API for Self Service

Overview

This API is used to create shopping carts in the [Self Service Channel](#) with both configurable and non-configurable products. This API is an optional feature activated with the [license](#).

All API calls require an [API key](#) that is set up in Administration at *Sales Process > Integration > Self Service*. Each call may also have additional requirements to succeed.

Some API calls can be done anonymously without specifying a reference to a shopping cart, i.e. the external entity ID (`_externalId`). Each documented call will highlight this. When required, the external entity ID can be specified as a request parameter `[apiURL]/[apiMethod]?_externalId=abc` or in a request header as `X-External-ID` .

API General Info

Response Content Type

The API only supports the `json` response type.

Successful Responses

Response Codes

- **200** - The request was executed successfully and the server returned some data. The return content type is specified inside a response header called `content-type` . Example content types are `application/xml` , `application/json` , `application/zip` and `text/plain` .
- **204** - The request was executed successfully and the server returned no data. If the `Location` header was present then the server responded with a redirect.

Error Responses

If the API call resulted in an error then the error code will be returned along with the error object in the response body. The error object contains the attributes listed below:

- **type** - type of the error
- **cause** - the root cause of the error
- **message** - message containing details of the related error
- **uuid** - a unique identifier of the error which also may appear in the exception logs
- **timestamp** - the time of the error

Note: For chunked responses, the socket will be closed prematurely if an error occurs after the initial chunk containing the status code has been sent.

Response Codes

- **400** - The request contains invalid parameters or there are invalid settings. Consult the error message and/or check Administration for invalid settings.
- **401** - The user is not authenticated.
- **403** - The user is not authorized access to the specified endpoint.
- **404** - The specified endpoint does not exist or a request parameter is invalid. Consult the error message.
- **405** - The specified endpoint does not support the requested operation. Consult the error message.
- **409** - The specified endpoint is in a state incompatible with the requested operation. Consult the error message.
- **500** - Unknown error occurred while processing the request. Consult the CPQ logs for more information.
- **504** - The specified concurrent calls limit was exceeded and the server was not able to serve the response in a specified timeout limit. That error is always served with the xml body containing the description.

API Versions

This API is versioned using the format `/self-service-api-v1.3/`.

Version	Changes	Status
/self-service-api-v1.3/	Starting the configuration with provided model needs updates the shopping cart's context attributes, otherwise compatible with version 1.2. Added support for fetching images from shopping cart item properties. Added support for finding alternatives when resolving conflicts. Added support for the `group` parameter. It holds the name of the focused configuration group, i.e. the group that is going to be displayed to the user after the call. When supplied, consistency checking is limited to parameters of the group. Supplying the parameter usually improves performance, in many cases by quite a lot.	Active
/self-service-api-v1.2/	Base version of the API since Tacton CPQ 9.0 release. Compacted shopping cart item representation, otherwise compatible with version 1.1.	Deprecated
/self-service-api-v1.1/	BoM attribute of type *Money* changed into *Price*. Otherwise compatible with version 1.	Removed
/self-service-api/	Original version of the API.	Removed

API Keys

All operations in the application require a context of an organization and role, and the API for self-service is no exception. Functions in this API require an API key which the administrator setup in advance at *Sales Process > Integration > Self Service.* It is possible to have multiple API keys registered.

All API calls require the `_key` value to be passed in the call. This can be passed in a request parameter:

```
[apiURL]/[apiMethod]?_key=XYZ
```

or in the request header as `X-KEY` . This value is verified against the key set in the basic setup.

Setting	Description
Name	Name of the key.
Key	The key to use for authentication. It needs to be provided in all API calls. The key also needs to be unique among self-service keys and keys defined for [Lead Generator](#).
Role	The role in which the API acts when authenticated with this key.
Organization	The organization in which the API acts when authenticated with this key.
Proposal Template	The proposal template used for generating draft proposals. Selected from a list of [*ProposalModel*](#) object instances (document templates using [Word templates](#) are not supported). The *Base Object* attribute needs to be set to *ShoppingCart* for the selected *ProposalModel* instance. Optional: if not set, the [generate draft proposal](#) endpoint will return 404 HTTP response.
Product Price Model	The price model used for pricing shopping cart items. The *Base Object* attribute needs to be set to *ShoppingCartItem* for the selected *PriceModel* instance. It is not possible to select between different price model views via the API. The API will return pricing data for the first view that has *Used for:* set to *Configuration Summary*.

Translations

Most endpoints support the optional query parameter `language` or request header `X-Language` . Both expect a language code as value. Translatable strings in the response will be

translated if used.

See [Languages and Translations](#) for more details.

Product Catalog

The API can access all [product catalog pages](#) that has *To Type = ShoppingCartItem*. All product catalog calls are anonymous (i.e. does not require an `_externalId`).

Any transformations on the catalog page will be executed when the shopping cart item is created.

List Product Catalog Pages

Setting	Description
Method	GET
Address	/self-service-api-v1.3/catalog/tabs
Path Parameters	N/A
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	`language` parameter or `X-Language` header (with language code).
Usage	List product catalog pages.
Exceptions	- 400: Not licensed to use the API. - 403: Unauthorized (no authentication key provided).
Return	List of product catalog pages (in JSON format) with following details: - `name` : Page name (ID). - `title` : Page title. - `type` : Type of the base object. - `isConfigurable` : True if listed products are configurable, false otherwise. - `isExternalSearch` : True if listed products are retrieved with an external search, false otherwise. - `filters` : List of page filters. - `columns` : List of page columns.

List all product catalog pages   ▼

```
/self-service-api-v1.3/catalog/tabs
```

List Products in Product Catalog Page

Setting	Description
Method	GET
Address	/self-service-api-v1.3/catalog/search/[tab]
Path Parameters	`tab` - Name (id) of the catalog page.
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	Static optional request parameters: - `language` parameter or `X-Language` header (with language code). - `limit` - Limit the number of results to the specified value. - `offset` - Offset the list of results with the specified value. - `sort` - Sort results on the specified field. - `order` - If "asc" sort the results in ascending order, otherwise descending (default behavior). Dynamic optional request parameters: - `filterName=filterValue` - List of filter values to apply for the search.
Usage	List products from the specified product catalog page, with optional search criteria.
Exceptions	- 400: Not licensed to use the API. - 403: Unauthorized (no authentication key provided). - 404: Catalog tab not found.
Return	List of products from the specified catalog page (in JSON format), matching the specified criteria, with the following details: - `total` - Number of results. - `rows` - List of products. For each specifying: - `#id` - Id of the product.

Setting	Description
	- `columnAttribute` - Values for custom attributes in the catalog page columns.

≔ List all products ▼

List all products from the catalog page named `1ecf110fe8ed49dda9b95e0739a2d58f` .

```
/self-service-api-v1.3/catalog/search/1ecf110fe8ed49dda9b95e0739a2c
```

≔ List products with filter and ascending sort ▼

List products from catalog page `1ecf110fe8ed49dda9b95e0739a2d58f` , use filter `productId=CMX` , sort results by `description` in ascending order.

```
/self-service-api-v1.3/catalog/search/1ecf110fe8ed49dda9b95e0739a2c
```

✏ Note

To access images of catalog page products you must use Image Support.

⚠ Warning

It is not possible to list products for catalog page with external search.

Configurator

All configuration calls can be executed anonymously or by providing a reference to a shopping cart, i.e. providing an external entity ID (`_externalId`). To create a shopping cart item of the configuration it is required to have an external entity ID.

The configurations calls are the same for both product models and TCX models but the content of returned data in the responses will vary depending on the type.

Anonymous configurations will return a `configState` (encoded parameter representing the current configuration state) and a `steps` structure with details on all configuration steps, groups,

and fields. In each subsequent configurator call, the current `configState` needs to be provided as a parameter and each response will provide the updated `configState`.

Summary of returned data (in JSON format) for anonymous calls:

Data	Description
`configState`	The current state of the configuration as an encoded parameter. Required in each subsequent calls.
`steps`	The list of configuration steps with a nested list of step members (groups and parameters).
`response`	The configurator [response](#) (if configuration action succeeded or not).

Configurations with an `_externalId` will be persisted and the response will return a unique `configId`. In each subsequent call, the `configId` needs to be provided as a parameter. The value of `configId` doesn't change in the scope of the same configuration and starting a new configuration will return a new unique `configId` for the `_externalId`. This allows for maintaining multiple parallel configurations for the same external entity.

Summary of returned data (in JSON format) for configuration calls of known entity:

Data	Description
`configId`	Unique ID for the configuration. It is required in each subsequent calls. This ID is not applicable after configuration has been added to cart.
`tab`	The catalog tab for the configurable product.
`productId`	The strong reference to the revision of the product being configured.
`numberOfUncommittedMandatoryFields`	The number of uncommitted [mandatory fields](#) remaining in this configuration.
`steps`	The list of configuration steps with nested list of the step members (groups and parameters).

Data	Description
response	The configurator response (if configuration action succeeded or not).
visualization	Holds the visualization object properties, if available. Can be used as the visualization object for the tactonVis JavaScript function.
pricing	The list of pricing fields and their values per BoM item and summary. The response will return the view tagged as *Configurator Result* in the product price model for the API key.
bom	The list of BoM items and their sub-items including BoM attribute values.

All configuration calls can be executed with the optional parameter `uomSystemId` (Unit System ID). If provided then the parameters with units in the generated response will have 2 new properties:

- `uomConvertedValue` has the value converted in the provided unit system.
- `uomConvertedItemId` is the effective unit item id of the unit of the calculated value. The `valueDescription` property will contain the display value converted in the provided unit system (including unit symbol if applicable).

Configuration Responses

Operation executed successfully:

```
"response": {
        "status": "OK"
},
```

Operation executed successfully but some parameter committed values were not applied:

```
"response": {
  "status": "OK",
  "not-applied": [
    {
      "name": "Rear Brake",
      "value": "ACM0023"
    }
  ],
  "message": "Some parameters were not applied"
}
```

Operation caused a resolvable conflict; the `changed` element list parameters with their current value (`oldValue`) and the new value proposed by the configuration engine (`newValue`):

```
"response": {
        "status": "RESOLVABLE",
        "changed": [
                {
                        "name": "front_brake_type_field",
                        "description": "Front Brake",
                        "oldValue": "ACM0023",
                        "oldValueDescription": "Disc Brakes S-Type",
                        "newValue": "ACM0022",
                        "newValueDescription": "V-brakes X"
                }
        ]
},
```

Alternative solutions for the conflict resolution have been requested; the `changedLists` element is a list of lists where each entry in the outer list is a list of parameters with their current value (`oldValue`) and the new value proposed by the configuration engine (`newValue`):

```
    "response": {
        "status": "RESOLVABLE",
        "changedLists": [
            [
                {
                    "name": "req_disc_brake_field",
                    "description": "Disc Brakes",
                    "oldValue": "Yes",
                    "oldValueDescription": "Yes",
                    "newValue": "No",
                    "newValueDescription": "No"
                },
                {
                    "name": "req_front_brakes_field",
                    "description": "Front Brakes",
                    "oldValue": "Yes",
                    "oldValueDescription": "Yes",
                    "newValue": "No",
                    "newValueDescription": "No"
                }
            ],
            [
                {
                    "name": "bike_size_field",
                    "description": "Bike Size",
                    "oldValue": "medium",
                    "oldValueDescription": "Medium",
                    "newValue": "small",
                    "newValueDescription": "Small"
                },
                {
                    "name": "req_front_brakes_field",
                    "description": "Front Brakes",
                    "oldValue": "Yes",
                    "oldValueDescription": "Yes",
                    "newValue": "No",
                    "newValueDescription": "No"
                }
            ]
        ]
    }
```

Operation caused non-resolvable conflict and the submitted value was not committed:

```
"response": {
        "status": "not OK"
}
```

Compatible Configuration Features

Product Models

The following configuration features are compatible with API for Self Service:

Feature	Supported
Needs	Needs are accessed as combination of groups and fields in the special needs step.
Sizing	Sizing is performed as the user steps from needs to the assembly configuration steps.
Assembly Configuration	Each top assembly is represented as a separate step, each with its set of groups and fields. The exposed auxiliary properties can be used to identify the type of field. Things to consider: - Status Message - Partially supported. Does not expose the short description. - Image - Image can be retrieved with Image from Model. - Table - Table columns should be the repeated fields within the table subgroups. - Field Detailed View - Long descriptions are not available from domain elements. Element images can be retrieved with Image from Model. - Custom Visualization Image - Requires the additional call Add Custom Visualization Image.
Options	If Options are displayed in the model view, a group of options will have auxiliary property *tc_options_module*. The value is *id* of the module containing the options. An options group can be configured with *Feature Details*. The *id* of these features are stored in property *tc_options_feature_details*. If there are multiple features, the *id* are separated by semicolon and space (;). Within each options group, there is a separate field for each option, identified by the property *tc_options_variant*. The value of this property is the *id* of the variant.
Post Calculations	Post calculations are exposed via fields showing calculated attributes. Post calculations are not supported in anonymous calls.
Module Variant Catalog	**Contact Tacton to enable this feature**. Variants for catalog positions can be set with the additional module variant catalog calls.

Feature	Supported
Info Group	The info group is exposed as a group with fields within the assembly configuration step.
CAD Automation	Models with CAD automation can use the additional CAD automation calls.
Summary	Summary is saved to the *summary* attribute on ShoppingCartItem object.
Outputs	Outputs are saved to selected attributes on the ShoppingCartItem object.
Parts	Not supported.
Mandatory fields	The number of uncommitted mandatory fields is exposed on the configuration and on every top level group. Additionally every mandatory field (committed or not) will have the isMandatory flag set to true. The type of mandatory field ('confirmation' or 'input') is exposed via the 'tc_mandatory' field property.
Notify when changed	Notify when changed fields are exposed via the 'tc_elevated' field property.

Price Lookup

For details on the feature, see Configuration Price Lookup.

When using price lookup with Options, the group of options will have an auxiliary property *tc_price_lookup_mode* with value *list*. The object `price-details` on the group will contain mappings between parameter name and the formatted price of respective variant (option).

When using the feature with fields, the field will have an auxiliary property *tc_price_lookup_mode* with either value *list* or *delta*. The object `price-details` on the field will contain mappings between domain element name and the formatted price of respective Variant.

> 🖊 **Note**
>
> The price lookup feature is not supported for anonymous configurations.

TCX Models

Feature	Supported
Step Configuration	The configuration is exposed as a set of steps, each with its set of groups and fields.

Feature	Supported
External Calculation	External calculations with base object *ShoppingCartItem* will be used.
UI Customization	Not supported, customization are not applied to the `steps` response.
Auxiliary Properties	Auxiliary properties are exposed on groups and fields. See warning below.
Module Variant Catalog	Variants for catalog positions can be set with the additional [module variant catalog calls](#).
Integration Attributes	Available for configurations saved as a shopping cart item.
Configuration Outputs	Outputs are saved to selected attributes on the ShoppingCartItem object.

> ⚠️ **Warning**
>
> Auxiliary properties is planned to be removed in future versions of API for Self Service. They will be replaced with a new set of named properties specific for this API. Avoid creating custom auxiliary properties in the TCX model only to transfer data through the API.

Needs and Sizing for Product Models

Product models which implement [Needs](#) or [Sizing](#) will receive corresponding steps in the configuration state, if input to any is required to advance in the configuration.

If a product model uses the concept of *contextual needs* then values for those needs must be set at [start of configuration](#). This is achieved by assigning an alias to the need (where needs are managed on the product model) and then providing the alias as a request parameter with the desired value.

> ≣ **Sample** ▼
>
> Assuming there are two needs defined on a product model, using aliases `region` and `size`, the following request parameters could be used to set the values for these needs:
>
> ```
> &region=europe&size=large
> ```

Shopping cart item context attributes need to be mapped for each of the needs (where needs are managed on the product model). Doing this will persist the needs values provided at start of configuration with the shopping cart item. Persisted needs are a prerequisite for reconfiguration of product models (with needs) to work.

Field Display Properties

The API for Self Service provides information on the display of each field (parameter) in the configurator call responses. This data can help craft the integration's UI to adapt to various scenarios.

The information is represented by a combination of entries in the `properties` map on each parameter in the response:

- `tc_gui_input` - indicates the expected widget to be used by the integration's UI.
- `guitype` - indicates the effective widget to be used by the integration's UI, as evaluated based on the configurator [field display rules](#).
    - Typically `tc_gui_input` (if present) and `guitype` have the same value, but overall `guitype` should take precedence.
    - In some cases when `guitype` evaluates to `readonly`, the `tc_gui_input` should indicate what read-only widget is expected to be used.
    - For TCX models, the *UI Customization* settings are not taken into account when evaluating `guitype` value.
- `hidden` - indicates if the field is hidden.
- `readonly` - indicates if the field is read-only, either based on the model constraints or the view designer customization (the evaluated *Editable If* expression).

Start Configuration

Setting	Description
Method	POST
Address	/self-service-api-v1.3/config/start/[tab]/[productId]
Path parameters	- `tab` - Catalog page (tab) name. - `productId` - ID of the product to be configured (retrieved from catalog page item list).
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `contextAttribute=contextValue` - Key value pairs for setting context needs for product models.
Other parameters	- `_externalId` - External entity ID, passed either as a request parameter or request header (`X-External-ID`). - `language` parameter or `X-Language` header (with language code). - `excludePricing` - request parameter. If set to true, then pricing calculation won't be run, response will not provide

Setting	Description
	`pricing` attribute and BoM attributes from pricing will not get any value. - `group` : the name of the focused configuration group. - [Contextual Needs]().
Usage	Start and check-out a configuration for the specified product from the catalog. After this you can perform configuration changes with the commit, uncommit and step API calls. The configuration will not be saved (and checked-in) in the shopping cart unless you perform the add to cart API call.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: Product not found
Return	[Configurator Data]()

☰ Start anonymous configuration ▼

Start an anonymous configuration for product `mbd76aa0c0eb4b6e8565579e0edbc5c2` listed in the catalog page `1ecf110fe8ed49dda9b95e0739a2d58f` .

```
/self-service-api-v1.3/config/start/1ecf110fe8ed49dda9b95e0739a2d58
```

☰ Start configuration with _externalId ▼

Start a configuration for product `m90737cadc8840d8852135424100a3e8` listed in the catalog page `1ecf110fe8ed49dda9b95e0739a2d58f` with `_externalId=d486cb2d` .

```
/self-service-api-v1.3/config/start/1ecf110fe8ed49dda9b95e0739a2d58
```

Commit a Parameter

Setting	Description
Method	POST

Setting	Description
Address	/self-service-api-v1.3/config/commit
Path Parameters	N/A
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `parameter` - Unique parameter name (URL encoded). - `value` - Value of the parameter to be committed (URL encoded).
Other Parameters	- `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `configId` - Request parameter (URL encoded). Required to continue a specific configuration started for specific external entity. - `configState` - Request parameter (URL encoded). Required to continue a configuration that started anonymously. - Additional `parameter` and `value` pairs for multi-commit. - `language` parameter or `X-Language` header (with language code). - `excludePricing` - request parameter. If set to true, then pricing calculation won't be run, response will not provide `pricing` attribute and BoM attributes from pricing will not get any value. - `group` : the name of the focused configuration group.
Usage	Commit a parameter, i.e. select a value for a field, thus preventing the engine from changing the value of that parameter.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: No data found (provided configId is not associated with any ongoing configuration for that external ID) - 404: Mandatory query parameter `parameter` / `value` not provided
Return	[Configurator Data](#)

Optional parameter for units of measure is `selectedUomItemId` , the unit of measure the provided value corresponds to. If different from the unit specified for the field domain then the value will be converted from the provided unit to the domain unit. No conversion is done if either `selectedUomItemId` is not specified or the field's domain has no connected unit item.

If several parameters are commited then `selectedUomItemId` should be provided in the respective order in the same way as `value` .

☰ Commit a parameter   ▼

Commit parameter `bike_model_field=calvin_pulse` in the configuration with `configId=6a67a3539a684647841cc8b9e8a73919` started for `_externalId=d486cb2d` .

```
/self-service-api-v1.3/config/commit?_externalId=d486cb2d&configId=
```

Example of committing several parameters at the same time.

```
/self-service-api-v1.3/config/commit?_externalId=d486cb2d&configId=
```

Commit parameter `max_weight_field=12000` in the configuration with specified `configState` .

```
/self-service-api-v1.3/config/commit?configState=*URL_ENCODED_CONF]
```

Commit several parameters.

```
/self-service-api-v1.3/config/commit?configId=6a67a3539a684647841cc
```

With the example body:

```
X-KEY: MY_KEY
X-EXTERNAL-ID: MY_EXTERNAL_ID
Content-Type: multipart/form-data; boundary=MyBoundary
--MyBoundary
Content-Disposition: form-data; name="parameter"
PaintType
--MyBoundary
Content-Disposition: form-data; name="value"
Solid
--MyBoundary
Content-Disposition: form-data; name="parameter"
Color
--MyBoundary
Content-Disposition: form-data; name="value"
green
--MyBoundary--
```

Uncommit a Parameter

Setting	Description
Method	POST
Address	/self-service-api-v1.3/config/uncommit
Path Parameters	N/A
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `parameter` - Unique parameter name (URL encoded).
Other Parameters	- `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `configId` - Request parameter (URL encoded). Required to continue a specific configuration started for specific external entity. - `configState` - Request parameter (URL encoded). Required to continue a configuration started anonymously. - `language` parameter or `X-Language` header (with language code). - `excludePricing` - request parameter. If set to true, then pricing calculation won't be run, response will not provide `pricing` attribute and BoM attributes from pricing will not get

Setting	Description
	any value. - `group` : the name of the focused configuration group.
Usage	Uncommit a parameter, i.e. unlock a field so that the engine again chooses its value.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: No data found (provided configId is not associated with any ongoing configuration for that external ID) - 404: Mandatory query parameter `parameter` not provided
Return	[Configurator Data](#)

▤ **Uncommit a parameter**   ▼

Uncommit parameter `bike_model_field` in configuration with `configId=6a67a3539a684647841cc8b9e8a73919` , started by external entity `_externalId=d486cb2d` .

```
/self-service-api-v1.3/config/uncommit?_externalId=d486cb2d&configI
```
◀ ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ ▶

Accept Conflict Resolution

Setting	Description
Method	POST
Address	/self-service-api-v1.3/config/accept
Path Parameters	N/A
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `parameter` - Unique parameter name (URL encoded). - `value` - Value of the parameter to be committed (URL encoded).
Other Parameters	- `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `configId` - Request parameter. Required to continue a

Setting	Description
	specific configuration started for specific external entity - `configState` - Request parameter (URL encoded). Required to continue a configuration started anonymously. - `language` parameter or `X-Language` header (with language code). - `excludePricing` - request parameter. If set to true, then pricing calculation won't be run, response will not provide `pricing` attribute and BoM attributes from pricing will not get any value. - `group` : the name of the focused configuration group.
Usage	• Commit a parameter and accept suggestion by the configuration engine on how to resolve conflicts: the `parameter` and `value` from the `response` data in the commit call which caused a conflict should be used in the the accept call. • Explicitly specify parameters to commit base on the provided [conflict resolution alternatives](), including the `parameter` and `value` from the `response` data in the commit call which caused a conflict.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: No data found (provided configId is not associated with any ongoing configuration for that external ID) - 404: Mandatory query parameter `parameter` / `value` not provided
Return	[Configurator Data]()

☰ Accept conflict resolution ▼

Accept parameter `front_brake_type_field=ACM0022` in the configuration with specified `configState` .

```
/self-service-api-v1.3/config/accept?configState=*URL_ENCODED_CONF]
```

☰ Accept alternative conflict resolution ▼

Accept the conflict resolution in the configuration with specified `configState` where `front_brake_type_field=ACM0022` was the commit that triggered the conflict, `bike_size_field=small` and `req_front_brakes_field=No` were one of the conflict resolution alternatives provided by the engine in the [request conflict resolution alternatives](#) call.

```
/self-service-api-v1.3/config/accept?configState=*URL_ENCODED_CONF]
```

Request Conflict Resolution Alternatives

Setting	Description
Method	POST
Address	/self-service-api-v1.3/config/alternatives
Path Parameters	N/A
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `parameter` - Unique parameter name (URL encoded) that triggered the conflict. - `value` - Value of the parameter (URL encoded) that triggered the conflict.
Other Parameters	- `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `configId` - Request parameter. Required to continue a specific configuration started for specific external entity - `configState` - Request parameter (URL encoded). Required to continue a configuration started anonymously. - `language` parameter or `X-Language` header (with language code). - `excludePricing` - request parameter. If set to true, then pricing calculation won't be run, response will not provide `pricing` attribute and BoM attributes from pricing will not get any value. - `group` : the name of the focused configuration group.
Usage	Request alternative conflict resolutions from the configuration engine. The parameter and value from the `response` data in

Setting	Description
	the commit call which caused a conflict should be used in this call.
Exceptions	- 400: Not licensed to use the API - 400: Finding alternatives is not supported for multi-commit actions - 403: Unauthorized (no authentication key provided) - 404: No data found (provided configId is not associated with any ongoing configuration for that external ID) - 404: Mandatory query parameter `parameter` / `value` not provided
Return	[Configurator Data](#)

≔ **Request conflict resolution alternatives** ▼

Request conflict resolution alternatives for commit `front_brake_type_field=ACM0022` in configuration with `configId=6a67a3539a684647841cc8b9e8a73919`, started by external entity `_externalId=d486cb2d`.

```
/self-service-api-v1.3/config/alternatives?_externalId=d486cb2d&cor
```
◄ ▬▬▬▬▬▬▬▬▬▬▬ ►

✎ **Note**

If the provided `parameter` and `value` don't trigger a configuration conflict, the submitted `value` will be committed.

Change Step

Setting	Description
Method	POST
Address	/self-service-api-v1.3/config/step
Path Parameters	N/A
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).

Setting	Description
	- `step` - Name of the step, as listed in the 'steps' list element of the configuration response (URL encoded).
Other Parameters	- `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `configId` - Request parameter (URL encoded). Required to continue a specific configuration started for specific external entity. - `configState` - Request parameter (URL encoded). Required to continue a configuration started anonymously. - `language` parameter or `X-Language` header (with language code). - `excludePricing` - request parameter. If set to true, then pricing calculation won't be run, response will not provide `pricing` attribute and BoM attributes from pricing will not get any value. - `group` : the name of the focused configuration group.
Usage	Navigate the configuration to the specified step.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: No data found (provided configId is not associated with any ongoing configuration for that external ID) - 404: Mandatory query parameter `step` not provided
Return	Configurator Data

≡ **Change configuration step** ▼

Navigate to `step=DetailedConfigurationStep` in the configuration with `configId=36b96a6ceb3c4e6c95255ac204e462be` started for `_externalId=d486cb2d` .

```
/self-service-api-v1.3/config/step?_externalId=d486cb2d&configId=36
```
◀ ▶

Cancel Configuration

Setting	Description
Method	POST
Address	/self-service-api-v1.3/config/cancel
Path Parameters	N/A
Required Parameters	`configId` - Request parameter (URL encoded).
Other Parameters	N/A
Usage	Cancel the given configuration. All configuration changes will be discarded.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: No data found (provided configId is not associated with any ongoing configuration for that external ID)
Return	JSON object with `status=ok` string.

≣ **Cancel a configuration** ▼

Cancel an ongoing configuration with `configId=8c2930359fc94324989750abd7409c1c` .

```
/self-service-api-v1.3/config/cancel?configId=8c2930359fc943249897
```

Save Configuration

See [Add Item to Cart](#).

Restart Configuration

Also known as reconfigure or reconfiguration.

Setting	Description
Method	POST
Address	/self-service-api-v1.3/config/start

Setting	Description
Path Parameters	N/A
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-External-ID`). - `configId` - Request parameter (URL encoded). Required to continue a specific configuration started for specific external entity.
Other Parameters	- `language` parameter or `X-Language` header (with language code). - `group` : the name of the focused configuration group.
Usage	Choose a shopping cart item for which to perform reconfiguration. After this API call you can continue performing configuration changes on the saved configuration using commit, step, etc. The new changes will not be saved in the shopping cart item unless you save them using the ([Add Item to Cart](#)) API call.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: Required parameters not provided - 404: No data found - 404: Cannot reconfigure unsaved configuration (will happen when attempting the reconfigure call on a configuration that has not yet been added to the Cart)
Return	[Configurator Data](#)

≡ **Reconfigure shopping cart item** ▼

Reconfigure a *ShoppingCartItem* with
`configId=8c2930359fc94324989750abd7409c1c` .

```
/self-service-api-v1.3/config/start?configId=8c2930359fc9432498975(
```
◄ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ►

⚠ **Warning**

> Starting the reconfiguration of the shopping cart item creates a temporary version of that item. That temporary version will be returned when listing cart items, until the reconfiguration is finished or cancelled. See Temporary Versions for more details.

Get Bill of Materials

Setting	Description
Method	GET
Address	/self-service-api-v1.3/config/bom
Path Parameters	N/A
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	- `_externalId` - Request parameter. Required to get BoM for configuration started for specific external entity. - `configId` - Request parameter (URL encoded). Required to get BoM for configuration started for specific external entity. - `configState` - Request parameter (URL encoded). Required to get BoM for configuration started anonymously. - `language` parameter or `X-Language` header (with language code).
Usage	Get the bill of materials for the current anonymous configuration. This operation should only be used during the configuration session.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 400: Mandatory query parameters not provided - 404: No data found (provided configId is not associated with any ongoing configuration for that external ID)
Return	Configurator Data

> ≔ **Get bill of materials** ▼
>
> Get BoM for the configuration with specified `configState` .

```
/self-service-api-v1.3/config/bom?configState=*URL_ENCODED_CONFIG_S
```

Get BoM for the `configId=121b7ffd3085460faacf0a6796102105` started for `_externalId=d486cb2d`.

```
/self-service-api-v1.3/config/bom?configId=121b7ffd3085460faacf0a67
```

Get Visualization

Setting	Description
Method	POST
Address	/configurator-api/visualization
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `configState` - Request parameter (URL encoded). Required to continue a configuration that started anonymously.
Other Parameters	- `group` : the name of the focused configuration group.
Usage	Returns a visualization object for the current state of the configuration.
Return	The visualizationObject in JSON format to be passed on to the Visualization.

Get visualization for configState ▼

Get the visualization object for the current configuration.

```
/configurator-api/visualization?_key=abc123&configState=H4sIAAAAAA
```

Get Module Variant Catalogs

Setting	Description
Method	GET
Address	/self-service-api-v1.3/config/variant-catalogs
Path Parameters	N/A
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	- `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `configId` - Request parameter (URL encoded). Required to access module variant catalogs for a configuration. - `language` parameter or `X-Language` header (with language code).
Usage	Get all module variant catalogs for a configuration.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: No data found (provided configId is not associated with any ongoing configuration for that external ID)
Return	JSON object with `catalogs` (incl. `pages`), `variantBom` , `positions` , `columns` , `domains` , and `filters` .

▤ **Get module variant catalog for configuration** ▼

Get all module variant catalogs for configuration with
`configId=36b96a6ceb3c4e6c95255ac204e462be` and `_externalId=d486cb2d` .

```
/self-service-api-v1.3/config/variant-catalogs?_externalId=d486cb2d
```

◀ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ▶

Get Variants from Module Variant Catalog Page

Setting	Description
Method	GET

Setting	Description
Address	/self-service-api-v1.3/config/variant-catalogs/[catalogid]/pages/[pageid]/positions/[positionid]/variants
Path Parameters	- `[catalogid]` - Id of the module variant catalog. - `[pageid]` - Id of the variant catalog page. - `[positionid]` - Id of the position realized by the module variant catalog.
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	- `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `configId` - Request parameter (URL encoded). Required to access module variant catalogs for a configuration. - `language` parameter or `X-Language` header (with language code). - `page` - Page 1 for first 100 results. Page 2 for 101-200 etc.
Usage	Get the first 100 variants for a module variant catalog page (without variable filters).
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: No data found (provided configId is not associated with any ongoing configuration for that external ID)
Return	JSON object with `filters` and `items` .

≡ **Get module variant catalog for configuration** ▼

Get all variants from `catalogId=id1` , `pageid=id2` , `positionId=id3` where `configId=36b96a6ceb3c4e6c95255ac204e462be` and `_externalId=d486cb2d` .

```
/self-service-api-v1.3/config/variant-catalogs/id1/pages/id2/positi
```
◀ ━━━━━━━━━━━━━━ ▶

✏️ **Filters**

Locked filters (`lockedFilterIds`) are automatically applied when returning variants for a page. Suggested filters (`suggestedFilterIds`) are currently not possible to send

in the GET call. The alternative is to do this filtering in the external web application.

Add Variant to Variant Catalog BoM

Setting	Description
Method	POST
Address	/self-service-api-v1.3/config/variant-bom
Path Parameters	N/A
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `items` - Post body with list of items.
Other Parameters	- `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `configId` - Request parameter (URL encoded). Required to access module variant catalogs for a configuration. - `language` parameter or `X-Language` header (with language code).
Usage	Add variants to variant catalog BoM. To modify a BoM, repeat the call with changed quantities and variants. Use *columnValues* to set a value for undefined global features.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: No data found (provided configId is not associated with any ongoing configuration for that external ID)
Return	JSON object with `items` stripped from non-editable column values.

Add one variant to variant catalog BoM ▼

Add one variant to the variant catalog BoM for `configId=36b96a6ceb3c4e6c95255ac204e462be` and `_externalId=d486cb2d` .

```
/self-service-api-v1.3/config/variant-bom?_externalId=d486cb2d&conf
◀ ━━━━━━━━━━━━━ ▶
```

With the example body:

```
{
"items": [
{
"variantId": "Y2RkMjU2MDgtZTAxMS00MTcyLWI4YjAtMzVlNjg4YWZkODJk.Zjcz
"quantity":1,
"columnValues": {
"30ee3a2b-4d4d-44c0-96dd-6f7dc5fb871c": "M-0001-A",
"6a390e0a-ac5a-4eea-b87d-ed2440010543": "1.5 kW Special Motor",
"caa38fb9-1f69-4eb7-b77d-124f04a9ea01": "Yes",
"b1751e191ef344c5a9f84bf5a713f48b": "Red"
}
}
]
}
```

> ✏️ **Note**
>
> Position limits regarding variant BoM (max quantity sum, max quantity per variant, max different variants) must be applied by the external web application using the API. Position limits are retrieved with the call get module variant catalogs.

> ✏️ **Note**
>
> If a variant has any undetermined values for global features (e.g. *unspecified* or multi-value) then you must use *columnValues* in the POST call to set those features. Otherwise, the system will randomly pick a value. If you try to modify a feature that is already determined (i.e. it has a single value) then it will be ignored.

Shopping Cart

There can exist only one active shopping cart for specific `_externalId` . Any operation (regardless of the HTTP method) that requires the shopping cart will create a new, empty shopping cart if no active one exists. Submitting a Request for Firm Proposal will effectively discard the active cart.

All shopping cart API operations require external entity ID (`_externalId`) parameter.

List Complete Cart Information

Setting	Description
Method	GET
Address	/self-service-api-v1.3/cart/complete

Setting	Description
Path parameters	N/A
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`).
Other parameters	N/A
Usage	Get the complete shopping cart with all its items. Including item BoM (with parts and BoM properties) and the IPA with costing breakdown.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided)
Return	JSON object with two top-level elements, `ShoppingCart` object and `ShoppingCartItems` list.

> 🖊️ **Note**
>
> When the optional parameter `includeUploadedProductConfiguration` is set to `true` and an uploaded TCX model is used, the product-configuration section will include additional information, if available.

≔ Get all cart details ▼

Show complete cart information, with all its items details, for shopping cart with `_externalId=d486cb2d` .

```
/self-service-api-v1.3/cart/complete?_externalId=d486cb2d
```

List Items in Cart

Setting	Description
Method	GET

Setting	Description
Address	/self-service-api-v1.3/cart/items
Path Parameters	N/A
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`).
Other Parameters	`language` parameter or `X-Language` header (with language code). `returnAllShoppingCartItems` - When included and set to `true` , all items in the cart will be returned, regardless of their state.
Usage	List content of the shopping cart.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter or header not provided (`_externalId` or `X-External-ID`). - 403: Unauthorized (no authentication key provided).
Return	List of items in the shopping cart (in JSON format), with following details: - `itemCount` - Number of shopping cart items. - `items` - The list of items. Properties per item: - `#id` - The shopping cart item ID. Required to reconfigure the item. - `product` - The name of the product (same as in product catalog page). - `productId` - The id of the product (same as in product catalog page). - `productReference` - The strong reference to the version of the product being configured. - `tab` - The product catalog page for the product. - `isConfigurable` - True if this product is configurable, false otherwise. - `isProductOutdated` - True if the model has been updated for the configurable product, false otherwise. - `isCurrencyOutdated` - True if the selected currency for the shopping cart item has been updated, false otherwise. - `isAnyCurrencyRateOutdated` - True if currency rates have

Setting	Description
	been changed and item uses costs.

been changed and item uses costs.
- `isPriceOutdated` - True if any of `isPriceModelOutdated`, `isPriceDatatableOutdated`, and `isPriceContextOutdated` is true, false otherwise.
- `isPriceModelOutdated` - True if the product price model has been updated, false otherwise.
- `isPriceDatatableOutdated` - True if any data table used by the product price model has been updated, false otherwise.
- `isPriceContextOutdated` - True if any contextual attribute used by the product price model, including conversion factors of currency converted data table columns, has been updated, false otherwise.
- `isCheckedOut` - True if the shopping cart item is undergoing modifications, mostly happens when the configuration has not been finished yet, false otherwise.
- `lifeCycleState` - Life Cycle Status if the shopping cart item object.
- `hasAnyUncommittedMandatoryFields` - True if the configurable product has any uncommitted [mandatory fields](#) remaining, false otherwise.
- `pricing` - The list of pricing fields and their values per BoM item and summary. The response will return the view tagged as *Configurator Result* in the product price model for the API key.
- `summary` - The summary of the shopping cart item, based on the catalog transformation to `summary` attribute.
- `qty` - The quantity of the shopping cart item.
- Any custom attributes on the ShoppingCartItem object.
- `configResult` - Containing the following data:
- `safecookie` - The configuration safe-cookie of how commits were applied.
- `requiredParameterList` - The list of all configuration parameters.
- `bom` - The list of all BoM items and their sub-items.
- `integrationAttributes` - The list of non-empty integration attributes.
- `resultLayout` - A deprecated list of steps and their members.

⁝☰ List shopping cart items ▼

List items in the cart with `_externalId=d486cb2d`.

```
/self-service-api-v1.3/cart/items?_externalId=d486cb2d
```

⚠️ **Warning**

If a shopping cart item has been added to the cart and then the reconfiguration of that item has been started, a temporary version of that item exists. That temporary version will be returned when listing cart items, until the reconfiguration is finished or cancelled. See [Temporary Versions](#) for more details.

Add Item to Cart

Setting	Description
Method	POST
Address	/self-service-api-v1.3/cart/items
Path Parameters	N/A
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`) - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). Changing the external ID will effectively add the item to the other cart (creating it if necessary).
Other Parameters	Configuration with `configId` : - `configId` - Required request parameter. See [Start Configuration](#). - `qty` - Optional request parameter. Specifies the quantity of the product to be added. Anonymous configuration without `configId` : - `configState` - Required request parameter (URL encoded). See [Start Configuration](#). - `tab` - Required request parameter. Name of the catalog tab where the product is listed. - `productId` - Required request parameter. Unique ID of the item to be added. See [List Catalog Items](#). Item must be of type "Product". - `qty` - Optional request parameter. Specifies the quantity of the product to be added.

Setting	Description
	Non-configurable product : - `tab` - Required request parameter. Name of the catalog tab where the product is listed. - `productId` - Required request parameter. Unique ID of the item to be added. See [List Catalog Items](). Item can't be of type "Product". - `qty` - Optional request parameter. Specifies the quantity of the product to be added. - `language` parameter or `X-Language` header (with language code).
Usage	Add an item to the cart, either a configured product or a non-configurable product. For configurable products this call also automatically checks-in the configuration.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided) - 404: Product does not exist in the catalog - 404: Product needs to be configured before adding it to cart - 404: Insufficient data provided for the operation to complete (in most cases means that not all parameters from specific use case as listed above in 'Other parameters' were provided).
Return	Details of the newly added item (in JSON format) as defined in [List Items in Cart]().

:=≡ **Add shopping cart item** ▼

Add to cart with `_externalId=d486cb2d` a product that has been configured and the returned `configId=75f4a85b4597426085bdd91aff4069e2` .

```
/self-service-api-v1.3/cart/items?_externalId=d486cb2d&configId=75f
◄ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ►
```

Add to cart `_externalId=d486cb2d` a non-configurable product with `productId=m5d44cfa703c4e958cc145f70535bdea` , listed in `tab=valve` in `qty=10` .

```
/self-service-api-v1.3/cart/items?_externalId=d486cb2d&tab=valve&pr
```
◀ ━━━━━━━━━━━━━━━━━━━━━━━━ ▶

Add Item from External Search to Cart

A shopping cart item can be created from an externally sourced product. One way to list these types of products is to utilize a product catalog tab with external search enabled (`/self-service-api-v1.3/catalog/search/[tab]`). The custom application can also retrieve this list through other mechanisms outside Tacton CPQ.

The POST call to `/self-service-api-v1.3/cart/items` expects an `_externalId`, `tab`, and a body with attribute values for the external product. The list of attributes needs to match the object attributes used for the product catalog `tab`. They are required to succeed with any transformations specified on the product catalog tab.

> ☰ **Add an item from an external search result** ▼
>
> Add shopping cart item based on external search result with `_externalId=d486cb2d` and `tab=08431bd63c2440979fa803ef28bc0200` .
>
> ```
> /self-service-api-v1.3/cart/items?_externalId=d486cb2d&tab=08431bd6
> ```
> ◀ ━━━━━━━━━━━━━━━━━━━━━━ ▶
>
> With the example body:
>
> ```
> {
> "level": "1",
> "image": null,
> "description": "Coffee Mug",
> "name": "abc123",
> "ItemID": "abc123",
> "variantCatalogs": null
> }
> ```

Update Item in Cart

Setting	Description
Method	PUT
Address	/self-service-api-v1.3/cart/items/[shoppingCartItemId]

Setting	Description
Path Parameters	`shoppingCartItemId` - Unique ID of the cart item to be updated.
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	- `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). Assigning a new external id will move the item to the new cart. - `qty` - Quantity of the item in the shopping cart. - `language` parameter or `X-Language` header (with language code). - `customObjectAttribute` - Any custom attribute of the ShoppingCartItem object. Some internal parameters are restricted and an attempt to update them will result in an error. Those are: searchLayout, shoppingCar, json, pricingJson, tempPricingJson, productId, and product. - `language` parameter or `X-Language` header (with language code).
Usage	Update an existing shopping cart item. To reconfigure an existing shopping cart item, see [Reconfigure]().
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided) - 404: No data found - 404: Attempt to update a restricted parameter
Return	Details of the updated item (in JSON format).

≡ **Update quantity for shopping cart item** ▼

Update shopping cart item `65af4b9ae10e44dd8c1b9ffb5d28c51e` with `qty=10` .

```
/self-service-api-v1.3/cart/items/65af4b9ae10e44dd8c1b9ffb5d28c51e?
```
◀ ▬▬▬▬▬▬▬▬ ▶

Delete Item from Cart

Setting	Description
Method	DELETE
Address	/self-service-api-v1.3/cart/items/[shoppingCartItemId]
Path Parameters	`shoppingCartItemId` - Unique ID of the cart item to be updated.
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`).
Other Parameters	N/A
Usage	Delete an item from the shopping cart.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided) - 404: No data found
Return	Success message.

> ≔ **Delete shopping cart item** ▼
>
> Delete item `06b76addbde5427a88acc724c9905af2` from shopping cart with `_externalId=d486cb2d` .
>
> ```
> /self-service-api-v1.3/cart/items/06b76addbde5427a88acc724c9905af2?
> ```
> ◄ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ►

List Cart Attributes

Setting	Description
Method	GET
Address	/self-service-api-v1.3/cart
Path parameters	N/A

Setting	Description
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`).
Other parameters	N/A
Usage	Read cart attributes as object key-value pairs.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided)
Return	Cart object.

☰ **Get attributes values for shopping cart** ▼

Show attributes from shopping cart with `_externalId=d486cb2d` .

```
/self-service-api-v1.3/cart?_externalId=d486cb2d
```

Update Cart Attributes

Setting	Description
Method	PUT
Address	/self-service-api-v1.3/cart
Path Parameters	N/A
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`).
Other Parameters	The following parameters are supported: - `customObjectAttribute` - Any custom attribute of the ShoppingCart object. Some internal attributes are restricted and

Setting	Description
	an attempt to update them will result in an error. Those are: externalId.
Usage	Update Cart attributes.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided) - 404: Attempt to update a restricted parameter
Return	Updated cart object.

Update currency for shopping cart ▼

Update attribute `currency=PLN` in shopping cart with `_externalId=d486cb2d`.

```
/self-service-api-v1.3/cart?_externalId=d486cb2d&currency=PLN
```

Delete Cart

Setting	Description
Method	DELETE
Address	/self-service-api-v1.3/cart
Path Parameters	N/A
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`).
Other Parameters	N/A
Usage	Delete shopping cart. If a request for firm proposal has been submitted, the cart cannot be deleted.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided:

Setting	Description
	_externalId/X-External-ID - 403: Unauthorized (no authentication key provided) - 404: Attempt to update a restricted parameter
Return	Success message.

> ☰ **Delete shopping cart** ▼
>
> Delete shopping cart with `_externalId=d486cb2d` .
>
> ```
> /self-service-api-v1.3/cart?_externalId=d486cb2d
> ```

Recalculate Prices

Setting	Description
Method	POST
Address	/self-service-api-v1.3/cart/recalculate-prices
Path Parameters	N/A
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`).
Other Parameters	`updateDatatables` - If present and true, the data tables are updated to their latest version. This parameter is only relevant if data tables are retained in pricing settings.
Usage	Recalculate prices of all cart items and the cart itself. Used for example when pricing model has changed.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided)
Return	Updated cart object.

Refresh Currency

Setting	Description
Method	POST
Address	/self-service-api-v1.3/cart/refresh-currency
Path Parameters	N/A
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`).
Other Parameters	N/A
Usage	Refresh the used currency for the cart and all the cart items. Used for example when the currency conversion factor has changed or when the currency was changed. This method also recalculates costs on all cart items when currency rates have been changed (indicated by the `"isAnyCurrencyRateOutdated": true` flag on the [cart item](#), only applicable if the cart item uses costs).
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided)
Return	Updated cart object

```
/self-service-api-v1.3/cart/refresh-currency?_externalId=d486cb2d
```

Upgrade Shopping Cart Items

Setting	Description
Method	POST
Address	/self-service-api-v1.3/cart/upgrade-products
Path Parameters	N/A
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`).
Other Parameters	N/A
Usage	Start a batch job for upgrading all configurable items in the cart with the latest revisions of the product models. The job is executed asynchronously and its status can be checked by calling [Check product upgrade job status](). To check if product needs update you can use `isProductOutdated` when listing shopping cart items.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided)
Return	JSON object with `message` and `state` elements, where message is a detailed description of the job state and state can be one of: - `pending` - The batch job has been submitted and is pending execution. - `no-run` - No shopping cart items with outdated product found in the shopping cart.

Upgrade configurable shopping cart items ▼

Upgrade configurable products for shopping cart with `_externalId=d486cb2d` .

```
/self-service-api-v1.3/cart/upgrade-products?_externalId=d486cb2d
```

Check Product Upgrade Job Status

Setting	Description
Method	GET
Address	/self-service-api-v1.3/cart/upgrade-products
Path Parameters	N/A
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`).
Other Parameters	N/A
Usage	Check a status of a batch job for upgrading all configurable items in the cart with the latest revisions of the product models.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided)
Return	JSON object with `message` and `state` elements, where message is a detailed description of the job state and state can be one of: - `pending` - The job is pending execution. - `running` - The job is currently running. - `not-found` - No active product upgrade jobs found for the shopping cart.

> 📋 **Example** ▼
>
> Check status of the job for upgrading configurable products in shopping cart with `_externalId=d486cb2d` .

```
/self-service-api-v1.3/cart/upgrade-products?_externalId=d486cb2d
```

Proposal

Generate Draft Proposal

Setting	Description
Method	GET
Address	/self-service-api-v1.3/proposal/draft
Path parameters	N/A
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - external entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`).
Other parameters	`language` parameter or `X-Language` header (with language code).
Usage	Generate and download the draft proposal for items in the shopping cart.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided) - 404: Proposal generation has not been set up by the Administrator - 500: Server encountered error when generating document
Return	PDF file containing draft proposal.

≔ Generate draft proposal ▼

Generate draft proposal for shopping cart with `_externalId=d486cb2d` .

```
/self-service-api-v1.3/proposal/draft?_externalId=d486cb2d
```

Request for Firm Proposal

Setting	Description
Method	POST
Address	/self-service-api-v1.3/proposal/firm-requests
Path Parameters	N/A
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - All mandatory attributes defined for object RequestForFirmProposal.
Other Parameters	N/A
Usage	Create a request for firm proposal from the direct sales channel (resolved manually by a direct sales user). This operation will effectively discard an active shopping cart. See Shopping Cart for details.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided)
Return	Empty success message.

> ≔ **Request firm proposal** ▼
>
> Request firm proposal for shopping cart with `_externalId=d486cb2d` .
>
> ```
> /self-service-api-v1.3/proposal/firm-requests?_externalId=d486cb2d
> ```

List Requests for Firm Proposal

Setting	Description
Method	GET

Setting	Description
Address	/self-service-api-v1.3/proposal/firm-requests
Path Parameters	N/A
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`).
Other Parameters	N/A
Usage	List all requests for firm proposal in the direct sales channel containing status and last modified date.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided)
Return	Success message.

List requests for firm proposal ▼

List requests for firm proposal with `_externalId=d486cb2d` .

```
/self-service-api-v1.3/proposal/firm-requests?_externalId=d486cb2d
```

Documents

Add Attachment to Shopping Cart

Setting	Description
Method	POST
Address	/self-service-api-v1.3/cart/attachment
Path Parameters	N/A

Setting	Description
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `file` - Attachment file can only be uploaded using multipart/form-data.
Other Parameters	N/A
Usage	Add an attachment to the cart.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 400: Mandatory parameter not provided: file - 403: Unauthorized (no authentication key provided)
Return	Details of the newly added attachment (in JSON format).

≣ Add attachment to shopping cart ▼

Add an attachment to shopping cart with `_externalId=d486cb2d` .

```
/self-service-api-v1.3/cart/attachment?_externalId=d486cb2d
```

Add Attachment to Shopping Cart Item

Setting	Description
Method	POST
Address	/self-service-api-v1.3/cart/items/[shoppingCartItemId]/attachment
Path Parameters	`shoppingCartItemId` - Unique ID of the cart item to be updated.
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`).

Setting	Description
	- `file` - Attachment file can only be uploaded using multipart/form-data.
Other Parameters	N/A
Usage	Add an attachment to the cart item.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 400: Mandatory parameter not provided: file - 403: Unauthorized (no authentication key provided)
Return	Details of the newly added attachment (in JSON format).

≣ **Add attachment to shopping cart item** ▼

Add an attachment to shopping cart item `a1e3e3a1778a4cccb90d7f4fb0b8ccbc` for shopping cart with `_externalId=d486cb2d` .

```
/self-service-api-v1.3/cart/items/a1e3e3a1778a4cccb90d7f4fb0b8ccbc/
```
◄ ═══════════════════════════════ ►

Delete Shopping Cart Attachment

Setting	Description
Method	DELETE
Address	/self-service-api-v1.3/cart/attachment
Path Parameters	N/A
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `attachmentId` - Unique ID of the attachment. This value can be retrieved from the listing of attachments.

Setting	Description
Other Parameters	N/A
Usage	Delete attachment from the shopping cart.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 400: Mandatory parameter not provided: attachmentId - 403: Unauthorized (no authentication key provided)
Return	Success message.

‚áú‚â° **Delete shopping cart attachment** ‚ñº

Delete `attachmentId=0ac3c7fe4b9d47f7a8a1130c486cc110` from shopping cart with `_externalId=d486cb2d` .

```
/self-service-api-v1.3/cart/attachment?_externalId=d486cb2d&attachm
```

Delete Shopping Cart Item Attachment

Setting	Description
Method	DELETE
Address	/self-service-api-v1.3/cart/items/[shoppingCartItemId]/attachment
Path Parameters	`shoppingCartItemId` - Unique ID of the cart item to be updated.
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `attachmentId` - Unique ID of attachment. This value can be retrieved from the listing of attachments.
Other Parameters	N/A

Setting	Description
Usage	Delete attachment from the shopping cart item.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 400: Mandatory parameter not provided: attachmentId - 403: Unauthorized (no authentication key provided)
Return	Success message.

≔ **Delete shopping cart item attachment** ▼

Delete `attachmentId=0ac3c7fe4b9d47f7a8a1130c486cc110` from item `0f6f517b6c224b2eb1fea1b93974fe40` in shopping cart with `_externalId=d486cb2d` .

```
/self-service-api-v1.3/cart/items/0f6f517b6c224b2eb1fea1b93974fe40/
```
◄ ▬▬▬▬▬▬▬▬▬ ►

List Shopping Cart Documents

Setting	Description
Method	GET
Address	/self-service-api-v1.3/cart/document/list
Path Parameters	N/A
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`).
Other Parameters	`language` - Optional request parameter. The language to list documents for. Only applicable for dynamic documents. Defaults to `en` if not specified.
Usage	Retrieve all documents associated with the shopping cart, data excluded. Note: any available documents for related Shopping Cart Items will also be included in the list.

Setting	Description
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided)
Return	Document list in JSON format.

List Shopping Cart Item Documents

Setting	Description
Method	GET
Address	/self-service-api-v1.3/cart/items/[shoppingCartItemId]/document/list
Path Parameters	`shoppingCartItemId` - Unique ID of the cart item to be updated.
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`).
Other Parameters	`language` - Optional request parameter. The language to list documents for. Only applicable for dynamic documents. Defaults to `en` if not specified.
Usage	Retrieve all documents associated with the shopping cart item, data excluded.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided)
Return	Document list in JSON format.

Download Shopping Cart Document

Setting	Description
Method	GET
Address	/self-service-api-v1.3/cart/document/download

Setting	Description
Path Parameters	N/A
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `documentType` - Document resource type. - `documentId` - Document resource ID.
Other Parameters	`language` - Optional request parameter. The language to download documents for. Only applicable for dynamic documents. Defaults to `en` if not specified. - `unitSystemNameOrId` - The name or id of the UoM system to download the document for. Only applicable for dynamic documents. The latest version of the UoM system will be used.
Usage	Download a document.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided)
Return	The document.

Download Shopping Cart Item Document

Setting	Description
Method	GET
Address	/self-service-api-v1.3/cart/items/[shoppingCartItemId]/document/download
Path Parameters	`shoppingCartItemId` - Unique ID of the cart item to be updated.
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - external entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `documentType` - Document resource type. - `documentId` - Document resource ID.

Setting	Description
Other Parameters	`language` - Optional request parameter. The language to download documents for. Only applicable for dynamic documents. Defaults to `en` if not specified. - `unitSystemNameOrId` - The name or id of the UoM system to download the document for. Only applicable for dynamic documents. The latest version of the UoM system will be used.
Usage	Download a document.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided)
Return	The document.

Download all Shopping Cart Documents

Setting	Description
Method	GET
Address	/self-service-api-v1.3/cart/document/download-all
Path Parameters	N/A
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `documentCategory` - Document category to download. A category of each document is included in the list documents response.
Other Parameters	`language` - Optional request parameter. The language to download documents for. Only applicable for dynamic documents. Defaults to `en` if not specified. - `unitSystemNameOrId` - The name or id of the UoM system to download documents for. Only applicable for dynamic documents. The latest version of the UoM system will be used.
Usage	Download all shopping cart documents of the specified category as one zip file.

Setting	Description
Exceptions	- 400: Not license to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided)
Return	The compressed archive of documents.

Download all Shopping Cart Item Documents

Setting	Description
Method	GET
Address	/self-service-api-v1.3/cart/items/[shoppingCartItemId]/document/download-all
Path Parameters	`shoppingCartItemId` - Unique ID of the cart item to be updated.
Required parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `documentCategory` - The document category to download. A category of each document is included in the list documents response.
Other Parameters	`language` - Optional request parameter. The language to download documents for. Only applicable for dynamic documents. Defaults to `en` if not specified. - `unitSystemNameOrId` - The name or id of the UoM system to download documents for. Only applicable for dynamic documents. The latest version of the UoM system will be used.
Usage	Download all Shopping Cart Item documents of the specified category as one zip file.
Exceptions	- 400: Not licensed to use the API - 400: Mandatory parameter/header not provided: _externalId/X-External-ID - 403: Unauthorized (no authentication key provided)
Return	The compressed archive of documents.

Image Support

All image API calls are anonymous.

Image From Object Attribute

Setting	Description
Method	GET
Address	/self-service-api-v1.3/[type]/[id]/image/[property]
Path Parameters	- `type` - The object type. - `id` - ID of the object instance. - `property` - Name of the object attribute (property) holding the image or zipped resources.
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	`file` - Request parameter. Indicates a specific file name if the accessed property is zipped resources.
Usage	Images that are properties on a *Product* instance, either directly (*Image* attribute) or as part of a zipped resources (*Binary* attribute). The typical use case for uploaded TCX models.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 403: Resource not accessible - 404: Image not found
Return	Image file with respective Content-Type.

Get image from attribute ▼

Get image from the attribute `thumbnail` on the product instance with id `m90737cadc8840d8852135424100a3e8` .

```
/self-service-api-v1.3/Product/m90737cadc8840d8852135424100a3e8/ima
```

Get image from resource zip ▼

Get the `file=logo.png` from the attribute `resources` on the product instance with id `m90737cadc8840d8852135424100a3e8` .

```
/self-service-api-v1.3/product/m90737cadc8840d8852135424100a3e8-b52
```

Image From Product Model

Setting	Description
Method	GET
Address	/self-service-api-v1.3/product/[productId]/image/[section]/[sectionId]
Path Parameters	- `productId` - ID of the *Product* object instance. - `section` - Identifier of the model section. Supported values are: tab, group, field, field-view (the image displayed in the detailed view), enumerated domain ID, module ID. - `sectionId` - ID of the model view section. If the section is an enumerated domain ID, this needs to be an ID of an element from that domain. If the section is a module ID, this needs to be an ID of a variant of that module.
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	N/A
Usage	Images in the design view for product models created in Administration. Those images are not tied to any specific attribute, but instead to a specific section of the model view (tab, group, or field) or to a specific enumerated domain / module (for component pictures).
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 403: Resource not accessible - 404: Image not found
Return	Image file with respective Content-Type. If the path parameters point to a valid combination of enumerated domain ID / element ID or module ID / variant ID but a respective component has no image uploaded in the product modeling, a default 'no picture found' image will be returned (PNG file, 100x75 px).

Image From Shopping Cart Item

Setting	Description
Method	GET
Address	/self-service-api-v1.3/cart/items/[shoppingCartItemId]/image/[property]
Path Parameters	- `shoppingCartItemId` - Unique ID of the cart item. - `property` - Name of the cart item property holding the image.
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	N/A
Usage	Images that are properties of the shopping cart item.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 403: Resource not accessible - 404: Image not found
Return	Image file with respective Content-Type.

```
/self-service-api-v1.3/cart/items/95cdd6c6d484497aa0f510884744d557/
```

Visualization

It is recommended to use the Tacton Visualization JavaScript Library when integrating the visualization in a web application utilizing the API for Self Service. This library contains the basic functionality of the iFrame hosting the 3D visualization and is available as part of every tenant.

≣ Load JavaScript library ▼

Load the JavaScript library provided by your tenant.

```
<script src="https://[apiURL]/js/tacton.vis.js"></script>
```

Initialize Visualization

The main function provided by this javascript library is called *tactonVis* and sets up the visualization based on the current configuration result. Subsequent calls are needed to keep the visualization updated with the current configuration.

```
tactonVisObject = tactonVis(containerHTMLElement, visualizationObject, conf
```

Parameter	Details
containerHTMLElement	The parent HTMLElement into which the visualization iframe will be inserted or where it currently exists in case of a subsequent call. Typically a DIV container on the web site where the visualization should be displayed.
visualizationObject	The JSON object returned when requesting visualization-relevant information [Get Visualization](). It contains the location of all files needed to initialize the visualization:

Parameter	Details		
	assetsPath	String	URL where the visual assets are located.
	customImagesRootPath	String	*Optionally* URL where the [custom visualization images](#) are located.
	iframePath	String	URL of the visualization version being used.
	screenshotRenderer	String	URL of the server processing screenshot generation requests.
	sessionServer	String	Location of the web socket server processing session sharing requests.
	tvkbPath	String	URL where the visual logic is located.
	tvkbReference	String	Name of the visual logic file being used.

It contains information about available buttons on the visualization UI:

debugMode	Boolean	If the button to copy the current configuration to

Parameter	Details		
			the clipboard is available.
	`fullscreen`	Boolean	If the button to switch the visualization to fullscreen is available.
	`share`	Boolean	If the button to open the sharing dialog is available.
	`visaction`	Boolean	If the buttons executing project specific functionality are available.

It contains visualization relevant configuration information:

`currentStep`	String	Name of the configuration step that is currently selected.
`customImageData`	String	Available and [selected custom visualization images]().
`messagingData`	[{String,String}]	Array of visualization relevant configuration attributes and their current values.

It contains information about the visualization:

`name`	String	Name of the visualization object in CPQ.

Parameter	Details

screenshots	[{Object}]	Array of objects containing information about available screenshots.

It contains style information:

aspectRatio	String	Aspect ratio of the Visualization (e.g. "1:1").
primaryColor	String	Color used for loading spinner (e.g. "#ff00ff").

It is possible to adjust the JSON object before passing it on, to adjust the visualization according to the Self Service use case. For Example removing buttons on the visualization UI.

And it is also possible to add additional parameters to the *visualizationObject* that are not provided by the configurator in a Self Service use case:

currentGroup	String	The current configuration group name being displayed. This parameter may be required by a visualization if it was designed to use it.
useCSP	Boolean	Determines if the Content Security Policy meta tags will be rendered in the visualization iframe. By default, the lack of this parameter evaluates to true.

Parameter	Details
configResponseObject	*Optionally* The response sub-object returned in every [configuration result](#) (JSON). It contains information to display configuration errors or conflicts. This information will not be passed if parameter is omitted.
triggerObject	*Optionally* The object to send builtin triggers ("toggleFullscreen", "share", "copyconfig") or vis actions (button names) to the visualization.

Return value	Details
tactonVisObject	*tactonVis* returns this object on which the visualization messaging handlers can be registered, see [Send Messages to VIZ](#) for more details.

Order of processing inside the visualization:

1. Configuration update
2. Update group
3. Handling of triggers

> **Example** ▼

```
// initialize the visualization
var data = {};
data["_key"] = "abc123";
data["configState"] = response.configState;
// response as returned from /configurator-api/start, /configurator
var containerHTMLElement;
var visualizationObject;
var xhttp = new XMLHttpRequest();
xhttp.open("POST", "https://NAME.tactoncpq.com/configurator-api/vis
xhttp.setRequestHeader('Content-type', 'application/x-www-form-urle
xhttp.onreadystatechange = function(event) {
if (xhttp.readyState == XMLHttpRequest.DONE && xhttp.status == 200)
visualizationObject = JSON.parse(xhttp.responseText);
containerHTMLElement = document.getElementById(visualizationObject.
}
};
xhttp.onerror = function(event) {
//error code here
};
let body = "_key=" + data["_key"] + "&configState=" + encodeURICom
xhttp.send(body);
```

Trigger Visualization events

To trigger specific visualization events, a subsequent call of the tactonVis object is needed.

Event	Type	Description
toggleFullscreen	triggers	Toggles the fullscreen mode
share	triggers	Shows the share dialog including QR code
copyconfig	triggers	Copies the current configuration into the clipboard to be used in the VIZstudio preview
<buttonname>	buttonTriggers	Trigger action button

Example ▼

```
// send a fullscreen trigger
function sendFullscreenTrigger() {
tactonVis(containerHTMLElement, visualizationObject, {}, {"triggers
}
// send a share trigger
function sendShareTrigger() {
tactonVis(containerHTMLElement, visualizationObject, {}, {"triggers
}
// send a copyconfig trigger
function sendCopyConfigTrigger() {
tactonVis(containerHTMLElement, visualizationObject, {}, {"triggers
}
// send a vis action trigger
function sendVisActionTrigger(name) {
tactonVis(containerHTMLElement, visualizationObject, {}, {"buttonTr
}
```

VIZ Message Handling

The "tactonVisObject", provided as a result of calling the "tactonVis" function, is required to update the external web application, or the configuration when interacting directly with visualization. The messageHandler callback function will be called every time the visualization detects relevant user interaction or other backend events. The callback function will be called with a data object parameter containing information about the event. For example, the current configuration attributes set on the visualization.

> ✏️ **Note**
>
> Subsequent calls to the *onMessage* method will register multiple callbacks.

Types of messages sent to the messageHandler:

Action	Details	Example Message
multicommit	Configuration change request sent from visualization	{"action": "multicommit", parameters: {"button1": true, "button2": false}}
group	Group change request sent from visualization	{"action": "group", "value": "options"}
button_visibility	Vis action visibility change notification sent from visualization	{"action": "button_visibility", parameters: {"button1": true, "button2": false}}

Add Custom Visualization Image

Setting	Description
Method	POST
Address	/self-service-api-v1.3/custom-images/[configId]/[customImageId]
Path Parameters	- `configId` - ID of the configuration as reported when starting the configuration. - `customImageId` - ID of the custom image widget receiving the file upload. The ID is found in property `tc_custom_image_id` on group member with property `tc_gui_type` = `custom_viz_image` (URL encoded).
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `file` - Request parameter. Name of file, also it's ID.
Other Parameters	File data provided as binary in request body.
Usage	Upload image file for a custom image field.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: incorrect input data
Return	Success message

Upload file `logo.png` for custom image field `cust_img_1` in the configuration with config id `36b96a6ceb3c4e6c95255ac204e462be` .

```
/self-service-api-v1.3/custom-images/36b96a6ceb3c4e6c95255ac204e462
```

List Custom Visualization Images

Setting	Description
Method	GET
Address	/self-service-api-v1.3/custom-images/[configId]/[customImageId]/get-custom-image-file-list
Path Parameters	- `configId` - ID of the configuration as reported when starting the configuration. - `customImageId` - ID of the custom image widget. The id is found in property `tc_custom_image_id` on group member with property `tc_gui_type` = `custom_viz_image` (URL encoded).
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	N/A
Usage	Get the list of uploaded image files for a custom image field.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: incorrect input data
Return	List of uploaded files (in JSON format) with following details: - `files` - List of files, for each specifying: - `key` - File key (unique ID). - `path` - Relative file path. - `lastModified` - Integer of epoch time. - `url` - Url to retrieve original image file. - `thumbnailUrl` - Url to retrieve generated thumbnail image file.

Setting	Description
	- `size` - File size as descriptive string. - `sizeInBytes` - File size as integer value.

📋 **Get list of custom uploaded images to field** ▼

Get the list of uploaded files for custom image field `cust_img_1` in the configuration with config id `36b96a6ceb3c4e6c95255ac204e462be` .

```
/self-service-api-v1.3/custom-images/36b96a6ceb3c4e6c95255ac204e462
```

Select Custom Visualization Image

Setting	Description
Method	PUT
Address	/self-service-api-v1.3/custom-images/[configId]/[customImageId]
Path Parameters	- `configId` - ID of the configuration as reported when starting the configuration. - `customImageId` - ID of the custom image widget receiving the file upload. The id is found in property `tc_custom_image_id` on group member with property `tc_gui_type` = `custom_viz_image` (URL encoded).
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `file` - Request parameter. Id/name of file to be selected.
Other Parameters	N/A
Usage	Select uploaded image for a custom image field.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: incorrect input data
Return	Success message

≣ **Select custom visualization image** ▼

Select file `logo.png` for custom image field `cust_img_1` in the configuration with config id `36b96a6ceb3c4e6c95255ac204e462be` .

```
/self-service-api-v1.3/custom-images/36b96a6ceb3c4e6c95255ac204e462
```
◄ ━━━━━━━━━━━━━━━━━━ ►

List Selected Custom Visualization Images

Setting	Description
Method	GET
Address	/self-service-api-v1.3/custom-images/[configId]/get-selected-custom-images
Path Parameters	`configId` - ID of the configuration as reported when starting the configuration.
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	N/A
Usage	Get a list of all selected images for all custom visualization image fields in this configuration.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: incorrect input data
Return	List of uploaded files (in JSON format) with following details: - `selections` - List of file selections, for each specifying: - `id` - The custom image id. - `value` - File key (unique ID).

≣ **Get a list of all selected custom visualization images** ▼

Get a list of the selected file for each custom image field in the configuration with config id `36b96a6ceb3c4e6c95255ac204e462be` .

```
/self-service-api-v1.3/custom-images/36b96a6ceb3c4e6c95255ac204e462
```

Delete Custom Visualization Image

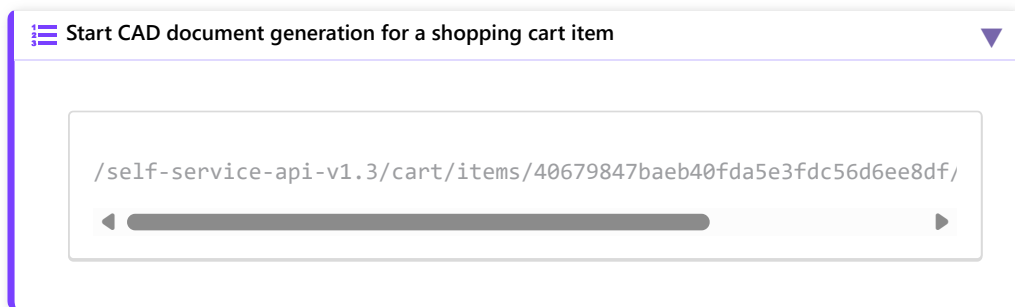| Setting	Description
Method	DELETE
Address	/self-service-api-v1.3/custom-images/[configId]/[customImageId]
Path Parameters	- `configId` - ID of the configuration as reported when starting the configuration. - `customImageId` - ID of the custom image widget receiving the file upload. The id is found in property `tc_custom_image_id` on group member with property `tc_gui_type` = `custom_viz_image` (URL encoded).
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `file` - Request parameter. Id/name of file to be deleted.
Other Parameters	N/A
Usage	Delete image previously uploaded to a custom image field.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: incorrect input data
Return	Success message

Delete custom visualization image

Delete file `logo.png` for custom image field `cust_img_1` in the configuration with config id `36b96a6ceb3c4e6c95255ac204e462be` .

```
/self-service-api-v1.3/custom-images/36b96a6ceb3c4e6c95255ac204e462
```
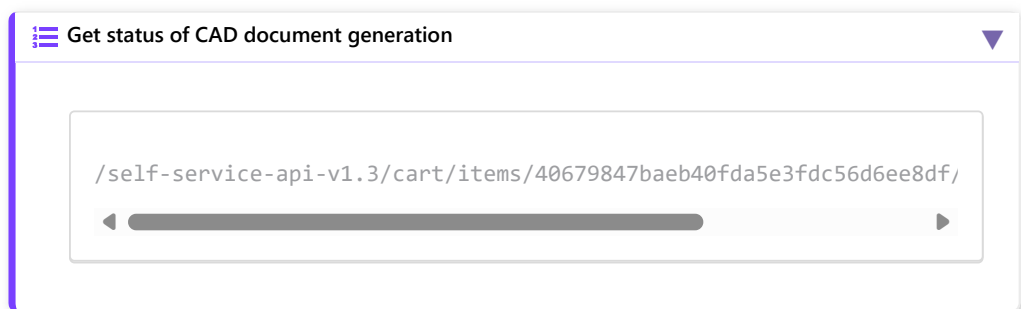
CAD Automation

Start CAD Document Generation

Setting	Description
Method	POST
Address	/self-service-api-v1.3/cart/items/[shoppingCartItemId]/cad/start-cad-automation
Path Parameters	`shoppingCartItemId` - Unique ID of the cart item to be updated.
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	N/A
Usage	Start CAD document generation for the product within this shopping cart item.
Exceptions	- 400: Not licensed to use the API - 400: CAD automation job already queued - 400: Product does not support CAD automation - 403: Unauthorized (no authentication key provided)
Return	JSON object with `status` element with value `ok` if successful. Otherwise an error HTTP code and error JSON.

▤ **Start CAD document generation for a shopping cart item** ▼

/self-service-api-v1.3/cart/items/40679847baeb40fda5e3fdc56d6ee8df/

◄ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ►

Status of CAD Document Generation

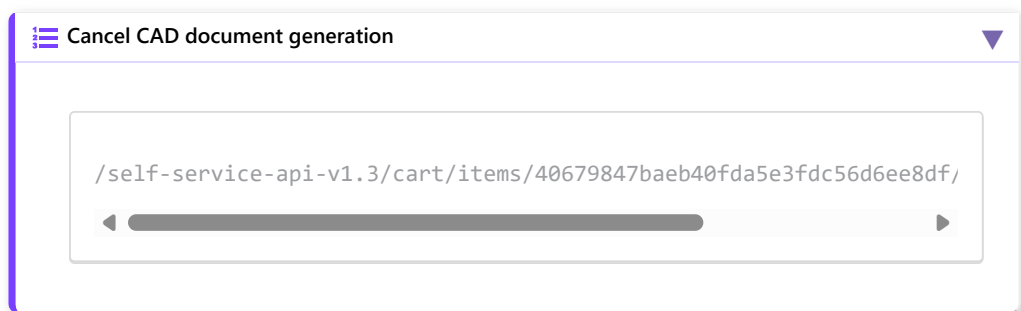| Setting	Description
Method	GET

Setting	Description
Address	/self-service-api-v1.3/cart/items/[shoppingCartItemId]/cad/cad-automation-status
Path Parameters	`shoppingCartItemId` - Unique ID of the cart item to be updated.
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	N/A
Usage	Check the status of CAD document generation of the shopping cart item.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided)
Return	JSON object with `status` and `error` elements, where error is empty or contains a message. Status can be one of: - `no cad automation` - This item cannot generate CAD. - `not generated` - Generation has not started. - `pending` - The job is not finished. - `Finished` - The job is finished.

☰ Get status of CAD document generation ▼

```
/self-service-api-v1.3/cart/items/40679847baeb40fda5e3fdc56d6ee8df/
```

Cancel CAD Document Generation

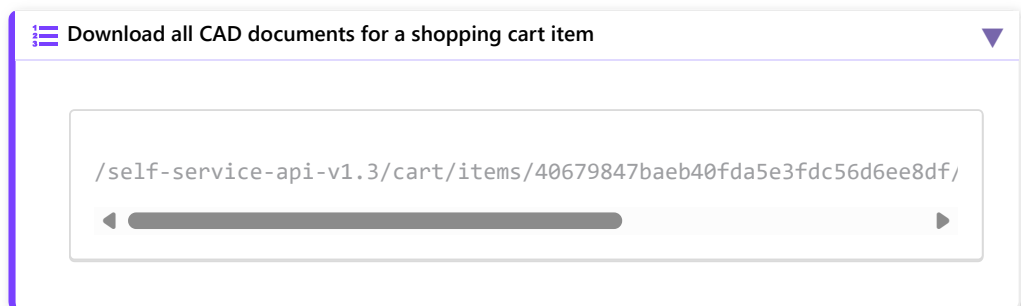| Setting	Description
Method	POST
Address	/self-service-api-v1.3/cart/items/[shoppingCartItemId]/cad/cancel-cad-automation
Path Parameters	`shoppingCartItemId` - Unique ID of the cart item to be updated.

Setting	Description
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	N/A
Usage	Cancel the generation of CAD documents for all products within this shopping cart.
Exceptions	- 400: Not licensed to use the API - 400: CAD Automation job is not pending - 400: No CAD Automation job found - 403: Unauthorized (no authentication key provided)
Return	JSON object with `status` element with value `ok` if cancel was successful. Otherwise an error HTTP code and error JSON.

☰ Cancel CAD document generation ▼

```
/self-service-api-v1.3/cart/items/40679847baeb40fda5e3fdc56d6ee8df/
```
◄ ━━━━━━━━━━━━━━━━━━━━ ►

Download all CAD Documents

Setting	Description
Method	GET
Address	/self-service-api-v1.3/cart/items/[shoppingCartItemId]/cad/download-all-generated-cad-documents
Path Parameters	`shoppingCartItemId` - Unique ID of the cart item to be updated.
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	N/A

Setting	Description
Usage	Download all generated CAD documents of all products within this shopping cart.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: No data found
Return	The compressed archive of documents.

≡ **Download all CAD documents for a shopping cart item** ▼

```
/self-service-api-v1.3/cart/items/40679847baeb40fda5e3fdc56d6ee8df/
```

◀ ▬▬▬▬▬▬▬▬▬▬▬ ▶

Download CAD Document List

Setting	Description
Method	GET
Address	/self-service-api-v1.3/cart/items/[shoppingCartItemId]/cad/get-generated-cad-document-list
Path Parameters	`shoppingCartItemId` - Unique ID of the cart item to be updated.
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	N/A
Usage	Retrieve a list of all CAD documents of all cart items within this shopping cart.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided)
Return	List of generated CAD documents for the shopping cart item (in JSON format), with following details: - `files` - List of files, for each specifying: - `type` - The document type, for example `jpg-2d`

Setting	Description
	- `file` - The relative path of the file. This value can be used when downloading specific CAD documents.

> ≡ Get a list of all CAD documents for a shopping cart ▼
>
> ```
> /self-service-api-v1.3/cart/items/40679847baeb40fda5e3fdc56d6ee8df/
> ```

Download Specific CAD Document

Setting	Description
Method	GET
Address	/self-service-api-v1.3/cart/items/[shoppingCartItemId]/cad/download-generated-cad-document
Path Parameters	`shoppingCartItemId` - Unique ID of the cart item to be updated.
Required Parameters	- `_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`). - `file` - Request parameter. Value is copied from result of the document list call above.
Other Parameters	N/A
Usage	Download one specific CAD document of one of the products within this shopping cart.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: File not found
Return	CAD document with respective Content-Type.

> ≡ Download a specific CAD document for shopping cart item ▼

```
/self-service-api-v1.3/cart/items/40679847baeb40fda5e3fdc56d6ee8df/
```

Deprecated Methods

The methods below are deprecated and have been removed in self-service-api-v1.3.

Commit a Parameter (Deprecated)

> ✏️ **Note**
>
> This operation is deprecated and doesn't properly support all possible parameter names and value types.

Setting	Description
Method	POST
Address	/self-service-api-v1.2/config/commit/[parameter]/[value]
Path Parameters	- `parameter` - Unique parameter name. - `value` - Value of the parameter to be committed.
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	- `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `configId` - Request parameter. Required to continue a specific configuration started for specific external entity. - `configState` - Request parameter. Required to continue a configuration started anonymously.
Usage	Commit a parameter, i.e. select a value for a field, thus preventing the engine from changing the value of that parameter.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: No data found (provided configId is not associated with any ongoing configuration for that external ID)
Return	[Configurator Data](#)

Uncommit a Parameter (Deprecated)

✏️ Note

This operation is deprecated and doesn't properly support all possible parameter names.

Setting	Description
Method	POST
Address	/self-service-api-v1.2/config/uncommit/[parameter]
Path Parameters	`parameter` - Unique parameter name.
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	- `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `configId` - Request parameter. Required to continue a specific configuration started for specific external entity. - `configState` - Request parameter. Required to continue a configuration started anonymously.

Setting	Description
Usage	Uncommit a parameter, i.e. unlock a field so that the engine again chooses its value.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: No data found (provided configId is not associated with any ongoing configuration for that external ID)
Return	[Configurator Data](#)

> 📋 **Example** ▼
>
> Uncommit parameter `bike_model_field` in configuration with `configId=6a67a3539a684647841cc8b9e8a73919`, started by external entity with `_externalId=d486cb2d`.
>
> ```
> /self-service-api-v1.2/config/uncommit/bike_model_field?_externalId
> ```

Accept Conflict Resolution (Deprecated)

> ✏️ **Note**
>
> This operation is deprecated and doesn't properly support all possible parameter names.

Setting	Description
Method	POST
Address	/self-service-api-v1.2/config/accept/[parameter]/[value]
Path Parameters	- `parameter` - Unique parameter name. - `value` - Value of the parameter to be accepted.
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	- `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `configId` - Request parameter. Required to continue a specific configuration started for specific external entity.

Setting	Description
	- `configState` - Request parameter. Required to continue a configuration started anonymously.
Usage	Commit a parameter and accept proposals by the configuration engine on how to resolve conflicts.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: No data found (provided configId is not associated with any ongoing configuration for that external ID)
Return	[Configurator Data](#)

≡ **Example** ▼

Accept parameter `front_brake_type_field=ACM0022` in the configuration with specified `configState`.

```
/self-service-api-v1.2/config/accept/front_brake_type_field/ACM0022
```

◀ ━━━━━━━━━━━━━━━━━━ ▶

Change Configuration Step (Deprecated)

✎ **Note**

This operation is deprecated and doesn't properly support all possible step names.

Setting	Description
Method	POST
Address	/self-service-api-v1.2/config/step/[step]
Path Parameters	`step` - Name of the step, as listed in the `steps` list element of the configuration response.
Required Parameters	`_key` - Self Service API key, passed either as a request parameter or request header (`X-KEY`).
Other Parameters	- `_externalId` - External entity ID, passed either as a request parameter or request header (`X-EXTERNAL-ID`). - `configId` - Request parameter. Required to continue a

Setting	Description
	specific configuration started for specific external entity. - `configState` - Request parameter. Required to continue a configuration started anonymously.
Usage	Navigate the configuration to specified step.
Exceptions	- 400: Not licensed to use the API - 403: Unauthorized (no authentication key provided) - 404: No data found (provided configId is not associated with any ongoing configuration for that external ID)
Return	[Configurator Data](#)

> ☰ **Example** ▼
>
> Navigate to `step=DetailedConfigurationStep` in the configuration with `configId=36b96a6ceb3c4e6c95255ac204e462be` started for `_externalId=d486cb2d` .
>
> ```
> /self-service-api-v1.2/config/step/DetailedConfigurationStep?_exter
> ```
> ◄ ▬▬▬▬▬▬▬▬▬▬ ►