***Version <19.10>***

**Nosto Integration Guide**

**November 2018**

**Purpose of this document:**

The main purpose of this document is to outline general concepts describing the workflow, configuration and logging view.

This document includes a step by step guide related to Business Manager configuration and integration details.

SFCC Business Manager guide will help in selecting or updating values for each attribute using Merchant section of Business Manager. Also included in the guide are steps to add new attribute values using Administration section of Business Manager.

**Version History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comment** |
| Version v1.0 | 13/11/2018 | Nosto | Initial Draft Version |
| Version v2.0 | 20/11/2018 | Nosto | Updated based on change requests arrived during UAT phase |
| Version v3.0 | 12/2/2019 | Nosto | Updated 18.10 to 19.10 |

Contents

[**In Scope** 5](#_Toc530585835)

[Category Page 5](#_Toc530585836)

[Product Page 5](#_Toc530585837)

[Cart 5](#_Toc530585838)

[Order 5](#_Toc530585839)

[Customer 5](#_Toc530585840)

[**Out of Scope** 5](#_Toc530585841)

[Option Product Type 5](#_Toc530585842)

[Search 5](#_Toc530585843)

[Placements 5](#_Toc530585844)

[**Cartridge Installation** 6](#_Toc530585845)

[Cartridge Path for Controllers 6](#_Toc530585846)

[Cartridge Path for SFRA 6](#_Toc530585847)

[**Functional Specifications** 7](#_Toc530585848)

[How Nosto works 7](#_Toc530585849)

[Adding the Nosto Script 7](#_Toc530585850)

[Adding the Page Tagging 7](#_Toc530585851)

[Adding the Product Tagging 8](#_Toc530585852)

[Adding the Category/ Brand Tagging 9](#_Toc530585853)

[Adding the Cart Tagging 9](#_Toc530585854)

[Adding the Order Tagging 10](#_Toc530585855)

[Adding the Customer Information 11](#_Toc530585856)

[**Site Configurations** 12](#_Toc530585857)

[Merchant Tools 12](#_Toc530585858)

[Custom Preferences 12](#_Toc530585859)

[Nosto Brand Configuration (Category Level) 13](#_Toc530585860)

[**Customized Code** 16](#_Toc530585861)

[Controller Implementation 16](#_Toc530585862)

[**htmlhead.isml** 16](#_Toc530585863)

[**pt\_storefront.isml** 16](#_Toc530585864)

[**pt\_cart.isml** 17](#_Toc530585865)

[**footer.isml** 17](#_Toc530585866)

[**pt\_productdetails.isml** 18](#_Toc530585867)

[**pt\_categorylanding.isml** 18](#_Toc530585868)

[**pt\_productsearchresult.isml** 18](#_Toc530585869)

[**pt\_orderconfirmation.isml** 19](#_Toc530585870)

[SFRA Implementation 19](#_Toc530585871)

[**htmlHead.isml** 19](#_Toc530585872)

[**homePage.isml** 20](#_Toc530585873)

[**cart.isml** 20](#_Toc530585874)

[**pageFooter.isml** 20](#_Toc530585875)

[**productDetails.isml** 21](#_Toc530585876)

[**bundleDetails.isml** 21](#_Toc530585877)

[**setDetails.isml** 22](#_Toc530585878)

[**searchResults.isml** 22](#_Toc530585879)

[**catLanding.isml** 22](#_Toc530585880)

[**confirmation.isml** 23](#_Toc530585881)

[**page.isml** 23](#_Toc530585882)

[**checkout.isml** 23](#_Toc530585883)

# **In Scope**

## Category Page

Nosto utilizes meta tags to track what category or brand a certain visitor is viewing or what page type the currently viewed page is. This needs to be added to all category and brand/vendor pages.

## Product Page

Product tagging is a series of meta attributes that describe the product for Nosto. This is added to all product pages.

## Cart

Cart tagging is a series of meta attributes that describe the contents of the shopping cart for Nosto. This needs to be added to all pages except the order-confirmation page.

## Order

Order tagging is a series of meta attributes that notify Nosto of what products finally ended up in a finalised order. This needs to be added to the order-confirmation page.

## Customer

Customer information allows you to annotate contact details for the current customer. This needs to be added to all pages except the order-confirmation page.

# **Out of Scope**

## Option Product Type

## Search

## Placements

Retailers has to do customization to place placements.

# **Cartridge Installation**

To install the Nosto integration for the first time on their instance, user may perform the following steps:

1. Clone repository to local machine.
2. Import the cartridge (**int\_nosto)** into your workspace and link them to the Server Connection. **int\_nosto\_sfra** cartridge is also provided but it is not mandatory to use the cartridge as it is just representation of how the cartridge will work with SFRA.
3. To import site’s metadata, create zip file for metadata folder (metadata.zip). Import metadata.zip into your instance (Meta Data and Custom Objects and Schedules)
4. Assign the **int\_nosto** cartridges to site’s cartridge path. It should be configured in below sequence. The cartridge path sequence should look like this,

## Cartridge Path for Controllers

**“int\_nosto:** **app\_storefront\_controllers:** **app\_storefront\_core”**

## Cartridge Path for SFRA

**“int\_nosto:** **app\_storefront\_base”**

1. Add **int\_nosto\_sfra** cartridge before int\_nosto to understand what changes are needed in the base cartridge and to understand how the cartridge works. **int\_nosto\_sfra** is created just to simulate how the cartridge will work and retailers has to make sure similar changes are copied over to respective base cartridge as mentioned in the implementation section.
2. User can now make use of the cartridge.

# **Functional Specifications**

This guide introduces retailer how to implement Nosto on e-commerce store that does not yet have a dedicated plugin solution. First, it’s important to understand a few of the concepts before diving in to the documentation, to make sure that retailer as an integrator understand the different steps and tools required.

* Nosto initializes by using a snippet of Javascript that starts the dialogue to your dedicated account.
* Nosto depends on structured data that we extract from your site. At the bare minimum you need to go through all the steps listed in this document.

### How Nosto works

Nosto’s functionality is based on hidden html tags that must be placed in store’s page source. Tagging contextual details inside hidden Nosto specific tag-fields in standard html format allows Nosto to read data from the target site. The embed script added before this step looks for Nosto specific tag fields and sends the data stored inside the tags to Nosto’s service, meaning that context or actual data Nosto reads is mapped in the tags.

### Adding the Nosto Script

To start tracking visits and content the Nosto script needs to be active on all pages within the store where the user might navigate. $accountId is a variable and it must be configurable by the merchant. Also note that it must be possible to configure $accountId for each store view (e.g. each language) respectively i.e. Retailer need a separate Nosto account for each storefront.

The code snippet must be placed within the <head> section of retailer sites HTML content. Retailer can find their stores account IDs from the account list within the Nosto admin.

NOTE: This is handled in cartridge via site preference so no need to add this line in the code but this is just for reference.

**<script src="//connect.nosto.com/include/$accountID" async></script>**

### Adding the Page Tagging

#### **Home Page**

Page type should be “front” for front/ Home Page

< div class ="nosto\_page\_type" style="display:none">front</div>

#### **Cart Page**

Page type should be “cart” for Cart Page

< div class ="nosto\_page\_type" style="display:none">cart</div>

### Adding the Product Tagging

The service incorporates a crawling mechanism that collects and stores these products in our catalogue for use within the different delivery methods of Nosto. Please note that the meta attributes need to exist within the page source when the page has rendered and Nosto is unable to crawl details injected via Google Tag Manager or other dynamic sources based on Javascript.

Product page will also have the nosto\_category tag which represents the primary category of the product.

**Example**:

<div class="nosto\_page\_type" style="display:none">product</div>

<div class="nosto\_category" style="display:none">/Boats/Canoes</div>

<div class="nosto\_product">   
  <span class="product\_id">Canoe123</span>  
  <span class="name">Acme Canoe</span>  
  <span class="url">https://example.com/canoe123</span>  
  <span class="image\_url">https://image.example.com/canoe1.jpg</span>  
  <span class="availability">Instock</span>  
  <span class="price">999.50</span>

 <span class="list\_price">1299.00</span>  
  <span class="price\_currency\_code">USD</span>  
  <span class="category">/interior</span>  
  <span class="category">/interior/towels</span>

 <span class="description">This foldable canoe is easy to travel with.</span>

 <span class="brand">ACME</span>

 <span class="review\_count">20</span>

 <span class="review\_value">3.0</span>

 <span class="alternate\_image\_url"><http://www.example.com/image1.jpg></span>

 <span class="alternate\_image\_url"><http://www.example.com/image2.jpg></span>

 <span class="alternate\_image\_url"><http://www.example.com/image2.jpg></span>

 <span class="custom\_fields">

   <span class="occasion">free time</span>

   <span class="material">plastic</span>

 </span>

 <span class="nosto\_sku">

   <span class="id">Canoe123-1</span>

   <span class="name">Canoe - Red XL</span>

   <span class="price">1269.00</span>

   <span class="list\_price">1299.00</span>

   <span class="url">http://www.example.com/product/CANOE123#/1-size-xl/13-color-red</span>

   <span class="image\_url">http://www.example.com/product/images/CANOE123-1.jpg</span>

   <span class="availability">InStock</span>

   <span class="custom\_fields">

     <span class="color">red</span>

     <span class="size">XL</span>

   </span>

 </span>

 <span class="nosto\_sku">

   <span class="id">Canoe123-2</span>

   <span class="name">Canoe - Orange S</span>

   <span class="price">999.00</span>

   <span class="list\_price">1199.00</span>

   <span class="url">http://www.example.com/product/CANOE123#/1-size-s/13-color-orange</span>

   <span class="image\_url">http://www.example.com/product/images/CANOE123-2.jpg</span>

   <span class="availability">OutOfStock</span>

   <span class="custom\_fields">

     <span class="color">orange</span>

     <span class="size">S</span>

   </span>

 </span>

</div>

**Note**:

Master product selects first available variant product price by default.

### Adding the Category/ Brand Tagging

These values are then used for dynamic filtering for categories and brands applied through the Nosto admin UI or exposure of certain pop-up campaigns for page types.

The category tagging should be exposed whenever a user is viewing a certain category.

<div class="nosto\_page\_type" style="display:none">category</div>  
<div class="nosto\_category" style="display:none">/Boats/Canoes</div>

The brand tagging should be exposed whenever a user is viewing a certain brand or vendor.

<div class="nosto\_page\_type" style="display:none">category</div>  
<div class="nosto\_category" style="display:none">Acme</div>

### Adding the Cart Tagging

The service incorporates a crawling mechanism that collects and stores these products in our catalogue for use within the different delivery methods of Nosto. Cart tagging lets Nosto track add to cart events and also use the cart contents to tailor recommendations and utilise the data for other feature types like triggered emails or Facebook retargeting.

<div class="nosto\_cart" style="display:none">  
    <div class="line\_item">  
        <span class="product\_id">Canoe123</span>

       <span class="sku\_id">Canoe123-1</span>

       <span class="quantity">1</span>  
        <span class="name">Acme Canoe</span>  
        <span class="unit\_price">999.00</span>  
        <span class="price\_currency\_code">EUR</span>  
    </div>  
   
    <div class="line\_item">  
        <span class="product\_id">Canoe245</span>

       <span class="sku\_id">Canoe245-1</span>  
        <span class="quantity">3</span>  
        <span class="name">Acme Large Canoe</span>  
        <span class="unit\_price">19.00</span>  
        <span class="price\_currency\_code">EUR</span>  
    </div>  
</div>

### Adding the Order Tagging

The order tagging should be outputted on the order-thank-you page after a customer has finalized payment. If you are using a 3rd party payment provider and you do not have edit access to your order-thank-you page please reach out to Nosto support or your assigned Technical Solutions Manager for further assistance in setting up the tracking.

<div class="nosto\_page\_type" style="display:none">order</div>  
<div class="nosto\_purchase\_order" style="display:none">  
    <span class="order\_number">1445</span>  
   
    <div class="buyer">  
        <span class="email">john.doe@example.com</span>  
        <span class="first\_name">John</span>  
        <span class="last\_name">Doe</span>  
        <span class="marketing\_permission">false</span>  
    </div>  
   
    <div class="purchased\_items">  
        <div class="line\_item">  
            <span class="product\_id">Canoe123</span>

           <span class="sku\_id">Canoe123-1</span>  
            <span class="quantity">1</span>  
            <span class="name">Acme Canoe</span>  
            <span class="unit\_price">999.00</span>  
            <span class="price\_currency\_code">EUR</span>  
        </div>  
    
        <div class="line\_item">  
            <span class="product\_id">Canoe245</span>

           <span class="sku\_id">Canoe245-1</span>  
            <span class="quantity">3</span>  
            <span class="name">Acme Large Canoe</span>  
            <span class="unit\_price">19.00</span>  
            <span class="price\_currency\_code">EUR</span>  
        </div>  
    </div>  
</div>

**Note**:

marketing\_permission tag is set as false by default.

Retailer needs to include their own logic for marketing permissions. They can do this by updating the getMarketingPermission() function in nostoHelper.js.

### Adding the Customer Information

This information can be outputted from your ecommerce platform, CRM system or any other mechanisms you have in place onsite that can modify the tagging.

The customer tagging should be exposed on every page whenever we have the necessary information available in the front-end context.

<div class="nosto\_customer" style="display:none">  
  <span class="email">john.doe@example.com</span>  
</div>

The email address is the only required parameter in the tagging, but you can enrich the profile with other optional attributes.

<span class="first\_name">John</span>  
 <span class="last\_name">Doe</span>  
 <span class="marketing\_permission">false</span>

The marketing\_permission flag denotes whether the customer has consented to email marketing. If the marketing\_permission field is omitted, we assume that the current customer has not given their consent and Nosto will refrain from sending out any personalized triggered emails.

Marketing permission is false by default but if a user has explicitly agreed to receive marketing then you can set it to true manually. In practice this means reading and mapping the value from opt-in for marketing in your platform e.g. a consumer explicitly subscribed for marketing emails when checking out.

**Note**:

marketing\_permission tag is set as false by default.

Retailer needs to include their own logic for marketing permissions. They can do this by updating the getMarketingPermission() function in nostoHelper.js.

# **Site Configurations**

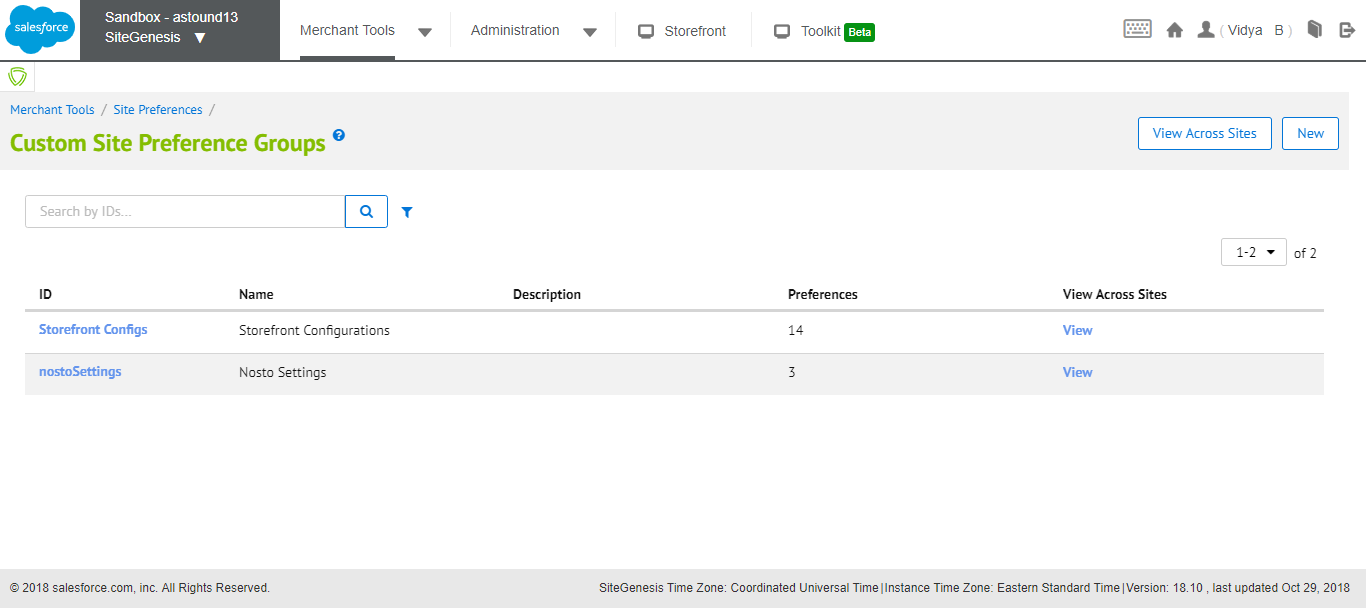
## Merchant Tools

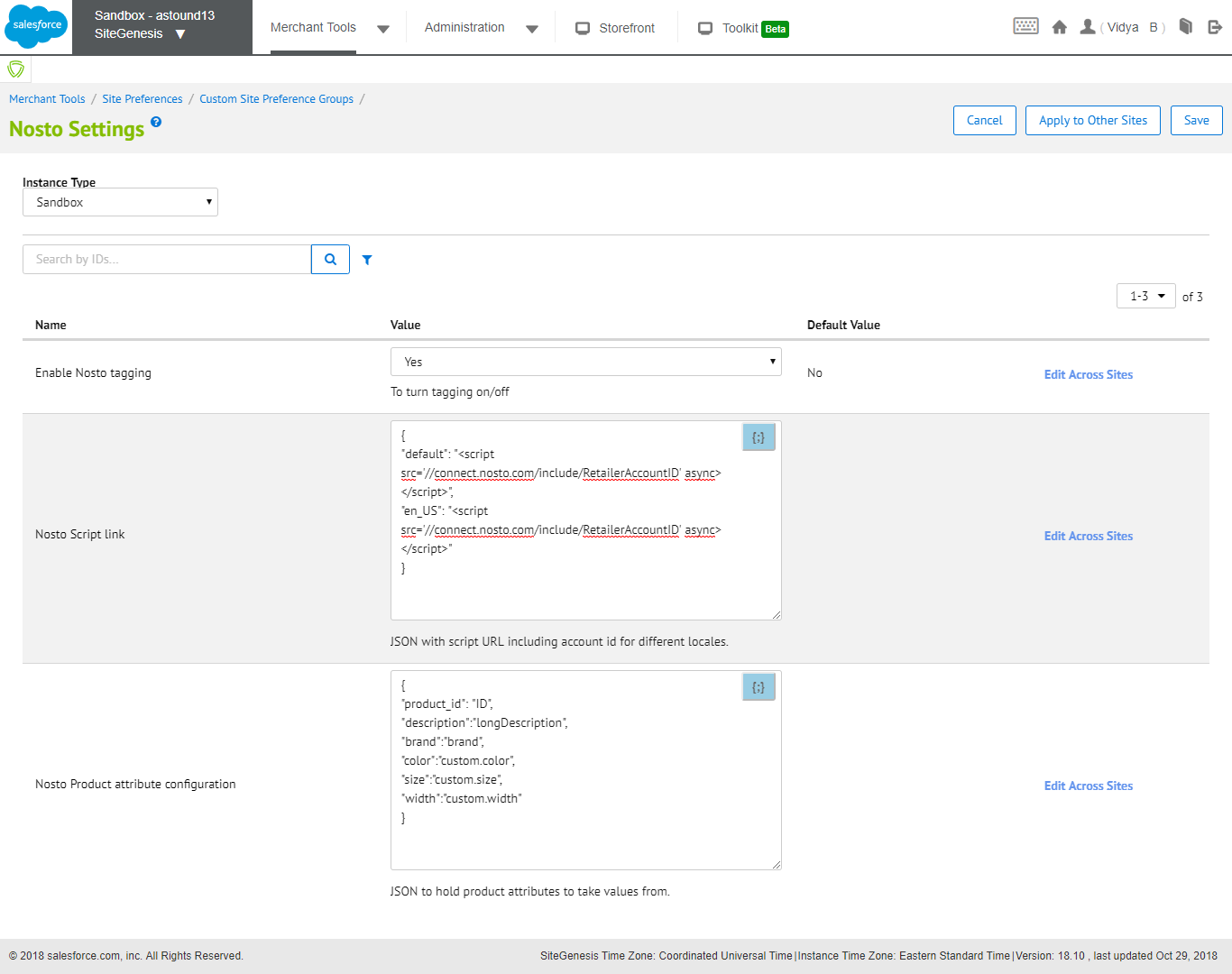
Merchant Tools section is used to manage site specific aspects for the application. Here user can manage site specific configurations of Nosto.

### Custom Preferences

(Merchant Tools > Site Preferences > Custom Preferences)

Custom Preferences are used for several configurations of Nosto cartridge as explained below:





#### Enable Nosto tagging

This site preference is to enable/ disable Nosto functionality at site level

#### Nosto Script link

The complete <script> tag with account ID will be part of site preference. JSON with script URL including account id for different locales.

The site preference value will be added in <head> of every page if the functionality is enabled.

#### Nosto Product attribute configuration

This preference contains product level attributes to take values from and generates tags.

Example:

Below preference will generate <span class=”size”>14</span>

{

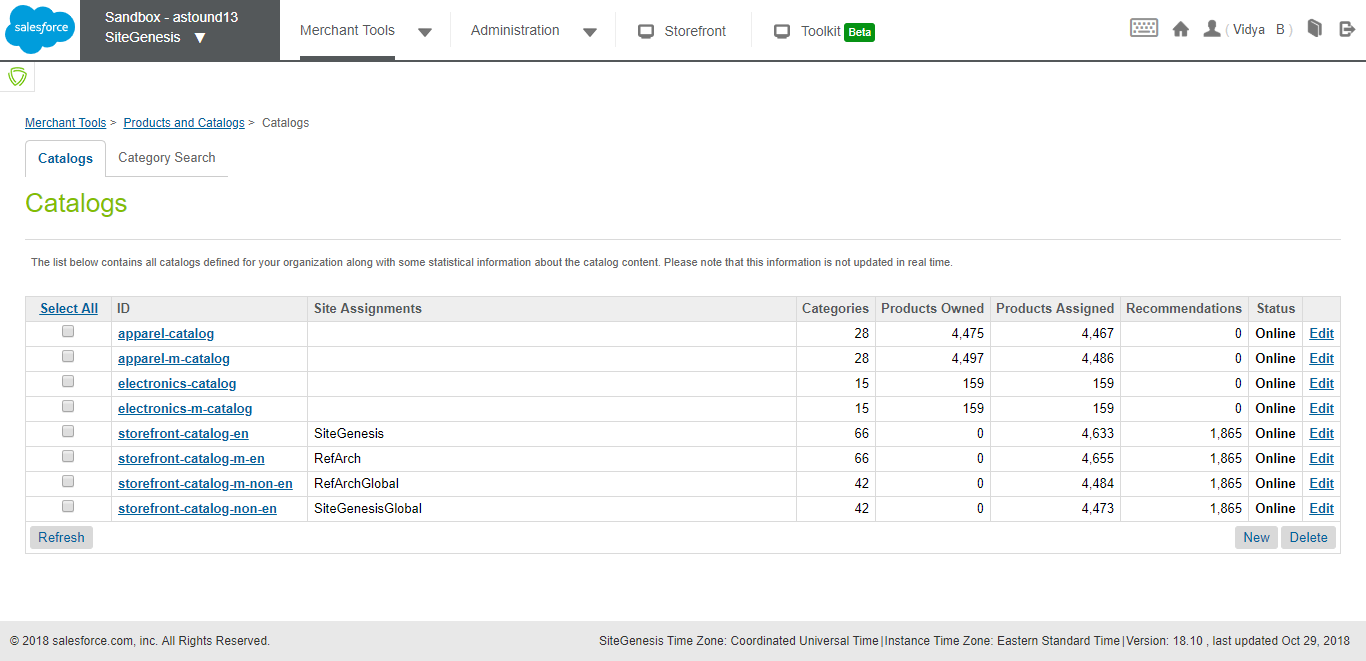
"size":"custom.size"

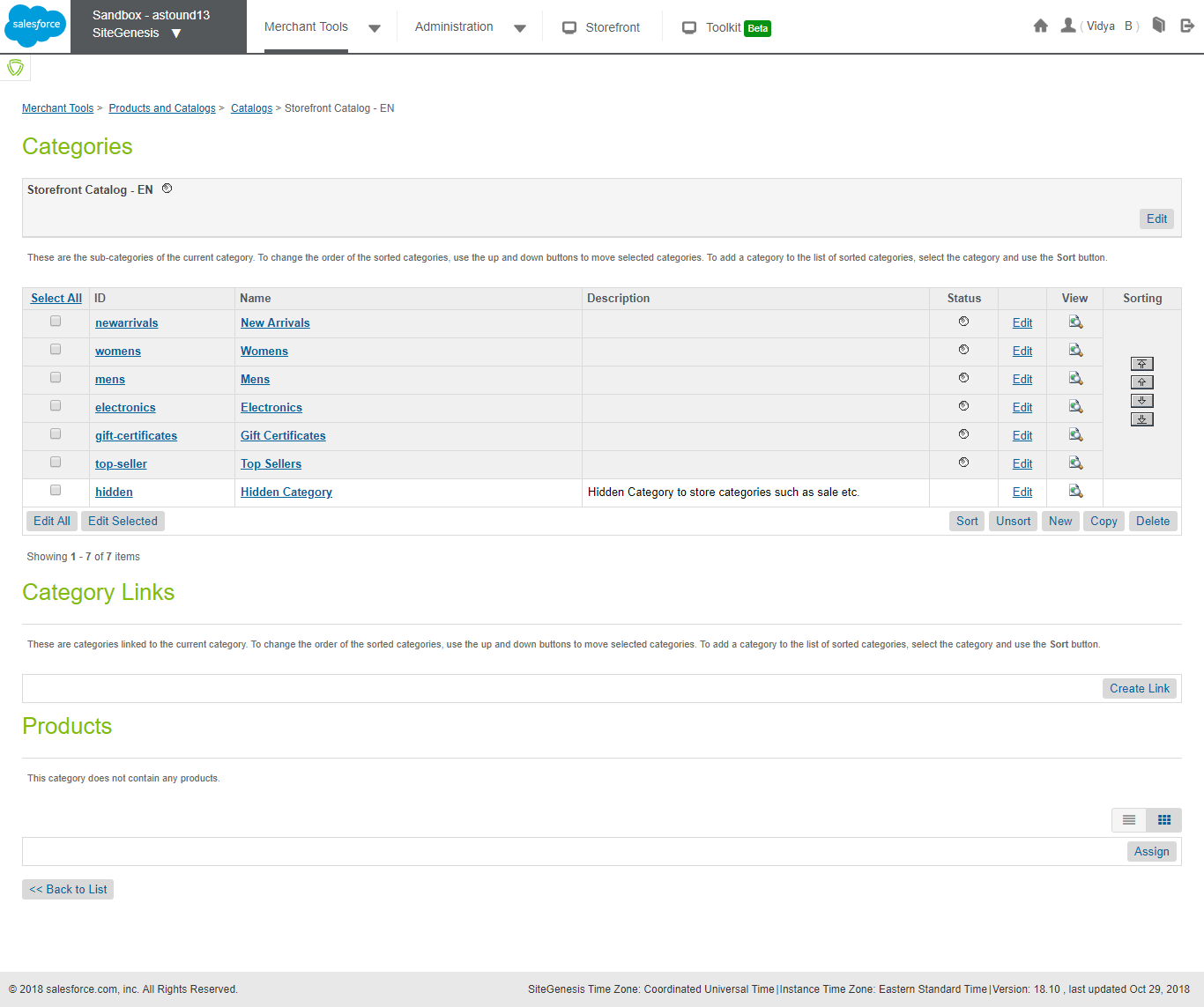
}

### Nosto Brand Configuration (Category Level)

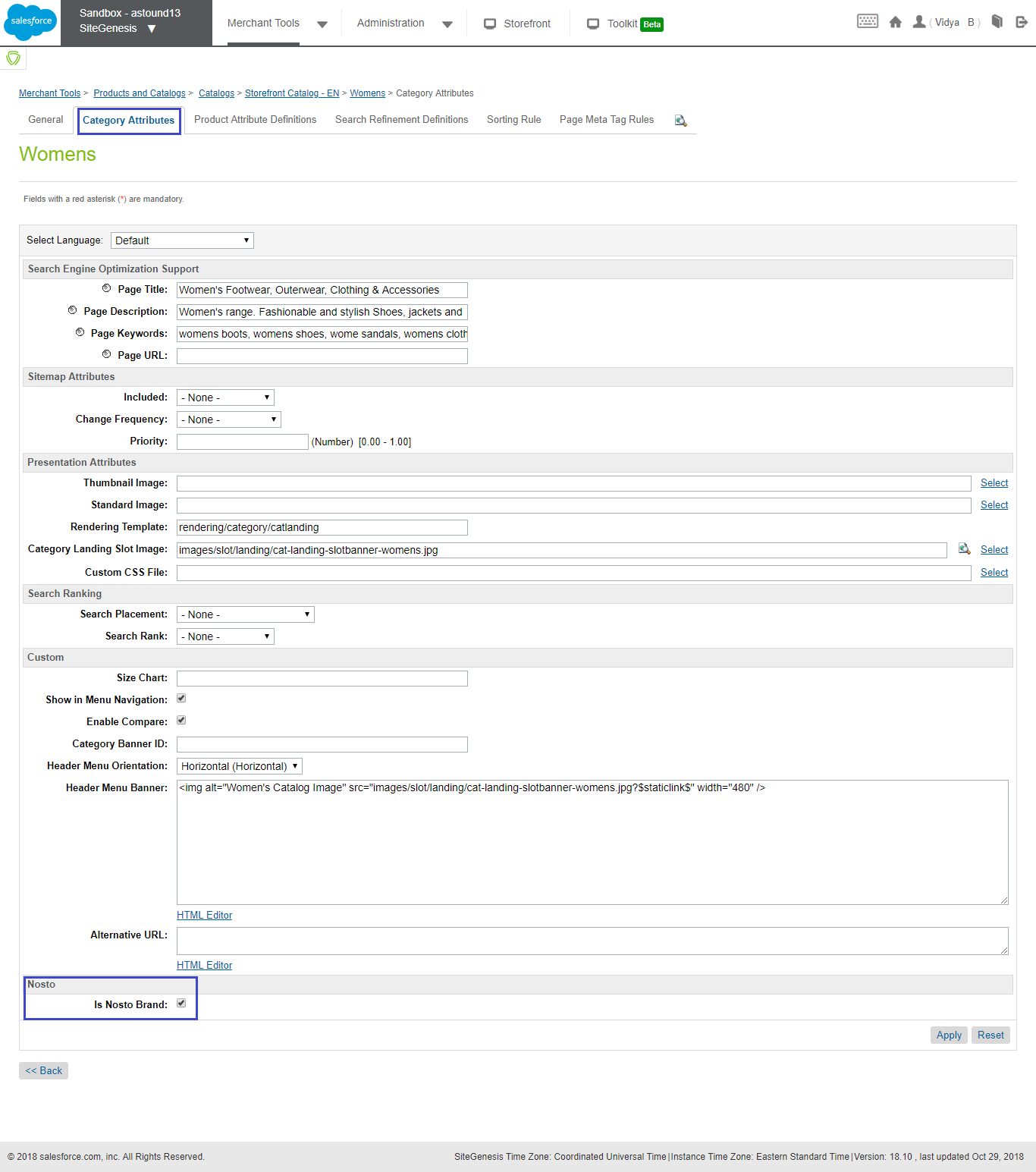
(Merchant Tools > Products and Catalogs > Catalogs)

Navigate to the targeted catalog and click on it to open it in edit mode as shown in below:





Edit targeted category > Open Category Attribute tab

****

In case ‘**Is Nosto Brand’** check box is selected then it will consider respective category as Nosto Brand.

# **Customized Code**

## Controller Implementation

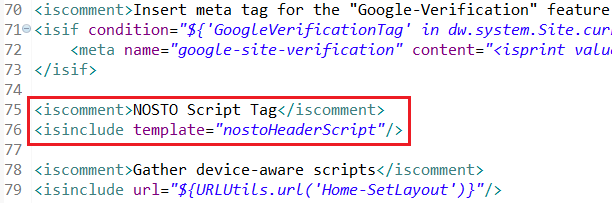
**NOTE**: Path for files given below is based on SG but retailer must make sure they are adding the code in their relevant cartridge.

### **htmlhead.isml**

Path:

SiteGenesis\_core/cartridge/templates/default/components/header/htmlhead.isml

Below code is used to inject the nosto script tags into the header.



<iscomment>NOSTO Script Tag</iscomment>

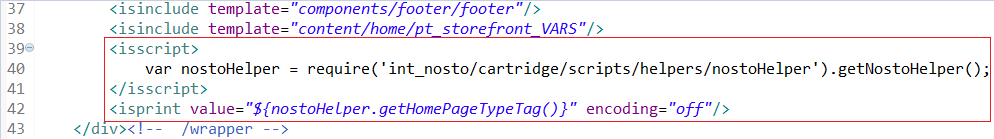
<isinclude template="nostoHeaderScript"/>

### **pt\_storefront.isml**

Path:

SiteGenesis\_core/cartridge/templates/default/content/home/pt\_storefront.isml

Below code is used to inject the nosto page type tag into the homepage.



<isscript>

    var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

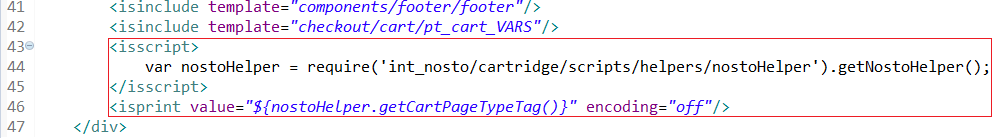
<isprint value="${nostoHelper.getHomePageTypeTag()}" encoding="off"/>

### **pt\_cart.isml**

Path:

SiteGenesis\_core/cartridge/templates/default/checkout/cart/pt\_cart.isml

Below code is used to inject the nosto page type tag into the cart page.



<isscript>

    var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

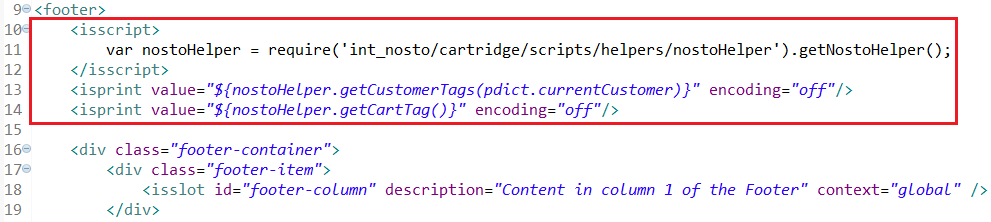
<isprint value="${nostoHelper.getCartPageTypeTag()}" encoding="off"/>

### **footer.isml**

Path:

SiteGenesis\_core/cartridge/templates/default/components/footer/footer.isml

Below code is used to inject the customer and cart tags into the page.



<isscript>

var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

<isprint value="${nostoHelper.getCustomerTags(pdict.currentCustomer)}" encoding="off"/>

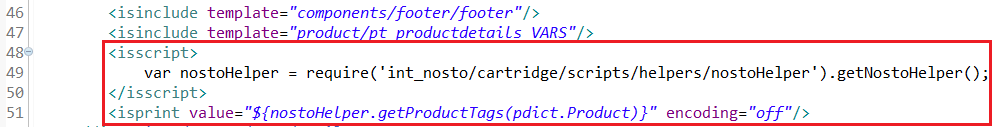
<isprint value="${nostoHelper.getCartTag()}" encoding="off"/>

### **pt\_productdetails.isml**

Path:

SiteGenesis\_core/cartridge/templates/default/product/pt\_productdetails.isml

Below code is used to inject the product tags into the PDP.



<isscript>

var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

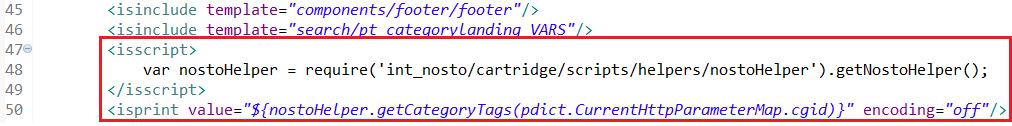
<isprint value="${nostoHelper.getProductTags(pdict.Product)}" encoding="off"/>

### **pt\_categorylanding.isml**

Path:

SiteGenesis\_core/cartridge/templates/default/search/pt\_categorylanding.isml

Below code is used to inject the category tags into the category page.



<isscript>

var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

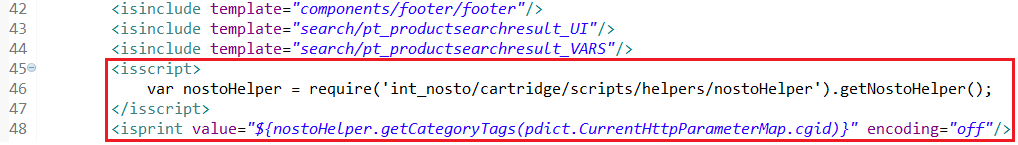
<isprint value="${nostoHelper.getCategoryTags(pdict.CurrentHttpParameterMap.cgid)}" encoding="off"/>

### **pt\_productsearchresult.isml**

Path:

SiteGenesis\_core/cartridge/templates/default/search/pt\_productsearchresult.isml

Below code is used to inject the category tags into the search page.



<isscript>

var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

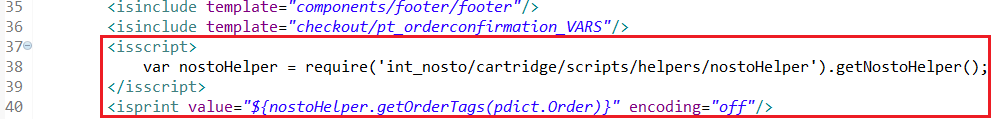
<isprint value="${nostoHelper.getCategoryTags(pdict.CurrentHttpParameterMap.cgid)}" encoding="off"/>

### **pt\_orderconfirmation.isml**

Path:

SiteGenesis\_core/cartridge/templates/default/checkout/pt\_orderconfirmation.isml

Below code is used to inject the order tags into the order confirmation page.



<isscript>

var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

<isprint value="${nostoHelper.getOrderTags(pdict.Order)}" encoding="off"/>

## SFRA Implementation

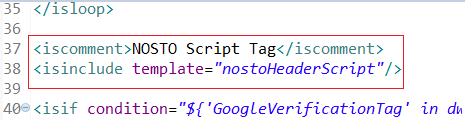
**NOTE**: Path for files given below is based on SFRA but retailer must make sure they are adding the code in their relevant cartridge. (Sample files with below changes are present in int\_nosto\_sfra cartridge for comparison purpose.)

### **htmlHead.isml**

Path:

app\_storefront\_base/cartridge/templates/default/common/htmlHead.isml

Below code is used to inject the nosto script tags into the header.



<iscomment>NOSTO Script Tag</iscomment>

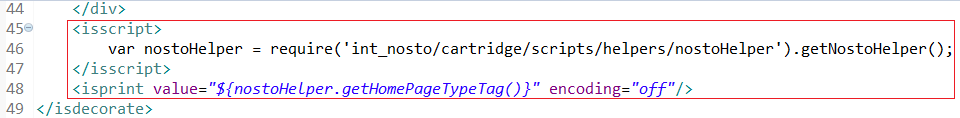
<isinclude template="nostoHeaderScript"/>

### **homePage.isml**

Path:

app\_storefront\_base/cartridge/templates/default/home/homePage.isml

Below code is used to inject the nosto page type tag into the homepage.



<isscript>

var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

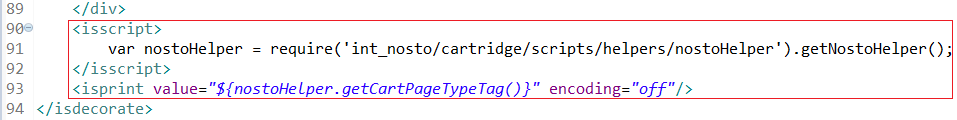
<isprint value="${nostoHelper.getHomePageTypeTag()}" encoding="off"/>

### **cart.isml**

Path:

app\_storefront\_base/cartridge/templates/default/cart/cart.isml

Below code is used to inject the nosto page type tag into the cart page.



<isscript>

    var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

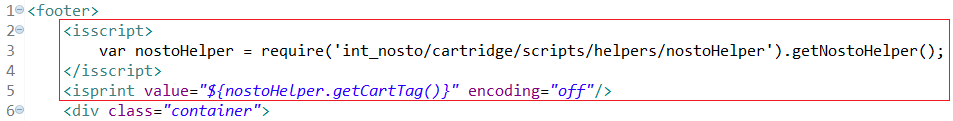
<isprint value="${nostoHelper.getCartPageTypeTag()}" encoding="off"/>

### **pageFooter.isml**

Path:

app\_storefront\_base/cartridge/templates/default/components/footer/pageFooter.isml

Below code is used to inject the nosto cart tags into the footer.



<isscript>

var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

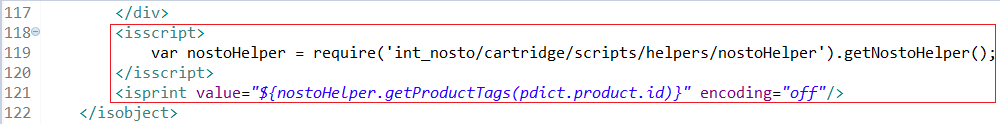
<isprint value="${nostoHelper.getCartTag()}" encoding="off"/>

### **productDetails.isml**

Path:

app\_storefront\_base/cartridge/templates/default/product/productDetails.isml

Below code is used to inject the nosto product tags into the page.



<isscript>

var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

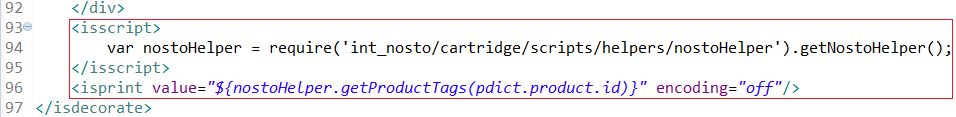
<isprint value="${nostoHelper.getProductTags(pdict.product.id)}" encoding="off"/>

### **bundleDetails.isml**

Path:

app\_storefront\_base/cartridge/templates/default/product/bundleDetails.isml

Below code is used to inject the nosto product bundle tags into the page.



<isscript>

var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

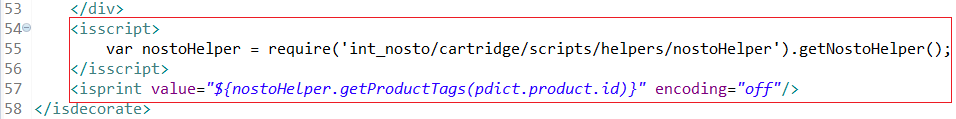
<isprint value="${nostoHelper.getProductTags(pdict.product.id)}" encoding="off"/>

### **setDetails.isml**

Path:

app\_storefront\_base/cartridge/templates/default/product/setDetails.isml

Below code is used to inject the nosto product set tags into the page.



<isscript>

var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

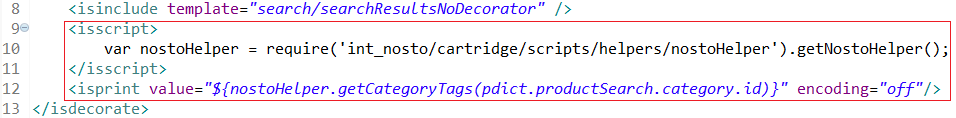
<isprint value="${nostoHelper.getProductTags(pdict.product.id)}" encoding="off"/>

### **searchResults.isml**

Path:

app\_storefront\_base/cartridge/templates/default/search/searchResults.isml

Below code is used to inject the nosto category tags into the category page.



<isscript>

var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

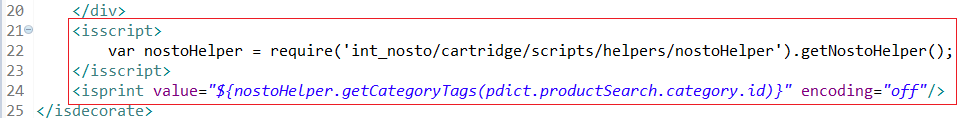
<isprint value="${nostoHelper.getCategoryTags(pdict.productSearch.category.id)}" encoding="off"/>

### **catLanding.isml**

Path:

app\_storefront\_base/cartridge/templates/default/rendering/category/catLanding.isml

Below code is used to inject the nosto category tags into the category page.



<isscript>

var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

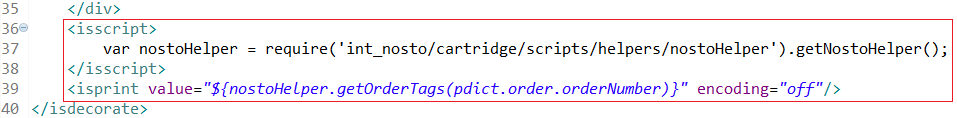
<isprint value="${nostoHelper.getCategoryTags(pdict.productSearch.category.id)}" encoding="off"/>

### **confirmation.isml**

Path:

app\_storefront\_base/cartridge/templates/default/checkout/confirmation/confirmation.isml

Below code is used to inject the nosto order tags into the order confirmation page.



<isscript>

var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

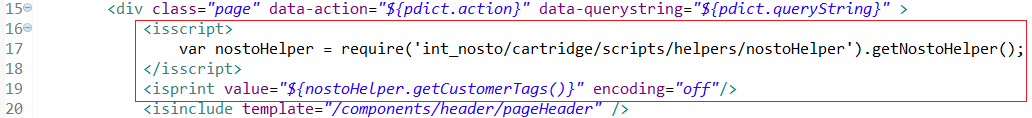
<isprint value="${nostoHelper.getOrderTags(pdict.order.orderNumber)}" encoding="off"/>

### **page.isml**

Path:

app\_storefront\_base/cartridge/templates/default/common/layout/page.isml

Below code is used to inject the nosto customer tags into the page.



<isscript>

var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

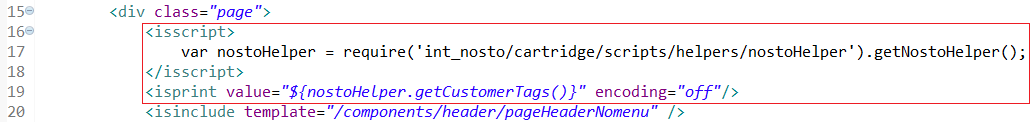
<isprint value="${nostoHelper.getCustomerTags()}" encoding="off"/>

### **checkout.isml**

Path:

app\_storefront\_base/cartridge/templates/default/common/layout/checkout.isml

Below code is used to inject the nosto customer tags into the checkout pages.



<isscript>

var nostoHelper = require('int\_nosto/cartridge/scripts/helpers/nostoHelper').getNostoHelper();

</isscript>

<isprint value="${nostoHelper.getCustomerTags()}" encoding="off"/>