**GA360 Integration**

Adding Google Analytics to Aurora B2C Store, using Google Tag Manager

Requirements Outline

Provide the reporting and tracking abilities on Aurora Store that allows to track and understand the customer's behaviour, what they are looking for, and whether business is addressing their needs.

**Implementation Guide:**

We need the below things for implementation of GA360 on Aurora store

Google analytics – Analytics Tool

Google tag manager – Tag Management.

GTM configuration script tag – Integrate GTM code snippet in Aurora B2C.

Firstly, we need to create account on both Google Analytics and google tag manager.

Google Analytics – [Click here](https://teams.microsoft.com/l/file/611EDE70-7F71-43C9-95F5-B727E51AB01A?tenantId=189de737-c93a-4f5a-8b68-6f4ca9941912&fileType=docx&objectUrl=https%3A%2F%2Fhclo365.sharepoint.com%2Fsites%2Finnovationfactory%2FShared%20Documents%2FGoogle%20Analytics%20360%20Integration%2FGoogle%20Analytics_CreateAccount_Steps.docx&baseUrl=https%3A%2F%2Fhclo365.sharepoint.com%2Fsites%2Finnovationfactory&serviceName=teams&threadId=19:2f9390ba8ee248eba516c36b7287dcc4@thread.tacv2&groupId=633ba5ef-06b3-46a7-ae83-2044288cf9e5) to get the steps for account creation

Google Tag Manger – [Click here](https://teams.microsoft.com/l/file/83B5F016-61A6-4BE2-BE69-77CFE58410DC?tenantId=189de737-c93a-4f5a-8b68-6f4ca9941912&fileType=docx&objectUrl=https%3A%2F%2Fhclo365.sharepoint.com%2Fsites%2Finnovationfactory%2FShared%20Documents%2FGoogle%20Analytics%20360%20Integration%2FGTM%20Account%20creation%20Steps.docx&baseUrl=https%3A%2F%2Fhclo365.sharepoint.com%2Fsites%2Finnovationfactory&serviceName=teams&threadId=19:2f9390ba8ee248eba516c36b7287dcc4@thread.tacv2&groupId=633ba5ef-06b3-46a7-ae83-2044288cf9e5) to get the steps for account creation

Once the account creation is done then we have to add the GTM Configuration script tag in the gtmService.js file and initialize the same as below.



**Google Tag Manger**

Please refer [the pdf](https://teams.microsoft.com/l/file/D0E1FF5C-5842-4A0D-A7F8-2791AE617F58?tenantId=189de737-c93a-4f5a-8b68-6f4ca9941912&fileType=pdf&objectUrl=https%3A%2F%2Fhclo365.sharepoint.com%2Fsites%2Finnovationfactory%2FShared%20Documents%2FGoogle%20Analytics%20360%20Integration%2FGoogle%20Tag%20Manger%20configuration%20sheet.pdf&baseUrl=https%3A%2F%2Fhclo365.sharepoint.com%2Fsites%2Finnovationfactory&serviceName=teams&threadId=19:2f9390ba8ee248eba516c36b7287dcc4@thread.tacv2&groupId=633ba5ef-06b3-46a7-ae83-2044288cf9e5) for the creation on tags, triggers and variables in Google Tag Manager.

**Add GTM Configurations to storeconf table:**

Run this query to store the Google Tag Manager information (id, authorization, env) for Aurora in the storeconf table.

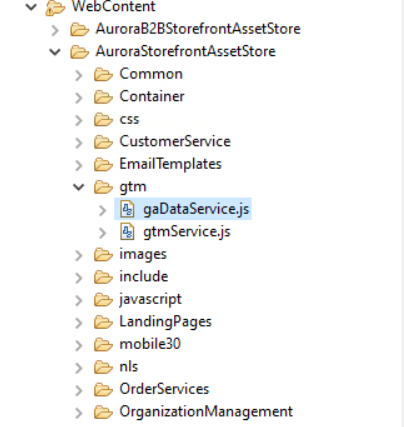
* INSERT INTO storeconf (storeent\_id, name, value) VALUES (1, 'google.tag.manager.container.id', 'GTM-T9CV742'), (1, 'google.tag.manager.auth', 'rkaDopVOtire-Dsdy3HJ4Q'), (1, 'google.tag.manager.preview', 'env-1'));

**Aurora B2C Store**

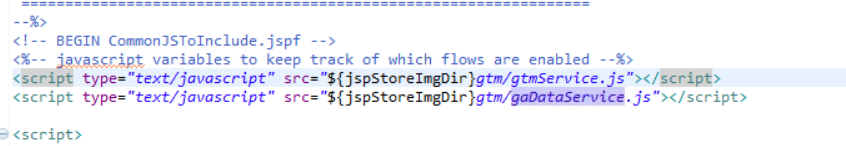
Created the two files for GA360 and kept under

“**crs-web/WebContent/AuroraStorefrontAssetStore/gtm**” folder

1. **gtmService.js** - specific to GTM configurations and initializing the GTM for AuroraB2C.
2. **gtmDataService.js** - Files that includes various methods to record the google analytics for various page load, product details checkout etc.



1. **Include both the files in common JS to Include file to initialize GTM**



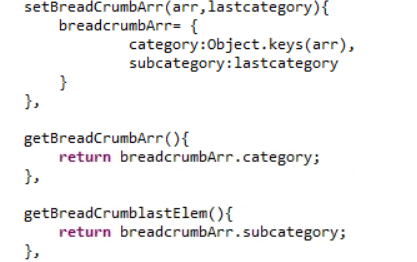
In gtmService.js, initailizeGTM () method is written.

The initializeGTM() Method has all the configurations related to the GTM account created.

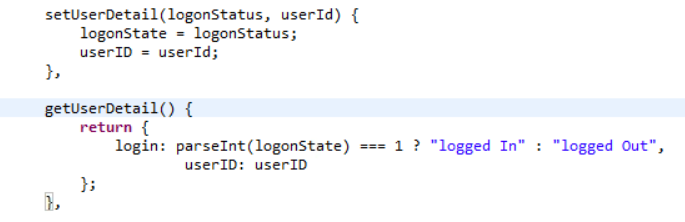
Its added in a script tag in the common js file to be included.

In gaDataService.js, we are calling different common functions which will help us to get the common information like,

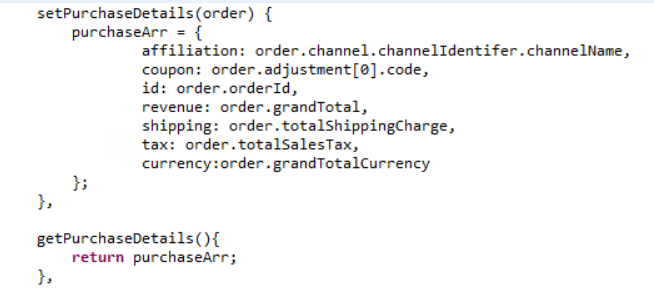
* setBreadCrumbArr() – this function help user for creating the breadcrumb array. This is called from /crs-web/WebContent/Widgets\_801/com.ibm.commerce.store.widgets.BreadcrumbTrail/BreadcrumbTrail\_UI.jspf file.
* getBreadCrumbArr() – this function help us to retrieve the category from breadcrumb array.
* getBreadCrumblastElem() – this function help us to get the last element I.e. the current page name.



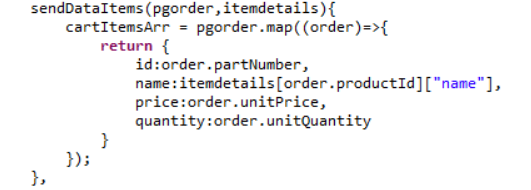
* setUserDetail() – this function help to set the user details like login status and userId. This function is called from /crs-web/WebContent/Widgets\_801/com.ibm.commerce.store.widgets.BreadcrumbTrail/BreadcrumbTrail\_UI.jspf file.
* getUserDetail() – this function returns the user details.



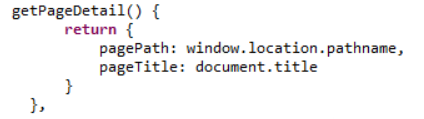
* setPurchaseDetails() – this helps to create a purchase array with information like coupon code, order id, revenue generated, shipping, tax, currency, etc. This function is called from /crs-web/WebContent/AuroraStorefrontAssetStore/ShoppingArea/CheckoutSection/SingleShipment/SingleShipmentOrderTotalsSummary.jsp
* getPurchaseDetails() – this function returns purchase array.



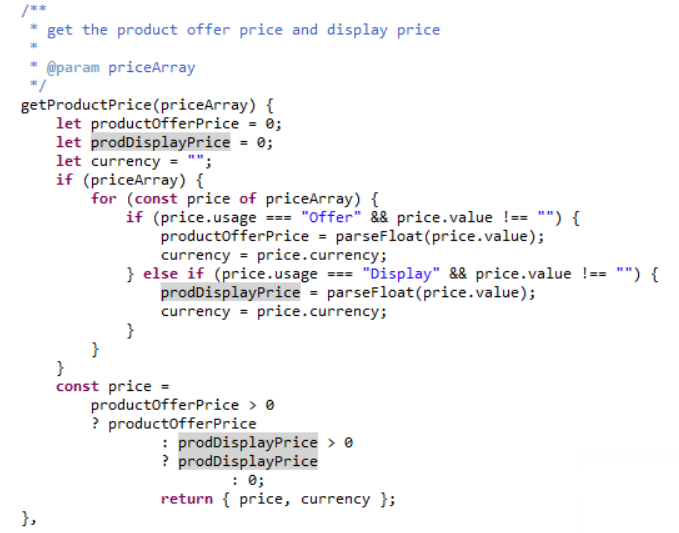
* sendDataItems() – this function is used to create an array of items in cart.



* getPageDetail() – this function is used to get the page path and title.

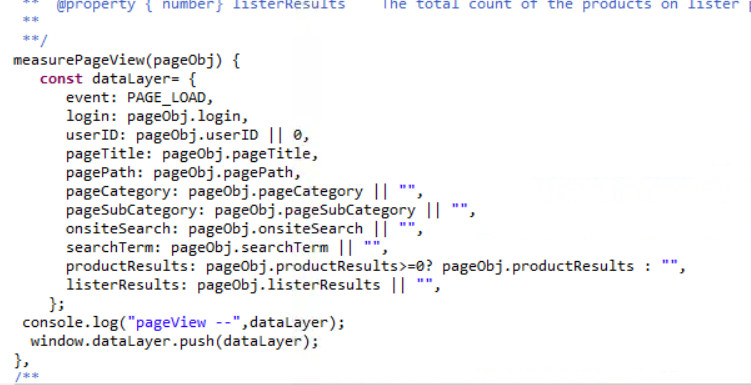


getProductPrice() – this function will help to get the price of the product from the price object.



1. **Pageview events**

In this, we will be sending page related information to GA.The measurePageview() method defines in gtmService.js file. Its responsible for adding object (required information which we want to send) in data layer.



We have sent the pageview event from following pages

* Home Page
* Lister page
* Search Page
* PDP Page
* Content Page
* Checkout Page

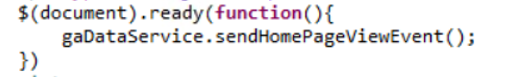
The methods are defined in the gaDataService.js file and we are calling those methods from the different pages with required arguments.

For Example:

**Home Page**

**(/crs-web/WebContent/AuroraStorefrontAssetStore/Container/HomePageContainer.jsp)**

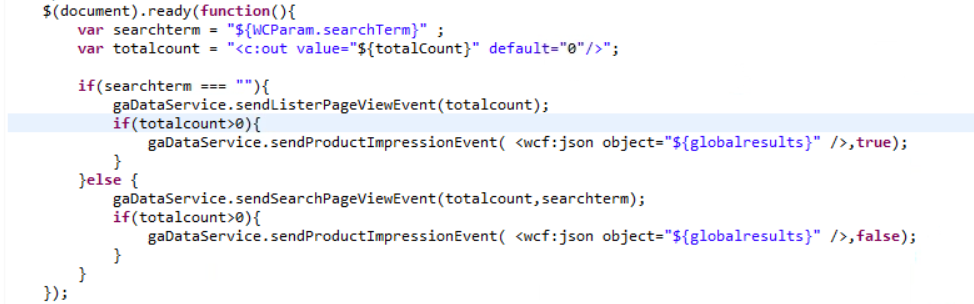
In HomePageContainer Page, need to call the SendHomePageViewEvent() method whose definition is in gaDataService.js file.



**Lister pageview, Search Pageview, Product Impression events**

**(/crs-web/WebContent/Widgets\_801/com.ibm.commerce.store.widgets.CatalogEntryList/CatalogEntryList\_UI.jspf)**

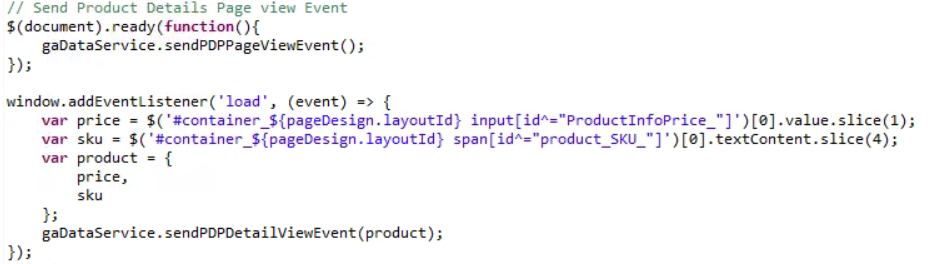
sendListerPageViewEvent() will be called on the product listing page. sendSearchPageViewEvent() will be called when user searches for some item from search bar. sendProductImpressionEvent() will be called on the product category page.



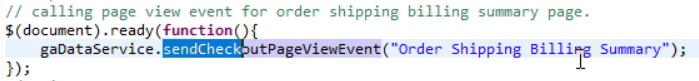
**Product Page Detail pageview**

**(/crs-web/WebContent/AuroraStorefrontAssetStore/Container/ProductPageContainer.jsp)**

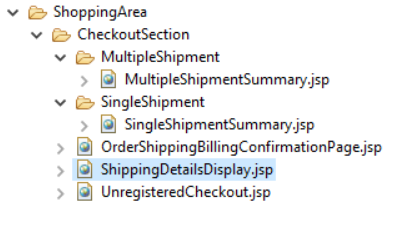
sendPDPPAgeViewEvent() will be called from the product details page.



**Checkout Page (ShoppingDetailsDisplay.jsp)**

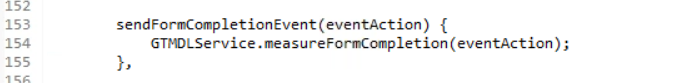


**We are calling the above pageview event from following list of pages as well for sending checkout event during the checkout process.**

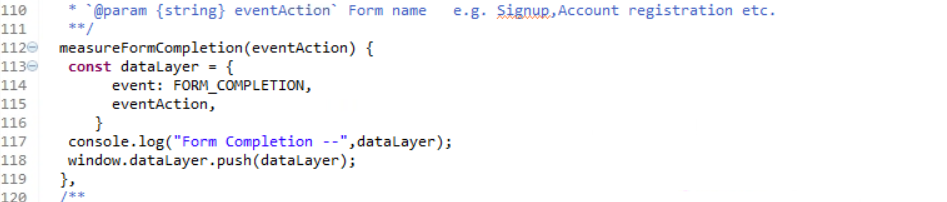


1. **Form Completion Event**

In gaDataService.js file, we have defined the common method with required parameters for form Completion Event.



This methods call the measureFormCompletion() method of gtmService.js for adding the object in datalayer.



We have called the sendFormCompletionEvent from the following the forms:

* SignIn
* Register

Sign In: (/crs-web/WebContent/Widgets\_801/com.ibm.commerce.store.widgets.GlobalLogin/GlobalLoginSignInRegistration\_UI.jspf)

Called sendFormCompletionEvent() from GlobalLoginSignInRegistration\_UI.jspf file after user successfully logged in.



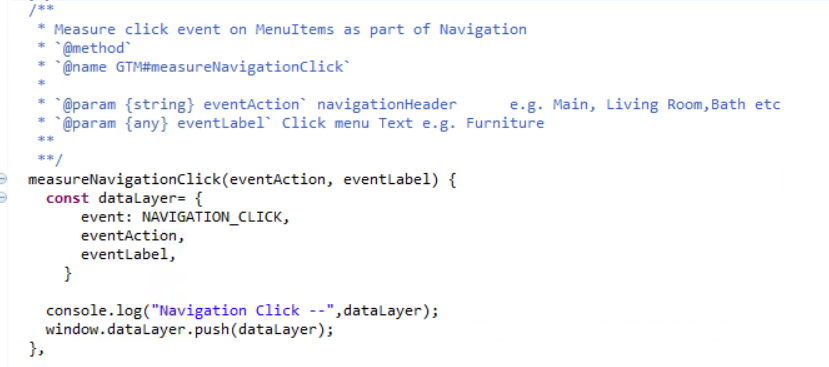
Register: (/crs-web/WebContent/Widgets\_801/com.ibm.commerce.store.widgets.RegistrationForm/RegistrationForm\_UI.jspf)

Called sendFormCompletionEvent() RegistrationForm\_UI.jspf file after the registration is successful.



1. **Navigation Click**

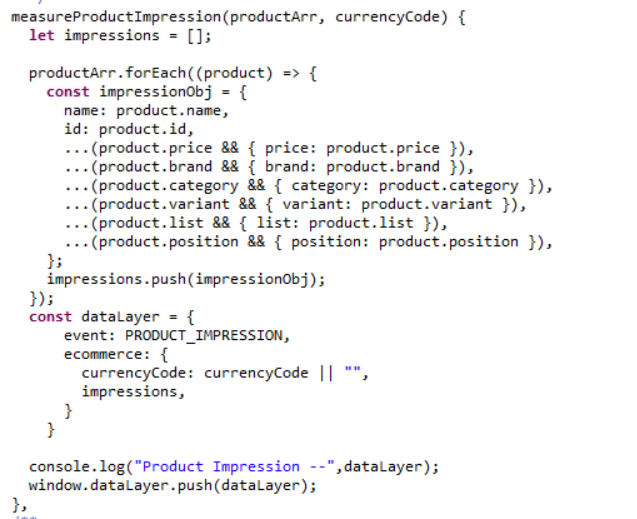
In gtmService.js file, we have defined the common method with required parameters for navigation click Event. This method will be called when we click on any menu item.



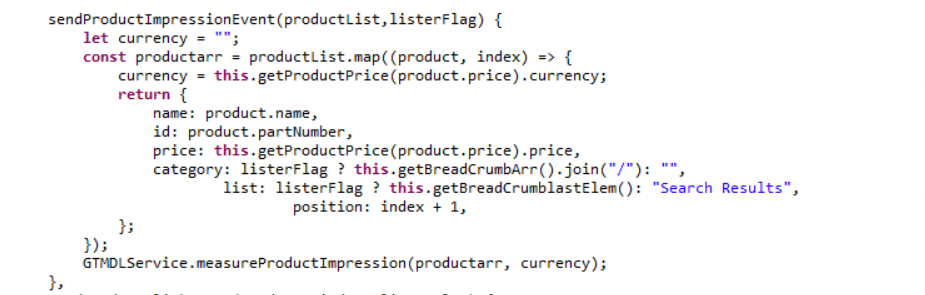
1. **Product Impression**

Product Impression is how many products visible on the page when user visited the page.

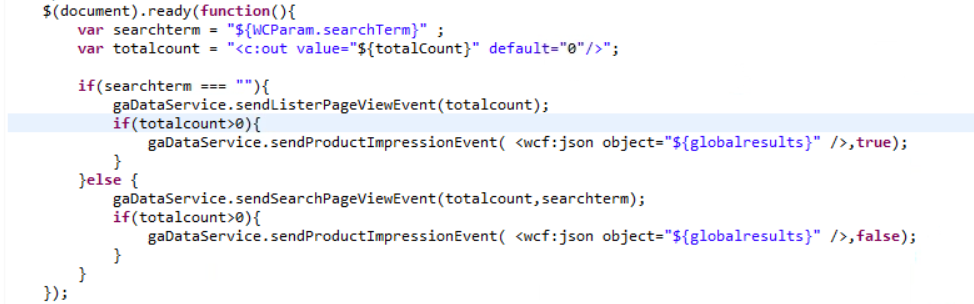
In gtmService.js, we have defined the method which will data the product data to data layer.



This method is getting called from sendproductImpressionevent() method of gaDataService.js file which will send the array of products to GTM service.



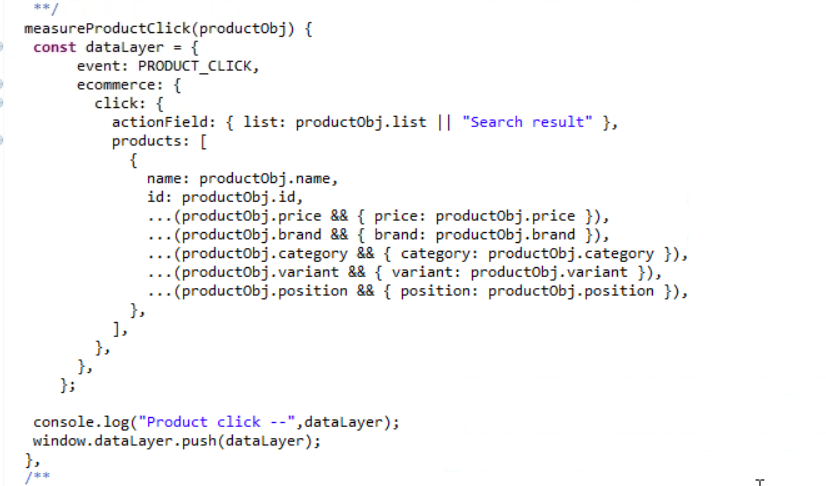
This common method is called from Lister page where products are getting rendered. (CatalogEntryList\_UI.jspf) If there is no category, then it means are products are rendered as part of search Result otherwise as part of selected Category.



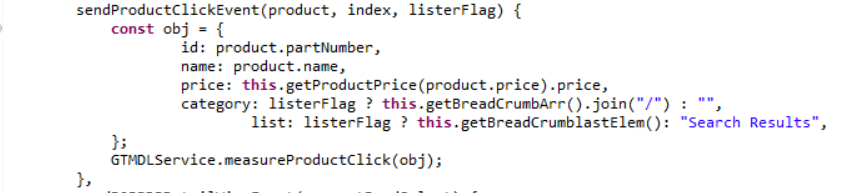
1. **Product Click**

Product click is the event which we sent when user clicks on a product from lister page.

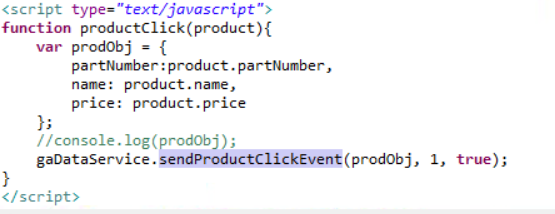
In gtmService.js, we have defined the method which will send the product data to data layer.



This method is getting called from sendProductClickEvent () method of gaDataService.js file which will send the clicked product info to method of gtmService.js.



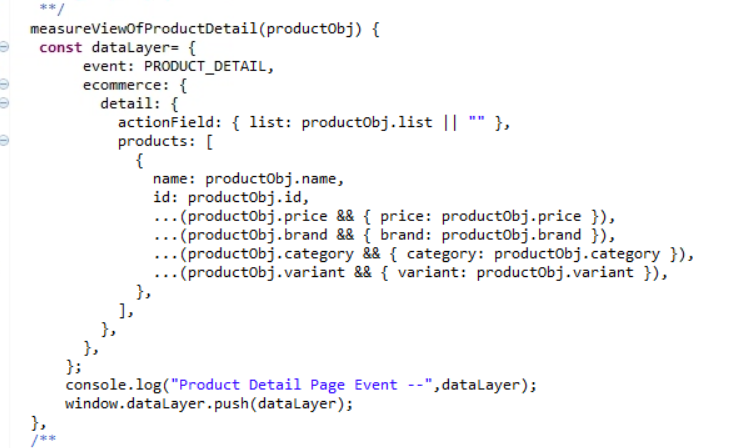
This common method is called from CataogEntryDisplay\_GridView\_UI.jspf where products are getting rendered.



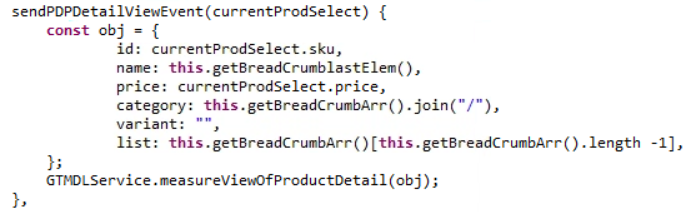
1. **Product Detail View**

The product detail View will be sent when user view the detail page of product.

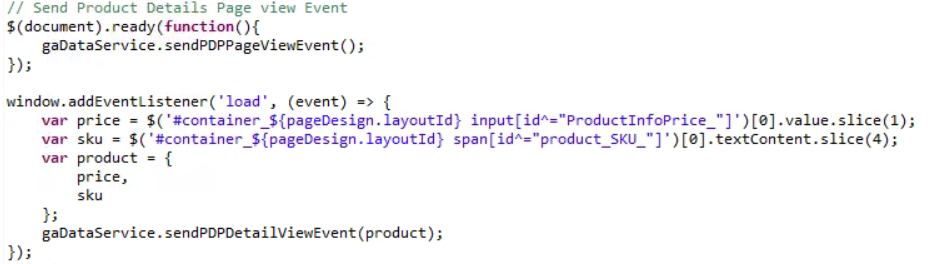
The Method which will add product data to data layer is defined in gtmService.js.



This method is getting called from sendPDPDetailViewEvent() in gaDataService.js file which provide product object to the above method.



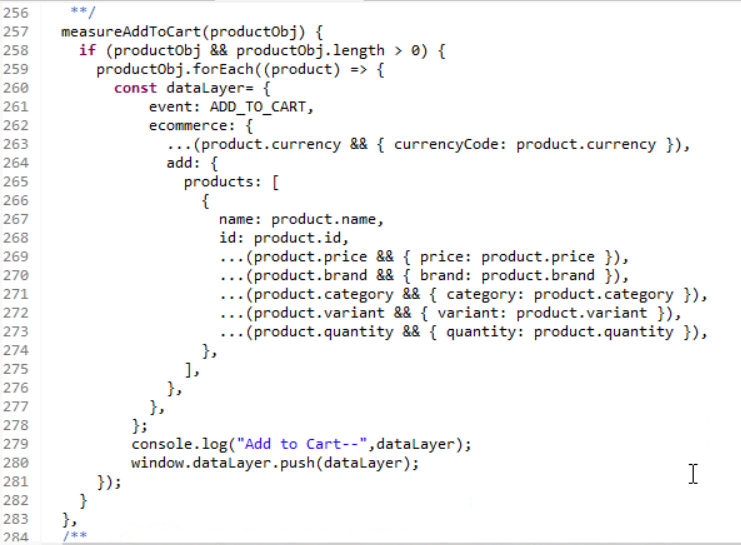
This method is called from product detail page i.e. ProductPageContainer.jsp



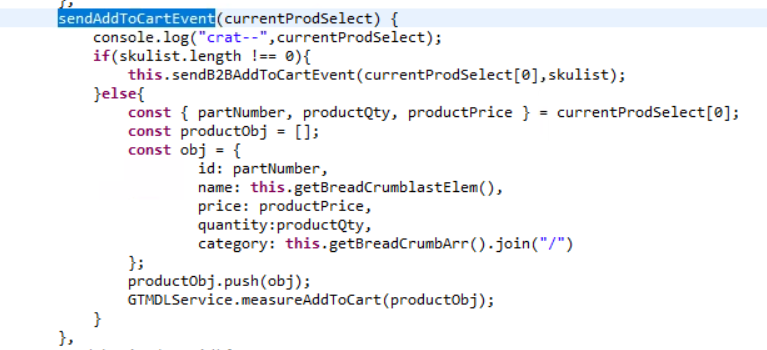
1. **Add to Cart**

We send the AddToCart event when user clicks on add to cart button and order is added in cart. We send the product information which is added in cart for datalayer.

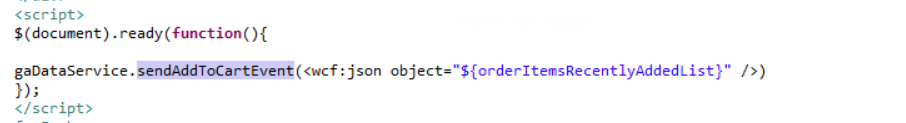
The Main method which takes product object and send data to datalayer is defined in gtmService.js



This method is called from gaDataService.js where we do the manipulation on data and get the required info of the product and pass it to above method for adding in datalayer.



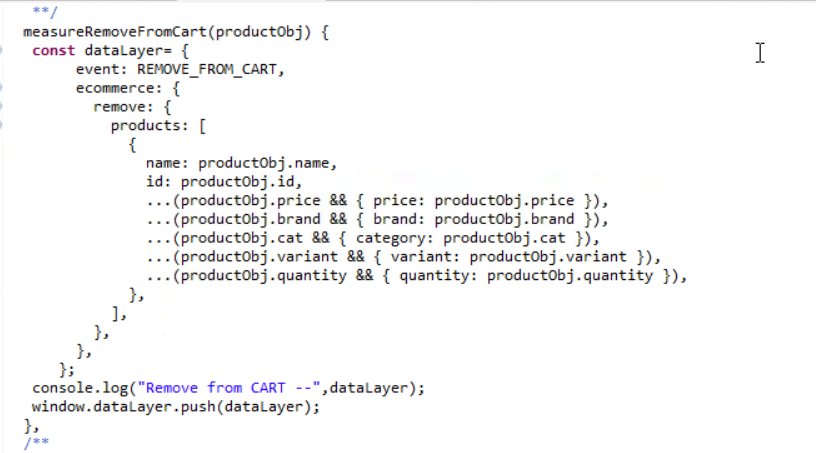
This method is called from detail page i.e. MiniShopCartDisplay\_UI.jspf where add to cart button is present. We pass selected product object which is added in the cart.



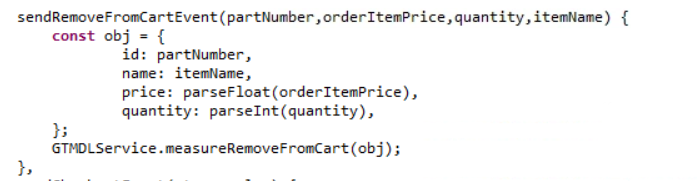
1. **Remove from Cart**

We need to call this event when user remove some item from the cart.

The main method is defined in gtmService.js where object structure is defined and it will add the object to data layer.



This method is called from gaDataService.js which send the removed product object to the above method.



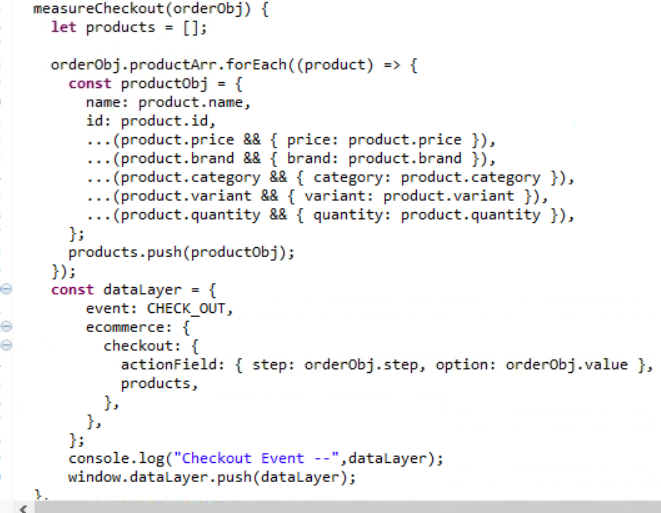
This method is called with item object from cart page where remove button is present. We called the above method on click of remove cart button.



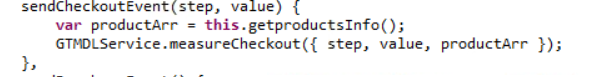
1. **Checkout**

We send the checkout information when user enters in checkout process. We need to send this event in each step completed by the user i.e. shipping and Billing and payment Info.

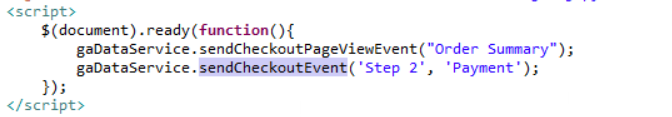
The main method of checkout is defined in gtmService.js file which add product data and step number and step value to data layer.



This Method is getting called from gaDataService.js which will form the data for the above method.

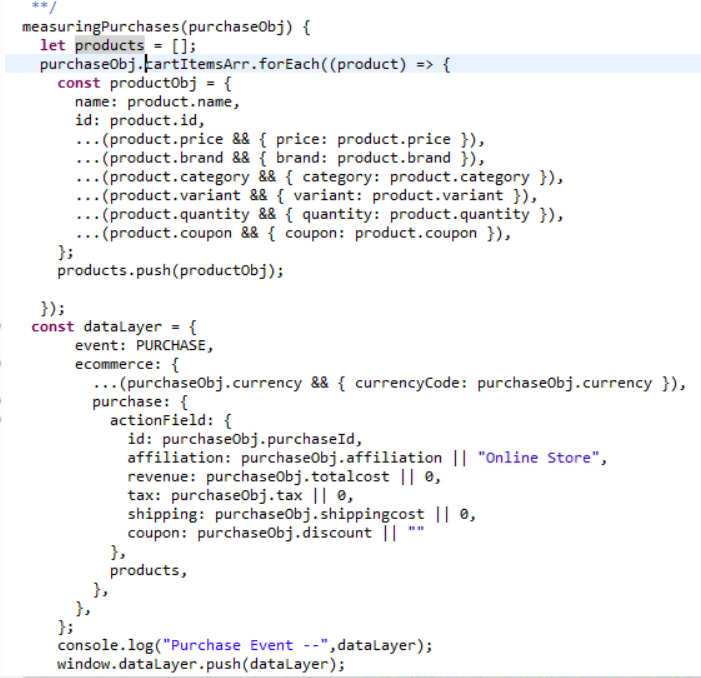


We pass the information to this method from checkout file.

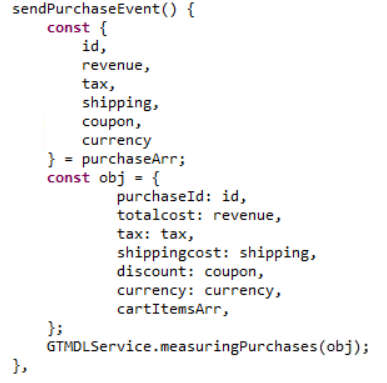


1. **Purchase**

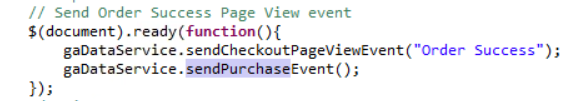
When send this event when user has completed the checkout process and placed an order and get confirmation page. The main method is defined the gtmService.js which adds the purchase order object to datalayer.



This method is getting called from gaDataService.js which the object for the above method.

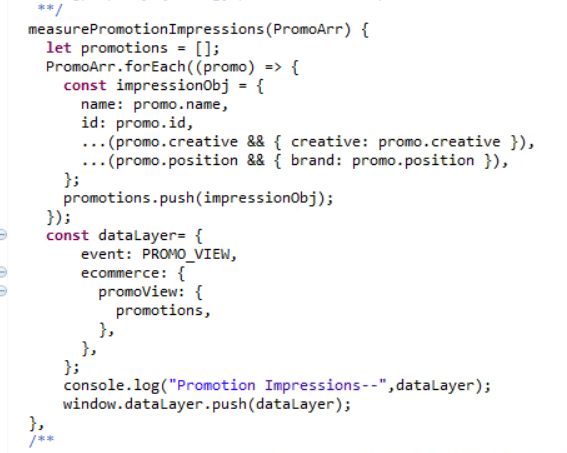


The Cart information and added items information is passed from the checkout page when order is confirmed. The below changes are made in the OrderConfirmationPage.

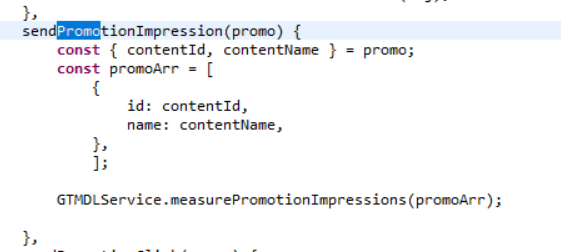


1. **Promotion View**

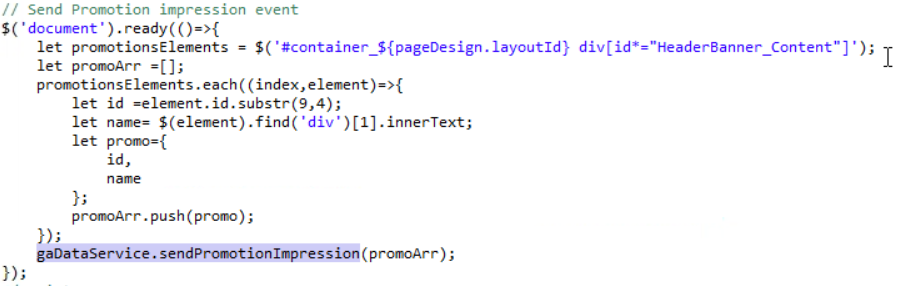
This pageview event is called when promotions are visible on the page opened by user. We send the promotion information rendered on the page. The main method is defined in gtmService.js which will send promotion data to data layer.



This method is called from gaDataService.js from where we are sending the data to above method.



We are accessing this function where we see the promotions on the page. We pass the content id and Content name to the function.



1. **Promotion Click**

The functionality to capture the Promotion Click event is implemented. Whenever the Promotion click functionality will be implemented in Aurora Store, we can address the Promotion click event from that page.