Bluecore

Bluecore Integration for Demandware

Version 16.2.0

demandware LINK

Table of Contents

1. Summary

Whether you send one thousand or one million emails a day, each one should be as unique as the customer that opens it. Bluecore’s platform makes creating personalized email campaigns easy by reducing the number of touchpoints in a marketer’s traditional workflow down to one. Audience segmentation, template creation, and campaign deployment is reduced to a handful of clicks within a single, easy-to-use interface. Live analysis on terabytes of behavioral and product data make emails dynamically unique to every customer– no tech support required.

Bluecore’s customer experience platform is designed to simplify the process of ingesting terabytes of behavioral data and automatically acting on precise insights, driving engagement and conversion rates that defy industry standards. Bluecore’s customer experience platform enables the personalized digital communications based on both individual behavior and catalog data.

The Bluecore integration with Demandware is a cartridge that enables customers to fully implement Bluecore technology on their storefront, which is easily installed on the site via Demandware's cartridge system. The cartridge automatically deploys Bluecore tracking across the site. This will collect vast amounts of customer behavioral data, in addition to capturing live data about the storefront’s product catalog. The cartridge enables use of the entire Bluecore product suite, in just a few easy steps.

2. Component Overview

2.1 Functional Overview

On-site Activity Tracking

The Bluecore cartridge will track the following on-site user activities:

- On-site search
- View category page
- View product
- Add product(s) to cart
- Purchase
- Opt-in to email marketing

If necessary, additional user activities can also be uploaded outside of the Demandware cartridge. Contact your Bluecore Customer Success team for more details.

On-site Product Catalog Tracking

The Bluecore cartridge will track your on-site product catalog in real-time, enabling use cases such as price change triggers. The list of product attributes tracked will depend on the cartridge configuration chosen with your Bluecore Customer Success team.

Customer Attribute Tracking

The Bluecore cartridge will track the following customer attributes:

- Email Address: A customer's unique email address.
- Sign-up Date: The date that a customer signed up on your site.
- Lifetime Value: The total value of a customer's transactions.

If necessary, additional customer attributes can also be uploaded outside of the Demandware cartridge. Contact your Bluecore Customer Success team for more details.

2.2 Use Cases

Abandonment Messaging

Remarket to users that have taken an on-site action and then abandoned. Abandonment actions include, but may not be limited to: on-site search, category browse, product browse, and cart.

Behavioral Lifecycle Messaging

Remarket to users that have made purchases or have taken other custom on-site actions, if applicable.

Product Notification Messaging

Remarket to users that have interacted with products, where attributes of the product (e.g., price) have subsequently changed.

Product Recommendations

Capture the product catalog to power product recommendations in remarketing messages.

Customer Analytics

Report on customer analytics based on customers' holistic interactions with the website and remarketing messages.

Audience Segmentation

Segment customers into different marketing groups based on customers' holistic interactions with the website and remarketing messages.

Predictive Marketing

Build predictive models based on the activities that the customer has taken on-site.

2.3 Limitations, Constraints

The Bluecore cartridge provides all implementation necessary to support the full Bluecore solution set. The actual use cases and solutions available to you are dependent on your Bluecore contract.

2.4 Compatibility

The Bluecore cartridge has been developed and tested against Commerce Cloud Digital 16.9.

2.5 Privacy, Payment

The only customer profile information collected by the cartridge is email address. No other personally identifiable information is collected. No credit card data is parsed or collected.

3. Implementation Guide

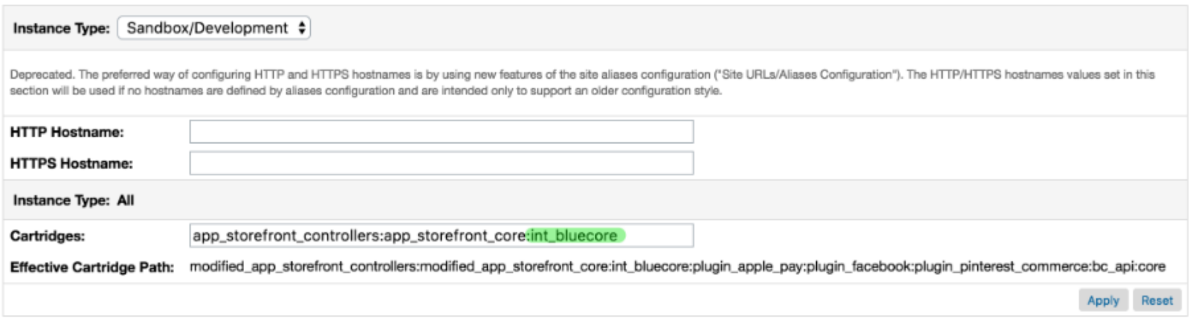
3.1 Setup

3.1.1 Uploading the cartridge via Demandware Studio

1. Open Demandware Studio.
2. Click File > Import > General > Existing Files Into Workspace.
3. Browse to the int_bluecore directory.
4. Click Finish.
5. Click OK when prompted to link the cartridge to the sandbox.

3.1.2 Registering the cartridge in Demandware

1. Navigate to Business Manager: Administration > Sites > Manage Sites > site.
2. Click the Settings tab.
3. In the Cartridges field, add: int_bluecore to the end of the list (separate cartridges by colons).



Instance Type: Sandbox/Development

Deprecated. The preferred way of configuring HTTP and HTTPS hostnames is by using new features of the site aliases configuration ("Site URLs/Aliases Configuration"). The HTTP/HTTPS hostnames values set in this section will be used if no hostnames are defined by aliases configuration and are intended only to support an older configuration style.

HTTP Hostname:	
HTTPS Hostname:	

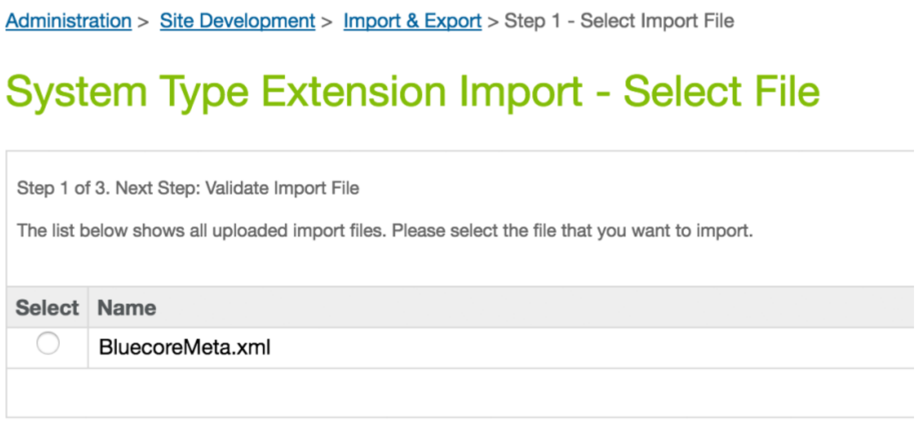
Instance Type: All

Cartridges:	app_storefront_controllers:app_storefront_core:int_bluecore
Effective Cartridge Path:	modified_app_storefront_controllers:modified_app_storefront_core:int_bluecore:plugin_apple_pay:plugin_facebook:plugin_pinterest_commerce:bc_api:core

4. Click Apply.

3.1.3 Import Custom Site Preferences into Business Manager

1. Log in to the Demandware Business Manager.
2. Navigate to **Administration > Site Development > Import & Export**.
3. Click Upload.
4. Browse for the BluecoreMeta.xml file contained in the metadata folder.



Administration > Site Development > Import & Export > Step 1 - Select Import File

System Type Extension Import - Select File

Step 1 of 3. Next Step: Validate Import File

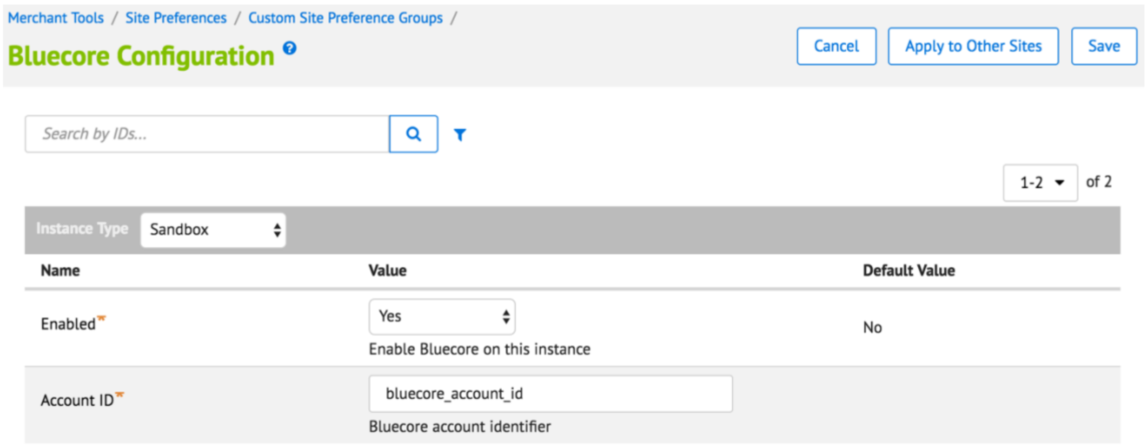
The list below shows all uploaded import files. Please select the file that you want to import.

Select	Name
○	BluecoreMeta.xml

5. Click **Upload**.
6. Click the **<< Back** button at the bottom of the page.
7. Click **Import** in the Meta Data section.
8. Select the **BluecoreMeta.xml** file.
9. Click **Next >>**.
10. Click **Refresh**.
11. Click **Import**.

3.1.4 Configure Custom Preferences using the Business Manager

1. Log into the Demandware Business Manager.
2. Select the correct site from the tabs across the top of the page. Note: Each site on which Bluecore will exist will display.
3. Click Merchant Tools > Site Preferences > Custom Preferences.



Instance Type: Sandbox

Name	Value	Default Value
Enabled*	Yes Enable Bluecore on this instance	No
Account ID*	bluecore_account_id Bluecore account identifier	

4. Fill in the Site Preferences as directed. Parameter values will be provided by Bluecore during integration. Your Account ID will be provided by your Customer Success team if you do not already have it.
5. Click **Save**.

3.2 Custom Code

Bluecore needs custom code written in both controllers (or pipelines) and templates. All custom code ultimately does the same thing; inserts a script tag into the response HTML. The script tag interacts with an external script the front end pulls in.

We'd prefer to add code to controllers/pipelines, as they are less likely to change. However, because some controllers/pipelines return the entire HTML (including the <html> tag), not just a section of it, we need to insert our script tag into the template rather than in the respective controller/pipeline to avoid having the script tag fall outside the html tag. Files like pt_productsearchresult_VARS.isml are designed for exactly this.

3.2.1 Templates

templates/default/components/header/header.isml

Add the following line:

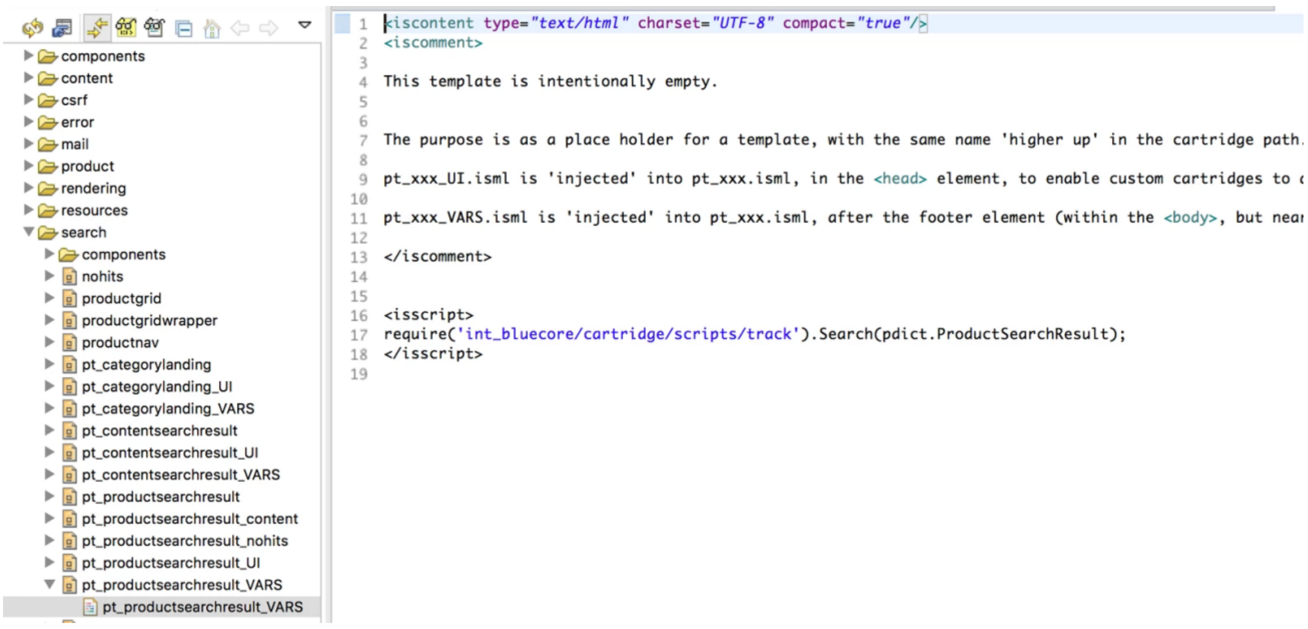
```
<isinclude template="bluecore/snippetinclude"/>
```

This included Bluecore template tells the site to pull in the Bluecore Javascript snippet to handle and relay the script payloads we'll be sending to the front-end. It's included on templates/default/components/header/header.isml as we need our script pulled in on all pages.

templates/search/pt_productsearchresult_VARS/pt_productsearchresult_VARS.isml

Add an `isscript` tag that calls our `track.js`'s Search function.

```
<isscript>
require('int_bluecore/cartridge/scripts/track').Search(pdict.ProductSearchResult);
</isscript>
```



```
<iscontent type="text/html" charset="UTF-8" compact="true"/>
<iscomment>

This template is intentionally empty.


The purpose is as a place holder for a template, with the same name 'higher up' in the cartridge path.

pt_xxx_UI.isml is 'injected' into pt_xxx.isml, in the <head> element, to enable custom cartridges to 

pt_xxx_VARS.isml is 'injected' into pt_xxx.isml, after the footer element (within the <body>, but near

</iscomment>


<isscript>
require('int_bluecore/cartridge/scripts/track').Search(pdict.ProductSearchResult);
</isscript>
```

If your implementation does not use this file, please copy this code into a template that gets pulled into every search result page. If you have any questions regarding this, see section 4.3 for support.

templates/checkout/cart/pt_cart_VARS/pt_cart_VARS.isml

Add an `isscript` tag that calls our `track.js`'s ViewCart function.

```
<isscript>
require('int_bluecore/cartridge/scripts/track').ViewCart(pdict.cart);
</isscript>
```

templates/account/wishlist/pt_wishlist_VARS/pt_wishlist_VARS.isml

Add an `isscript` tag that calls our `track.js`'s ViewWishlist function.

```
<isscript>
require('int_bluecore/cartridge/scripts/track').ViewWishlist();
</isscript>
```

templates/checkout/pt_orderconfirmation_VARS/pt_orderconfirmation_VARS.isml

Add an **isscript** tag that calls our **track.js**'s Checkout function.

```
<isscript>
require('int_bluecore/cartridge/scripts/track').Checkout(pdict.Order);
</isscript>
```

3.2.2 Controllers

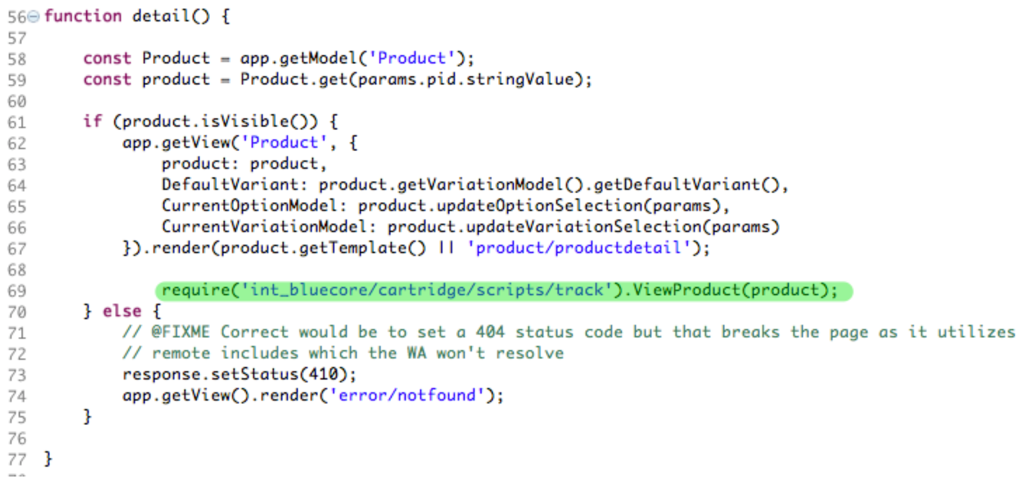
If your site uses pipelines, skip this section and continue to 3.2.3 Pipelines

controllers/Product.js (view product)

Add the following line to the detail function, after the template is rendered:

```
require('int_bluecore/cartridge/scripts/track').ViewProduct(product);
```

Example:



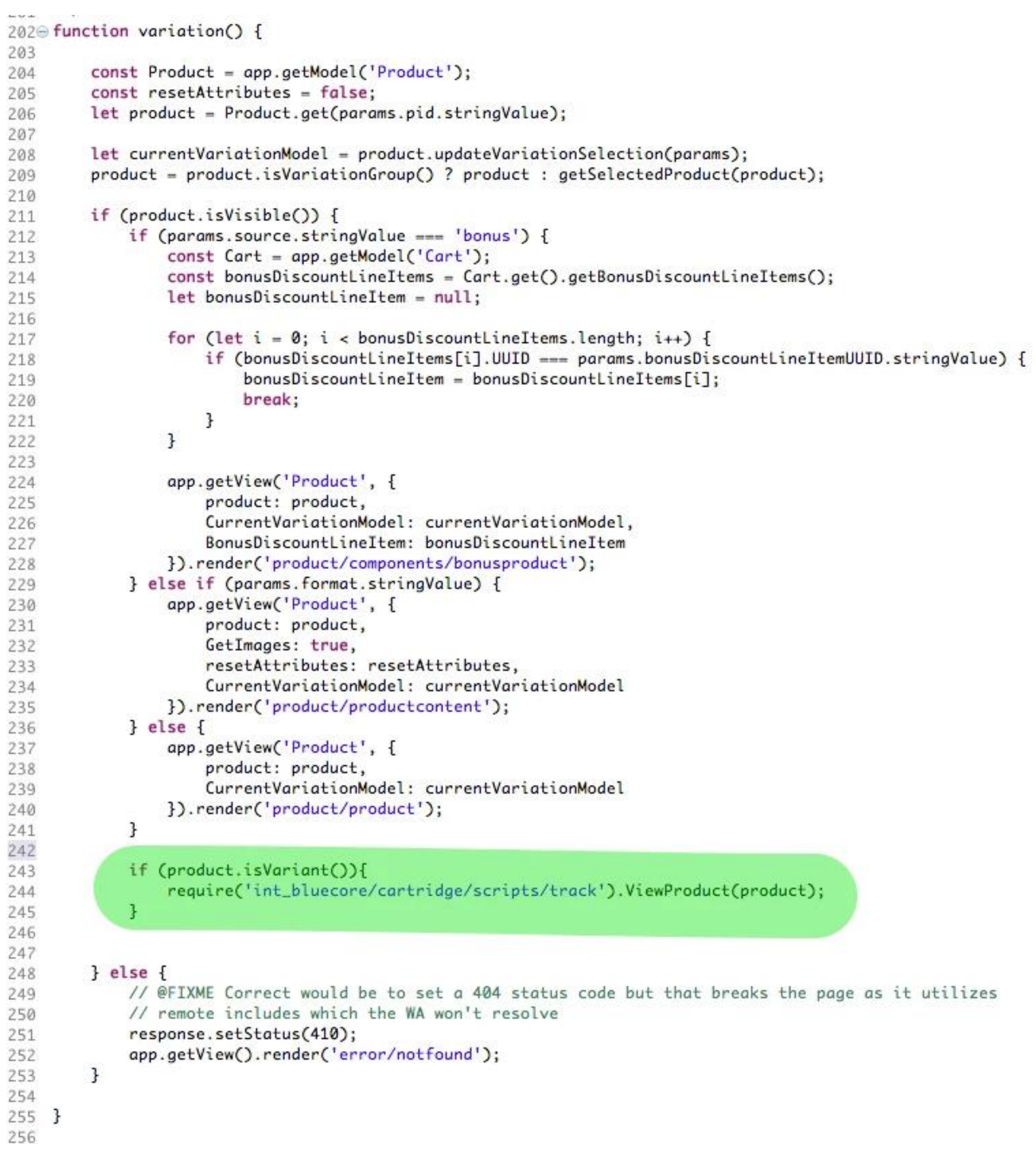
```
function detail() {

    const Product = app.getModel('Product');
    const product = Product.get(params.pid.stringValue);

    if (product.isVisible()) {
        app.getView('Product', {
            product: product,
            DefaultVariant: product.getVariationModel().getDefaultVariant(),
            CurrentOptionModel: product.updateOptionSelection(params),
            CurrentVariationModel: product.updateVariationSelection(params)
        }).render(product.getTemplate() || 'product/productdetail');

        require('int_bluecore/cartridge/scripts/track').ViewProduct(product);
    } else {
        // @FIXME Correct would be to set a 404 status code but that breaks the page as it utilizes
        // remote includes which the WA won't resolve
        response.setStatus(410);
        app.getView().render('error/notfound');
    }

}
```

This appends a script tag to the HTML response when a user lands on a Product Detail Page. This function returns just a section of html, so it's OK to append our script here.

Add the same line, with the condition that the product be a variant, within the variation function after the template is rendered:

```
if (product.isVariant()){
    require('int_bluecore/cartridge/scripts/track').ViewProduct(product);
}
```

Example:



```
function variation() {

    const Product = app.getModel('Product');
    const resetAttributes = false;
    let product = Product.get(params.pid.stringValue);

    let currentVariationModel = product.updateVariationSelection(params);
    product = product.isVariationGroup() ? product : getSelectedProduct(product);

    if (product.isVisible()) {
        if (params.source.stringValue === 'bonus') {
            const Cart = app.getModel('Cart');
            const bonusDiscountLineItems = Cart.get().getBonusDiscountLineItems();
            let bonusDiscountLineItem = null;

            for (let i = 0; i < bonusDiscountLineItems.length; i++) {
                if (bonusDiscountLineItems[i].UUID === params.bonusDiscountLineItemUUID.stringValue) {
                    bonusDiscountLineItem = bonusDiscountLineItems[i];
                    break;
                }
            }

            app.getView('Product', {
                product: product,
                CurrentVariationModel: currentVariationModel,
                BonusDiscountLineItem: bonusDiscountLineItem
            }).render('product/components/bonusproduct');
        } else if (params.format.stringValue) {
            app.getView('Product', {
                product: product,
                GetImages: true,
                resetAttributes: resetAttributes,
                CurrentVariationModel: currentVariationModel
            }).render('product/productcontent');
        } else {
            app.getView('Product', {
                product: product,
                CurrentVariationModel: currentVariationModel
            }).render('product/product');
        }

        if (product.isVariant()){
            require('int_bluecore/cartridge/scripts/track').ViewProduct(product);
        }


    } else {
        // @FIXME Correct would be to set a 404 status code but that breaks the page as it utilizes
        // remote includes which the WA won't resolve
        response.setStatus(410);
        app.getView().render('error/notfound');
    }

}
```

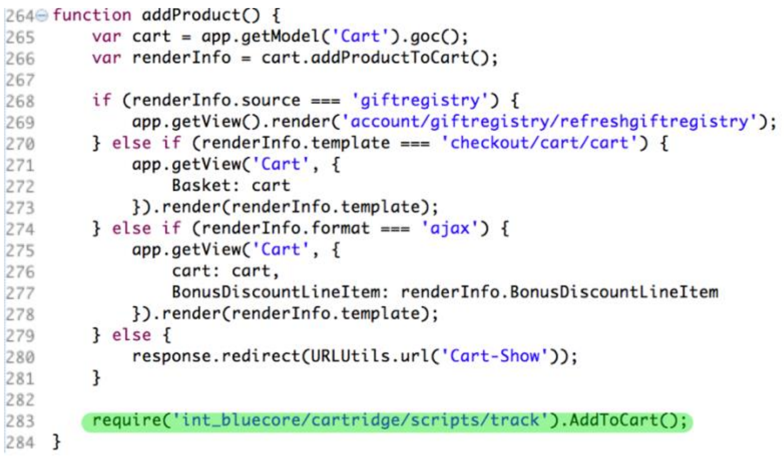
This allows us to register viewing behavior on quick views. We only need to know when the user sees a full variation (i.e., if they've selected color AND size), which is the reason for the "if" statement.

controllers/Cart.js (add to cart)

Add the following line to the end of the addProduct function:

```
require('int_bluecore/cartridge/scripts/track').AddToCart();
```

Example:



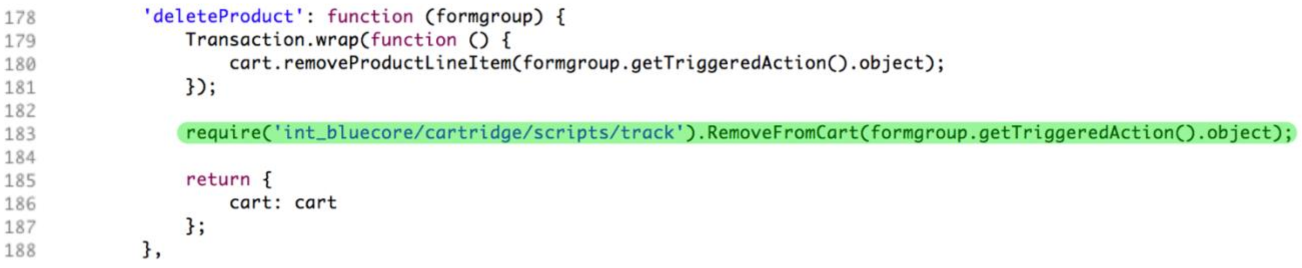
```
function addProduct() {
    var cart = app.getModel('Cart').goc();
    var renderInfo = cart.addProductToCart();

    if (renderInfo.source === 'giftregistry') {
        app.getView().render('account/giftregistry/refreshgiftregistry');
    } else if (renderInfo.template === 'checkout/cart/cart') {
        app.getView('Cart', {
            Basket: cart
        }).render(renderInfo.template);
    } else if (renderInfo.format === 'ajax') {
        app.getView('Cart', {
            cart: cart,
            BonusDiscountLineItem: renderInfo.BonusDiscountLineItem
        }).render(renderInfo.template);
    } else {
        response.redirect(URLUtils.url('Cart-Show'));
    }

    require('int_bluecore/cartridge/scripts/track').AddToCart();
}
```

controllers/Cart.js (remove from cart)

Add the following line to the submitForm function under the deleteProduct handler:

```
require('int_bluecore/cartridge/scripts/track').RemoveFromCart(formgroup.getTriggeredAction().object);
```

Example:



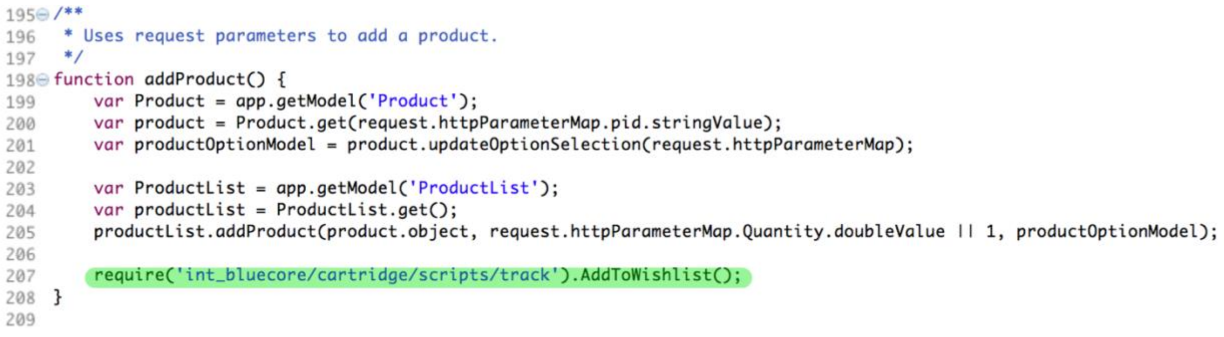
```
        'deleteProduct': function (formgroup) {
            Transaction.wrap(function () {
                cart.removeProductLineItem(formgroup.getTriggeredAction().object);
            });

            require('int_bluecore/cartridge/scripts/track').RemoveFromCart(formgroup.getTriggeredAction().object);

            return {
                cart: cart
            };
        },
```

controllers/Wishlist.js (add to wishlist)

Add the following line to the addProduct function:

```
require('int_bluecore/cartridge/scripts/track').AddToWishlist();
```

Example:



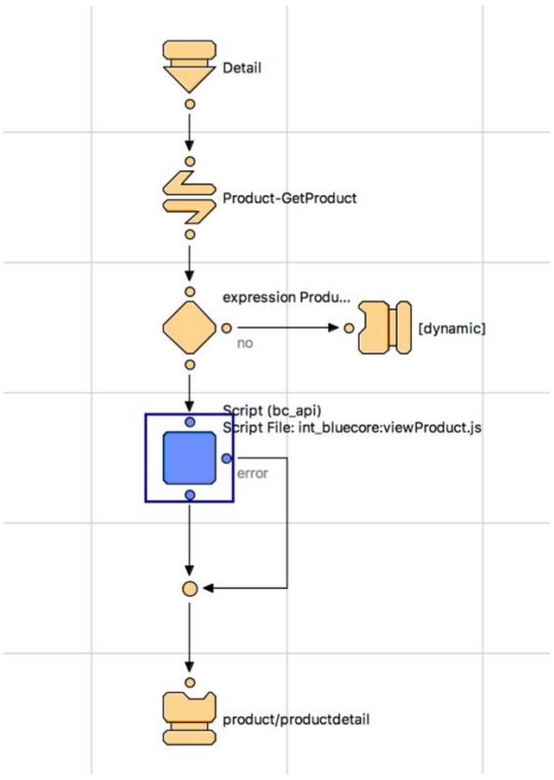
```
/**
 * Uses request parameters to add a product.
 */
function addProduct() {
    var Product = app.getModel('Product');
    var product = Product.get(request.httpParameterMap.pid.stringValue);
    var productOptionModel = product.updateOptionSelection(request.httpParameterMap);

    var ProductList = app.getModel('ProductList');
    var productList = ProductList.get();
    productList.addProduct(product.object, request.httpParameterMap.Quantity.doubleValue || 1, productOptionModel);

    require('int_bluecore/cartridge/scripts/track').AddToWishlist();
}
```

3.2.3 Pipelines

Use the following steps if your site uses pipelines.

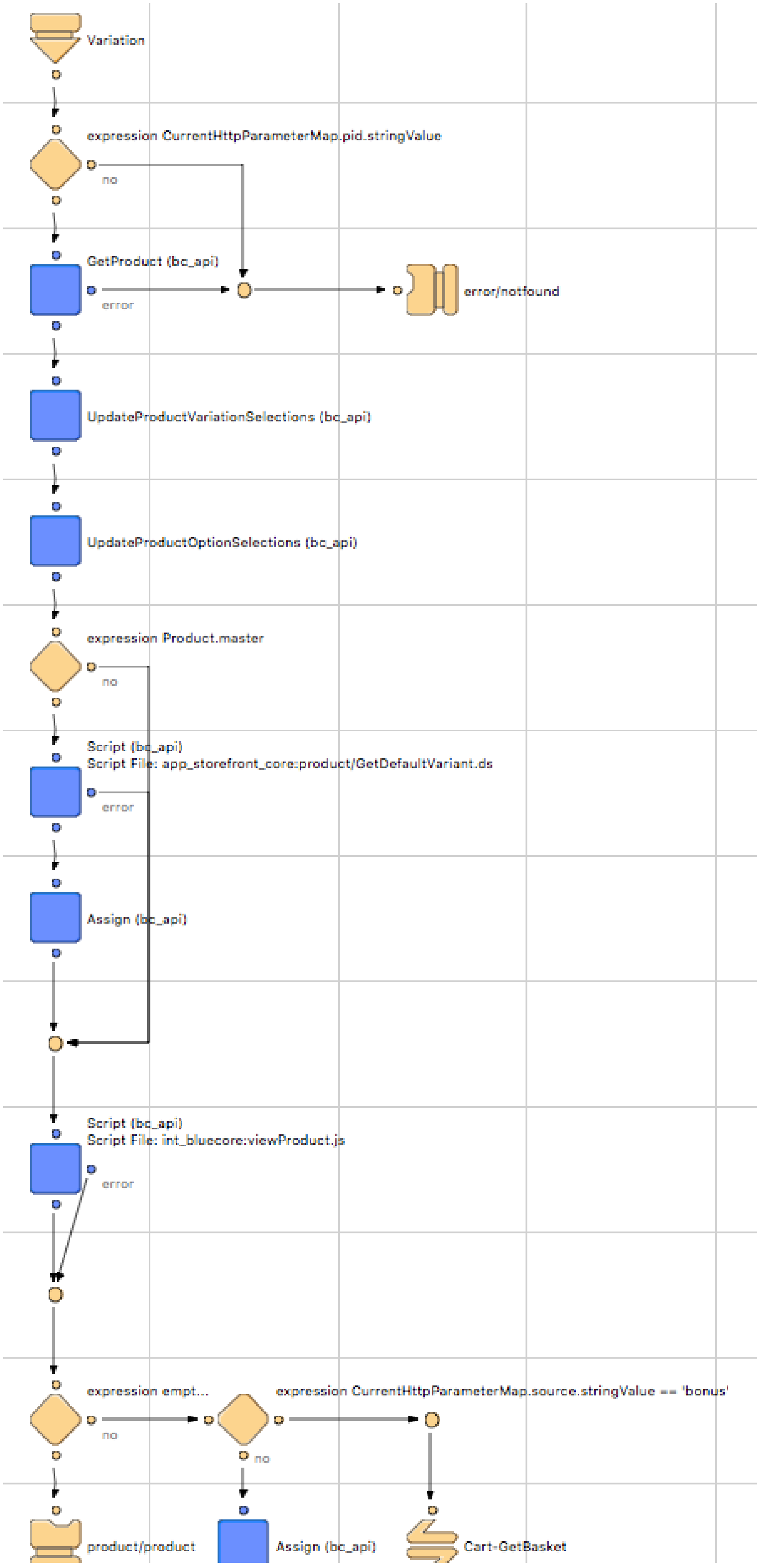
Product-Detail Pipeline

Add the script int_bluecore/cartridge/scripts/viewProduct.js to the Product-Detail pipeline between the Product-GetProduct and the template interaction node.



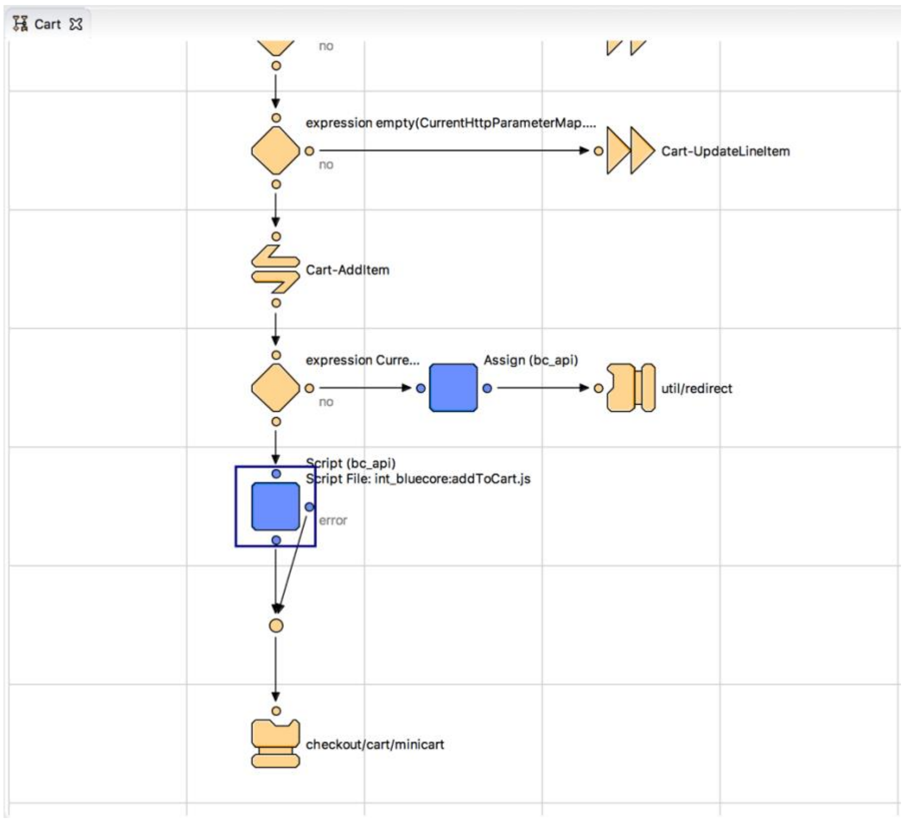
Product-Variation Pipeline

Add the script int_bluecore/cartridge/scripts/viewProduct.js to the Product-Variation pipeline after fetching the product, and before the template interaction nodes.


Variation
expression CurrentHttpParameterMap.pid.stringValue
no
GetProduct (bc_api)
error
error/notfound
UpdateProductVariationSelections (bc_api)
UpdateProductOptionSelections (bc_api)
expression Product.master
no
Script (bc_api)
Script File: app_storefront_core:product/GetDefaultVariant.ds
error
Assign (bc_api)
Script (bc_api)
Script File: int_bluecore:viewProduct.js
error
expression empt...
no
expression CurrentHttpParameterMap.source.stringValue == 'bonus'
no
product/product
Assign (bc_api)
Cart-GetBasket

Cart-AddProduct Pipeline

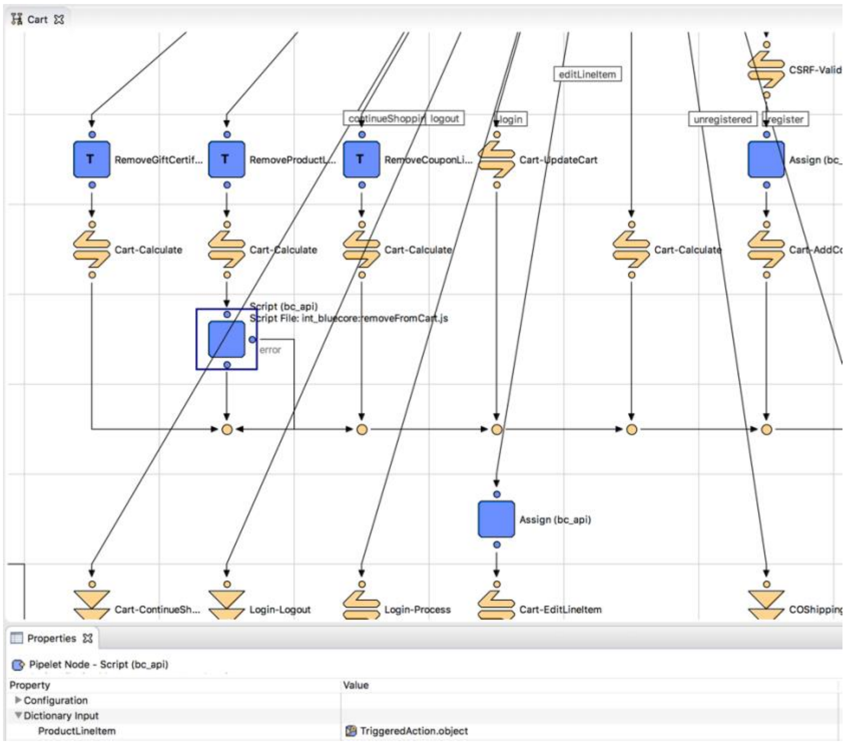
Add the script int_bluecore/cartridge/scripts/addToCart.js to the Cart-AddProduct pipeline before the template interaction node.



Cart-Show Pipeline (removeFromCart)

Add the script int_bluecore/cartridge/scripts/removeFromCart.js to the pipeline under submitForm > deleteProduct.

Set ProductLineItem to TriggeredAction.object for the input dictionary.



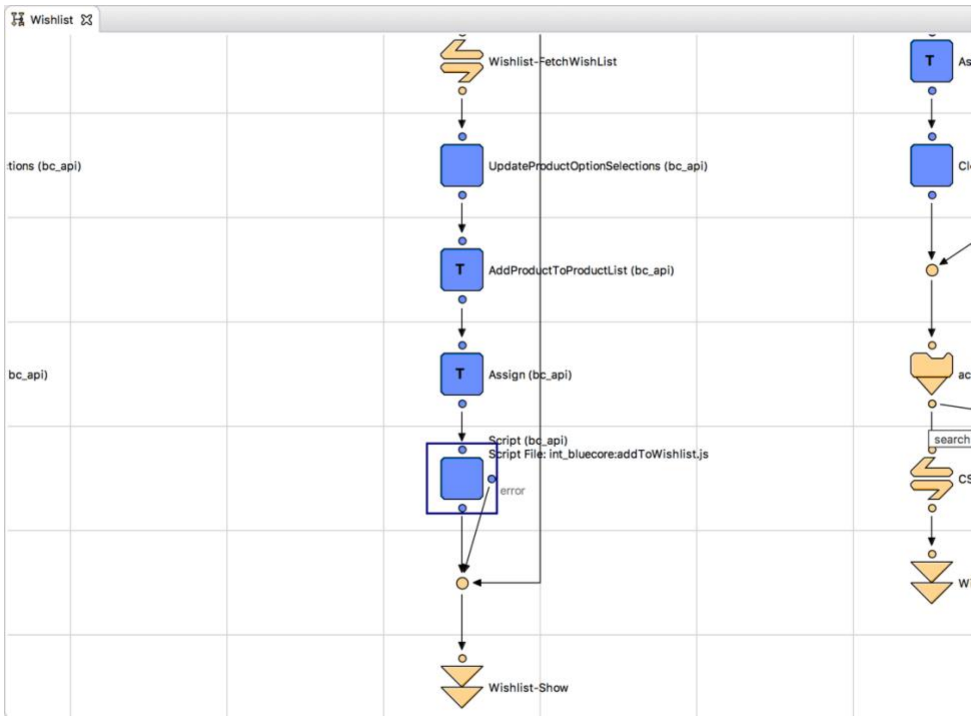
Properties

Pipelet Node - Script (bc_api)

Property	Value
▸ Configuration	
▾ Dictionary Input	
ProductLineItem	TriggeredAction.object

Wishlist-Add Pipeline

Add the script int_bluecore/cartridge/scripts/addToWishlist.js to the Wishlist-Add pipeline before the Wishlist-Show jump node.


Wishlist
Wishlist-FetchWishList
UpdateProductOptionSelections (bc_api)
AddProductToProductList (bc_api)
Assign (bc_api)
Script (bc_api)
Script File: int_bluecore:addToWishlist.js
error
Wishlist-Show
search

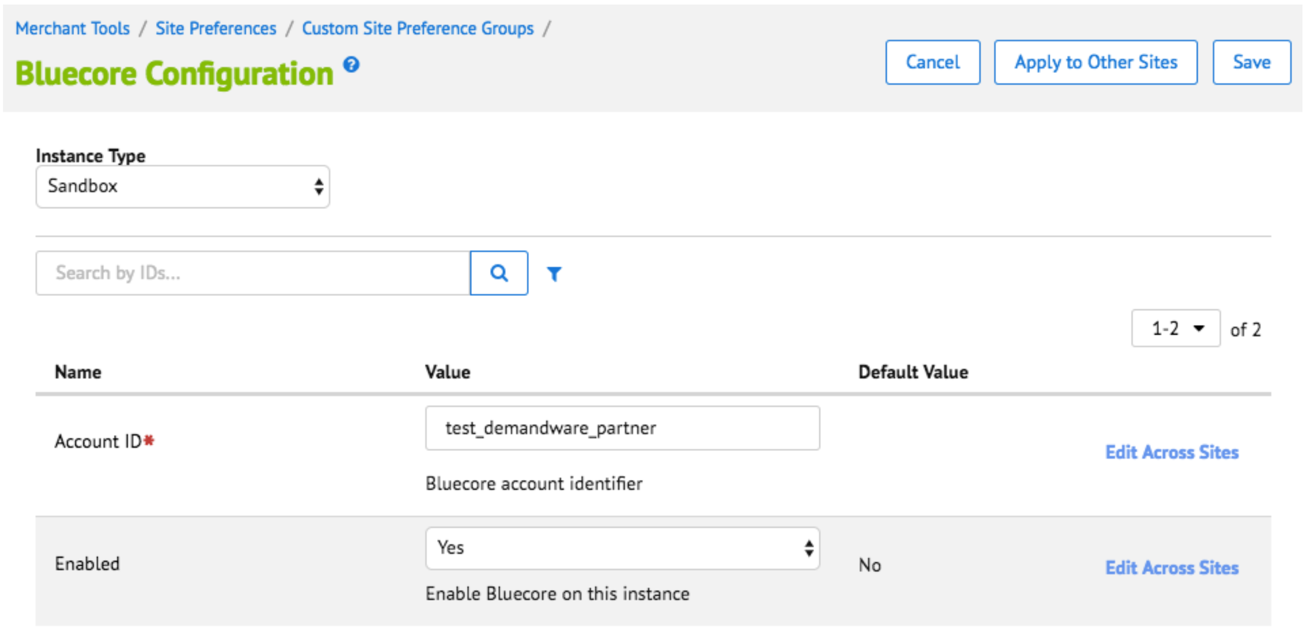
3.3 External Interfaces

The Bluecore cartridge automatically handles all communication required between your storefront and Bluecore via asynchronous Javascript added to the site. The Javascript is asynchronous and only performs tracking functions; it does not impact the viewable page content in any way. The Javascript has no impact on the load times of your site. Optionally, you may send additional customer and product information to Bluecore via direct file upload into the Bluecore UI, or automated file feed to a Bluecore SFTP location. Contact your Customer Success team for more details.

3.4 Testing

3.4.1 Testing Setup

For the purposes of testing, please fill out the custom preferences in section 3.1.4 the following way:

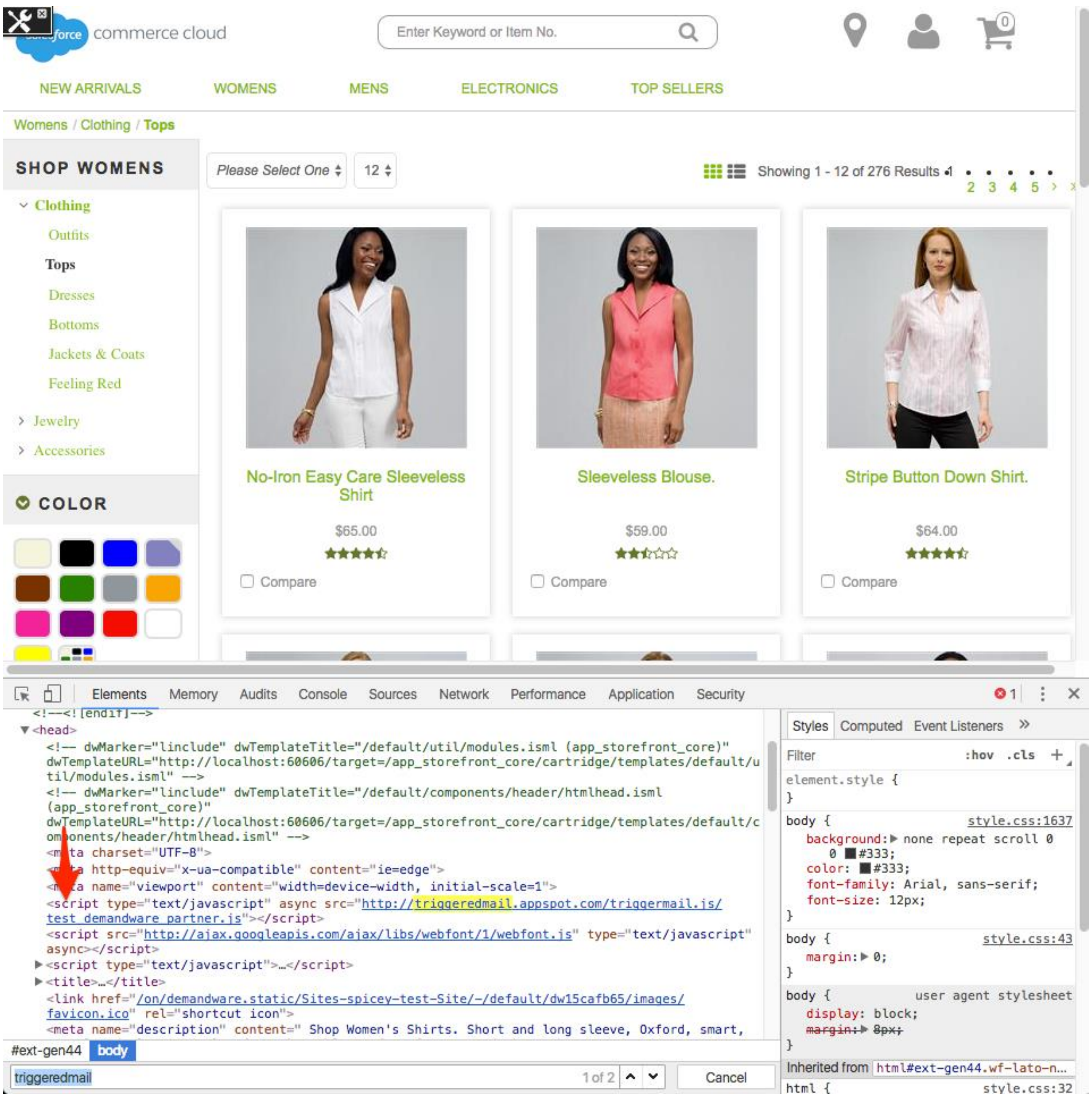


Instance Type: Sandbox

Name	Value	Default Value
Account ID*	test_demandware_partner Bluecore account identifier	
Enabled	Yes Enable Bluecore on this instance	No

If your account team has provided you with a more specific Account ID, please use that instead of "test_demandware_partner" (case sensitive).

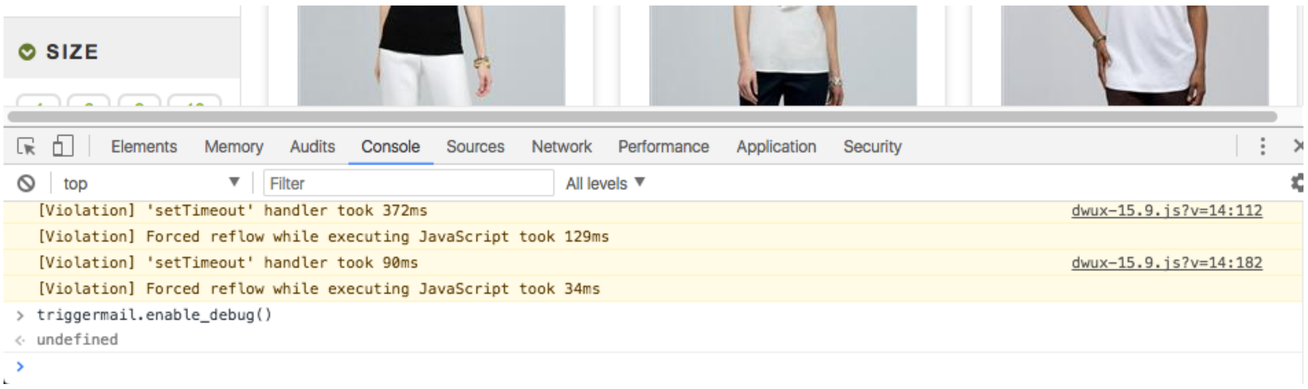
After enabling, you should start seeing our Javascript snippet being pulled in on the front end:


```
<!--<![endif]-->
▼<head>
  <!-- dwMarker="linclude" dwTemplateTitle="/default/util/modules.isml (app_storefront_core)"
  dwTemplateURL="http://localhost:60606/target=/app_storefront_core/cartridge/templates/default/u
  til/modules.isml" -->
  <!-- dwMarker="linclude" dwTemplateTitle="/default/components/header/htmlhead.isml
  (app_storefront_core)"
  dwTemplateURL="http://localhost:60606/target=/app_storefront_core/cartridge/templates/default/c
  omponents/header/htmlhead.isml" -->
  <meta charset="UTF-8">
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <script type="text/javascript" async src="http://triggeredmail.appspot.com/triggermail.js/
  test_demandware_partner.js"></script>
  <script src="http://ajax.googleapis.com/ajax/libs/webfont/1/webfont.js" type="text/javascript"
  async></script>
 ▶<script type="text/javascript">…</script>
 ▶<title>…</title>
  <link href="/on/demandware.static/Sites-spicey-test-Site/-/default/dw15cafb65/images/
  favicon.ico" rel="shortcut icon">
  <meta name="description" content=" Shop Women's Shirts. Short and long sleeve, Oxford, smart,
```

```
element.style {
}
body {                                   style.css:1637
  background:▶ none repeat scroll 0
    0 ■#333;
  color: ■#333;
  font-family: Arial, sans-serif;
  font-size: 12px;
}
body {                                   style.css:43
  margin:▶ 0;
}
body {                     user agent stylesheet
  display: block;
  ~~margin:▶ 8px;~~
}
Inherited from html#ext-gen44.wf-lato-n…
html {                                   style.css:32
```

If the script is not being pulled in properly, then there has been an issue with the steps in 3.1.x, or with the first template step in section 3.2.1 (adding to templates/default/components/header/header.isml).

With the external script pulled in, the "triggermail" object gets created and placed on the window. There is a method on triggermail we can now call to help us test: "triggermail.enable_debug()":

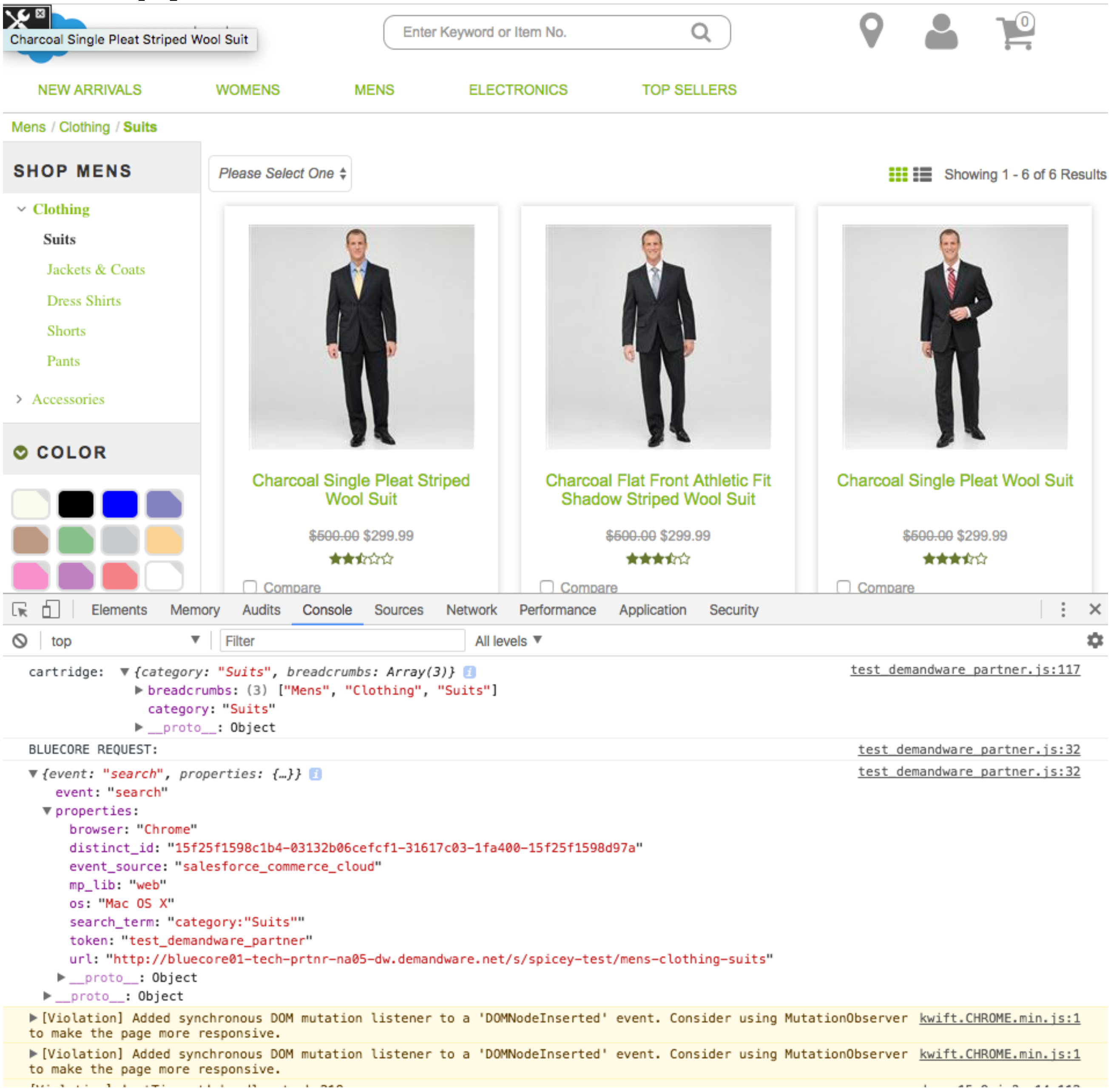

```
[Violation] 'setTimeout' handler took 372ms                            dwux-15.9.js?v=14:112
[Violation] Forced reflow while executing JavaScript took 129ms
[Violation] 'setTimeout' handler took 90ms                             dwux-15.9.js?v=14:182
[Violation] Forced reflow while executing JavaScript took 34ms
> triggermail.enable_debug()
< undefined
>
```

After calling the method, we'll start seeing the payloads the cartridge should be sending to the frontend, as well as the subsequent payloads sent to Bluecore via the external script.

Now we can start testing the collection of individual events.

3.4.2 Event Testing

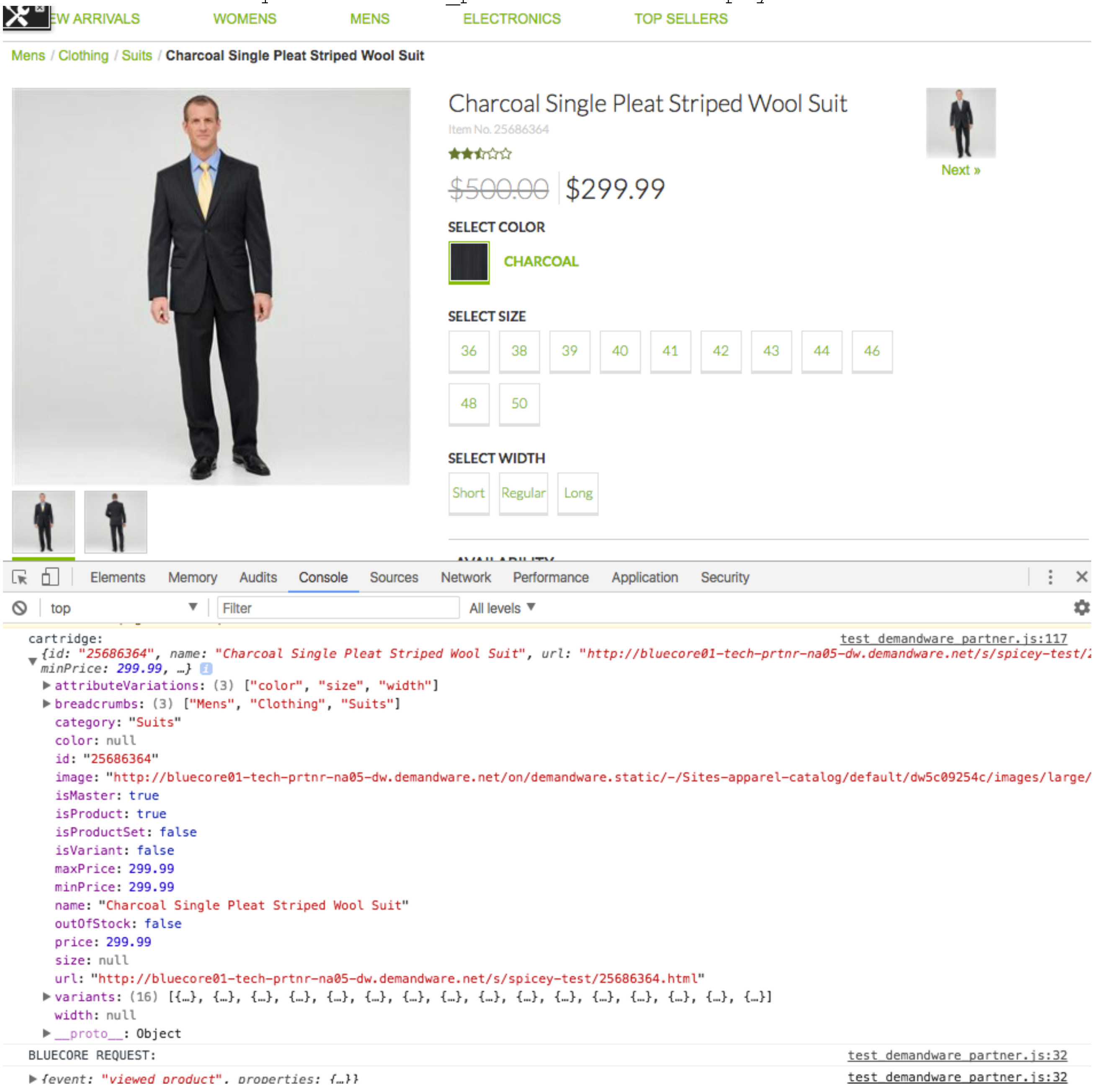
Searching:
Upon landing on a category or search page, we should see the data handed to the frontend via the cartridge, as well as the Bluecore "search" payload:



```
cartridge:  ▼{category: "Suits", breadcrumbs: Array(3)}          test_demandware_partner.js:117
              ▶breadcrumbs: (3) ["Mens", "Clothing", "Suits"]
               category: "Suits"
              ▶__proto__: Object
BLUECORE REQUEST:                                                 test_demandware_partner.js:32
▼{event: "search", properties: {…}}                               test_demandware_partner.js:32
   event: "search"
  ▼properties:
     browser: "Chrome"
     distinct_id: "15f25f1598c1b4-03132b06cefcf1-31617c03-1fa400-15f25f1598d97a"
     event_source: "salesforce_commerce_cloud"
     mp_lib: "web"
     os: "Mac OS X"
     search_term: "category:"Suits""
     token: "test_demandware_partner"
     url: "http://bluecore01-tech-prtnr-na05-dw.demandware.net/s/spicey-test/mens-clothing-suits"
    ▶__proto__: Object
  ▶__proto__: Object
▶[Violation] Added synchronous DOM mutation listener to a 'DOMNodeInserted' event. Consider using MutationObserver  kwift.CHROME.min.js:1
to make the page more responsive.
▶[Violation] Added synchronous DOM mutation listener to a 'DOMNodeInserted' event. Consider using MutationObserver  kwift.CHROME.min.js:1
to make the page more responsive.
```

If the cartridge event is not seen, there is likely an issue including the <isscript> tag in templates/search/pt_productsearchresult_VARS/pt_productsearchresult_VARS.isml

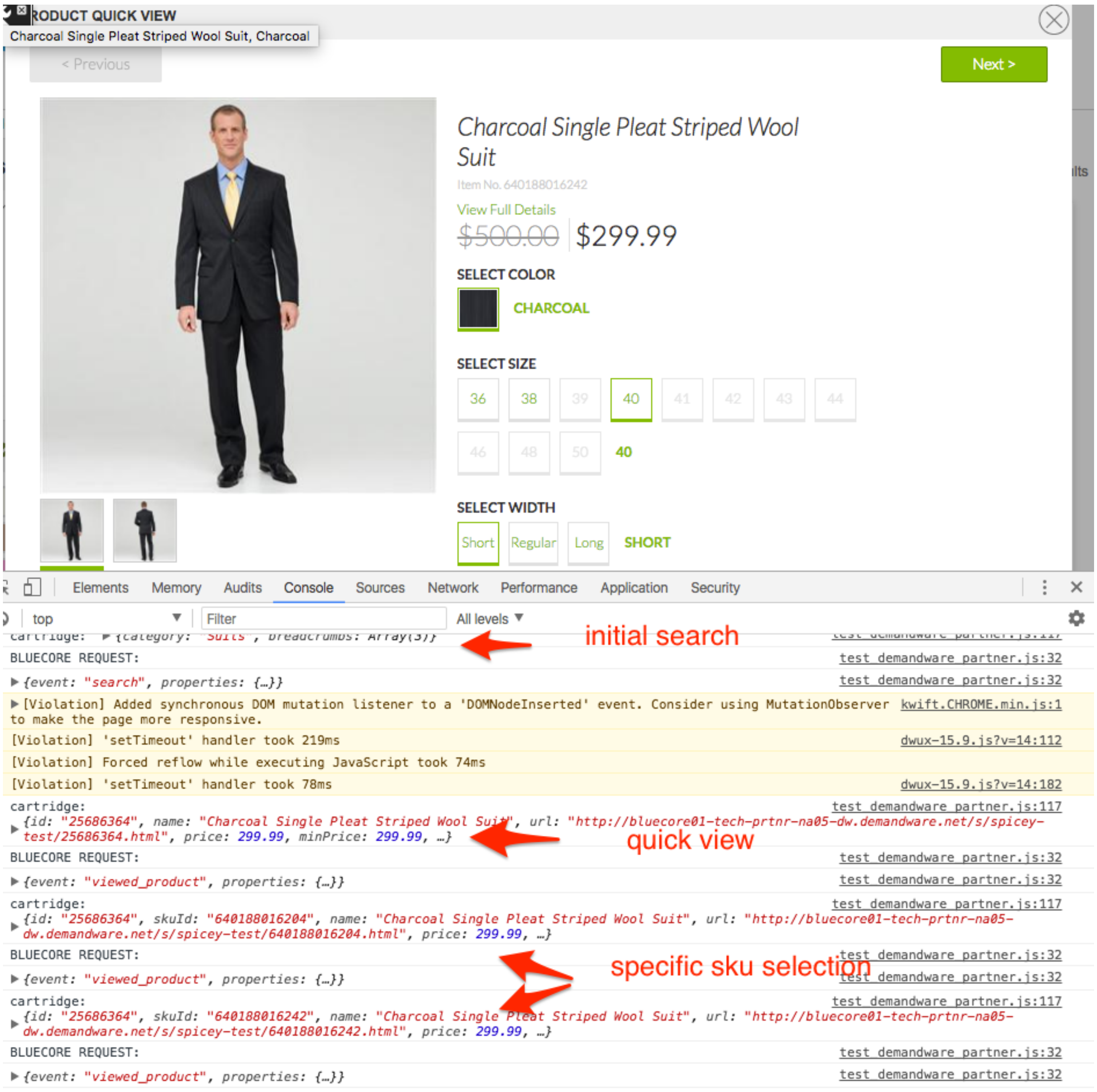
PDP Page Loads:
Upon landing on a PDP, we should see a larger cartridge payload, as well as its subsequent "viewed_product" Bluecore payload:



```
cartridge:                                                                           test_demandware_partner.js:117
▼{id: "25686364", name: "Charcoal Single Pleat Striped Wool Suit", url: "http://bluecore01-tech-prtnr-na05-dw.demandware.net/s/spicey-test/2
 minPrice: 299.99, …}
  ▶attributeVariations: (3) ["color", "size", "width"]
  ▶breadcrumbs: (3) ["Mens", "Clothing", "Suits"]
   category: "Suits"
   color: null
   id: "25686364"
   image: "http://bluecore01-tech-prtnr-na05-dw.demandware.net/on/demandware.static/-/Sites-apparel-catalog/default/dw5c09254c/images/large/
   isMaster: true
   isProduct: true
   isProductSet: false
   isVariant: false
   maxPrice: 299.99
   minPrice: 299.99
   name: "Charcoal Single Pleat Striped Wool Suit"
   outOfStock: false
   price: 299.99
   size: null
   url: "http://bluecore01-tech-prtnr-na05-dw.demandware.net/s/spicey-test/25686364.html"
  ▶variants: (16) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}]
   width: null
  ▶__proto__: Object
BLUECORE REQUEST:                                                                     test_demandware_partner.js:32
▶{event: "viewed_product", properties: {…}}                                           test_demandware_partner.js:32
```

If there is an issue seeing these events, it's likely an issue calling our track product event within the "detail" section of the Controller (or pipeline)

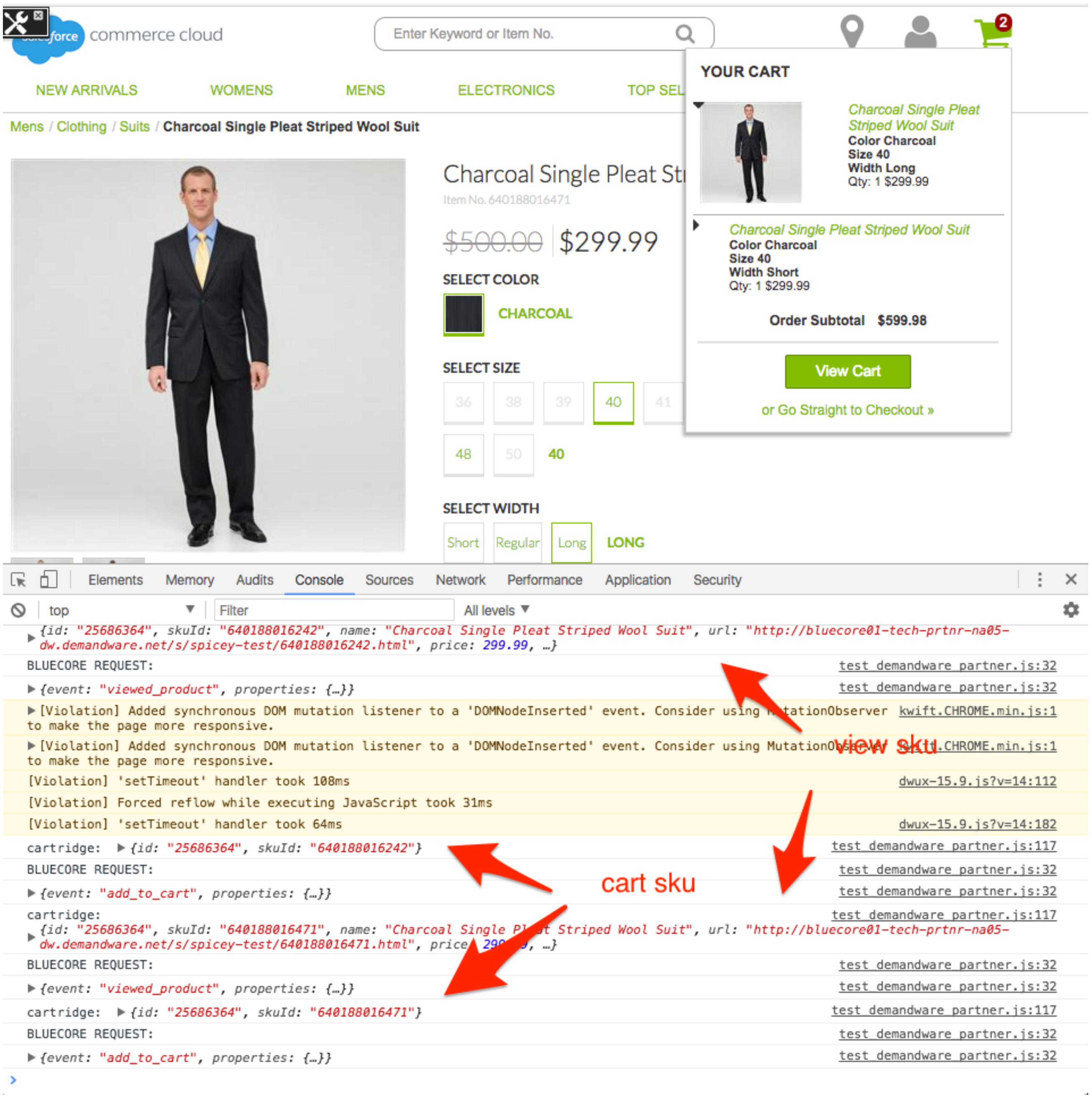
Sku Clicks and Quick Views:
Typical demandware sites reuse the "Product-variation" controller (or pipeline) for both the selection of skus, and quick views. Thus, when we call our track product event within the "variation" flow, we should see payloads fire on quick view clicks, as well specific sku clicks (either within the quick view modal or on PDPs):



```
cartridge:  ▶{category: "Suits", breadcrumbs: Array(3)}                       test_demandware_partner.js:117
BLUECORE REQUEST:                                                              test_demandware_partner.js:32
▶{event: "search", properties: {…}}                                            test_demandware_partner.js:32
▶[Violation] Added synchronous DOM mutation listener to a 'DOMNodeInserted' event. Consider using MutationObserver   kwift.CHROME.min.js:1
to make the page more responsive.
[Violation] 'setTimeout' handler took 219ms                                    dwux-15.9.js?v=14:112
[Violation] Forced reflow while executing JavaScript took 74ms
[Violation] 'setTimeout' handler took 78ms                                     dwux-15.9.js?v=14:182
cartridge:                                                                     test_demandware_partner.js:117
▶{id: "25686364", name: "Charcoal Single Pleat Striped Wool Suit", url: "http://bluecore01-tech-prtnr-na05-dw.demandware.net/s/spicey-
 test/25686364.html", price: 299.99, minPrice: 299.99, …}
BLUECORE REQUEST:                                                              test_demandware_partner.js:32
▶{event: "viewed_product", properties: {…}}                                    test_demandware_partner.js:32
cartridge:                                                                     test_demandware_partner.js:117
▶{id: "25686364", skuId: "640188016204", name: "Charcoal Single Pleat Striped Wool Suit", url: "http://bluecore01-tech-prtnr-na05-
 dw.demandware.net/s/spicey-test/640188016204.html", price: 299.99, …}
BLUECORE REQUEST:                                                              test_demandware_partner.js:32
▶{event: "viewed_product", properties: {…}}                                    test_demandware_partner.js:32
cartridge:                                                                     test_demandware_partner.js:117
▶{id: "25686364", skuId: "640188016242", name: "Charcoal Single Pleat Striped Wool Suit", url: "http://bluecore01-tech-prtnr-na05-
 dw.demandware.net/s/spicey-test/640188016242.html", price: 299.99, …}
BLUECORE REQUEST:                                                              test_demandware_partner.js:32
▶{event: "viewed_product", properties: {…}}                                    test_demandware_partner.js:32
```

If there is an issue seeing the quick view, or sku selections (which trigger only when you select a full variant, i.e. a color, size, and width), there is likely an issue calling our track product event within the "variation" flow, or the site does not use the out-of-the-box Product-Variation flow.

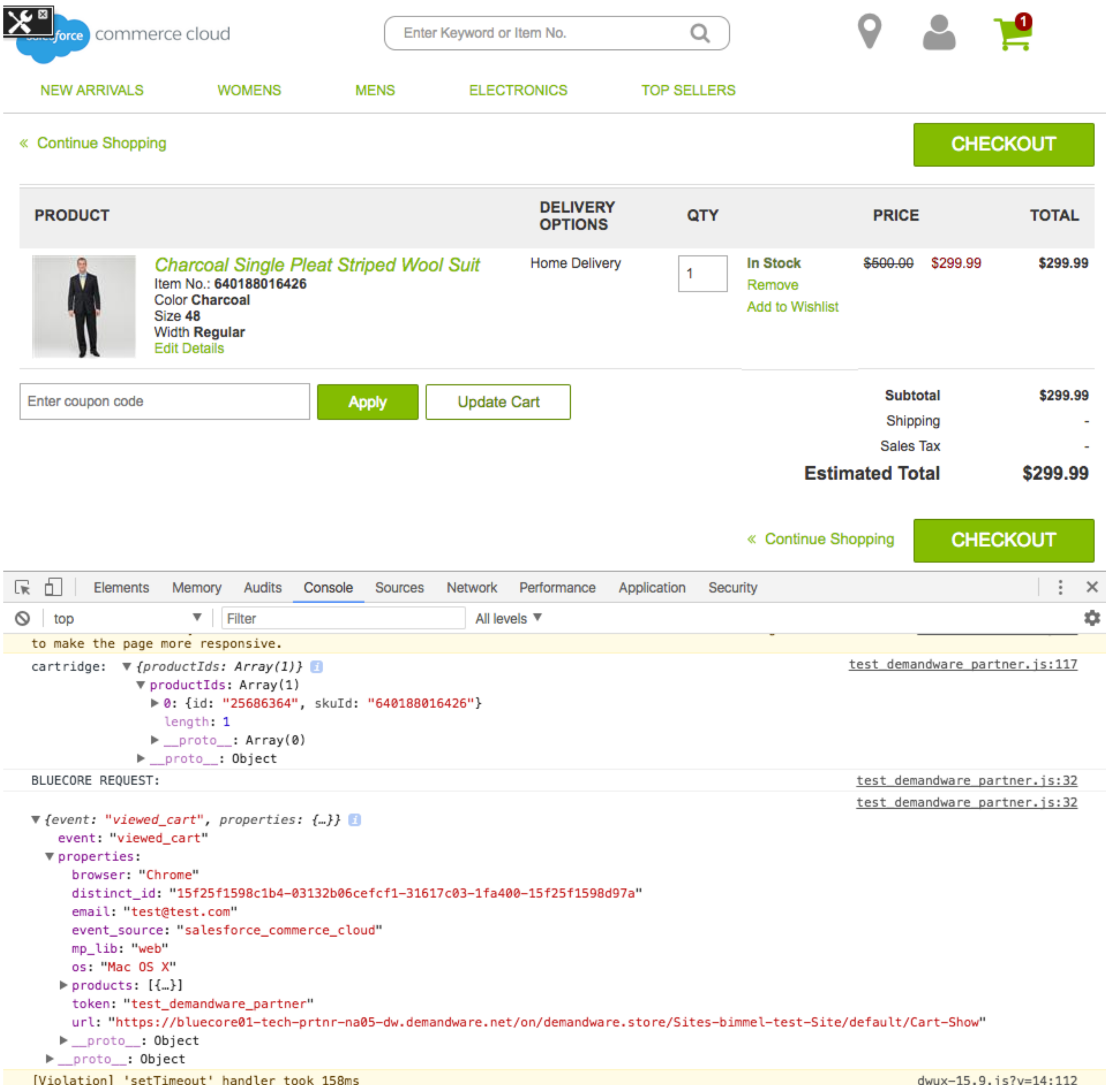
Add to Cart:



```
 ▶{id: "25686364", skuId: "640188016242", name: "Charcoal Single Pleat Striped Wool Suit", url: "http://bluecore01-tech-prtnr-na05-
  dw.demandware.net/s/spicey-test/640188016242.html", price: 299.99, …}
BLUECORE REQUEST:                                                                          test_demandware_partner.js:32
▶{event: "viewed_product", properties: {…}}                                                test_demandware_partner.js:32
▶[Violation] Added synchronous DOM mutation listener to a 'DOMNodeInserted' event. Consider using MutationObserver kwift.CHROME.min.js:1
to make the page more responsive.
▶[Violation] Added synchronous DOM mutation listener to a 'DOMNodeInserted' event. Consider using MutationObserver kwift.CHROME.min.js:1
to make the page more responsive.
[Violation] 'setTimeout' handler took 108ms                                                 dwux-15.9.js?v=14:112
[Violation] Forced reflow while executing JavaScript took 31ms
[Violation] 'setTimeout' handler took 64ms                                                  dwux-15.9.js?v=14:182
cartridge:  ▶{id: "25686364", skuId: "640188016242"}                                        test_demandware_partner.js:117
BLUECORE REQUEST:                                                                          test_demandware_partner.js:32
▶{event: "add_to_cart", properties: {…}}                                                   test_demandware_partner.js:32
cartridge:                                                                                 test_demandware_partner.js:117
 ▶{id: "25686364", skuId: "640188016471", name: "Charcoal Single Pleat Striped Wool Suit", url: "http://bluecore01-tech-prtnr-na05-
  dw.demandware.net/s/spicey-test/640188016471.html", price: 299.99, …}
BLUECORE REQUEST:                                                                          test_demandware_partner.js:32
▶{event: "viewed_product", properties: {…}}                                                test_demandware_partner.js:32
cartridge:  ▶{id: "25686364", skuId: "640188016471"}                                        test_demandware_partner.js:117
BLUECORE REQUEST:                                                                          test_demandware_partner.js:32
▶{event: "add_to_cart", properties: {…}}                                                   test_demandware_partner.js:32
>
```

If cartridge events are not being displayed, there was likely an error in adding our addToCart functionality to the Cart controller/pipeline "addProduct" flow.

View Cart



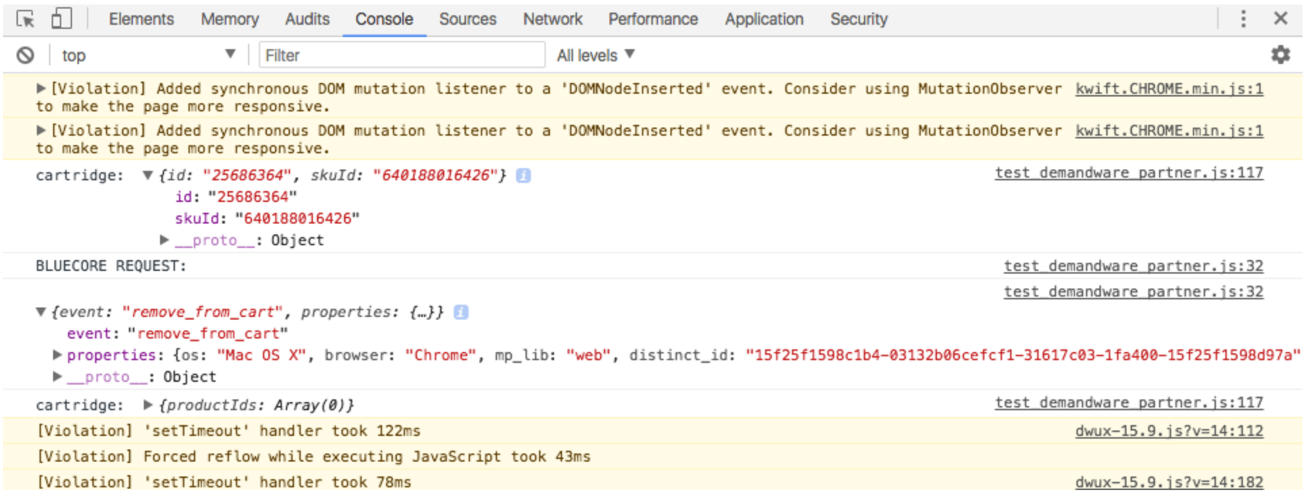
PRODUCT	DELIVERY OPTIONS	QTY		PRICE	TOTAL
Charcoal Single Pleat Striped Wool Suit Item No.: **640188016426** Color **Charcoal** Size **48** Width **Regular** Edit Details	Home Delivery	1	**In Stock** Remove Add to Wishlist	~~$500.00~~ $299.99	$299.99

Subtotal	$299.99
Shipping	-
Sales Tax	-
Estimated Total	**$299.99**

```
  to make the page more responsive.
  cartridge:  ▼{productIds: Array(1)}                                      test_demandware_partner.js:117
                ▼productIds: Array(1)
                  ▶0: {id: "25686364", skuId: "640188016426"}
                   length: 1
                  ▶__proto__: Array(0)
                ▶__proto__: Object
  BLUECORE REQUEST:                                                          test_demandware_partner.js:32
                                                                             test_demandware_partner.js:32
  ▼{event: "viewed_cart", properties: {…}}
      event: "viewed_cart"
    ▼properties:
        browser: "Chrome"
        distinct_id: "15f25f1598c1b4-03132b06cefcf1-31617c03-1fa400-15f25f1598d97a"
        email: "test@test.com"
        event_source: "salesforce_commerce_cloud"
        mp_lib: "web"
        os: "Mac OS X"
      ▶products: [{…}]
        token: "test_demandware_partner"
        url: "https://bluecore01-tech-prtnr-na05-dw.demandware.net/on/demandware.store/Sites-bimmel-test-Site/default/Cart-Show"
      ▶__proto__: Object
    ▶__proto__: Object
  [Violation] 'setTimeout' handler took 158ms                                          dwux-15.9.js?v=14:112
```

This event is a result of adding the ViewCart function call to templates/checkout/cart/pt_cart_VARS/pt_cart_VARS.isml in section 3.2.1.

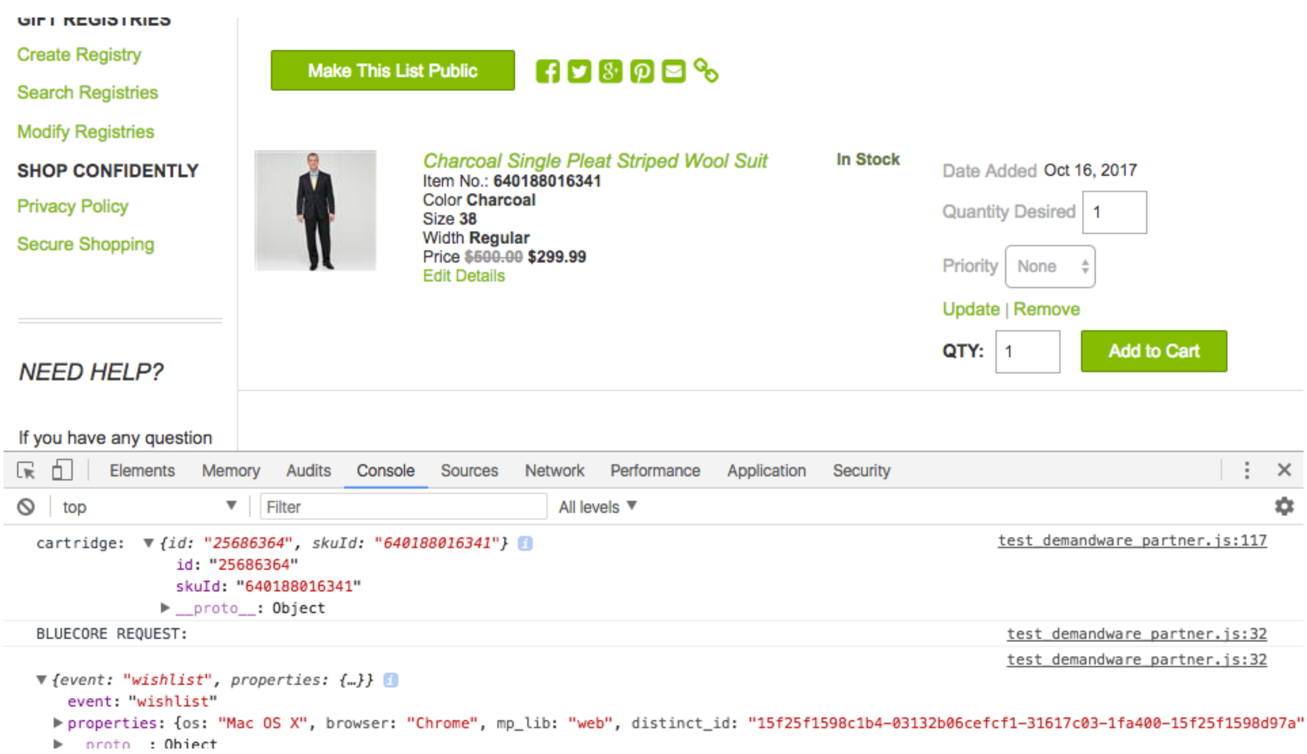
Remove from Cart



```
▶[Violation] Added synchronous DOM mutation listener to a 'DOMNodeInserted' event. Consider using MutationObserver  kwift.CHROME.min.js:1
to make the page more responsive.
▶[Violation] Added synchronous DOM mutation listener to a 'DOMNodeInserted' event. Consider using MutationObserver  kwift.CHROME.min.js:1
to make the page more responsive.
cartridge:  ▼{id: "25686364", skuId: "640188016426"}                                          test_demandware_partner.js:117
                id: "25686364"
                skuId: "640188016426"
              ▶__proto__: Object
BLUECORE REQUEST:                                                                               test_demandware_partner.js:32
                                                                                                test_demandware_partner.js:32
▼{event: "remove_from_cart", properties: {…}}
    event: "remove_from_cart"
  ▶properties: {os: "Mac OS X", browser: "Chrome", mp_lib: "web", distinct_id: "15f25f1598c1b4-03132b06cefcf1-31617c03-1fa400-15f25f1598d97a"
  ▶__proto__: Object
cartridge:  ▶{productIds: Array(0)}                                                            test_demandware_partner.js:117
[Violation] 'setTimeout' handler took 122ms                                                      dwux-15.9.js?v=14:112
[Violation] Forced reflow while executing JavaScript took 43ms
[Violation] 'setTimeout' handler took 78ms                                                       dwux-15.9.js?v=14:182
```

This event corresponds to adding the RemoveFromCart functionality in 3.2.2 / 3.2.3.

Wishlist



Charcoal Single Pleat Striped Wool Suit
Item No.: 640188016341
Color Charcoal
Size 38
Width Regular
Price ~~$500.00~~ $299.99
Edit Details

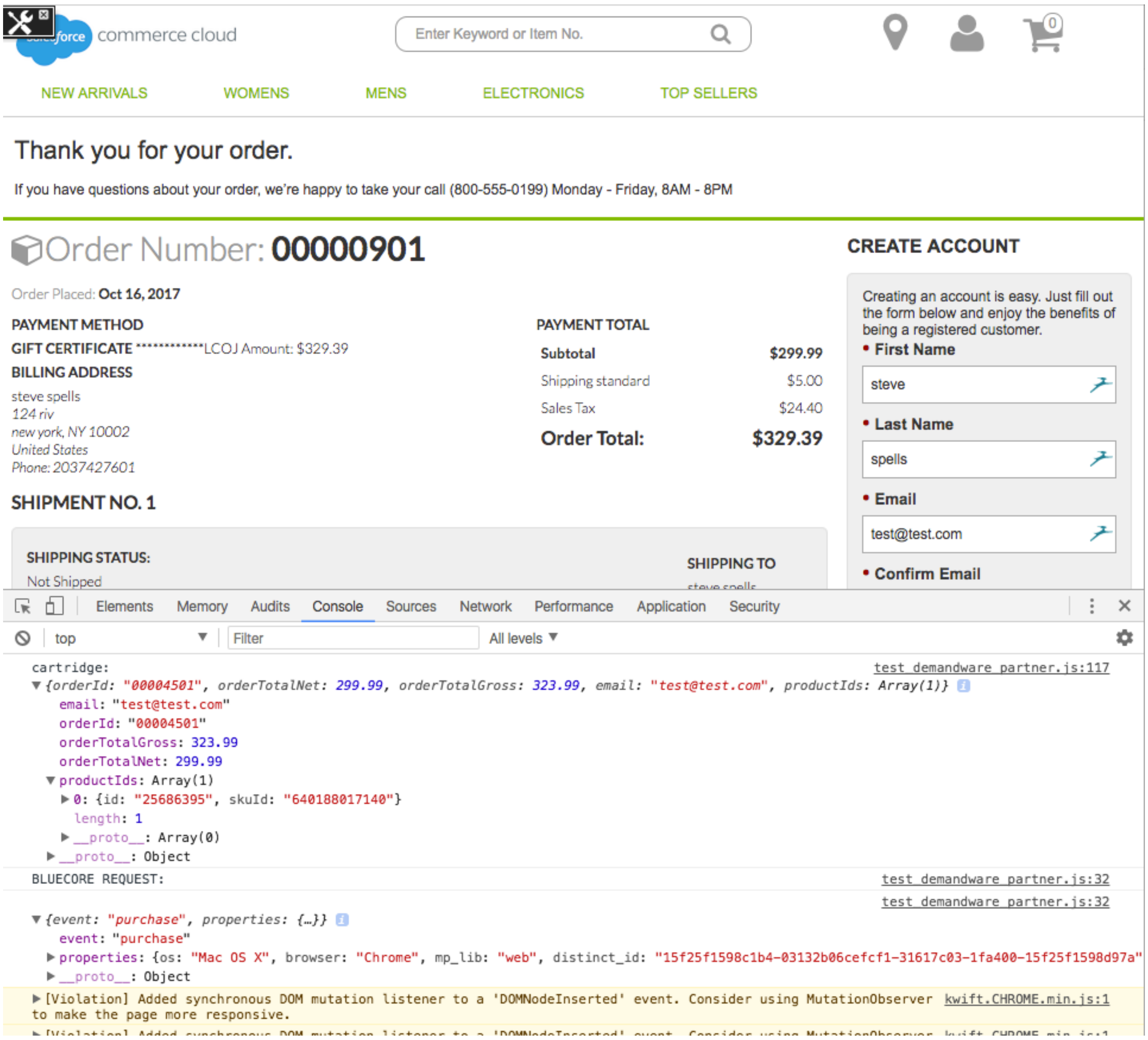
In Stock

Date Added Oct 16, 2017
Quantity Desired 1
Priority None
Update | Remove
QTY: 1 Add to Cart

```
cartridge:  ▼{id: "25686364", skuId: "640188016341"}                                          test_demandware_partner.js:117
                id: "25686364"
                skuId: "640188016341"
              ▶__proto__: Object
BLUECORE REQUEST:                                                                               test_demandware_partner.js:32
                                                                                                test_demandware_partner.js:32
▼{event: "wishlist", properties: {…}}
    event: "wishlist"
  ▶properties: {os: "Mac OS X", browser: "Chrome", mp_lib: "web", distinct_id: "15f25f1598c1b4-03132b06cefcf1-31617c03-1fa400-15f25f1598d97a"
  ▶__proto__: Object
```

This event fires only if both the AddToWishlist and ViewWishlist functions are being called appropriately in 3.2.2 and 3.2.1 respectively.

Checkout



Thank you for your order.

If you have questions about your order, we're happy to take your call (800-555-0199) Monday - Friday, 8AM - 8PM

Order Number: 00000901

Order Placed: Oct 16, 2017

PAYMENT METHOD

GIFT CERTIFICATE ************LCOJ Amount: $329.39

BILLING ADDRESS

steve spells
124 riv
new york, NY 10002
United States
Phone: 2037427601

PAYMENT TOTAL

Subtotal	$299.99
Shipping standard	$5.00
Sales Tax	$24.40
Order Total:	$329.39

SHIPMENT NO. 1

SHIPPING STATUS:
Not Shipped

SHIPPING TO

CREATE ACCOUNT

Creating an account is easy. Just fill out the form below and enjoy the benefits of being a registered customer.

• First Name: steve

• Last Name: spells

• Email: test@test.com

• Confirm Email

```
cartridge:                                                                          test_demandware_partner.js:117
▼{orderId: "00004501", orderTotalNet: 299.99, orderTotalGross: 323.99, email: "test@test.com", productIds: Array(1)}
   email: "test@test.com"
   orderId: "00004501"
   orderTotalGross: 323.99
   orderTotalNet: 299.99
  ▼productIds: Array(1)
   ▶0: {id: "25686395", skuId: "640188017140"}
    length: 1
   ▶__proto__: Array(0)
  ▶__proto__: Object
BLUECORE REQUEST:                                                                    test_demandware_partner.js:32
                                                                                     test_demandware_partner.js:32
▼{event: "purchase", properties: {…}}
   event: "purchase"
  ▶properties: {os: "Mac OS X", browser: "Chrome", mp_lib: "web", distinct_id: "15f25f1598c1b4-03132b06cefcf1-31617c03-1fa400-15f25f1598d97a"
  ▶__proto__: Object
▶[Violation] Added synchronous DOM mutation listener to a 'DOMNodeInserted' event. Consider using MutationObserver kwift.CHROME.min.js:1
to make the page more responsive.
```

This event corresponds to the Checkout functionality added in 3.2.1.

3.4.3 Notes on Testing

- If Cartridge payloads are being logged, but Bluecore Requests are not, please contact the support team in 4.3, as this is most likely an issue of handling the payload the demandware cartridge has handed off, which is something that can be resolved within the external script pulled in.
- When QA'ing viewed_product events, you may see 2 Bluecore events fired: the "viewed_product" event as well as a "patch". The patch event fires only a small percentage of the time (1% by default) to help keep the Bluecore catalog up to date.
- When QA'ing, you may also see "identify" events, which are handled purely by the external Javascript. Those do not need to be QA'd.

4. Operations, Maintenance

4.1 Data Storage

The Bluecore cartridge does not store any data within Demandware. All data tracked and collected by the cartridge is stored by Bluecore at its secure data centers.

4.2 Availability

The availability of the cartridge itself is expected to be 100%. The availability of the tracking interfaces and APIs that the cartridge communicates with is dependent on your Bluecore Master Services Agreement or Statement of Work. Please contact to your Bluecore Customer Success team for further details.

4.3 Support

Bluecore has several support resources:

1. For product documentation, visit support.bluecore.com.

2. For further help, please contact support@bluecore.com and a Product Specialist will assist you. Product Specialists are available Monday through Friday, 9:00 AM to 6:00 PM Eastern Standard Time (excluding US holidays).
3. Depending on your Bluecore Master Services Agreement or Statement of Work, you may also have a dedicated Bluecore Customer Success team for general day-to-day support.

5. User Guide

5.1 Roles, Responsibilities

No recurring tasks or responsibilities are required after initial implementation of the Bluecore cartridge.

5.2 Storefront Functionality

The Bluecore cartridge does not add any new storefront functionality.

6. Known Issues

N/A

7. Release History

Version	Date	Changes
16.1.0	Oct. 5, 2016	Initial release
16.2.0	Oct. 11, 2017	▪ Updated documentation ▪ Fetch category from master when parsing a variant ▪ Allow variants products to include attributevariation parsing ▪ Provide fallback support to product images options ▪ Convert tabs to spaces ▪ Pull out category fallback logic into own function ▪ Update parsedProducts to fetch peer variants on variant PDPs ▪ Convert tabs to spaces ▪ Update viewProduct to support Product input in pipeline ▪ Update custom attribute fetching on root parsed product