PUBLIC
2026-03-27

SAP Signavio Process Manager API Guide

Content

1 SAP Signavio Process Manager API Guide

With the SAP Signavio Process Manager APIs, you can integrate the application with third-party applications, allowing you to manage all diagram, folder, and dictionary data. You can do things such as create, update or retrieve your data in addition to other operations.

Each API provides a collection of endpoints related to a specific resource. You can learn about the functionality provided by each API in the Resource APIs section, as well as general information about requests and responses, and examples of using the API.

For more information, visit our complete getting started tutorial available on the SAP community page at Getting Started with SAP Signavio Process Manager API.

Using a Postman Collection

If you wish to send requests to this API using Postman, we provide a ready-to-use collection for you to download and reuse. You can find it at the SAP Samples GitHub repository .

2 Access and Licensing

API access requires setting up technical user accounts with the appropriate licenses and permissions.

Setting up Technical Users

To integrate with the SAP Signavio Process Manager API, we advise that you create technical user accounts and assign only the **API Edition** license to these technical users. Don't add any other licenses to your technical users. The dedicated **API Edition** license enables technical users to bypass the single sign-on login process, facilitating seamless API operations.

When creating technical user accounts, we recommend using dedicated technical email addresses, such as *integrations-one@[companyname].com*, *integrations-two@[companyname].com*, and so on, instead of personal emails. You would need access to these email addresses. Assigning the **API Edition** license to technical users works like assigning any other license to a regular user. For more information, see User Accounts.

> → Tip
>
> It's best practice to always use *one* API technical user account for each integration scenario or integration system.

Make sure that the technical users have the required permissions, such as the ability to read processes or to create dictionary items. Granting permissions is done by adding access rights. For more information, see Manage Access Rights.

> ⚠ Restriction
>
> With the **API Edition** license assigned to technical user accounts, users are unable to access the application through the user interface.

Getting the License

The **API Edition** license is provided at no additional cost.

If there is already one or more **API Edition** licenses available in your workspace, go ahead and assign them to the technical user accounts. If an **API Edition** license isn't readily available in your workspace, request one by proceeding as follows:

1. Contact SAP Support by creating a support case in SAP for Me. For more information, see Reporting an Issue in the *SAP Signavio Support Guide*.
2. Include your Tenant ID and region or server when submitting the incident. Our support team will assign licenses to your workspace and from there, you can assign them to the technical user accounts.

> ⚠ Caution
>
> Utilizing the credentials of a user associated with any license other than **API Edition** consumes a paid license, which incurs additional cost.

Password Policy

Use the SAP Signavio Process ManagerSAP Signavio Process Modeler technical user's password and not your SSO password.

- Create a technical, non-SSO, user in User Management. For more information, see User Accounts in the SAP Signavio Process ManagerSAP Signavio Process Modeler Workspace Admin Guide, OR
- Reset the password for your SSO-created user in User Management. For more information, see Changing User Settings in the same guide.

3 Authentication

To access an SAP Signavio Process Manager API, the user must first be authenticated. Authentication is done by dispatching a token request to acquire a JWT (JSON Web Token) and cookie. For this, a user e-mail address and password are required.

Token Request

Dispatch a request with the following information to perform authentication.

Method	POST
Endpoint	`/auth/v1/token`
Headers	• **Content-Type**: `application/x-www-form-urlencoded`
Form Parameters	• **tokenonly**: `true` • **name**: Authenticating user's email address • **password**: Authenticating user's password • **tenant**: Unique identifier of the target workspace. Only necessary if the authenticating user is a member of multiple workspaces.

→ Tip

To find the workspace tenant ID:

1. Log in to SAP Signavio Process Manager or SAP Signavio Process Modeler.
2. In the explorer, on the *Help* menu, select *Workspace information*
3. Copy the *Workspace ID*.

Token Response

The response returns one of the following codes:

- **200 OK**: Authentication successful
- **400 Bad Request**: Request contained a `password` query string parameter, instead of a request body form parameter, which violates security policy.
- **401 Unauthorized**: Authentication failed due to incorrect credentials.

> ⚠ Caution
>
> If you omit the `tokenonly=true` parameter, successful authentication results in a **302 Found** response status, instead of **200 OK**.

Using the Cookie and Authentication Token

From the response, you can access the cookie and the JWT:

- The cookie is returned in a response field named `JSESSIONID`. Copy the corresponding value to obtain the cookie.
- The JWT is returned in the response body. The token forms the entirety of the body's content. Therefore you can obtain the JWT by copying the whole response body.

When subsequently making a request to the API:

- Include the cookie in the request's `Cookie` header with the value '`JSESSIONID={value}`', substituting the cookie value.
- Include the JWT by adding an `x-signavio-id` request header with the token as its value.

> ⓘ Note
>
> The authentication token is valid for 8 hours upon receipt. On reaching this time limit, the token expires and a new token must be obtained.

Base URL

API endpoints are relative to the base URL, which is specific to your region. For the token request, replace the `baseUrl` in the following path:

```
https://{baseUrl}/auth/v1/token
```

The available base URLs are:

Region	Base URL
Australia	`api.au.signavio.cloud.sap`
Canada	`api.ca.signavio.cloud.sap`
EU	`api.eu.signavio.cloud.sap`
Japan	`api.jp.signavio.cloud.sap`
Singapore	`api.sgp.signavio.cloud.sap`
South Korea	`api.kr.signavio.cloud.sap`
USA	`api.us.signavio.cloud.sap`

Examples

Token request, cURL version

Example

```
curl --location 'https://<<baseUrl>>/auth/v1/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'tokenonly=true' \
--data-urlencode 'name=<<username>>' \
--data-urlencode 'password<<password>>' \
--data-urlencode 'tenant=<<tenantId>>'
```

Token request, Python version

Example

```
import requests
def authenticate():
    # Provide values for the following variables
    base_url = ''
    username = ''
    password = ''
    tenantId = ''
    payload = {
        'name': username,
        'password': password,
        'tokenonly': 'true',
        'tenant': tenantId
    }
    headers = {
        'Content-Type': 'application/x-www-form-urlencoded',
    }
    login_url = base_url + '/auth/v1/token'
    # Send request
    response = requests.request("POST", login_url, data=payload,
headers=headers)
    # Decode token from response body
    auth_token = response.content.decode('utf-8')
    return {
        'auth_token': auth_token
    }
authenticate()
```

Related Information

Postman Collection for SAP Signavio Authentication API

4 URLs and Resource Representation

Find out how URLs map to resources in SAP Signavio Process ManagerAPIs as well as how resources are represented.

URL Scheme

All requests use the following URL scheme, where the `{id}` and `{extension}` path segments are optional:

```
https://{baseUrl}/spm/v1/{resourceName}
https://{baseUrl}/spm/v1/{resourceName}/{id}
https://{baseUrl}/spm/v1/{resourceName}/{id}/{extension}
```

The `baseUrl` value is region-specific and depends on your location. The available regions are:

Region	Base URL
Australia	`api.au.signavio.cloud.sap`
Canada	`api.ca.signavio.cloud.sap`
EU	`api.eu.signavio.cloud.sap`
Japan	`api.jp.signavio.cloud.sap`
Singapore	`api.sgp.signavio.cloud.sap`
South Korea	`api.kr.signavio.cloud.sap`
USA	`api.us.signavio.cloud.sap`

Resource Name

Each type of resource has its own name that is used in URLs, for example `directory` (for folders) and `model`.

> Example
>
> Requesting `https://{baseUrl}/spm/v1/directory` returns a workspace's root folders.

Not all resources implement all HTTP methods. Requesting a resource without an ID is only possible with the GET method and not all resources support this method.

ID

The ID is a unique identifier. Requesting a resource URL with an ID returns all information available about the resource. A resource is usually divided into several sub-resources, each identified by its extension.

Extension

The extension defines a sub-resource that contains attributes or other resources. For example, a model has the extension `info` that includes the attributes `title`, `description`, `creation date`.

Resource Representations

HTTP resource representations consist of a JSON object with three properties:

1. `href`: the resource URI, relative to the API base URI
2. `rel`: the resource type
3. `rep`: a JSON object or array representation of the resource content

```
{
  "href": "<resourceUri>",
  "rel": "<resourceType>",
  "rep": <representation>
}
```

Resource URI

Use this relative URI to get or manipulate the resource's data. Every URI starts with a forward slash and the resource name (depending on its type), followed by URL path segments for the ID and an optional extension type.

For example, `/model/e78c0b90010f489fa9025aaa27cc6174/info` is the relative URI of the `info` resource of the model with the ID `e78c0b90010f489fa9025aaa27cc6174`. Use the PUT method on this URI to change the model's title and description.

Resource Type

In most cases, the resource type is the same as the extension and is used to qualify the resource. For example, in `/model/e78c0b90010f489fa9025aaa27cc6174/info` the resource type is `info`.

Representation

The resource representation can either be a JSON object or a JSON array that contains the resource's content. The structure differs from one resource type to another.

5 Resource APIs

5.1 Dictionary

With the Dictionary API, you can create and manage dictionary entries and dictionary categories.

> **Note**
>
> The terms 'dictionary' and 'glossary' can be used interchangeably. In this documentation, the term 'dictionary' is used, while the API endpoints use the term 'glossary'.

In general, the functions provided by this API enable you to perform CRUD actions, in other words creating, retrieving, updating, and deleting dictionary entries and categories.

Prerequisites

Before using the API, you must obtain an authentication token. Refer to Authentication [page 6] for further guidance. Once you have the token, include it in all requests by adding an `x-signavio-id` request header with the token as the header's value.

Requests

Requests to the Dictionary API are dispatched to the following URLs accordingly:

Resource Type	URL
Dictionary entry	`https://{baseUrl}/spm/v1/glossary`
Dictionary category	`https://{baseUrl}/spm/v1/glossarycategory`

In both cases, the base URL is region-specific. The available base URLs are:

Region	Base URL
Australia	`api.au.signavio.cloud.sap`
Canada	`api.ca.signavio.cloud.sap`
EU	`api.eu.signavio.cloud.sap`
Japan	`api.jp.signavio.cloud.sap`
Singapore	`api.sgp.signavio.cloud.sap`

Region	Base URL
South Korea	`api.kr.signavio.cloud.sap`
USA	`api.us.signavio.cloud.sap`

Retrieving Information

You can retrieve dictionary information in one of two ways: a general search or requesting individual resources. This applies to both dictionary entries and dictionary categories.

The first way is to perform a search. Sending a `GET` request to `/spm/v1/glossary` or `/spm/v1/glossarycategory` returns a list of dictionary entries or dictionary categories respectively. In both cases, the search can be refined using additional query parameters.

For example, when searching dictionary entries, you can use parameters to:

- Filter by the entry's type or initial letter.
- Provide a search term.
- Sort or limit the results.

When searching dictionary categories, you can use parameters to:

- Control whether sub-categories or hidden categories are returned.
- Allow administrators to circumvent access restrictions.

> → Tip
>
> When searching, we recommend setting the `include` parameter to `all`. This executes a search in all dictionary items, including meta-attribute values and dictionary items from a category and its subcategories.

The second way to retrieve information is to request resources individually. Sending a `GET` request to `/spm/v1/glossary/{id}` or `/spm/v1/glossarycategory/{id}` returns a specific dictionary entry or dictionary category respectively.

Refer to the API Reference [page 15] or the API Documentation on the SAP Business Accelerator Hub for a comprehensive description of request information.

Creating, Updating, and Deleting Information

The Dictionary API provides a range of actions for keeping your dictionary up-to-date. For both dictionary entries and dictionary categories, new resources can be created and existing resources can be updated or deleted.

Creation is done by dispatching a `POST` request to the URL of the corresponding resource type. Attributes of the new resource are provided in accompanying form data.

→ Tip

To discover custom dictionary attributes' IDs and data structures, use the SAP Signavio Process Manager user interface to create a dictionary item in the corresponding category. Then, retrieve the dictionary entry via the API and use the JSON representation of the dictionary entry as a template.

Updating is done by dispatching a `PUT` request to the URL of the corresponding resource. Similarly to creation, updated attributes are provided in accompanying form data.

⚠ Caution

Existing metadata values aren't deleted if you send a `PUT` request without a `metaDataValues` attribute. This differs from other optional parameters. For example, an existing description of a dictionary entry is deleted if the request body doesn't contain the `description` parameter.

Deleting is done by dispatching a `DELETE` request to the URL of the corresponding resource.

⚠ Caution

Deleting dictionary entries and dictionary categories is permanent. Deleted resources can't be restored.

Refer to the API Reference [page 15] or the API Documentation on the SAP Business Accelerator Hub for a comprehensive description of request information.

Responses

For a complete listing of all response types per endpoint, refer to the API Reference [page 15] or the API Documentation on the SAP Business Accelerator Hub. This includes responses to successful operations as well as potential errors.

Example

This example demonstrates searching for a dictionary entry.

The following script executes a dictionary full-text search and prints the result:

{·} Sample Code

```
import requests
from conf import *
from authentication.authenticate import authenticate
# user variables
query = "test"
limit = 10
offset = 0
# Provide appropriate base URL
base_url = ''
dictionary_url = base_url + "/spm/v1/glossary"
query_url = "{0}?q={1}".format(dictionary_url, query)
auth_data = authenticate()
```

```
# set credentials, response format
cookies = {
    "JSESSIONID": auth_data["jsesssion_ID"]
}
headers = {"Accept": "application/json", "x-signavio-id":
auth_data["auth_token"]}
search_request = requests.get(
    query_url, cookies=cookies, headers=headers, data={limit: limit, offset:
offset}
)
result = str(search_request.content)
print("results: " + result)
```

The following is an example response:

Output Code


```
[
  {
    "rel": "gitem",
    "href": "/glossary/2dfa3594fbf54f4db948e985af47c571",
    "rep": {
      "glossaryId": "8cce891eaabe4294bfa90dfd560be22c",
      "linkedByCount": 0,
      "replacedItemIds": [],
      "linkedCount": 0,
      "formats": {
        "description": []
      },
      "id": "2dfa3594fbf54f4db948e985af47c571",
      "categoryName": "IT Systems",
      "category": "/glossarycategory/a52d17797a2442669a5b4e9c66dcefb0",
      "title": "new IT system entry",
      "color": "#61AEF2",
      "description": "A new IT system.",
      "hasAttachments": true,
      "metaDataValues": {},
      "occurrence": 0,
      "attachments": [
        {
          "id": "c70a72f86e3d4ce6b2df8ff011d8eb37",
          "label": "Signavio",
          "url": "http://www.signavio.com"
        }
      ],
      "language": "de_de"
    }
  },
  {
    "rel": "info",
    "href": "/glossary",
    "rep": {
      "size": 1
    }
  }
]
```


For more examples, refer to the API Documentation on the SAP Business Accelerator Hub.

Related Information

API Documentation on SAP Business Accelerator Hub

5.1.1 API Reference

An overview of the endpoints provided by the Dictionary API.

Common Request Information

All endpoints listed here are relative to `{baseUrl}/spm/v1`. This is explained further in URLs and Resource Representation [page 9].

Unless otherwise indicated, all requests should include the following headers:

`x-signavio-id`	The authentication token
`Cookie`	`JSESSIONID=(jsessionid)`
`Accept`	`application/json`

Common Response Information

Unless otherwise indicated, all successful responses return:

- A status code of `200 OK`
- The header `Content-Type=application/json`

All endpoints listed here could potentially return the following errors:

`Status Code 500, plugin.notEnabled`	The custom dictionary categories feature isn't available for your workspace.
`Status Code 404, no access`	The accessing user doesn't have the permission to access the corresponding dictionary category.

Search for Dictionary Entries

```
GET /glossary
```

Query Parameters

`letter`	Filters by the initial letter of the dictionary entry's title.
`q`	Filters by a full-text search term (ignored when `letter` is specified).
`category`	Filters by entry type - `ORG_UNIT`, `DOCUMENT`, `ACTIVITY`, `STATE`, or `IT_SYSTEM`, representing the pre-defined dictionary categories Organizational Units, Documents, Activities, Events, IT Systems, respectively, or a Dictionary category's ID.
`sort`	If set to `title`, sorts entries by title; otherwise, the search sorts results by a dynamically determined relevance score.
`limit`	Limits the number of results to the given number.
`offset`	Within the returned results, return results starting at offset, useful in combination with the `limit` parameter (paging).
`include`	Optional. If set to `all`, dictionary entries from a category and its subcategories are listed.

> ### Note
>
> Use paging and adhere to entry limits when fetching dictionary entries. You can retrieve a maximum of 1000 records per request. To fetch the first 1000 records, use the query parameter `?limit=1000`. To access the next 1000 records, use the query parameters `?limit=1000&offset=1000`, and continue this pattern for subsequent records. For more information, see Restrictions and Limits [page 82].

Response

Returns a JSON representation of the search results. All dictionary entries are contained in a top-level array (bounded by `[` and `]`), and contain the key-value pair `"rel": "gitem"` (for 'glossary item'). The important fields inside the content of their `rep` object have the same semantics as is defined when creating dictionary entries. They're:

- `id`
- `title`
- `category`
- `description`
- `attachments`

Some fields provide additional information on the entry's category:

- `categoryName`
- `color`

> ### Note
>
> If the category ID in the `category` query string isn't found, the response includes all categories instead of having a 404 Not Found status.

> ### Example
>
> Example response:

```
[
  {
    "rel":"gitem",
```

```
    "href":"/glossary/2dfa3594fbf54f4db948e985af47c571",
    "rep":{
      "glossaryId":"8cce891eaabe4294bfa90dfd560be22c",
      "linkedByCount":0,
      "replacedItemIds":[

      ],
      "linkedCount":0,
      "formats":{
        "description":[

        ]
      },
      "id":"2dfa3594fbf54f4db948e985af47c571",
      "categoryName":"IT Systems",
      "category":"/glossarycategory/a52d17797a2442669a5b4e9c66dcefb0",
      "title":"new IT system entry",
      "color":"#61AEF2",
      "description":"A new IT system.",
      "hasAttachments":true,
      "metaDataValues":{

      },
      "occurrence":0,
      "attachments":[
        {
          "id":"c70a72f86e3d4ce6b2df8ff011d8eb37",
          "label":"Signavio",
          "url":"http://www.signavio.com"
        }
      ],
      "language":"de_de"
    }
  },
  {
    "rel":"info",
    "href":"/glossary",
    "rep":{
      "size":1
    }
  }
]
```


Potential Errors

`Status Code 500, glossary.cannotParseQuery`	The query parameter `q` is invalid.

Retrieve a Specific Dictionary Entry

```
GET /glossary/<id>
```

Response

Returns a JSON object representing the dictionary entry.

Example

Example response:


```
{
```

```
  "granted_revision":"/itemrevision/47225a21d39b429ebe2ead3a191a4676",
  "granted_revision_number":1,
  "formats":{
    "description":[

    ]
  },
  "color":"#800000",
  "granted_revision_user":"",
  "author":"/user/36821979e14b417dac355a748f896874",
  "granted_revision_date":"2017-10-17 10:44:46 +0200",
  "description":"The new back end system needs to support the legacy data
format. Otherwise, the integration with older 3rd-party integrations breaks.",
  "language":"en",
  "replacedItemIds":[

  ],
  "type":"UNDEFINED",
  "authorCompany":"Signavio",
  "title":"Support legacy data format",
  "categoryName":"Requirements",
  "headRevisionNum":1,
  "authorName":"John Doe",
  "metaDataValues":{
    "meta-summary":""
  },
  "comment":"",
  "id":"e44ace669662486286c70183c8655388",
  "category":"/glossarycategory/e8be7975fe4446138e5495a6a3e9c13a",
  "hasAttachments":false,
  "glossaryId":"be5e25dd63d747f5bd8635b0e0c2f38c",
  "updated":"2017-10-17 10:44:46 +0200",
  "headRevision":"/itemrevision/47225a21d39b429ebe2ead3a191a4676"
}
```


Retrieve a Specific Dictionary Entry with Metadata

This operation retrieves the specified dictionary entry along with additional meta information, including the category containing the entry.

```
GET /glossary/<id>/info
```

Response

Returns a JSON object representing the dictionary entry.

Example

Example response:


```
[
  {
    "rel":"link",
    "href":"/glossary/e44ace669662486286c70183c8655388/link",
    "rep":[

    ]
  },
  {
    "rel":"subscription",
```

```
    "href":"/glossary/e44ace669662486286c70183c8655388/subscription",
    "rep":{
      "sendingInterval":"NONE"
    }
  },
  {
    "rel":"outgoings",
    "href":"/glossary/e44ace669662486286c70183c8655388/outgoings",
    "rep":[

    ]
  },
  {
    "rel":"info",
    "href":"/glossary/e44ace669662486286c70183c8655388/info",
    "rep":{
      "granted_revision":"/itemrevision/47225a21d39b429ebe2ead3a191a4676",
      "granted_revision_number":1,
      "formats":{
        "description":[

        ]
      },
      "color":"#800000",
      "granted_revision_user":"",
      "author":"/user/36821979e14b417dac355a748f896874",
      "granted_revision_date":"2017-10-17 10:44:46 +0200",
      "description":"The new back end system needs to support the legacy data
format. Otherwise, the integration with older 3rd-party integrations breaks.",
      "language":"en",
      "replacedItemIds":[

      ],
      "type":"UNDEFINED",
      "authorCompany":"Signavio",
      "title":"Support legacy data format",
      "categoryName":"Requirements",
      "headRevisionNum":1,
      "authorName":"John Doe",
      "metaDataValues":{

      },
      "comment":"",
      "id":"e44ace669662486286c70183c8655388",
      "category":"/glossarycategory/e8be7975fe4446138e5495a6a3e9c13a",
      "hasAttachments":false,
      "glossaryId":"be5e25dd63d747f5bd8635b0e0c2f38c",
      "updated":"2017-10-17 10:44:46 +0200",
      "headRevision":"/itemrevision/47225a21d39b429ebe2ead3a191a4676"
    }
  },
  {
    "rel":"parents",
    "href":"/glossary/e44ace669662486286c70183c8655388/parents",
    "rep":[
      {
        "rel":"cat",
        "href":"/glossarycategory/e8be7975fe4446138e5495a6a3e9c13a",
        "rep":{
          "decisionInput":false,
          "color":"#800000",
          "hidden":false,
          "workflowSyncUpToDate":false,
          "publishingMode":"AUTO",
          "type":"UNDEFINED",
          "isStandard":false,
          "workflowSyncEnabled":false,
          "name":"Requirements",
```

```
            "decisionDataObjectKey":"",
            "id":"/glossarycategory/e8be7975fe4446138e5495a6a3e9c13a",
            "glossaryId":"be5e25dd63d747f5bd8635b0e0c2f38c",
            "order":1,
            "linkingPublishedModelsUpdateMode":"UPDATE"
          }
        }
      ]
    }
  ]
```

Retrieve a List of All Dictionary Categories

```
GET /glossarycategory
```

Query Parameters

`allCategories`	(Optional) If set to `true`, all categories' representations are returned at once, without the need to send additional queries for sub-categories.
`showHidden`	(Optional) If set to `true`, categories configured as `hidden` are included in the result.
`considerAllPrivilege`	Optional. To circumvent access restrictions as a member of the workspace's administrators group, set this parameter to `true`.

Response

Returns a list of all dictionary categories. The content data is partly made up of information as described in creating dictionary categories. Additionally, each category contains the following information:

`ChildCategories`	Array of prefixed IDs of categories that have this category as parent category.
`ChildCategoryCount`	The number of child categories.
`GlossaryId`	ID of the workspace's Dictionary (not relevant for API access).
`ItemCount`	The number of dictionary entries in this category (excluding those in sub-categories).
`Items`	Array of IDs of the contained dictionary entries (deprecated).
`OldCategory`	If the category corresponds to one of the six standard categories, this field is set. For example, some reports consider the content of these categories. The following `oldCategories` exist: • `ORG_UNIT` - Organizational units • `DOCUMENT` - Documents • `ACTIVITY` - Activities

- `STATE` - Events
- `IT_SYSTEM` - IT systems
- `NONE` - Everything else

Example

Example response:


```
[
  {
    "rel":"cat",
    "href":"/glossarycategory/b64046bf36654833bdf66231bb1f3cd7",
    "rep":{
      "childCategories":[

      ],
      "id":"/glossarycategory/b64046bf36654833bdf66231bb1f3cd7",
      "order":0,
      "items":[

      ],
      "oldCategory":"ORG_UNIT",
      "color":"#C22F1E",
      "hidden":false,
      "name":"Organizational Units",
      "childCategoryCount":0,
      "itemCount":0,
      "glossaryId":"8cce891eaabe4294bfa90dfd560be22c"
    }
  },
  {
    "rel":"cat",
    "href":"/glossarycategory/ec6e5cba209e44f9a6a072860622fc0a",
    "rep":{
      "childCategories":[

      ],
      "id":"/glossarycategory/ec6e5cba209e44f9a6a072860622fc0a",
      "order":1,
      "items":[

      ],
      "oldCategory":"DOCUMENT",
      "color":"#F0AD26",
      "hidden":false,
      "name":"Documents",
      "childCategoryCount":0,
      "itemCount":0,
      "glossaryId":"8cce891eaabe4294bfa90dfd560be22c"
    }
  },
  {
    "rel":"cat",
    "href":"/glossarycategory/ab5852a9aecd45bba7c3bf40635d5c3f",
    "rep":{
      "childCategories":[

      ],
      "id":"/glossarycategory/ab5852a9aecd45bba7c3bf40635d5c3f",
      "order":2,
      "items":[

      ],
      "oldCategory":"ACTIVITY",
      "color":"#B8CC00",
```

```
      "hidden":false,
      "name":"Activities",
      "childCategoryCount":0,
      "itemCount":0,
      "glossaryId":"8cce891eaabe4294bfa90dfd560be22c"
    }
  },
  {
    "rel":"cat",
    "href":"/glossarycategory/85686412c24e4dd2a04f03df07345d8c",
    "rep":{
      "childCategories":[
        "/glossarycategory/02407da1bca6441e82192cf97560d2d3"
      ],
      "id":"/glossarycategory/85686412c24e4dd2a04f03df07345d8c",
      "order":3,
      "items":[

      ],
      "oldCategory":"STATE",
      "color":"#00A7AC",
      "hidden":false,
      "name":"Events",
      "childCategoryCount":1,
      "itemCount":0,
      "glossaryId":"8cce891eaabe4294bfa90dfd560be22c"
    }
  },
  {
    "rel":"cat",
    "href":"/glossarycategory/02407da1bca6441e82192cf97560d2d3",
    "rep":{
      "childCategories":[

      ],
      "id":"/glossarycategory/02407da1bca6441e82192cf97560d2d3",
      "order":0,
      "items":[

      ],
      "color":"#00A7AC",
      "parentCategory":"/glossarycategory/85686412c24e4dd2a04f03df07345d8c",
      "hidden":false,
      "name":"mySecondCustomCategory",
      "childCategoryCount":0,
      "itemCount":0,
      "glossaryId":"8cce891eaabe4294bfa90dfd560be22c"
    }
  },
  {
    "rel":"cat",
    "href":"/glossarycategory/a52d17797a2442669a5b4e9c66dcefb0",
    "rep":{
      "childCategories":[

      ],
      "id":"/glossarycategory/a52d17797a2442669a5b4e9c66dcefb0",
      "order":4,
      "items":[

      ],
      "oldCategory":"IT_SYSTEM",
      "color":"#61AEF2",
      "hidden":false,
      "name":"IT Systems",
      "childCategoryCount":0,
      "itemCount":0,
      "glossaryId":"8cce891eaabe4294bfa90dfd560be22c"
```

```
    }
  },
  {
    "rel":"cat",
    "href":"/glossarycategory/d442e84547584812be868602f5901aee",
    "rep":{
      "childCategories":[

      ],
      "id":"/glossarycategory/d442e84547584812be868602f5901aee",
      "order":5,
      "items":[
        "be19b2b56a714e0e85f2230c7bcb339d",
        "7426f361937a475997772fadddd70781",
        "f7b9ea1625174c138cb0b92bead675f3"
      ],
      "oldCategory":"NONE",
      "color":"#7F7F7F",
      "hidden":false,
      "name":"Others",
      "childCategoryCount":0,
      "itemCount":3,
      "glossaryId":"8cce891eaabe4294bfa90dfd560be22c"
    }
  },
  {
    "rel":"cat",
    "href":"/glossarycategory/2b73a954c03c4e6182fac8c896513bb4",
    "rep":{
      "childCategories":[

      ],
      "id":"/glossarycategory/2b73a954c03c4e6182fac8c896513bb4",
      "order":6,
      "items":[
        "4cfba70728694e4da6da8fa5279a5d39",
        "5d10dcbbc2c5450f865e337d6403d9b2",
        "a0d9c01f153e451d9a528469984360c6"
      ],
      "color":"#fffaaa",
      "hidden":false,
      "name":"myFirstCustomCategory",
      "childCategoryCount":0,
      "itemCount":3,
      "glossaryId":"8cce891eaabe4294bfa90dfd560be22c"
    }
  }
]
```


Potential Errors

Status Code 500, `glossary.cannotParseQuery`:	Internal issue, contact support.
Status Code 500, `usermanagement.apionlyaccess`:	Access for users using a web browser isn't allowed. Make sure the HTTP headers are set correctly to `Accept: application/json`.

Retrieve a Specific Dictionary Category

```
GET /glossarycategory/<id>
```

Query Parameters

`allCategories`	(Optional) If set to `true`, all categories' representations are returned at once, without the need to send additional queries for sub-categories.
`showHidden`	(Optional) If set to `true`, categories configured as `hidden` are included in the result.
`considerAllPrivilege`	Optional. To circumvent access restrictions as a member of the workspace's administrators group, set this parameter to `true`.

Response

Returns a list of this category's sub-categories (`rel` is `cat`), and an object with information on the category itself (`rel` is `info`). Sending a `GET` request to `/glossarycategory/{id}/info` fetches the information object only.

Example

Example response:


```
[
  {
    "rel":"priv",
    "href":"/glossarycategory/851e9da35ccc4931969efe5c7bc10a40/priv",
    "rep":[
      "glossary.write",
      "all"
    ]
  },
  {
    "rel":"info",
    "href":"/glossarycategory/851e9da35ccc4931969efe5c7bc10a40/info",
    "rep":{
      "id":"/glossarycategory/851e9da35ccc4931969efe5c7bc10a40",
      "order":2,
      "oldCategory":"ACTIVITY",
      "color":"#FFFFFF",
      "hidden":false,
      "name":"Aktivitäten",
      "childCategoryCount":2,
      "itemCount":1,
      "glossaryId":"1435c148625d411ba68bd711185db688"
    }
  },
  {
    "rel":"cat",
    "href":"/glossarycategory/cdc39ba2cd4b49e883d89bf20fcdce3c",
    "rep":{
      "childCategories":[

      ],
      "id":"/glossarycategory/cdc39ba2cd4b49e883d89bf20fcdce3c",
      "order":0,
      "color":"#FFFFFF",
      "parentCategory":"/glossarycategory/851e9da35ccc4931969efe5c7bc10a40",
```

```
      "hidden":false,
      "name":"child1",
      "childCategoryCount":0,
      "itemCount":0,
      "glossaryId":"1435c148625d411ba68bd711185db688"
    }
  },
  {
    "rel":"cat",
    "href":"/glossarycategory/5d5b871a4f0e4957aaf82281d5baed2b",
    "rep":{
      "childCategories":[

      ],
      "id":"/glossarycategory/5d5b871a4f0e4957aaf82281d5baed2b",
      "order":1,
      "color":"#FFFFFF",
      "parentCategory":"/glossarycategory/851e9da35ccc4931969efe5c7bc10a40",
      "hidden":false,
      "name":"child2",
      "childCategoryCount":0,
      "itemCount":0,
      "glossaryId":"1435c148625d411ba68bd711185db688"
    }
  }
]
```


Create a Dictionary Entry

```
POST /glossary
```

Request Headers

`Content-Type`	`application/x-www-form-urlencoded`

Form Parameters

`title`	The entry's title
`category`	The desired dictionary category's ID
`description`	(Optional) A description of the entry
`force`	(Optional) Set to `true` if you want to create a new dictionary entry even if an entry with the same name already exists.
`attachments`	Add a single attachment as an object. For multiple attachments, add them as an array of objects: `[ {"url":"http://www.example.com", "label":"Example"} ]`

`metaDataValues` (Optional) A JSON object containing ID-to-value mappings of custom attributes, for example:

```
{
  "meta-my-single-line-text": "one",
  "meta-my-multi-line-text": "two\nthree\nfour",
  "meta-my-dictionary-link": "/glossary/
3aabd26a654c4123a71c597210810000",
  "meta-my-date": "2018-12-24T16:46:00.000Z",
  "meta-my-number": "42",
  "meta-my-drop-down": "ci1545668214217710650210",
  "meta-my-url": {
    "label": "Example",
    "url": "http://www.example.com/"
  },
  "meta-my-boolean": true
}
```

Response

Returns a JSON object containing the created dictionary item.

> **Note**
>
> To discover custom dictionary attributes' IDs and data structures, use the SAP Signavio Process Manager user interface to create a dictionary item in the corresponding category. Then retrieve the dictionary entry via the API, for example by executing a sufficiently specific search and use the JSON representation of the dictionary entry as a template.

> **Example**
>
> Example response:


```
{
  "rel":"gitem",
  "href":"/glossary/2dfa3594fbf54f4db948e985af47c571",
  "rep":{
    "glossaryId":"8cce891eaabe4294bfa90dfd560be22c",
    "linkedByCount":0,
    "replacedItemIds":[

    ],
    "linkedCount":0,
    "formats":{
      "description":[

      ]
    },
    "id":"2dfa3594fbf54f4db948e985af47c571",
    "categoryName":"IT Systems",
    "category":"/glossarycategory/a52d17797a2442669a5b4e9c66dcefb0",
    "title":"new IT system entry",
    "color":"#61AEF2",
    "description":"A new IT system.",
    "hasAttachments":true,
    "metaDataValues":{
      "meta-summary":""
    },
    "occurrence":0,
    "attachments":[
      {
        "id":"c70a72f86e3d4ce6b2df8ff011d8eb37",
```

```
        "label":"Signavio",
        "url":"http://www.signavio.com"
      }
    ],
    "language":"de_de"
  }
}
```

Potential Errors

Status Code 409, An error occurred (glossary.duplicateItem)	`{ "message": "An error occurred (glossary.duplicateItem)", "title": "(TITLE)" }` An entry with the title `(TITLE)` already exists. Re-send with form parameter `force=true` to create the entry anyway. The existing item won't be overwritten.
Status Code 500, UpdatingModelFailed	An error occurred while trying to manipulate additional data related to the dictionary entry.

Update a Dictionary Entry

```
PUT /glossary/<id>/info
```

Request Headers

Content-Type	`application/x-www-form-urlencoded`

Form Parameters

title	The entry's title
category	The desired dictionary category's ID
description	(Optional) A description of the entry
force	(Optional) Set to `true` if you want to create a new dictionary entry even if an entry with the same name already exists.
attachments	Add a single attachment as an object. For multiple attachments, add them as an array of objects: `[ {"url":"http://www.example.com", "label":"Example"} ]`

`metaDataValues` (Optional) A JSON object containing ID-to-value mappings of custom attributes, for example:

```
{
  "meta-my-single-line-text": "one",
  "meta-my-multi-line-text": "two\nthree\nfour",
  "meta-my-dictionary-link": "/glossary/
3aabd26a654c4123a71c597210810000",
  "meta-my-date": "2018-12-24T16:46:00.000Z",
  "meta-my-number": "42",
  "meta-my-drop-down": "ci1545668214217710650210",
  "meta-my-url": {
    "label": "Example",
    "url": "http://www.example.com/"
  },
  "meta-my-boolean": true
}
```

> **Note**
>
> Existing metaDataValues aren't deleted if you send a `PUT` request without a `metaDataValues` attribute. This differs from other optional parameters. For example, an existing description of a dictionary entry is deleted if the request body doesn't contain the `description` parameter.

Delete a Dictionary Entry

> **Caution**
>
> Deleting dictionary entries is permanent, deleted dictionary entries can't be restored.

```
DELETE /glossary/<id>
```

Response

A successful response returns:

```
{"success" : true}
```

Potential Errors

`Status Code 404, glossary.notFound`	No entry with the given ID exists.

Create a Dictionary Category

```
POST /glossarycategory
```

Request Headers

`Content-Type`	`application/x-www-form-urlencoded`

Form Parameters

`name`	The category's name
`color`	A HEX color to display with the category, for example `#800000` (dark red)
`order`	The order the category takes among categories within the same context
`parentCategory`	The category's parent category. Leave the parameter unset to create the category on the top-most level.
`hidden`	If set to `true`, the category is invisible to all users.
`restricted`	If set to `true`, the category may only be accessible to users who own a specific privilege.

Response

Returns the newly created category.

Example

Example response:

```
{
  "rel":"cat",
  "href":"/glossarycategory/02407da1bca6441e82192cf97560d2d3",
  "rep":{
    "childCategories":[

    ],
    "id":"/glossarycategory/02407da1bca6441e82192cf97560d2d3",
    "order":0,
    "items":[

    ],
    "color":"#00A7AC",
    "parentCategory":"/glossarycategory/85686412c24e4dd2a04f03df07345d8c",
    "hidden":false,
    "name":"mySecondCustomCategory",
    "childCategoryCount":0,
    "itemCount":0,
    "glossaryId":"8cce891eaabe4294bfa90dfd560be22c"
  }
}
```

Potential Errors

`Status Code 500, glossaryCategory.noValidOrOnlyExistingNamesGiven`	The chosen name is most likely not given, or given as an empty or whitespace character string. Choose a different string and try again.
`Status Code 500, glossarycategory.nameAlreadyExistsException`	There's already a dictionary category with this name. Choose a different name, and try again.

Status Code 500, glossarycategory.cannotHaveThreeLevelsOfCategories	There may not be three levels of dictionary categories. Categories at the top level may only have sub-categories without further sub-categories.

Update a Dictionary Category

```
PUT /glossarycategory/<id>
```

Request Headers

Content-Type	`application/x-www-form-urlencoded`

Request Parameters

name	The category's name
color	A HEX color to display with the category, for example `#800000` (dark red)
order	The order the category takes among categories within the same context
parentCategory	The category's parent category. Leave the parameter unset to create the category on the top-most level.
hidden	If set to `true`, the category is invisible to all users.
restricted	If set to `true`, the category may only be accessible to users who own a specific privilege.

Potential Errors

Status Code 500, glossarycategory.cannotHideOrRestrictDefaultCategory	The new 'Others' standard category can't be hidden nor can access to it be restricted.
Status Code 500, glossarycategory.cannotHaveThreeLevelsOfCategories	There may not be three levels of dictionary categories. Categories at the top level may only have sub-categories without further sub-categories.
Status Code 500, glossarycategory.cannotMoveStandardCategory	The five former standard categories and the new 'Others' category can't be moved to another parent category: • Organizational Units • Documents • Activities • Events

- IT Systems
- Others

Delete a Dictionary Category

> ⚠ Caution
>
> Deleting dictionary categories is permanent, deleted dictionary entries can't be restored.

```
DELETE /glossarycategory/<id>
```

Query Parameters

`moveContent` If set to `true`, before deleting the category with all its sub-categories, the content is moved to the 'Others' category. Otherwise, sub-categories and all contained dictionary entries and the sub-categories' contained dictionary entries are deleted.

Response

A successful response returns:

```
{"success" : true}
```

Potential Errors

`Status Code 500, glossarycategory.cannotMoveStandardCategory`

The five former standard categories and the new 'Others' category can't be moved to another parent category:

- Organizational Units
- Documents
- Activities
- Events
- IT Systems
- Others

5.1.2 Custom Attribute Values

In SAP Signavio Process Manager, you can create custom attributes.

Several different data types are supported:

- Single-line text
- Multi-line text

- Dictionary link
- Document/URL
- Boolean

All types, except for Boolean and multi-line text, can be lists.

Examples

Example Values of Data Types

Type	Example Value
Single-line text	`"singlelinetext" : "a single line"`
Single-line text (list)	`"singlelinetextlist" : [ "single line text item 1", "single line text item 2" ]`
Multi-line text	`"multilinetext" : "Many lines: \nLine two. \n\n\nLine three after an empty line."`
Dictionary link	`"dictionarylink" : [{ "id" : "43e92bb841934ab796cae15f51f05cde", "category" : "/glossarycategory/c82004460f7f413e9612b23f8d8e3840", "title" : "Signavio" }]`
Dictionary link (list)	`"dictionarylinklist" : [{ "id" : "43e92bb841934ab796cae15f51f05cde", "category" : "/glossarycategory/c82004460f7f413e9612b23f8d8e3840", "title" : "Signavio" }, { "id" : "745fb87e7396450493cb7d7bdb1a482c", "category" : "/glossarycategory/886ead53075b4e07926c9a3ee1a62ef1", "title" : "SAPERION Workflow" }]`
Document URL	`"documenturl" : { "label" : "Signavio Homepage", "url" : "http://www.signavio.com" }`

Type	Example Value
Document URL (list)	`"documenturllist" : [{ "label" : "Signavio homepage", "url" : "http://www.signavio.com" }, { "label" : "Signavio process editor", "url" : "https://editor.signavio.com" }]`
Boolean	`"booleanattributeid" : true`

5.1.3 Creating Custom Attributes for Dictionary Categories

The `/spm/v1/meta` endpoint enables you to add custom attributes to a dictionary category.

To add a custom attribute, dispatch a `POST` request to `/spm/v1/meta` with the following form parameters:

`glossaryBindings`	The dictionary category for which the custom attribute should be created.
`name`	The desired name of the custom attribute.
`type`	The type of the custom attribute.
`isList`	Optional. If the custom attribute should be a list.
`category`	Optional. Restricts the attribute for a specific category for the custom attribute of type "`MetaDataGlossaryLink`".

If the request is successful, a response is returned with the following:

- A response code of `200 OK`.
- A response body containing a JSON representation of the custom attribute.

Example

The following Python script creates a new custom attribute of type "`MetaDataStringInfo`" for dictionary category:

```
import requests
from conf import *
from authentication.authenticate import authenticate
# user variables
glossaryBindings = {"category": "<category_ID>", "order": 5}
name = "A sample custom attribute"
attributeType = "MetaDataStringInfo"
isGlossaryDefinition = True
parent = "/meta"
isList = True
custom_attribute_url = base_url + "/spm/v1/meta"
```

```
auth_data = authenticate()
# set credentials, response format
cookies = {
    "JSESSIONID": auth_data["jsesssion_ID"]
}
headers = {"Accept": "application/json", "x-signavio-id":
auth_data["auth_token"]}
create_custom_attribute_request = requests.post(
    custom_attribute_url,
    cookies=cookies,
    headers=headers,
    data={
        "glossaryBindings": glossaryBindings,
        "name": name,
        "type": attributeType,
        "isGlossaryDefinition": isGlossaryDefinition,
        "parent": parent,
        "isList": isList,
    },
)
print("creating entry: " + str(create_custom_attribute_request.content))
```

Example response:

```
{
  "rel": "meta",
  "href": "/meta/meta-asamplecustomattributefordi",
  "rep": {
    "glossaryBindings": [
      { "category": "9ea30b047e75497d919028c110b2f980", "order": 9 }
    ],
    "lineWrap": false,
    "defaultValue": "",
    "length": 0,
    "description": "",
    "isGlossaryDefinition": true,
    "type": "MetaDataStringInfo",
    "isList": true,
    "multilanguage": false,
    "stencilsetBindings": [],
    "readonly": false,
    "name": "A sample custom attribute for dictionary",
    "id": "meta-asamplecustomattributefordi"
  }
}
```

5.1.4 Multi-Language Dictionary Attributes

Dictionary attributes can be stored in multiple languages. You can add and retrieve them via the API.

Retrieving the Available Languages

Sending a `GET` to `/spm/v1/configuration?category=all` returns a list of workspace configurations as defined by the workspace administrator, including the content languages.

In the response, check the object whose `href` ends with "languages". Its corresponding `rep` object contains an escaped JSON string in the `value` property, which represents the currently defined list of content languages.

Checking If an Attribute Is Multi-Language

If an attribute has been created to support multiple languages, it has a property of `multilanguage=true`. This property is returned in the response to a `GET /spm/v1/meta` request.

Note that this doesn't necessarily mean that translations are available in all configured languages, only that the attribute supports adding them.

Retrieving Multi-Language Attributes

The default language of an attribute is determined by the chosen locale, for example `"en_us"` or `"de_de"`. When an attribute supports multiple languages, the translations of its values are accessible via a property whose name follows this format: `<attribute>_<locale>`.

For example, when retrieving a dictionary entry via `/spm/v1/glossary`, assuming a default locale of `"en_us"` and `"de_de"` as a secondary language:

1. The `description` property in the response contains the English version.
2. If a German translation was provided, its translation is contained in the `description_de_de` property.

Adding Multi-Language Attributes

Similarly to when retrieving them, adding multi-language attributes requires suffixing their name with the locale of the corresponding translation.

When creating a dictionary entry with `POST /spm/v1/glossary`, you can supply as a parameter:

1. The attribute value in the default language, for example in the `description` parameter.
2. The translation of the attribute value into any secondary languages by adding the locale as a suffix to the attribute name, for example in the `description_de_de` parameter.

5.2 Directory

With the Directory API, you can manage both directories and directory metadata as well as publish content in SAP Signavio Process Manager.

In general, the functions provided by this API enable you to perform CRUD actions on directories and their metadata, in other words creating, retrieving, updating, and deleting them.

Prerequisites

Before sending requests, ensure that you followed authentication procedures as described in Authentication [page 6] to obtain an authentication token. Include this token in all requests by adding an `x-signavio-id` request header with the token as the header's value.

Requests

Requests to the Directory API are sent to `https://{baseUrl}/spm/v1/directory`. The base URL is region-specific and should be substituted by the correct base URL for your region:

Region	Base URL
Australia	`api.au.signavio.cloud.sap`
Canada	`api.ca.signavio.cloud.sap`
EU	`api.eu.signavio.cloud.sap`
Japan	`api.jp.signavio.cloud.sap`
Singapore	`api.sgp.signavio.cloud.sap`
South Korea	`api.kr.signavio.cloud.sap`
USA	`api.us.signavio.cloud.sap`

Managing Information

You can use this API to retrieve directory metadata. Endpoints are provided to obtain metadata for individual directories or all items within a specific directory.

You can also send requests for managing directories individually, such as creating, updating, and deleting them.

> ### Example
>
> The endpoint `PUT /spm/v1/directory/{id}` allows you to move the directory identified by `{id}` to a new location.

Refer to the API Reference [page 38] or the API Documentation on the SAP Business Accelerator Hub for comprehensive descriptions of each request and its parameters.

Responses

For a complete listing of all response types per endpoint, refer to the API Reference [page 38] or the API Documentation on the SAP Business Accelerator Hub. This includes responses to successful operations as well as potential errors.

Example

The following code example gets the root folder's meta data:

Source Code

```
import json, requests
from conf import *
from authentication.authenticate import authenticate
dir_url = base_url + '/spm/v1/directory'
auth_data = authenticate()
# set credentials, response format
cookies = {
    'JSESSIONID': auth_data['jsesssion_ID']
}
headers = {
    'Accept': 'application/json',
    'x-signavio-id':  auth_data['auth_token']
}
get_dir_meta_request = requests.get(dir_url,
    cookies=cookies,
    headers=headers)
print(get_dir_meta_request)
```

The following is an example response:

Output Code

```
[{
    "rel": "dir",
    "href": "/directory/961aa342dc4d4115b9ded669517a18d8",
    "rep": {
        "type": "public",
        "visible": true,
        "created": "2016-03-07 10:34:30 +0100",
        "description": "",
        "name": "Shared Documents",
        "deleted": false
    }
}, {
    "rel":"dir",
    "href":"/directory/ced8a7b57de04de4ba75a57193442ac8",
    "rep": {
        [...],
        "type":"private",
    }
}, {
    "rel":"dir",
    "href":"/directory/c15ac64587f740b0a9eba2cbac5342e3",
        "rep": {
        [...],
        "type":"trash",
```

```
    }
}, {
    "rel":"glos",
    "href":"/glossarymanager/5999186c164f42dfbd62683459cefb2d",
    "rep": {
        "visible":true,
        "uri":"/p/glossary?originId=5999186c164f42dfbd62683459cefb2d"
    }
}]
```

Related Information

API Documentation on SAP Business Accelerator Hub

5.2.1 API Reference

An overview of the endpoints provided by the Directory API.

Common Request Information

All endpoints listed here are relative to `{baseUrl}/spm/v1`. This is explained further in URLs and Resource Representation [page 9].

Unless otherwise indicated, all requests should include the following headers:

`x-signavio-id`	The authentication token
`Cookie`	`JSESSIONID=(jsessionid);LBROUTEID=(lbrouteid)`
`Accept`	`application/json`

Common Response Information

Unless otherwise indicated, all successful responses return:

- A status code of `200 OK`
- The header `Content-Type=application/json`

Retrieve the Workspace Root Folders' Metadata

```
GET /directory
```

Response

Returns four resources, which represent a workspace's root folders:

- `Shared Documents` folder
- `My Documents` folder
- `Trash`
- `Dictionary`

The actual folder names may differ depending on the workspace language. Each resource's `rel` and `type` properties differ depending on the resource type. The folders `Shared Documents`, `My Documents`, and `Trash` have the `rel` field set to `dir` and are tagged with a `type` field, which can be either `public`, `private` or `trash`. Such type fields are only defined for root folder structures. For the Dictionary, the `rel` field is set to `glos` (from 'glossary').

→ Tip

Although the `href` property values start with a `/`, they're relative to the base path rather than being absolute URL paths.

Example

Example response

```
[{
    "rel": "dir",
    "href": "/directory/961aa342dc4d4115b9ded669517a18d8",
    "rep": {
        "type": "public",
        "visible": true,
        "created": "2016-03-07 10:34:30 +0100",
        "description": "",
        "name": "Shared Documents",
        "deleted": false
    }
}, {
    "rel":"dir",
    "href":"/directory/ced8a7b57de04de4ba75a57193442ac8",
    "rep": {
        [...],
        "type":"private",
    }
}, {
    "rel":"dir",
    "href":"/directory/c15ac64587f740b0a9eba2cbac5342e3",
        "rep": {
        [...],
        "type":"trash",
    }
},{
    "rel":"glos",
    "href":"/glossarymanager/5999186c164f42dfbd62683459cefb2d",
    "rep": {
        "visible":true,
        "uri":"/p/glossary?originId=5999186c164f42dfbd62683459cefb2d"
    }
}]
```

Retrieve a Folder's Contents

```
GET /directory/<id>
```

Response

Returns a JSON array listing the related resources of the folder.

Example


```
[
  {
    "rel": "info",
    "href": "/directory/961aa342dc4d4115b9ded669517a18d8/info",
    "rep": {
      "visible": true,
      "created": "2016-08-27 13:23:09 +0200",
      "description": "",
      "name": "Shared Documents",
      "type": "public",
      "deleted": false
    }
  },
  {
    "rel": "parents",
    "href": "/directory/961aa342dc4d4115b9ded669517a18d8/parents",
    "rep": {}
  },
  {
    "rel": "dir",
    "href": "/directory/40d7fed893894624878a43f5e99943e7",
    "rep": {}
  },
  {
    "rel": "mod",
    "href": "/model/fea8a2ac6d184bf2852ecbaa11622a51",
    "rep": {
      "rev": 1,
      "parent": "/directory/961aa342dc4d4115b9ded669517a18d8",
      "granted_revision": "",
      "type": "Prozesslandkarte",
      "deleted": false,
      "namespace": "http://www.signavio.com/stencilsets/processmap#",
      "author": "/user/d2e2757a6e6f4856a7843e6ceb69a20e",
      "revision": "/revision/d44c345964624d4483b3c61399e4f4be",
      "isLicensedStencilSet": true,
      "created": "2016-04-07 16:02:59 +0200",
      "updated": "2016-04-07 16:02:59 +0200",
      "description": "The Description",
      "name": "Value Chain: Procurement",
      "comment": "",
      "parentName": "Shared Documents"
    }
  }
]
```


Potential Errors

Status Code 403	No permission to list the directory content.
Status Code 404	The directory doesn't exist.

Navigating to Root Folders

The root folder structures contain a link to their actual folder representation in the `href` property. In the case of the example response to this request, a `GET` request to `/directory/961aa342dc4d4115b9ded669517a18d8` returns the folder structure of the `Shared Documents` folder. Generally, a folder's JSON representation is returned by `GET /directory/<id>`. This folder structure is an array that contains the items `info`, `parent`, `dir` and `mod`. The `GET` request returns a JSON array with the following relevant items:

- The `info` item contains in especially the name, the creation date and a description.
- The `parents` item contains a reference (`href`) to the `parents` resource. A `GET` request to this reference returns an array listing all parent folders and their parents, recursively. Each item keeps a reference to the actual resource representation in the `href` field and also a short description. The order of the parents is bottom-up, so the direct parent folder is the first item of the array, the parent's parent folder is second and so on.

  > → Tip
  >
  > The `parents` item lists all the folder's ancestors, not just its direct parent folder.

- Each `dir` item represents a sub-folder, and contains a reference to the folder resource of this sub-folder.
- Each `mod` item represents a model within the folder, and contains a reference to the corresponding model resource.

Navigating to Sub-Folders

You can navigate through the folder tree step by step, by using the `href` references of the `dir` and `parents` items. The following example explains navigating to the folder `Obtainment - Example processes` and getting the model structure of the diagram `obtainment example`:

1. `GET /directory` returns the four root folder structures. The folder tagged `public` has the `href` value `/directory/961aa342dc4d4115b9ded669517a18d8`.
2. `GET /directory/961aa342dc4d4115b9ded669517a18d8` returns the folder structure of `Shared Documents`. It contains a `dir` item for each sub-folder of `Shared Documents`. One of the sub-folders is named `Obtainment - Example processes` and has the `href` value `/directory/40d7fed893894624878a43f5e99943e7`.
3. `GET /directory/40d7fed893894624878a43f5e99943e7` finally returns the `Obtainment - Example processes` folder. It contains multiple items of the type `mod` that represent the model diagrams in the folder. One of them is named `obtainment example` and has the `model_id` `98e75r98wehjt9o54r0w7805u`.
4. `GET /model/98e75r98wehjt9o54r0w7805u` returns the model object of the diagram.

`GET /dictionary/model/<id>` returns a model's object structure.

Create a New Folder

```
POST /directory
```

Request Headers

`Content-Type`	`application/x-www-form-urlencoded`

Form Parameters

name	The new folder's name
parent	A reference to the parent folder that will contain the new folder, in the format `/directory/<parent_folder_id>`

Response

Returns the successfully created folder and returns its meta-data.

> → Remember
>
> The expected HTTP response status is `200 OK` rather than `201 Created`.

Potential Errors

Status Code 500	Failed to create folder.

Rename a Folder and Update Its Description

```
PUT /directory/<id>/info
```

Request Headers

`Content-Type`	`application/x-www-form-urlencoded`

Form Parameters

name	The folder's new name
description	The folder's new description

Response

Returns the updated JSON representation of the folder.

Potential Errors

`Status Code 403`	No permission to rename the directory.
`Status Code 404`	The directory doesn't exist.

Move a Folder

```
PUT /directory/<id>
```

> **Note**
>
> This operation moves the given directory to a new parent directory. It's also used for soft deletion – in other words, "moving to trash" – if the parent ID is the ID of the Trash.

Request Headers

`Content-Type`	`application/x-www-form-urlencoded`

Form Parameters

parent	Reference to the new parent folder in the form `/directory/<parent_folder_id>`

Response

Returns a JSON representation of the moved folder with its updated parents property.

Potential Errors

`Status Code 403`	No permission to move the directory.
`Status Code 404`	The directory doesn't exist.
`Status Code 409`	Moving the directory isn't allowed.

Delete a Folder

```
DELETE /directory/<id>
```

> **Caution**
>
> This operation deletes the given directory and all items within. No recovery is possible.

Response

Returns a JSON snippet:

```
{ "success": true }
```

Potential Errors

`Status Code 403`	No permission to delete the directory.
`Status Code 404`	The directory doesn't exist.

5.3 Import and Export

This API allows users to import and export artifacts related to SAP Signavio Process Manager.

Prerequisites

Before using the API, you must obtain an authentication token. Refer to Authentication [page 6] for further guidance. Once you have the token, include it in all requests by adding an `x-signavio-id` request header with the token as the header's value.

Requests

When importing or exporting, requests are generally sent to endpoints of the following format:

```
https://{baseUrl}/spm/v1/{resourceType}/{resourceId}/{representation}
```

- `resourceType`: The type of resource being imported or exported, for example `model` or `revision`.
- `resourceId`: The unique identifier of the resource.
- `representation`: The type of representation the resource is being imported or exported as, for example `png` or `bpmn_2_0_xml`.

Some exceptions to this format exist. Refer to the API Reference [page 45] or the API documentation on the SAP Business Accelerator Hub for a comprehensive description of the available endpoints.

The base URL is region-specific. The available regions are:

Region	Base URL
Australia	`api.au.signavio.cloud.sap`
Canada	`api.ca.signavio.cloud.sap`
EU	`api.eu.signavio.cloud.sap`
Japan	`api.jp.signavio.cloud.sap`
Singapore	`api.sgp.signavio.cloud.sap`
South Korea	`api.kr.signavio.cloud.sap`
USA	`api.us.signavio.cloud.sap`

Importing and Exporting Models

Import and export functionality is available for a variety of formats, including:

- PNG
- SVG
- JSON
- BPMN XML
- DMN XML

To export the latest version of a model (also called the head revision), set `resourceType` to `model` and provide the model's ID as `resourceId`.

To export a specific version of a model, set `resourceType` to `revision` and provide the revision's ID as `resourceId`.

Responses

For a complete description of all response types per endpoint, refer to the API Reference [page 45] or the API Documentation on the SAP Business Accelerator Hub. This includes responses to successful operations as well as potential errors.

Related Information

API Documentation on SAP Business Accelerator Hub

5.3.1 API Reference

An overview of the endpoints provided by the Import and Export API.

Common Request Information

All endpoints listed here are relative to `{baseUrl}/spm/v1`. URLs are explained further in URLs and Resource Representation [page 9].

Unless otherwise indicated, all requests should include the following headers:

`x-signavio-id`	The authentication token
`Cookie`	`JSESSIONID=(jsessionid)`
`Accept`	`application/json`

Common Response Information

Unless otherwise indicated, all successful responses return:

- A status code of `200 OK`
- The header `Content-Type=application/json`

All endpoints listed here could potentially return the following errors:

`Status Code 500`	Internal server error

Export a Diagram Revision as BPMN 2.0

You can export the latest revision (also called the head revision) of a diagram via this endpoint:

```
GET /model/<model_id>/<content_type>
```

You can export a specific revision of a diagram via this endpoint:

```
GET /revision/<revision_id>/<content_type>
```

Replace `content_type` with the desired export format. Available options are:

- `json`
- `bpmn2_0_xml`
- `png`
- `svg`

Query Parameters

`updateShapeLabels`	`true`, result: .json, shape labels are updated based on linked dictionary items.
`mergeDictionaryEntryAttributes`	`true`, result: .json, diagram attributes from linked dictionary entries are added to the attributes of their linking shapes.
`mergeLinkedObjectAttributes`	`true`, result: .json, diagram attributes from other diagrams that are linked via collapsed subprocesses are added to the attributes of their linking collapsed subprocess.
`excludeRiskManagement`	`true`, result: .json, risk and control attributes, which are linked to dictionary items, are excluded from the .json, so updating a model via `PUT` request doesn't overwrite the Dictionary links.
`exportSubprocesses`	`true`, result: .json, linked subprocesses are part of the export alongside the main BPMN 2.0 diagram. **Note** This is supported only for the `bpmn2_0_xml` format.

Response

Returns a JSON, XML, SVG or PNG representation of the diagram.

Export a Diagram Revision as DMN 1.2 XML

To fetch the DMN 1.2 XML representation of a diagram's latest revision, send two separate requests: one to generate the XML and then a second to download the XML.

Generating the XML

Use this endpoint to generate the DMN XML:

```
GET /model/<id>/dmn
```

- The response is a JSON document rather than an `HTTP 303 See Other` redirect.
- The response returns a `Content-Type` of `application/json;charset=utf-8`.

Example

Example response:

```
{
  "href": "/dmn",
  "rel": "exp",
  "rep": {
    "downloadUrl": "https://editor.signavio.com/p/dmn-xml-download/
06620b1c66c54efdbec2e7881d70",
    "messages": [],
    "success": true
  }
}
```

Downloading the XML

Use this endpoint to download the DMN XML:

```
GET /dmn-xml-download/<download_id>
```

For the request:

- The `download_id` is available in the response to the `GET /model/<id>/dmn` request.
- The `Accept` request header should have the value `*/*`.

For the response:

- The `Content-Type` response header has the value `application/octet-stream`.

Example

Example response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<definitions namespace="http://www.signavio.com/dmn/1.1/diagram/
ff8ab5d5f46646319ea13f59e2000000.xml"
```

```
  exporterVersion="11.15.5" name="Rules example" id="id-
e2d3f079e41341b49b449ce7d417a25a"
  sigExt:revisionId="0a542e98f63645689d3db01460e3c8c7"
sigExt:revisionNumber="2"
  xmlns="http://www.omg.org/spec/DMN/20151101/dmn.xsd"
  xmlns:sig="http://www.signavio.com/dmn/1.1/diagram/
ff8ab5d5f46646319ea13f59e2000000.xml"
  xmlns:sigExt="http://www.signavio.com/schema/dmn/1.1/"
  xmlns:feel="http://www.omg.org/spec/FEEL/20140401">
    <extensionElements>
      <sigExt:diagramMetaData name="processvaliduntil" value=""/>
      <sigExt:diagramMetaData name="om" value=""/>
    </extensionElements>
    <itemDefinition isCollection="false" name="Age" label="Age"
id="id-66d6c4c25cc223088664eeb5b81d97bd">
      <typeRef>feel:string</typeRef>
    </itemDefinition>
    <inputData name="Age" id="id-90931069787b7884dd33803bfc71a127"
label="Age" sigExt:shapeId="sid-EB04D41F-0B58-455A-9DA8-B6EFEB2DA8C2"
sigExt:diagramId="ff8ab5d5f46646319ea13f59e2cb00b4">
      <extensionElements/>
      <variable typeRef="sig:Age" name="Age"
id="id-90931069787b7884dd33803bfc71a127_variable"/>
    </inputData>
</definitions>
```

Import a Diagram from a BPMN 2.0 XML Document

```
POST /bpmn2_0-import
```

Request Headers

Content-Type	multipart/form-data

Form Parameters

bpmn2_0file	File name and content of the BPMN 2.0 XML file
filename	File name
directory	The directory where the content of the BPMN 2.0 XML file should be imported into
modelid	(Optional) The ID of an existing model for which to create a new revision

Response

```
"[true]\r\n"
```

5.4 Model

With the Model API, you can manage process models and their revisions.

In general, the functions provided by this API enable you to perform CRUD actions on models, namely creating, retrieving, updating, and deleting them. Additional functionality includes creating and managing a model's revisions, and performing syntax checks.

> **Note**
>
> To access the API reference for extracting BPMN 2.0 in various formats, see Export a Diagram Revision. This allows you to retrieve diagrams and revisions as XML or JSON, providing complete access to the BPMN model, including details about its tasks and transitions.

Prerequisites

Before sending requests, ensure that you followed authentication procedures as described in Authentication [page 6] to obtain an authentication token. Include this token in all requests by adding an `x-signavio-id` request header with the token as the header's value.

Requests

Requests to the Model API are sent to `https://{baseUrl}/spm/v1/model` or `https://{baseUrl}/spm/v1/revision`. The base URL is region-specific and should be substituted by the correct base URL for your region:

Region	Base URL
Australia	`api.au.signavio.cloud.sap`
Canada	`api.ca.signavio.cloud.sap`
EU	`api.eu.signavio.cloud.sap`
Japan	`api.jp.signavio.cloud.sap`
Simgapore	`api.sgp.signavio.cloud.sap`
South Korea	`api.kr.signavio.cloud.sap`
USA	`api.us.signavio.cloud.sap`

Models and Their Revisions

The Model API allows you to create, retrieve, update, and delete models (the CRUD operations), providing full control over the model's life cycle. For models with revisions, you can retrieve a list of these revisions along with each revision's metadata.

You can also create, retrieve, update, and delete specific revisions of a model.

Find out more in the API Reference [page 52] or the API documentation on the SAP Business Accelerator Hub.

Model Field Length Restrictions

When creating or updating models in the editor, follow these field length restrictions to avoid breaking an integration:

Field	Location	Max Field Length
Model Name	Header	255 characters
Documentation	Main Attribute	16,777,215 characters
Name	Main Attribute	16,777,215 characters

The Syntax Checker

This API provides a function for performing a syntax check on a BPMN 2.0 process diagram. Learn more in the API Reference [page 60].

Publishing Models

There are a couple of options for publishing models:

- To publish the latest revision of a model, use the `POST /publish` endpoint. Find out more in Rename a Diagram or Change Its Published Revision [page 58].
- To publish a previous revision of a model, use the `PUT /model/{id}/info` endpoint. Find out more in Rename a Diagram or Change Its Published Revision [page 58]

Responses

For a complete listing of all response types per endpoint, refer to the API Reference [page 52] or the API Documentation on the SAP Business Accelerator Hub. This includes responses to successful operations as well as potential errors.

Examples

The following code example gets a list with all the revisions of a particular model, together with each revision's metadata:

Source Code

```
import requests

 from conf import *
 from authentication.authenticate import authenticate

 # user variables
 # ID of the diagram revision you want to retrieve the revision list for
 diagram_ID = '<diagram_ID>'

revision_url = base_url + '/spm/v1/model/' + diagram_ID + '/revisions'
auth_data = authenticate()
# set credentials, response format
cookies = {'JSESSIONID': auth_data['jsesssion_ID']}
headers = {'Accept': 'application/json',
           'x-signavio-id':  auth_data['auth_token']}
get_revisions_request = requests.get(revision_url,
                                     cookies=cookies,
                                     headers=headers)
print(get_revisions_request.text)
```

The following is an example response:

Output Code

```
[
  {
      "rel": "info",
      "href": "/model/beb279a43d1d4080808b048510e7e091/revisions",
      "rep": {
          "size": 3
      }
  },
  {
      "rel": "revision",
      "href": "/revision/3e4e1ae44c934ff79fbf9fa7f77dc078",
      "rep": {
          "rev": 3,
          "author": "/user/36821979e14b417dac355a748f896874",
          "authorName": "John Doe",
          "created": "2017-10-17 11:59:42 +0200",
          "isDeployed": false,
          "comment": "",
          "authorCompany": "Signavio",
          "status": "none"
      }
  },
  {
      "rel": "revision",
      "href": "/revision/c3bcc71abcca4e3aa2b84e49c668bf7c",
      "rep": {
          "rev": 2,
          "author": "/user/36821979e14b417dac355a748f896874",
          "authorName": "John Doe",
          "created": "2017-10-17 11:59:37 +0200",
          "isDeployed": false,
          "comment": "",
```

```
            "authorCompany": "Signavio",
            "status": "none"
        }
    },
    {
        "rel": "revision",
        "href": "/revision/a2a25a97821b4b09acf3617ba70e2863",
        "rep": {
            "rev": 1,
            "author": "/user/36821979e14b417dac355a748f896874",
            "authorName": "John Doe",
            "created": "2017-10-17 11:54:33 +0200",
            "isDeployed": false,
            "comment": "",
            "authorCompany": "Signavio",
            "status": "none"
        }
    }
]
```


Related Information

API Documentation on SAP Business Accelerator Hub

5.4.1 API Reference

An overview of the endpoints provided by the Model API.

Common Request Information

All endpoints listed here are relative to `{baseUrl}/spm/v1`. URLs are explained further in URLs and Resource Representation [page 9].

Unless otherwise indicated, all requests should include the following headers:

`x-signavio-id`	The authentication token
`Cookie`	`JSESSIONID=(jsessionid)`
`Accept`	`application/json`

Common Response Information

Unless otherwise indicated, all successful responses return:

- A status code of `200 OK`

- The header `Content-Type=application/json`

All endpoints listed here could potentially return the following errors:

`Status Code 500`	Internal server error

Retrieve Diagram Data

```
GET /model/<id>
```

Request

Retrieves the meta data of a model. If you don't add any sub-endpoints, the following data is retrieved:

- `info`
- `views`
- `status`
- `parents`
- `subscription`
- `priv`

> **Note**
>
> To access the API reference for extracting BPMN 2.0 in various formats, see Export a Diagram Revision. This allows you to retrieve diagrams and revisions as XML or JSON, providing complete access to the BPMN model, including details about its tasks and transitions.

Sub-endpoints for specific data:

`info`	Basic diagram info, for example the last update or its parent folder. This sub-endpoint also returns details about the publication status of the diagram, including information such as the publication time, the published revision, and the publishing user. As demonstrated in this endpoint's example response, all fields prefixed with '`granted`' describe publication status.
`link`	All linked diagrams and files
`linkedBy`	All diagrams linking to this diagram
`glossary`	All linked dictionary items with their complete data
`glossaryinfo`	All linked dictionary items with their basic data, for example title, description, or custom attributes.
`language`	Desired content language, for example "de_de" or "zh_cn". If this parameter isn't provided or an item isn't translated into this language, then the values of the workspace default language are used as a fallback.
`parents`	Parent directory information
`comments`	User comments for this model
`status`	Approval-workflow states of the current revision

approvalExpiration	Expiration date of the approval for the diagram
views	Stakeholder-specific views for this model
priv	Privileges (access rights) of the current user for this diagram
subscription	Settings for the notification subscription for this diagram set for the current user

The log entry `isDeployed` refers to a deprecated feature and is always `false`.

Example


```
{
  "deleted": false,
  "formats": {},
  "name": "Delivery-to-payment",
  "isLicensedStencilSet": true,
  "namespace": "http://b3mn.org/stencilset/bpmn2.0#",
  "type": "Business Process Diagram (BPMN 2.0)",
  "granted_revision": "/revision/ac5b0f5c9d61437480492e50cafb1b86",
  "granted_revision_number": 3,
  "granted_revision_date": "2024-10-22 10:48:28 +0200",
  "granted_revision_user": "/user/567c007cc30245bd90efc4983da25fc6",
  "granted_revision_user_name": "John Doe",
  "parentName": "End-to-end processes",
  "authorName": "John Doe",
  "authorCompany": "SAP",
  "created": "2024-10-22 10:19:11 +0200",
  "parent": "/directory/6fb49f379f7547ccb133c2aae920e6e7",
  "author": "/user/567c007cc30245bd90efc4983da25fc6",
  "revision": "/revision/ac5b0f5c9d61437480492e50cafb1b86",
  "rev": 3,
  "comment": "",
  "updated": "2024-10-22 10:48:19 +0200",
  "status": {
    "id": "none",
    "deleted": false,
    "publish": false,
    "approve": false
  },
  "isDeployed": false,
  "sri": "sri:eu:1bd018cbea0844eb96cf3d5f1e39d8f5:spm:model/
258ec2c424954e2492cce933a9985a04",
  "sri_path": "sri:eu:1bd018cbea0844eb96cf3d5f1e39d8f5:spm:directory/
b0eff8603c254d7d8f5bf5a8aa4cbf10;sri:eu:1bd018cbea0844eb96cf3d5f1e39d8f5:spm:d
irectory/6fb49f379f7547ccb133c2aae920e6e7",
  "sri_revision": "sri:eu:1bd018cbea0844eb96cf3d5f1e39d8f5:spm:model/
258ec2c424954e2492cce933a9985a04:revision/ac5b0f5c9d61437480492e50cafb1b86",
  "description": "",
  "name_en": "Delivery-to-payment"
}
```


Retrieve All Revision IDs of a Diagram

```
GET /model/<id>/revisions
```

Query Parameters

offset	The offset of the first revision to be returned
limit	The maximum number of revisions to be returned
query	A search query to filter the revisions

Response

Returns a diagram's list of revisions, together with each diagram revision's metadata.

Example

Example response


```
[
  {
    "rel":"info",
    "href":"/model/beb279a43d1d4080808b048510e7e091/revisions",
    "rep":{
      "size":3
    }
  },
  {
    "rel":"revision",
    "href":"/revision/3e4e1ae44c934ff79fbf9fa7f77dc078",
    "rep":{
      "rev":3,
      "author":"/user/36821979e14b417dac355a748f896874",
      "authorName":"John Doe",
      "created":"2017-10-17 11:59:42 +0200",
      "isDeployed":false,
      "comment":"",
      "authorCompany":"Signavio",
      "status":"none"
    }
  },
  {
    "rel":"revision",
    "href":"/revision/c3bcc71abcca4e3aa2b84e49c668bf7c",
    "rep":{
      "rev":2,
      "author":"/user/36821979e14b417dac355a748f896874",
      "authorName":"John Doe",
      "created":"2017-10-17 11:59:37 +0200",
      "isDeployed":false,
      "comment":"",
      "authorCompany":"Signavio",
      "status":"none"
    }
  },
  {
    "rel":"revision",
    "href":"/revision/a2a25a97821b4b09acf3617ba70e2863",
    "rep":{
      "rev":1,
      "author":"/user/36821979e14b417dac355a748f896874",
      "authorName":"John Doe",
      "created":"2017-10-17 11:54:33 +0200",
      "isDeployed":false,
      "comment":"",
      "authorCompany":"Signavio",
      "status":"none"
    }
  }
```

```
]
```

Create a New Diagram Draft

```
GET /editorcreate
```

> → Remember
>
> This operation creates an empty diagram, but it doesn't store the diagram. To store a diagram, refer to Store a New Diagram [page 57].

Request Headers

`Content-Type`	`application/x-www-form-urlencoded`

Form Parameters

`stencilset`	The notation URI value for the stencil set. For example: `http://b3mn.org/stencilset/bpmn2.0#`

Response

- Body: A JSON representation of the empty diagram
- Status code: `201 Created`.

Field Length

When you create a diagram, consider the limits applied to fields:

- Model name: 255 characters.
- Documentation attribute: 16.777.215 characters.
- Name attribute: 16.777.215 characters.
- URL custom attribute: 255 characters.

Response Headers

A response from this endpoint includes the additional header:

`Location`	The new model's ID.

> ⓘ Note
>
> The `Location` response header value is the model ID, not a relative URL.

Open the URL `<baseUrl>/spm/v1/editor?id=<model_id>` in a web browser to view a diagram in the graphical editor.

Store a New Diagram

```
POST /model
```

Request Headers

Content-Type	application/x-www-form-urlencoded

> → Remember
>
> Although the request content type is `application/x-www-form-urlencoded` the `json_xml` parameter value must be JSON. Take care to correctly encode the JSON content for this content type.

Form Parameters

name	diagram name
parent	`/directory/<parent_directory_id>`
namespace	For example, `http://b3mn.org/stencilset/bpmn2.0#`
json_xml	JSON representation
id	(Optional) ID of the diagram, as provided when creating a new diagram draft [page 56].
svg_xml	(Optional) An SVG representation
comment	(Optional) A revision comment

Response

Returns a JSON representation of the diagram.

> Example
>
> Example response

```
[{
    [...]
}, {
    "rel": "parents",
    "href": "/model/83b98a0d07c7482a934c6140ea07b434/parents",
    "rep": [{
        "rel": "dir",
        "href": "/directory/40d7fed893894624878a43f5e99943e7",
        "rep": {
            "visible": true,
            "created": "2016-03-07 10:34:41 +0100",
        "description": "",
        "name": "End-to-end processes",
        "parent": "/directory/fa9149ed892348cb884b5bea7ba5ab2b",
        "deleted": false
        }
    },{
        "rel": "dir",
        [...]
    }]
}, {
    "rel": "info",
    "href": "/model/83b98a0d07c7482a934c6140ea07b434/info",
    "rep": {
```

```
        "rev": 2,
        "parent": "/directory/40d7fed893894624878a43f5e99943e7",
        "granted_revision": "/revision/d7d8a73e051b4f96b8fd6dc52d2ca754",
        "type": "Business Process Diagram (BPMN 2.0)",
        "deleted": false,
        "namespace": "http://b3mn.org/stencilset/bpmn2.0#",
        "author": "/user/d2e2757a6e6f4856a7843e6ceb69a20e",
        "revision": "/revision/7ea59453149d4974947bfe8cc1eb406e",
        "isLicensedStencilSet": true,
        "created": "2016-03-29 13:41:26 +0200",
        "updated": "2016-11-07 11:09:30 +0100",
        "description": "The diagram description",
        "name": "Delivery-to-payment",
        "comment": "created view for management",
        "parentName": "End-to-end processes"
    }
}, {
    "rel": "revision",
    "href": "/revision/7ea59453149d4974947bfe8cc1eb406e",
    "rep": {
        "author": "/user/d2e2757a6e6f4856a7843e6ceb69a20e",
        "rev": 2,
        "created": "2016-11-07 11:09:30 +0100",
        "comment": "created view for marketing",
        "isDeployed": false
    }
}, {
    "rel": "revision",
    [...]
}]
```


Rename a Diagram or Change Its Published Revision

```
PUT /model/<id>/info
```

Renames the model specified with the ID, or changes the model revision that is published in SAP Signavio Process Collaboration Hub.

Request Headers

`Content-Type`	`application/x-www-form-urlencoded`

Form Parameters

`name`	Diagram name
`granted_revision`	`/revision/<revision_id>`

Response

Returns a JSON representation of the diagram with updated properties.

Potential Errors

`Status Code 400`	Bad request
`Status Code 404`	Model not found

Copy a Diagram

```
POST /model
```

Request Headers

Content-Type	application/x-www-form-urlencoded

Form Parameters

copy	true
id	Diagram to copy, in the format /model/<model_id>
parent	The parent folder that will contain the copy, in the format /directory/<parent_folder_id>
name	The new folder's name

Response

Returns a JSON representation of the diagram with updated properties.

Move a Diagram

```
PUT /model/<id>
```

Request Headers

Content-Type	application/x-www-form-urlencoded

Form Parameters

parent	The new containing folder of the diagram, in the format /directory/<parent_folder_id>

Response

Returns a JSON array with the items related to this model.

Update a Diagram By Adding a New Revision

```
PUT /model/<id>
```

> **Note**
>
> This updates the specified diagram by creating a new diagram revision. You can still retrieve and restore older diagram revisions.

This request is the same kind as when moving a diagram [page 59]. If you use values for `name` and `parent` that are different to the current values, you'll change the name or location of the original diagram.

→ Tip

You can also use this endpoint to update a model's attributes. To do so, follow these steps:

1. Retrieve the model whose attributes you'd like to update by exporting it in JSON format [page 46].
2. In the exported JSON representation, find the attributes you'd like to alter and make your changes.
3. Upload the updated version of your diagram using this endpoint. Provide your updated JSON representation in the `json_xml` form parameter.

Request Headers

`Content-Type`	`application/x-www-form-urlencoded`

Form Parameters

`name`	The diagram name
`parent`	The parent folder, in the format `/directory/<parent_folder_id>`
`comment`	Revision comment
`json_xml`	JSON representation of the new revision
`svg_xml`	(Optional) SVG representation of the new revision

Field Length

When you update a diagram, consider the limits applied to fields:

- Model name: 255 characters.
- Documentation attribute: 16.777.215 characters.
- Name attribute: 16.777.215 characters.
- URL custom attribute: 255 characters.

Response

Returns a JSON array with the items related to this model.

Perform a Syntax Check on a BPMN 2.0 Diagram

```
POST /syntaxchecker
```

→ Remember

Use the JSON representation of the BPMN 2.0 process diagram, rather than the BPMN XML, and encode it as a form parameter value in the form-encoded request. Don't use the JSON or XML representation as the entire request body.

Request Headers

Content-Type	application/x-www-form-urlencoded

Form Parameters

data_json	JSON representation of your diagram as a string
ns	http://b3mn.org/stencilset/bpmn2.0#
isJson	true

Response

Returns a JSON Object with error codes.

Example

Example response

```
{
  "rel": "syntaxchecker",
  "href": "/syntaxchecker",
  "rep": [
    {
      "should": {},
      "must": {
        "sid-435FD0FE-A44E-4FFA-810D-A528FA0C781B": [
        "BPMN_NO_TARGET"
        ]
      },
      "guidelineId": "syntax_checker"
    }
  ]
}
```

Publish an Item

Publish the latest revision of one or more diagrams to SAP Signavio Process Collaboration Hub.

```
POST /publish
```

Note

To publish older revisions, send a separate request for each diagram. Refer to Rename a Diagram or Change Its Published Revision [page 58]

Request Headers

Content-Type	application/x-www-form-urlencoded

Form Parameters

`models` An identifier for the model to be published in the format `/model/<id>`. Include one `models` parameter for each model to be published. For example, publishing three models requires a request with three parameters:

```
models: /model/df7b0129b7a44e91b136f75b6d393a61
models: /model/8da583419c74476aade0fb7a9ad8391b
models: /model/9357469d470c4147b37b0cc31f7b0ca
```

`mode` `publish`

Response

Returns a JSON array containing the published models.

Published Revisions

All endpoints that take the ID of a model or dictionary entry as a parameter by default include information about the latest revision (also called head revision) of the object in the response.

To retrieve information about the published revision for these endpoints, you can add the parameter `signavio-rev-mode` with the value `PUBLISHED` to the request.

For objects not currently published, then a `403 (Forbidden)` error code is returned.

Using this parameter when requesting a folder results in a limited response. Sub-folders are only included in the response if they contain at least one published object somewhere down the line. Also, only published objects are included in the response. If a folder contains no published objects, then the request also results in a 403 error.

5.4.2 Creating Custom Attributes for a BPMN Diagram

The `/spm/v1/meta` endpoint enables you to add custom attributes for BPMN diagrams.

To add a custom attribute, dispatch a `POST` request to `/spm/v1/meta` with the following form parameters:

`stencilsetBindings`	The namespace and the stencil for which the custom attribute is created. In this case, the namespace is BPMN2.0 and the stencil is "Task".
`name`	The desired name of the custom attribute.
`type`	The type of the custom attribute.
`isList`	If the custom attribute should be a list.

If the request is successful, a response is returned with the following:

- A response code of `200 OK`.
- A response body containing a JSON representation of the custom attribute.

Example

The following Python script creates a new custom attribute for the task stencil of BPMN diagram:

```
import requests
from conf import *
from authentication.authenticate import authenticate
# user variables
stencilsetBindings = {
    "namespace": "http://b3mn.org/stencilset/bpmn2.0#",
    "stencil": "Task",
    "order": 48,
}
name = "A sample custom attribute"
attributeType = "MetaDataStringInfo"
isGlossaryDefinition = True
parent = "/meta"
isList = True
custom_attribute_url = base_url + "/spm/v1/meta"
auth_data = authenticate()
# set credentials, response format
cookies = {
    "JSESSIONID": auth_data["jsesssion_ID"],
    "LBROUTEID": auth_data["lb_route_ID"],
}
headers = {"Accept": "application/json", "x-signavio-id":
auth_data["auth_token"]}
create_custom_attribute_request = requests.post(
    custom_attribute_url,
    cookies=cookies,
    headers=headers,
    data={
        "stencilsetBindings": stencilsetBindings,
        "name": name,
        "type": attributeType,
        "isGlossaryDefinition": isGlossaryDefinition,
        "parent": parent,
        "isList": isList,
    },
)
```

Example response:

```
{
  "rel": "meta",
  "href": "/meta/meta-asamplecustomattribute",
  "rep": {
    "glossaryBindings": [],
    "lineWrap": false,
    "defaultValue": "",
    "length": 0,
    "description": "",
    "isGlossaryDefinition": true,
    "type": "MetaDataStringInfo",
    "isList": true,
    "multilanguage": false,
    "stencilsetBindings": [
      {
        "namespace": "http://b3mn.org/stencilset/bpmn2.0#",
        "stencil": "Task",
        "order": 51
      }
    ],
    "readonly": false,
    "name": "A sample custom attribute",
    "id": "meta-asamplecustomattribute"
```

```
  }
}
```

5.5 Search

The search function performs a full-text search of all content types.

You can configure a search in various ways, such as by limiting the result size and restricting the set of content types to be searched.

Prerequisites

Before using the API, you must obtain an authentication token. Refer to Authentication [page 6] in this guide for help on getting the token.

Once you have the token, include it in all requests to the Search API by adding an `x-signavio-id` request header with the token as the header's value.

Performing the Search

A search is performed by dispatching a request to the following URL:

```
https://{baseUrl}/spm/v1/search?q={searchTerm}
```

Substitute values for the variables in this URL:

- The base URL is region-specific. Refer to the section Base URL [page 65] to find the one corresponding to your region.
- The search term is the text used for the search.

This is all that's required to perform a basic search. Refer to the API Reference [page 68] or the API Documentation on the SAP Business Accelerator Hub for a comprehensive description of request information.

Configuring the Search

You can configure the search further by adding query parameters to the request. Configuration options include:

- `limit`: The maximum number of results to return
- `offset`: The index of the first result to return
- `types`: A list of content types the search should include

Refer to the API Reference [page 68] or the API Documentation on the SAP Business Accelerator Hub for more information about these parameters.

Responses

A successful response returns a JSON array containing folder and model object representations.

In the response object with the property `rel="search"`, the `totalNrOfResults` property gives the total number of results. When there are more than 250 results, the API limits the response array to the first 250 results. You can use the `offset` request parameter to fetch subsequent results, for example by setting `offset=250` to return the second 'page' of results.

Refer to the API Reference [page 68] or the API Documentation on the SAP Business Accelerator Hub for a comprehensive description of response information, including response codes, the data schema, and example values.

Base URL

API endpoints are relative to the base URL, which is specific to your region. The available base URLs are:

Region	Base URL
Australia	`api.au.signavio.cloud.sap`
Canada	`api.ca.signavio.cloud.sap`
EU	`api.eu.signavio.cloud.sap`
Japan	`api.jp.signavio.cloud.sap`
Singapore	`api.sgp.signavio.cloud.sap`
South Korea	`api.kr.signavio.cloud.sap`
USA	`api.us.signavio.cloud.sap`

Example

The following Python script searches for the term `calculate` and returns the first 10 results:

Source Code

```
import requests
from conf import *
# set search parameters
query = 'calculate'
limit = 10
offset = 0
types = (
    # remove items to filter by content type
    'MODEL',
    'MODEL_REVISION',
    'SHAPE',
    'FILE',
    'FILE_REVISION',
    'DIR',
    'COMMENT'
```

```
)
# setup request (add appropriate base URL)
base_url = ''
search_url = base_url + '/spm/v1/search'
query_url = '{0}?q={1}'.format(search_url, query)
# obtain your authentication token and set the value here
token = ''
# set credentials and response format
headers = {
    'Accept': 'application/json',
    'x-signavio-id': token
}
# execute search
search_request = requests.get(
    query_url,
    headers = headers,
    params = {
        limit: limit,
        offset: offset,
        types: types
    }
)
result = str(search_request.content)
print(result)
```

The following shows a sample of an example response including one result:

Output Code


```
[
    {
        "rel":"mod",
        "href":"/model/242f5bed25024ebe8c96a72b956d5be0",
        "rep":{
            "granted_revision":"",
            "parent":"/directory/889c590193d54db18fe9b0fbc0470bba",
            "rev":1,
            "granted_revision_user":"",
            "created":"2016-08-19 14:33:56 +0200",
            "author":"/user/f40100476cfb4b058b7454cb1db1a83a",
            "isLicensedStencilSet":true,
            "description":"",
            "granted_revision_date":"",
            "type":"Entscheidungsdiagramm (DMN 1.1)",
            "authorCompany":"Signavio",
            "revision":"/revision/d7d28f5804f343f995e1c9d406829ce4",
            "parentName":"Prozessbeispiele",
            "deleted":false,
            "authorName":"John Doe",
            "numberOfNewComments":0,
            "isDeployed":false,
            "name":"Calculate Discount",
            "namespace":"http://signavio.com/stencilsets/dmn-1.0#",
            "comment":"",
            "fields":[
                "MODEL",
                "SHAPE",
                "Score:4.5136113"
            ],
            "updated":"2016-08-19 14:33:56 +0200",
            "status":{
                "deleted":false,
                "publish":false,
                "id":"none"
            }
        }
```

```
    },
    {
        "rel":"search",
        "href":"/search",
        "rep":{
            "types":[
                "MODEL",
                "MODEL_REVISION",
                "SHAPE",
                "FILE",
                "FILE_REVISION",
                "DIR",
                "COMMENT"
            ],
            "offset":0,
            "showMatchInfos":true,
            "limit":20,
            "fields_json":[
                {
                    "textQueries":[
                        {
                            "text":"calculate"
                        }
                    ],
                    "name":"Alle",
                    "id":"search.searchableFields.all"
                },
                {
                    "plainQueries":[
                        "MODEL",
                        "MODEL_REVISION",
                        "SHAPE",
                        "FILE",
                        "FILE_REVISION",
                        "DIR",
                        "COMMENT"
                    ],
                    "name":"doctype",
                    "id":"doctype"
                }
            ],
            "totalNrOfResults":1,
            "relevantResources":[

            ]
        }
    }
]
```


Related Information

API Documentation on SAP Business Accelerator Hub

5.5.1 API Reference

An overview of the endpoints provided by the Search API.

Performing a Full-Text Search

```
GET /search
```

This endpoint is relative to `{baseUrl}/spm/v1`. URLs are explained further in URLs and Resource Representation [page 9].

Request Headers

`x-signavio-id`	The authentication token
`Cookie`	`JSESSIONID=(jsessionid);LBROUTEID=(lbrouteid)`
`Accept`	`application/json`
`Content-Type`	`application/x-www-form-urlencoded`

Query Parameters

`q`	Search string, for example "Obtainment"
`limit`	(Optional) Maximum number of search results, up to 250
`offset`	Index of first search result to return
`types`	List of content types the search should include, from: `MODEL`, `MODEL_REVISION`, `SHAPE`, `FILE`, `FILE_REVISION`, `DIR` and `COMMENT`.

Response

- `Status Code: 200 OK`
- `Content-Type: application/json`

Returns a JSON array containing folder and model object representations. In the response object with `rel="search"`, the `totalNrOfResults` property gives the total number of results. When there are more than 250 results, the API limits the response array to the first 250 results. You can use the `offset=250` request parameter to fetch the second 'page' of results.

Example

Example response


```
[
  {
    "rel":"mod",
    "href":"/model/242f5bed25024ebe8c96a72b956d5be0",
    "rep":{
      "granted_revision":"",
      "parent":"/directory/889c590193d54db18fe9b0fbc0470bba",
      "rev":1,
```

```
      "granted_revision_user":"",
      "created":"2016-08-19 14:33:56 +0200",
      "author":"/user/f40100476cfb4b058b7454cb1db1a83a",
      "isLicensedStencilSet":true,
      "description":"",
      "granted_revision_date":"",
      "type":"Entscheidungsdiagramm (DMN 1.1)",
      "authorCompany":"Signavio",
      "revision":"/revision/d7d28f5804f343f995e1c9d406829ce4",
      "parentName":"Prozessbeispiele",
      "deleted":false,
      "authorName":"John Doe",
      "numberOfNewComments":0,
      "isDeployed":false,
      "name":"Calculate Discount",
      "namespace":"http://signavio.com/stencilsets/dmn-1.0#",
      "comment":"",
      "fields":[
        "MODEL",
        "SHAPE",
        "Score:4.5136113"
      ],
      "updated":"2016-08-19 14:33:56 +0200",
      "status":{
        "deleted":false,
        "publish":false,
        "id":"none"
      }
    }
  },
  {
    "rel":"search",
    "href":"/search",
    "rep":{
      "types":[
        "MODEL",
        "MODEL_REVISION",
        "SHAPE",
        "FILE",
        "FILE_REVISION",
        "DIR",
        "COMMENT"
      ],
      "offset":0,
      "showMatchInfos":true,
      "limit":20,
      "fields_json":[
        {
          "textQueries":[
            {
              "text":"calculate"
            }
          ],
          "name":"Alle",
          "id":"search.searchableFields.all"
        },
        {
          "plainQueries":[
            "MODEL",
            "MODEL_REVISION",
            "SHAPE",
            "FILE",
            "FILE_REVISION",
            "DIR",
            "COMMENT"
          ],
          "name":"doctype",
          "id":"doctype"
```

```
        }
      ],
      "totalNrOfResults":1,
      "relevantResources":[

      ]
    }
  }
]
```

6 Custom Attributes

Use the `/meta` endpoint to manage custom attributes for diagrams and dictionary categories.

> **Note**
>
> The terms 'dictionary' and 'glossary' can be used interchangeably. In this documentation, the term 'dictionary' is used, while the API endpoints use the term 'glossary'.

Creating Custom Attributes for BPMN Diagrams

To learn more, refer to Creating Custom Attributes for a BPMN Diagram [page 62]

Creating Custom Attributes for Dictionary Categories

To learn more, refer to Creating Custom Attributes for Dictionary Categories [page 33].

Viewing Custom Attributes IDs

To locate the technical ID of your custom attribute for processes or objects, navigate to SAP Signavio Process Manager *Setup* > *Define notations/attributes* > *Custom attributes*.

Retrieving the Current Custom Attributes

To retrieve a list of all custom attribute definitions, dispatch a `GET` request to `/spm/v1/meta` .

If the request is successful, a response is returned with the following:

- A response code of `200 OK`.
- A response body containing JSON representations of all available custom attributes definitions.

> **Example**
>
> An example response.

```
{
   "rel":"meta",
   "href":"/meta/meta-cause",
   "rep":{
```

```
      "glossaryBindings":[
         {
            "category":"65f53b120d354bc5bdda69e0b9368d2e",
            "order":0
         }
      ],
      "lineWrap":false,
      "defaultValue":"",
      "length":0,
      "description":"",
      "isGlossaryDefinition":true,
      "type":"MetaDataStringInfo",
      "isList":false,
      "multilanguage":false,
      "stencilsetBindings":[

      ],
      "readonly":false,
      "name":"Cause",
      "id":"meta-cause"
   }
},
{
   "rel":"meta",
   "href":"/meta/meta-customstart",
   "rep":{
      "multilanguage":false,
      "glossaryBindings":[

      ],
      "stencilsetBindings":[
         {
            "namespace":"http://b3mn.org/stencilset/bpmn2.0#",
            "stencil":"StartNoneEvent",
            "order":1
         }
      ],
      "readonly":false,
      "defaultValue":"",
      "name":"customStart",
      "id":"meta-customstart",
      "isGlossaryDefinition":true,
      "category":"",
      "type":"MetaDataGlossaryLink",
      "isList":false
   }
},
{
   "rel":"meta",
   "href":"/meta/meta-customstartlist",
   "rep":{
      "multilanguage":false,
      "glossaryBindings":[

      ],
      "stencilsetBindings":[
         {
            "namespace":"http://b3mn.org/stencilset/bpmn2.0#",
            "stencil":"StartNoneEvent",
            "order":2
         }
      ],
      "readonly":false,
      "defaultValue":"",
      "name":"customStartList",
      "description":"",
      "id":"meta-customstartlist",
      "isGlossaryDefinition":true,
```

```
        "type":"MetaDataModelLink",
        "isList":true
    }
}
```

7 Custom Data Types

Learn how to define your own custom data types.

> **Note**
>
> This feature is only available in the Enterprise Plus Edition of SAP Signavio Process Manager.

In SAP Signavio Process Manager you can define custom attributes for use in models and the dictionary - these are referred to as custom data types. You can also define custom data types in the dictionary, called central data types. Such data types can be used in Decision Model & Notation (DMN 1.2) diagrams for modeling complex business decisions. Custom data types also provide data lookup from external sources.

Dictionary Category Configuration

To define custom data types in the dictionary inside a dictionary category, a special setting must be activated. Refer to Dictionary [page 11] for more information.

Change your predefined dictionary category by additionally adding the following request parameter when creating or updating a category:

```
decisioninput=true
```

The dictionary category's JSON object representation now includes these new properties:

```
{
  "decisionDataObjectKey": "meta-metadatadataobject",
  "decisionInput": true,
  [...]
}
```

While the `decisionDataObjectKey` property's value is usually `meta-metadatadataobject` for all categories, implementations shouldn't rely on this. Still, the following examples use this property name, and refer to it as the 'decision data property'.

Creating Custom Data Types

To create a custom dictionary type, send a `POST /spm/v1/glossary` request with type-specific content described in the following section and that endpoint's body parameters described in the SAP Business Accelerator Hub.

Boolean

Boolean types only allow a property to be either true or false, and are typically represented as a checkbox.

Properties in the decision data property's JSON object value:

type:	`boolean`

Example result:

```
{
    "title":"MyBooleanType",
    "category":"/glossarycategory/657e2a77bbfe4abb9dae463aab07d99d",
    [...]
    "metaDataValues":{
        "meta-metadatadataobject":{
           "type":"boolean"
        }
     }
}
```

String

String types allow any text input.

Properties in the decision data property's JSON object value:

type:	`string`
ignoreCase:	Optional. Turns off case-sensitivity, if set to `true`.

Example result:

```
{
    "title":"MyStringType",
    "category":"/glossarycategory/657e2a77bbfe4abb9dae463aab07d99d",
    [...]
    "metaDataValues":{
        "meta-metadatadataobject":{
           "type":"string",
           "ignoreCase":true
        }
     }
}
```

Number

The number type allows the user to define a double-precision decimal number, and set a minimum and maximum value as well as a unit for it.

Properties in the decision data property's JSON object value:

type: `number`

min: Optional. Minimum value (double)

max: Optional. Maximum value (double)

unit: Optional. The quantity the number represents (String), one of:

- `percentage`
- `weight`
- `length`
- `temperature`
- `currency`

weight: Optional. If `unit` is set to `weight` this parameter defines the weight unit (String), one of:

- `mg`: milligram
- `g`: gram
- `kg`: kilogram
- `oz`: ounce
- `lbs`: pound
- `t`: ton

length: Optional. If `unit` is set to `length` this parameter defines the length unit (String), one of:

- `mm`: millimeter
- `cm`: centimeter
- `m`: meter
- `km`: kilometer
- `in`: inch
- `ft`: foot
- `yd`: yard

temperature: Optional. If `unit` is set to `temperature` this parameter defines the temperature unit (String), one of:

- `c`: Celsius
- `f`: Fahrenheit
- `k`: Kelvin

currency: Optional. If `unit` is set to `currency` this parameter defines the currency code (String), one of: `EUR`, `USD`, `AUD`, `BGN`, `BRL`, `CAD`, `CHF`, `CLP`, `CZK`, `DKK`, `GBP`, `HUF`, `MXN`, `NZD`, `PLN`, `RON`, `RUB`, `SEK`, `SGD`, `ZAR`

Example result:

```
{
    "title": "MyNumberType",
    "category": "/glossarycategory/657e2a77bbfe4abb9dae463aab07d99d",
    [...]
    {
```

```
    "metaDataValues":{
        "meta-metadatadataobject":{
           "type":"number",
           "min":0.3,
           "max":42,
           "unit":"weight",
           "weight":"kg"
        }
    }
}
```

Enumeration

Enumeration types allow a value to be one of a given list of values. They're typically represented as a drop-down menu. The list of permitted values is provided at creation or modification time.

Properties in the decision data property's JSON object value:

type: `enumeration`

enumItems: Zero or more items, provided in a JSON array. Each item is a JSON Object and contains and `id` and a `title`.

Example result:

```
{
   "title":"MyEnumType",
   "category":"/glossarycategory/657e2a77bbfe4abb9dae463aab07d99d",
   [...]
   "metaDataValues":{
      "meta-metadatadataobject":{
         "type":"enumeration",
         "enumItems":[
            {
               "id":"0",
               "title":"A"
            },
            {
               "id":"1",
               "title":"B"
            },
            {
               "id":"2",
               "title":"C"
            }
         ]
      }
   }
}
```

Date, Time, and DateTime

Date types allow the user to define data types that are correlated to dates and/or times.

Properties in the decision data property's JSON object value:

type: `date`

datetype: Defines the type of this date (String), one of:

- `date`: This data type describes a date.
- `time`: This data type describes a specific time of day.
- `datetime`: This data type describes a certain point in time including the date.

Example result:

```
{
   "title":"MyDate",
   "category":"/glossarycategory/657e2a77bbfe4abb9dae463aab07d99d",
   [...]
   "metaDataValues":{
      "meta-metadatadataobject":{
         "type":"date",
         "datetype":"time"
      }
   }
}
```

Hierarchy

Hierarchy types allow values to be one of a list of defined values. They're similar to enumerations, but define a hierarchy of values instead of a flat list. Hierarchies are typically represented by a tree control in web applications. The list of permitted values is provided at creation or modification time.

Properties in the decision data property's JSON object value:

type: `hierarchy`

hierarchyItems: Zero or more in a JSON array. Each item is a JSON object and contains:

- `nodeid`: The identifier of that item (String). Node IDs should be unique within the entire hierarchy.
- `title`: A simple string that represents this item (String).
- `children` Optional. JSON array that contains the sub-items as JSON objects (the structure of sub-items is the same as the parent items).

Example result:

```
{
  "title":"MyHierarchy",
  "category":"/glossarycategory/657e2a77bbfe4abb9dae463aab07d99d",
  [...]
  "metaDataValues": {
    "meta-metadatadataobject": {
    "type": "hierarchy",
    "hierarchyItems": [
      {
        "nodeid": "1",
        "title": "A",
```

```
        "children": [{
          "nodeid": "3",
          "title": "Aa"
        }, {
          "nodeid": "4",
          "title": "Ab"
        }]
    }, {
        "nodeid": "2",
        "title": "B",
        "children": [{
          "nodeid": "5",
          "title": "Ba"
        }, {
          "nodeid": "6",
          "title": "Bb"
        }]
      }
    ]}
  }
}
```

Complex

Complex types are record types composed of various other types. A record can contain a person's address, composed of their name (String), age (Number), gender, or even nested data like an address type: the main record contains a composite field for street name (String) and street number (Number).

Properties in the decision data property's JSON object value:

type:	`complex`
relations:	(zero or more in a JSON array):
referenced subtypes	A nested list of JSON objects, each containing:

relationId	Zero-based unique number used to define paths along complex types
title	Title of type reference
relationType	"glossary"
value	`/glossary/(ID)`; `ID`: Identifier of the dictionary entry that is referenced
isList	Optional. Defines if this complex data type contains a list of these objects or only one (Boolean). Default is `false`.
gitemTitle	Only in response. Current title of the linked dictionary entry in correct translation.
invalid	Only in response. `true` if the referenced dictionary entry was found.

Example request:

```
{
   "title":"MyComplex Type",
   "category":"/glossarycategory/657e2a77bbfe4abb9dae463aab07d99d",
   [...]
   "metaDataValues":{
```

```
      "meta-metadatadataobject":{
         "type":"complex",
         "relations":[
            {
               "title":"mySubAttribute1",
               "value":"/glossary/28faf8576c9f4555b67e5fa8c5a12f75",
               "relationType":"glossary",
               "relationId":0
            },
            {
               "title":"mySubAttribute2",
               "value":"/glossary/9b466fdcee7f4d58a71326af331fb72d",
               "relationType":"glossary",
               "relationId":1
            }
         ]
      }
   }
}
```

Example result:

```
{
   "title":"MyComplex Type",
   "category":"/glossarycategory/657e2a77bbfe4abb9dae463aab07d99d",
   "metaDataValues":{
      "meta-metadatadataobject":{
         "type":"complex",
         "relations":[
            {
               "title":"mySubAttribute1",
               "value":"/glossary/28faf8576c9f4555b67e5fa8c5a12f75",
               "gitemTitle":"MyStringType",
               "relationType":"glossary",
               "relationId":0,
               "invalid":false
            },
            {
               "title":"mySubAttribute2",
               "value":"/glossary/9b466fdcee7f4d58a71326af331fb72d",
               "gitemTitle":"MyNumberType",
               "relationType":"glossary",
               "relationId":1,
               "invalid":false
            }
         ]
      }
   }
}
```

No Data Type

Dictionary entries can be created without a data definition. This applies to dictionary entries in a category enabled for data modeling that has no specific data type.

Properties in the decision data property's JSON object value:

type:	none

Example result:

```
{
   "title":"MyStringType",
   "category":"/glossarycategory/657e2a77bbfe4abb9dae463aab07d99d",
   [...]
   "metaDataValues":{
      "meta-metadatadataobject":{
         "type":"none"
      }
   }
}
```

8 Restrictions and Limits

Usage of SAP Signavio Process Manager APIs is monitored to ensure that our services are available fairly and reliably

To ensure a consistent level of service and prevent any disruptions, clients may access the API according to the following policy:

Aspect	Limits
API rate limit	50 requests per 60 seconds
Concurrency limits	Read: 5 Write: 1
Maximum number of items fetched when retrieving dictionary resources (`GET /spm/v1/glossary`)	1000 per request

9 Troubleshooting

Find help with problems when using SAP Signavio Process Manager APIs.

Common Errors

This table lists commonly-encountered errors and their likely causes:

> → Tip
>
> It is not guaranteed that the REST API will return HTTP 4xx (client error) status responses if you use the wrong request URL, request method, or request `Content-Type` header value. If you encounter unexpected errors, verify the request information.

Error	Possible Causes and Resolutions
Status Code 403, ACCESS VIOLATION or similar	A privilege is needed for the operation, adjust your workspace's user management. The user needs to be among the workspace's administrators to perform the operation.
Status Code 423	This error can have the following causes: • You've reached the limit of failed login attempts. • Your workspace has single sign-on (SSO) enforced and your requests are being redirected to the authentication service. This can happen if you're trying to access the API using an account lacking the "ApiAccess" license, which allows an account to avoid SSO and authenticate directly in SAP Signavio. Refer to Access and Licensing [page 4] for more information. ⓘ Note In this case, your user is locked. Please create a support request to unlock your account.
Status Code 500, platform.jsonexception	The provided parameters cannot be resolved to a valid JSON document. Its structure may be faulty and important keys and values may be missing.
Status Code 500, platform.ioexception	Redirect to dictionary browser page failed. Make sure you set the `Accept` request header value to `application/json`.
Status Code 500, platform.servletexeption	Redirect to dictionary browser page failed. Make sure you set the `Accept` request header value to `application/json`.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon  : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap