

ccrz.ccLogicWLAddTo

Overview

Called by [ccrz.ccApiWishList](#) during the addTo() method with the intention of creating new Wish List Items from a set of passed in data.

- **processInput** - Used to aggregate/prescreen input parameters and set defaults. Also, hashes the original input map.
- **fetchWishList** - If a Wish List ID is passed in, verifies it is a valid Wish List.
- **fetchActive** - Fetches the active Wish List if no ID is passed in.
- **fetchProduct** - Fetches the products to add by either SKUs or SFIDs.
- **createWishListItems** - Creates the new Wish List Items as sObjects, and prepares them for insert.
- **insertWishListItems** - Perform actual DML insert of new Wish List Items.
- **refetch** - A possible refetch of the Wish List.
- **prepReturn** - Cleans the returning map from any intermediate keys.

process

The process method is the main method that invokes the overall chain of methods listed above and is an override of ccLogic.process. Any exceptions thrown during processing will be thrown out of the method, i.e. there are no catches.

Main Flow

```

Map<String, Object> process(Map<String, Object> inputData){

    // Process Input Data And Set Fall Back Values.
    processInput(inputData);

    // If Wish List ID Is Passed In --> Verify It Is A Real Wish
    List
    fetchWishList(inputData);

    // If No Wish List ID Is Passed In, Fetch The Active Wish List
    (This May Create A New Wish List If No Active One Exists)
    fetchActive(inputData);

    // Fetch The Products To Add By Either SKUs or SFIDs
    fetchProduct(inputData);

    // Create The New Wish List Items As sObjects, And Prepare Them
    For Insert
    createWishListItems(inputData);

    // Perform Actual DML Insert Of New Wish List Items
    insertWishListItems(inputData);

    // If ccAPI.SZ_REFETCH == TRUE -> Refetch According To Passed
    In Sizing
    refetch(inputData);

    // Clean Up Return Data And Prepare It For Return
    prepReturn(inputData);

    return inputData;
}

```

Methods

processInput

Responsible for setting any default / fallback values, while also copying over the original input data back into the workingMap keyed as 'originalInputData' for the purpose of preserving the original input.

Output Parameters

Parameter Key	Type	Description
originalInputData	Map<String, Object>	A Map<> of the original input data, created in order to preserve original input values.

fetchWishList

If **ccrz.ccAPIWishList.ID** is passed in, then attempt to fetch the wish list in order to verify it is an actual valid wish list. This is accomplished by an API call to **ccrz.ccAPIWishList.fetch**.

Input Parameters

Parameter Key	Type	Description
ccrz.ccAPIWishList.ID	String	The Force.com ID of the wish list to add an item to.

fetchActive

If no **ccrz.ccAPIWishList.ID** is passed in, or if it is not valid, then this method is responsible for retrieving the current active wishlist for the user. This is accomplished by an API call to **ccrz.ccAPIWishList.getActive**.

Output Parameters

Parameter Key	Type	Description
ccrz.ccAPIWishList.ID	String	The Force.com ID of the active wish list for the current user.

fetchProduct

The fetchProduct method is responsible for retrieving all the products to add to the Wish List. This is accomplished by an API call to **ccrz.ccAPIProduct.fetch**. The fetch call is necessary for retrieving any Composite Products to ensure they are added.

Input Parameters

Parameter Key	Type	Description
ccrz.ccAPIProduct.PRODUCTSKULIST	Set<String>	A set containing the SKUs of the products to be retrieved for adding.
ccrz.ccAPIProduct.PRODUCTIDLIST	Set<String>	A set containing the Force.com IDs of the products to be retrieved for adding.

Output Parameters

Parameter Key	Type	Description
ccrz.ccApiProduct.PRODUCTLIST	List<Map<String, Object>>	A List containing the new products to add to the Wish List.

createWishListItems

Handles the sObject creation of the Wish List Items (this includes only major lines for complex products). Using the passed in **ccrz.ccApiProduct.PRODUCTLIST**, a Map of line items will be constructed and keyed by their corresponding product SFID. All new items will have a quantity set to 1 and the price will be set to 0.00 by default.

Input Parameters

Parameter Key	Type	Description
ccrz.ccAPIWishList.ID	String	The Force.com ID of the Wish List to add the new items too.
ccrz.ccApiProduct.PRODUCTLIST	List<Map<String, Object>>	A List containing the new products to create the Wish List Items from.

Output Parameters

Parameter Key	Type	Description
ccrz.ccAPIWishList.WISHLIST_ITEMS	Map<String, ccrz__E_CartItem__c>	A Map of sObjects containing the new Wish List Items with all data populated, and ready to be inserted, keyed by their corresponding products Force.com ID.

insertWishListItems

The **insertWishListItems** method is responsible for performing the actual DML INSERT on the Wish List Items. It also sets the **ccrz__ParentCartItem__c** on all minor lines to create the necessary association.

Input Parameters

Parameter Key	Type	Description
ccrz.ccAPIWishList. WISHLIST_ITEMS	Map<String, ccrz__E_CartItem__c>	A Map of sObjects containing the new Wish List Items with all data populated, and ready to be inserted, keyed by their corresponding products Force.com ID.

refetch

Method responsible for re-fetching Wish Lists via a **ccrz.ccAPIWishList.fetch** call. This is only called if **ccrz.ccAPI.SZ_REFETCH** is set to TRUE in the **ccrz.ccAPIWishList.ENTITYNAME** sizing block. By default this is set to FALSE. All passed-in sizing will be honored.

prepReturn

The prepReturn method is responsible for cleaning up any data and preparing it for its final return. For this logic class, this method is simply made available as an extension point and does not perform any action.

Override Example

The following example shows how to override the fetchProduct method to further restrict the product types allowed to be added to the Wish List.

Fetch Product Method Override

```
global virtual with sharing class myCustom_ccLogicWLAddTo extends ccrz.
ccLogicWLAddTo {

    global virtual override Map<String,Object> fetchProduct
(Map<String,Object> inputData){
        Map<String,Object> retMap = super.fetchProduct
(inputData);

        if(ccrz.ccUtil.isKeyValued(retMap, ccrz.ccApiProduct.
PRODUCTLIST)){
            List<Map<String, Object>> filteredProductList =
new List<Map<String, Object>>();

            for(Map<String, Object> product :
(List<Map<String, Object>>)retMap.get(ccrz.ccApiProduct.PRODUCTLIST)){
                if('PRODUCT'.equalsIgnoreCase((String)
product.get('productType'))){
                    filteredProductList.add
(product);
                }
            }

            // Replace PRODUCTLIST With Our Custom Filtered
Products
            retMap.put(ccrz.ccApiProduct.PRODUCTLIST,
filteredProductList);
        }

        return retMap;
    }
}
```

Component Usage

Adding to a wishlist or creating a new wishlist invokes a call to `ccApiWishList.addTo` which calls this logic class.

Add To Wishlist	Total QTY
	2 ITEM(S)
New Wish List - 2017-09-19 03:28:01	
Name	
Create Wishlist	

On the front end, the Handlebar template is cc_WishlistPicker.

Add To Wishlist	
-----------------	--