Checkout Groupings (After)

This action occurs when groupings need to be re-evaluated after a shopper moves to the checkout page. This can happen when: going to the checkout page for the first time, refreshing checkout from the cart, setting/updating item destinations, changing applied discounts, selecting a shipping method from a ship group, or when applying a buy one get one (BOGO) discount. Changes made to checkout items in this action do not persist in Kibo.

<table>
<thead>
<tr>
<th>Action Type</th>
<th>Embedded</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Action ID</td>
<td>embedded.commerce.carts.addItem.after</td>
</tr>
<tr>
<td>Runs multiple custom functions?</td>
<td>No</td>
</tr>
</tbody>
</table>

**JavaScript File Structure**

Action files share the following basic structure:

```javascript
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.

**Context: CheckoutGroupingsContext**

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

**Get Methods**

- `name`

**Exec Methods**

- `name.name`

**Context Objects Available to All Actions**

- `name.name`

**Get**
**get.cart**

Obtains a response that includes information about the current cart.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Example:

```java
context.get.cart();
```

Response:

For information about the properties in the response, refer to the [REST API Help](https://example.com/rest-api-help).