

Update Return (Before)

This action manipulates the HTTP request or response before the UpdateReturn operation occurs in Kibo.

| | |
|--|---|
| Action Type | HTTP |
| Full Action ID | http.commerce.returns.updateReturn.before |
| Runs multiple custom functions? | Yes |

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.

Context: HTTP

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

REST API Operation

This action corresponds to the [commerce/returns/updateReturn](#) operation.

HTTP Request

PUT `api/commerce/returns/{returnId}?responseFields={responseFields}`

Request Body

Use `context.request.body` to read/write the HTTP request body using this action.

Response Body

Use `context.response.body` to write the HTTP response body using this action.

Context Objects Available to All HTTP Actions

- [request](#)
- [response](#)

Context Methods Available to All HTTP Actions

- [request.get](#)
- [response.get](#)
- [response.set](#)
- [response.set](#)
- [response.remove](#)
- [response.redirect](#)
- [response.end](#)
- [get.resource](#)
- [get.resourceStatus](#)
- [items.urlHelper.getUrl](#)

Context Objects Available to All Actions

- [apiContext](#)
- [configuration](#)

Context Objects Available to All HTTP Actions

request

Accesses the current HTTP request. In the case of Before actions, updates can be made to the request before Kibo processes it.

| Properties | Type | Description |
|------------|--------|--|
| params | object | The input parameters to the service or webpage. |
| headers | object | The request headers. |
| method | string | The request method. |
| url | string | The request URL. |
| path | string | The request path. |
| cookies | object | The request cookies. *Available only for Storefront HTTP actions. |

| Properties | Type | Description |
|------------|---------|--|
| query | object | The request query. |
| href | string | The request href. |
| secure | Boolean | Indicates whether the request uses HTTPS. |
| ip | string | The request IP address. |
| ips | string | The request secure IP address. |
| body | object | The request body of the API operation associated with this action. |

Example:

```
context.request.url;
```

response

Accesses the current HTTP response. For both Before and After actions, updates can be made to the response before Kibo processes it.

| Properties | Type | Description |
|------------|---------|--|
| header | object | The response header collection. |
| viewData | object | The viewData collection used by the storefront rendering engine. *Available only for Storefront HTTP actions. |
| viewName | string | The response viewName value. *Available only for Storefront HTTP actions. |
| body | object | The response body of the API operation associated with this action. |
| status | integer | The response status code. |
| message | string | The response message. |
| length | integer | The response length. |
| type | string | The response type. |

Example:

```
context.response.header;
```

Context Methods Available to All HTTP Actions

request.get

Returns an HTTP header value for the specified header key.

| Parameters | Type | Description |
|------------|--------|-----------------|
| key | string | The header key. |

Example:

```
context.request.get(field);
```

Response:

```
"object"
```

response.get

Gets a field from the response.

| Parameters | Type | Description |
|------------|--------|----------------------|
| key | object | The field to obtain. |

Example:

```
context.response.get(field);
```

Response:

```
"string"
```

response.set

Sets the HTTP headers for the response.

| Parameters | Type | Description |
|------------|--------|----------------------------------|
| values | object | The values for the HTTP headers. |

Example:

```
context.response.set({ myCustomHeader: "howdy" });
```

Response: N/A

response.set2

Updates a field in the response.

| Parameters | Type | Description |
|------------|--------|-------------------------|
| key | string | The key of the field. |
| value | string | The value of the field. |

Example:

```
context.response.set("name", "stuff");
```

Response: N/A

response.remove

Removes an HTTP header from the response.

| Parameters | Type | Description |
|------------|--------|----------------------------|
| key | string | The HTTP header to remove. |

Example:

```
context.response.remove(header);
```

Response: N/A

response.redirect

Redirects the incoming URL.

| Parameters | Type | Description |
|------------|--------|----------------------|
| url | string | The destination URL. |

Example:

```
context.response.redirect("http://someOtherSite/foo");
```

Response: N/A

response.end

Ends the response so that other actions or Kibo logic can run. Also, signals the callback to complete.

| Parameters | Type | Description |
|------------|------|-------------|
| N/A | N/A | N/A |

Example:

```
context.response.end();
```

Response: N/A

get.resource

Returns the currently persisted value of the requested resource. ****Not available for all calls.**

| Parameters | Type | Description |
|------------|------|-------------|
| N/A | N/A | N/A |

Example:

```
context.get.resource();
```

Response: N/A

get.resourceStatus

Gets the HTTP resource status.

| Parameters | Type | Description |
|------------|------|-------------|
| N/A | N/A | N/A |

Example:

```
context.get.resourceStatus();
```

Response: N/A

items.urlHelper.getUrl

Gets the current URL.

| Parameters | Type | Description |
|------------|------|-------------|
| type | | |
| object | | |
| config | | |

Example:

```
context.items.urlHelper.getUrl(type, object, config);
```

Context Objects Available to All Actions

apiContext

Accesses tenant information.

| Properties | 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. |
| 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. |

| Properties | Type | Description |
|---------------------|---------|--|
| 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.

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

Example:

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