

# Version Open APIs

Overview.....	2
Authentication.....	2
Asynchronous request processing and tracking.....	2
Get Version v1.....	3
Request.....	3
Response.....	3
Get Versions v1.....	5
Request.....	5
Response.....	5
Duplicate Version v1.....	7
Request.....	7
Response.....	8
Activate Version v1.....	9
Request.....	9
Response.....	9
Get Products of the specific Version v1.....	10
Request.....	10
Response.....	11
Get Specific Products (Products Catalog) v1.....	15
Request.....	15
Response.....	16
Extract products Excel v1.....	17
Request.....	17
Response.....	17
Get Asynchronous Request Status v1.....	18
Request.....	18
Response.....	18
Get Asynchronous Request Execution Summary v1.....	19
Request.....	19
Response.....	20
Errors.....	22



## Overview

The current document describes Open API services over the *Version* entity of the DealHub CPQ system. In the CPQ system the *Version* is an entity that consolidates the configuration of products, prices, discount policies, and various business flows to allow sales reps to ensure the delivery of accurate quotes to buyers.

Some of the APIs below are asynchronous. Thus the response system will return the `request_id` parameter so that the consuming system will be able to track the result of the asynchronous request execution.

## Authentication

Current API document designed for system-to-system integration with token-based authentication. The header of each request described below must include a DealHub Authentication Token. This token should be generated by the CPQ administrator and shared with the consuming application in advance.

Authorization: Bearer <TOKEN\_THAT\_WAS\_PROVIDED\_BY\_CPQ\_ADMIN>

## Asynchronous request processing and tracking

For an asynchronous API, the system will respond with the `request_id` parameter. The same `request_id` will be part of the WebHook event that is published by CPQ once the request is complete. For more information on CPQ events, please refer to the ***Open API WebHooks*** document.



## Get Version v1

Retrieve precise *Version* based on version name or version ID.

### Request

type: GET

URI: /api/v1/version

Request example:

api/v1/version/id/{version\_id}

api/v1/version/name/{version\_name}

Optional Request Parameters - N/A

Name	Description	Empty Value	Valid values
version_id	ID of the version	no	max length 16 chars
name	Name of the version	yes	max length 120 chars

### Response

Code: 200 (Success)

Response format: JSON

Response example:

```
{
  "name": "Master_v1",
  "comment": "",
  "status": "ACTIVE",
  "version_id": "67657"
}
```

Fields details:

Name	Description	Empty Value	Format/Notes
name	Name of the version	no	
comment	Comment provided during version creation	yes	
status	Status of the version	no	
version_id	ID of the version	no	

## Get Versions v1

Retrieve the *Versions* list based on filter criteria.

If no parameters are provided in the request the system will retrieve a list of all *Versions* under a specific account.

## Request

type: GET

URI: /api/v1/versions

Request example:

api/v1/versions?status=active

## Optional Request Parameters

Name	Description	Valid values
status	Search for <i>Version(s)</i> based on status	"DRAFT" "ACTIVE" "DEACTIVATED"

## Response

As a response the system will return the list of *Version(s)* that matching given criteria

Code: 200 (Success)

Response format: JSON

Response example:

```
[
  {
    "name": "v3a",
    "comment": "",
    "status": "DEACTIVATED",
    "version_id": "068DoObd1i5GuF14"
  },
  {
    "name": "msd6",
```



```
    "comment": "",  
    "status": "DEACTIVATED",  
    "version_id": "0AjX3aABgLN9f8uG"  
  }  
]
```

Fields details:

Name	Description	Empty Value	Format/Notes
name	Name of the version	no	
comment	Comment provided during version creation	yes	
status	Status of the version	no	
version_id	ID of the version	no	



## Duplicate Version v1

By utilizing this API the consuming system will create a new Version entity within the CPQ system based on an existing Version entity.

For scenarios where an account would like to duplicate Version from its Sandbox environment to its Production environment, the request payload should include the `to_account_id` parameter. In such a scenario the request should be sent from the Sandbox account. Version duplication based on another account ID is allowed only if both accounts have the same domain in the CPQ.

## Request

type: POST

URI: `/api/v1/version/duplicate`

Query body format: JSON

### Mandatory Request Parameters

Name	Description	Valid values/Notes
version_id	ID of the version that should be duplicated (aka source version)	max length 16 chars
new_version_name	Name of new version	max length 120 chars

### Optional Request Parameters

Name	Description	Valid values
comment	Comments to be added for the newly created version	max length 4000 chars
to_account_id	Account ID of the newly created version. Relevant in case the source version on sandbox env needs to be duplicated to the production environment. The request is done from the Sandbox account.	max length 16 chars



Query body example:

```
{
  "version_id" : "765379899",
  "new_version_name" : "New Version",
  "comment" : "Adding more BroadBand services",
  "to_account_id" : "8764827348"
}
```

## Response

Code: 200 (Success)

As a response the system will return the ID of the asynchronous request for version duplication.

This request\_id will be published in the relevant Version event when request processing will complete.

Expected response example in case of success:

```
{
  "request_id": "4D114DAD97"
}
```





## Activate Version v1

By utilizing this service the consuming system will be able to activate the draft *Version* entity within the CPQ system.

### Request

type: POST

URI: /api/v1/version/activate

Query body format: JSON

Mandatory Request Parameters

Name	Description	Valid values/Notes
version_id	ID of the <i>Version</i> that should be activated	max length 16 chars

Optional Request Parameters - N/A

Query body example:

```
{
  "version_id" : "765379899"
}
```

### Response

As a response the system will return the ID of the asynchronous request for *Version* activation.

Expected response example:

```
{
  "request_id": "4D114DAD97"
}
```

## Get Products of the specific Version v1

### Request

The data returned by this service includes primarily product attributes information and basic price details.

By utilizing this service, the consuming system will be able to retrieve a list of the products configured for a specific Version.

- **DRAFT Status:** If the specified Version is in 'DRAFT' status, the output response will always include the full products list.
- **ACTIVE Status:** If the specified Version is in 'ACTIVE' status, the output response can include either all products in the catalog or only modified products. Modified products are newly added or updated SKUs compared to the Version preceding the current active Version.
- **DEACTIVATED Status:** If the specified Version is in 'DEACTIVATED' status, the output response can include either the full list of products in the catalog or only modified products. The list of 'MODIFIED' products remains the same as when the Version was 'ACTIVE', i.e., SKUs identified as new or updated during the initial Version activation compared to its preceding active Version.

If the `limit` value in the request exceeds the maximum limit, the system will return a response based on the maximum limit.

type: GET

URI:

`/api/v1/version/{version_id}/products`

Request example:

`/api/v1/version/{version_id}/products?product_option=MODIFIED&pricing_option=MAX`

### Mandatory Request Parameters

Name	Description	Valid values/Notes
version_id	ID of the <i>Version</i>	max length 16 chars

## Optional Request Parameters

Name	Description	Valid values/Notes
offset	Part of the offset-based pagination. Parameter helps the system to identify how many products should be skipped.	0 - 10000
limit	Part of the offset-based pagination. Parameter determines the maximum number of products returned by this API	1 - 1000
product_option	Specify if to fetch all products (ALL) or only new/updated products of the version (MODIFIED)  By default (if this parameter is not specified) CPQ will return all products	<b>MODIFIED</b> <b>ALL</b>  Default value: <b>ALL</b>
pricing_option	Specify which price to return. CPQ allows setting multiple price options for each product (based on business conditions, aka rules).  For SKUs that have multiple fixed prices (price per rule), the system will return the price based on the requested pricing option. Otherwise, it will return 0.	<b>MIN</b> <b>MAX</b> <b>FIRST</b>  Default value: <b>MIN</b>

## Response

System will retrieve products catalog as configured within CPQ in context of specific Version.

Code: 200 (Success)

Response format: JSON

Response example:

```
{
  "info": {
    "version_id": "76472",
    "version_status": "ACTIVE",
    "compared_with_version_id": "76326",
    "more_results_matching_the_request": true,
  }
}
```



```
"offset": "1",
"limit": "25"
},
"products": [
  {
    "sku": "56545",
    "name": "Product A",
    "type": "PRODUCT",
    "description": "Description of product A",
    "price": {
      "EUR": "10",
      "USD": "12.5"
    },
    "attributes": [
      {
        "attr_x": "value X"
      },
      {
        "attr_y": "value Y"
      }
    ]
  },
  {
    "sku": "56546",
    "name": "Product B",
    "type": "PRODUCT",
    "description": "Description of product B",
    "price": {
      "EUR": "18",
      "USD": "21.5"
    },
    "attributes": [
      {
        "attr_x": "value XX"
      },
      {
        "attr_y": "value YY"
      }
    ]
  },
  {
    "sku": "56545_b",
    "name": "Bundle X",
    "type": "BUNDLE",
    "description": "Description of bundle X",
    "price": {
      "EUR": "10"
    },
    "attributes": [
```

```

    {
      "attr_x": "value XYZ"
    },
    {
      "attr_y": "value YZ"
    }
  ],
  "bundle_items": [
    "56545",
    "56546"
  ]
}
]
}

```

## Fields details:

Name		Description	Empty Value	Format/Notes
info		Basic information about returned payload	no	
	version_id	The ID of the version for which products were retrieved (e.g. version-B)	no	
	version_status	The status of the version for which products were retrieved	no	
	compared_with_version_id	The ID of the version (version-A) that was replaced by version-B (during <b>initial activation of version-B</b> ).	yes	Empty if version-B is still in draft status.
	error	Error message relevant for retrieval of products by SKU. In case not all of the requested data could be retrieved. E.g. the consuming app requests to receive 5 products, while some of them do not exist in the specified version.	yes	
	more_results_matching_the_request	Flag that specifies to consuming system if there are more entries on the Dealhub side that are matching the request criteria		
	offset	Offset as provided in HTTP request. If		

		not explicitly provided in the request, the system will use the default offset (0)		
	limit	Limit as provided in HTTP request. If not explicitly provided, the system will use the default limit (1000).		
products	List of products matching specified criteria. Each element in the list contains the following information:			
	sku	Product catalog unique identifier	no	
	name	Product name	no	
	type	Type of the product	no	PRODUCT BUNDLE
	description	Product description	yes	
	price	List of price & currency pairs as defined for SKU.  For a bundle with type "Products Summary" the bundle price is calculated in run-time during actual quote generation, thus in response payload of this call the system will return 0 price.	no	Currency format ISO-4217
	attributes	List of all product attributes defined for the version	no	
	bundle_items	List of Product SKUs in the bundle. Will be instantiated if product type == BUNDLE.	yes	

## Get Specific Products (Products Catalog) v1

### Request

By utilizing this service the consuming system will be able to retrieve specific products from the catalog based on the SKU(s).

If some SKUs could not be found in a specific *Version* the system will return status 200, with error in the 'info' structure.

type: POST

URI:

/api/v1/version/product\_by\_sku

Query body format: JSON

Query body example:

```
{
  "version_id": "23687268JS5",
  "pricing_option": "MAX",
  "sku": [
    "Sku-123-dvvv",
    "Sku-123-dvv2",
    "Sku-123-dvv3"
  ]
}
```

Query body explanation:

Name	Description	Mandatory	Valid values
version_id	ID of the version for which product(s) details need to be retrieved. If not provided - system will lookup for specified SKU(s) in the <i>ACTIVE Version</i> .	no	max length 16 chars
pricing_option	Specify which price to return. CPQ allows setting multiple price options for each product (based on business conditions, aka rules).  For SKUs that have multiple fixed prices (price per rule), the system will return the	no	

Name	Description	Mandatory	Valid values
	price based on the requested pricing option. Otherwise, it will return 0.  <b>MIN</b> - default if no explicit pricing options requested <b>MAX</b> <b>FIRST</b>		
sku	List of SKU which should be retrieved from the DealHub system	yes	max length 200 chars

## Response

Same as the response payload described in [Get Version Products \(Products Catalog\)](#) (above)





## Extract products Excel v1

By utilizing this service the consuming system will be able to retrieve products related information in a form of Excel file (same file format as available today via CPQ UI). Such information can be retrieved per the specific *Version*.

### Request

type: GET

URI: /api/v1/version/products\_excel/{version\_id}

#### Mandatory Request Parameters

Name	Description	Valid values/Notes
version_id	ID of the <i>Version</i>	max length 16 chars

Optional Request Parameters - N/A

### Response

As a response the system will return the byte array that represents the products with pricing rules and information extracted in the Excel format.

## Get Asynchronous Request Status v1

Every asynchronous API (e.g.: duplicate quote, activate quote, etc) will return to the consuming system the `request_id`. Upon completion of the request processing, CPQ will publish relevant events to the configured end-point (see WebHook Open API for more information).

However, in some cases, the consumer side might need to query CPQ for the status of the request instead of listening to WebHook events. The current service will query CPQ to fetch the status of the specific asynchronous request (polling mode).

### Request

type: GET

URI: `/api/v1/request/{request_id}/status`

#### Mandatory Request Parameters

Name	Description	Valid values/Notes
<code>request_id</code>	ID of the asynchronous request (as received in response to duplicate/activate version)	max length 16 chars

Optional Request Parameters - N/A

### Response

As a response the system will return available information for the given request ID.

Expected response example:

```
{
  "request_id": "4D114DAD97",
  "status": "done",
  "error_description": "",
  "error_code": ""
}
```

## Response details

Name	Description	Empty value	Valid Values/Notes
request_id	ID of asynchronous request	no	
status	Status of asynchronous request	no	"in-progress" "done" "failed"
error_description	Description of the error is request status "failed"	yes	
error_code	Code of the error if request status "failed"	yes	

## Get Asynchronous Request Execution Summary v1

This REST service is designed to return the results of asynchronous processing.

An asynchronous API (e.g., product upload) will return a request\_id to the consuming system. Upon completion of the request processing, CPQ will publish relevant events to the configured endpoint (see WebHook Open API for more information).

In cases where asynchronous processing finishes with errors, the consumer can query CPQ for a summary of the request execution.

## Request

type: GET

URI: /api/v1/request/{request\_id}/summary

### Mandatory Request Parameters

Name	Description	Valid values/Notes
request_id	ID of the asynchronous request (as received in	max length 16 chars

	response to duplicate/activate version)	
--	---	--

Optional Request Parameters - N/A

## Response

Expected response example:

```
{
  "status": "done",
  "summary": {
    "success_count": 1,
    "errors_count": 0,
    "warnings": [
      "Missing currency in price item for SKU KY014X, using default currency: USD"
    ],
    "errors": []
  }
}
```

## Response details

Name		Description	Empty value	Valid Values/Notes
status	The status of the asynchronous request.		no	Valid values: In Progress, Completed, Error
summary	Summary of the executed request. Relevant only if request processing is complete.		yes	
	success_count	The number of successfully processed records.	no	
	errors_count	The number of records that could not be processed.	no	
	warnings	List of warnings that occurred during the execution of an asynchronous process.  Note that warning/s do not impact the	yes	

Name		Description	Empty value	Valid Values/Notes
		execution process and do not increase the error count.		
	errors	List of errors that occurred during the execution of an asynchronous process.	yes	

## Errors

HTTP code	When to return	Error Message
400	Invalid payload format	Invalid payload format. Supported format: JSON
400	Parameter within request payload exceed its limit(s)	The request parameter <PARAM_NAME> exceeds its limits. Allowed maximum length: <MAX_LENGTH>
400	Request payload missing one or more mandatory fields	Request payload missing mandatory field(s): <comma separated list of missing mandatory fields>
400	One or more parameters are invalid (length, valid value etc.)	Invalid parameter format (<param_name>: <error>)
400	The system got a request with an ID that does not exist in DealHub (version ID, account ID etc.)	Entity (ID = <unknown_id>) not found
400	Mismatch between domain name of the source version account and target version accounts during the duplicate request.	The domain name of the source version account does not match the target version account
400	Version activation fails due to errors. The system will not list all validation errors to the consuming app as fixing Version errors require manual admin intervention in any case. This error will be published in the Failure event as a response to an async request of Version activation.	Version could not be activated due to validation errors. Please contact the administrator to fix all errors before activation.
400	Received activation request for an active Version	Version (id = <version_id>) already active.
400	New version name is not unique	A new version name should be unique. Please change the name and try again.
400	Received request to fetch 'MODIFIED' version products (products catalog) for 'DRAFT' version	'MODIFIED' products option is not available for version in 'DRAFT' status. Fetch 'ALL' products instead.

HTTP code	When to return	Error Message
400	No explicit version ID was provided in request, thus CPQ system approached 'ACTIVE' version to fetch requested products (by SKU)	Could not find the 'ACTIVE' version.
400	Invalid value sent for one of the parameters	<PARAMETER_NAME> - Invalid parameter value. Valid value(s): <VALID_VALUES>
400	The consuming app requests to receive a few products (based on SKU), while some of them do not exist in the specified version.	The following SKUs not found: <SEMICOLON_SEPARATED_LIST_OF_SKUS>
403	If the request is not authenticated (authentication token is missing or invalid)	Unauthenticated
400	Too many SKUs in the fetch request.	The number of requested items exceeds the allowed limit of <MAX_LIMIT>. Reduce the number of SKUs.
400	SKU does not exist in the system	The SKU [SKU] at [p4 -> toner A -> p5] does not exist in the products list. Please add the SKU to the system products or remove it from the product hierarchy and try again.
400	The field SKU/ Label name is empty	Invalid data found at [p4 -> toner A ] ,SKU/ Label name field cannot be empty.
400	Type field is empty	Invalid data found at [p4 -> toner A -> p5] .The field type is not defined. Choose from type [PRODUCT, LABEL, or BUNDLE).
400	Type field value is invalid	Invalid data found at [p4 -> toner A -> p5] .The field type is invalid. Valid types are [PRODUCT, LABEL, and BUNDLE).
400	label name is exceed the max characters limit	Invalid data found at [p4 -> toner A -> p5] .A label name cannot exceed 50 characters.
400	SKU exceed the max characters limit	Invalid data was found at [p4 -> toner A -> p5] .A SKU cannot exceed 200 characters.

HTTP code	When to return	Error Message
400	Invalid structure, label must have at least one child (product/bundle)	Invalid data found at [p4 -> toner A -> p5] .Label [label name] is set without any related products. Please remove the label or add related products.
400	Invalid structure, In case type label & set as mandatory → it has at least one child that is a product or bundle and set as mandatory	Invalid data found at [p4 -> toner A -> p5] .Label [label name] is a mandatory label without any mandatory products. Please set mandatory product(s) or change its restriction.
400	Invalid structure, item that set as alternative→does not have at least one more item product/label as an alternative at the same level	Invalid data found at [p4 -> toner A -> p5]. [product/bundle/label NAME] has incomplete configuration. Please ensure there is at least one more alternative at the same level for proper setup.
400	Invalid structure, item that set as alternative & mandatory →all product/label at the same level is not set as alternative & mandatory	Invalid data found at [p4 -> toner A -> p5] .for [nameOrSku] .You have an alternative option in this level, and one is set as mandatory. Please set all fields as both mandatory and alternative.
400	Invalid structure, mandatory field is empty.	Invalid data found at [p4 -> toner A -> p5] .The item "[nameOrSku]" has an empty mandatory field.
400	Invalid structure, mandatory field is not set as either true or false	Invalid data was found at [p4 -> toner A -> p5]. The item "[nameOrSku]" has invalid mandatory field value. The valid types are [TRUE, FALSE).
400	Invalid structure, alternative field is empty.	Invalid data was found at [p4 -> toner A -> p5]. The item "[nameOrSku]" has an empty alternative field.
400	Invalid structure, alternative field is not set as either true or false	Invalid data was found at [p4 -> toner A -> p5]. The item "[nameOrSku]" has invalid alternative field value. The valid types are [TRUE, FALSE).
400	element contains more than 10 levels	Invalid data found for [nameOrSku] at [p4 -> toner A -> p5]. You have reached the maximum limit of 10 levels that can be added to a product or a label.



HTTP code	When to return	Error Message
400	Delete product that has a children in the hierarchy tree, this product cannot be deleted	Unable to delete product [SKU] that has related product/s in the product hierarchy. Please remove any associated child products from the product hierarchy and try again.
400	Delete product that set as mandatory in the hierarchy tree, this product cannot be deleted	Unable to delete product [SKU] that was set as mandatory in the product hierarchy. Please remove it from the product hierarchy and try again.
400	Delete product that set as alternative in the hierarchy tree, this product cannot be deleted	Unable to delete product [SKU] that was set as an alternative in the product hierarchy. Please remove it from the product hierarchy and try again.
400	If the items in the product hierarchy has 50,000 items, an error message will appear	You can only add a maximum of 50,000 products or labels to the product hierarchy.
400	Invalid version status	Specified version cannot be modified: Invalid version status.