Verify External Password

If you are importing existing customer accounts to Kibo or using two customer systems simultaneously, you need to store an external password hash to migrate existing customer accounts seamlessly into Kibo, without having to require shoppers to create new passwords for the platform. Using the AddAccounts API operation, you can set the `externalPassword` property to store the password hash for the external customer account. You can then use this action, which runs during authentication, to code a function that verifies that the hashed version of a cleartext password provided by a shopper matches the external password hash for the account.

Action Type	Embedded
Full Action ID	embedded.commerce.customer.accounts.verifyExternalPassword
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.

Context: Password

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

Microservice Operation
This action corresponds to the microservice that verifies an external password.

Get Methods

- get.clearTextPassword
- get.externalPassword

Exec Methods

- exec.setSuccess

Context Objects Available to All Actions

- apiContext
- configuration

Get

get.clearTextPassword

Returns the cleartext password from an auth-token request.

Parameters	Type	Description
N/A	N/A	N/A

Example:

```
context.get.clearTextPassword();
```

Response:

```
"string"
```

get.externalPassword

Returns the external password hash stored on the customer record.

Parameters	Type	Description
N/A	N/A	N/A

Example:

```
context.get.externalPassword();
```

Response:

```
"string"
```

Exec

exec.setSuccess

Indicates to the caller that the cleartext password matches the external password value (when hashed using the same mechanism that generated the value stored in externalPassword).

Parameters	Type	Description
successValue	Boolean	True or false value. A true value prompts the calling service (Customer) to remove the externalPassword value on the customer record, hash the cleartext password, and store the new hashed value in the customer record. A false value results in an authentication failure on the originating auth-token request.

Example:

```
context.exec.setSuccess(true);
```

Response: N/A

Context Objects Available to All Actions

apiContext

Accesses Kibo eCommerce 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.

Properties	Type	Description
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.

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

Example:

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