

Talon.One Engagement Controllers (SiteGenesis) Integration

Version 105.2.0



Table of Contents

1. Summary.....	3
2. Overview.....	4
<i>Functional Overview.....</i>	4
<i>Use Cases.....</i>	5
<i>Limitations, Constraints.....</i>	12
<i>Compatibility</i>	12
<i>Privacy, Payment</i>	12
3. Implementation Guide.....	13
<i>Setup of Business Manager</i>	13
<i>Configuration.....</i>	13
<i>Custom Code.....</i>	18
<i>External Interfaces.....</i>	47
4. User Guide	48
<i>Logs.....</i>	48
<i>Storefront Functionality</i>	49
5. Known Issues	50
6. Release History	50

1. Summary

Talon.One is a cloud-based Promotion Engine designed to allow expanding enterprises to create, manage and track coupon codes, discounts, referral rewards and loyalty programs all in one holistic platform. Talon.One is a platform that allows you to create highly customizable promotional campaigns thanks to the *Campaign Manager*.

The centralized platform enables users to create targeted promotions on a very granular level to improve customer experience.

With Talon.One you instantly unlock a whole host of powerful features including: Coupon generation, coupon tracking, referral management, automated discounts and notifications, loyalty program backend logic, default campaign budgets, and single sign-on.

The full features are too extensive to list, but with this customers can trigger any promotion solution they like based on customer profile data, interactions, location, device and segment data they obtain through 3rd parties.

Talon.One allows enterprises to run loyalty programs based on points, memberships, and multi-tiered KPIs, personalize each incentive/reward per customer attribute, and utilize the earn and burn module to buyers' wallet spend. It comes with an application programming interface (API), which allows businesses to integrate the platform with several third-party systems such as Salesforce, and more.

Platform relies on **Applications** inside the Campaign Manager. They are the receiving end of the integration layer and typically represent a region where you do business..

Each Application contains **campaigns**. These are your promotional campaigns, based on your marketing decisions. They are defined by budgets, start and end dates, attributes, and the promotion logic.

The logic of your campaign is implemented by **rules**. They usually have a set of conditions and effects such as *check if the coupon code is valid, or generate a referral code..*

2. Overview

The Talon.One promotion cartridge enables commerce cloud to integrate with the Talon.One promotion service. The purpose of the document is to guide through an easy installation of Talon.One cartridge onto a commerce cloud store.

The integration is based on the SiteGenesis demo store, provided by SFCC.

Note:- Our assumption is that we won't be using any SFCC coupons, promotion & campaigns.

The integration consists of an archive with contents as described in the below table.

Name	Purpose
cartridges	Contains Talon.One integration cartridges
cartridges\int_talonone_controllers	This cartridge contains the SiteGenesis specific changes required for Talon.One integration
cartridges\int_talonone	This cartridge contains the API calls for Talon.One integration and common code
cartridges\app_talonone_controllers	This cartridge references all the changes done to OOB SiteGenesis files. Please do not include this in the cartridge path and this is for reference purposes only
cartridges\bm_talonone	This cartridge references all the changes done on BM Talon.One transactions and is included in the BM cartridge path.
metadata	Contains system object extensions and configurations required for the integration
documentation	Contains this document "Talon.One SiteGenesis Integration Document"

2.1 Functional Overview

Talon.One Key Features:-

- Activity Dashboard
- Campaign Management
- CRM
- Customer Activity Tracking
- Customer Profiles
- Discount Management
- Referral Management
- Loyalty Cards
- Loyalty Program
- Referral Tracking
- Reporting/Analytics
- Third Party Integrations
- Rewards Management

Integration API

Use the Integration API to send external information to the Talon.One Rule Engine or to retrieve data in **high-load environments**. It's the main API for all integration use-cases.

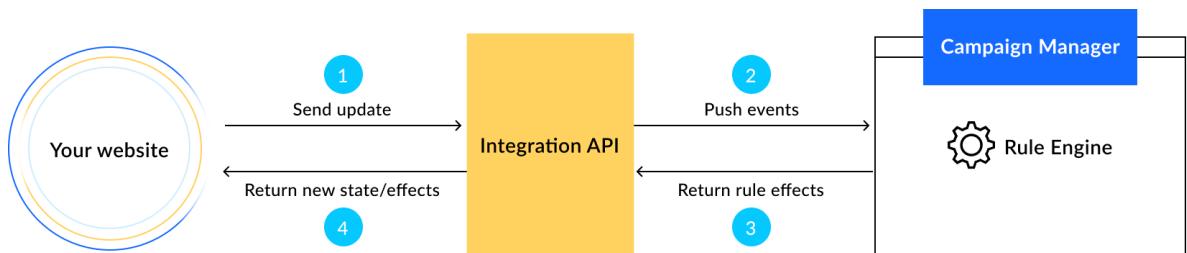
Typical use cases are:

- Sharing order/cart data, such as cart items, delivery address or payment information.
- Get promotion campaign [effects](#).
- Sharing customer data, such as total sales or loyalty memberships.
- Getting customer-related data.

Here is a usual example of workflow with the Integration API:

1. Your webshop sends a request via the Integration API.
2. The request is transformed in [event](#) and sent to the Rule Engine.
3. The Rule Engine uses this event to find the rules that match the current event.
4. The Rule Engine returns the effect(s) of the matching rules to the integration layer.
5. The integration layer applies the effect(s) as needed.

Represented visually, the relation between the Campaign Manager, your webshop, and the Rule Engine is as follows:



The only coupling of your integration with campaigns is through the Rule Engine: the integration must send the attributes that the Rule Engine requires.

2.2 Use Cases

The different effect types are:-

2.2.1 Discount

The effects happen when you use a Discount effect.

2.2.1.1 SetDiscount (Order Level Discount)

It indicates that a discount should be set on the total shopping cart value of the current order with the given label and amount.

This discount should overwrite any existing discount with the same name. The most recent integration state update always returns the latest values for all effects, effectively overwriting any previous effects.

The screenshot shows a commerce cloud website interface. At the top, there's a navigation bar with links for NEW ARRIVALS, WOMENS, MENS, ELECTRONICS, and TOP SELLERS. A search bar is also present. The main content area displays a product in the cart: a "Floral Scoop Neck Tank Dress" (Item No.: 701644026622, Color Black & White, Size L). The cart summary shows a Subtotal of \$445.00, an Order Discount of -\$44.50, and an Estimated Total of \$400.50. Below the cart, there are fields for Enter coupon code and Enter Referral Code, each with an "Apply" button. A yellow box highlights the "Order Discount: 10 % off for cart item total" message.

2.2.1.2 SetDiscountPerItem (Product Level Discount)

This effect schema is returned when you use the Discount individual items or Discount individual items pro rata effect in a rule.

It indicates that a discount per item should be applied on the specific item specified in the effect.

The screenshot shows a commerce cloud website interface. The product in the cart is a "Striped Silk Tie" (Item No.: 793775370033, Color Turquoise). The cart summary shows a Subtotal of \$161.95, an Order Discount of -\$16.19, and an Estimated Total of \$161.95. Below the cart, there are fields for Enter coupon code and Enter Referral Code, each with an "Apply" button. A yellow box highlights the "- 10% off for item 793775370033" message.

2.2.1.3 SetDiscountPerAdditionalCost (Shipping Level Discount)

It indicates that a discount that should be applied on a specific additional cost.

FREE 2-DAY SHIPPING FOR ORDERS OVER \$300


commerce cloud

SEARCH
LOCATION
USER PROFILE
CART (6)

NEW ARRIVALS WOMENS MENS ELECTRONICS TOP SELLERS

[« Continue Shopping](#) [CHECKOUT](#)

PRODUCT	DELIVERY OPTIONS	QTY	PRICE	TOTAL
 Striped Silk Tie Item No.: 793775370033 Color Turquoise Edit Details	Home Delivery	<input type="text" value="6"/>	<small>In Stock</small> Remove Add to Wishlist	\$39.50 \$29.99 \$179.94
<input type="text" value="Enter coupon code"/> Apply Update Cart			<small>Subtotal</small> Shipping Ground <small>Shipping Discount</small> Sales Tax	\$179.94 \$7.99 -\$4.00 \$9.20 Estimated Total \$193.13
<input type="text" value="Enter Referral Code"/> Apply				

[« Continue Shopping](#) [CHECKOUT](#)

2.2.1.4 addFreeItem

It indicates that a free item should be added to the shopping cart in the current session.

FREE 2-DAY SHIPPING FOR ORDERS OVER \$300


commerce cloud

SEARCH
LOCATION
USER PROFILE
CART (9)

NEW ARRIVALS WOMENS MENS ELECTRONICS TOP SELLERS

[« Continue Shopping](#) [CHECKOUT](#)

PRODUCT	DELIVERY OPTIONS	QTY	PRICE	TOTAL
 Boat Neck Sleeveless Sheath Dress. Item No.: 701644358969 Color Black Size M Edit Details	Home Delivery	<input type="text" value="2"/>	<small>In Stock</small> Remove Add to Wishlist	\$119.00 \$238.00
 Floral Scoop Neck Tank Dress Item No.: 701644026622 Color Black & White Size L Edit Details	Home Delivery	<input type="text" value="5"/>	<small>In Stock</small> Remove Add to Wishlist	\$89.00 \$445.00
 Floral Sheath Dress <small>- Free B with C and D</small> Item No.: 701643698189 Color Surf Spray & Navy Size 8 Edit Details	Home Delivery	<input type="text" value="2"/>	<small>Pre-Order The expected in-stock date is Jun 1, 2029.</small> Remove Add to Wishlist	\$109.00 <small>List Price \$248.00</small> 0 x Free B with C and D <small>Item Total \$0.00</small>
<input type="text" value="Enter coupon code"/> Apply Update Cart			<small>Subtotal</small> Shipping Ground <small>Sales Tax</small>	\$683.00 \$15.99 \$34.95 Estimated Total \$733.94
<input type="text" value="Enter Referral Code"/> Apply				

[« Continue Shopping](#) [CHECKOUT](#)

2.2.2 Coupons

The effects happen when you use a Discount effect with valid coupon codes

2.2.2.1 SetDiscount (Order Level Discount)

The screenshot shows the Salesforce Commerce Cloud checkout interface. At the top, there's a banner for "FREE 2-Day SHIPPING FOR ORDERS OVER \$300". Below it, the "commerce cloud" logo is visible. A search bar with placeholder text "Enter Keyword or Item No." and a magnifying glass icon is positioned next to user icons for location, profile, and cart (with a "3" notification).

The main content area shows a shopping cart with one item: a "Floral Scoop Neck Tank Dress". The cart summary table includes columns for PRODUCT, DELIVERY OPTIONS, QTY, PRICE, and TOTAL.

A yellow box highlights the "Coupon" section, which displays a coupon code "7YNMYM9W" and a message stating "10% discount on session total (-\$55.40)". A "Remove" button is also present here.

Below the coupon section, a message says "Order Discount: 10% discount on session total".

The cart summary table shows the following details:

	Subtotal	\$267.00
Order Discount	- \$55.40	
Shipping Ground	\$9.99	
Sales Tax	\$11.08	
Estimated Total	\$232.67	

At the bottom, there are "Continue Shopping" and "CHECKOUT" buttons.

2.2.2.2 SetDiscountPerItem (Product Level Discount)

This screenshot shows the same checkout interface as the previous one, but with a different item in the cart: a "3/4 Sleeve V-Neck Top". The cart summary table includes columns for PRODUCT, DELIVERY OPTIONS, QTY, PRICE, and TOTAL.

A yellow box highlights the "Coupon" section, which displays a coupon code "EVBNDL93" and a message stating "eligibility (-\$14.40)". A "Remove" button is also present here.

The cart summary table shows the following details:

	Subtotal	\$129.60
Shipping Ground	\$7.99	
Sales Tax	\$6.88	
Estimated Total	\$144.47	

At the bottom, there are "Continue Shopping" and "CHECKOUT" buttons.

2.2.2.3 SetDiscountPerAdditionalCost (Shipping Level Discount)

[« Continue Shopping](#)[CHECKOUT](#)

PRODUCT	DELIVERY OPTIONS	QTY	PRICE	TOTAL
 3/4 Sleeve V-Neck Top Item No.: 701642923466 Color Butter Size S Edit Details	Home Delivery	<input type="text" value="6"/>	<small>In Stock</small> Remove Add to Wishlist	\$24.00 \$144.00
Coupon Coupon Number: HELLO Shipping Discount with Coupon "HELLO" (-\$4.00)				
<input type="text" value="Enter coupon code"/>	Apply	Update Cart	Subtotal	\$144.00
<input type="text" value="Enter Referral Code"/>	Apply		<small>Shipping Ground</small> Shipping Discount <small>Sales Tax</small>	<small>\$7.99</small> -\$4.00 <small>\$7.40</small>
			Estimated Total	\$155.39

[« Continue Shopping](#)[CHECKOUT](#)

2.2.2.4 addFreeItem

[« Continue Shopping](#)[CHECKOUT](#)

PRODUCT	DELIVERY OPTIONS	QTY	PRICE	TOTAL
 Neutral Floral Dress Item No.: 701644024901 Color Dark Stone Size 8 Edit Details	Home Delivery	<input type="text" value="2"/>	<small>In Stock</small> Remove Add to Wishlist	\$129.00 \$258.00
 Short Sleeve Belted Shirtdress Item No.: 701643563814 Color Ink Size 4 Edit Details	Home Delivery	<input type="text" value="2"/>	<small>In Stock</small> Remove Add to Wishlist	\$149.00 \$110.99 \$221.98
 Pleated Dress With Front Sash. - Free B with C and D of coupon Item No.: 701644397364 Color Admiral Navy Size 12 Edit Details	Home Delivery	<input type="text" value="1"/>	<small>In Stock</small> Remove Add to Wishlist	\$129.00 <small>List Price \$129.00</small> Free B with C and D of coupon <small>Item Total \$0.00</small>
Coupon Coupon Number: 5RRKWV3C Free B with C and D of coupon (-\$129.00)				
<input type="text" value="Enter coupon code"/>	Apply	Update Cart	Subtotal	\$479.98
<input type="text" value="Enter Referral Code"/>	Apply		<small>Shipping Ground</small> <small>Sales Tax</small>	<small>\$9.99</small> <small>\$24.50</small>
			Estimated Total	\$514.47

[« Continue Shopping](#)[CHECKOUT](#)

2.2.3 Loyalty

2.2.3.1 AddLoyaltyPoints

It indicates that a defined amount of loyalty points was successfully added to the customer's loyalty wallet.

The points only persist when the session is closed.

[« Continue Shopping](#)**CHECKOUT**

PRODUCT	DELIVERY OPTIONS	QTY	PRICE	TOTAL
 Neutral Floral Dress Item No.: 701644024901 Color Dark Stone Size 8 Edit Details	Home Delivery	<input type="text" value="2"/>	In Stock Remove Add to Wishlist	\$129.00 \$149.00 \$110.99 \$258.00
 Short Sleeve Belted Shirtdress Item No.: 701643563814 Color Ink Size 4 Edit Details	Home Delivery	<input type="text" value="2"/>	In Stock Remove Add to Wishlist	\$149.00 \$149.00 \$110.99 \$221.98
Coupon Coupon Number: LOYALTY Remove				
<input type="text" value="Enter coupon code"/> Apply Update Cart				
<input type="text" value="Enter Referral Code"/> Apply				
			Subtotal	\$479.98
			Shipping Ground	\$9.99
			Sales Tax	\$24.50
			Estimated Total	\$514.47
			Loyalty Points to Earn:	100.00
« Continue Shopping CHECKOUT				

Thank you for your order.

If you have questions about your order, we're happy to take your call (800-555-0190) Monday - Friday, 8AM - 8PM

 Order Number: **00000105**

Order Placed: Jul 26, 2022

PAYMENT METHOD

Credit Card

Debit

Visa

MasterCard

American Express

Discover

PayPal

Bank Transfer

EFT

ACH

Check

Bank Wire

Bank Draft

Money Order

Banker's Draft

Draft

Bank Note

Bank Check

Bank Deposit

Bank Deposit Note

Bank Deposit Check

The screenshot shows a product in the cart:

- Product:** Boat Neck Sleeveless Sheath Dress
- Delivery Options:** Home Delivery
- Quantity:** 1
- Price:** \$119.00
- Total:** \$119.00

Coupon applied: DISLOYAL (Loyalty Points to Spend: 80.00)

Cart Summary:

Subtotal	\$119.00
Shipping	-
Sales Tax	-
Estimated Total	\$119.00

Buttons: Continue Shopping, CHECKOUT

2.2.4 Referral

Referral codes are intended for your customers to share with other people so that they too become customers. Each referral code is related to an advocate and their friend:

- Advocate: The customer who invited their friend via a referral code.
- Friend: The person who receives the code from an advocate.

When a referral code is successful, the configured promotion will be applied to that referral.

2.2.4.1 SetDiscount

The screenshot shows a product in the cart:

- Product:** Boat Neck Sleeveless Sheath Dress
- Delivery Options:** Home Delivery
- Quantity:** 1
- Price:** \$119.00
- Total:** \$119.00

Referral applied: BHJP-DCMM (Order Discount: 10% Off)

Cart Summary:

Subtotal	\$119.00
Order Discount	-\$11.90
Shipping	-
Sales Tax	-
Estimated Total	\$107.10

Buttons: Continue Shopping, CHECKOUT

2.3 Limitations, Constraints

TBC

2.4 Compatibility

Available since Commerce Cloud Platform Release 22.7, Site Genesis 105.2.0

The cartridge is available for installations on storefronts that support both Controller and SFRA implementations.

2.5 Privacy, Payment

N/A

3. Implementation Guide

3.1 Setup of Business Manager

This guide requires that the latest SG code is uploaded alongside the SG latest reference applications.

The prepared test data setup is using the RefArchGlobal reference site as a base.

The Talon.One LINK Cartridge contains several cartridges that are required for full functionality. Additionally, Controller and SFRA support is broken out into two separate cartridges, thereby facilitating the installation and use of one or the other models.

Import all three cartridges into UX studio and associate them with a Server Connection.

Site Cartridge Assignment Install the cartridge.

1. Navigate to Administration > Sites > Manage Sites
2. Click on the Site Name for the Storefront Site that will add Talon.One Engagement Functionality
3. Select the "Settings" tab
4. For Sitegenesis "int_talonone_controllers:int_talonone" need to be added to the cartridge path,

i.e. Sitegenesis-based Site Path:

int_talonone_controllers:int_talonone:app_storefront_controllers:app_storefront_core

[Administration](#) > [Sites](#) > [Manage Sites](#) > SiteGenesis - Settings

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

SiteGenesis - 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 are intended only to support an older configuration style.

HTTP Hostname:

HTTPS Hostname:

Instance Type: All

Cartridges:

Effective Cartridge Path:

5. Repeat steps 2 – 4 for each Storefront Site where Talon.One Engagement will be implemented.

Metadata import

1. Navigate to the metadata folder of the project and open the site_template folder.
2. Open the sites folder and edit the 'siteIDHere' folder to the site ID of the site you want.
3. Add a folder for each site you need TalonOne Engagement on.
4. Navigate to Administration > Site Development > Site Import & Export
5. Zip the Talon.One Engagement_site_template folder and import it.

3.2 Configuration

3.2.1 Service configuration

The initial configuration was imported from services.xml file included in the metadata folder.

Go to: Administration → Operations → Services

- You would see the service with name **talonone.http.service**

- You would see the linked services profile: **talonone.http.profile**
- You would see the linked services credentials: **talonone.http.credentials**

Type	Profile	Credentials	Status
HTTP	talonone.http.profile	talonone.http.credentials	Live

Fields with a red asterisk (*) are mandatory. Click **Apply** to save the details. Click **Reset** to revert to the last saved state.

Name: talonone.http.service
Type: HTTP
Enabled:
Service Mode: Live
Log Name Prefix: talonone
Communication Log Enabled:
Force PRD Behavior in Non-PRD Environments:

Profile: talonone.http.profile
Credentials: talonone.http.credentials

<< Back to List **Apply** **Reset**

Service's profile used for configuration of general connection properties.

- Go to: Administration → Operation → Services → Profiles tab
- You would see the profile with name talonone.http.profile

Fields with a red asterisk (*) are mandatory. Click **Apply** to save the details. Click **Reset** to revert to the last saved state.

This profile is used by 1 service.

Name: talonone.http.profile
Connection Timeout (ms): 1,000
Enable Circuit Breaker:
Max Circuit Breaker Calls:
Circuit Breaker Interval (ms):
Enable Rate Limit:
Max Rate Limit Calls:
Rate Limit Interval (ms):

<< Back to List **Apply** **Reset**

Service's credentials used for authentication to the Talon.One: url to be added.

- Go to: Administration → Operation → Services → Credentials tab
 - You must see in credentials list credentials with name talonone.http.credentials
 - Open talonone.http.credentials and setup the URL fields as provided by the Talon.One Solutions team. Talon.One is configured with only the URL and no username or password.
- URL: Talon.One URL eg:- <https://tryzens.europe-west1.talon.one/v2>

talonone.http.credentials

Fields with a red asterisk (*) are mandatory. Click **Apply** to save the details. Click **Reset** to revert to the last saved state.
These credentials are used by 1 service.

Name:*	talonone.http.credentials
URL:	https://tryzens.europe-west1.talon.one/v2
User:	
Password:	*****

Apply **Reset**

[**<< Back to List**](#)

3.2.2 General configuration

Talon.One configuration values should be set properly in the Business Manager, so that the API requests are made successfully.

3.2.2.1 SitePreferences

The properties to enable the functionality are on the main level in Site Preferences.

- Navigate to Site > Site Preferences > Custom Site Preferences.
- Select the Talon One Configuration preference group.

Full list of the available site preferences can be seen below:

Configuration	Purpose
isTalonOneEnabled	Used to Enable/Disabled Talon.One for the site.
API_key_prefix	The API prefix value with corresponding version, we have got from Talon.One ApiKey-v1
API_Key	Add the Talon.One API key, which we have created from Talon.One Campaign Manager as provided by the Talon.One team
profile_id_prefix	Prefix to identify if it's a test, or production site Eg:- test_RefArch, or prod_RefArch
productAttributes	Add the supporting attribute available for products. Eg:- color, size
talonOneAttributeObjects	List of Attributes needed to be created in campaign manager
isLoyalPointsEnabled	Based on the value, decides whether or not to display loyalty points on the site.
isReferralCodeEnabled	Based on the value, decides whether or not to display referral code block on the site.

1-8 of 8

Name	Value	Default Value
Is Talon One Enabled or not (isTalonOneEnabled)	Yes	<input type="button" value="Edit Across Sites"/>
API Key Prefix (API_key_prefix)	ApiKeyv1 (ApiKeyv1)	<input type="button" value="Edit Across Sites"/>
API Key (API_Key) (String)	579e6788ceb79e78cad18f6f089eda4985a2dca5e431b72ea659838c48d7ef	<input type="button" value="Edit Across Sites"/>
profile id prefix (profile_id_prefix)	None	<input type="button" value="Edit Across Sites"/>
Product Attributes (productAttributes)	<input type="button" value="Add"/> color <input checked="" type="radio"/> size <input type="radio"/>	<input type="button" value="Edit Across Sites"/>
TalonOne Attribute Objects (talonOneAttributeObjects) Eg: "entity" : CustomerProfile or CustomerSession or CartItem "type" : string or Number or Boolean or Time	<pre>{ "attributes": [{ "entity": "CustomerSession", "name": "SiteId", "title": "SiteId", "type": "string", "description": "Current Site ID", "suggestions": [], "editable": true }] }</pre> <p>Eg: "entity" : CustomerProfile or CustomerSession or CartItem "type" : String or N...</p>	<input type="button" value="Edit Across Sites"/>
Is loyalty points enabled or not (isLoyalPointsEnabled)	Yes	<input type="button" value="Edit Across Sites"/>
SFCC Admin can disable/ enabled loyal points in total displayed in 'Order Summary' and 'Order Confirmation page'	SFCC Admin can disable/ enabled loyal points in total displayed in 'Order Summa...'	
Is referral code enabled or not (isReferralCodeEnabled)	Yes	<input type="button" value="Edit Across Sites"/>

3.2.2.2 System Objects

Basket

Attribute	Purpose
talononeRejectedFreelitem	List of all Talon.One Rejected Free Items
referralCode	Contains the entered referral code
talononeCouponCodes	Contains all the coupon codes , which are acceptable by Talon.One

Order

Attribute	Purpose
customerProfileID	Talon.One customer profile id
customerSessionID	Talon.One customer session id
talononeRejectedFreelitem	List of all Talon.One Rejected Free Items
talononeCouponCodes	Contains all the coupon codes , which are acceptable by Talon.One
referralCode	Contains the entered referral code

PriceAdjustment

Attribute	Purpose

isTalonOneAdjustment	Is set true when a Talon.One promotions are updated against the lineitem priceadjustment
isTalonOneFreeItem	Is set true when a Talon.One promotions has free item and are updated against the lineitem priceadjustment
talonOneFreeItemQty	Contains the number of free item qty
talonOneLineItemUuid	Line Item Unique Id
talonOnePriceAdjustment	Contains the unique adjustment Id
talonOnePromotionRuleName	Promotion Rule Name

ProductionLineItem

Attribute	Purpose
hasTalonOneFreeItem	Is set true when a Talon.One promotions has free item

3.2.2.3 Jobs

The job iterates over the configured preference attributes value of “talonOneAttributeObjects” to create custom attributes on campaign manager, if attributes are not created yet.

It's a one time execution. Until if any new custom attributes are needed or to be configured.

Note:- This job must be executed each time a new sitepreference attribute is added in order to create the corresponding Campaign Manager attribute.

Jobsteps

3.3 Custom Code

Integration may vary based on the customers' storefront. Site Genesis version 105.2.0 is used as a reference to demonstrate Talon.One integration

3.3.1 Template Modification

The following template changes should be made regardless of whether a controller integration approach are being used:

3.3.1.1 default/checkout/cart/cart.isml

a) Add Code:

```
<iscomment> Talonone - Begin: Displaying an error message, while the
talonone free item is in OOS </iscomment>
    <isset name="isTalonOneEnabled"
value="${require('* /cartridge/scripts/util/talonOnePreferences').isEnabled()
}" scope="page" />
        <isif condition="${isTalonOneEnabled &&
pdict.isTalonOneAddFreeItemerror}">
            <div class="error-form talonone-freeitem-oos-error">
                <i class="fa fa-exclamation-triangle fa-2x pull-left"></i>
                ${Resource.msg('talononeerror.freeitem.cart.or.checkout','ta
lonOne',null)}
            </div>
        </isif>
    <iscomment> Talonone - End: Displaying an error message, while the
talonone free item is in OOS </iscomment>
```

Before <form action="\${URLUtils.httpContinue()}">
 method="post" name="\${pdict.CurrentForms.cart.htmlName}" id="cart-items-form"> tag as shown
 below:-

```

71 |             <isif condition="${!enableCheckout}">
72 |                 <div class="error-form">
73 |                     <i class="fa fa-exclamation-triangle fa-2x pull-left"></i>
74 |                     ${Resource.msg('cart.limitedQuantity','checkout',null)}
75 |                 </div>
76 |             </isif>
77 |
78 |             <iscomment> Talonone - Begin: Displaying an error message, while the talonone free item is in OOS </iscomment>
79 |             <isset name="isTalonOneEnabled" value="${require('/cartridge/scripts/util/talonOnePreferences').isEnabled()}" scope="page" />
80 |             <isif condition="${!isTalonOneEnabled && pdict.isTalonOneAddFreeItemError}">
81 |                 <div class="error-form talonone-freeitem-oos-error"> You, last week • TALBLD-23 FreeItem OOS error message on SG & SFRA
82 |                     <i class="fa fa-exclamation-triangle fa-2x pull-left"></i>
83 |                     ${Resource.msg('talononeerror.freeitem.cart.or.checkout','talonOne',null)}
84 |                 </div>
85 |             </isif>
86 |             <iscomment> Talonone - End: Displaying an error message, while the talonone free item is in OOS </iscomment>
87 |
88 |             <iscomment>show the basket when we have data</iscomment>
89 |             <form action="${URLUtils.httpContinue()}" method="post" name="${pdict.CurrentForms.cart.htmlName}" id="cart-items-form">
90 |                 <fieldset>

```

b) Add Code :-

```

<iscomment> Talonone - Begin: Updating the talonone promotion callout
message </iscomment>
                                                <isif
condition="${isTalonOneEnabled}">
                                                <isprint
value="${pa.custom.talonOnePromotionRuleName}" encoding="off" />
                                                <iselse/>
                                                <isprint
value="${pa.promotion.calloutMsg}" encoding="off"/>
                                            </isif>
                                            <iscomment>
Talonone - End: Updating the talonone promotion callout message </iscomment>

```

Added this inside <div class="promo-adjustment"> after the condition check

```

<isif condition="${lineItem.quantityValue > 1 && lineItem.quantityValue != pa.quantity}">
    <isprint value="${pa.quantity}" /> x
</isif> tag as shown below:-

```

```

283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300

```

```

<iscomment>Display the promotion name for each price adjustment.</iscomment>
<isloop items="${lineItem.priceAdjustments}" var="pa" status="prAdloopstatus">
    <div class="promo-adjustment">
        <isif condition="${lineItem.quantityValue > 1 && lineItem.quantityValue != pa.quantity}">
            <isprint value="${pa.quantity}" /> x
        </isif>
        <iscomment> Talonone - Begin: Updating the talonone promotion callout message </iscomment>
        <isif condition="${isTalonOneEnabled}">
            <isprint value="${pa.custom.talonOnePromotionRuleName}" encoding="off" />
        <iselse/>
            <isprint value="${pa.promotion.calloutMsg}" encoding="off"/>
        </isif>
        <iscomment> Talonone - End: Updating the talonone promotion callout message </iscomment>
    </div>
</isloop>

```

c) Add Code:-

```

<iscomment> TalonOne Begin: Removing the applied / not applied label from
coupons </iscomment>
                                                <isif condition="${!isTalonOneEnabled}">
                                                    <isif
condition="${FormCoupon.object.applied}">
                                                        <span class="bonus-
item">${Resource.msg('global.applied','locale',null)}</span>
                                                    <iselse/>

```

```

        <span class="bonus-item">${Resource.msg('global.notapplied','locale',null)}</span>
            </isif>
        </isif>
        <iscomment> TalonOne End: Removing the applied / not applied label from coupons </iscomment>

```

Added this inside the loop of <isloop items=" \${pdict.CurrentForms.cart.coupons}" var="FormCoupon" status="loopstateCoupons"> after <td class="item-total"> tag as shown below:-

```

565      <td class="item-price"></td>
566
567      <td class="item-total">
568
569          <iscomment> TalonOne Begin: Removing the applied / not applied label from coupons </iscomment>
570          <isif condition="${listTalonOneEnabled}">
571              <isif condition="${FormCoupon.object.applied}">
572                  <span class="bonus-item">${Resource.msg('global.applied','locale',null)}</span>
573              <iselse/>
574                  <span class="bonus-item">${Resource.msg('global.notapplied','locale',null)}</span>
575              </isif>
576          <iscomment> TalonOne End: Removing the applied / not applied label from coupons </iscomment>
577      </td>
578
579      </tr>|   You, 3 weeks ago • TALBLD-23 Updating the cart calculation in SFCC a...
580
581
582  </isloop>

```

d) Add Code:-

```

<!--TalonOne Start -->
    <isif
condition=" ${pdict.talon.referral.isReferralEnabled &&
pdict.talon.referral.code}">
        <tr class="rowcoupons">
            <td class="item-image">
                <!-- BLANK IMAGE CELL -->
            </td>
            <td class="item-details">
                <div
class="name">${Resource.msg('label.referral.Referral','talonOne',null)}</di
v>
                <div class="cartcoupon clearfix">
                    <span
class="label">${Resource.msg('label.referral.ReferralTitle','talonOne',null
)}</span>
                    <span class="value">
${pdict.talon.referral.code}
                    </span>
                </div>
            </td>
            <td class="item-quantity"></td>
            <td class="item-quantity-details">
                <button class="textbutton"  value="Remove"
onclick="event.preventDefault();location.href =
'${pdict.talon.referral.removeReferralCodeUrl}';">
                    ${Resource.msg('global.remove','locale',null)}
                </button>
            </td>
            <td class="item-price"></td>

```

```

        <td class="item-total">
            <span class="bonus-
item">${Resource.msg('global.applied','locale',null)}</span>
        </td>
    </tr>
</isif>
<!--TalonOne End -->
```

Added this before <isloop items="\${pdict.Basket.bonusDiscountLineItems}" var="bonusDiscountLineItem" status="loopstate"> tag as shown below:-

```

583         <!--Talonone start -->
584         <isif condition="${pdict.talon.referral.isReferralEnabled && pdict.talon.referral.code}">
585             <tr class="rowcoupons">
586                 <td class="item-image">
587                     <!-- BLANK IMAGE CELL -->
588                 </td>
589                 <td class="item-details">
590                     <div class="name">${Resource.msg('label.referral.Referral','talonOne',null)}</div>
591                     <div class="cartcoupon clearfix">
592                         <span class="label">${Resource.msg('label.referral.ReferralTitle','talonOne',null)}</span>
593                         <span class="value">
594                             ${pdict.talon.referral.code}
595                         </span>
596                     </div>
597                 </td>
598                 <td class="item-quantity"></td>
599                 <td class="item-quantity-details">
600                     <button class="textbutton" value="Remove" onclick="event.preventDefault();location.href = '${pdict.talon.referral.removeReferralCodeUrl}';">
601                         ${Resource.msg('global.remove','locale',null)}
602                         Unlink Krishnan CC, last week * For TALBLD-27 PR ...
603                     </button>
604                 </td>
605                 <td class="item-price"></td>
606                 <td class="item-total">
607                     <span class="bonus-item">${Resource.msg('global.applied','locale',null)}</span>
608                 </td>
609             </tr>
610         </isif>
611         <!--TalonOne End -->
612
613         <iscomment>Bonus discount line items</iscomment>
614
615         <isloop items="${pdict.Basket.bonusDiscountLineItems}" var="bonusDiscountlineItem" status="loopstate">
616
```

e) Add Code:-

```

<iscomment> Talonone - Begin: Updating the talonone promotion callout
message </iscomment>
                    <isif
condition ="${isTalonOneEnabled}">
                        <isprint
value ="${priceAdjustment.custom.talonOnePromotionRuleName}" encoding="off"
/>
                        <iselse/>
                        <isprint
value ="${priceAdjustment.promotion.calloutMsg}" encoding="off" />
                    </isif>
                    <iscomment> Talonone - End:
Updating the talonone promotion callout message </iscomment>
```

Add this after \${Resource.msg('summary.orderdiscount','checkout',null)} tag as shown below:-

```

851 <tfoot>
852     <isloop items ="${pdict.Basket.priceAdjustments}" var="priceAdjustment" status="cliloopstate">
853         <tr>
854             <td colspan="6" class="cart-promo">
855                 <div>
856                     <span class="label">${Resource.msg('summary.orderdiscount','checkout',null)}</span>
857                     <span class="value">
858                         <iscomment> Talonone - Begin: Updating the talonone promotion callout message </iscomment>
859                         <isif condition ="${isTalonOneEnabled}">
860                             <isprint value ="${priceAdjustment.custom.talonOnePromotionRuleName}" encoding="off" />
861                         <iselse/>
862                             <isprint value ="${priceAdjustment.promotion.calloutMsg}" encoding="off" />
863                         </isif>
864                         <iscomment> Talonone - End: Updating the talonone promotion callout message </iscomment>
865                     </span>
866                 </div>
867             </td>
868         </tr>
869     </isloop>
870 </tfoot>
```

f) Add Code :-

```
<!--TalonOne Start: Referrals Points -->
    <isif
condition="${pdict.talon.referral.isReferralEnabled}">
        <isinclude template="checkout/cart/referral.isml" />
    </isif>
<!--TalonOne End: Referrals Points -->
```

Add this after `<div class="cart-coupon-code">.... </div>` tag as shown below:-

```
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
```

```
<div class="cart-coupon-code">
    <input type="text" name="${pdict.CurrentForms.cart.referralCode.htmlName}" id="${pdict.CurrentForms.cart.referralCode.htmlName}" placeholder="${Resource.msg('lable1.referral.EnterReferralCode', 'talonOne', null)}" />

    <button type="submit" value="${pdict.CurrentForms.cart.addReferral.htmlName}" name="${pdict.CurrentForms.cart.addReferral.htmlName}" id="add-referral">
        ${Resource.msg('global.apply', 'locale', null)}
    </button>

    <comment>Update Cart</comment>
    <button class="simple" type="submit" value="${pdict.CurrentForms.cart.updateCart.htmlName}" name="${pdict.CurrentForms.cart.updateCart.htmlName}" id="update-cart">
        ${Resource.msg('global.updatecart', 'locale', null)}
    </button>

    <isif condition="${pdict.ReferralError && pdict.ReferralError=='REFERRAL_CODE_MISSING'}">
        <div class="error">
            ${Resource.msg('lable1.referral.error.missing.code', 'talonOne', null)}
        </div>
    </isif>
    <iselseif condition="${pdict.ReferralError && pdict.ReferralError=='NO_ACTIVE_PROMOTION'}">
        <div class="error">
            ${Resource.msg('lable1.referral.error.noactivepromotion', 'talonOne', null)}
        </div>
    </iselseif>
    <iselseif condition="${pdict.CouponStatus != null && pdict.CouponStatus.error}">
        <div class="error">
            ${Resource.msgf('lable1.referral.error.couponstatus', null, pdict.CurrentForms.cart.referralCode.htmlValue)}
        </div>
    </iselseif>
</div>
<!--TalonOne Start: Referrals Points -->
<isif condition="${pdict.talon.referral.isReferralEnabled}">
    <isinclude template="checkout/cart/referral.isml" />
</isif>
<!--TalonOne End: Referrals Points -->
```

```
You, 3 weeks ago • TALON-23 Updating the
```

3.3.1.2 default/checkout/cart/referral.isml

a) It's a complete new isml. Copy and paste this Code to file referral.isml:-

```
<div class="cart-coupon-code cart-actions-top">

    <input type="text"
name="${pdict.CurrentForms.cart.referralCode.htmlName}"
    id="${pdict.CurrentForms.cart.referralCode.htmlName}"
    placeholder="${Resource.msg('lable1.referral.EnterReferralCode', 'talonOne', null)}" />

    <button type="submit"
value="${pdict.CurrentForms.cart.addReferral.htmlName}"
    name="${pdict.CurrentForms.cart.addReferral.htmlName}" id="add-referral">
        ${Resource.msg('global.apply', 'locale', null)}
    </button>

    <isif condition="${pdict.ReferralError && pdict.ReferralError == 'REFERRAL_CODE_MISSING'}">
        <div class="error">
            ${Resource.msg('lable1.referral.error.missing.code', 'talonOne', null)}
        </div>
    </isif>
    <iselseif condition="${pdict.ReferralError && pdict.ReferralError == 'NO_ACTIVE_PROMOTION'}">
        <div class="error">
            ${Resource.msg('lable1.referral.error.noactivepromotion', 'talonOne', null)}
        </div>
    </iselseif>
    <iselseif condition="${pdict.CouponStatus != null && pdict.CouponStatus.error}">
        <div class="error">
            ${Resource.msgf('lable1.referral.error.couponstatus', null, pdict.CurrentForms.cart.referralCode.htmlValue)}
        </div>
    </iselseif>
</div>
```

```

        ${Resource.msgf("cart.NO_ACTIVE_PROMOTION", "talonOne", "", 
pdict.CurrentForms.cart.referralCode.htmlValue)}
    </div>
    <iselseif condition="${pdict.ReferralStatus != null}">
        <isif condition="${pdict.ReferralStatus.error}">
            <div class="error">
                ${pdict.ReferralStatus.code}
            </div>
        <iselseif condition="${pdict.ReferralStatus.error == 
false}">
            <div class="text-success">
                ${pdict.ReferralStatus.code}
            </div>
        </isif>
    </isif>
</div>

```

3.3.1.3 default/checkout/summary/summary.isml

a) Add code:-

```

<td class="item-total">
    <iscomment> TalonOne Begin: Removing the applied 
/ not applied label from coupons </iscomment>
    <isset name="isTalonOneEnabled" 
value="${require('*/*cartridge/scripts/util/talonOnePreferences').isEnabled()}" 
scope="page" />
    <isif condition="${!isTalonOneEnabled}">
        <isif condition="${couponLineItem.applied}">
            <span class="coupon-
applied">${Resource.msg('summary.applied', 'checkout', null)}</span>
        <iselse/>
            <span class="coupon-not-
applied">${Resource.msg('summary.notapplied', 'checkout', null)}</span>
        </isif>
    </isif>
    <iscomment> TalonOne End: Removing the applied / 
not applied label from coupons </iscomment>
</td>

```

Added this inside the loop of <isloop items="\${pdict.Basket.couponLineItems}" var="couponLineItem" status="cliloopstate"> after <td class="item-total"> tag as shown below:-

```

168 |           <td class="item-quantity-details">
169 |             <div class="item-edit-details">
170 |               <a href="${URLUtils.url('cart-show')}">${Resource.msg('global.editdetails','locale',null)}</a>
171 |             </div>
172 |           </td>
173 |
174 |           <td class="item-total">
175 |             <!-- TalonOne Begin: Removing the applied / not applied label from coupons -->
176 |             <isset name="isTalonOneEnabled" value="${require('/cartridge/scripts/util/talonOnePreferences').isEnabled()}" scope="page">
177 |               <ifif condition="!isTalonOneEnabled">
178 |                 <ifif condition="!couponlineItem.applied">
179 |                   <span class="coupon-applied">${Resource.msg('summary.applied','checkout',null)}</span>
180 |                 <else/>
181 |                   <span class="coupon-not-applied">${Resource.msg('summary.notapplied','checkout',null)}</span>
182 |                 </ifif>
183 |               </ifif>
184 |             <!-- TalonOne End: Removing the applied / not applied label from coupons -->
185 |           </td>
186 |
187 |           </tr>
188 |         </isif>
189 |       </isloop>
190 |
191 |     <isloop items="${pdict.Basket.priceAdjustments}" var="priceAdjustment" status="cliloopstate">
192 |
193 |

```

b) Add code :-

```

<!-- TalonOne Start -->
<ifif condition="${pdict.talon.loyaltyEnabled}">
<table class="order-totals-table">
<tbody>
  <tr class="order-sales-tax">
    <td><isprint
value="${pdict.talon.loyalty.loyaltyTitle}" />: </td>
    <td><isprint
value="${pdict.talon.loyalty.loyaltyNetChange}" /></td>
  </tr>
</tbody>
</table>
</ifif>
<!-- TalonOne End -->

```

Added this after `<div class="place-order-totals">... </div>` tag as shown below:-

```

215 |           <div class="place-order-totals">
216 |             <isordertotals p_lineitemctrn="${pdict.Basket}" p_showshipmentinfo="${false}" p_shipmenteditable="${false}" p_totallabel="${Resource.msg('summary', 'label', null)}" p_totalvalue="${pdict.Basket.totalValue}" p_tota...
217 |           </div>
218 |
219 |
220 |           <!-- TalonOne Start -->
221 |           <ifif condition="${pdict.talon.loyaltyEnabled}">
222 |             <table class="order-totals-table">
223 |               <tbody>
224 |                 <tr class="order-sales-tax">
225 |                   <td><isprint value="${pdict.talon.loyalty.loyaltyTitle}" />: </td>
226 |                   <td><isprint value="${pdict.talon.loyalty.loyaltyNetChange}" /></td>
227 |                 </tr>
228 |               </tbody>
229 |             </table>
230 |           </ifif>
231 |           <!-- TalonOne End -->
232 |
233 |           <form action="${URLUtils_https('cosummary-Submit')}" method="post" class="submit-order">
234 |             <fieldset>

```

3.3.1.4 default/checkout/minisummary.isml

Add Code:-

```

<!-- TalonOne Start -->
<ifif condition="${pdict.talon.loyaltyEnabled}">
<table class="order-totals-table">
<tbody>
  <tr class="order-sales-tax">
    <td><isprint value="${pdict.talon.loyalty.loyaltyTitle}" />: </td>
  </tr>
</tbody>
</table>

```

```

        <td><isprint
value="\${pdict.talon.loyalty.loyaltyNetChange}"></td>
    </tr>
</tbody>
</table>
</isif>
<iscomment>TalonOne End</iscomment>
```

Added this after <div class="checkout-order-totals"> <isif condition="\\${checkoutstep > 3}"> </isif> tag as shown below :-

```

36
37     <iscomment>render the order totals</iscomment>
38     <div class="checkout-order-totals">
39         <isif condition="\${checkoutstep > 3}">
40             <isordertotals p_lineitemctnr="\${pdict.Basket}" p_showshipmentinfo="\${true}" p_shipmenteditable="\${true}" p_totallabel="\${Resource
41             <iselse/>
42             <isordertotals p_lineitemctnr="\${pdict.Basket}" p_showshipmentinfo="\${false}" p_shipmenteditable="\${false}" p_totallabel="\${Resource
43         </isif>
44
45         <iscomment>TalonOne Start</iscomment>
46         <isif condition="\${pdict.talon.loyalty.isEnabled}">      Unnikrishnan CC, 6 days ago • Code review and changes
47             <table class="order-totals-table">
48                 <tbody>
49                     <tr class="order-sales-tax">
50                         <td><isprint value="\${pdict.talon.loyalty.loyaltyTitle}"></td>
51                         <td><isprint value="\${pdict.talon.loyalty.loyaltyNetChange}"></td>
52                     </tr>
53                 </tbody>
54             </table>
55         </isif>
56         <iscomment>TalonOne End</iscomment>
57
58     </div>
59
```

3.3.1.5 default/checkout/components/order/ordertotals.isml

Add Code:-

```

<iscomment>TalonOne Start</iscomment>
    <isif condition="\${pdict.talon.loyalty.isLoyaltyEnabled}">
        <tr class="order-sales-tax">
            <td><isprint
value="\${pdict.talon.loyalty.loyaltyTitle}"></td>
            <td><isprint
value="\${pdict.talon.loyalty.loyaltyNetChange}"></td>
        </tr>
    </isif>
    <isif condition="\${pdict.talon.referral.isReferralEnabled &&
pdict.talon.isConfirmationPage && pdict.talon.referral.code}">
        <tr class="order-sales-tax">
            <td><isprint
value="\${Resource.msg('referral.summary.referralApplied','talonOne',null)}"/>
        </td>
            <td><isprint value="\${pdict.talon.referral.code}"></td>
        </tr>
    </isif>
    <iscomment>TalonOne End</iscomment>
```

Add this after <td class="order-value"><isprint value="\\${orderTotalValue}"></td> tag as shown below:-

```
123  
124     <!--comment>order total</comment>  
125     <tr class="order-total">  
126         <if condition="${LineItemCtnr.totalGrossPrice.available}">  
127             <isset name="orderTotalValue" value="${LineItemCtnr.totalGrossPrice}" scope="page"/>  
128         <else/>  
129             <isset name="orderTotalValue" value="${LineItemCtnr.getAdjustedMerchandiseTotalPrice(true).add(LineItemCtnr.giftCertificateT  
130             </if>  
131  
132             <td><print value="${pdict.p_totallabel}" /></td>  
133             <td class="order-value"><print value="${orderTotalValue}" /></td>  
134             <!--comment>TalonOne Start</comment>  
135         <if condition="${pdict.talon.loyalty.isLoyaltyEnabled}">  
136             <tr class="order-sales-tax">  
137                 <td><print value="${pdict.talon.loyalty.loyaltyTitle}" /></td>  
138                 <td><print value="${pdict.talon.loyalty.loyaltyNetChange}" /></td>  
139             </tr>  
140         </if>  
141         <if condition="${pdict.talon.referral.isReferralEnabled && pdict.talon.isConfirmationPage && pdict.talon.referral.code}">  
142             <tr class="order-sales-tax">  
143                 <td><print value="${Resource.msg('referral.summary.referralApplied','talonOne',null)}" /></td>  
144                 <td><print value="${pdict.talon.referral.code}" /></td>  
145             </tr>  
146         </if>  
147         <!--comment>TalonOne End</comment>  
148     </tr>  
149     </tbody>
```

3.3.2 Controller Modification

If using a controller based SiteGenesis integration, additionally follow the instructions in this chapter.

3.3.2.1 Account.js

a) Add code:-

```
// Talonone - Begin: Calling the talonone customer profile
    var talonOneHelper =
require('*cartridge/scripts/helper/talonOneHelper');
    talonOneHelper.createCustomerProfile(request.locale);
// Talonone - End: Calling the talonone customer profile
```

Add this after if (isProfileUpdateValid && hasEditSucceeded) condition as shown below:-

```
129     if (isProfileUpdateValid && hasEditSucceeded) {
130         // Talonone - Begin: Calling the talonone customer profile
131         var talonOneHelper = require('~/cartridge/scripts/helper/talonOneHelper');
132         talonOneHelper.createCustomerProfile(request.locale);
133         // Talonone - End: Calling the talonone customer profile
134         response.redirect(URLUtils_https('Account-Show'));
135     } else { Unnikrishnan CC, last month • As a business we want to share customer data with...
136         response.redirect(URLUtils_https('Account-EditProfile', 'invalid', 'true'));
137     }

```

b) Add Code:-

```
// Talonone - Begin: Calling the talonone customer profile
    var talonOneHelper =
require('*cartridge/scripts/helper/talonOneHelper');
    talonOneHelper.createCustomerProfile(request.locale);
// Talonone - End: Calling the talonone customer profile
```

Add this after target = session.custom.TargetLocation: as shown below:-

```
433         app.getForm('profile').clear();
434         target = session.custom.TargetLocation;
435         // Talonone - Begin: Calling the talonone customer profile
436         var talonOneHelper = require('/cartridge/scripts/helper/talonOneHelper');
437         talonOneHelper.createCustomerProfile(request.locale);
438         // Talonone - End: Calling the talonone customer profile
439         if (target) {
440             delete session.custom.TargetLocation;
441             //TODO make sure only path, no hosts are allowed as redirect target
442             dw.system.Logger.info('Redirecting to "{0}" after successful login', target);
443             response.redirect(target);
444         } else {
445             response.redirect(URLUtils.https('Account-Show', 'registration', 'true'));
446         }

```

3.2.2.2 Cart.js

a) Add Code

```
var TalonOnePreferences =
require('*cartridge/scripts/util/talonOnePreferences');
var talonOneHelper = require('*cartridge/scripts/helper/talonOneHelper');
```

Include this code at the top as shown below:-

```
14 var Transaction = require('dw/system/Transaction');
15 var URLUtils = require('dw/web/URLUtils');
16
17 /* Script Modules */
18 var app = require('~cartridge/scripts/app');
19 var guard = require('~cartridge/scripts/guard');
20
21 var TalonOnePreferences = require('*cartridge/scripts/util/talonOnePreferences');
22 var talonOneHelper = require('*cartridge/scripts/helper/talonOneHelper');
23
24 /**
25 * Redirects the user to the last visited catalog URL if known, otherwise redirects to
26 * a hostname-only URL if an alias is set, or to the Home>Show controller function in the default
27 * format using the HTTP protocol.| You, 3 weeks ago • TALBLD-23 Updating the cart calculation in SFCC
28 */
29 function continueShopping() {
30
31     var location = require('~cartridge/scripts/util/Browsing').lastCatalogURL();
32
33     if (location) {
```

b) In show() add the following code block:

```
// Talonone - Begin:
    var talon = talonOneHelper.talonConfig(false, 'Cart',
cart.object.custom.referralCode);
// Talonone - End:

    app.getView('Cart', {
        cart: app.getModel('Cart').get(),
        RegistrationStatus: false,
        talon: talon
    }).render('checkout/cart/cart');
```

As shown below:-

```
40 /**
41 * Invalidates the login and shipment forms. Renders the checkout/cart/cart template.
42 */
43 function show() {
44     var cartForm = app.getForm('cart');
45     app.getForm('login').invalidate();

46     cartForm.get('shipments').invalidate();
47     var cart = app.getModel('Cart').get();

48
49     // Talonone - Begin:
50     var talon = talonOneHelper.talonConfig(false, 'Cart', cart.object.custom.referralCode);
51     // Talonone - End:

52
53     app.getView('Cart', {
54         cart: app.getModel('Cart').get(),
55         RegistrationStatus: false,
56         talon: talon
57     }).render('checkout/cart/cart');

58 }
| You, 3 weeks ago • TALBLD-23 Updating the cart calculation in SFCC a..
```

c) Add the following code block after function show()

```
// Talonone - Begin:
/**
 * For remove talon referral
 */
function removeReferral() {
    var cart = app.getModel('Cart').goc();
    Transaction.wrap(function () {
        cart.object.custom.referralCode = '';
        cart.calculate();
    });
    var url = URLUtils.url('Cart-Show').toString();
    response.redirect(url);
}

// Talonone - End:
```

As shown below:-

```
43     function show() {
44         var cartForm = app.getForm('cart');
45         app.getForm('login').invalidate();
46
47         cartForm.get('shipments').invalidate();
48         var cart = app.getModel('Cart').goc();
49
50         // Talonone - Begin:
51         var talon = talonOneHelper.talonConfig(false, 'Cart', cart.object.custom.referralCode);
52         // Talonone - End:
53
54         app.getView('Cart', {
55             cart: app.getModel('Cart').get(),
56             RegistrationStatus: false,
57             talon: talon
58         }).render('checkout/cart/cart');
59
60     }
61     // Talonone - Begin:
62     /**
63      * For remove talon referral
64      */
65     function removeReferral() {
66         var cart = app.getModel('Cart').goc();
67         Transaction.wrap(function () {
68             cart.object.custom.referralCode = '';
69             cart.calculate();
70         });
71         var url = URLUtils.url('Cart-Show').toString();
72         response.redirect(url);
73
74     }
75     // Talonone - End:
```

d) Add the following code block after addCoupon function

```
// TalonOne Start: Referrals Points
    //Add a referral if a referral was entered correctly and is active.
    'addReferral': function (formgroup) {
        var result = {
            cart: cart,
            EnableCheckout: true,
            dontRedirect: true
        };
    }
}
```

```

        if (TalonOnePreferences.isEnabled() &&
TalonOnePreferences.isReferralEnabled()) {
    var CSRFProtection = require('dw/web/CSRFProtection');

    if (!CSRFProtection.validateRequest()) {
        app.getModel('Customer').logout();
        app.getView().render('csrf/csrffailed');
        return null;
    }

    if (formgroup.referralCode.htmlValue) {
        //status =
cart.addCoupon(formgroup.referralCode.htmlValue);

        if (cart.object.custom.referralCode) {
            result.ReferralStatus = {
                code: Resource.msg('error.referral.used',
'talonOne', null),
                error: true
            };
            return result;
        }

        Transaction.wrap(function () {
            cart.object.custom.referralCode =
formgroup.referralCode.htmlValue;
        });
    }

    var applyReferral =
talonOneHelper.addTalononeReferral(cart.object, 'open');
    if (applyReferral) {
        result.ReferralStatus = {
            code: applyReferral.message,
            error: applyReferral.error
        };
        if (!applyReferral.error){
            Transaction.wrap(function (){
                cart.calculate();
            });
        } else {
            Transaction.wrap(function () {
                cart.object.custom.referralCode = '';
            });
        }
    } else {
        // no status means valid but inactive referral
        result.ReferralError = 'NO_ACTIVE_PROMOTION';
    }
} else {
    // no coupon code supplied
    result.ReferralError = 'REFERRAL_CODE_MISSING';
}
}

```

```

        } else {
            result.ReferralStatus = {
                code: Resource.msg('error.referral.disabled',
'talonOne', null),
                error: true
            };
        }
        return result;
    },

```

As shown below

```

150     return result;
151 },
152     // TalonOne Start: Referrals Points
153     //Add a referral if a referral was entered correctly and is active.
154     'addReferral': function (formgroup) {          Unnikrishnan CC, last week • Latest Pull merged
155         var result = {
156             cart: cart,
157             EnableCheckout: true,
158             dontRedirect: true
159         };
160
161         if (TalonOnePreferences.isEnabled() && TalonOnePreferences.isReferralEnabled()) {
162             var CSRFProtection = require('dw/web/CSRFProtection');
163
164             if (!CSRFProtection.validateRequest()) {
165                 app.getModel('Customer').logout();
166                 app.getView().render('csrf/csrffailed');
167                 return null;
168             }
169
170             if (formgroup.referralCode.htmlValue) {
171                 //status = cart.addCoupon(formgroup.referralCode.htmlValue);
172
173                 if (cart.object.custom.referralCode) {
174                     result.ReferralStatus = {
175                         code: Resource.msg('error.referral.used', 'talonOne', null),
176                         error: true
177                     };
178                     return result;
179                 }
180
181                 Transaction.wrap(function () {
182                     cart.object.custom.referralCode = formgroup.referralCode.htmlValue;
183                 });
184
185                 var applyReferral = talonOneHelper.addTalononeReferral(cart.object, 'open');
186                 if (applyReferral) {
187                     result.ReferralStatus = {
188                         code: applyReferral.message,
189                         error: applyReferral.error
190                     };
191                     if (!applyReferral.error){
192                         Transaction.wrap(function (){
193                             cart.calculate();
194                         });
195                     } else {
196                         Transaction.wrap(function () {
197                             cart.object.custom.referralCode = '';
198                         });
199                     }
200                 } else {
201                     // no status means valid but inactive referral
202                     result.ReferralError = 'NO_ACTIVE_PROMOTION';
203                 }
204             } else {
205                 // no coupon code supplied
206                 result.ReferralError = 'REFERRAL_CODE_MISSING';
207             }
208         } else {
209             result.ReferralStatus = {
210                 code: Resource.msg('error.referral.disabled', 'talonOne', null),
211                 error: true
212             };
213         }
214         return result;
215     },
216     'calculateTotal': function () {
217         // nothing to do here - calculate has been done
218     }

```

e) Add the following code block on deleteCoupon function

Talon.One Controllers SG Integration v105.2.0

```

// Talonone - Begin: Removing/Updating free lineitem based on the talonone
response effects. Also updating the basket custom attribute with coupon codes
    if (TalonOnePreferences.isEnabled()) {
        var couponLineItem =
formgroup.getTriggeredAction().object;
        talonOneHelper.removeFreeLineItemFromBasket(cart.object,
couponLineItem);

        var couponCodes = [];
        if (!empty(cart.object.custom.talononeCouponCodes)) {
            var appliedCoupons =
cart.object.custom.talononeCouponCodes;
            appliedCoupons.forEach(function (key) {
                if (key !== couponLineItem.couponCode) {
                    couponCodes.push(key);
                }
            });
        }

        cart.object.custom.talononeCouponCodes =
couponCodes;
    }

        cart.removeCouponLineItem(formgroup.getTriggeredAction()
.object);
        cart.calculate();
    } else {
        cart.removeCouponLineItem(formgroup.getTriggeredAction()
.object);
    }

// Talonone - End: Removing/Updating free lineitem based on
the talonone response effects. Also updating the basket custom attribute with
coupon codes

```

As shown below:-

```

244     'deleteCoupon': function (formgroup) {
245         transaction.wrap(function () {
246
247             // Talonone - Begin: Removing/Updating free lineitem based on the talonone response effects. Also updating the basket custom attribute with coupon codes
248             if (TalonOnePreferences.isEnabled()) {
249                 var couponLineItem = formgroup.getTriggeredAction().object;
250                 talonOneHelper.removeFreeLineItemFromBasket(cart.object, couponLineItem);
251
252                 var couponCodes = [];
253                 if (!empty(cart.object.custom.talononeCouponCodes)) {
254                     var appliedCoupons = cart.object.custom.talononeCouponCodes;
255                     appliedCoupons.forEach(function (key) {
256                         if (key !== couponLineItem.couponCode) {
257                             couponCodes.push(key);
258                         }
259                     });
260
261                     cart.object.custom.talononeCouponCodes = couponCodes;
262
263                     cart.removeCouponLineItem(formgroup.getTriggeredAction().object);
264                     cart.calculate();
265                 } else {
266                     cart.removeCouponLineItem(formgroup.getTriggeredAction().object);
267                 }
268             }
269             // Talonone - End: Removing/Updating free lineitem based on the talonone response effects. Also updating the basket custom attribute with coupon codes
270
271         });
272
273         return {
274             cart: cart
275         };
276     },

```

f) Add the following code block on deleteProduct function

```

// Talonone - Begin: On Talonone free item product lineitem removal, we are
// updating the rejected free item id's to Talonone request.
    if (TalonOnePreferences.isEnabled()) {
        talonOneHelper.talononeRejectedFreeItem(cart.object,
formgroup.getTriggeredAction().object);
    }
    // Talonone - End: On Talonone free item product lineitem
removal, we are updating the rejected free item id's to Talonone request.

```

As shown below :-

```

286     'deleteProduct': function (formgroup) {
287
288         // Talonone - Begin: On Talonone free item product lineitem removal, we are updating the rejected free item id's to Talonone request.
289         if (TalonOnePreferences.isEnabled()) {
290             talonOneHelper.talononeRejectedFreeItem(cart.object, formgroup.getTriggeredAction().object);
291         }
292         // Talonone - End: On Talonone free item product lineitem removal, we are updating the rejected free item id's to Talonone request.
293
294         Transaction.wrap(function () {
295             cart.removeProductLineItem(formgroup.getTriggeredAction().object);
296         });
297
298         You, 3 weeks ago * TALBLD-23 Updating the cart calculation in SFCC a...
299
300         return {
301             cart: cart
302         };
303     },

```

g) Add the following code block on updateCart function

```

// Talonone - Begin: On Talonone free item product lineitem removal, we are
// updating the rejected free item id's to Talonone request.
    if (TalonOnePreferences.isEnabled()) {
        talonOneHelper.talononeRejectedFreeItem(cart
.object, item.object);
    }
    // Talonone - End: On Talonone free item product
lineitem removal, we are updating the rejected free item id's to Talonone
request.

```

As shown below:-

```

315     'updateCart': function () {
316
317         Transaction.wrap(function () {
318             var shipmentitem, item;
319
320             // remove zero quantity line items
321             for (var i = 0; i < session.forms.cart.shipments.childCount; i++) {
322                 shipmentitem = session.forms.cart.shipments[i];
323
324                 for (var j = 0; j < shipmentitem.items.childCount; j++) {
325                     item = shipmentitem.items[j];
326
327                     if (item.quantity.value === 0) {
328                         // Talonone - Begin: On Talonone free item product lineitem removal, we are updating the rejected free item id's to Talonone request.
329                         if (TalonOnePreferences.isEnabled()) {
330                             talonOneHelper.talononeRejectedFreeItem(cart.object, item.object);
331                         }
332                         // Talonone - End: On Talonone free item product lineitem removal, we are updating the rejected free item id's to Talonone request.
333
334                         cart.removeProductLineItem(item.object);
335
336                     }
337                 }
338
339                 session.forms.cart.shipments.accept();
340                 cart.checkInStoreProducts();
341             });
342
343             return {
344                 cart: cart,
345                 EnableCheckout: true
346             };
347         },

```

h) Add the following code block on submitForm()

```

// Talonone - Begin:
    var talon = talonOneHelper.talonConfig(false, 'Cart',
formResult.cart.object.custom.referralCode);

```

```

// Talonone - End:

if (formResult.dontRedirect) {
    app.getView({
        Basket: formResult.cart.object,
        EnableCheckout: formResult.EnableCheckout,
        CouponStatus: formResult.CouponStatus,
        CouponError: formResult.CouponError,
        ReferralStatus: formResult.ReferralStatus,
        ReferralError: formResult.ReferralError,
        talon: talon
    }).render('checkout/cart/cart');
} else {
    response.redirect(URLUtils_https('Cart-Show'));
}

```

As shown below:-

```

352
353     if (formResult) {
354         cartAsset = app.getModel('Content').get('cart');
355
356         pageMeta = require('~cartridge/scripts/meta');
357         pageMeta.update(cartAsset);
358         // Talonone - Begin:
359         var talon = talonOneHelper.talonConfig(false, 'Cart', formResult.cart.object.custom.referralCode);
360         // Talonone - End:
361
362         if (formResult.dontRedirect) {
363             app.getView([
364                 {Basket: formResult.cart.object, You, 3 weeks ago * TALBLD-23 Updating the cart calculation in SFCC a...},
365                 {EnableCheckout: formResult.EnableCheckout, },
366                 {CouponStatus: formResult.CouponStatus, },
367                 {CouponError: formResult.CouponError, },
368                 {ReferralStatus: formResult.ReferralStatus, },
369                 {ReferralError: formResult.ReferralError, },
370                 {talon: talon}
371             ]).render('checkout/cart/cart');
372         } else {
373             response.redirect(URLUtils_https('cart-show'));
374         }
375     }
376 }
377

```

i) Add the following code block at the end:-

```

/** Remove Referral
 * @see {@link module:controllers/Cart~removeReferral} */
exports.removeReferral = guard.ensure(['get'], removeReferral);

```

As shown below:-

```

594     exports.AddToWishlist = guard.ensure(['get', 'https', 'loggedIn'], addToWishlist, {
595         scope: 'wishlist'
596     });
597     /** Adds bonus product to cart.
598     * @see {@link module:controllers/Cart~addBonusProductJson} */
599     exports.AddBonusProduct = guard.ensure(['post'], addBonusProductJson);
600     /** Remove Referral
601     * @see {@link module:controllers/Cart~removeReferral} */
602     exports.removeReferral = guard.ensure(['get'], removeReferral);
603

```

3.2.2.3 COBilling.js

a) Add the following code block on returnToForm function

```
// Talonone - Begin:
```

```

var talonOneHelper =
require('*-/cartridge/scripts/helper/talonOneHelper');
var talon = talonOneHelper.talonConfig(false, '', '');
// Talonone - End:

if (params) {
    app.getView(require('~/cartridge/scripts/object').extend(params, {
        Basket: cart.object,
        talon: talon,
        ContinueURL: URLUtils.https('COBilling-Billing')
    })).render('checkout/billing/billing');
} else {
    app.getView({
        Basket: cart.object,
        talon: talon,
        ContinueURL: URLUtils.https('COBilling-Billing')
    }).render('checkout/billing/billing');
}

```

As shown below:-

```

79 function returnToForm(cart, params) {
80     var pageMeta = require('~/cartridge/scripts/meta');
81
82     // if the payment method is set to gift certificate get the gift certificate code from the form
83     if (!empty(cart.getPaymentInstrument()) && cart.getPaymentInstrument().getPaymentMethod() === PaymentInstrument.METHOD_GIFT_CERTIFICATE) {
84         app.getForm('billing').copyFrom({
85             giftCode: cart.getPaymentInstrument().getGiftCertificateCode()
86         });
87     }
88
89     pageMeta.update({
90         pageTitle: Resource.msg('billing.meta.pagetitle', 'checkout', 'SiteGenesis Checkout')
91     });
92
93     // Talonone - Begin:
94     var talonOneHelper = require('*-/cartridge/scripts/helper/talonOneHelper');
95     var talon = talonOneHelper.talonConfig(false, '', '');
96     // Talonone - End:
97
98     if (params) {
99         app.getView(require('~/cartridge/scripts/object').extend(params, [
100             {Basket: cart.object}], Unikrishnan CC, 3 weeks ago * As a business we want to update the cart in SFCC ...
101             talon: talon,
102             ContinueURL: URLUtils.https('COBilling-Billing')
103         ])).render('checkout/billing/billing');
104     } else {
105         app.getView({
106             Basket: cart.object,
107             talon: talon,
108             ContinueURL: URLUtils.https('COBilling-Billing')
109         }).render('checkout/billing/billing');
110     }
111 }

```

b) Add the following code block on updateSummary function

```

// Talonone - Begin:
var talonOneHelper =
require('*-/cartridge/scripts/helper/talonOneHelper');
var talon = talonOneHelper.talonConfig(false, '', '');
// Talonone - End:

app.getView({
    checkoutstep: 4,
    Basket: cart.object,
    talon: talon
}).render('checkout/minisummary');

```

As shown below :-

```

614 /**
615  * Updates the order totals and recalculates the basket after a coupon code is applied.
616  * Renders the checkout/minisummary template, which includes the mini cart order totals and shipment summary.
617 */
618 function updateSummary() {
619
620     var cart = app.getModel('Cart').get();
621
622     Transaction.wrap(function () {
623         cart.calculate();
624     });
625
626     // Talonone - Begin:
627     var talonOneHelper = require('*cartridge/scripts/helper/talonOneHelper');
628     var talon = talonOneHelper.talonConfig(false, '', '');
629     // Talonone - End:
630
631     app.getView({
632         checkoutstep: 4,
633         Basket: cart.object,
634         talon: talon
635     }).render('checkout/minisummary');
636 }
637

```

3.2.2.4 COShipping.js

- a) Add the following code block on start function

```

// Talonone - Begin:
var talonOneHelper =
require('*cartridge/scripts/helper/talonOneHelper');
var talon = talonOneHelper.talonConfig(false, '', '');
// Talonone - End:

// Go to billing step, if we have no product line items, but only gift
certificates in the basket, shipping is not required.
if (cart.getProductLineItems().size() === 0) {
    app.getController('COBilling').Start();
} else {
    pageMeta = require('~/cartridge/scripts/meta');
    pageMeta.update({
        pageTitle: Resource.msg('singleshipping.meta.pagetitle',
'checkout', 'SiteGenesis Checkout')
    });
    app.getView({
        ContinueURL: URLUtils.https('COShipping-SingleShipping'),
        Basket: cart.object,
        HomeDeliveries: homeDeliveries,
        talon: talon
    }).render('checkout/shipping/singleshipping');
}

```

As shown below:-

```

97     Transaction.wrap(function () {
98         cart.calculate();
99     });
100
101    // Talonone - Begin:
102    var talonOneHelper = require('/cartridge/scripts/helper/talonOneHelper');
103    var talon = talonOneHelper.talonConfig(false, '', '');
104    // Talonone - End:
105
106    // Go to billing step, if we have no product line items, but only gift certificates in the basket, shipping is not required.
107    if (cart.getProductLineItems().size() === 0) {
108        app.getController('COBilling').Start();
109    } else {
110        pageMeta = require('~/cartridge/scripts/meta');
111        pageMeta.update({
112            pageTitle: Resource.msg('singleShipping.meta.pageTitle', 'checkout', 'SiteGenesis Checkout')
113        });
114        app.getView({
115            ContinueURL: URLUtils_https('COShipping-SingleShipping'),
116            Basket: cart.object,
117            HomeDeliveries: homeDeliveries,
118            talon: talon
119        }).render('checkout/shipping/singleShipping');
120    }
121
122    /**
123     * Handles the selected shipping address and shipping method. Copies the
124     * address details and gift options to the basket's default shipment. Sets the
125     * selected shipping method to the default shipment.
126     *
127     * @transactional
128     * @param {module:models/CartModel~CartModel} cart - A CartModel wrapping the current Basket.
129     */
130
131    function handleShippingSettings(cart) {
132        Transaction.wrap(function () {

```

b) Add the following code block on updateShippingMethodList

```

session.custom.bypassTalonOneCalculation = true;
for (i = 0; i < applicableShippingMethods.length; i++) {
    method = applicableShippingMethods[i];

    cart.updateShipmentShippingMethod(cart.getDefaultShipment().getID(),
method.getID(), method, applicableShippingMethods);
    cart.calculate();
    shippingCosts.put(method.getID(),
cart.preCalculateShipping(method));
}
delete session.custom.bypassTalonOneCalculation;

```

As shown below:-

```

330    applicableShippingMethods = cart.getApplicableShippingMethods(address);
331    shippingCosts = new HashMap();
332    currentShippingMethod = cart.getDefaultShipment().getShippingMethod() || ShippingMgr.getDefaultShippingMethod();
333
334    // Transaction controls are for fine tuning the performance of the data base interactions when calculating shipping methods
335    Transaction.begin();
336
337    session.custom.bypassTalonOneCalculation = true;
338    for (i = 0; i < applicableShippingMethods.length; i++) {
339        method = applicableShippingMethods[i];
340
341        cart.updateShipmentShippingMethod(cart.getDefaultShipment().getID(), method.getID(), method, applicableShippingMethods);
342        cart.calculate();
343        shippingCosts.put(method.getID(), cart.preCalculateShipping(method));
344    }
345    delete session.custom.bypassTalonOneCalculation;
346
347    Transaction.rollback();
348
349    Transaction.wrap(function () {
350        cart.updateShipmentShippingMethod(cart.getDefaultShipment().getID(), currentShippingMethod.getID(), currentShippingMethod, applicableShippingMethods);
351        cart.calculate();
352    });
353
354    session.forms.singleShipping.shippingAddress.shippingMethodID.value = cart.getDefaultShipment().getShippingMethodID();
355

```

3.2.2.5 COSummary.js

a) Add the following code block on start function

```

// Talonone Loyalty - Begin:
    var talonOneHelper =
require('*cartridge/scripts/helper/talonOneHelper');
    var talon = talonOneHelper.talonConfig(false, '', '');
    // Talonone Loyalty - End:

    var pageMeta = require('~/cartridge/scripts/meta');
    var viewContext =
require('app_storefront_core/cartridge/scripts/common/extend').immutable(context,
{
    Basket: cart.object,
    talon: talon
});

```

As shown below:-

```

27  function start(context) {
28      var cart = Cart.get();
29
30      // Checks whether all payment methods are still applicable. Recalculates all existing non-gift certificate payment
31      // instrument totals according to redeemed gift certificates or additional discounts granted through coupon
32      // redemptions on this page.
33      var COBilling = app.getController('COBilling');
34      if (!COBilling.validatePayment(cart)) {
35          COBilling.Start();
36          return;
37      } else {
38          Transaction.wrap(function () {
39              cart.calculate();
40          });
41
42          Transaction.wrap(function () {
43              if (!cart.calculatePaymentTransactionTotal()) {
44                  COBilling.Start();
45              }
46          });
47
48      // Talonone Loyalty - Begin:
49      var talonOneHelper = require('*cartridge/scripts/helper/talonOneHelper');
50      var talon = talonOneHelper.talonConfig(false, '', '');
51      // Talonone Loyalty - End:
52
53      var pageMeta = require('~/cartridge/scripts/meta');
54      var viewContext = require('app_storefront_core/cartridge/scripts/common/extend').immutable(context, {
55          Basket: cart.object,
56          talon: talon
57      });
58      pageMeta.update({pageTitle: Resource.msg('summary.meta.pagetitle', 'checkout', 'SiteGenesis Checkout')});
59      app.getView(viewContext).render('checkout/summary/summary');
60  }

```

b) Add the following code block on submit function

```

// Talonone - Begin: Calling the talonone customer session on closed state
    var talonOneHelper =
require('*cartridge/scripts/helper/talonOneHelper');
    talonOneHelper.updateCloseOrderSession(placeOrderResult.Order);
    // Talonone - End: Calling the talonone customer session on closed
state

```

As shown below:-

```

63 /**
64 * This function is called when the "Place Order" action is triggered by the
65 * customer.
66 */
67 function submit() {
68     // Calls the COPlaceOrder controller that does the place order action and any payment authorization.
69     // COPlaceOrder returns a JSON object with an order_created key and a boolean value if the order was created successfully.
70     // If the order creation failed, it returns a JSON object with an error key and a boolean value.
71     var placeOrderResult = app.getController('COPlaceOrder').Start();
72     if (placeOrderResult.error) {
73         start({
74             | PlaceOrderError: placeOrderResult.PlaceOrderError
75         });
76     } else if (placeOrderResult.order_created) {
77
78         // Talonone - Begin: Calling the talonone customer session on closed state
79         var talonOneHelper = require('*cartridge/scripts/helper/talonOneHelper');
80         talonOneHelper.updateCloseOrderSession(placeOrderResult.order);
81         // Talonone - End: Calling the talonone customer session on closed state
82
83         showConfirmation(placeOrderResult.order);
84     }
85 }
86

```

c) Add the following code block on showConfirmation function

```

// Talonone - Begin:
var talonOneHelper =
require('*cartridge/scripts/helper/talonOneHelper');
var talon = talonOneHelper.talonConfig(true, '',
order.custom.referralCode);
// Talonone - End:

var pageMeta = require('~cartridge/scripts/meta');
pageMeta.update({pageTitle: Resource.msg('confirmation.meta.pagetitle',
'checkout', 'SiteGenesis Checkout Confirmation')});
app.getView({
    Order: order,
    talon: talon,
    ContinueURL: URLUtils.https('Account-RegistrationForm') // needed by
registration form after anonymous checkouts
}).render('checkout/confirmation/confirmation');

```

As shown below:-

```

93 function showConfirmation(order) {
94     if (!customer.authenticated) {
95         // Initializes the account creation form for guest checkouts by populating the first and last name with the
96         // used billing address.
97         var customerForm = app.getForm('profile.customer');
98         customerForm.setValue('firstname', order.billingAddress.firstName);
99         customerForm.setValue('lastname', order.billingAddress.lastName);
100        customerForm.setValue('email', order.customerEmail);
101        customerForm.setValue('orderNo', order.orderNo);
102        customerForm.setValue('orderUUID', order.getUUID());
103    }
104
105    app.getForm('profile.login.passwordconfirm').clear();
106    app.getForm('profile.login.password').clear();
107
108    // Talonone - Begin:
109    var talonOneHelper = require('*cartridge/scripts/helper/talonOneHelper');
110    var talon = talonOneHelper.talonConfig(true, '', order.custom.referralCode);
111    // Talonone - End:
112
113    var pageMeta = require('~cartridge/scripts/meta');
114    pageMeta.update({pageTitle: Resource.msg('confirmation.meta.pagetitle', 'checkout', 'SiteGenesis Checkout Confirmation')});
115    app.getView({
116        Order: order,
117        talon: talon,
118        ContinueURL: URLUtils.https('Account-RegistrationForm') // needed by registration form after anonymous checkouts
119    }).render('checkout/confirmation/confirmation');
120

```

3.2.2.5 Login.js

Add the following code block on handleLoginForm function

```
// Talonone - Begin: Calling the talonone customer profile
    var talonOneHelper =
require('*cartridge/scripts/helper/talonOneHelper');
    talonOneHelper.createCustomerProfile(request.locale);
// Talonone - End: Calling the talonone customer profile
```

As shown below:-

```
93  function handleLoginForm () {
94      var loginForm = app.getForm('login');
95
96      loginForm.handleAction({
97          login: function () {
98              // Check to see if the number of attempts has exceeded the session threshold
99              if (RateLimiter.isOverThreshold('FailedLoginCounter')) {
100                  RateLimiter.showCaptcha();
101              }
102
103              var success = Customer.login(loginForm.getValue('username'), loginForm.getValue('password'), loginForm.getValue('rememberme'));
104
105              if (!success) {
106                  loginForm.get('loginsucceeded').invalidate();
107                  app.getView('Login').render();
108                  return;
109              } else {
110                  loginForm.clear();      Unnikrishnan cc, last month • As a business we want to share customer data with...
111              }
112
113              RateLimiter.hideCaptcha();
114              // Talonone - Begin: Calling the talonone customer profile
115              var talonOneHelper = require('*cartridge/scripts/helper/talonOneHelper');
116              talonOneHelper.createCustomerProfile(request.locale);
117              // Talonone - End: Calling the talonone customer profile
118
119              // In case of successful login
120              // Redirects to the original controller that triggered the login process.
121              response.redirect(getTargetUrl());
122
123          }
124      });
125  },
```

3.3.3 Model Modification

The following model changes should be made regardless of whether a controller integration approach are being used:

3.3.3.1 CartModel.js

a) Add the following code block at the top.

```
var TalonOnePreferences =
require('*cartridge/scripts/util/talonOnePreferences');
```

As shown below:-

```
22  var ShippingMgr = require('dw/order/ShippingMgr');
23  var StoreMgr = require('dw/catalog/StoreMgr');
24  var TransientAddress = require('~cartridge/scripts/models/TransientAddressModel');
25  var UUIDUtils = require('dw/util/UUIDUtils');
26  var TalonOnePreferences = require('*cartridge/scripts/util/talonOnePreferences');
```

b) Add the following code block on addProductItem function

```
/**
```

```

        * Adds a product to the cart and recalculates the cart.
        * By default, when a bundle is added to cart, all its child products
are added too, but if those products are
        * variants then the code must replace the master products with the
selected variants that are passed in the
        * HTTP params as childPids along with any options.
        * @params {request.httpParameterMap.childPids} - comma separated list
of
        * product IDs of the bundled products that are variations.
        *
        * @transactional
        * @alias module:models/CartModel~CartModel/addProductItem
        * @param {String} pid - ID of the product that is to be added to the
basket.
        * @param {Number} quantity - The quantity of the product.
        * @param {dw.catalog.ProductOptionModel} productOptionModel - The
option model of the product that is to be added to the basket.
        * @param {Boolean} isFreeDiscountItem - The value set to identify
whether the product are added from Talonone as a free product item.
        */
    addProductItem: function (product, quantity, productOptionModel,
isFreeDiscountItem) {
        var cart = this;
        Transaction.wrap(function () {
            var i;
            if (product) {
                var productInCart;
                var productLineItem;
                var productLineItems = cart.object.productLineItems;
                var quantityInCart;
                var quantityToSet;
                var shipment = cart.object.defaultShipment;

                for (var q = 0; q < productLineItems.length; q++) {
                    if (productLineItems[q].productID === product.ID) {
                        productInCart = productLineItems[q];
                        break;
                    }
                }

                if (productInCart) {
                    if (productInCart.optionModel) {
                        productLineItem =
cart.createProductLineItem(product, productOptionModel, shipment);
                        if (quantity) {
                            productLineItem.setQuantityValue(quantity);
                        }
                    } else {
                        quantityInCart = productInCart.getQuantity();
                        quantityToSet = quantity ? quantity + quantityInCart
: quantityInCart + 1;
                        productInCart.setQuantityValue(quantityToSet);
                    }
                }
            }
        })
    }
}

```

```

        }
    } else {
        productLineItem = cart.createProductLineItem(product,
productOptionModel, shipment);

        if (quantity) {
            productLineItem.setQuantityValue(quantity);
        }
    }

    /**
     * By default, when a bundle is added to cart, all its child
products are added too, but if those products are
     * variants then the code must replace the master products
with the selected variants that get passed in the
     * HTTP params as childPids along with any options. Params:
CurrentHttpParameterMap.childPids - comma separated list of
     * pids of the bundled products that are variations.
     */
    if (request.httpParameterMap.childPids.stringValue &&
product.bundle) {
        var childPids =
request.httpParameterMap.childPids.stringValue.split(',');

        for (i = 0; i < childPids.length; i++) {
            var childProduct = Product.get(childPids[i]).object;

            if (childProduct) {
                childProduct.updateOptionSelection(request.httpP
arameterMap);

                var foundLineItem =
this.getBundledProductLineItemByPID(lineItem, childProduct.isVariant() ?
childProduct.masterProduct.ID : childProduct.ID);

                if (foundLineItem) {
                    foundLineItem.replaceProduct(childProduct);
                }
            }
        }
    }

    // Talonone - Begin: Avoiding unnecessary cart calculate
calls, to avoid quota limitation while free items are added to cart based on
talonone response effect
    if (!(TalonOnePreferences.isEnabled() &&
isFreeDiscountItem)) {
        cart.calculate();
    }
    // Talonone - End: Avoiding unnecessary cart calculate
calls, to avoid quota limitation while free items are added to cart based on
talonone response effect

```

```

        }
    });
},

```

As shown below:-

```

162 /**
163 * Adds a product to the cart and recalculates the cart.
164 * By default, when a bundle is added to cart, all its child products are added too, but if those products are
165 * variants then the code must replace the master products with the selected variants that are passed in the
166 * HTTP params as childPids along with any options.
167 * @param {request.httpParameterMap.childPids} - comma separated list of
168 * product IDs of the bundled products that are variations.
169 *
170 * @transactional
171 * @alias module:models/CartModel~CartModel/addProductItem
172 * @param {String} pid - ID of the product that is to be added to the basket. You, 4 weeks ago • TALBLD-23 Updating the cart calculation in SFCC a...
173 * @param {Number} quantity - the quantity of the product.
174 * @param {dw.catalog.ProductoptionModel} productoptionModel - The option model of the product that is to be added to the basket.
175 * @param {Boolean} isFreeDiscountItem - The value set to identify whether the product are added from Talonone as a free product item.
176 */
177 addProductItem: function (product, quantity, productOptionModel, isFreeDiscountItem) {
178     var cart = this;
179     Transaction.wrap(function () {
180         var i;
181         if (product) {
182             var productInCart;
183             var productLineItem;
184             var productLineItems = cart.object.productLineItems;
185             var quantityInCart;
186             var quantityToSet;
187             var shipment = cart.object.defaultShipment;
188
189             for (var q = 0; q < productLineItems.length; q++) {
190                 if (productLineItems[q].productId === product.ID) {
191                     productInCart = productLineItems[q];
192                     break;
193                 }
194             }
195
196             if (productInCart) {
197                 if (productInCart.optionModel) {
198                     productLineItem = cart.createProductLineItem(product, productOptionModel, shipment);
199                     if (quantity) {
200                         productLineItem.setQuantityValue(quantity);
201                     }
202                 } else {
203                     quantityInCart = productInCart.getQuantity();
204                     quantityToSet = quantity ? quantity + quantityInCart : quantityInCart + 1;
205                     productInCart.setQuantityValue(quantityToSet);
206                 }
207             } else {
208                 productLineItem = cart.createProductLineItem(product, productOptionModel, shipment);
209
210                 if (quantity) {
211                     productLineItem.setQuantityValue(quantity);
212                 }
213             }
214
215             /**
216             * By default, when a bundle is added to cart, all its child products are added too, but if those products are
217             * variants then the code must replace the master products with the selected variants that get passed in the
218             * HTTP params as childPids along with any options. Params: CurrentHttpParameterMap.childPids - comma separated list of
219             * pids of the bundled products that are variations.
220             */
221             if (request.httpParameterMap.childPids.stringValue && product.bundle) {
222                 var childPids = request.httpParameterMap.childPids.stringValue.split(',');
223
224                 for (i = 0; i < childPids.length; i++) {
225                     var childProduct = Product.get(childPids[i]).object;
226
227                     if (childProduct) {
228                         childProduct.updateOptionsSelection(request.httpParameterMap);
229
230                         var foundLineItem = this.getBundledProductLineItemByPID(lineItem, childProduct.isVariant() ? childProduct.masterProduct.ID : childProduct.ID);
231
232                         if (foundLineItem) {
233                             foundLineItem.replaceProduct(childProduct);
234                         }
235                     }
236                 }
237
238             // Talonone - Begin: Avoiding unnecessary cart calculate calls, to avoid quota limitation while free items are added to cart based on talonone response effect
239             if (!TalonOnePreferences.isEnabled() && isFreeDiscountItem) {
240                 cart.calculate();
241             }
242             // Talonone - End: Avoiding unnecessary cart calculate calls, to avoid quota limitation while free items are added to cart based on talonone response effect
243         }
244     });
}

```

c) Add Code:-

```

// Talonone - Begin: Creating couponline item based on the talonone response
// effects
try {
    if (TalonOnePreferences.isEnabled()) {
        var talonOneHelper =
require('*cartridge/scripts/helper/talonOneHelper');

```

```

        addCouponToBasketResult =
talonOneHelper.addTalononeCoupon(cart.object, couponCode, 'open');

            if (!(addCouponToBasketResult instanceof
dw.order.CouponLineItem)) {
                return {CouponStatus: 'COUPON_CODE_UNKNOWN'};
            }
        } else {
            addCouponToBasketResult = Transaction.wrap(function () {
                return cart.object.createCouponLineItem(couponCode,
campaignBased);
            });
        }
    } catch (e) {
        if (e.errorCode) {
            return {CouponStatus: e.errorCode};
        }
        return {CouponStatus: 'COUPON_CODE_ALREADY_IN_BASKET'};
    }
    // Talonone - End: Creating couponline item based on the
talonone response effects

```

As shown below:-

```

294     addCoupon: function (couponCode) {
295         if (couponCode) {
296             var cart = this;
297             var campaignBased = true;
298             var addCouponToBasketResult;
299
300             // Talonone - Begin: Creating couponline item based on the talonone response effects
301             try {
302                 if (TalonOnePreferences.isEnabled()) {
303                     var talonOneHelper = require('/cartridge/scripts/helper/talonOneHelper');
304                     addCouponToBasketResult = talonOneHelper.addTalononeCoupon(cart.object, couponCode, 'open');
305
306                     if (!(addCouponToBasketResult instanceof dw.order.CouponLineItem)) {
307                         return {CouponStatus: 'COUPON_CODE_UNKNOWN'};
308                     }
309                 } else {
310                     addCouponToBasketResult = Transaction.wrap(function () {
311                         return cart.object.createCouponLineItem(couponCode, campaignBased);
312                     });
313                 }
314             } catch (e) {
315                 if (e.errorCode) {
316                     return {CouponStatus: e.errorCode};
317                 }
318                 return {CouponStatus: 'COUPON_CODE_ALREADY_IN_BASKET'};
319             }
320             // Talonone - End: Creating couponline item based on the talonone response effects
321
322             Transaction.wrap(function () {
323                 cart.calculate();
324             });
325             return {CouponStatus: addCouponToBasketResult.statusCode};
326         }

```

3.3.4 View Modification

The following view changes should be made regardless of whether a controller integration approach are being used:

3.3.4.1 CartView.js

Add the following code block at prepareView function.

```
// Talonone - Begin: Recalculating the cart items, when a discount free
lineitem are added with respect to talonone response effects
    var TalonOnePreferences =
require('*cartridge/scripts/util/talonOnePreferences');
    var cartHelper =
require('*cartridge/scripts/cart/cartHelpers');
    if (TalonOnePreferences.isEnabled()) {
        cartHelper.addFreeItemCartCalculate(cart);

        if (session.custom.isTalonOneAddFreeItemError) {
            this.isTalonOneAddFreeItemerror = true;
        }

        delete session.custom.isTalonOneAddFreeItemError;

        // Refreshes shipments.
        session.forms.cart.shipments.copyFrom(cart.shipments);
        // Refreshes coupons.
        session.forms.cart.coupons.copyFrom(cart.couponLineItems);
    }
// Talonone - End: Recalculating the cart items, when a discount
free lineitem are added with respect to talonone response effects
```

As shown below:-

```
27  prepareView: function () {
28
29
30  var cart = this.Basket;
31  if (cart) {
32
33      // Refreshes shipments.
34      session.forms.cart.shipments.copyFrom(cart.shipments);
35      // Refreshes coupons.
36      session.forms.cart.coupons.copyFrom(cart.couponLineItems);
37      // Refreshes the cart calculation.
38      Transaction.wrap(function () {
39          | Cart.get(cart).calculate();
40      });
41      You, 3 weeks ago * TALBLLD-23 Updating the cart calculation in SFCC a...
42
43      // Talonone - Begin: Recalculating the cart items, when a discount free lineitem are added with respect to talonone response effects
44      var TalonOnePreferences = require('*cartridge/scripts/util/talonOnePreferences');
45      var cartHelper = require('*cartridge/scripts/cart/cartHelpers');
46      if (TalonOnePreferences.isEnabled()) {
47          cartHelper.addFreeItemCartCalculate(cart);

48          if (session.custom.isTalonOneAddFreeItemError) {
49              this.isTalonOneAddFreeItemerror = true;
50          }

51          delete session.custom.isTalonOneAddFreeItemError;

52          // Refreshes shipments.
53          session.forms.cart.shipments.copyFrom(cart.shipments);
54          // Refreshes coupons.
55          session.forms.cart.coupons.copyFrom(cart.couponLineItems);
56      }
57
58      // Talonone - End: Recalculating the cart items, when a discount free lineitem are added with respect to talonone response effects
59
60      var validationResult = Cart.get(cart).validateForCheckout();
61      this.EnableCheckout = validationResult.EnableCheckout;
62      this.BasketStatus = validationResult.BasketStatus;
```

3.3.5 Form Modification

The following form changes should be made regardless of whether a controller integration approach are being used:

3.3.5.1 cart.xml

Add the following code block

After <field formid="couponCode" type="string" mandatory="false"/>

```
TalonOne Start: Referrals Points -->
  <field formid="referralCode" type="string" mandatory="false"/>
  <!--TalonOne End: Referrals Points -->
```

After <action formid="addCoupon" valid-form="false"/>

```
<!--TalonOne Start: Referrals Points -->
  <action formid="addReferral" valid-form="false"/>
  <!--TalonOne End: Referrals Points -->
```

As shown below

```
17      <action formid="deleteGiftCertificate" valid-form="false"/>
18    </list>
19  </list>
20
21  <!-- coupons -->
22  <list formid="coupons">
23    <action formid="deleteCoupon" valid-form="false"/>
24  </list>
25
26  <field formid="couponCode" type="string" mandatory="false"/>
27  <!--TalonOne Start: Referrals Points -->
28  <field formid="referralCode" type="string" mandatory="false"/>
29  <!--TalonOne End: Referrals Points -->      Unnikrishnan CC, last week • For TALBL-27 P
30
31  <!-- functional buttons -->
32  <action formid="calculateTotal" valid-form="false"/>
33  <action formid="addCoupon" valid-form="false"/>
34  <!--TalonOne Start: Referrals Points -->
35  <action formid="addReferral" valid-form="false"/>
36  <!--TalonOne End: Referrals Points -->
37  <action formid="updateCart" valid-form="false"/>
38  <action formid="checkoutCart" valid-form="false"/>
39  <action formid="continueShopping" valid-form="false"/>
40  <action formid="register" valid-form="false"/>
41  <action formid="unregistered" valid-form="false"/>
42  <action formid="logout" valid-form="false"/>
```

3.3.6 Scripts Modification

The following scripts changes should be made regardless of whether a controller integration approach are being used:

3.3.5.1 calculate.js

Add the following code block

At top

```
var HookMgr = require('dw/system/HookMgr');
var TalonOnePreferences =
require('*-/cartridge/scripts/util/talonOnePreferences');
```

As shown below:-

```
var HookMgr = require('dw/system/HookMgr');          You, last month • TALBLD-17 Integration A
var TalonOnePreferences = require('*-/cartridge/scripts/util/talonOnePreferences');

/**
```

Add the following code block

Before calculateTax(basket);

```
// Talonone - Begin: Calling talonone services for each calculate updation
if (TalonOnePreferences.isEnabled() &&
empty(session.custom.bypassTalonOneCalculation) &&
!session.custom.bypassTalonOneCalculation ) {
    HookMgr.callHook('dw.order.calculateDiscounts',
'calculateDiscounts', basket);
}
// Talonone - Begin: Calling talonone services for each calculate
updation
```

As shown below

```
73     PromotionMgr.applyDiscounts(basket);
74
75     // since we might have bonus product line items, we need to
76     // reset product prices
77     calculateProductPrices(basket);
78
79     // =====
80     // =====      CALCULATE TAX      =====
81     // =====
82
83     // Talonone - Begin: Calling talonone services for each calculate updation
84     if (TalonOnePreferences.isEnabled() && empty(session.custom.bypassTalonOneCalculation) && !session.custom.bypassTalonOneCalculation ) {
|       HookMgr.callHook('dw.order.calculateDiscounts', 'calculateDiscounts', basket);
}
// Talonone - Begin: Calling talonone services for each calculate updation
85
86     calculateTax(basket);
87     You, last month • TALBLD-17 Integration API Service frame work for ...
88
89     // =====
90     // =====      CALCULATE BASKET TOTALS      =====
91     // =====
92
93     basket.updateTotals();
94
95
```

3.3.5.2 hooks.json

Add the following code block at the end after {"name":/checkout/applepay.js"},

```
{
    "name": "dw.order.calculateDiscounts",
    "script":
"../../../../int_talonone/cartridge/scripts/helper/calculateHelper.js"
}
```

As shown below

```
1  {
2      "hooks": [
3          {
4              "name": "dw.ocapi.shop.basket.calculate",
5              "script": "./cart/calculate.js"
6          },
7          {
8              "name": "dw.extensions.applepay.paymentAuthorized.authorizeOrderPayment",
9              "script": "./checkout/applepay.js"
10         },
11         {
12             "name": "dw.order.calculateDiscounts",
13             "script": "../../../../../int_talonone/cartridge/scripts/helper/calculateHelper.js"
14         }
15     ]
16 }
17 }
```

3.4 External Interfaces

All requests are done through Talon.One API's with the API key Provided by Talon.One.

The full reference guide, along with the resource structure for requests & responses can be found in the portal

<https://docs.talon.one/docs/dev/integration-api/overview/>

<https://docs.talon.one/docs/dev/integration-api/api-effects/>

3.4.1 Dry Request

A dry request is a test request that is checked by Talon.One but that is not executed.

To mark an Integration API request as dry, add the optional query parameter dry=true. No data about this request and its response is stored in Talon.One's database.

Also, the full reference guide for dry request can be found in this portal:-

<https://docs.talon.one/docs/dev/integration-api/dry-requests/>

Note:- No changes related to this are handled as part of the cartridge.

3.4.2 Return Cart Items

For this, a new return request for the specified cart item is required. The return request can be invoked to automatically switch the session state from closed to partially returned or cancelled.

Also, the full reference guide for return item can be found in this portal:-

<https://docs.talon.one/integration-api/#tag/Customer-sessions/operation/returnCartItems>

Note:- No changes related to this are handled as part of the cartridge.

3.4.3 Service Down

In any case, if the services are unavailable, we will not handle any discounts. If a discount has already been applied, it will not be removed, and if no discount has been applied, it will remain the same. The Talon.One service and custom logs would contain information about the service being unavailable.

4. User Guide

4.1 Logs

Configuration to view Talon.One logs:- On Administration → Operations → Custom Log Settings.

Ensure that the Log Category to be leveled as DEBUG or INFO & Log files are checked with FATAL, ERROR, WARN, INFO as shown below:-

The screenshot shows the 'Custom Log Settings' page under 'Administration / Operations'. It includes sections for 'Custom Log Filters' and 'Custom Log Targets'. In the filters section, there's a note about custom log category names not including 'custom' as a prefix. Below this are tables for 'Active' and 'Log Category' with rows for 'root' at level 'INFO'. The 'Log Files' section has checkboxes for FATAL, ERROR, WARN, and INFO, all of which are checked. The 'Log Center' and 'Request Log' sections are also visible.

To view the request & response in service log, ensure that the communication Log Enabled is checked on service configurations.

The screenshot shows the 'talonone.http.service' configuration page under 'Administration > Operations > Services'. A green message at the top says 'Service changes have been applied.' The 'Communication Log Enabled' checkbox is checked. Other settings include Name: 'talonone.http.service', Type: 'HTTP', Enabled: checked, Service Mode: 'Live', Log Name Prefix: 'talonone', and Force PRD Behavior in Non-PRD Environments checked. At the bottom are 'Apply' and 'Reset' buttons.

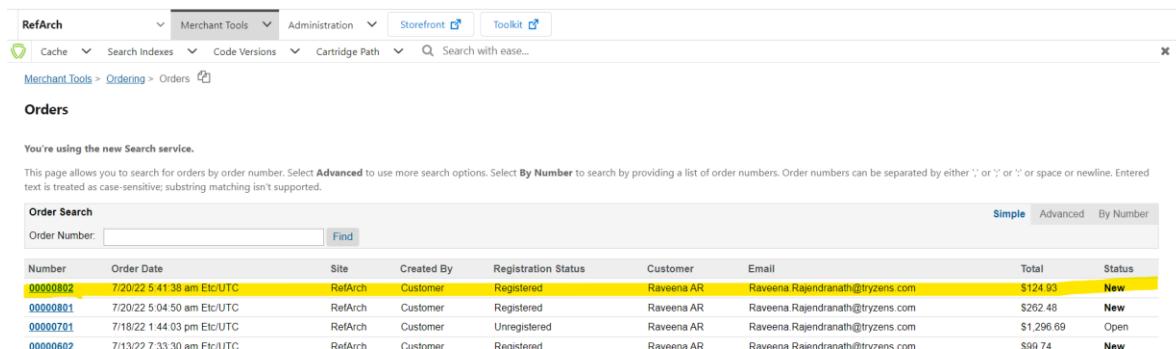
The integration includes the following logs:-

- Service communication logs – starts with “service-talonone-***”. These logs contain every request and response to the Talon.One.
- Custom error info are logged under “custom-talonone-***”, files depending on the case.

4.2 Storefront Functionality

The newly created order can be viewed in Business Manager:

Merchant Tools → Ordering → Orders

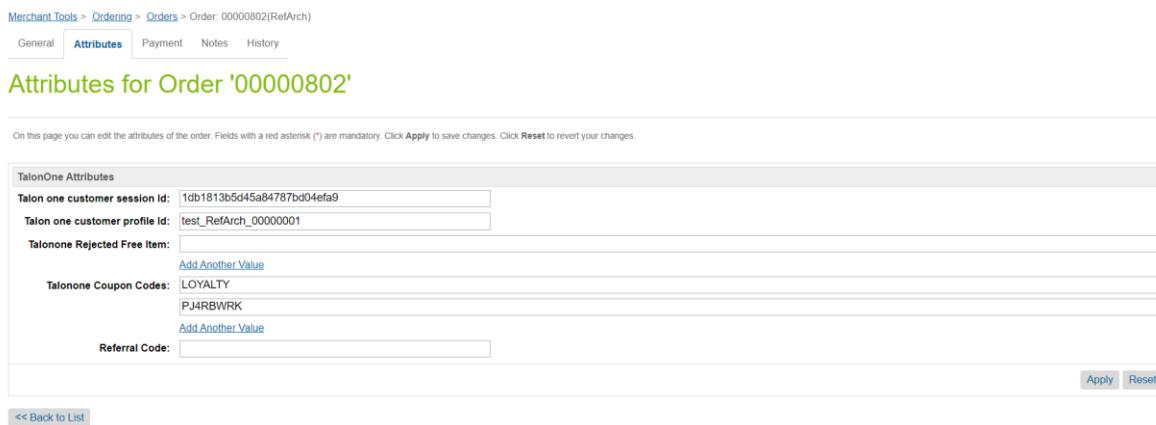


The screenshot shows the Business Manager interface with the 'Orders' section selected under 'Merchant Tools > Ordering > Orders'. The page displays a table of orders with columns: Number, Order Date, Site, Created By, Registration Status, Customer, Email, Total, and Status. One order is highlighted in yellow: 00000802, placed on 7/20/22 at 5:41:38 am Etc/UTC by RefArch, registered customer Raveena AR, total \$124.93, status New.

Number	Order Date	Site	Created By	Registration Status	Customer	Email	Total	Status
00000802	7/20/22 5:41:38 am Etc/UTC	RefArch	Customer	Registered	Raveena AR	Raveena.Rajendranath@tryzens.com	\$124.93	New
00000801	7/20/22 5:04:50 am Etc/UTC	RefArch	Customer	Registered	Raveena AR	Raveena.Rajendranath@tryzens.com	\$262.48	New
00000701	7/18/22 1:44:03 pm Etc/UTC	RefArch	Customer	Unregistered	Raveena AR	Raveena.Rajendranath@tryzens.com	\$1,296.69	Open
00000602	7/13/22 7:33:30 am Etc/UTC	RefArch	Customer	Registered	Raveena AR	Raveena.Rajendranath@tryzens.com	\$99.74	New

Talon.One Session Id, Profile Id, Rejected Free Items, Applied Coupon Codes, & Applied Referral

Code can be inspected in the Attributes tab of the order:



The screenshot shows the 'Attributes' tab for Order 00000802. It displays TalonOne specific attributes: customer session ID (1db1813b5d45a84787bd04efa9), customer profile ID (test_RefArch_00000001), rejected free items (empty), coupon codes (LOYALTY, PJ4RBWRK), and a referral code field (empty). Buttons for 'Apply' and 'Reset' are at the bottom right.

Attributes for Order '00000802'

TalonOne Attributes

Talon one customer session Id: 1db1813b5d45a84787bd04efa9

Talon one customer profile Id: test_RefArch_00000001

Talonone Rejected Free Item: [Add Another Value](#)

Talonone Coupon Codes: LOYALTY
PJ4RBWRK
[Add Another Value](#)

Referral Code:

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

5. Known Issues

The LINK Cartridge has no known issues.

6. Release History

Version	Date	Changes
21.10.0	2021-10-25	Initial release