

External Query API

Overview	1
Assumptions	2
Authentication	2
External Query Request	3
External Query Response	6
Success Response	6
Error Response	8



Overview

This document aims to explain integration calls that can be triggered from the CPQ system to any 3rd party system that is compliant with the interface described below.

External Query feature provides our customers the ability to retrieve information essential for the quote generation from any external source. The solution is based on a HTTP request/response in JSON format.

Note that the data sent in the HTTP request is flexible and will be combined based on the definition of the DealHub account administrator.

Assumptions

1. Depending on the version configuration settings, the external call could be triggered by the CPQ user during the quote generation process, or systemically upon quote creation.
2. The query is executed per system, in case the quote is in multi-system mode

Authentication

During external query setup within the CPQ system, the administrator will be able to specify the authentication type that the external system requires.

The administrator will be able to specify one of the following authentication types:

- Token
When the Token authentication type is selected, the administrator will need to provide the actual token (that is shared by the provider of the endpoint). This secret token will be persisted in the system and will be sent in the header of each request under the following field
`Authorization: Bearer <TOKEN_SHARED_BY_ENDPOINT_PROVIDER>`
- Basic
When the Basic authentication type is selected, the administrator will need to provide the username and the password (as shared by the provider of the endpoint). These credentials will be persisted in the system and will be sent in the header of each request under the following field
`Authorization: Basic <CREDENTIALS_SHARED_BY_ENDPOINT_PROVIDER>`
- None
This option will be used in the case when the endpoint provider does not have any authentication mechanism, or the endpoint provider uses a Shared Access Signature (SAS) where the token is embedded into the URI of the endpoint.



External Query Request

Request type: POST

URL: <EXTERNAL_SYSTEM_ENDPOINT_URL_CONFIGURED_BY_ADMIN>

Query body format: JSON

Query body example (basic):

```
{
  "playbook_data": {
    "general.currency": "USD",
    "general.geo": "North America",
    "group1.customer_type": "Enterprise",
    "group2.number": 4323.11,
    "group3.proposal_date": "2021-03-25T15:39:25Z"
  },
  "product_data": [
    {
      "ITEM NAME": "Whiteboard",
      "SKU": "B-12345-W",
      "attribute1": "1237"
    },
    {
      "ITEM NAME": "Junior Kit",
      "SKU": "B-56-TRIPLE",
      "bundle_line_items": [
        {
          "ITEM NAME": "Pencil",
          "SKU": "sku-1",
          "attribute1": "9"
        },
        {
          "ITEM NAME": "Eraser",
          "SKU": "sku-2",
          "attribute1": "2"
        },
        {
          "ITEM NAME": "Sharpener",
          "SKU": "sku-3"
        }
      ]
    },
    {
      "ITEM NAME": "Sharpener",
      "SKU": "sku-3"
    }
  ]
}
```

Query body explanation:

Name	Description	Mandatory
playbook_data	<p>The list of playbook questions as configured by the CPQ account administrator.</p> <p>The syntax of each element in the list will look as: <playbook_group_id.question_id>:<answer></p> <p>Note that depending on the playbook group setting the <answer> might have single value (non-repeatable group definition) or multiple values (repeatable group definition). Refer to the request examples below 📌 for more details.</p> <p>If the CPQ administrator didn't map any data when configuring the API External Query object for specific <i>Version</i>, there will be no playbook data sent in the request.</p>	no
product_data	<p>An array of objects, where each object represents a product or a bundle that is part of a proposal at the moment of the HTTP request.</p> <p>The list of elements in each object depends on the configuration set up by the admin within CPQ. An admin can configure the system to include both built-in product fields and custom attributes in the object structure.</p> <p>Bundle structure - When a quote contains a bundle, the object structure will also include bundled line items. Each bundled line item follows the same structure as the product object.</p> <p>If a custom attribute is not applicable for a specific product, it will not be included in the request. This ensures that only relevant information is sent in the request payload. (see attribute1 in the example above)</p>	no

Query body example (group5 is repeatable and have 4 different answers):

```
{
  "playbook_data": {
    "general.currency": "USD",
```



```
"general.geo": "North America",
"group1.customer_type": "Enterprise",
"group2.number_of_something": 4323.11,
"group3.proposal_date": "2021-03-25T15:39:25Z",
"group5.repeatableNumericQuestion": [
    "1702339200000",
    "1702399200000",
    "1702339298000",
    "1502339200000"]
}
}
```

Query body example (group5 is repeatable, 2nd occurrence does not have an answer):

```
{
  "playbook_data": {
    "general.currency": "USD",
    "general.geo": "North America",
    "group1.customer_type": "Enterprise",
    "group2.number_of_something": 4323.11,
    "group3.proposal_date": "2021-03-25T15:39:25Z",
    "group5.repeatableQuestion": [
      "op2",
      "",
      "op1",
      "op2"]
  }
}
```

Query body example (repeatable multi-select text list without an answer in last occurrence):

```
{
  "playbook_data": {
    "general.currency": "USD",
    "general.geo": "North America",
    "group1.customer_type": "Enterprise",
    "group2.number_of_something": 4323.11,
    "group3.proposal_date": "2021-03-25T15:39:25Z",
    "group5.repeatableMultiSelectTextListQuestion": [
      "op2; op4",
      "op4",
      "op1; op2; op3",
      ""]
  }
}
```

External Query Response

Success Response

The response (HTTP-200) payload will contain the data required for the specific quote generation (e.g. subscription information relevant for the renewal quote).

Note that the structure of the response is flexible. The system will inject received information into the playbook based on the keys sent in the response payload. The keys (question IDs) should correspond to the questions implemented by the account administrator in the CPQ.

Response data might contain more data than what the administrator configured. Any additional information in the response will be **ignored**.

Expected response example:

```
{
  "playbook_data": [
    {
      "id1": "id1_val",
      "id2": "A;B;C",
      "id3": "some_val"
    },
    {
      "id1": "id1_val",
      "id2": "A;D;N",
      "id3": "another_val"
    }
  ]
}
```

Query body explanation:

Name	Description
playbook_data	External data. List of the specific question IDs with the respective data that should be populated into the CPQ playbook.

If the external query response is not received (timeout), the system will present a warning message to the seller: "External query call timed out. Could not receive data for the group (<GROUP_NAME>) from <EXTERNAL_SYSTEM_URL_CONFIGURED_BY_ADMIN>"
The quote generation won't stop.



If the external query response does not contain any data while the response code is HTTP-200, the quote generation won't stop.

In both above scenarios, the submission of the quote will depend on the playbook definition. I.e. if the CPQ admin configured the playbook with a fallback (e.g. questions have default values or are defined as optional), the quote submission will continue.

Error Response

In the scenario where an external system need to return an error, it should follow the following response payload format (HTTP-400):

```
{  
  "message": "This is an example of the formatted error from the external  
system."  
}
```

An error message received from the external system will be presented to the user, but the quote generation won't stop.

The submission of the quote will depend on the playbook definition. I.e. if the CPQ admin configured the playbook with a fallback (e.g. questions have default values or are defined as optional), the quote submission will continue.