

Extole Integration

*Version 22.1.0*



**Table of Contents**

[Summary](#_30j0zll) **3**

[Component Overview](#_1fob9te) **4**

[Functional Overview](#_3znysh7) 4

[Prerequisites](#_2et92p0) 4

[Limitations, Constraints](#_tyjcwt) 4

[Compatibility](#_3dy6vkm) 4

[Implementation Guide](#_1t3h5sf) **5**

[General Implementation](#_4d34og8) 5

[Configure the Extole API Service Credentials](#_3rdcrjn) 11

[3.2 Open Commerce API (OCAPI) Enablement](#_lnxbz9) 11

[Coupon creation](#_35nkun2) 12

[3.3 Custom Code](#_lf18qofaonrb) 13

[Extole library](#_944r6lt1pwyu) 13

[Extole library](#_79p9ghgnd0ul) 13

[Extole Zone tags](#_3as4poj) 13

[Extole Conversion tag](#_1pxezwc) 14

[3.4 Styling Extole Campaign within Your Storefront](#_1ksv4uv) 15

[3.5 Testing](#_44sinio) 15

[Operations, Maintenance](#_2jxsxqh) **15**

[Data Storage](#_z337ya) 15

[Availability](#_3j2qqm3) 15

[Support](#_1y810tw) 15

[User Guide](#_4i7ojhp) **15**

[Roles, Responsibilities](#_2xcytpi) 16

[Business Manager](#_1ci93xb) 16

[Storefront Functionality](#_3whwml4) 16

[Known Issues](#_2bn6wsx) **16**

[Release History](#_qsh70q) **17**

# Summary

Extole is proud to power referral marketing programs for the world's best-known brands and to help challenger brands become the household names of tomorrow. Extole serves both B2C and B2B companies across a range of verticals, including financial services, retail, and consumer subscription markets. Extole technology powers hundreds of referral programs that help acquire millions of new customers.

This document describes how to implement Extole’s cartridge into the Salesforce Commerce Cloud site. The Extole cartridge is a self-contained cartridge that can easily integrate into any Salesforce Commerce Cloud project. The cartridge can be configured in Business Manager and contains all the elements necessary for you to perform a successful best practices implementation of Extole.

After the cartridge is deployed, configured, and integrated with the storefront templates, you will have all the power of Extole marketing programs applied to your site.

# Component Overview

## Functional Overview

The Extole cartridge integration provides the following functionality:

1. Extole’s core Javascript library.
2. Extole Zone Tags: Zone tags can be placed anywhere on a webpage with the help of a modular ISML template, where the zone tag name is passed as a parameter.
3. Extole Conversion Tag: Javascript tag that passes conversion events to Extole.
4. Extole Approval API: A description of the Extole API Call sub-pipeline and example of integration with ExtoleOrderStatusUpdate job. The cartridge includes the scheduled order status update job, which sends “approve” or “decline” statuses to Extole based on order status (Confirmed, Declined, Failed) after a certain configured time period.
5. OCAPI coupon enablement: By following the integration guide, Extole will be able to read and manage your coupon codes systematically.

### Prerequisites

Before performing the steps in this document, ensure that you and your organization have satisfied the following criteria:

1. You have reached out to Extole and set up an Extole Account (there you can find your Extole Client ID, more details below).
2. You have downloaded and unzipped the Extole cartridge ZIP file from Salesforce Commerce Cloud Marketplace.
   1. The cartridge ZIP file contains the Extole cartridge package, which consists of four folders:
      1. int\_extole
      2. int\_extole\_controllers
      3. int\_extole\_sfra
      4. Metadata folder (contains Extole’s custom site preferences)

## Limitations, Constraints

*No limitations or constraints.*

## Compatibility

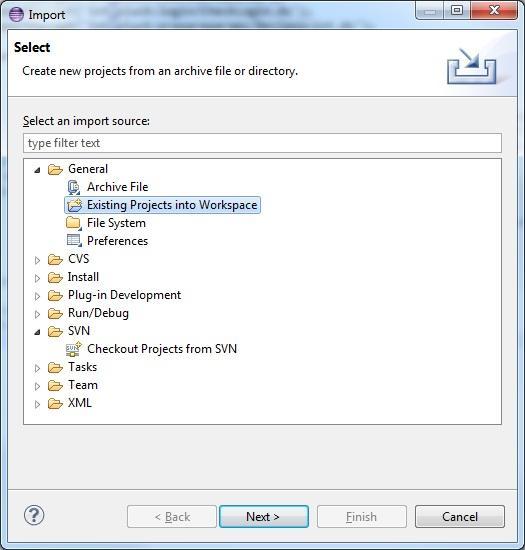
The cartridge is designed for Salesforce Commerce Cloud API version 22.4 (Compatibility Mode: 21.7) with SFRA version 6.0.0. The cartridge is designed with US locale but has multi-locale and multi-currency support.

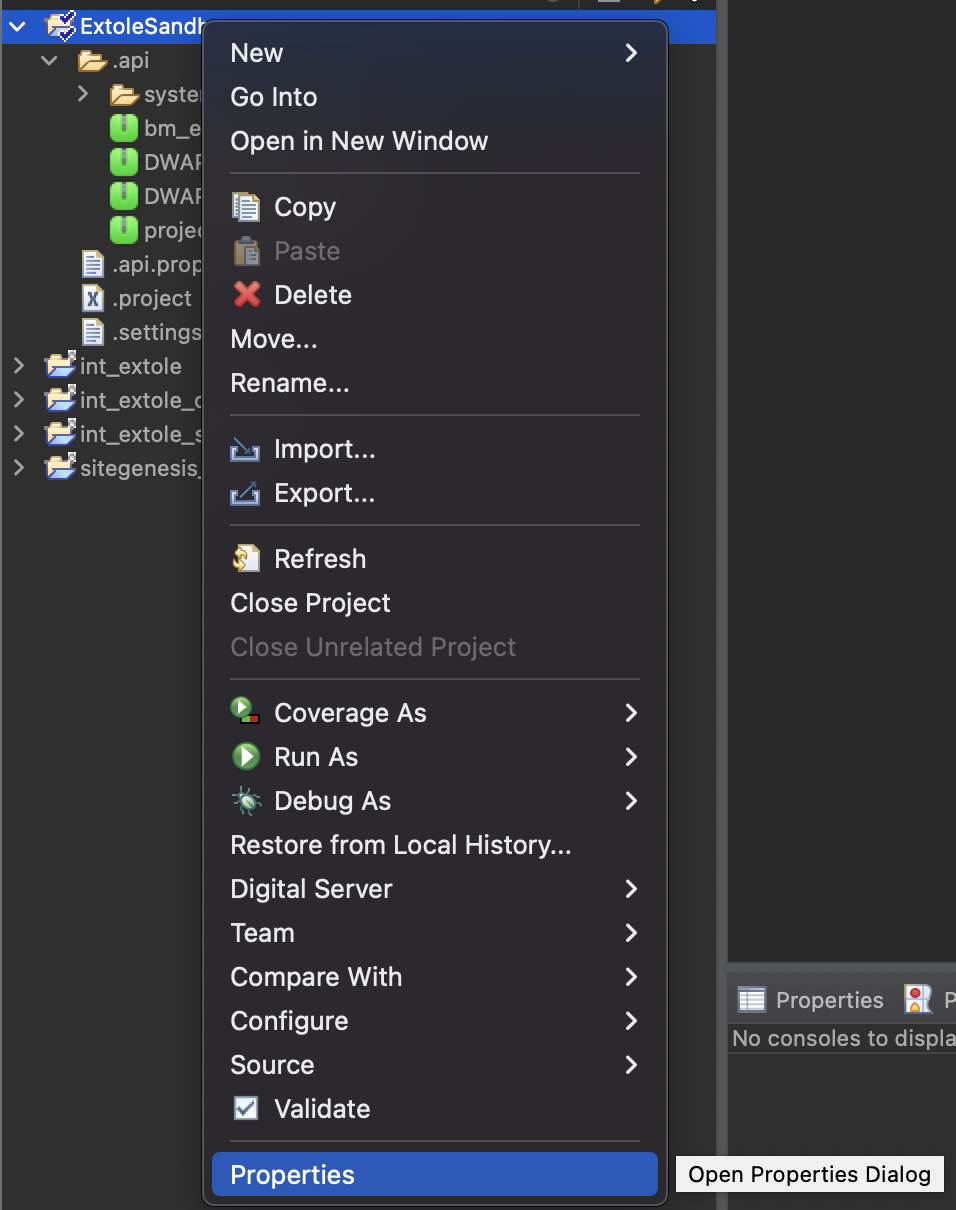
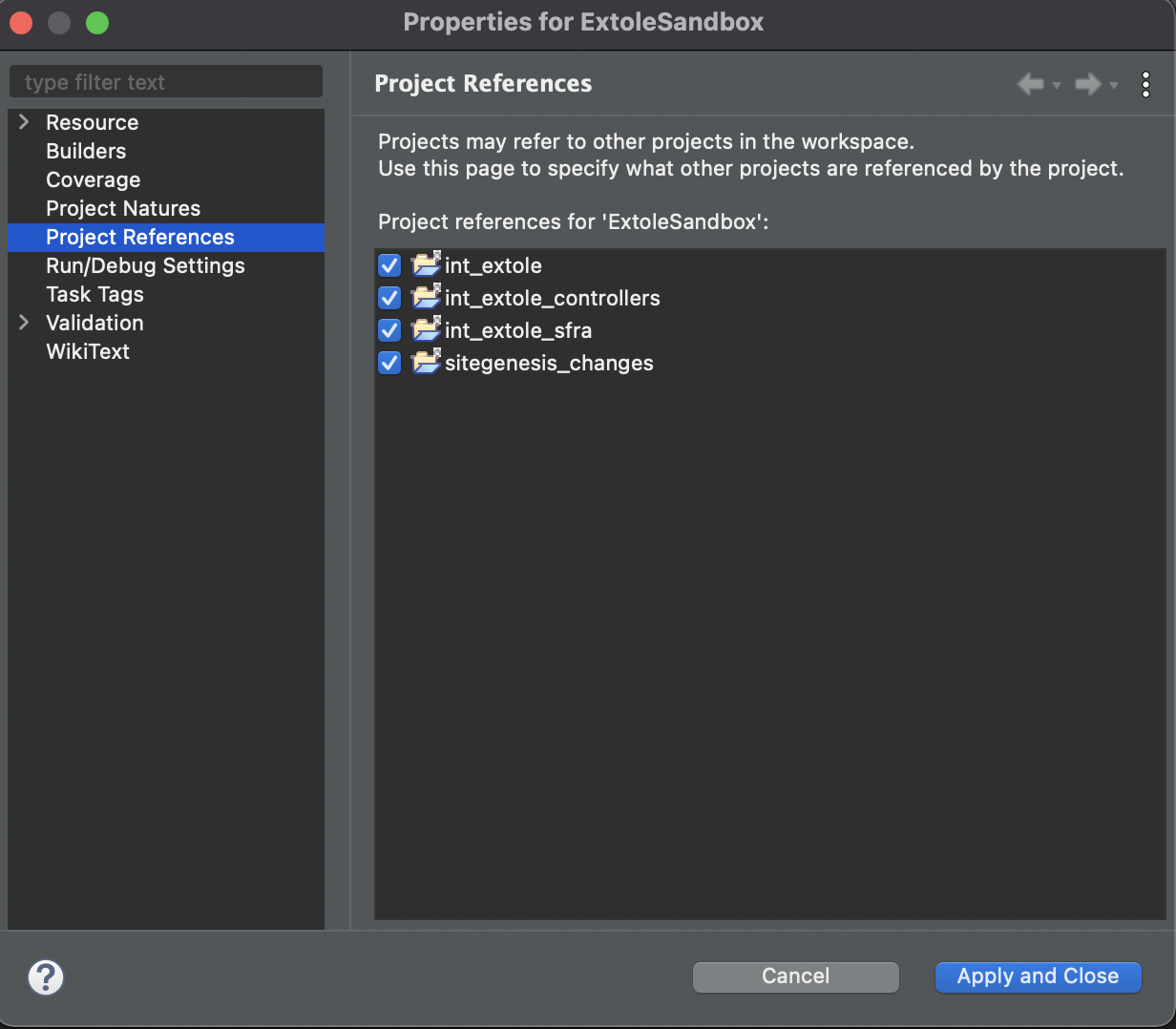
# *Implementation Guide*

## General Implementation

### Installing the Cartridge on sandbox

1. Open Salesforce Commerce Cloud Studio.
2. Import the downloaded cartridge ZIP file. On the import screen, select the folders **int\_extole** and **int\_extole\_sfra.**

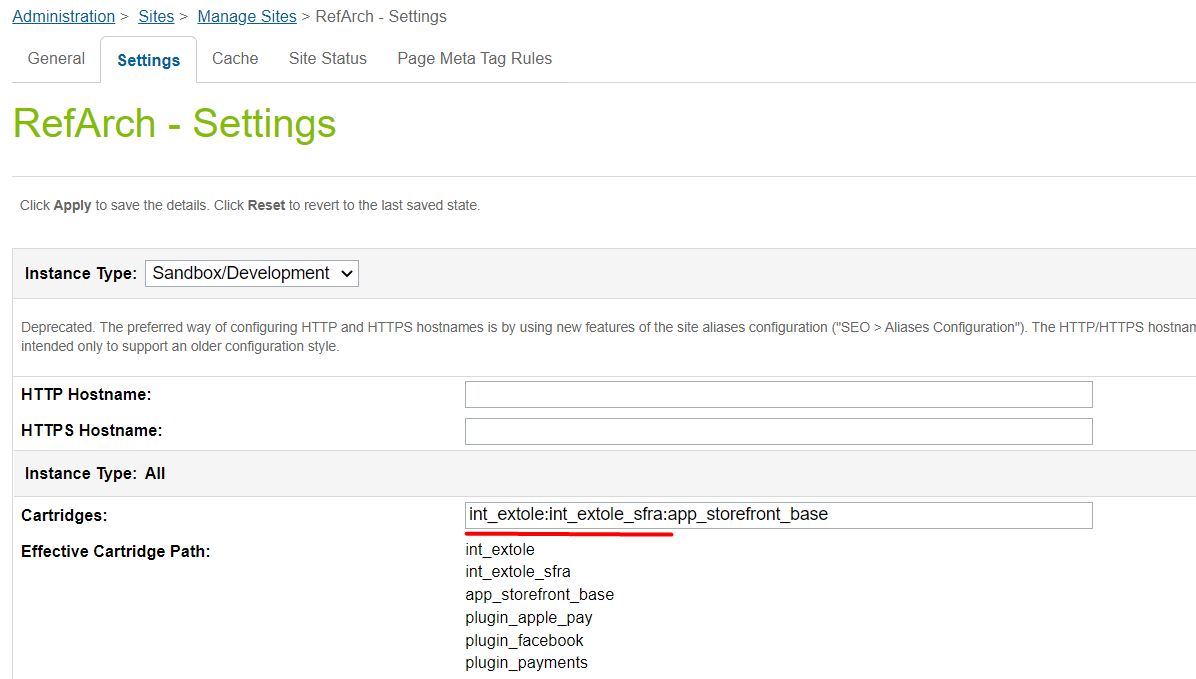
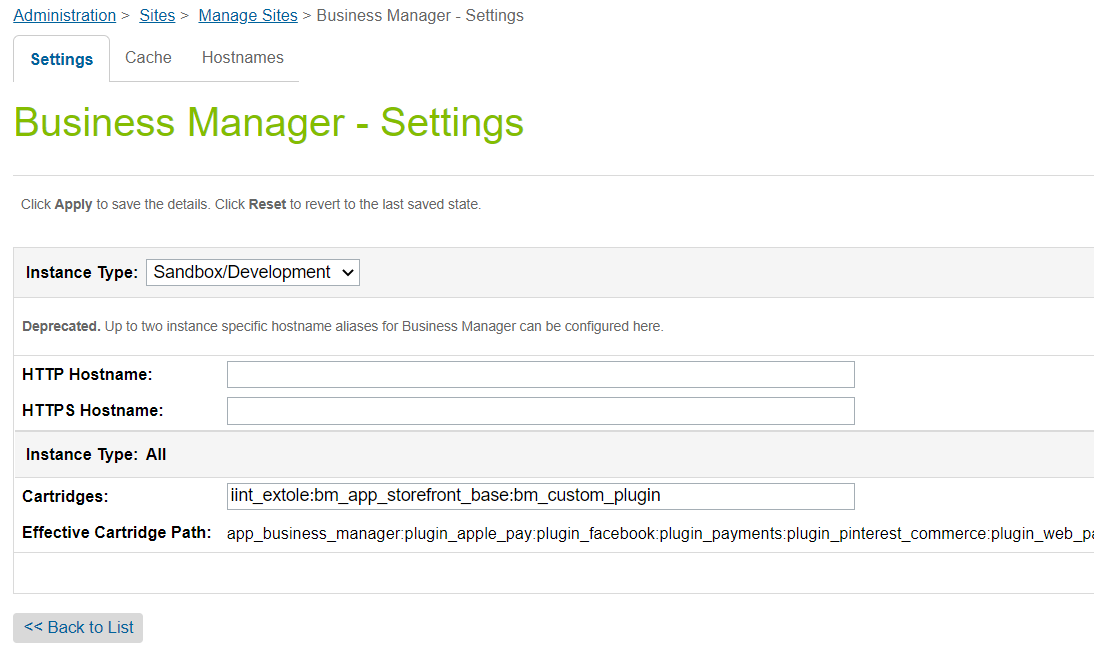
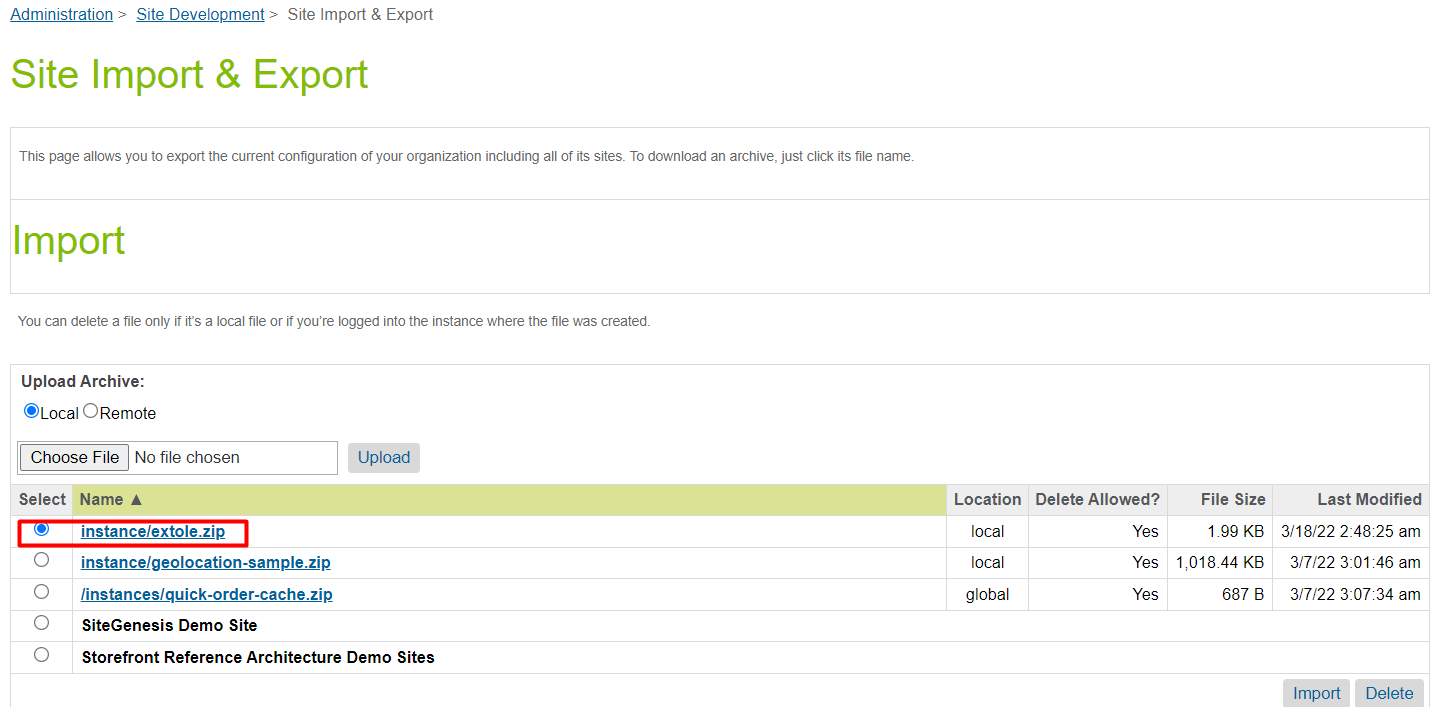


1. Link the cartridge to your sandbox. Right click on the **ExtoleSandbox** and select **Properties** from the dropdown menu. 
2. In the **Properties** menu, select **Project References** and check **int\_extole** and **int\_extole\_sfra**.

### Extole API Credentials

Please reach out to your Extole team to get your Extole API credentials: User ID and Password data.

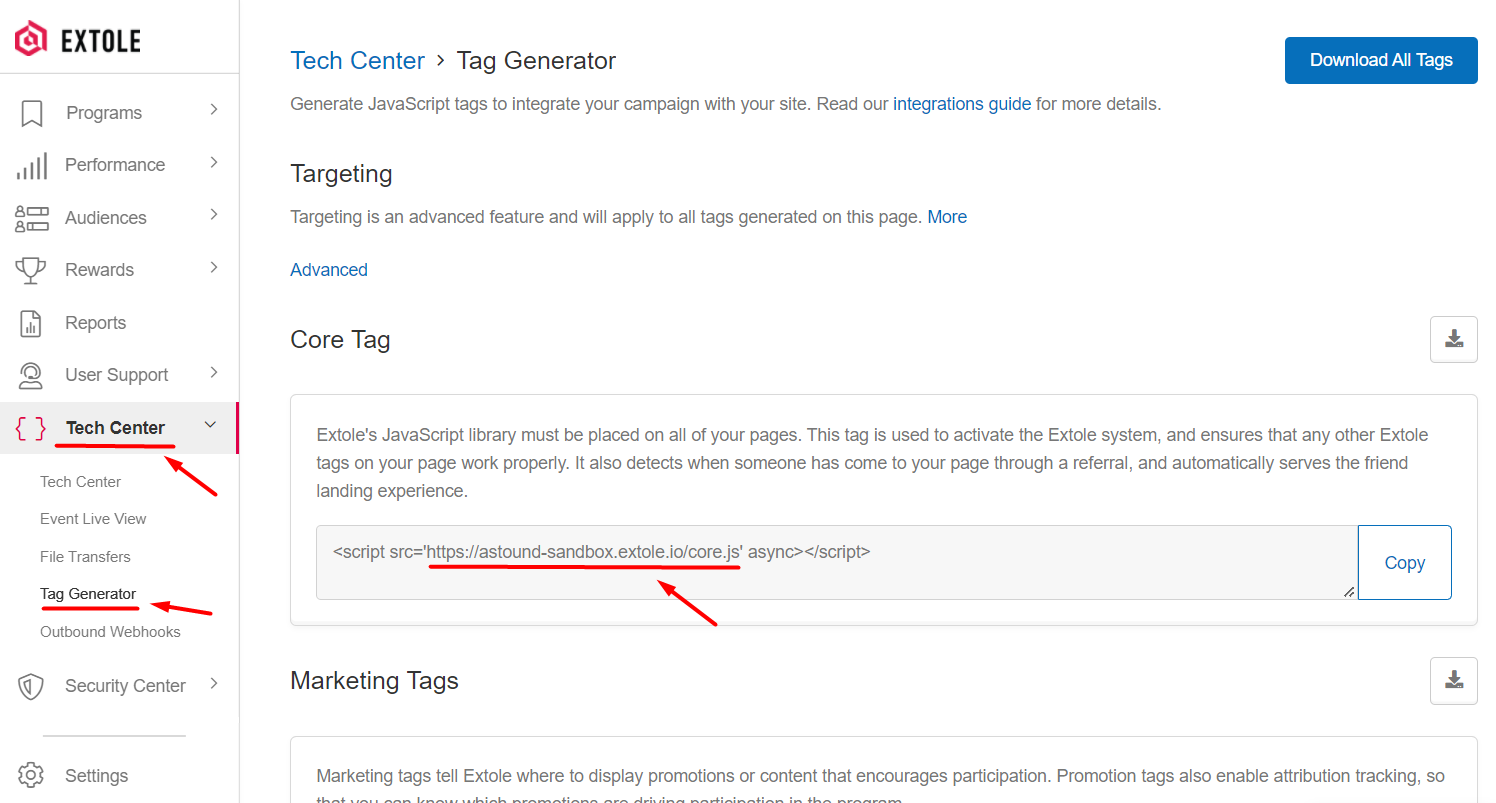
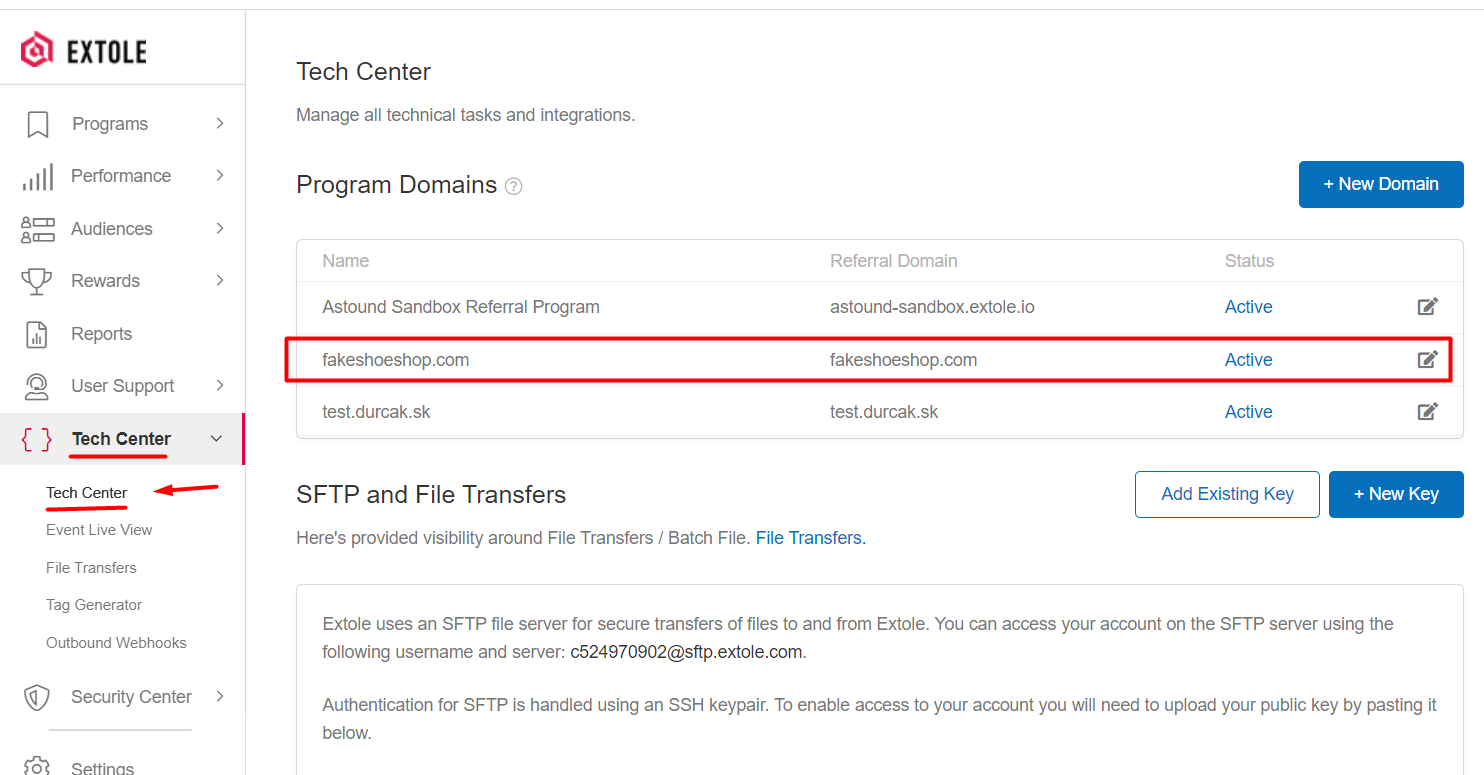
### Business Manager Setup

1. Go to **Business Manager** and open Administration > Sites > Manage Sites. Select your site, then select the **Settings** tab. In the cartridge path, at the beginning, add **int\_extole:int\_extole\_sfra:**
2. Navigate to Administration > Sites > Manage Sites and select **Business Manager** site. In the cartridge path, at the beginning, add **int\_extole:**  
   
3. Open Administration > Site Development > Site Import & Export.
4. Under the Upload Archive, select Local, then choose the **extole.zip** file and click the Upload button. All metadata, service, and job configurations will be imported.

### Extole Site Preferences Setup

In order to properly configure the Extole site preferences, you will need several pieces of information from your **Extole Account**. To get started, open your [Extole Account](https://my.extole.com/login). In a new browser window, pull up **Business Manager** and navigate to Sites > Your Site > Site Preferences > Custom Site Preferences > Extole.

#### Extole Core URL

1. Once you’ve logged into your [Extole Account](https://my.extole.com/login), navigate to the **Tech Center** in the left-hand menu.
2. From the dropdown options, click on **Tag Generator**.
3. Find the **Core Tag** and copy the URL.
4. Go to your other browser window with **Business Manager** open, and paste the URL into the Core Tag URL field of the Extole Custom Site Preferences page.
5. If you want to have the tag URL on a page with the same domain as your storefront (so it won’t be affected by third-party ad blockers), you will need to set up a **CNAME** for the core tag domain. To do so, return to your Extole Account and click on the **Tech Center**. Follow the instructions in [Extole’s Program Domain Setup Guide](https://success.extole.com/hc/en-us/articles/115012548127).
   1. **Example**: For the site URL **https://fakeshoeshop.com/**, the share experience microsite would have a URL like **https://share.fakeshoeshop.com/**, and the core tag would have the following URL: **https://share.fakeshoeshop.com/core.js**.

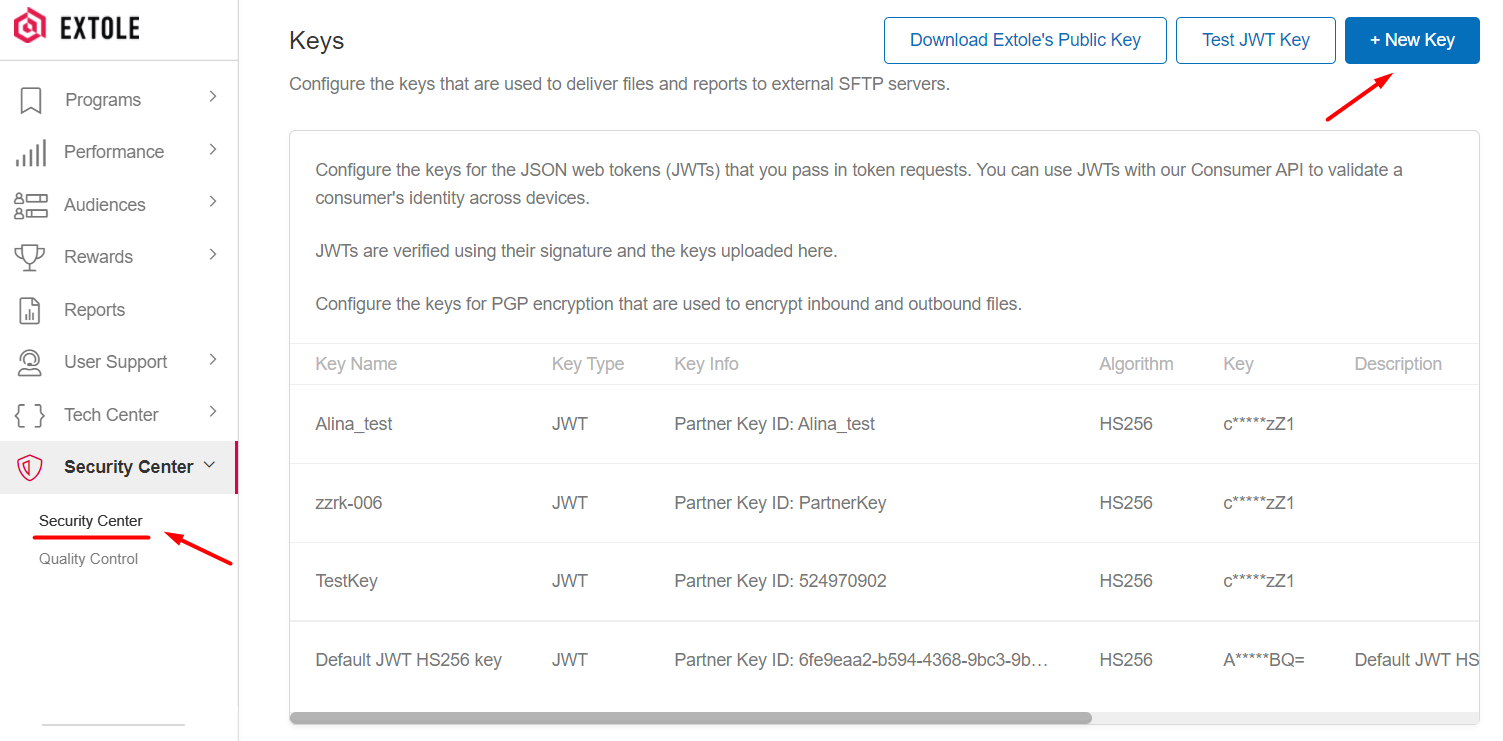
#### 

#### Extole Access Token

1. Into your Extole Account, click on **Security Center** in the left-hand menu.
2. Click on the **+ New Access Token** button. You will be asked to verify your identity. Once you have done so, name the key appropriately and hit Create.
   1. **Note**: Make sure to copy the token immediately. Once it’s created and you navigate away from this page, it will disappear.
3. Return to your other browser window with **Business Manager** open, and paste the token into the **API Access Token** field of the **Extole Custom Site Preferences** page. Graphical user interface, text, application

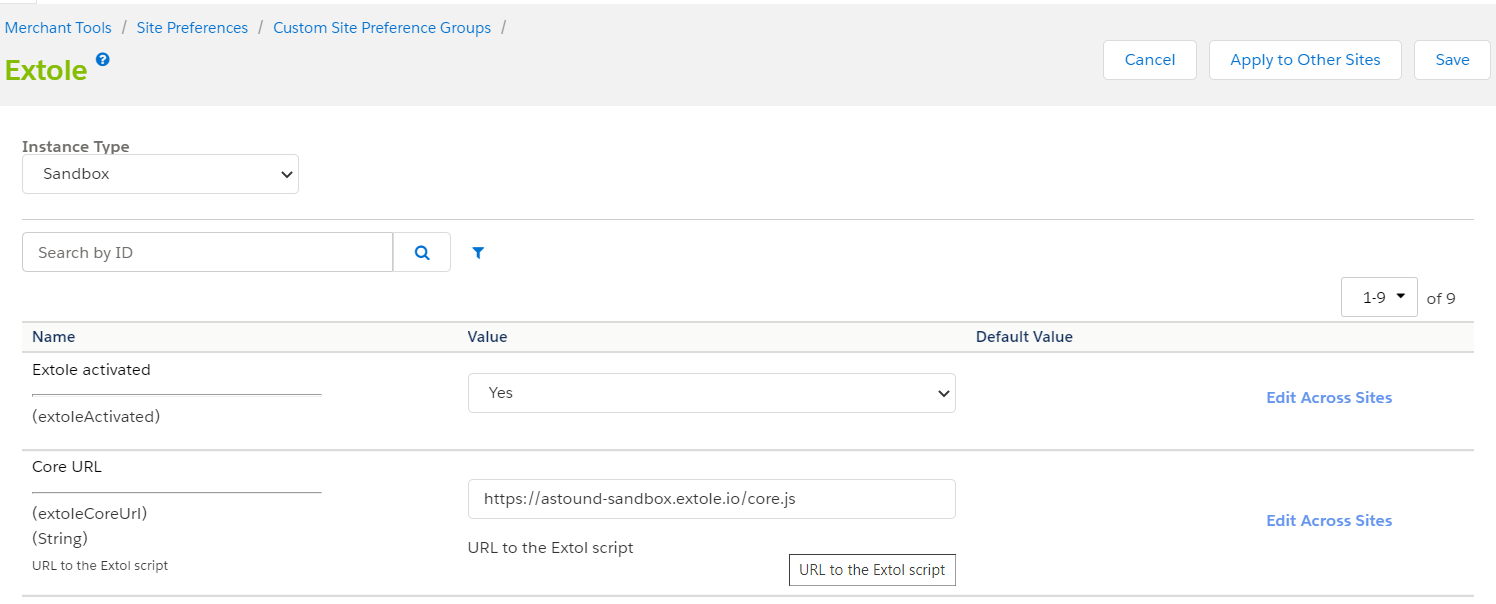
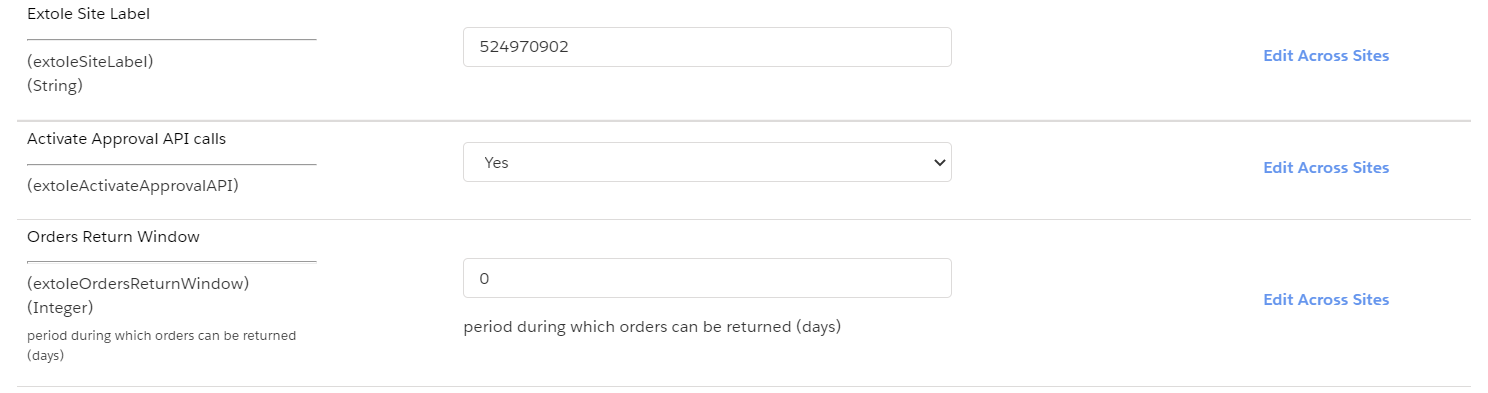
   Description automatically generated

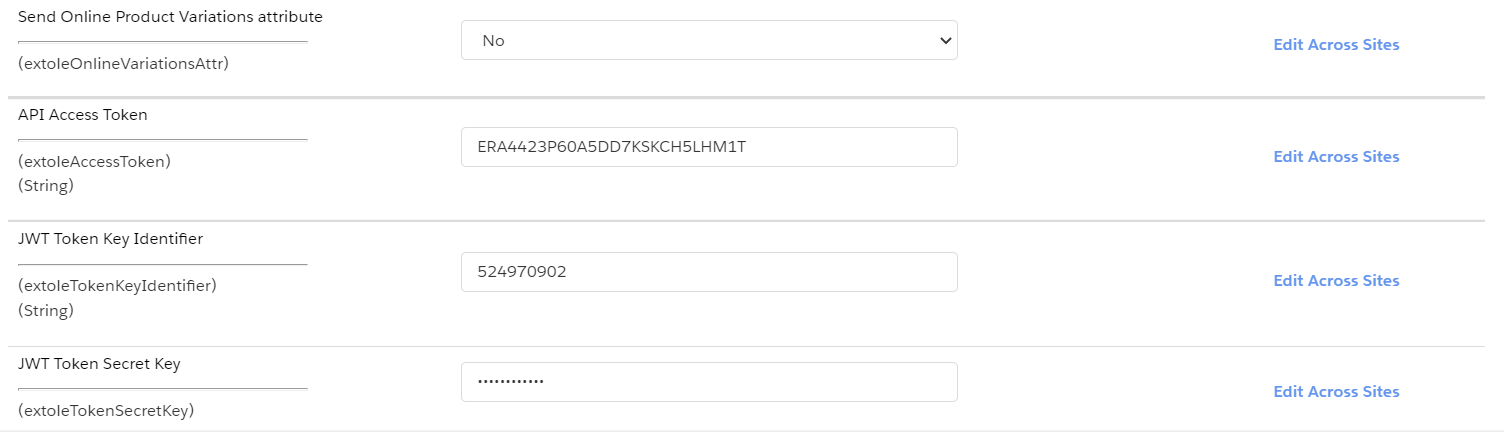
#### Extole JWT Token Key Identifier and Token Secret Key

1. In your Extole Account, click on **Security Center** in the left-hand menu.
2. Click the **+ New Key** button.
3. On the key creation page, fill in the necessary fields.
   1. **Key Name**: The Key Name is used as a reference to the Access Key in your Extole Account.
   2. **Key Type**: JWT.
   3. **Algorithm**: HS256.
   4. **Partner Key ID**: The unique identifier for the key. This is the same as the “kid” in the token header.
      1. Copy the Partner Key ID value and paste it into the **JWT Token Key Identifier** field on the **Extole Custom Site Preferences** page of **Business Manager**.
   5. **Key**: The key itself (32 or more characters in length).
      1. Copy the Key value and paste it into the **JWT Token Secret Key** field on the **Extole Custom Site Preferences** page of **Business Manager**.

#### Configure Remaining Custom Site Preferences

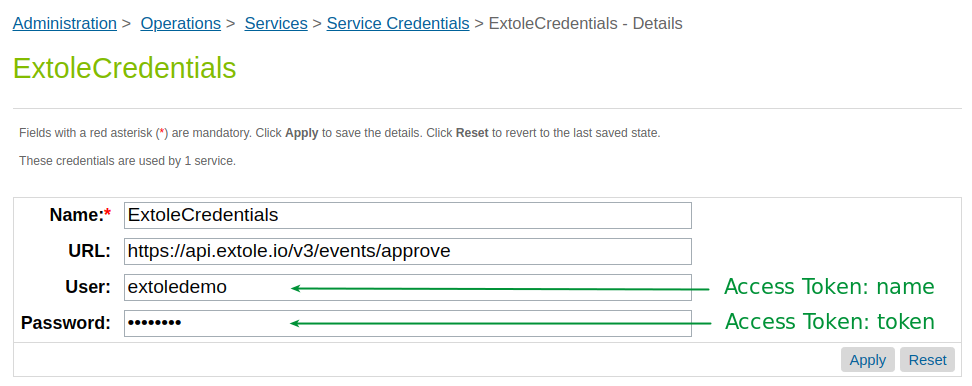
After collecting all of the necessary information from your **Extole Account** and pasting the data in the proper fields, make sure to configure all of the remaining fields properly on the **Extole Custom Site Preferences** page of **Business Manager**.

1. Make sure your **Instance Type** is set to **Sandbox**.
2. For the **Extole activated** option, select **Yes** from the dropdown menu.
3. Enter your **Site Label**, which is your Extole Client ID. You can find your Client ID from the Settings menu of your Extole account.
4. For the **Activate Approval API Calls** option, select **Yes** from the dropdown menu.
5. Set your **Orders Return Window** to the number of days you allow orders to be returned after purchase.
6. For the **Send Online Product Variations attribute**, select **No** from the dropdown menu.



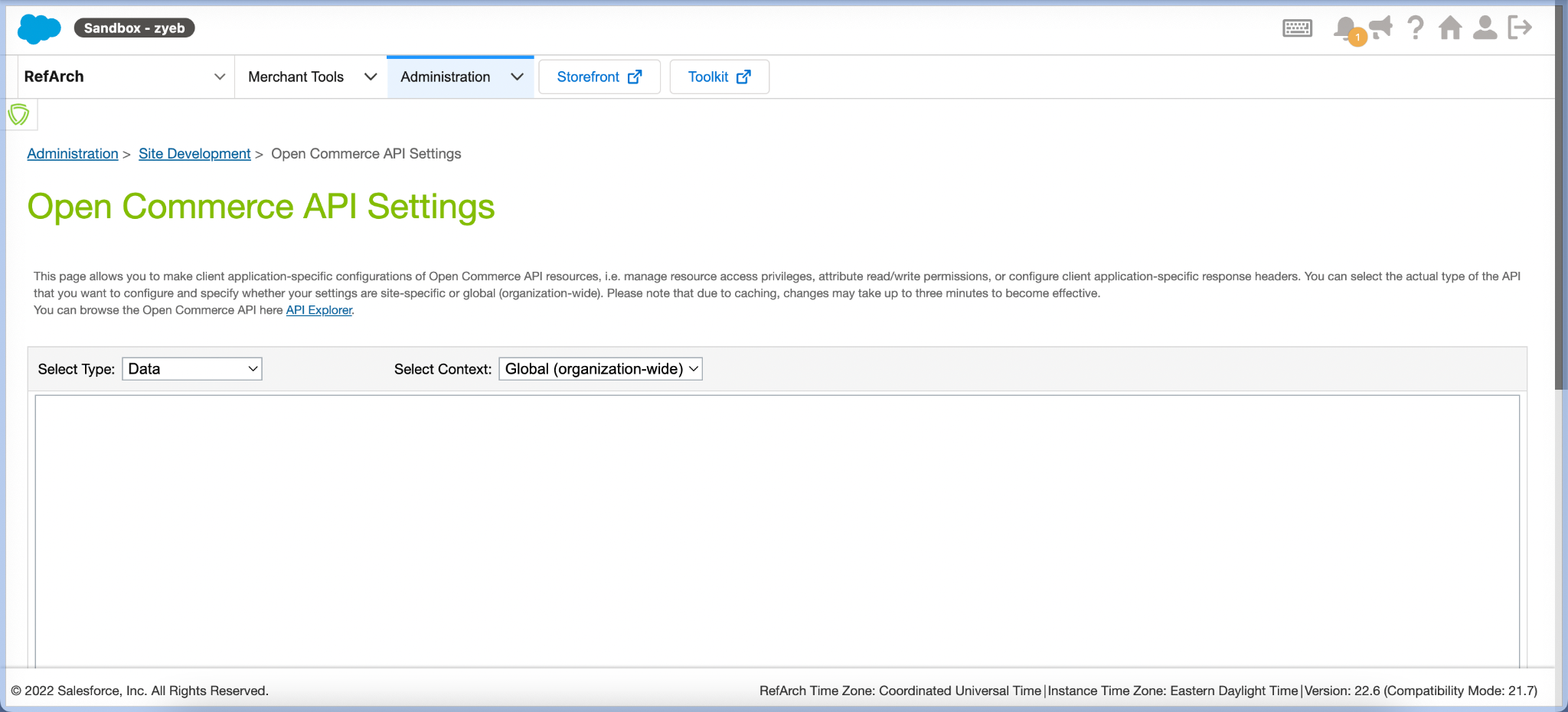
### Configure Extole API Service Credentials

The current cartridge uses the advantages of the Extole API V3.

1. Open Administration > Operations > Services > Service Credentials > ExtoleCredentials.
2. Type in a name for your credentials in the required **Name** field.
3. Paste in the URL for the Extole Events API endpoint: <https://api.extole.io/v3/events/approve>.
4. Enter the Access Token **Name** and **Token** values you created in the Extole Tech Center in the **User** and **Password** fields.

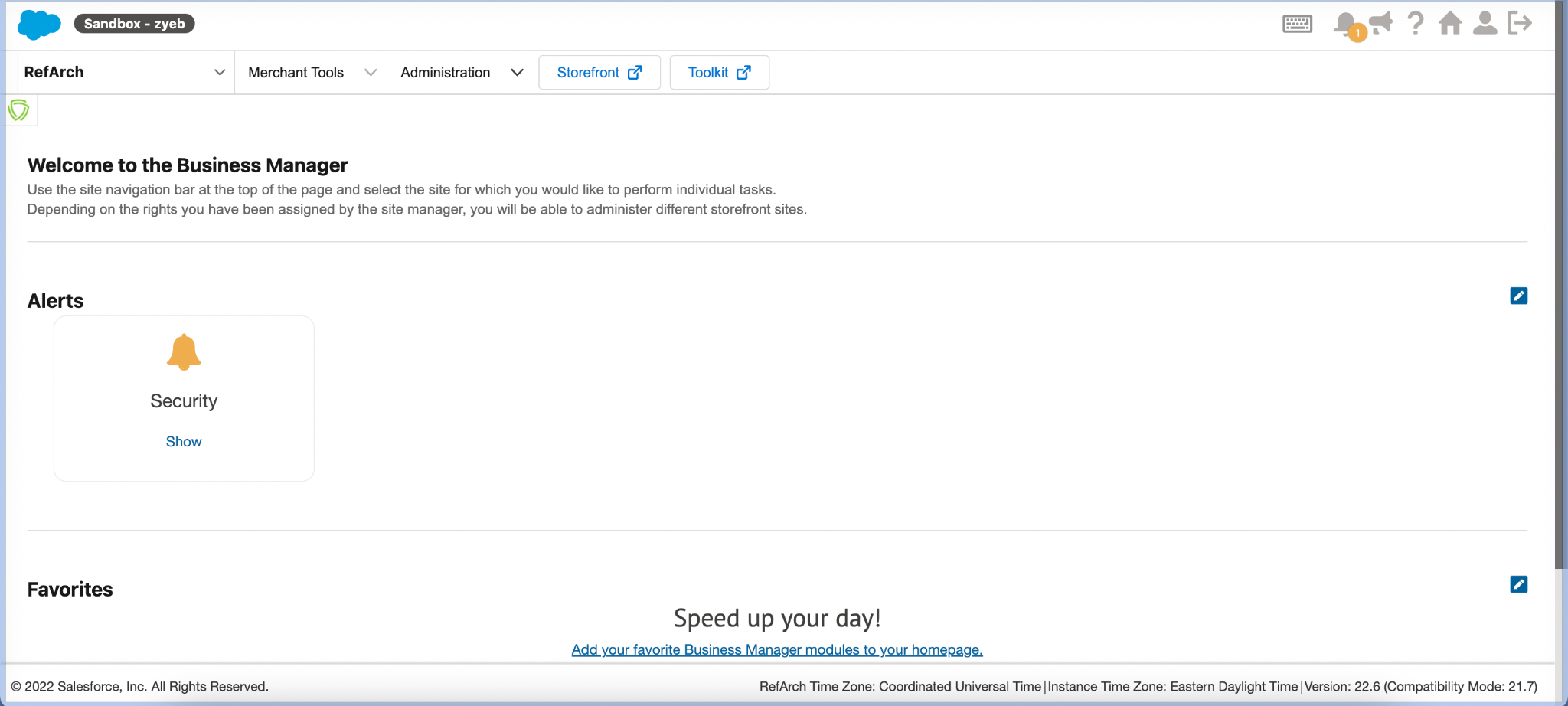
## 3.2 Open Commerce API (OCAPI) Enablement

Extole can programmatically manage coupon codes within the Salesforce Commerce Cloud system by interfacing with OCAPI. The following configurations in **Business Manager** are required.

1. Go to Business Manager > Administration > Site Development > Open Commerce API Settings.
2. From the Select Type dropdown, choose **Data**. Leave the Select Context dropdown set to **Global (organization-wide)**.
3. Update your **OCAPI settings** to include the clients section below (you may need to amend your settings), which will enable Extole to create, read, and update coupons.

| {  "\_v": "18.3",  "clients": [  {  "allowed\_origins": [  "http://my.extole.com",  "https://my.extole.com",  "http://api.extole.com",  "https://api.extole.com"  ],  "client\_id": "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa",  "resources": [  {  "resource\_id": "/sites/**{site\_id}**/coupons/\*\*",  "methods": ["get", "post","patch"],  "read\_attributes": "(\*\*)",  "write\_attributes": "(\*\*)",  "cache\_time": 900  },  {  "resource\_id": "/sites/**{site\_id}**/coupon\_search",  "methods": ["post"],  "read\_attributes": "(\*\*)",  "write\_attributes": "(\*\*)",  "cache\_time": 900  }  ]  }  ]  } |
| --- |

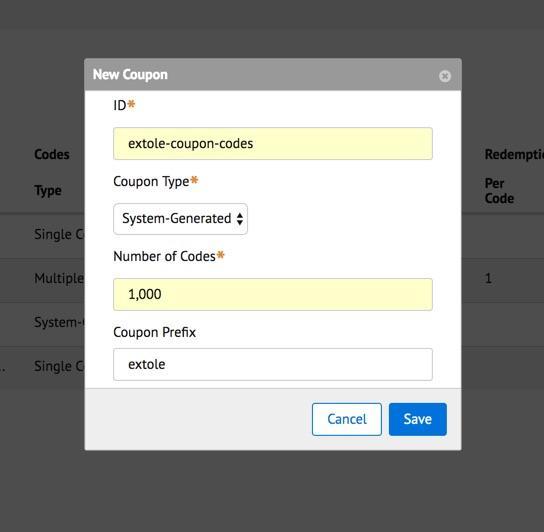
* 1. Make sure to replace **{site\_id}** with your actual Site ID. You can find your Site ID in the far-left corner of the Business Manager toolbar (e.g., see below: **RefArch**).



* 1. For the purposes of this document, test credentials were used. You will need to generate a new API Client ID in **Account Manager** and update these settings within each environment.

### 

### Coupon Creation

1. In **Business Manager**, navigate to Merchant Tools (desired Site) > Online Marketing > Coupons.
2. In the upper right, click **New**.
3. Configure your New Coupon.
   1. Type in an **ID** to identify your coupons.
   2. Select the **Coupon Type** “System-Generated”
   3. Enter the **Number of Codes** you would like.
   4. Add a **Coupon Prefix** (extole).
4. In your **Extole Account**, navigate to **Rewards** and click on the **+ New Reward** button.
5. Select **Salesforce Commerce Cloud Coupon** from the available coupon types and make sure to input the **Coupon ID** you configured in **Business Manager**.
6. Request help from your implementation manager if you need help with this action.

## 3.3 Custom Code

### Extole Javascript Library

The Extole Javascript library should preferably be embedded in the <header> tag on all pages of your website.

Paste the following code sample into your htmlHead.isml:

| <iscomment>Extole Library</iscomment>  <isinclude template=*"extole/core"*/> |
| --- |

Include the extole.js script file in your common/scripts.isml template:

| <script defer type="text/javascript" src="${URLUtils.staticURL('/js/extole.js')}"></script> |
| --- |

### Extole Zone Tags

Extole zone tags are placeholders that can be embedded anywhere on your site. Once embedded, zone tags serve program content you’ve configured in the Extole Campaign Editor, including banners, CTAs, pop-ups, overlays, and so on.

An example of a common tag placement is a global header. In this case, you would add the zone tag to the header of every page on your site.

If the zone tag you want to use doesn’t need to be placed in a specific location (like for a popup), then the tag must be placed at the bottom of your isml template.

Zone tags can pass additional information such as product, category, or customer data within the ‘data’ parameter to personalize your customers’ sharing experience.

#### Recommended Zone Tags & Placements

| Zone Tag Name | Placement |
| --- | --- |
| global\_header | Place in the header of the site, all pages. |
| global\_footer | Place in the footer of the site, all pages. |
| homepage, homepage\_module | Place in homepage banner containers |
| category | Place in all categories on product listing pages (PLP). |
| product | Place on all product details pages (PDP). |
| social | Place in the PDP social sharing buttons area. |
| my\_account | Place on the My Account page. |
| embedded\_stats | Place on the My Account page. |
| registration | Place on the My Account page, but only when a customer first registers. |
| confirmation | Place on the order confirmation page. |

Each tag can be customized, meaning you can specify what information needs to be added to the tag. For example, take a look at the zone tag integration below:

| <iscomment>  Extole zone tag example  </iscomment>  <isinclude url="${URLUtils.url('Extole-Zonetag',  'name', 'global\_header',  'elementId', 'extole\_zone\_global\_header',  'predefined\_data', 'customer, product',  'mapped\_data', 'brand: product.brand',  'data', 'name\_test: value\_test',  'productId', pdict.product.id,  'categoryId', pdict.productSearch.category.id,  'orderId', pdict.CurrentRequest.httpParameterMap.orderID.value,  'orderToken', pdict.CurrentRequest.httpParameterMap.orderToken.value  )}" /> |
| --- |

#### Zone Tag Parameters

In the above example, ***'Extole-Zonetag'*** is the name of the controller that renders the tag on the page, which is followed by the tag parameters. Tag parameters have the format ***‘parameter name1’, ‘value1’, ‘parameter name2’, ‘value2’****.*

| Parameter | Value |
| --- | --- |
| name (required) | The name of the tag, such as global\_header. See the list of valid tag names above. |
| elementId (required) | The Extole popup element ID. |
| predefined\_data (optional) | A comma-separated string. It could include objects like *customer, category, product, order, jwt*.  Predefined data configuration is stored in the file **'int\_extole/cartridge/scripts/extole/defaultConfig.json'**. |

#### Predefined Data Configuration

The predefined data configuration includes the name of the SFCC objects like *customer, category, product, order, jwt*. See the configurations for each of the listed object types below:

| {  "customer": {  "first\_name": "customer.profile.firstName",  "last\_name": "customer.profile.lastName",  "email": "customer.profile.email",  "partner\_user\_id": "customer.$customerId"  },  "category": {  "title": "category.$categoryTitle",  "image\_url": "category.$categoryImageURL",  "description": "category.pageDescription",  "content.url": "category.$categoryURL"  },  "product": {  "title": "product.$productTitle",  "image\_url": "product.$productImageURL",  "description": "product.pageDescription",  "url": "product.$productURL",  "partner\_content\_id": "product.ID"  },  "order": {  "partner\_conversion\_id": "order.orderNo",  "cart\_value": "order.$cartValue",  "coupon\_code": "order.$couponList",  "first\_name": "order.billingAddress.firstName",  "last\_name": "order.billingAddress.lastName",  "email": "order.customerEmail",  "partner\_user\_id": "order.$orderCustomerId"  },  "jwt": {  "email": "customer.profile.email",  "customerId": "customer.profile.customerNo"  }  } |
| --- |

Each object has an attributes/pathToValues pair defined, where *attributes* is the name of the attribute for the Extole tag, and *pathToValues* is the path to the attribute value.

#### Attributes

If the attribute name starts with a letter it is used as a path to a SFCC object. See the list of available attributes below.

| Attribute Name | Description |
| --- | --- |
| ***'mapped\_data'*** (optional) | A comma-separated string of attributes / pathToValues pairs — {attributeName: sfccObject.attribute}. Here, *attributes* is the name of the attribute for the Extole tag, and *pathToValues* is the path to the attribute value of the SFCC object. You can specify additional properties for objects: *customer, category, product, order, jwt*. These properties will be added to predefined data at this tag. |
| ***'data'*** (optional) | JSON string of *attribute:value* pairs. This data is added to the tag without processing. |
| ***'productId'*** (optional) | Selected Product ID for the PDP page. This parameter is added to get information about the product. |
| ***'categoryId'*** (optional) | Selected Category ID for the categories page. This parameter is added to get information about the category. |
| ***'orderId'*** and ***'orderToken'*** (optional) | Order ID and order token for the “Thank you” page. This parameter is added to get information about the order. |
| ***'jwt'*** (optional) | A comma-separated string of *attribute:value pairs*. Add the JWT token to the tag and include required JWT attributes plus attributes specified in formatted string. |

#### Computed Attributes

If the name of the attribute starts with $ like *order.$orderTotal*, this attribute value should be a result of a call to the orderTotal function. See the list of available computed attributes below.

| Attribute Name | Description |
| --- | --- |
| customer.$customerId | Get Customer ID for the anonymous user or Customer Number for registered user |
| category.$categoryTitle | Get category name. If the category name ID is empty, get category ID. |
| category.$categoryImageURL | Get category image URL. |
| category.$categoryURL | Get category page URL. |
| product.$productTitle | Get product name. If the product name ID is empty, get product ID. |
| product.$productImageURL | Get product image URL. |
| product.$productURL | Get product page URL. |
| order.$cartValue | Get the full order value before coupons. |
| order.$couponList | Get a list of coupon codes separated by commas. |
| order.$orderCustomerId | Get Order customer ID for the anonymous user or Customer Number for registered user |

#### Example Zone Tag Data

Tag data example:

| <isinclude url="${URLUtils.url('Extole-Zonetag',  'name', 'example',  'elementId', 'extole\_id\_example',  'predefined\_data', 'customer',  'mapped\_data', 'customerAge: customer.profile.custom.age',  'data', 'custom: extole\_custom\_data',  )}" /> |
| --- |

Results in the following create zone object:

| {  "example": "extole\_id\_example", "element\_id": "element\_id\_example",  // following attributes inserted because of predefined\_data: ‘customer’ parameter value  "email": "test@example.com",  "partner\_user\_id": "123",  "first\_name": "Robin",  "last\_name": "Hood",  // inserted because of ‘mapped\_data’, ‘customerAge: customer.profile.custom.age’  "customerAge": "25",  // inserted because of 'data', 'custom: extole\_custom\_data'  "custom": "extole\_custom\_data"  …  } |
| --- |

### 

### Salesforce Commerce Cloud Caching

Any inclusion of assets from the cartridge: template/zonetag, should be done so in a way that is uncached. This will avoid situations in which the wrong customer's information is being used in the tags.

Methods to include cartridge assets without caching:

* Using a remote include
* Using the cartridge ISM component

If you are seeing the wrong customer's data being passed, please identify how you are including zone tags in your codebase and see if any local includes can be changed to remote includes with caching disabled.

### Extole Conversion Tag

Most programs reward advocates when a conversion event occurs, such as a purchase. This can be communicated to Extole by adding the conversion tag to the conversion confirmation page on your site.

The conversion tag should pass information about the conversion to allow Extole to run reward rules, quality rules, and attribute the conversion to a referral when possible.

A Conversion tag needs to be added in your confirmation.isml page to track conversion events in Extole.

| <iscomment>  Extole conversion complete tag  </iscomment>  <isinclude url="${URLUtils.url('Extole-Zonetag', 'name', 'conversion', 'orderId', pdict.Order.orderNo, 'orderToken', pdict.Order.orderToken)}" /> |
| --- |

Example Conversion tag data in order confirmation page HTML:

| <script type=*"text/javascript"*>  /\* Start Extole \*/  (function(c,e,k,l,a){c[e]=c[e]||{};for(c[e].q=c[e].q||[];a<l.length;)k(l[a++],c[e])})(window,"extole",function(c,e){e[c]=e[c]||function(){e.q.push([c,arguments])}},["createZone"],0);  /\* End Extole \*/  extole.createZone({  name: 'conversion',  data: {  "first\_name":"Jon",  "last\_name":"Adams",  "email":"jadams@gmail.com",  "partner\_user\_id":" jadams@gmail.com ",  "partner\_conversion\_id":"00003109",  "cart\_value":"23.99",  "coupon\_code":"extole-KB6B-OIM9-H5I7-KHYQ"  }  });  </script> |
| --- |

### 

| Conversion Tag Attribute | Value |
| --- | --- |
| first\_name (recommended) | The first name of the person making the purchase. This is used for personalization and client customer support lookups. |
| last\_name (recommended) | The last name of the person making the purchase. This is used for personalization. |
| email (required) | The email address of the person making the purchase. This is used for quality detection (self referral, bad domains), referral matching, and optionally to trigger program alerts and promotions. |
| partner\_user\_id (recommended) | YOUR unique identifier for this person making the purchase such as an account id or member id. This is used to detect uniqueness, quality, and can help match if people switch emails. |
| partner\_conversion\_id (required) | YOUR order number that uniquely identifies this transaction. |
| cart\_value (required) | The value of the purchase. Ideally this is the gross cart value before coupons have been applied. This is used to let you view and analyze the revenue generated from your referral program. |
| coupon\_code (required) | The coupon code that may have been used to make the purchase. |

### 

### Online Product Variations Properties

In your IDE, update the detail.js script to reflect your product details. This entails updating selected product variation values from your product details page (PDP). These values are stored on the storefront in the ***window.extole.selectedProduct*** property.

**Note:** Don't forget to build your Javascript code after incorporating these changes.

1. Open **app\_storefront\_base/cartridge/client/default/js/product/detail.js**.
2. To the function `updateAttribute`, add the following code to line 70:

| // Extole: update user selected product attribute  $('body').trigger('extole:zoneTagUpdate', response.data); |
| --- |



## 3.4 Styling Your Extole Campaign within Your Storefront

Please contact your Extole CSM or Implementation Manager to create new campaign designs and integrate them with your site.

## 3.5 Testing

Test cases to verify the integration can be found in the [EXT] Code implementation [EXT] Code implementation Tests.xlsx file.

# Operations, Maintenance

## Data Storage

*Intentionally left blank.*

## Availability

*Intentionally left blank.*

## Support

For technical support please contact your Extole contact, or for cartridge specific questions you may email [support@extole.com](mailto:support@extole.com).

# User Guide

## Roles, Responsibilities

*Intentionally left blank.*

## Business Manager

*Configuration options described above, in 3.1.*

## Storefront Functionality

*Intentionally left blank.*

# Known Issues

*Intentionally left blank.*

# Release History

| **Version** | **Date** | **Changes** |
| --- | --- | --- |
| 1.0.0 | May 19, 2015 | Initial release |
| 1.0.1 | Jun 12, 2015 | Add Extole Conversion tag description |
| 15.1.0 | Jun 19, 2015 | * Document version update. * Added instructions for the zone tags implementation. |
| 16.1.0 | Oct 6, 2016 | Certified by Salesforce Commerce Cloud with JavaScript Controller Based Site Genesis |
| 17.1.0 | Feb 1, 2017 | * Updated to version 17.2 (compatibility mode 16.1), SiteGenesis 103.1.9. * Extole zone tags updated. * Extole Client Id replaced with Extole Client Name. * OCAPI-based coupon integration enabled. |
| 18.1.0 | Jan 1, 2018 | * Updated to compatibility mode 17.7, SiteGenesis 103.1.11. * Label added to Extole zone tags. * Job was migrated into new job framework. * All ds files migrated to js All backend logic moved from isml into .js files. |
| 18.2.0 | Mar 15, 2018 | Updated for Mobile-First Reference Architecture compatibility |
| 19.1.0 | June, 2019 | * Updated to compatibility mode 18.10, Salesforce Commerce Cloud API version 19.5. * migrated service to LocalService * changed cartridge structure * updated guide to reflect new Extole Dashboard * changed new AccessToken authorization |
| 21.1.0 | September 17, 2021 | * Update guidance around caching and screenshots |
| 22.1.0 | May 20, 2022 | * Updated to compatibility mode 21.7, Salesforce Commerce Cloud API version 22.4 with SFRA version 6.0.0 * Do not cache extole.io JS script * Upgrade My Account Dashboard security (JWT) * Upgrade Drop A Hint with selected attributes * Added tag customization * Add Production flag to the tag |