

Version: 4.2.0

Published date: January 2024

Copyright © 2024 by ESW. All rights reserved. Information contained herein is subject to change without notice.

SFCC-PWA INTEGRATION GUIDE



**SFCC – ESW PWA Storefront Cartridge**

+

Table of Contents

[1. ESW PWA cartridges 3](#_Toc155097892)

[1.1. Setup BM Cartridge 3](#_Toc155097893)

[**1.1.1.** **Setup cartridge path** 3](#_Toc155097894)

[1.1.2. Import site data 3](#_Toc155097895)

[**1.1.3.** **Configure PWA related custom preferences** 3](#_Toc155097896)

[2. Configure countries and locales 4](#_Toc155097897)

[2.1. Configure countries custom object 4](#_Toc155097898)

[**2.2.** **Configure Locales** 5](#_Toc155097899)

[3. Code customization 6](#_Toc155097900)

[3.1. Files to override to enable ESW functionalities: 6](#_Toc155097901)

[3.2. Add ESW related configurations in default.js 6](#_Toc155097902)

[3.3. ESW related configurations 7](#_Toc155097903)

[3.4. Init ESW component 8](#_Toc155097904)

[3.5. Override registration and login forms 8](#_Toc155097905)

[3.6. Override Product Item component 10](#_Toc155097906)

[3.7. Override Product view component 11](#_Toc155097907)

[3.8. Override “useDerivedProduct” hook 12](#_Toc155097908)

[3.9. Override “use-add-to-cart-modal” hook 14](#_Toc155097909)

[3.10. Override site-utils.js 15](#_Toc155097910)

[3.11. Override order detail page 16](#_Toc155097911)

[3.12. Override Cart page 17](#_Toc155097912)

[3.13. Override Cart page 19](#_Toc155097913)

[4. Update content slots and messages 19](#_Toc155097914)

[4.1. Update geo ip alert message 19](#_Toc155097915)

[4.2. Update return prohibit message 20](#_Toc155097916)

[4.3. Update prices in callout messages and content 20](#_Toc155097917)

[5. Custom attributes added in PWA 21](#_Toc155097918)

[6. Custom Object Added in PWA 22](#_Toc155097919)

[7. Order confirmation hooks 22](#_Toc155097920)

# ESW PWA cartridges

The ESW PWA cartridge is developed to use with SFCC pwa-kit (<https://github.com/SalesforceCommerceCloud/pwa-kit>). To use this cartridge make sure you already have a working PWA environment.

## Setup BM Cartridge

### **Setup cartridge path**

1. Add following cartridges to your cartridge path
   1. int\_eshopworld\_core
   2. int\_eshopworld\_pwa
2. The cartridge path should be looks like:
   1. int\_eshopworld\_core:int\_eshopworld\_pwa:app\_storefront\_base:modules

### Import site data

After setting up the cartridge path, you can import PWA related custom objects, preferences and content slots from following paths:

* To import system objects
  + link\_eshopworld/sitesdata/meta/systemobjects.xml
  + link\_eshopworld/sitesdata/meta/systemobjects\_pwa.xml
* To import custom objects
  + link\_eshopworld/sitesdata/meta/customobjects.xml
  + link\_eshopworld/sitesdata/meta/customobjects\_pwa.xml
* To import content assets
  + link\_eshopworld/sitesdata/sites/RefArch/libraries.xml
  + link\_eshopworld/sitesdata/sites/RefArch/content-assets.xml

### **Configure PWA related custom preferences**

After importing site data you can configure custom preferences as follows:

1. Goto ` Merchant Tools > Site Preferences > Custom Site Preference Groups > ESW PWA Integration Configuration`
2. Provide the “PWA Storefront URL“ (This is needed because on MRT environment we are not getting server reference)
3. Provide the “ESW PWA URL Expansion Pairs” as follows:
   1. InventoryCheckFailurePageUrl|cart
   2. BaseUrl|
   3. ContinueShoppingUrl|Base url
   4. BackToCartUrl|cart

A screenshot of a computer

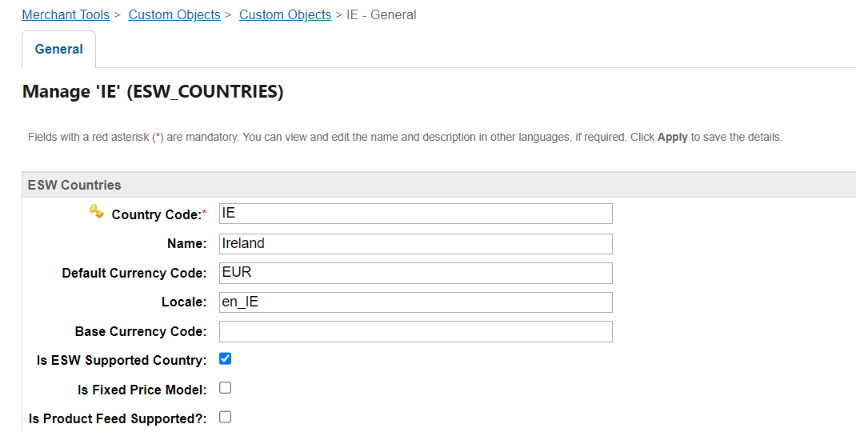
Description automatically generated

# Configure countries and locales

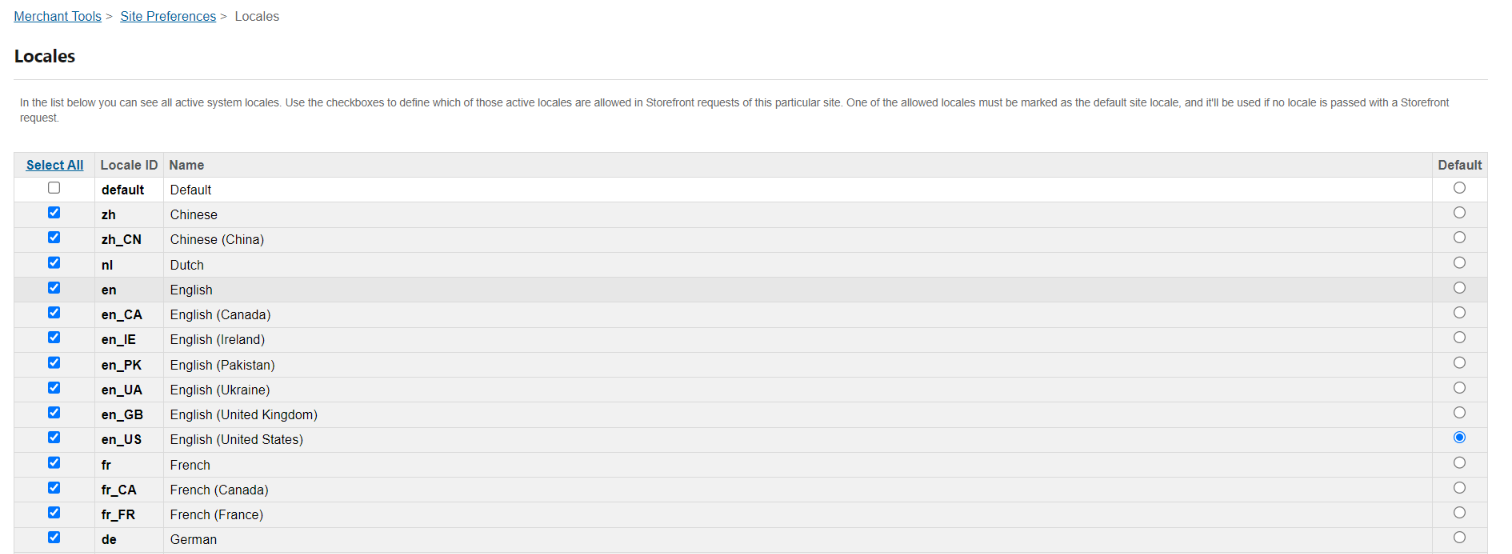
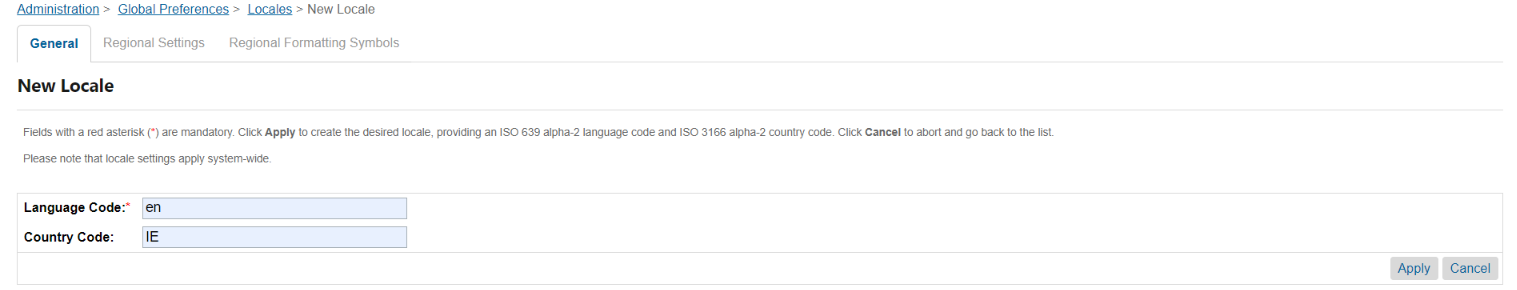
### Configure countries custom object

Countries should be configured in following way:

* Goto ` Merchant Tools > Custom Objects > Manage Custom Objects` and select ESW\_COUNTRIES A screenshot of a computer

  Description automatically generated
* Select country
* Add locale for the country as shown in screenshot 
* **The locale must be in the format of {lang\_Country} e.g. en\_IE, en\_US, en\_CA etc**

### **Configure Locales**

* Now go to “Merchant Tools > Site Preferences > Locales” and check if your given locale is present
* If yes then activate the locale (Mark the locale and press Apply button) 
* If locale is not present then add locale from “Administration > Global Preferences > Locales > New Locale” 
* This will generate locale in form of en\_IE

# Code customization

* After setting up and running up the pwa kit successfully we can add ESW related changes in it.

## Files to override to enable ESW functionalities:

Following files need to be placed in the relevant directories of pwa-kit to enabled ESW checkout:

* config/
  + default.js (To add ESW cartridge related URLs)
  + sites.js (Override this file so that add only one base country configs)
* override/
  + main.jsx (Add timezone cookie)
  + components/
    - footer/
      * index.js (To call esw init component)
    - forms/
      * login-fields.jsx (Prefilled login email for post checkout registration)
      * registration-fields.jsx (Prefilled registration fields for post checkout registration)
    - product-item/
      * index.jsx (Display return prohibit and restricted product message)
    - product-view/
      * index.jsx (Display return prohibit and restricted product message)
  + hooks/
    - use-derived-product.js (Setting return restricted product message)
    - use-add-to-cart-modal.js (Override checkout button in add to cart modal)
  + utils/
    - site-utils.js (Getting site.js configurations from the BM instead of site.js file)
  + pages
    - account/
      * order-detail.jsx (Display order tracking information from ESW)
  + cart/
    - partials/
      * cart-cta.jsx (override checkout button)
    - index.jsx
  + routes.jsx
* translations/
  + en-US.json (Add message for restricted products, retailer will need to update relevant translation file)

## Add ESW related configurations in default.js

* Go to /configs folder
* Add esw-config.js file

## ESW related configurations

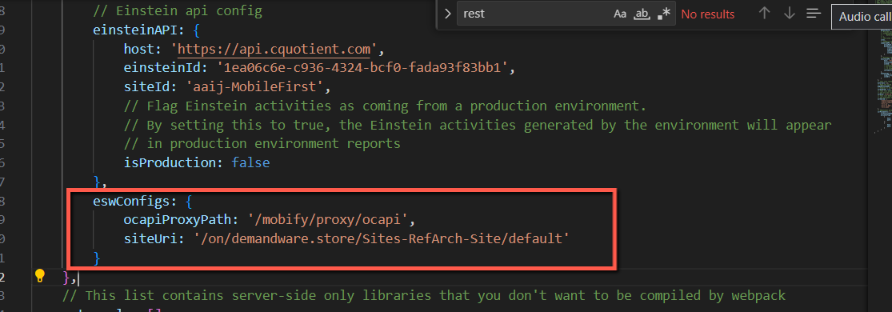
Add the following configuration in default.js file in `app:

        eswConfigs: {

            ocapiProxyPath: '/mobify/proxy/ocapi',

            siteUri: '/on/demandware.store/Sites-RefArch-Site/default'

        }

The config will look like: 

Make sure to add correct path for eswConfigs.

The sites.js file should have following content

module.exports = [

    {

        id: 'RefArch',

        l10n: {

            supportedCurrencies: ['USD'], //  Retailer's base currency

            defaultCurrency: 'USD', //  Retailer's base currency

            defaultLocale: 'en-US', //  Retailer's base country locale

            supportedLocales: [

                {

                    id: 'en-US', //  Retailer's base country locale

                    alias: 'us', // country code in lowercase for URL

                    preferredCurrency: 'USD', // Retailer's base currency

                    supportedLocales: ['en-US'], // Retailer's base country locale

                    isFixedPriceModel: true,

                    isSupportedByESW: true,

                    actualCurrency: 'USD',

                    countryCode: 'US' // country code in uppercase for PWA cartridge use

                }

            ]

        }

    }

]

## Init ESW component

This component is responsible for the following:

* Getting ESW related configurations
* Display geo ip alert if user changes the location
* Cart rebuild when user come back to cart from the ESW checkout

To add this component please do the following:

* Go to /overrides/components/footer/index.jsx
  + Copy the file from pwa-kit if its not already there
* Import EswInit component

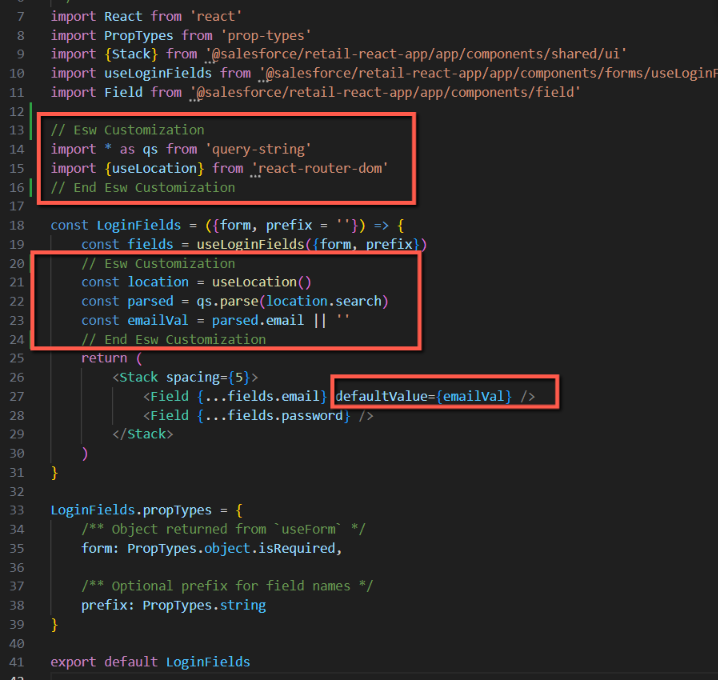
import {EswInit} from '../../esw/components/esw-init'

* Add EswInit component as shown in figureA screen shot of a computer program

  Description automatically generated
* **EswInit component call APIs during initial rendering of the app**

## Override registration and login forms

This is require if you are using shopper registration (<https://know.eshopworld.com/space/AE/848003073/Checkout+Shopper+Registration>) feature of ESW. These forms are required to prefill the values of relevant fields in login and registration forms. Retailer can override these forms by following steps:

* Go to /overrides/components/forms
  + Create the folder if not already exist
* Copy and paste login-fields.jsx from ESW pwa cartridge
  + Add following codes in file 
* Copy and paste registration-fields.jsx from ESW pwa cartridge
  + Add following codes in file A screenshot of a computer program

    Description automatically generated

## Override Product Item component

This is require if you are using Return prohibited (<https://know.eshopworld.com/space/AE/255852638/Global+Product+Returns+Prohibition>) feature of ESW. We will override this component by adding “EswReturnProhibitMsg” component. To add component follow these steps:

* Go to /overrides/components/product-item/index.jsx
  + If file is not there then get it from pwa cartridge
* Import the component

import {EswReturnProhibitMsg} from '../../esw/components/product-return-prohibit-msg'

* Call the component as shown A screen shot of a computer program

  Description automatically generated
* **We will call this component in /overrides/pages/cart/index.jsx in later steps**

## Override Product view component

This is require if you are using Return prohibited (<https://know.eshopworld.com/space/AE/255852638/Global+Product+Returns+Prohibition>) feature of ESW. We will override this component by adding “EswReturnProhibitMsg” component. To add component follow these steps:

* Go to /overrides/components/product-view/index.jsx
  + If file is not there then get it from pwa cartridge
* Import the component

import {EswReturnProhibitMsg} from '../../esw/components/product-return-prohibit-msg'

* Call the component as shown A screen shot of a computer program

  Description automatically generated

## Override “useDerivedProduct” hook

This is require if you are using restricted product (<https://know.eshopworld.com/space/AE/255557739/Global+Product+Restriction>) feature of ESW. We will override this hook to add restrictions message. To override this hook please follow these steps:

* Go to /overrides/hooks/use-derived-product.js
  + If file is not there then get it from pwa cartridge
* Add following code as shown A screenshot of a computer program

  Description automatically generated
* Since we have updated hook which is using translation to show restricted product message. We also need to add the key in our translation file
  + Go to /translations/en-US.json
  + Add following code in the file

  "esw.product.restricted.msg": {

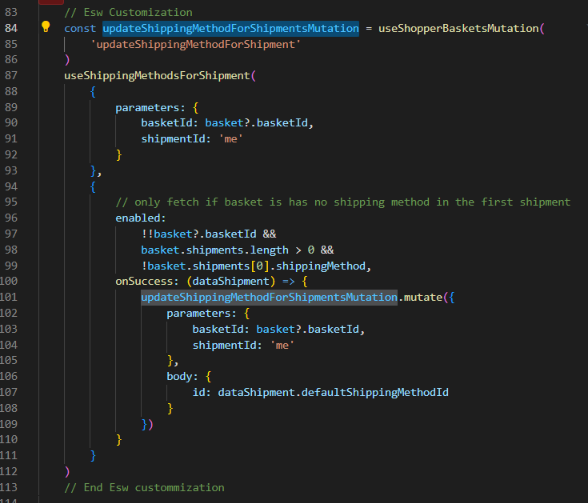
    "defaultMessage": "{eswProductRestrictedMsg}"

  },

* + **Retailer have to use specific translation file to add translation**

## Override “use-add-to-cart-modal” hook

This is require to implement ESW checkout on add to cart modal. To override the hook follow these steps:

* Go to /overrides/hooks/use-derived-product.js
  + If file is not there then get it from pwa cartridge
* Add code to add default shipment to basket
* Import required ESW modules

// Esw Customization

import { EswCheckoutBtn } from '../esw/components/checkout-btn'

import {

    useShippingMethodsForShipment,

    useShopperBasketsMutation

} from '@salesforce/commerce-sdk-react'

// end Esw Customization

* In place of checkout Button add following:

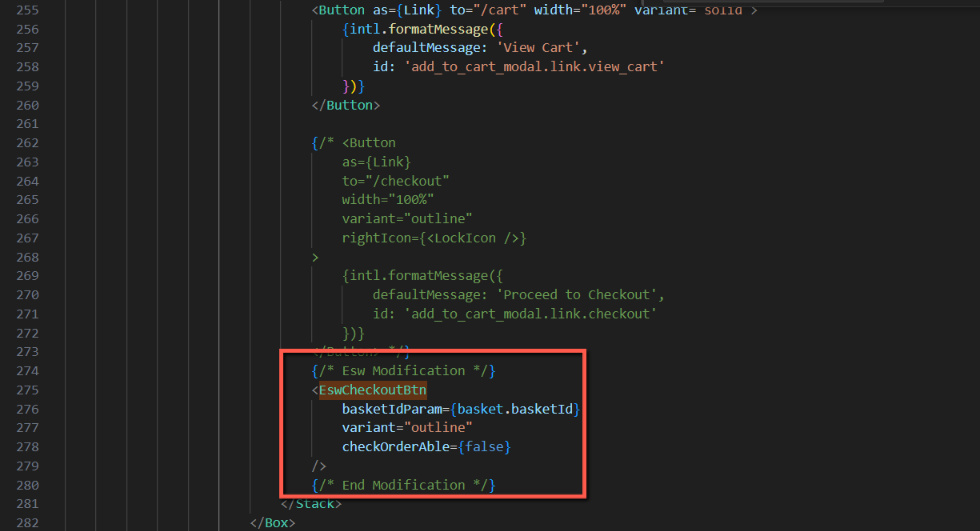
<EswCheckoutBtn

   basketIdParam={basket.basketId}

   variant="outline"

   checkOrderAble={false}

/>

* As shown in screenshot 
* Make sure to change it in two places (<ModalBody/> and <ModalFooter/>) components

## Override site-utils.js

This is required to get countries from the BM configurations. To override this file please follow these steps:

* Go to /overrides/utils/site-utils.js
  + If file is not there then get it from pwa cartridge
* Import esw-services.js in the file

import {getSupportedCountries} from '../esw/esw-services'

* Update the function getSites()

/\*\*

 \* ESW customized function to return the list of sites that has included their respective aliases

 \* @return {array} sites - list of sites including their aliases

 \*/

export const getSites = () => {

    let sitesFromConfig = getConfig().app

    // sites.js from the BM

    getSupportedCountries()

        .then((response) => response.json())

        .then((data) => (sitesFromConfig.sites = data.allowedCountries))

    let {sites = [], siteAliases = {}} = sitesFromConfig || {}

    if (!sites.length) {

        throw new Error("Can't find any sites from the config. Please check your configuration")

    }

    return sites.map((site) => {

        const alias = siteAliases[site.id]

        return {

            ...site,

            ...(alias ? {alias} : {})

        }

    })

}

* As shown in screenshot A screen shot of a computer

  Description automatically generated

## Override order detail page

This is require to display order tracking information from ESW. To override this file please follow these steps:

* Go to /overrides/pages/account/order-detail.jsx
  + If file is not there then get it from pwa cartridge
* Import EswOrderTracking component

import ESWOrderTracking from '../../esw/components/order-tracking/index'

* Call the component as shown in screenshot A screen shot of a computer

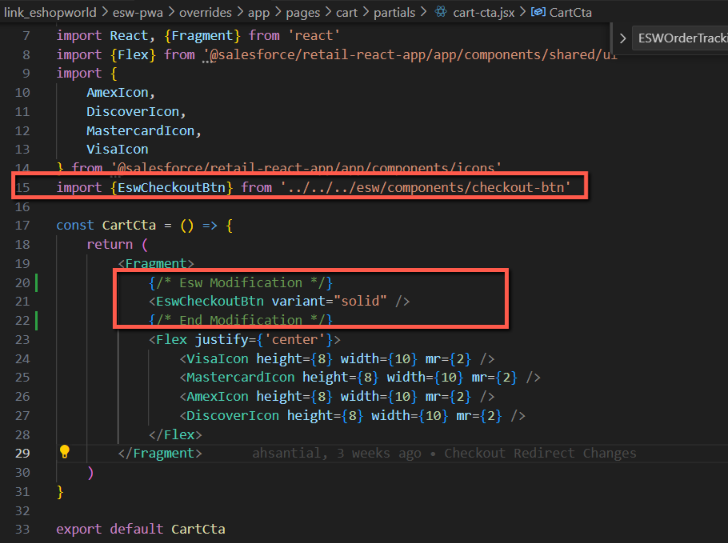
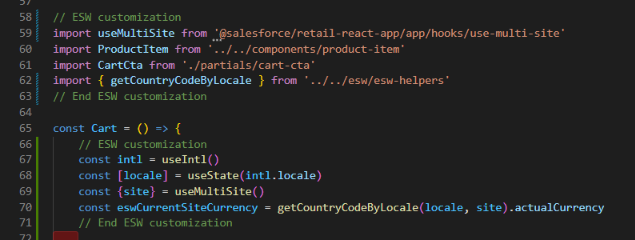
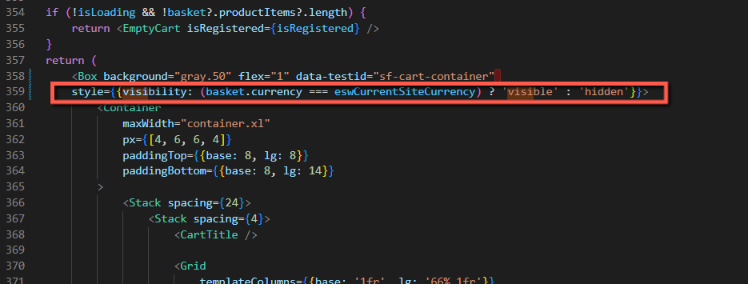
  Description automatically generated

## Override Cart page

This is require to override checkout functionality. So that if country is supported by ESW then it will redirect to esw checkout otherwise OOTB pwa checkout will work. To override this file please do the following:

* Go to /overrides/pages/cart/partials/cart-cta.jsx
  + If file is not there then get it from pwa cartridge
* Import EswCheckoutBtn component

import {EswCheckoutBtn} from '../../../esw/components/checkout-btn'

* Call the component as shown in screenshot 
* After completing above step, call the component in cart page so that
  + Go to /overrides/pages/cart/index.jsx
  + Modify import of cart-cta as follows 
* Add condition on product item grid so that product price should not blink on page load
* Since we have updated page so we need to override routes too
  + Go to /overrides/routes.jsx
  + Import Cart

import Cart from './pages/cart'

* + Call this in routes as

{

path: '/cart',

component: Cart,

exact: true

}

## Override Cart page

As MRT is not providing user’s country so we are getting it from the client timezone. For this reason we need to add “esw.shopperTimezone” cookie which we will get on the server to get the shopper country to show more accurate geo ip related results. To override this:

* + Go to /overrides/main.jsx
  + Add following piece of code in it

// Esw: Customization

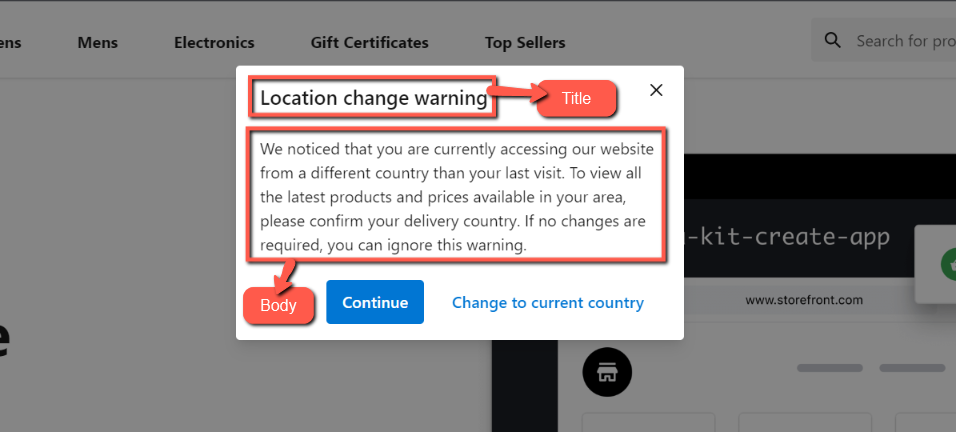
document.cookie = "esw.shopperTimezone=" + Intl.DateTimeFormat().resolvedOptions().timeZone + ";secure"

# Update content slots and messages

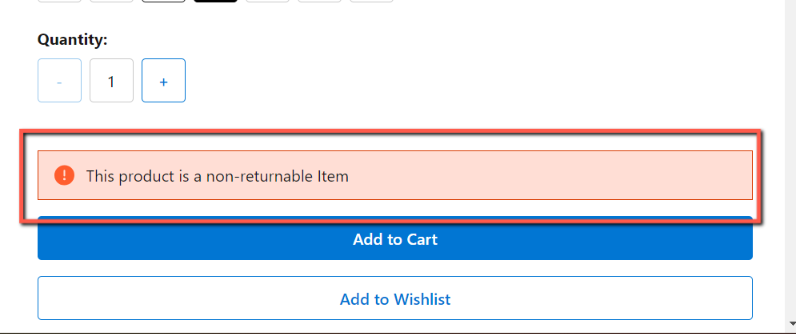
This section describes how to update different messages we are getting from the BM

## Update geo ip alert message

If retailer is using geo lookup (<https://know.eshopworld.com/space/AE/255754260/ESW+Storefront+Components#ESW-General-Configuration-Site-Preferences>) then its alert messages can be customized from content asset “eswGeoIpChangeWarning”. The title is the name of the content asset while the body of the content asset



## Update return prohibit message

If retailer is using return prohibit feature then its message can be updated from the body of content asset “esw-display-return-prohibited-message” 

## Update prices in callout messages and content

This is used to calculate prices from retailer’s currency to shopper currency. To enable this feature please follow (<https://know.eshopworld.com/space/AE/475332611/Localized+Front-End+Prices>). We then need to convert prices in shopper currency on the frontend. Please use convertPrice() function from esw-helpers to convert prices after getting in the DOM.

**Working example can be found in ‘/overrides/pages/esw-example-content-details/index.jsx` and can be accessed via route `/content/esw-test-content` **

# Custom attributes added in PWA

Product custom attributes

|  |  |
| --- | --- |
| **ESW Attribute** | **Data Type** |
| c\_eswReturnProhibited | Boolean - true/false if  product return is prohibited  in current shopper country. |
| c\_eswReturnProhibitMsg | String - Message to show for return prohibit products |
| c\_eswRestrictedProduct | Boolean - true/false if  product is restricted  in current shopper country. |
| c\_eswRestrictedProductMsg | String - Message to show if product is restricted |

Basket custom attributes

|  |  |  |
| --- | --- | --- |
| c\_eswShopperCurrency | String - 2 Char ISO | - |

Custom preferences attributes

|  |  |  |
| --- | --- | --- |
| eswPwaUrl | String – Storefront URL for PWA | Example: http://localhost:3000 |
| eswPwaUrlExpansionPairs | Set of Strings: (URI configurations for different actions) | Values:  InventoryCheckFailurePageUrl|cart  BaseUrl|  ContinueShoppingUrl|Base url  BackToCartUrl|cart |

# Custom Object Added in PWA

The Following new custom object are added for PWA

**ESW\_PWA\_ORDER\_DATA:**

No manual configuration is necessary for this custom object, as it's exclusively utilized by the cartridge to facilitate the post-registration feature for order number storage. The cartridge autonomously manages both the creation and destruction of this object, ensuring seamless functionality without any additional administrative overhead.

# Order confirmation hooks

These hooks can be found and configured from related documentation. No special work requires for PWA