

Headless (OCAPI) Configuration guide

**Version**: 4.7.1

**Published date**: July 2025

Copyright © 2025 by ESW. All rights reserved. Information contained herein is subject to change without notice.



**SFCC - ESW Cartridge**

+

Contents

[1 Headless (OCAPI) Architecture 3](#_Toc201821759)

[1.1 Configure OCAPI 3](#_Toc201821760)

[1.2 Add Shop API Resources 3](#_Toc201821761)

[1.3 OCAPI Endpoint Customizations 4](#_Toc201821762)

[1.3.1 Product Detail Page (PDP) 4](#_Toc201821763)

[1.3.2 Product Listing Page (PLP/CLP) 6](#_Toc201821764)

[1.3.3 Cart or Basket Page 8](#_Toc201821765)

[1.3.4 ESW Checkout Redirect 11](#_Toc201821766)

[1.3.5 Order Confirmation 15](#_Toc201821767)

[1.3.6 Generic Price Conversion Endpoint 15](#_Toc201821768)

[1.3.7 Order Cancelation 16](#_Toc201821769)

[1.3.8 ESW Headless Cart Rebuild From Order 16](#_Toc201821769)

[2 ESW Headless Integration Configuration site preference 19](#_Toc201821770)

Headless (OCAPI) Architecture

Configure OCAPI

This section describes how to configure Shop API resources under OCAPI settings. ESW recommends that you configure Shop API version 23.1 or later for optimum utilization of the cartridge.

**Note**

You can generate a new client ID and password by logging in to the [Commerce Cloud Account Manager](https://account.demandware.com/dwsso/XUI/?realm=%2F&goto=https%3A%2F%2Faccount.demandware.com%3A443%2Fdwsso%2Foauth2%2Fauthorize%3Fresponse_type%3Dcode%26client_id%3D70e0d585-9115-4cd2-89cd-1c8758ed2ce2%26scope%3Dopenid%2520roles%2520tenantFilter%2520profile%26redirect_uri%3Dhttps%253A%252F%252Faccount.demandware.com%253A443%252Fdw%252Foidc%252Fopenid_connect_login%26nonce%3D2285cdce6299e%26state%3D3f09c85416588%26prompt%3D#login/).

Add Shop API Resources

For the cartridge to work, you must add Shop API resources to the cartridge.

To add resources:

1. On the target site, go to **Administration** > **Site Development** > **Open Commerce API Settings**.
2. From the **Select type**drop-down, select **Shop**.



1. From the **Select Context** drop-down, select the appropriate context, for example, Global (organization-wide) or site-specific.
2. Enter the OCAPI data API resources provided in the **ocapi-configurations.json** file in the **documentation**folder.
3. Update the **client\_id** with the one you generated from the Commerce Cloud Account Manager.
4. Click **Save**.

Note: **int\_eshopworld\_headless** is an overlay cartridge that includes all OCAPI customizations. You can duplicate the customizations in your cartridge by including the function calls in your OCAPI hook call files.

OCAPI Endpoint Customizations

Product Detail Page (PDP)

* + - 1. OCAPI Endpoint

GET http://hostname:port/dw/shop/v{OCAPI\_VERSION}/products/{id}?expand=prices

* + - 1. Request Parameters

| **ESW Parameters** | **Description** | **Notes** |
| --- | --- | --- |
| country-code | (Required) Two-letter ISO code of the shopper's country. | - |
| currency-code | (Optional) Three-letter ISO code of the shopper's local currency. | If the currency code is provided in the request, the ESW plugin uses this currency. If not provided, the ESW plugin uses the currency code provided in the **ESW Default Country Currency Mapping** site preference. |
| currency | (Optional) Three-letter ISO code of the shopper's local currency. | This is a standard OCAPI parameter. The parameter is required for Fixed Pricing and Override Price Books. |

* + - 1. OCAPI Response

Product Document (Out of the box pricing attributes are converted and assigned to ESW custom attributes.)

|  |  |  |
| --- | --- | --- |
| **ESW Attribute** | **Type** | **Product Base Pricing Attribute** |
| c\_eswPrice | Decimal | price |
| c\_eswPrice\_max | Decimal | price\_max |
| c\_eswPrice\_per\_unit | Decimal | price\_per\_unit |
| c\_eswPrice\_per\_unit\_max | Decimal | price\_per\_unit\_max |
| c\_eswPrices | Map[String, Number] | prices |
| c\_shopperCurrency | String - 3 Char ISO | - |
| c\_eswReturnProhibited | Boolean - true/false if  product return is prohibited  in current shopper country. |  |
| c\_eswRestrictedProduct | Boolean - true/false if  product is restricted  in current shopper country. |  |

* + - 1. Customizations

|  |  |  |
| --- | --- | --- |
| **Extension Point** | **ESW Extension Script** | **Notes** |
| dw.ocapi.shop.product.beforeGET | /int\_eshopworld\_headless/cartridge/scripts/ocapi/shop/hooks/setOverridePriceBook.js | This extension script is mainly used for Fixed Pricing and Override Price Book to get prices directly from the price book. |
| dw.ocapi.shop.product.modifyGETResponse | /int\_eshopworld\_headless/cartridge/scripts/ocapi/shop/hooks/pdpPriceHook.js | This extension script is used for dynamic/calculated prices. This script skips dynamic calculations for Fixed Pricing model countries. |

* + 1. Product Listing Page (PLP/CLP)

GET http://hostname:port/dw/shop/v20\_10/product\_search?expand=prices

You can also add search-related query parameters to the URL as well.

* + - 1. Request Parameters

|  |  |  |
| --- | --- | --- |
| **ESW Parameters** | **Description** | **Notes** |
| country-code | (Required) Two-letter ISO code of the shopper's country. | - |
| currency-code | (Optional) Three-letter ISO code of the shopper's local currency. | If the currency code is provided in the request, the ESW plugin uses this currency. If not provided, the ESW plugin uses the currency code provided in the **ESW Default Country Currency Mapping** site preference. |
| currency | (Optional) Three-letter ISO code of the shopper's local currency. | This is a standard OCAPI parameter. The parameter is required for Fixed Pricing and Override Price Books. |

* + - 1. OCAPI Response

Product Search Result Hits (Out of the box pricing attributes are converted and assigned to ESW custom attributes.)

|  |  |  |
| --- | --- | --- |
| **ESW Attribute** | **Data Type** | **Product Base Pricing Attribute** |
| c\_eswPrice | Decimal | price |
| c\_eswPrice\_max | Decimal | price\_max |
| c\_eswPrice\_per\_unit | Decimal | price\_per\_unit |
| c\_eswPrice\_per\_unit\_max | Decimal | price\_per\_unit\_max |
| c\_eswPrices | Map[String, Number] | prices |
| c\_shopperCurrency | String - 3 Char ISO | - |
| c\_eswReturnProhibited | Boolean - true/false if  product return is prohibited  in current shopper country. |  |
| c\_eswRestrictedProduct | Boolean - true/false if  product is restricted  in current shopper country. |  |

* + - 1. Customizations

|  |  |  |
| --- | --- | --- |
| **Extension Point** | **ESW Extension Script** | **Notes** |
| dw.ocapi.shop.product  .beforeGET | /int\_eshopworld\_headless/cartridge/scripts/okapi  /shop/hooks/setOverridePriceBook.js | This extension script is mainly used for Fixed Pricing and Override Price Book to get prices directly from the price book. |
| dw.ocapi.shop.product  \_search.modifyGETResponse | /int\_eshopworld\_headless/cartridge/scripts/okapi  /shop/hooks/plpPriceHook.js | This extension script is used for dynamic/calculated prices. This script skips dynamic calculations for Fixed Pricing model countries. |

* + 1. Cart or Basket Page

POST http://hostname:port/dw/shop/v{OCAPI\_VERSION}/baskets

* + - 1. Request Parameters

|  |  |  |
| --- | --- | --- |
| **ESW Parameters** | **Description** | **Notes** |
| country-code | (Required) Two-letter ISO code of the shopper's country. | - |
| currency-code | (Optional) Three-letter ISO code of the shopper's local currency. | If the currency code is provided in the request, the ESW plugin uses this currency. If not provided, the ESW plugin uses the currency code provided in the **ESW Default Country Currency Mapping** site preference. |

* + - 1. OCAPI Response

Basket Document.

|  |  |  |
| --- | --- | --- |
| **ESW Attribute** | **Data Type** | **Basket API Object - Base Pricing Attribute** |
| c\_eswOrderTotal | Decimal | totalGrossPrice |
| c\_eswShopperCurrency | String - 3 Char ISO | - |
| c\_eswEstimatedShippingTotal | Decimal | shippingTotalPrice |
| c\_eswOrderDiscount | Decimal | eswOrderDiscount |
| c\_ eswSubTotal | Decimal | eswSubTotal |

|  |  |  |
| --- | --- | --- |
| **ESW Attribute** | **Data Type** | **Basket API Object - ProductLineItem - Base Pricing Attribute** |
| c\_eswPrice | Decimal | price |
| c\_eswBasePrice | Decimal | basePrice (subtotal) |
| c\_eswPriceAfterOrderDiscount | Map[String, Number] | proratedPrice |
| c\_eswReturnProhibited | Boolean - true/false if  product return is prohibited  in current shopper country. |  |
| c\_eswRestrictedProduct | Boolean - true/false if  product is restricted  in current shopper country. |  |
| c\_eswUnitPrice | Decimal | basePrice (per unit) |

Note: If retailer wants to convert additional custom attributes prices at basket and product line item level, can use SFCC site preferences ([ESW Headless Integration Configuration](https://know.eshopworld.com/space/AE/255557757/Merchant+Tools#ESW-Headless-Integration-Configuration)).

* + - 1. Customizations

|  |  |  |
| --- | --- | --- |
| **Extension Point** | **ESW Extension Script** | **Notes** |
| dw.ocapi.shop.basket.afterPOST | /int\_eshopworld\_headless/cartridge/scripts/ocapi/shop/hooks/setOverridePriceBook.js | This extension script is mainly used for Fixed Pricing and Override Price Book to get prices directly from the price book. |
| dw.order.calculate | /int\_eshopworld\_headless/cartridge/scripts/cart/calculate.js | The standard SFCC calculate.js extension script calculates (or recalculates) the basket sent in the request.  The customized/ added function call in the standard extension script (eswBasketPriceConversions) is used for dynamic/calculated prices. This function skips dynamic calculations for Fixed Pricing model countries. |
| dw.ocapi.shop.basket.modifyPATCHResponse | /int\_eshopworld\_core/cartridge/scripts/headless/shop/hooks/basketHooks.js | **Used in Enhanced Order Flow Only**  This hook is responsible to call the ESW checkout Api. |
| dw.ocapi.shop.basket.beforePATCH | /int\_eshopworld\_core/cartridge/scripts/headless/shop/hooks/basketHooks.js | **Used in Enhanced Order Flow Only**  This hook is responsible to set override pricebooks and handle esw basket attributes. |

* + 1. ESW Checkout Redirect

SFCC Order OCAPI and ESW Checkout API

POST http://hostname:port/dw/shop/v{OCAPI\_VERSION}/orders

* + - 1. Request Parameters

|  |  |  |
| --- | --- | --- |
| **ESW Parameters** | **Description** | **Notes** |
| country-code | (Required) Two-letter ISO code of the shopper's country. | - |
| currency-code | (Optional) Three-letter ISO code of the shopper's local currency. | If the currency code is provided in the request, the ESW plugin uses this currency. If not provided, the ESW plugin uses the currency code provided in the **ESW Default Country Currency Mapping** site preference. |

**OCAPI Response**

orderResponse Document (Appended with ESW PreOrder Service Response)

**ESW PreOrder Service Response - SUCCESS**

"c\_eswPreOrderResponseStatus": "OK",

"c\_eswPreOrderResponse": {

        "preOrderCode": "8e2275b1-34a0-474c-97a5-f862a2dfa432",

        "orderNumber": "8e2275b1-34a0-474c-97a5-f862a2dfa432",

        "redirectUrl": "https://checkout-ui-plugus.sandbox.eshopworld.com/se/8e2275b1-34a0-474c-97a5-f862a2xxxxxxxx",

        "shopperAccessToken": "eyJhbGciOiJSUzI1NiIsImtpZCI6IjQ3NkQ1QzE5QzI1Nzc0NUVDOTQ2MDlGNTQ1REY0QjU2QzdERTM5NTNSUzI1NiIsInR5cCI6IkpXVCIsIng1dCI6IlIyMWNHY0pYZEY3SlJnbjFSZDlMVnNmZU9WTSJ9.eyJuYmYiOjE2MzM1Mjg3MjEsImV4cCI6MTYzMzUzMjMyMSwiaXN"

    }

**Note:** “shopperAccessToken” is an optional attribute in response and will only show if ESW checkout request created by registered shopper. This token will further use to achieve Stored Payment Card feature. For more details & how to enable this click [here](https://know.eshopworld.com/space/AE/723714049/Stored+Payment+Cards+Integration).

**ESW PreOrder Service Response - FAILED**

"c\_eswPreOrderResponseStatus": "ERROR",

"c\_eswPreOrderResponse": [

{

"message": "RetailerCartId is required.",

"code": 8

}

]

**Customizations**

|  |  |  |
| --- | --- | --- |
| **Extension Point** | **ESW Extension Script** | **Notes** |
| dw.ocapi.shop.order.beforePOST | /int\_eshopworld\_headless/cartridge/scripts/ocapi/shop/hooks/ orderHooks.js | This extension script is used for Fixed Pricing and Override Price Book to get prices directly from the price book based on the basket currency.  The orderHooks.js hook is used to set the following Basket attributes with faux information, which helps to create an order in SFCC:   * Shipping Address * Billing Address * Customer Email Address * ESW Payment Method   The script also sets the following product line item custom attributes that are required and must be saved during order creation:   * ESW Unit Price * ESW Delta Rounding Value |
| dw.ocapi.shop.order.afterPOST | /int\_eshopworld\_headless/cartridge/scripts/ocapi/shop/hooks/orderHooks.js | This extension script is used to set the following Order level custom attributes:   * eswFxrate * eswShopperCurrencyOrderDiscount * eswDeliveryOptions   This script also fetches and stores the override shipping method rates that you can use in the ESW Checkout service.  To store override shipping methods JSON with rates, the ESW plugin uses eswDeliveryOptions, which is an order level custom attribute available in the systemobjects\_headless.xmlmetadata file. |
| dw.ocapi.shop.order.modify POSTResponse | /int\_eshopworld\_headless/cartridge/scripts/ocapi/shop/hooks/orderHooks.js | This extension script is used to make a call to the ESW Checkout service and append the service response in order OCAPI’s [POST] orderResponse document. The response format is mentioned above. |
| dw.ocapi.shop.basket.modifyPOSTResponse | /int\_eshopworld\_headless/cartridge/scripts/ocapi/shop/hooks/basketHooks.js | This script is used to fetch the ESW override shipping methods in basket if selected country has override shipping methods enable |

* + - 1. Basket Creation Request Payload

The following code sample shows the minimum request payload that is required to create an ESW Checkout Order:

{

"product\_items":[

{

"product\_id":"750518699721",

"quantity":1

},

{

"product\_id":"750518699769",

"quantity":1

}

],

"shipments":[

{

"shipping\_method":{

"id":"POST"

}

}

],

"currency":"USD"

}

For information on Checkout API and its request body attributes, see the [Checkout API](https://devportal.eshopworld.com/reference/checkout). For information on the OCAPI endpoints, see [SFCC - Shop API Resources](https://documentation.b2c.commercecloud.salesforce.com/DOC2/index.jsp?topic=%2Fcom.demandware.dochelp%2FOCAPI%2Fcurrent%2Fshop%2FResources%2Findex.html).

* + 1. Order Confirmation

The ESW Order Confirmation is a webhook call initiated from the ESW system to the retailer system. Similar to other architectures, the Headless (OCAPI) Architecture also has a similar flow for Order Confirmation. For information on the Order Confirmation Integration, see [Webhook Overview](https://know.eshopworld.com/space/DT/177046186/Order+Confirmation+Webhook).

For order confirmation requests in Headless OCAPI, the EShopWorldSG-Notify endpoint is used for Sitegenisis compatible architecture and EShopWorld-Notify endpoint is used for SFRA compatible architecture. Configurations for this can be done using the ESW Metadata Items custom site preference under ESW Checkout Configuration (refer to the eShopWorld\_Checkout\_Integration.docx document).

* + 1. Generic Price Conversion Endpoint
       1. SFCC Custom Web Endpoint

**SiteGenisis compatible Architecture:** GET <https://hostname/on/demandware.store/Sites-SiteGenesis-Site/en_US/EswHL-PriceConversion?price=26&country=GB>

**SFRA compatible Architecture:** GET <https://hostname/on/demandware.store/Sites-RefArch-Site/en_US/EswRefArchHL-PriceConversion?price=26&country=GB>

**Request Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **ESW Parameters** | **Type** | **Description** | **Notes** |
| price | Decimal | (Required) Price of the product. | - |
| country | String | (Required) Two-letter ISO code of the shopper's country. | - |
| currency | String | (Optional) Three-letter ISO code of the shopper's local currency. | If the currency code is provided in the request, the ESW plugin uses this currency. If not provided, the ESW plugin uses the currency code provided in the **ESW Default Country Currency Mapping** site preference. |
| applyAdjust | Boolean | (Optional) Indicates if the country adjustment should be applied. | If provided, the country adjustment value is applied to the price accordingly. |
| applyRounding | Boolean | (Optional) Indicates if the prices should be rounded. | If provided, the prices are rounded accordingly. |

**Response**

The formatted converted price in the shopper currency.

For more information on how OCAPI response attributes can be used to display or represent the different prices on the Cart Page, see [Headless (OCAPI) Attributes Mapping with Cart](https://know.eshopworld.com/space/AE/294486123/Headless+(OCAPI)+Attributes+Mapping+with+Cart).

* + 1. Order Cancelation
       1. SFCC Custom Web Endpoint

The ESW Order Cancelation is a webhook call initiated from the ESW system to the retailer system.

For order cancelation requests in Headless OCAPI, the EShopWorldSG- ProcessWebHooks endpoint is used for Sitegenisis compatible architecture and EShopWorld-ProcessWebHooks endpoint is used for SFRA compatible architecture. Configurations for this can be done by importing systemobjects\_headless.xml from sitesdata/meta directory path.

### **ESW Headless Cart Rebuild from Order**

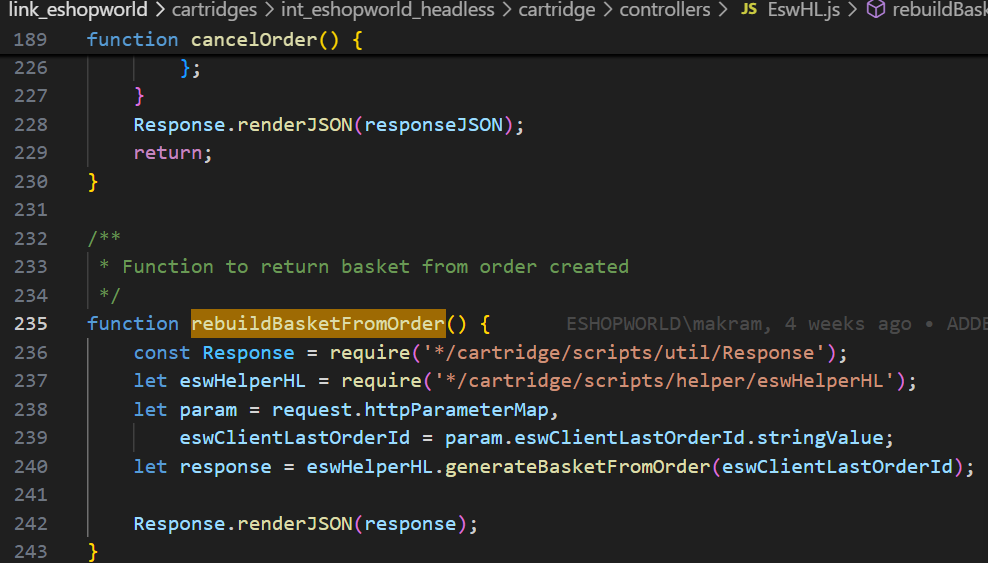
Overview

This update focuses on backend modifications for Cart Rebuild from Order, with no front-end changes. Retailers must pass the OrderID to the endpoint; any front-end adjustments are their responsibility. The EswHL.js file in Salesforce Commerce Cloud manages headless controller actions for ESW integration, handling basket and order interactions, including rebuilding a basket from a prior order, such as after an ESW checkout

**1. Controller Function: rebuildBasketFromOrder**

This controller function is exposed as a **POST endpoint** and is used to rebuild the customer’s basket from a previously created order (eswClientLastOrderId). This is useful when a customer returns to the site after an ESW redirect.

Code Snippet:



**Explanation:**

**Input Parameter:** eswClientLastOrderId (from request).

**Helper Call:** Uses generateBasketFromOrder method to reconstruct the basket.

**Response:** JSON response with reconstructed basket details.

**2.** **Helper Function:** generateBasketFromOrder(eswClientLastOrderId)

To generate a basket based on an existing order, typically used when the order was created but not finalized (e.g., failed, created, or new state).

**Key Steps:**

**Fetch the Order:**

Uses OrderMgr.getOrder(eswClientLastOrderId) to retrieve the order.

Validates if the order is in FAILED, CREATED, or NEW status.

**Rebuild the Cart:**

Calls eswHelper.rebuildCartUponBackFromESW(orderNo) to initiate a rebuild.

If the order status is CREATED, it is explicitly failed (OrderMgr.failOrder).

**Extract Line Items:**

Iterates over ProductLineItems and extracts productId, price, quantity.

**Extract Coupons:**

Calls getRetailerPromoCodes(order) to get coupon codes.

**Create/Fetch Basket:**

Uses BasketMgr.getCurrentOrNewBasket().

**Return Payload:**

Returns an object containing:

orderLineItems

basketId

couponCodes

basketItems

removeLineItems flag

Returned Object Format:

json

{

orderLineItems: { products: [...] },

basketId: "UUID",

couponCodes: [ { code: "PROMO123" }, ... ],

basketItems: [ { productId: "xyz", lineItemId: "uuid" }, ... ],

removeLineItems: true/false

}

**3. Helper Function: getRetailerPromoCodes(order)**

Extracts any coupon codes applied to the order.Checks if couponLineItems exist in the order and Loops through them and collects valid couponCode values.

**Returns:**

An array of objects like:

"couponCodes": [

        {

            "code": "Coupon Code"

        }

    ]

**4. Integration Instruction for Headless Implementation**

To integrate this functionality into a headless environment, developers must invoke the EShopWorld-RebuildBasketFromOrder endpoint using an HTTP POST request. This endpoint requires the query parameter eswClientLastOrderId to identify the target order.

Example Endpoint:

SiteGenisis compatible Architecture: Post [https://hostname/on/demandware.store/Sites- SiteGenesis-Site/en\_US/EswHL-](https://hostname/on/demandware.store/Sites-%09SiteGenesis-Site/en_US/EswHL-) RebuildBasketFromOrder?eswClientLastOrderId=OrderID

SFRA compatible Architecture: Post [https://hostname/on/demandware.store/Sites-RefArch- Site/en\_US/EswRefArchHL- RebuildBasketFromOrder?eswClientLastOrderId=OrderID](https://hostname/on/demandware.store/Sites-RefArch-%09Site/en_US/EswRefArchHL-PriceConversion?price=26&country=GB)

These endpoint is essential for reconstructing the customer's basket based on a previously created order, typically used when returning from the ESW checkout flow

**5. Conclusion**

The above code changes are intended solely for back-end modifications.

Retailers are required to pass the Order ID when triggering the endpoint.

Any front-end related changes should be handled by retailers according to their specific customizations.

. ESW Headless Integration Configuration site preference

**ESW Basket Pricing Mapping**: This preference contains the mapping between the Basket API object attributes and ESW basket custom attributes. The following code sample shows the mapping:

{

"c\_estimatedCharges": "eswEstimatedCharges",

"c\_cartSubtotal": "eswCartSubTotal",

"c\_total-discount": "eswTotalDiscount",

"productLineItems": {

"c\_regular-price": "eswRegularPrice",

"c\_sale-price": "eswSalePrice"

}

}

The following table lists the Basket API object attributes and ESW-specific basket custom attributes used in the mapping. The cartridge takes the value of basket-related attributes, converts these values to shopper prices, and stores the converted price in the ESW basket custom attributes.

|  |  |
| --- | --- |
| **Basket API object attributes** | **ESW basket custom attributes** |
| c\_estimatedCharges | eswEstimatedCharges |
| c\_cartSubtotal | eswCartSubTotal |
| c\_total-discount | eswTotalDiscount |
| c\_regular-price | eswRegularPrice |
| c\_sale-price | eswSalePrice |
| c\_available\_shipping\_methods | eswOverrideShipping |
|  |  |

**ESW Pricebook Keywords - Retailer Currencies/ Keywords**: This preference holds the price book ID keywords (separated by pipeline) that need to be replaced when converting the price book prices.

**Keywords Example**: Retailer Currencies, countries, and so on.

**Value Example**: usd|cad|gbp|abc

**Use Case**

When converting prices, ESW takes the prices from the retailer’s base currency price book and converts these prices to shopper currency prices. If the retailer's base currency price book ID is 'usd-list-prices'/'usd-sale-prices', ESW converts the prices and puts the converted price values against the IDs with 'eSWPriceOf-' as the suffix. This changes the ID from 'usd-list-prices'/ 'usd-sale-prices' to 'eSWPriceOf-usd-list-prices'/ 'eswPriceOf-usd-sale-prices' in the response.

However, as this naming convention could be confusing, the keywords in the price book IDs can be changed to a keyword called 'base'. This means that if the brand name is 'abc' and the price book ID is 'abc-list-prices', the ID will be 'eSWPriceOf-base-list-prices' in the response.