Implementation Guide – Controllers & Pipelines

***Version*** ***19.1.0 (Promo Only)***



**Table of Contents**

[1. Summary 2](#_Toc48741041)

[2. Component Overview 3](#_Toc48741042)

[2.1 Functional Overview 3](#_Toc48741043)

[2.2 Use Cases 3](#_Toc48741044)

[2.3 Limitations, Constraints 3](#_Toc48741045)

[2.4 Compatibility 3](#_Toc48741046)

[2.5 Privacy, Payment 3](#_Toc48741047)

[3. Implementation Guide 4](#_Toc48741048)

[3.1 Setup 4](#_Toc48741049)

[3.1.1 Deploying cartridge to a sandbox 4](#_Toc48741050)

[3.1.2 Sandbox setup 5](#_Toc48741051)

[3.2 Configuration 6](#_Toc48741052)

[3.2.1 Site Preferences configuration 6](#_Toc48741053)

[3.3 Custom Code (Both Controllers & Pipelines) 9](#_Toc48741054)

# Summary

This document describes how to implement Affirm cartridge in SiteGenesis site. This cartridge can be configured in the Business Manager and contains all elements necessary to perform a successful best practices implementation of Affirm.

# Component Overview

## Functional Overview

“Buy with Affirm” is a closed-end installment loan product offered by through Affirm’s technology platform. “Buy with Affirm” allows you to buy goods or services offered by online merchants with whom Affirm partners (“Merchants”). If you agree to use “Buy with Affirm,” Cross River Bank will pay the Merchant on your behalf in exchange for your promise to repay the same amount plus a finance charge as determined by your creditworthiness.

## Use Cases

On the PDP, PLP and Cart Page Customers can see Affirm promotions beneath the price displayed monthly payment.

## Limitations, Constraints

Affirm cartridge has the following limitation for purchasing products. List of goods that is excluded from cartridge support for buying: Gift Certificates

Cartridge handles default implementation of Gift Certificates in Salesforce platform. Any 3rd party integration that is not based on Salesforce Gift Certificate functionality will NOT be handled by the cartridge and requires additional development.

## Compatibility

Cartridge is designed and developed for: Salesforce platform version 19.1 (Compatibility Mode 19.1)

## Privacy, Payment

Affirm stores and processes your information maintaining physical, electronic and procedural safeguards. We maintain physical security measures to guard against unauthorized access to systems and use safeguards such as firewalls and data encryption. We enforce physical access controls to our buildings, and we authorize access to personal information only for those employees who require it to fulfill the responsibilities of their jobs.

# Implementation Guide

## Setup

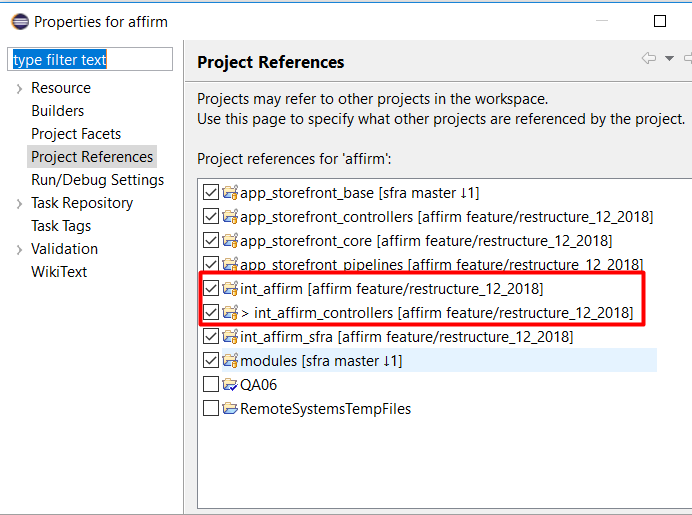
This section describes steps that should be completed before cartridge configuration in Business Manager.

### Deploying cartridge to a sandbox

1. Download cartridge source code.
2. Import the Affirm cartridges to a workspace in Salesforce UX Studio.



1. Add Affirm cartridges (int\_affirm, int\_affirm\_controllers) to Project Reference of Server Connection.

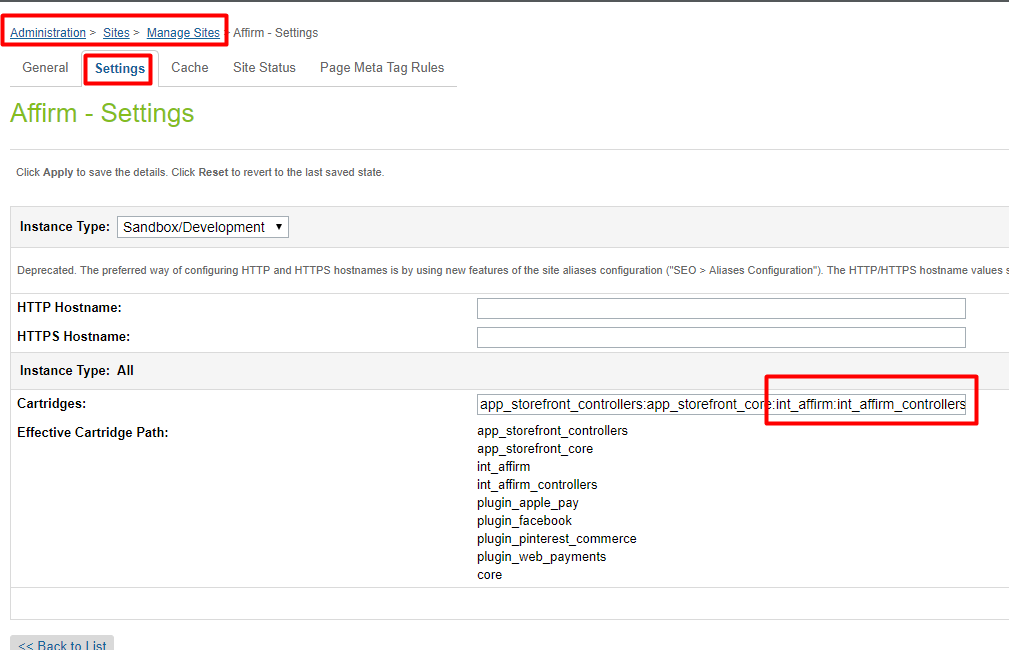


1. Wait until Studio completes workspace build and uploading source codes to a sandbox.

### Sandbox setup

1. Go to Business Manager -> Site -> Manage Sites. Select correct site, then select Settings tab. In cartridge path at the end add the following:

:int\_affirm:int\_affirm\_controllers



1. Download cartridge source code.
2. Find metadata/affirm/meta/system-objecttype-extensions.xml
3. Go to Administration > Site Development > Import & Export > Upload Import Files. Click the Upload button and select system-objecttype-extensions.xml  
   A screenshot of a cell phone

   Description automatically generated
4. Go to Meta Data > Import and import the uploaded file.A screenshot of a cell phone

   Description automatically generated
5. Verify that Site Preferences group was created with ID - AFFIRM\_PROMO and name – Affirm.

## Configuration

This section describes configuration of the sandbox.

### Site Preferences configuration

1. Go to Merchant Tools > Site Preferences > **Custom Preferences** > Affirm
2. Enable attribute - Affirm Online Status. This attribute defines status (enable/disable) of Affirm integration.
3. Select a value from dropdown Affirm Mode. This attribute defines in which mode cartridge will work. Allowable values are “Sandbox” and “Production”.
4. Add site preference attribute - Affirm Public Key with provided public key from Affirm.
5. Add site preference attribute - Affirm Private Key with provided private key from Affirm.
6. Enable attribute - Cart Promo Message. This attribute defines status (enable/disable) of Promotional messages on Cart Page.
7. Enable attribute - Product Promo Message. This attribute defines status (enable/disable) of Promotional messages on Product Detail Page.
8. Enable attribute - PLP Promo Message. This attribute defines status (enable/disable) of Promotional messages on Product Listing Page.
9. Add site preference value attribute - Default financing program. This attribute defines which financing program will be used by default (if any other financing programs have not been applied).
10. Add site preference value - Affirm Financing programs mapping. It allows to map financing programs to promo-id.

Please use the following format: {financing program name}|{promo-id}

Example:



1. Add site preference value - Financing programs customer group. This allows detection of financing programs based on customer group.
2. Please use the following format: {customerGroup}|{financingProgram},
3. Example: Registered|registeredFinProgram



1. Add site preference value - Financing programs cart total range. This allows detection of financing programs based on cart total range.
2. Please use the following format: {minValue}|{maxValue}|{financingProgram}.
3. Example: 100|200|CartRange



1. Add site preference value - Financing programs date range. This allows detection of financing programs based on date range.
2. Please use the following format: {startDate}|{endDate}|{financingProgram}, where startDate and endDate format is "yyyy-MM-dd".
3. Example: 2017-01-01|2017-02-28|myWinter



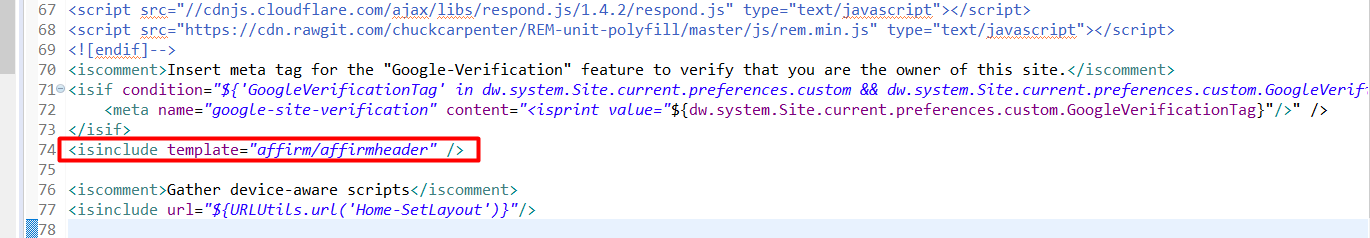
1. Add site preference value - Affirm Min Total. It allows setting minimal price value for displaying promotional messages. Products with price lower than this value will not have promotional message.
2. Add site preference attribute - Payment Min Total with minimal cart total when Affirm payment method will be available.
3. Add site preference attribute - Payment Max Total with maximal cart total when Affirm payment method will be available.

## Custom Code (Both Controllers & Pipelines)

This section describes changes that should be made to a merchant storefront cartridge.

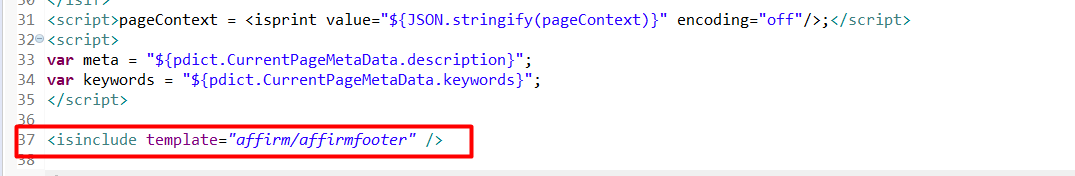
1. Find and open the below template: **storefront/cartridge/templates/default/components/header/htmlhead.isml**. Paste the following code after the “Google verification feature” as it shown on the screenshot below:

|  |
| --- |
| <isinclude template=*”affirm/affirmheader”* /> |



1. Find and open template – **storefront/cartridge/templates/default/components/footer/footer\_UI.isml**. Paste code after the last line:

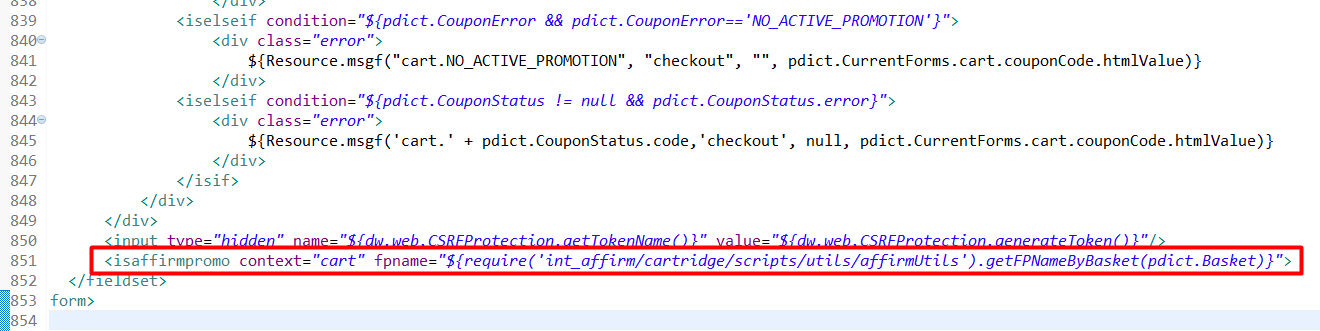
|  |
| --- |
| <isinclude template=*”affirm/affirmfooter”* /> |



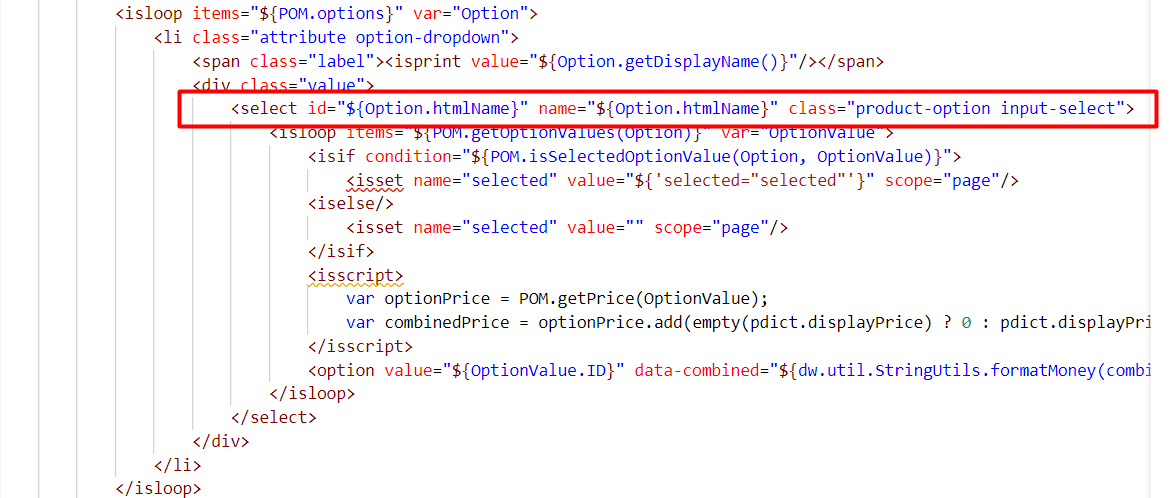
1. Find and open template – **storefront/default/checkout/cart/cart.isml.**

find the closing tag of div with class “cart-footer” (line 850). Paste code after the below:

|  |
| --- |
| <isaffirmpromo context=*”cart”* fpname=*”${require(‘int\_affirm/cartridge/scripts/utils/affirmUtils’).getFPNameByBasket(pdict.Basket)}”*> |



1. Find and open template: **storefront/cartridge/templates/default/product/components/options.isml.** Add “data-affirm-name” property to option <select> component



Replace with:

|  |
| --- |
| <select id="${Option.htmlName}" name="${Option.htmlName}" data-affirm-name="${Option.ID}" class="product-option input-select"> |

1. Find and open template: **storefront/cartridge/templates/default/product/productcontent.isml.** Paste code after “pricing component”:

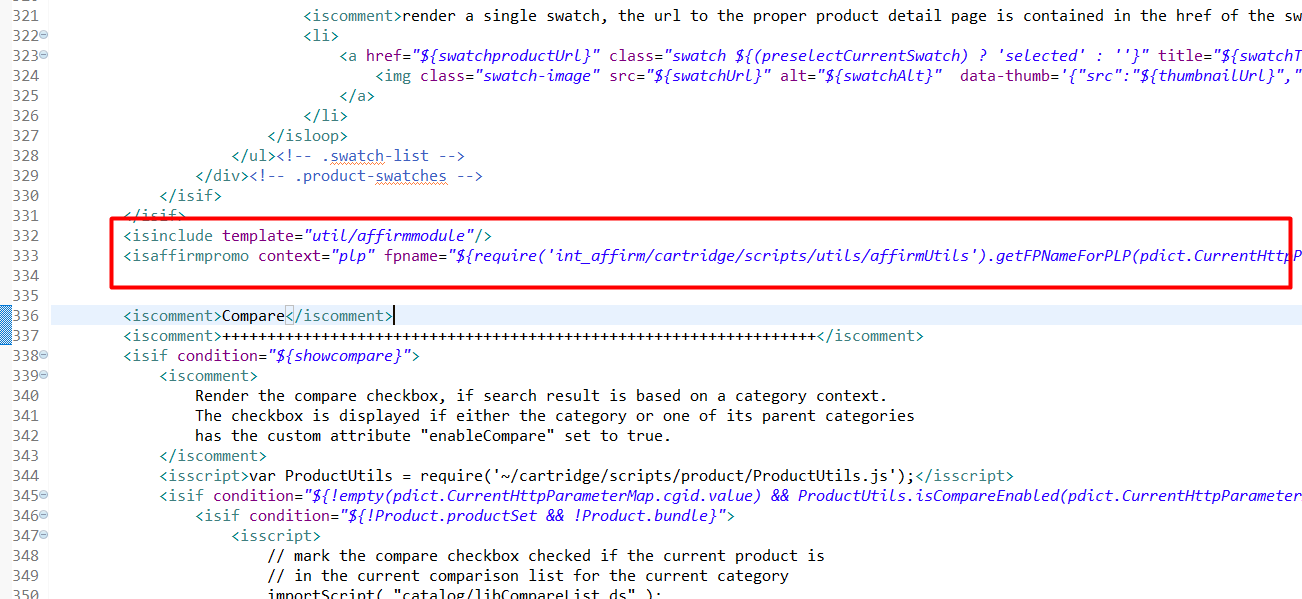
|  |
| --- |
| <isinclude template=*”util/affirmmodule”*/>  <isaffirmpromo context=*”pdp”* fpname=*”${require(‘int\_affirm/cartridge/scripts/utils/affirmUtils’).getFinancingProgramByProduct(pdict.Product, true)}”* /> |



1. Find and open template:

**storefront/cartridge/templates/default/product/producttile.isml**. Paste the code below after the “product swatches block”

|  |
| --- |
| <isinclude template=*”util/affirmmodule”*/>  <isaffirmpromo context=*”plp”* fpname=*”${require(‘int\_affirm/cartridge/scripts/utils/affirmUtils’).getFPNameForPLP(pdict.CurrentHttpParameterMap.cgid.value, Product)}”* price=*”${prices}”*/> |



1. Find and open template:

**storefront/cartridge/templates/default/product/producttopcontentPS.isml.** Find the following code (lines 118-120):



Paste code after:

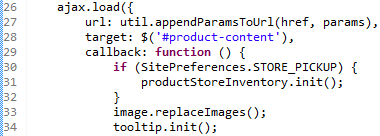
|  |
| --- |
| <isinclude template=*”util/affirmmodule”*/>  <isaffirmpromo context=*”pdp”* fpname=*”${require(‘int\_affirm/cartridge/scripts/utils/affirmUtils’).getFinancingProgramByProduct(pdict.Product, true)}”* /> |



1. Find and open template: **storefront/cartridge/templates/default/util/modules.isml.** Paste code after the last line:

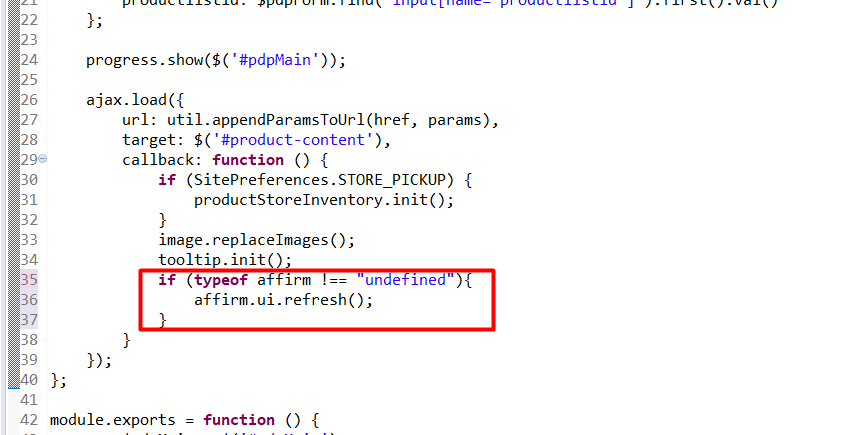
|  |
| --- |
| <isinclude template=*”util/affirmmodule”*/> |

1. Find and open client JS file – **storefront/cartridge/js/pages/product/variant.js**. Find the following code:

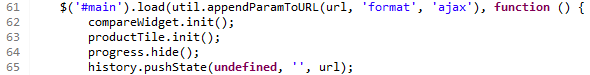


Paste code after:

|  |
| --- |
| **if** (**typeof** affirm !== “undefined”){  affirm.ui.refresh();  } |

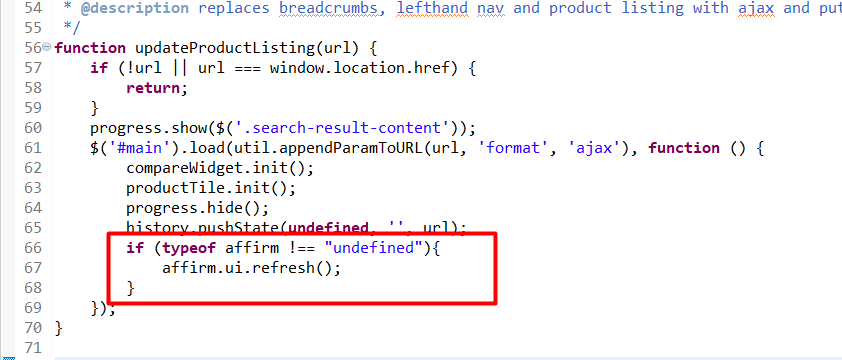


1. Find and open client JS file – **storefront/cartridge/js/pages/search.js**. Find the following code:

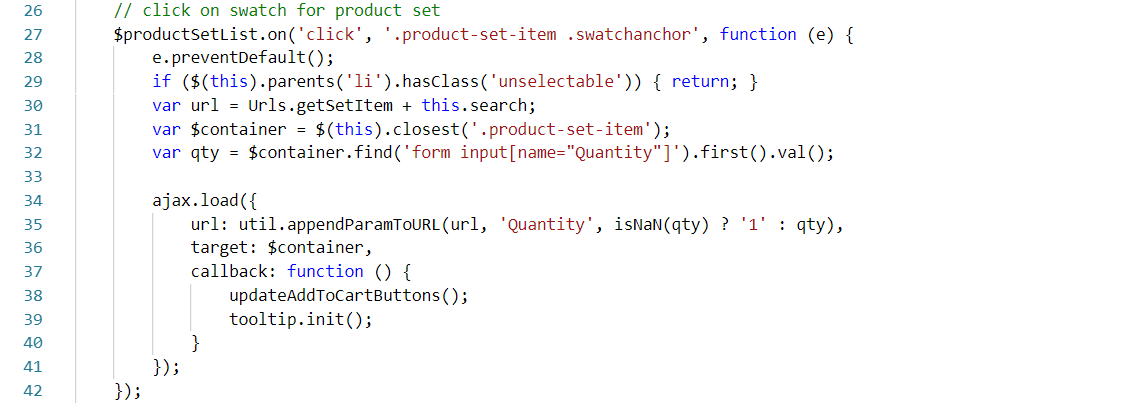


Paste code after:

|  |
| --- |
| **if** (**typeof** affirm !== “undefined”){  affirm.ui.refresh();  } |



1. Find and open client JS file – **storefront/cartridge/js/pages/product/productSet.js**. Find the following code:



Replace it with extended code:

|  |
| --- |
| var updateAffirmItems = function($container, oldSKU) {  var affirmItem = affirmItems.find(function (item) {  return item.sku === oldSKU;  });  affirmItem.sku = $container.find('input[name=pid]').first().val();  }  var updateBuyWithAffirmButton = function () {  if ($productSetList.find('.add-to-cart[disabled]').length > 0) {  $('#js-affirm-checkout-now').hide();  } else {  $('#js-affirm-checkout-now').show();  document.dispatchEvent(new CustomEvent('affirm-checkout-button-rendered'));  }  };  // click on swatch for product set  $productSetList.on('click', '.product-set-item .swatchanchor', function (e) {  e.preventDefault();  if ($(this).parents('li').hasClass('unselectable')) { return; }  var url = Urls.getSetItem + this.search;  var $container = $(this).closest('.product-set-item');  var qty = $container.find('form input[name="Quantity"]').first().val();  var oldSKU = $container.find('input[name=pid]').first().val();  ajax.load({  url: util.appendParamToURL(url, 'Quantity', isNaN(qty) ? '1' : qty),  target: $container,  callback: function () {  updateAddToCartButtons();  updateAffirmItems($container, oldSKU);  updateBuyWithAffirmButton();  tooltip.init();  }  });  }); |

**!!!IMPORTANT**

**To properly render “affirm-as-low” on PDP, PLP and Cart, at the end of handle functions that influence price on basket (ajax product price change, quantity based price change, pagination, search, add-on price change etc.) must call the affirm.ui.refresh function.**