



Extend Integration for Salesforce Commerce Cloud SFRA

Welcome to Extend! We provide an easy way for any merchant to sell extended warranties - generating new revenue, increasing purchase conversion, and dramatically improving the customer experience.

This guide will walk you through the process of getting live with Extend protection plans, so you can start selling and earning revenue as quickly and easily as possible. The process is pretty simple - and even more simple with a connected SFCC reference application SFRA v5.3.0 or later. Read on to learn more!

Official Extend Cartridge Link: <https://github.com/helloextend/salesforce-commerce-cloud>



Table of Contents

Extend Integration for Salesforce Commerce Cloud SFRA	1
Table of Contents	2
Component Overview	4
Cartridge Features	4
Use Cases	4
Compatibility	4
Privacy Policy on Payments	4
Implementation Guide	6
Setup of Business Manager	6
Upload cartridge & include in the cartridge path	6
Import metadata	6
Import custom jobs	6
Import service	6
Import Extend catalog	7
Configuration	7
Catalog setup	7
Custom preferences	7
Jobs	9
Extend Export Products	9
Extend Contracts Creation	10
Custom Code	11
Updates to calculate.js	11
Orders API	12
Custom preferences:	12
Testing	12
Extend SDK Analytics	17
Custom preferences	17
Tracking events	17
Testing	22
Enabling Extend	22
PDP Offers	22
Extend Plan	22
Cart	24





Extend Plans in Cart	24
Business Manager	25
ExtendContractsQueue custom object	26
Questions? Feedback?	28





Component Overview

1. Cartridge Features

This integration cartridge enables the following features:

- Automated product catalog export to Extend
- Present offers via the Product Display Page (PDP), an up-sell modal during the add-to-cart action, and within the Cart page for eligible products
- Purchase Extend Protection Plans for eligible products
- Contract creation and cancellation with Extend via the API

2. Use Cases

For a merchant, the primary use cases include:

- Maintaining core Extend functionalities directly within Business Manager
- Display Extend Protection Plan products in PDP & Cart
- Display an Extend Protection Plan up-sell modal in PDP & Cart
- Capture customer behavior with Extend's analytics module
- Generate contracts for newly purchased protection plans with Extend
- Cancel contracts for returned orders with Extend

For a customer, the primary use cases include:

- Attach an Extend Protection Plan to an eligible catalog product during their purchase journey
- View details, including coverage and terms & conditions about the Extend Protection Plans
- Remove Extend Protection Plans from Cart (either by removing the main product or just the warranty product) during their purchase journey

Compatibility

This cartridge has been developed and tested against Commerce Cloud Digital 19.10 and integrated on top of SFRA v.5.3.0 or later.

Privacy Policy on Payments

This cartridge implements a custom object for temporary storage of order data needed for Extend Protection Plan contract generation. This custom object (ExtendContractsQueue) was implemented based on a queue pattern that is consumed by a job configured to execute every 5 minutes. This data is not exposed publicly and used only in a server-to-server communication setup.





Extend does not handle payments from the end-user. The merchant is responsible for charging the end-user for the Extend Protection Plans purchased, using the existing tenders that are currently supported on the specific environment. The merchant is responsible for generating Extend Protection Plan contracts only after charging the end-user for the warranty plans purchased. Please visit <https://extend.com/privacy> for more information.





Implementation Guide

The SFRA version of this integration follows the extensibility and modularity design principles behind SFRA. It's been designed to live on top of the storefront-base-architecture components (app_storefront_base & modules). This means it's required that this cartridge is included in an environment that follows the SFRA recommendations.

1. Setup of Business Manager

1.1. Upload cartridge & include in the cartridge path

- 1.1.1. Upload the int_extend_sfra cartridge to the sandbox.
- 1.1.2. Go to Business Manager → Administration → Site → Manage Sites. Select the desired site and go to Settings. At the beginning of the Cartridge Path add "int_extend_sfra:" in order to include the cartridge in the current site's configuration.

1.2. Import metadata

- 1.2.1. Go to Business Manager → Administration → Site Development → Import & Export
- 1.2.2. Upload the following files from ./metadata/meta/ folder
 - system-objecttype-extensions.xml
 - custom-objecttype-definitions.xml
- 1.2.3. Go to Business Manager → Administration → Site Development → Import & Export → Metadata → Import. Both previous files should be available for import. Proceed to importing both.

1.3. Import custom jobs

- 1.3.1. Go to Business Manager → Administration → Operations → Import & Export
- 1.3.2. Upload the following file: ./metadata/jobs.xml
- 1.3.3. Go to Business Manager → Administration → Operations → Import & Export → Jobs → Import. The previous file should be available for import. Proceed to importing it.

1.4. Import service

- 1.4.1. Go to Business Manager → Administration → Operations → Import & Export
- 1.4.2. Upload the following file: ./metadata/services.xml





- 1.4.3. Go to Business Manager → Administration → Operations → Import & Export → Services → Import. The previous file should be available for import. Proceed to importing it.
- 1.4.4. Note: by default, the service URL is configured to point to the demo instance of the Extend API. Prior to going live with the integration, this will need to be changed and tested with the live production API.

1.5. Import Extend catalog

- 1.5.1. Select the site integrating with Extend
- 1.5.2. Go to Business Manager → Merchant Tools → Products and Catalogs → Import & Export
- 1.5.3. Upload the following file: ./metadata/catalog.xml. This will import 3 dummy option products, which are used in order to create dynamic product line items in the cart. We recommend keeping this as a separate master catalog, for visibility purposes.

2. Configuration

2.1. Catalog setup

- 2.1.1. The 3 Extend products that require configuration are:
 - EXTEND-12
 - EXTEND-24
 - EXTEND-36
- 2.1.2. In order to make the Extend products merchandisable, perform the following:
 - Assign them to your storefront catalog
 - Allocate an inventory record in the inventory file assigned to the current site
 - Configure this entry as perpetual, since these are **intangible goods** and **need to always be in-stock**
- 2.1.3. These Extend products do not require any description, pricing, or any other regular product attributes to be configured

2.2. Custom preferences

- 2.2.1. Go to Business Manager → Merchant Tools → Site Preferences → Custom Preferences → **Extend: Integration Preferences**
 - **Enable Extend:** Global switch for all things Extend related within the environment.
 - **Extend Store ID:** Merchant specific Store ID available from their Extend account. The Store ID is environment specific (Extend





production or Extend sandbox), and is required to make the API connections with Extend.

- **Extend Environment in Production:** Defines if Extend APIs call the Extend Production environment (Yes = <https://api.helloextend.com/>) or Extend Demo environment (No/None = <https://api-demo.helloextend.com/>). Access Token must be updated to correspond to the appropriate environment. Note: this is available from the Extend Merchant Portal and is environment specific.
- **Extend Access Token:** Merchant specific API key available from their Extend account. This key is environment specific (Extend production or Extend sandbox), and is required to make the API connections with Extend.
- **Extend Store Name:** This is the name of the user's store. Used for calls to Extend API. Note: Extend Store Name is by default configured to "Online Store".
- **Extend Integration Method:** Determines to which Extend API order data is sent.

Orders API on Order Create triggers the integration with Extend near real-time upon the order creation in SFCC Business Manager.

Orders API on Schedule creates a custom queue record (*ExtendOrderQueue*) upon order creation in the SFCC Business Manager, but will not integrate with Extend until the Job is run manually or on a schedule. The Job will consume the custom queue.

Contracts API on Schedule is a legacy feature whereby only Contract creation occurs (requires Extend API Version 2021-04-01 or earlier). Integration with Contracts API uses a Job that consumes a custom queue (*ExtendContractQueue*) when run manually or on a schedule. Please confirm with the Extend delivery team before selecting this option.

- **Product Image Size:** Determines the size of the product image stored on the Extend catalog. Accepted values are "small", "medium", "large", and "hi-res". This is the image that Extend uses when sending warranty contract emails to customers. Image Size is by default configured to large.
- **Extend Refund Key:** Unique key defined by the merchant to be used as part of Extend's contract cancellation process in SFCC if applicable.





2.2.2. Go to Business Manager → Merchant Tools → Site Preferences → Custom Preferences → **Extend: Storefront Preferences**

- **Extend SDK URL:** Denotes the SDK version used on the Storefront, specifically for the Extend PDP offer, Interstitial Modal offer, and Cart offer (individual settings below).
- **Enable in PDP Offers:** Allows merchants to enable or disable the Extend Product Detail Page (PDP) offer.
- **Enable Modal Offers:** Allows merchants to enable or disable the Extend Interstitial Modal offer.
- **Enable Cart Offers:** Allows merchants to enable or disable the Extend Cart offer.
- **Enable Analytics:** Invokes Extend's Analytics methods within the Extend SDK.

3. Jobs

3.1. Extend Export Products

This job uses Extend's API to traverse the storefront catalog and export each Simple and Variant Product that is currently merchandisable (online and in stock):

- 3.1.1. Go to Business Manager → Operations → Jobs. Select the Extend Products Export job.
- 3.1.2. Go to Job Steps and select Scope. Configure this to run in the scope of the site integrating with Extend.
- 3.1.3. Schedule this to run at the preferred cadence (Extend recommends once per day), at a time that's most suitable for a recurring job setup, ideally after the catalog is being updated.
- 3.1.4. Ensure this job is triggered at least once without errors and notify Extend after the successful run. Extend will confirm if the product sync was successful.

3.2. Extend Orders Creation

This job uses Extend's Orders API to consume the orders queue, stored in the ***ExtendOrdersQueue*** custom object. During SFCC order creation (after checkout), an instance of this object is created for each SFCC order. When this job is triggered, it sends an Order generation request to Extend and Extend will generate any necessary service contracts or lead tokens (for Post Purchase flow) and write them back onto the corresponding order line items. Upon successful order creation, the job removes the completed instances from the queue. This job is only required if the **Extend Integration Method** custom preference is set to **Orders API on Schedule**.





3.3. Extend Contracts Creation

This job uses Extend's Contracts API to consume the contracts queue, stored in the **ExtendContractsQueue** custom object. During SFCC order creation (after checkout), an instance of this object is created for each Extend Protection Plan on the SFCC order. When this job is triggered, it sends a contract generation request to Extend for each queued element. Upon successful contract creation, the job removes the completed instances from the queue. This job is only required if the **Extend Integration Method** custom preference is set to **Contracts API on Schedule**.

3.4. Send Historical Orders

This job uses Extend's Orders API to send historical orders from the past 2 years in SFCC to Extend during extension/cartridge installation for analytics and optimization reasons.

Note: For new merchants only who install the SFCC extension for the first time.

3.5. Create Refunds from SFCC

This job uses Extend's API to cancel an active contract on a canceled SFCC order (Order Status = "Canceled"). Please note that Extend will never refund the customer directly, as Extend does not capture the customer's payment information at any point. Customer refunds should be processed based on the merchant's regular refund process. This job will simply terminate active contract(s) after the customer's refund request is accepted by the merchant. All canceled Extend contracts are represented as credit lines on Extend's monthly invoice. For additional information on Extend's refund policy, please reach out to the Extend project team.

Note: This job is not required if the merchant's return and refund process occurs outside of SFCC as the contract cancellation requests to Extend will likely occur in a downstream system. Confirm with the Extend project team prior to scheduling this job.





4. Custom Code

4.1. Updates to calculate.js

Custom changes are required to the calculate.js hook. First, open the **calculate.js** hook script (`*/cartridge/scripts/cart/hooks/calculate.js`) and find function `calculateProductPrices(basket) {...}`.

Change the “handle non-catalog products” if statement to be:

```
if (!productLineItem.catalogProduct || productLineItem.custom.parentLineItemUUID)
```

```
96  /**
97   * @function calculateProductPrices
98   *
99   * Calculates product prices based on line item quantities. Set calculates prices
100  * on the product line items. This updates the basket and returns nothing
101  *
102  * @param {object} basket The basket containing the elements to be computed
103  */
104  function calculateProductPrices(basket) {
105      // get total quantities for all products contained in the basket
106      var productQuantities = basket.getProductQuantities();
107      var productQuantitiesIt = productQuantities.keySet().iterator();
108
109      // get product prices for the accumulated product quantities
110      var productPrices = new HashMap();
111
112      while (productQuantitiesIt.hasNext()) {
113          var prod = productQuantitiesIt.next();
114          var quantity = productQuantities.get(prod);
115          productPrices.put(prod, prod.priceModel.getPrice(quantity));
116      }
117
118      // iterate all product line items of the basket and set prices
119      var productLineItems = basket.getAllProductLineItems().iterator();
120      while (productLineItems.hasNext()) {
121          var productLineItem = productLineItems.next();
122
123          // handle non-catalog products
124          if (!productLineItem.catalogProduct || productLineItem.custom.parentLineItemUUID) {
125              productLineItem.setPriceValue(productLineItem.basePrice.valueOrNull);
126              continue;
127          }
128
129          var product = productLineItem.product;
130
131          // handle option line items
132          if (productLineItem.optionProductLineItem) {
133              // for bonus option line items, we do not update the price
134              // the price is set to 0.0 by the promotion engine
135              if (!productLineItem.bonusProductLineItem) {
```





5. Orders API

5.1. Custom preferences:

- 5.1.1. Go to Business Manager → Merchant Tools → Site Preferences → Custom Preferences → Extend: Integration Preferences
- 5.1.2. Configure the provided Extend credentials in the following fields (refer to Section 2.2 for additional info):
 - Extend Access Token
 - Extend Store ID
- 5.1.3. Configure the following Orders API specific preferences:
 - Extend Integration Method: set this to Orders API on Order Create (please confirm with the Extend project delivery team).

5.2. Testing

5.2.1. LeadToken:

Go to PDP → Choose product variation → Click “Add to Cart” button → Go to cart → Go to checkout page → Make a purchase.

Result:

Go to Merchant Tools → Ordering → Orders → Choose Your Order → Shipment → Choose ProductLineItem → Attributes





[Merchant Tools](#) > [Ordering](#) > [Orders](#) > [Order: 00005906\(RefArch\)](#) > [Shipment: 00032006](#) > Line Item: 640188017003M


General

Attributes

Attributes for "Charcoal Flat Front Athletic Fit

On this page you can edit the attributes of the product line item. Fields with a red asterisk (*) are mandatory. Click **Apply** to save changes.

Extend

 **Extend Contract ID:** No data is available

Extend Refund Statuses:

Lead Token:

5.2.2. **Extend Contract ID:**

Go to PDP → Choose product variation → Choose an extension plan → Click “Add to Cart” button → Go to cart → Go to checkout page → Make a purchase.

Result:

Go to Merchant Tools → Ordering → Orders → Choose Your Order → Shipment → Choose ExtendLineItem → Attributes





[Merchant Tools](#) > [Ordering](#) > [Orders](#) > [Order: 00006004\(RefArch\)](#) > [Shipment: 00032504](#) > Line Item: EXTEND-12


General

Attributes

Attributes for "Extend Product Protection: 1 Striped Wool Suit"

On this page you can edit the attributes of the product line item. Fields with a red asterisk (*) are mandatory. Click **Apply** to save.

Extend

 **Extend Contract ID:** 772b3e12-1436-41be-b783-a20b96eb1161
6ac9cf81-f623-4999-8c40-ae4c8143a111

Extend Refund Statuses:

Lead Token:

5.2.3. Refund status:

- Go to PDP → Choose product variation → Choose an extension plan → Click “Add to Cart” button → Go to cart → Go to checkout page → Make a purchase.
- Then Go to Merchant Tools → Ordering → Orders → Choose Your Order → Order Status → Choose **“Cancelled”** → Click **“Apply”** button.
- Then Go To Administration → Operations → Jobs → **“Create Refund from SFCC”** → Click “Run Now” button.

Result:

Go to Merchant Tools → Ordering → Orders → Choose Your Order → Shipment → Choose ExtendLineItem → Attributes:






[Merchant Tools](#) > [Ordering](#) > [Orders](#) > [Order: 00005106\(RefArch\)](#) > [Shipment: 00028006](#) > Line Item: EXTEND-12

General

Attributes

Attributes for "Extend Product Protection: 1

On this page you can edit the attributes of the product line item. Fields with a red asterisk (*) are mandatory. Click **Apply** to save.

Extend	
 Extend Contract ID:	276694fa-ee51-40b9-9096-45bc8c0593fd
Extend Refund Statuses:	<div><pre>{"276694fa-ee51-40b9-9096-45bc8c0593fd": "refund_paid"}</pre></div>
Lead Token:	<input type="text"/>

Go to Merchant Tools → Ordering → Orders → Choose Your Order → Attributes:





[Merchant Tools](#) > [Ordering](#) > [Orders](#) > Order: 00005106(RefArch)

General

Attributes

Payment

Notes

History

Attributes for Order '00005106'

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

Extend

Extend Refund Status: refund_paid (Refund Paid)



Refund Statuses:

refund_quoted: Refund amounts have been previewed by merchant/customer at least once, but merchant has not reported that they have refunded the customer yet.

refund_paid: Refund has been reported as paid by the merchant. Contract is canceled. Finance will credit the merchant in the next monthly invoice.

refund_denied: Customer is not eligible for a refund. Customer has consumed all entitlements associated with the contract. Get Refund via Contract ID.

ERROR: service call error.





6. Extend SDK Analytics

6.1. Custom preferences

- 6.1.1. Go to Business Manager → Merchant Tools → Site Preferences → Custom Preferences → Extend: Storefront Preferences
- 6.1.2. Configure the provided Extend credentials in the following fields (refer to Section 2.2 for additional info)
 - Extend Access Token
 - Extend Store ID
- 6.1.3. Ensure **“Extend Analytics”** is set to **“Yes”**.

Extend Analytics Switch	Yes
(extendAnalyticsSwitch)	

6.2. Tracking events

6.2.1. **Extend.trackOfferViewed:**

This method is triggered when an Extend warranty offer is rendered on the screen to a user.

Name	Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> Product-Productnav?lang=en_US&start=1&pid=25686514&format=ajax		Request Payload view source				
<input type="checkbox"/> offers?storeId=5bcd9cf9-807a-4389-ba29-6d38daf7d395&productId=25686...		▼ {type: "analytics",...}				
<input type="checkbox"/> Product-Variation?pid=25686514&dwvar_25686514_size...WL&Quantity=1&...		▼ object: {userId: "185D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "offer_viewed",...}				
<input type="checkbox"/> offers?storeId=5bcd9cf9-807a-4389-ba29-6d38daf7d395&productId=25686...		cardId: "08F2F4B4-7570-4A4E-99E5-416E0840CA88"				
<input type="checkbox"/> Product-Variation?pid=25686514&dwvar_25686514_wid...WL&Quantity=1...		▶ context: {,...}				
<input type="checkbox"/> tracking		event: "offer_viewed"				
		platform: "extend-sdk-client"				
		▼ properties: {productId: "750518548203", offerType: {area: "product_page", component: "buttons"}}}				
		▶ offerType: {area: "product_page", component: "buttons"}				
		productId: "750518548203"				
		▶ sessionState: {products: [], productsWithWarranties: [],...}				
		storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395"				
		timestamp: 1643298228798				
		userId: "185D4861-6DE1-4552-96E6-65624D3E8EB7"				
		type: "analytics"				

6.2.2. **Extend.trackProductAddedToCart:**

This method is triggered when an user adds a (non-Extend) product to the cart. This event accepts the product's “productId” and the number of units (quantity) added to the cart.

Name	Headers	Payload	Preview	Response	Initiator	Timing
https://api-demo.helloextend.com/tracking		Request Payload view source				
<input type="checkbox"/> Product-Productnav?lang=en_US&start=1&pid=25686514&format=ajax		▼ {type: "analytics",...}				
<input type="checkbox"/> offers?storeId=5bcd9cf9-807a-4389-ba29-6d38daf7d395&productId=25686...		▼ object: {userId: "185D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "product_added_to_cart",...}				
<input type="checkbox"/> Product-Variation?pid=25686514&dwvar_25686514_size...WL&Quantity=1&...		cardId: "08F2F4B4-7570-4A4E-99E5-416E0840CA88"				
<input type="checkbox"/> offers?storeId=5bcd9cf9-807a-4389-ba29-6d38daf7d395&productId=25686...		▶ context: {,...}				
<input type="checkbox"/> Product-Variation?pid=25686514&dwvar_25686514_wid...WL&Quantity=1...		event: "product_added_to_cart"				
<input type="checkbox"/> tracking		platform: "extend-sdk-client"				
<input type="checkbox"/> offers?storeId=5bcd9cf9-807a-4389-ba29-6d38daf7d395&productId=75051...		▼ properties: {productId: "750518548203", productQuantity: 1}				
<input type="checkbox"/> tracking		productId: "750518548203"				
<input type="checkbox"/> tracking		productQuantity: 1				
<input type="checkbox"/> Cart-AddProduct?format=ajax		▶ sessionState: {products: [{productId: "750518548203", productQuantity: 1}], productsWithWarranties: [],...}				
		storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395"				
		timestamp: 1643298587260				
		userId: "185D4861-6DE1-4552-96E6-65624D3E8EB7"				
		type: "analytics"				





6.2.3. Extend.trackOfferAddedToCart

This method is triggered when an user adds an Extend warranty to the cart. This event accepts the product's "productId", the number of units (quantity), and the Extend warranty's "planId".

Name	Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> tracking <input type="checkbox"/> Extend-AddExtendProduct <input type="checkbox"/> tracking <input type="checkbox"/> Extend-IsEligibleForWarranty?uuid=7896859ab60bb9ab25a226641f <input type="checkbox"/> Extend-IsEligibleForWarranty?uuid=635210df93d0915974167902d1		<div>▼ Request Payload view source</div> <div>▼ {type: "analytics",...}</div> <div>▼ object: {userId: "185D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "offer_added_to_cart",...}</div> <div> productId: "08F2F4B4-7570-4A4E-99E5-416E0840CA88"</div> <div> context: {,...}</div> <div> eventName: "offer_added_to_cart"</div> <div> platform: "extend-sdk-client"</div> <div>▼ properties: {productId: "750518548203", productQuantity: 1, warrantyQuantity: 1,...}</div> <div> offerType: {area: "cart_page", component: "modal"}</div> <div> planId: "10001-misc-elec-base-replace-3y"</div> <div> productId: "750518548203"</div> <div> productQuantity: 1</div> <div> warrantyQuantity: 1</div> <div>▶ sessionState: {products: [],...}</div> <div> storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395"</div> <div> timestamp: 1643290855612</div> <div> userId: "185D4861-6DE1-4552-96E6-65624D3E8EB7"</div> <div> type: "analytics"</div>				

6.2.4. Extend.trackOfferRemovedFromCart

This method is triggered when an user removes an Extend warranty from the cart. This event accepts the Extend warranty's "planId" and the associated "productId" of the product the warranty would have covered.

Name	Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> https://api-demo.helloextend.com/tracking <input type="checkbox"/> extend-iseligiblerforwarranty.ruuid=/o96859ab60bb9ab25a226641f <input type="checkbox"/> tracking <input type="checkbox"/> offers?storeId=5bcd9cf9-807a-4389-ba29-6d38daf7d395&productId=75051... <input type="checkbox"/> tracking		<div>▼ Request Payload view source</div> <div>▼ {type: "analytics",...}</div> <div>▼ object: {userId: "185D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "offer_removed_from_cart",...}</div> <div> productId: "08F2F4B4-7570-4A4E-99E5-416E0840CA88"</div> <div> context: {,...}</div> <div> eventName: "offer_removed_from_cart"</div> <div> platform: "extend-sdk-client"</div> <div>▼ properties: {productId: "750518548203", planId: "10001-misc-elec-base-replace-3y"}</div> <div> planId: "10001-misc-elec-base-replace-3y"</div> <div> productId: "750518548203"</div> <div>▶ sessionState: {products: [{productId: "750518548203", productQuantity: 1}], productsWithWarranties: [],...}</div> <div> storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395"</div> <div> timestamp: 1643291063318</div> <div> userId: "185D4861-6DE1-4552-96E6-65624D3E8EB7"</div> <div> type: "analytics"</div>				

6.2.5. Extend.trackOfferUpdated

This method is triggered when an user increments or decrements the quantity of a warranty that has already been added to the cart. This event takes the Extend warranty's "planId", the associated "productId" of the product the warranty covers, as well as an update object containing the set of updates to apply to the warranty offer. **If the quantity of warranty is updated to 0, the a Extend.trackOfferRemoved event is called**, and further updates to this planId/productId will result in a no-op until it is re-added via





Extend/trackOfferAddedToCart.

Name	× Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> Extend-IsEligibleForWarranty?uuid=7896859ab60bb9ab25a226641f						
<input type="checkbox"/> Extend-IsEligibleForWarranty?uuid=eeeab5379a83081e350fc5f3aa						
<input checked="" type="checkbox"/> tracking		<div>▼ Request Payload view source</div> <div>▼ {type: "analytics",...}</div> <div>▼ object: {userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "offer_updated",...}</div> <div>cardId: "08F2F4B4-7570-4A4E-99E5-416E0840CA88"</div> <div>▶ context: {...}</div> <div>eventName: "offer_updated"</div> <div>platform: "extend-sdk-client"</div> <div>▼ properties: {productId: "750518548203", planId: "10001-misc-elec-base-replace-1y",...}</div> <div>▶ newQuantities: {warrantyQuantity: 2, productQuantity: 2}</div> <div>▶ oldQuantities: {productQuantity: 1, warrantyQuantity: 1}</div> <div>planId: "10001-misc-elec-base-replace-1y"</div> <div>productId: "750518548203"</div> <div>▶ sessionState: {products: [...]}</div> <div>storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395"</div> <div>timestamp: 1643291432640</div> <div>userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7"</div> <div>type: "analytics"</div>				

6.2.6. Extend.trackProductRemovedFromCart

This method is triggered when an user removes a product from the cart that **does not** have a warranty offer associated with it. This event takes the “productId” of the product being removed from the cart.

Name	× Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> tracking		<div>▼ Request Payload view source</div> <div>▼ {type: "analytics",...}</div> <div>▼ object: {userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "product_removed",...}</div> <div>cardId: "08F2F4B4-7570-4A4E-99E5-416E0840CA88"</div> <div>▶ context: {...}</div> <div>eventName: "product_removed"</div> <div>platform: "extend-sdk-client"</div> <div>▼ properties: {productId: "750518548203"}</div> <div>productId: "750518548203"</div> <div>▶ sessionState: {products: [], productsWithWarranties: [...]}</div> <div>storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395"</div> <div>timestamp: 1643291664323</div> <div>userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7"</div> <div>type: "analytics"</div>				

6.2.7. Extend.trackProductUpdated

This method is triggered when an user increments or decrements the quantity of a product that has already been added to the cart. This event takes the “productId” of the product being updated, as well as an update object containing the set of updates to apply to the product. The product being updated must **not** be associated with a warranty offer.
If the “productQuantity” passed into this method is 0, then





Extend.trackProductRemoveFromCart is fired.

Name	Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> Extend-IsEligibleForWarranty?uuid=f48cc344ec25b9d94f038586e8		Request Payload view source				
<input type="checkbox"/> tracking		▼ {type: "analytics",...}				
<input type="checkbox"/> tracking		▼ object: {userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "product_updated",...}				
		▼ cartId: "08F2F484-7570-4A4E-99E5-416E0840CA88"				
		▶ context: {,...}				
		event: {eventName: "product_updated"				
		platform: "extend-sdk-client"				
		▼ properties: {productId: "750518699608M", oldProductQuantity: 1, newProductQuantity: 2}				
		newProductQuantity: 2				
		oldProductQuantity: 1				
		productId: "750518699608M"				
		▶ sessionState: {products: [{productId: "750518699608M", productQuantity: 2}], productsWithWarranties: [],...}				
		storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395"				
		timestamp: 1643292010508				
		userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7"				
		type: "analytics"				

6.2.8. Extend.trackCartCheckout

This method is triggered when a customer completes a purchase.

- After the Checkout flow is complete (at Order Confirmation page), there fires two tracking events - offer_sold, and total_cart_revenue

Name	Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> COShipping-GetApplicableShippingMethodsJSON?address...ess2=&country...		Request Payload view source				
<input type="checkbox"/> COShipping-GetApplicableShippingMethodsJSON?address...ountryCode=u...		▼ {type: "analytics",...}				
<input type="checkbox"/> COShipping-GetApplicableShippingMethodsJSON?address...s&stateCode=...		▼ object: {userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "offer_sold",...}				
<input type="checkbox"/> COBilling-SelectCreditCard?creditCardUUID=3a1e6870902235f9c0771e4bf1		▼ cartId: "08F2F484-7570-4A4E-99E5-416E0840CA88"				
<input type="checkbox"/> tracking		▶ context: {,...}				
<input type="checkbox"/> tracking		event: {eventName: "offer_sold"				
		platform: "extend-sdk-client"				
		▼ properties: {offerType: {area: "product_page", component: "buttons"}, planId: "10001-misc-elec-base-replace-3y",...}				
		offerType: {area: "product_page", component: "buttons"}				
		planId: "10001-misc-elec-base-replace-3y"				
		productId: "750518894485"				
		productQuantity: 1				
		warrantyQuantity: 1				
		▶ sessionState: {products: [], productsWithWarranties: [],...}				
		storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395"				
		timestamp: 1643292435532				
		userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7"				
		type: "analytics"				

Name	Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> COShipping-GetApplicableShippingMethodsJSON?address...ess2=&country...		Request Payload view source				
<input type="checkbox"/> COShipping-GetApplicableShippingMethodsJSON?address...ountryCode=u...		▼ {type: "analytics",...}				
<input type="checkbox"/> COShipping-GetApplicableShippingMethodsJSON?address...s&stateCode=...		▼ object: {userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "total_cart_revenue",...}				
<input type="checkbox"/> COBilling-SelectCreditCard?creditCardUUID=3a1e6870902235f9c0771e4bf1		▼ cartId: "08F2F484-7570-4A4E-99E5-416E0840CA88"				
<input type="checkbox"/> tracking		▶ context: {,...}				
<input type="checkbox"/> tracking		event: {eventName: "total_cart_revenue"				
		platform: "extend-sdk-client"				
		▼ properties: {cartTotal: 409.47}				
		cartTotal: 409.47				
		▶ sessionState: {products: [], productsWithWarranties: [],...}				
		storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395"				
		timestamp: 1643292435534				
		userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7"				
		type: "analytics"				

- After the Checkout flow is complete (at Order Confirmation page), there fires two tracking events called - product_sold and total_cart_revenue





Name	×	Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> tracking							
<input type="checkbox"/> tracking							
			<div>▼ Request Payloadview source</div> <div>▼ {type: "analytics",...}</div> <div>▼ object: {userId: "185D4861-60E1-4552-96E6-6562403E8EB7", eventName: "product_sold",...}</div> <div> cartId: "669FA4D1-70F4-4463-B816-28855403B9C5"</div> <div> ▶ context: {,...}</div> <div> eventName: "product_sold"</div> <div> platform: "extend-sdk-client"</div> <div> ▼ properties: {productId: "750518894485", productQuantity: 1}</div> <div> productId: "750518894485"</div> <div> productQuantity: 1</div> <div> ▶ sessionState: {products: [{productId: "750518894485", productQuantity: 1}], productsWithWarranties: [],...}</div> <div> storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395"</div> <div> timestamp: 1643292733992</div> <div> userId: "185D4861-60E1-4552-96E6-6562403E8EB7"</div> <div> type: "analytics"</div>				

Name	×	Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> tracking							
<input type="checkbox"/> tracking							
			<div>▼ Request Payloadview source</div> <div>▼ {type: "analytics",...}</div> <div>▼ object: {userId: "185D4861-60E1-4552-96E6-6562403E8EB7", eventName: "total_cart_revenue",...}</div> <div> cartId: "669FA4D1-70F4-4463-B816-28855403B9C5"</div> <div> ▶ context: {,...}</div> <div> eventName: "total_cart_revenue"</div> <div> platform: "extend-sdk-client"</div> <div> ▼ properties: {cartTotal: 325.48}</div> <div> cartTotal: 325.48</div> <div> ▶ sessionState: {products: [], productsWithWarranties: [],...}</div> <div> storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395"</div> <div> timestamp: 1643292733993</div> <div> userId: "185D4861-60E1-4552-96E6-6562403E8EB7"</div> <div> type: "analytics"</div>				





Testing

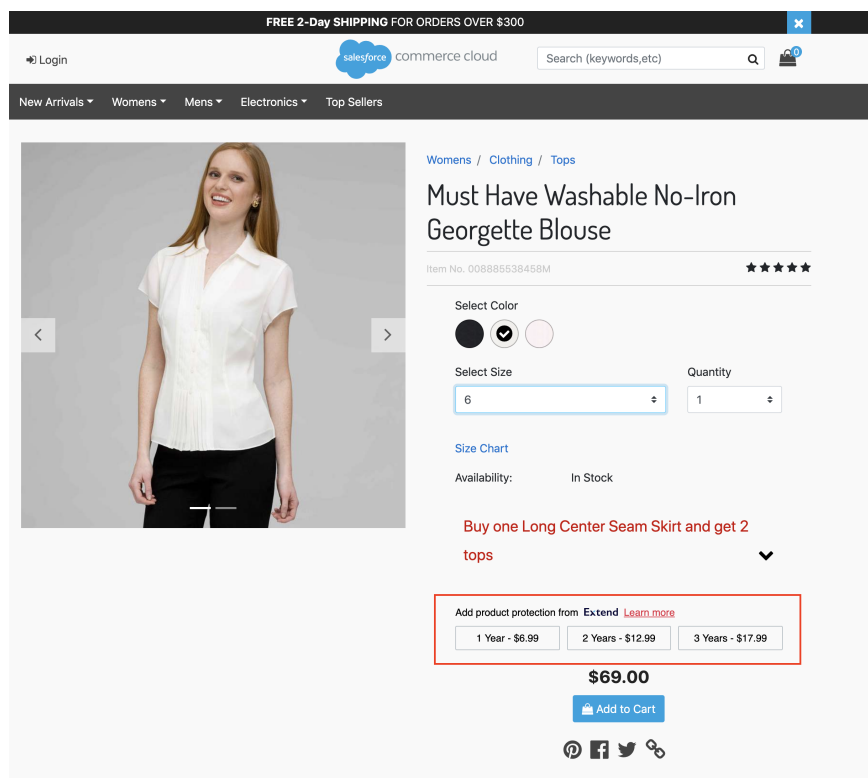
1. Enabling Extend

- Go to Business Manager → Merchant Tools → Site Preferences → Custom Preferences → Extend: Storefront Preferences
- Set the following configuration options to 'Yes'
 - Enable Extend
 - Enable PDP Offers
 - Enable Modal Offers
 - Enable Cart Offers
- Ensure the product catalog has been exported to Extend and your Extend representative enabled warranty plans in the system

2. PDP Offers

2.1. Extend Plan

Navigate to any simple or variant product that has an Extend Protection Plan enabled. Select a variation and confirm an Extend offer is shown.



2.2. Adding a Warranty to the Cart





Select an Extend Protection Plan and click Add to Cart. Confirm 2 products were added to the cart – the SFCC catalog product along with the Extend Protection Plan.

2.3. Interstitial Modal

Navigate to any simple or variant product that has an Extend Protection Plan enabled and click Add to Cart on the PDP without selecting any Extend Protection Plan. Confirm the Extend Interstitial Modal is presented. Selecting “Add Protection” will add the Extend Protection Plan to the cart, along with the catalog product. Selecting “No, thanks” will only add the catalog product to the cart.

The screenshot shows a product page for a 'Must Have Washable No-Iron Georgette Blouse' (Item No. 008885538458) priced at \$69.00. The product is available in 'IVORY' color. An interstitial modal is displayed over the product details, featuring the Extend logo and a photo of a smiling woman. The modal text reads: 'Protect your purchase from drops, breaks, spills, and defects with Extend.' It lists the plan's coverage: fast and free product replacements, accidental damage, and extended malfunction/wear-and-tear protection. Below this, there are three plan options: 1 Year (\$6.99), 2 Year (\$11.99, marked 'Best Value'), and 3 Year (\$17.99, marked 'Best Value'). At the bottom of the modal are two buttons: 'No, thanks' and 'Add protection'. The background of the modal is a blue gradient with a photo of a smiling woman holding a cardboard box.

Must Have Washable No-Iron Georgette Blouse
Item No. 008885538458
\$69.00
SELECT COLOR
IVORY

Protect your purchase from drops, breaks, spills, and defects with Extend.

This plan covers:

- Fast and free product replacements
- Accidental damage such as breaks, drops, and spills
- Extended malfunction and wear-and-tear protection

See plan details [↗](#) Offered and sold by Extend

Plan Duration	Price	Value Label
1 Year Protection Plan	\$6.99	
2 Year Protection Plan	\$11.99	Best Value
3 Year Protection Plan	\$17.99	Best Value

No, thanks Add protection






3. Cart

3.1. Extend Plans in Cart

Confirm an Extend plan has been added to the cart.


FREE 2-Day SHIPPING FOR ORDERS OVER \$300


[Login](#) 

[New Arrivals](#) [Womens](#) [Mens](#) [Electronics](#) [Top Sellers](#)

Your Cart

[Continue Shopping](#) 2 Items [Need Help? Call 1-800-555-0199](#)

Must Have Washable No-Iron Georgette Blouse
 Color: Ivory
Size: 6
In Stock
Each **\$69.00**
Quantity Total **\$69.00**

Extend Product Protection: 2 years for Must Have Washable No-Iron Georgette B...
 Extend Term: 2Y
In Stock
Each **\$12.99**
Quantity Total **\$12.99**

Enter Promo Code
 [Submit](#)

Shipping


Shipping cost **\$5.99**
Sales Tax **\$4.40**
Estimated Total \$92.38
[Checkout](#)

3.2. Up-sell button for eligible products in cart

Add an eligible catalog product to the cart, without selecting any Extend Protection Plan for it. Navigate to the cart page and confirm the cart offer button is presented.

Your Cart

[Continue Shopping](#) 1 Items [Need Help? Call 1-800-555-0199](#)

Must Have Washable No-Iron Georgette Blouse
 Color: Ivory
Size: 6
In Stock
Each **\$69.00**
Quantity Total **\$69.00**
[Add product protection from \\$6.99](#)

Enter Promo Code
 [Submit](#)

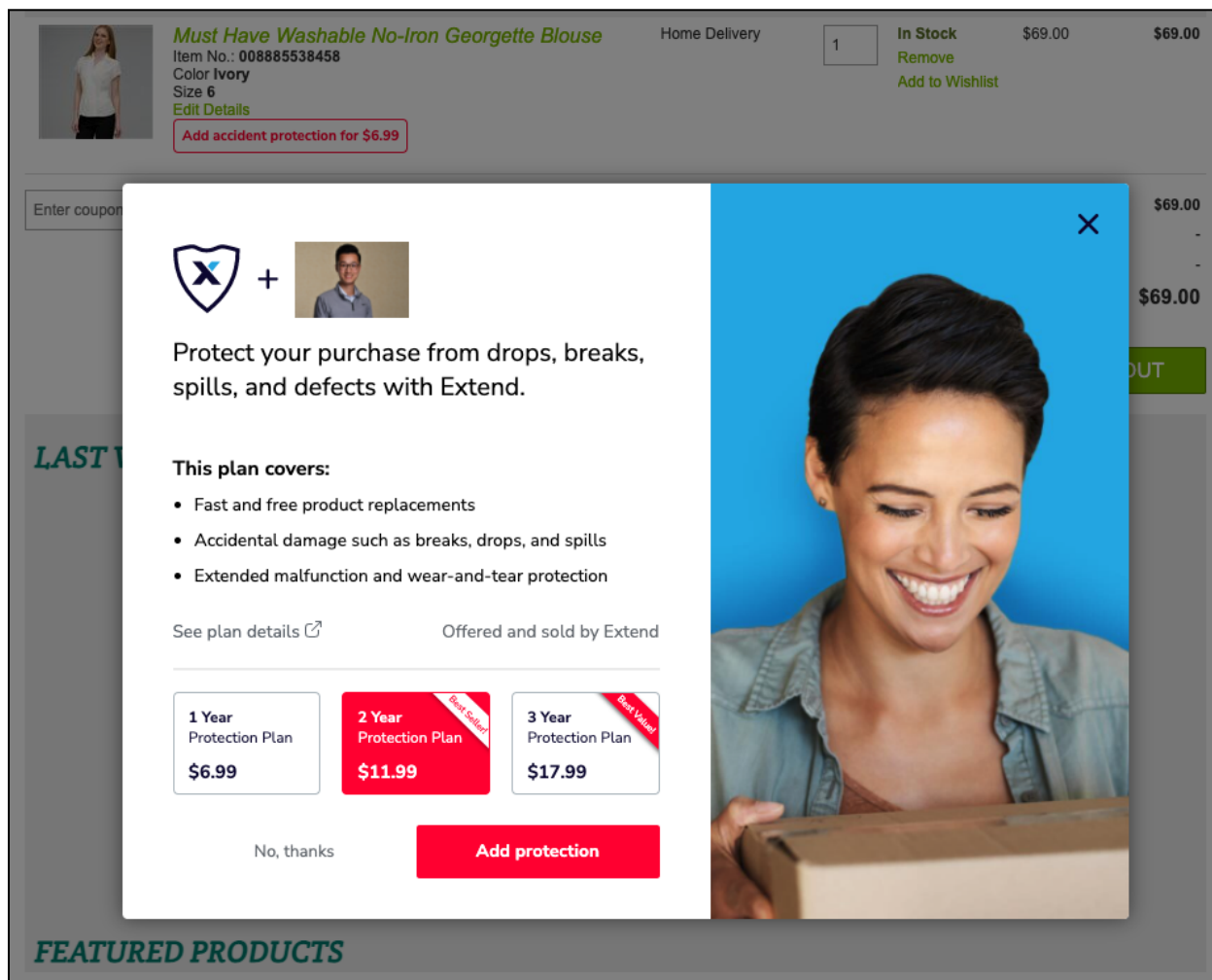
Shipping





3.3. Interstitial Modal in Cart

Click on the up-sell cart offer button and confirm the Extend Interstitial Modal is presented. Selecting “Add Protection” will add the corresponding Extend Protection Plan to the cart. Selecting “No, thanks” will close the modal.



4. Business Manager

4.1. ExtendContractsQueue custom object

If the **Extend Integration Method** custom preference is set to **Orders API on Schedule**, after placing an order, a new instance of this custom object is created. This holds all the






data needed for generating an Extend contract or leadToken.

[Merchant Tools](#) > [Custom Objects](#) > [Custom Objects](#) > 00010301 - General

General

Manage '00010301' (ExtendOrderQueue)

Fields with a red asterisk (*) are mandatory. You can view and edit the name and description in other languages, if required

Extend Order Data	
 Order Number:*	<input type="text" value="00010301"/>
Creation Date:*	<input type="text" value="06/13/2022"/> : <input type="text" value="9:39 am"/> MM/dd/yyyy h:mm a
Last Modified:*	<input type="text" value="06/13/2022"/> : <input type="text" value="9:39 am"/> MM/dd/yyyy h:mm a
Order Total:	<input type="text" value="76,225.00"/> (Number)
Site currency:	<input type="text" value="USD"/>
Order Customer:	<input "="" type="text" value='{"phone":"3333333333","email":"testing@gmail.com","name":"Testing Customer"}'/>
Order Shipping Address:	<input "="" type="text" value='{"address1":"testAddress1","address2":"testAddress2","city":"testCity"}'/>
Log information:	<input type="text"/>

4.2. Extend Order Creation Job

This job will process the ExtendContractsQueue object instances and will generate a contract for each instance. After successfully executing the request, the objects will be deleted.

4.3. ExtendContractsQueue custom object

If the **Extend Integration Method** custom preference is set to **Contracts API on Schedule**, after placing an order containing at least 1 Extend Protection Plan, a new instance of this custom object is created. This holds all the data needed for generating an Extend contract.





[Merchant Tools](#) > [Custom Objects](#) > [Custom Objects](#) > 8f6c7fc79e60c41637321dab6a - General

General

Manage '8f6c7fc79e60c41637321dab6a' (ExtendContractsQueue)

Fields with a red asterisk (*) are mandatory. You can view and edit the name and description in other languages, if required. Click **Apply** to save the details.

contractData	
Site Currency:	USD
Extend Warranty Plan:	{"purchasePrice":1364,"planId":"10001-misc-elec-base
SFCC Product:	{"referenceId":"008885538465M","purchasePrice":724
Last Modified:*	12/18/2019 : 9:50 pm MM/dd/yyyy h:mm a
Creation Date:*	12/18/2019 : 9:50 pm MM/dd/yyyy h:mm a
Order Customer:	{"phone":"4157477799","email":"kris@extend.com","nan
Log:	
Line Item UUID:*	8f6c7fc79e60c41637321dab6a
Order Total:	9,238.00 (Number)
Order Number:	00000907

4.4. Extend Contracts Creation Job

This job will process the ExtendContractsQueue object instances and will generate a contract for each instance. After successfully executing the request, the objects will be deleted.





Questions? Feedback?

At Extend, customer experience is our North Star, and that goes for both our merchant customers (you!) and your store's customers as they buy your products and our protection plans. So, we want to do anything we can to make your Extend integration successful and to make your customers' experiences outstanding.

If you have any questions, please don't hesitate to reach out to us any time or contact integrations@extend.com for help. And if you have any feedback about our documentation, integration process, or our APIs and SDK, we are all ears!

