Get PriceList Hierachy

This action occurs when determining the price list to use when looking up product pricing. Note that the action name is spelled "getPriceListHierachy", not "getPriceListHierarchy".

This arc doesn’t always fire immediately after pushing a change. Sometimes you need to change something like logging in or logging out, changing a product, or just waiting 10 or 15 minutes. Additionally, if there is no default price list set in the tenant, this arc won’t fire.

>

Action Type	Embedded
Full Action ID	embedded.commerce.catalog.storefront.products.getPriceListHierachy
Runs multiple custom functions?	No

JavaScript File Structure

Action files share the following basic structure:

```
module.exports = function(context, callback) {
  // Your custom code here
  callback();
};
```

When you code the custom function for an action, you have access to two arguments:

`callback` —This argument follows the established JavaScript callback pattern: it takes an error as the first argument (or null if there is no error) and a result as the second argument (if required).

`context` —This argument provides the function access to relevant objects and methods that interface with Kibo eCommerce.

Context: PriceList

The methods and objects documented here are available to this action through the use of the `context` argument.

Microservice Operation
This action corresponds to the microservice that gets a price list.

Get

context.get.priceListHierarchy()

Returns an array of the current price list hierarchy based on the context.

Parameter	Type	Description
N/A	N/A	N/A

Example:

```
context.get.priceListHierarchy();
```

This example further shows how this call could be used in the action file. In order to set a price list, you must pass an array of price lists as the second parameter of callback(), such as *callback(null, [pricelist1, pricelist2, pricelist3, pricelist4]).* A logged in user with a customer segment will have any customer segment specific pricelist codes in this array, while a guest user will just see the default.

The following action file will attempt to locate a price list entry for the current product in pricelist1, and if it exists, it will set that as the effectivePricelist. If it fails, it will look for the entry in pricelist2, and set that as the effectivePriceList if it exists. Otherwise, it will continue to look at the next price list. A logged in user with a customer segment will have any customer segment specific pricelist codes in this array, while a guest user will just see the default.

```
module.exports = function(context, callback) {
  if (context.apiContext.purchaseLocation){
      callback ( null, [context.apiContext.purchaseLocation] );
   } else{
      callback ( null, context.get.priceListHierarchy().slice(0) );
   }
};
```

For information about the properties in the response, refer to the REST API Help.

Context Objects Available to All Actions

apiContext

Accesses Kibo eCommerce tenant information.

Property	Type	Description
baseUrl	string	The base URL for the site.
basePciUrl	string	The base PCI URL for the site.
tenantPod	string	The name of the tenant pod in which the tenant resides.
appClaims	string	The application claims token.
appKey	string	The application key.

Property	Type	Description
tenantId	integer	Unique identifier for the tenant.
siteId	integer	Unique identifier for the site. This ID is used at all levels of a store, catalog, and tenant to associate objects to a site.
masterCatalogId	integer	Unique identifier for the master catalog.
catalogId	integer	The unique identifier for the product catalog. Catalogs are part of a master catalog.
currencyCode	string	The default three-letter ISO currency code for monetary amounts.
previewDate	date/time	The date and time that the content is being viewed. This might be a future date if the content is previewed with an active date range set in the future.
localeCode	string	The locale code per the country code provided. This code determines the localized content to use and display.
correlationId	string	The unique identifier of the API request associated with the event action, which might contain multiple actions.
isAuthorizedAsAdmin	Boolean	Indicates whether the Dev Account user is authorized as an admin.
userClaims	string	The user claims token.

Example:

```
context.apiContext.baseUrl;
```

configuration

Receives a JSON response that contains information about the configuration data set in the Action Management JSON editor.

Property	Type	Description
Varies	object	Custom fields and values that you can set in the Action Management JSON Editor.

Example:

```
context.configuration.customData;
```