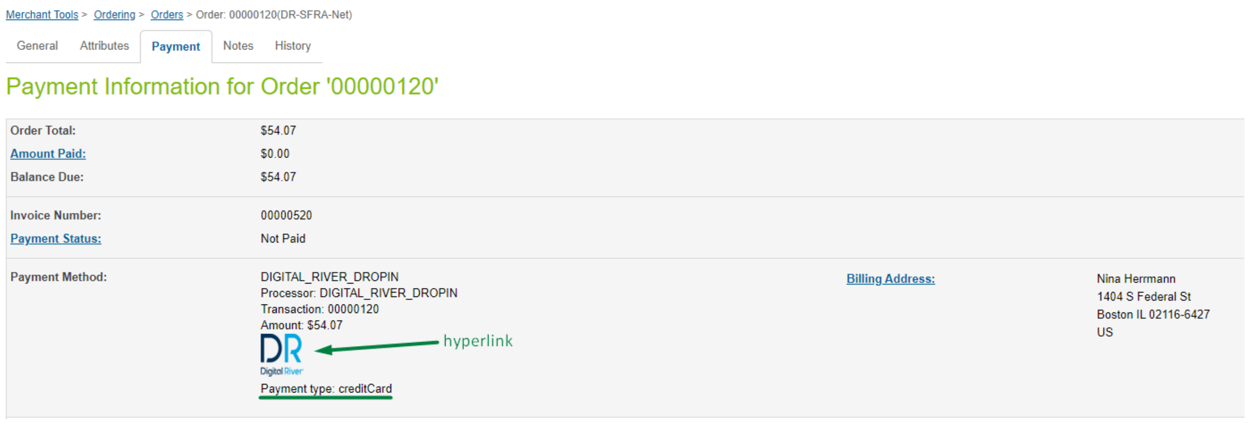
Service Name	Status
Create SKU	Not Checked
Create Checkout	Not Checked
Create Customer	Not Checked

Clicking the *Test Services* button will make calls to the indicated web services and evaluate response codes.

ORDER INFORMATION

DIGITAL_RIVER_DROPIN payment processor provides following order information in Business Manager

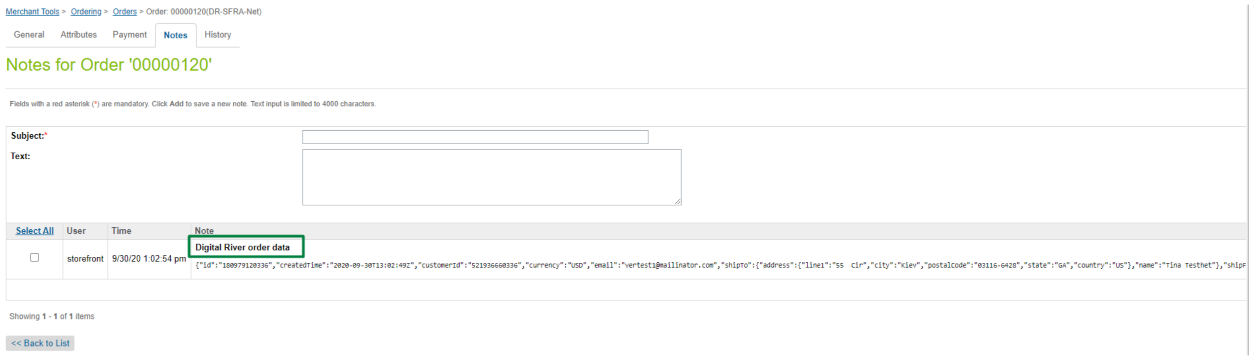
Merchant Tools > Orders> pick order > Payments
Payment type (creditCard, payPal, etc.) is mentioned
Hyperlink to order on Digital River dashboard


Merchant Tools > Ordering > Orders > Order: 00000120(DR-SFRA-Net)
General
Attributes
Payment
Notes
History
DR
Digital River
hyperlink

Payment Information for Order '00000120'

Order Total:	$54.07
Amount Paid:	$0.00
Balance Due:	$54.07
Invoice Number:	00000520
Payment Status:	Not Paid
Payment Method:	DIGITAL_RIVER_DROPIN Processor: DIGITAL_RIVER_DROPIN Transaction: 00000120 Amount: $54.07 Payment type: creditCard
Billing Address:	Nina Herrmann 1404 S Federal St Boston IL 02116-6427 US

Merchant Tools > Orders> pick order > Notes
Digital River "create order" response is been saved in order notes


Merchant Tools > Ordering > Orders > Order: 00000120(DR-SFRA-Net)

General | Attributes | Payment | Notes | History

Notes for Order '00000120'

Fields with a red asterisk (*) are mandatory. Click Add to save a new note. Text input is limited to 4000 characters.

Subject:*

Text:

Select All	User	Time	Note
☐	storefront	9/30/20 1:02:54 pm	**Digital River order data** `{"id":"180979120336","createdTime":"2020-09-30T13:02:49Z","customerId":"521936660336","currency":"USD","email":"vertest1@mailinator.com","shipTo":{"address":{"line1":"55  Cir","city":"Kiev","postalCode":"03116-6428","state":"GA","country":"US"},"name":"Tina Testhet"},"shipF`

Showing 1 - 1 of 1 items

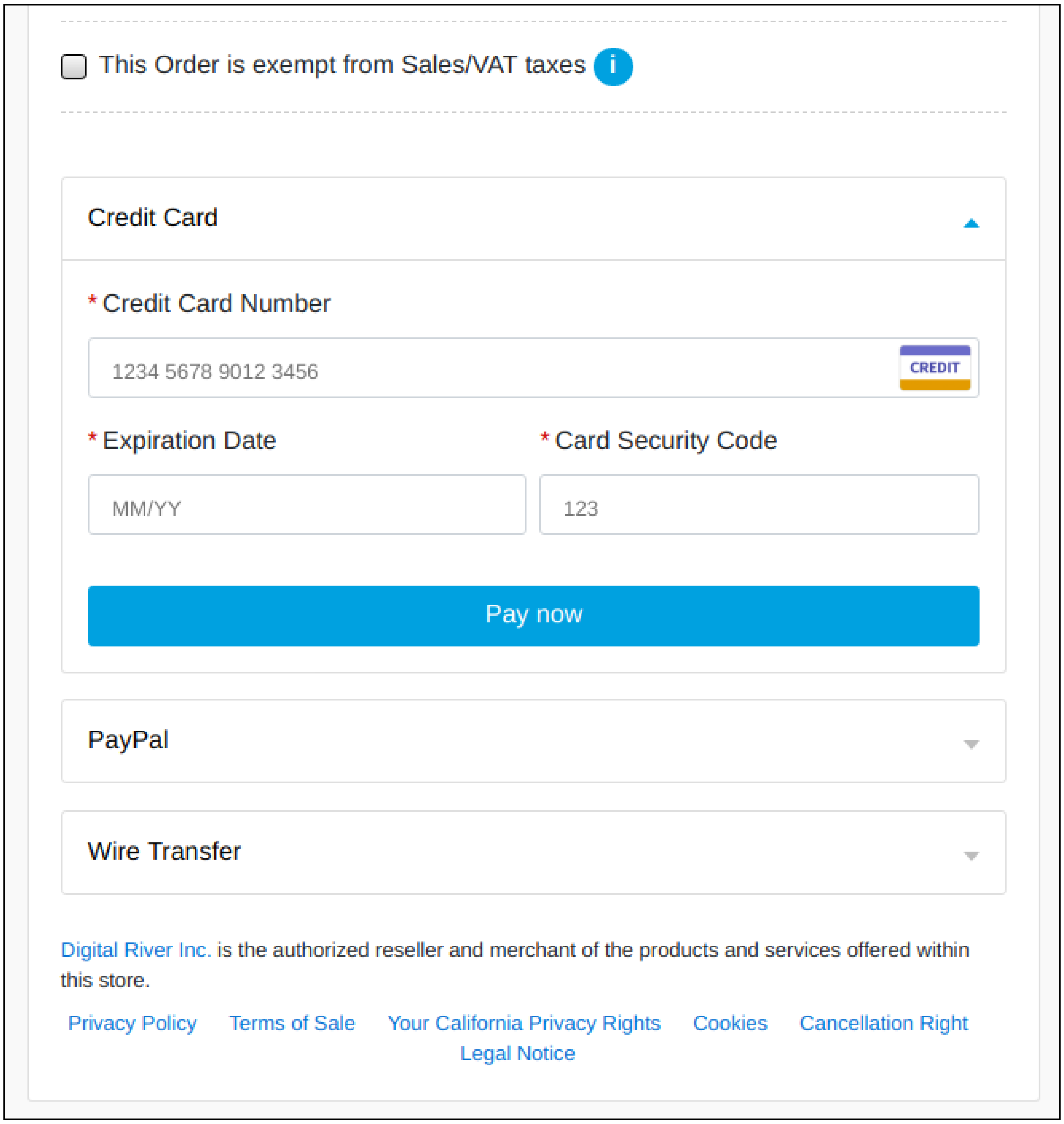
<< Back to List

Added *drExportedDate* custom attribute to the *Product*. It is updated each time the job is running.

Storefront Functionality

DigitalRiver.js library replaces built-in payment methods form and renders Digital River Drop-In payment methods and payment forms instead to fulfill the payment process.

Digital River payment methods available on billing page:


☐ This Order is exempt from Sales/VAT taxes

Credit Card

* Credit Card Number

1234 5678 9012 3456

* Expiration Date

MM/YY

* Card Security Code

123

Pay now

PayPal

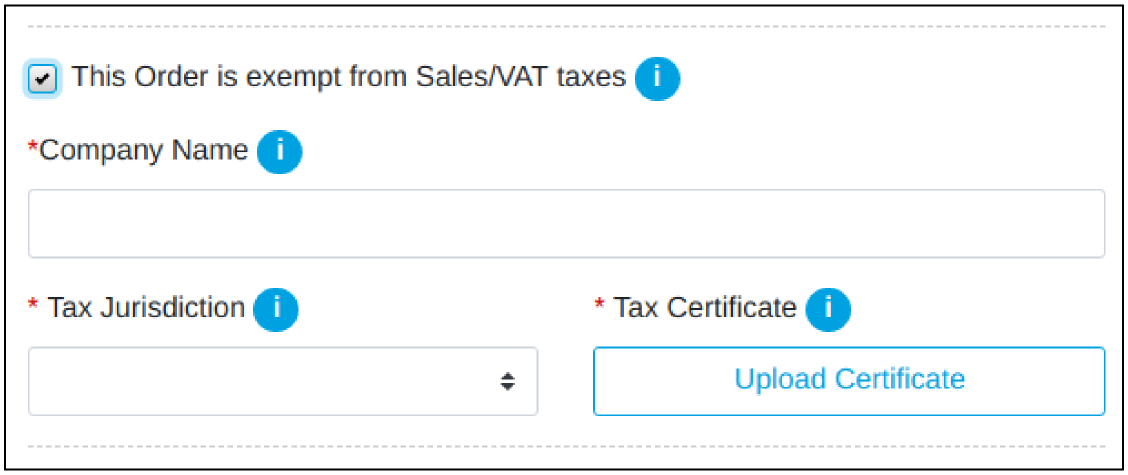
Wire Transfer

Digital River Inc. is the authorized reseller and merchant of the products and services offered within this store.

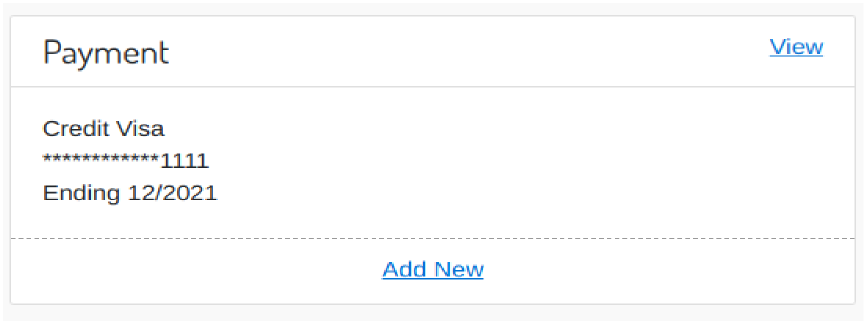
Privacy Policy Terms of Sale Your California Privacy Rights Cookies Cancellation Right Legal Notice

Added the ability for the customer to ask for tax exemptions - a checkbox to the Billing Form: *This Order is exempt from Sales/VAT taxes*.

Cartridge logic determines what additional fields should be displayed on the billing form.



When the Customer clicks on *Add New* link in *My Account* section,



the Drop-In form with the specific configuration will be shown allowing the customer to add a new payment within the Drop-In. A new source will be created.

8. Known Issues

Tested against Compatibility Mode: 21.7

Default locale was used for all implementations. The integration will work with any properly configured locale on the instance.

9. Release History

Version	**Date**	**Changes**
21.1.0	Dec 2020	Initial release
22.1.0	Feb 2022	Added Digital River customer credit
22.2.0	May 2022	
22.3.0	July 2022	