d.velop

CM100 (d.velop documents repo export API)

Table of Contents

1. d.velop documents repo export API	. 3
1.1. Introduction	. 3
1.1.1. About the d.velop documents repo export API	. 3
1.1.2. Scope of functions of d.velop documents repo export API	. 3
1.1.3. Prerequisites	
1.2. Using the API functions	. 4
1.2.1. Activating the API for the cloud	. 4
1.2.2. Activating the API for on-premises	. 4
1.2.3. Defining the export filter	
1.2.4. Requirements for metadata and download URLs	. 7
1.2.5. Exporting the source system configuration	
1.2.6. Configuring user rights for the cloud	. 9
1.2.7. Configuring user rights for on-premises	15
1.2.8. Using the divelop documents repo export client	16

1. d.velop documents repo export API

1.1. Introduction

This section provides you with general product information.

1.1.1. About the d.velop documents repo export API

The d.velop documents repo export API (export API) is a REST HTTP endpoint through which you can retrieve metadata, links to files and system configurations (master data). Starting from the migration interface, the export API assists with the export of documents from d.velop documents.

The output format for the metadata information is StandardDocumentImportFormat (JSON). This format is the defined target format for the migration API and contains practically all the metadata information for a d.3 document. Thus, this interface enables the general export of documents from d.velop documents. The document export feature is provided in both the cloud version and the on-premises version from 7.2.2 onward (with an additionally installed d.ecs istore instance).

This makes it easy to perform direct migrations from the d.velop on-premises platform to the cloud platform. It highlights potential solutions for customers to exit the cloud platform of d.velop cloud again (exit strategy) or to perform backups themselves.

Note

As of version 8.37.0, the d.velop documents repo export API is a fixed part of the d.velop documents dmsdocs component. You can activate the application here without installing additional software.

The export API is available in the cloud via a dmsdocs app endpoint.

In on-premises systems, you can install the export API via the d.3 server tool setup.

In older d.3 server environments (version 7), the export API can update the document information to the current format at runtime. As a result, you no longer have to perform any laborious update processes before a cloud migration.

Furthermore, you can export original files directly from storage systems with the export API as of d.ecs storage manager version 3.3.0. The current API development version does not support encrypted directory structures (document trees). Therefore, original files cannot be exported from these systems.

You can operate the export API either via HTTP or through a client solution via d.velop documents repo exporter.

1.1.2. Scope of functions of d.velop documents repo export API

The export API includes the following functions and key data:

- HTTP-based REST endpoint
- Available as a separate setup without updating the system
- Available through all platforms
- Direct migration of metadata from d.3 server version 7 to the current format
- Original files can also be exported from storage systems
- Output format is already in the appropriate import format for cloud migrations
- Delta export capabilities
- Export can be restricted using filters (time, category, document IDs)
- Stateless

Logging in d.velop logfile viewer (d.3 logview)

1.1.3. Prerequisites

As of the version Current 2023.Q4 (from 8.37.0) and Continuous (cloud), the export API is already available in the dmsdocs component. You can activate the application in the configuration.

For older versions of d.velop documents or d.3, you can update the d.velop documents repo export API via the d.3 server tools setup.

Prerequisites:

- d.3 server version 7.2.2 or higher
- d.ecs jstore version 1.3 or higher
- d.ecs storage manager version 3.3 or higher, if used

1.2. Using the API functions

Below you can learn about the various functions on the user interface of the d.velop documents repo export API (export API).

1.2.1. Activating the API for the cloud

If you want to use the API for the cloud, you first have to activate the API for each d.velop tenant.

This is how it works

- 1. Contact d.velop support. As the heading, use "Activation of the export interface for multi-tenants".
- 2. Set up a user with export permissions. Use the following API to do so: Configuring user rights for the cloud

You can access the API under:

https://<tenant>.d-velop.cloud/dmsdocs/r/<repository ID>/export

1.2.2. Activating the API for on-premises

You can install the export API (on-premises) in existing d.velop documents systems at a later point in time using the tool setup. This applies to all software versions of d.3 server from 7.2.2 onward. This standalone application provides a REST-capable web server that makes the export interface and the file download interface available.

Note

As of d.3 server version **8.37.0**, the export API can also be accessed via **dmsdocs.exe** as standard. A separate installation is thus not required.

In this case, you enter the parameter **ENABLE_EXPORTS = "1"** in the file **addon.ini**. You also configure a user with export permissions.

Note

For the migration of d.3 server version 7: Before the data export, use UpdateChecker to check whether the system can be migrated.

This is how it works

1. Start the application via Windows command prompt (cmd) as follows:

```
d.velop_documents_repo_export_API.exe -c <drive name>:\<path to
d3config> -u <exportUser> -p <Port (Default: 8333)>
```

You can access the export API under the following URL:

```
http://<local IP>:<Port>/repoexport/export
```

- 2. Ensure that the user you are using exists and, depending on the version, has administrative permissions or permissions for exporting data.
- 3. Use this user for every request to the export API with basic authentication. Alternatively, you can use the API key of an authorized user if you are using the dmsdocs app for the export.

1.2.3. Defining the export filter

You may have to add the base URL to URLs that have been returned by the API.

Warning

Security information: Note that authentication information is permitted to be sent only to the configured API endpoint.

The export process consists of multiple steps:

- 1. An export job is created that describes the filter and the number of parallel export processes.
- 2. A further query is performed for each returned batch URL in the "batches" list. For more information, see: Requirements for metadata and download URLs
- 3. After you call a batch URL, the result includes a determination of whether there is another page for the batch. Any "next" link present must be followed until the returned resource does not contain any further entry for "next".
- 4. Once all the parallel requests are complete, the export ends.

Request:

Initial request for an export using a filter to limit the documents provided for the export.

- dmsdocs app: PUT https://<base URL>/dmsdocs/r/<repository ID>/export
- Standalone application: PUT http://<local IP>:8333/repoexport/export
- Content type: application/json
- Accept: application/json

Tag name	Description		
documentTypeById	Category to which the filter is applied. Use this tag for GUIDs. If you do not define a category, all the documents are selected.		
documentTypeByD3Id	Category to which the filter is applied. Use this tag for d.3 IDs. If you do not define a category, all the documents are selected.		

Tag name	Description
modifiedAfter	Timestamp in accordance with ISO 8601. This timestamp is used for delta exports and marks the last time that a document was processed (overAllProcDate). Documents that were processed at exactly the time of the timestamp are not filtered out. The timestamp must be compatible with the underlying DBMS.
modifiedBefore	Timestamp in accordance with ISO 8601. This timestamp is used for delta exports and marks the last time that a document was processed (overAllProcDate). Documents that were processed at exactly the time of the timestamp are not filtered out. The timestamp must be compatible with the underlying DBMS.
doclds	Special list of document IDs (docIds) to be exported. The server can currently process a maximum of 100 document IDs.
batchSize	Maximum block size (number of documents) that can be retrieved via a GET request.
numberOfProcesses	The number of processes planned to access the export API in parallel.
migrated	Limits the filter to documents created as a result of migration.
use_synch_id_format	Export with a unique property ID for metadata.

Response:

Content type: application/json

```
"documentsToExportCount": "164",
             "filter": {
                         "modifiedAfter": "2017-11-07T13:27:37.643Z",
                         "modifiedBefore": "2023-01-01T01:01:01.001Z",
                         "numberOfProcesses": 4,
                         "documentTypesByD3Id": [],
                         "documentTypesById": [],
                         "docIds": [],
                         "batchSize": 200
             "batches": [
                         "/dmsdocs/r/a88362cb-f626-5fc2-9293-61f6eafc4b0b/export?
startDocId=DE00000000&batchNumber=0&mB=2021-11-07T13:27:37.643Z&mA=2017-11-0
7T13:27:37.643Z&dTD3Id=1e4b6,5fa04&bS=10&nP=4&",
                         "/dmsdocs/r/a88362cb-f626-5fc2-9293-61f6eafc4b0b/export?
startDocId=G20000001&batchNumber=1&mB=2021-11-07T13:27:37.643Z&mA=2017-11-0
7T13:27:37.643Z&dTD3Id=1e4b6,5fa04&bS=10&nP=4&",
                         "/dmsdocs/r/a88362cb-f626-5fc2-9293-61f6eafc4b0b/export?
\verb|startDocId=M100000007\&batchNumber=2\&mB=2021-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27:37.643Z\&mA=2017-11-07T13:27.643Z\&mA=2017-11-07T13:27.643Z\&mA=2017-11-07T13:27.643Z\&mA=2017-11-07T13:27.643Z\&mA=2017-11-07T13:27.643Z\&mA=2017-11-07T13:27.643Z\&mA=2017-11-07T13:27.643Z\&mA=2017-11-07T13:27.643Z\&mA=2017-11-07T13:27.643Z\&mA=2017-11-07T13:27.643Z\&mA=2017-11-07T13:27.643Z\&mA=2017-11-07T13:27.643Z\&mA=2017-11-07T13:27.643Z\&mA=2017-11-07T13:27.643Z\&mA=2017-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-07-11-0
7T13:27:37.643Z&dTD3Id=1e4b6,5fa04&bS=10&nP=4&",
                         "/dmsdocs/r/a88362cb-f626-5fc2-9293-61f6eafc4b0b/export?
startDocId=M200000007&batchNumber=3&mB=2021-11-07T13:27:37.643Z&mA=2017-11-0
7T13:27:37.643Z&dTD3Id=1e4b6,5fa04&bS=10&nP=4&"
```

Tag name	Description
documentsToEx- portCount	The number of documents that meet the filter criteria.
filter	Confirmation of the included filter criteria:
batches	Links to export batches (specified relatively to the export API). The metadata and download URL are provided via GET request under these addresses. In this case, the link contains the batch number and the associated filter criteria.

1.2.4. Requirements for metadata and download URLs

In the example below, you can learn how to use the batch URLs generated in the section "Defining the export filter" to request the relevant metadata for the filter and the download URLs of the associated usage files.

You obtain the URLs for retrieving the metadata by creating an export filter or by following the "next" links in the metadata.

Note

For a complete export, you have to retrieve all the pages from all the batches provided.

Request:

- dmsdocs app: GET https://<base URL>/<URL from batch or "next" link>
- Standalone application: GET http://<local IP>/<URL from batch or "next" link>
- Accept: application/json

Response:

Sample structure of a response (content type: application/json):

```
"docs": [
            "files": [
                    "fileId": 1,
                    "filename": "DE0000000.1",
                    "downloadUrl": "/dmsdocs/r/a88362cb-
f626-5fc2-9293-61f6eafc4b0b/Document/DE00000000/files/1/root"
                    "fileId": 1,
                    "downloadUrl": "/dmsdocs/r/a88362cb-
f626-5fc2-9293-61f6eafc4b0b/Document/DE00000000/files/1/P1",
                    "dependentExtension": "P1"
            "metadata": {
                "docId": "DE00000000",
                [...]
    ],
        "errorDocs" : [
                         "docId": "DE00000001",
                        "message": "<Beschreibung eines Exportfehlers>"
        ],
    "_links": {
            "href": "/dmsdocs/r/a88362cb-f626-5fc2-9293-61f6eafc4b0b/export?
startDocId=DE00000004&batchNumber=4&mB=2021-11-07T13:27:37.643Z&mA=2017-11-0
7T13:27:37.643Z&dTD3Id=1e4b6,5fa04&bS=1&nP=4&"
        },
```

Description of the tag:

Tag name	Description
_links.self	Link to your export batch.
_links.next	Link to the next batch that is used to provide additional documents and metadata. If no link is present, this batch has been fully queried.
docs	Item containing the metadata and information for the files.
errorDocs	List of errors that occurred while processing the export batch. Documents in this list could not be fully exported.
files	Combines all the information for the files into a metadata record.
metadata	Document information in the standard import format (importable format).

1.2.5. Exporting the source system configuration

Request: Querying master data properties

- dmsdocs app: GET https://<base URL>/dmsdocs/r/<repository ID>/objectmanagement/properties
- Standalone application: GET http://<local IP>:8333/repoexport/objectmanagement/properties
- Content type: application/json

Response:

Request: Querying master data categories

- dmsdocs app: GET https://<base URL>/dmsdocs/r/<repository ID>/objectmanagement/categories
- Standalone application: GET http://<local IP>:8333/repoexport/objectmanagement/categories
- Content type: application/json

Response:

1.2.6. Configuring user rights for the cloud

You can use the HTTP REST endpoint **user** from **dmsdocs.exe** to display existing users that are known to the d.3 server. You can grant export and migration permissions to these users. This HTTP endpoint can be called only by users with administrative rights.

You can restrict user lists using query parameters. You can also display and edit individual users.

Structure of an individual user object

Main route:

https://<tenant>.d-velop.cloud/dmsdocs/r/<repository ID>/user

Link to the individual user object:

d.3 ID of the user:

```
"d3Id": "user1",
```

IDP ID of the user:

```
"idpId": "7B841E93-EC4E-4790-B9D7-AD7F5DFCC82B"
```

Define whether you want the user to have export permissions:

```
"hasExportRight": true,
```

Define whether you want the user to have migration permissions:

```
"hasMigrationRight": true,
}
```

When you change a user permission, call the self link of the user object using a PUT request. Send the received information to the user as the JSON body in the request as well.

GET requests and query parameters

Objective: Querying all users

Request/subroute: GET https://<tenant>.d-velop.cloud/dmsdocs/r/<repository ID>/user

Sample end point response:

```
"user": [
            "_links": {
                "self": {
                     "href": "https://<baseUrl>/dmsdocs/r/<repoUuid>/user/
d3Id/user1",
                     "templated": false
            },
            "d3Id": "user1",
            "hasExportRight": true,
            "hasMigrationRight": true,
            "idpId": "7B841E93-EC4E-4790-B9D7-AD7F5DFCC82B"
        },
            "_links": {
                "self": {
                     "href": "https://<baseUrl>/dmsdocs/r/<repoUuid>/user/
d3Id/user2",
                     "templated": false
            "d3Id": "user2",
            "hasExportRight": false,
            "hasMigrationRight": false,
            "idpId": ""
        },
            "_links": {
                 "self": {
                     "href": "https://<baseUrl>/dmsdocs/r/<repoUuid>/user/
d3Id/user3",
                     "templated": false
            "d3Id": "user3",
            "hasExportRight": false,
            "hasMigrationRight": false,
            "idpId": ""
```

```
}
```

Objective: Displaying users with migration permissions

Request/subroute: **GET https://<tenant>.d-velop.cloud/dmsdocs/r/<repository ID>/user?hasMigration-Right**

Sample end point response:

Objective: Displaying users without migration permissions

Request/subroute: **GET https://<tenant>.d-velop.cloud/dmsdocs/r/<repository ID>/user?hasMigration-Right=false**

Sample end point response:

```
"user": [
            "_links": {
                "self": {
                     "href": "https://<baseUrl>/dmsdocs/r/<repoUuid>/user/
d3Id/user2",
                     "templated": false
            },
            "d3Id": "user2",
            "hasExportRight": false,
            "hasMigrationRight": false,
            "idpId": ""
        },
            "_links": {
                "self": {
                     "href": "https://<baseUrl>/dmsdocs/r/<repoUuid>/user/
d3Id/user3",
                     "templated": false
```

```
},
    "d3Id": "user3",
    "hasExportRight": false,
    "hasMigrationRight": false,
    "idpId": ""
}
```

Objective: Displaying users with export permissions

Request/subroute: **GET https://<tenant>.d-velop.cloud/dmsdocs/r/<repository ID>/user?hasExport-Right**

Sample end point response:

Objective: Displaying users without export permissions

Request/subroute: **GET https://<tenant>.d-velop.cloud/dmsdocs/r/<repository ID>/user?hasExport-Right=false**

Sample end point response:

Objective: Displaying users with IDP IDs

Request/subroute: GET https://<tenant>.d-velop.cloud/dmsdocs/r/<repository ID>/user?hasldpld

Sample end point response:

```
"user": [
      " links": {
        "self": {
          "href": "https://<tenant>.d-velop.cloud/dmsdocs/r/<repoId>/user/
d3Id/user1",
          "templated": false
      },
      "d3Id": "user1",
      "hasExportRight": false,
      "hasMigrationRight": true,
      "idpId": "7B841E93-EC4E-4790-B9D7-AD7F5DFCC82B"
    },
      " links": {
        "self": {
          "href": "https://<tenant>.d-velop.cloud/dmsdocs/r/<repoId>/user/
d3Id/user2",
          "templated": false
        }
      },
      "d3Id": "user2",
      "hasExportRight": false,
      "hasMigrationRight": false,
      "idpId": "61C15487-3A24-4FC0-B105-F3141114168D"
    },
  ]
```

Objective: Displaying users without IDP IDs

Request/subroute: **GET https://<tenant>.d-velop.cloud/dmsdocs/r/<repository ID>/user?hasl-dpld=false**

Sample end point response:

```
"user": [
      "_links": {
        "self": {
          "href": "https://<tenant>.d-velop.cloud/dmsdocs/r/<repoId>/user/
d3Id/user1",
          "templated": false
        }
      },
      "d3Id": "d3_wfl",
      "hasMigrationRight": false,
      "idpId": ""
    },
      "_links": {
        "self": {
          "href": "https://<tenant>.d-velop.cloud/dmsdocs/r/<repoId>/user/
d3Id/user1",
          "templated": false
        }
      },
      "d3Id": "jhor",
      "hasMigrationRight": false,
      "idpId": ""
    },
  ]
```

Objective: Querying individual users

Request/subroute: **GET https://<tenant>.d-velop.cloud/dmsdocs/r/<repository ID>/user/d3ld/user1**Sample end point response:

PUT requests

Objective: Granting and revoking migration permissions

Request/subroute: PUT https://<tenant>.d-velop.cloud/dmsdocs/r/<repository ID>/user/d3ld/<d3ld>
JSON body for request:

```
{
  "d3Id": "user1",
  "hasExportRight": true,
  "hasMigrationRight": false,
  "idpId": "7B841E93-EC4E-4790-B9D7-AD7F5DFCC82B"
}
```

Sample end point response:

1.2.7. Configuring user rights for on-premises

To use export API, users require the appropriate permissions.

Warning

Users thus get the permission to export each document. This permission is valid regardless of any restrictions to document types or compulsory watermarks applied.

It applies to both users of the standalone d.velop documents repo export API and users of the dmsdocs app.

User permissions for exporting data

The permissions you require for a full data export differ based on the version of the system to be exported. The standalone application requires a d.3 user for the data export. IDP users cannot be granted the required permissions.

Current versions

For the following versions of d.velop documents, users require the special **data export** permission. You can grant the permission from the d.3 administration.

- Annual 2022 Patch 09 and later (from 8.21.18)
- Current 2023.Q2 Patch 03 and later (from 8.32.16)
- Beta 2023/08/10 (from 8.33.1)
- Continuous (cloud) (from 8.36.0)

Out-of-date versions

In older systems, there is a check of whether users have the **administrator permission**. You have to use the same user when you start d.velop documents repo export API.

1.2.8. Using the d.velop documents repo export client

d.velop documents repo exporter is a client solution that can communicate directly with the export API (cloud and on-premises). d.velop documents repo exporter lets you simplify data exports for your customers.

This is a client application that you can start from the Windows command prompt. You can obtain the client via the d.3 server tools package. The client abstracts the HTTP communication to the export API and loads the configured documents to a definable target directory.

The client application provides the following functions:

- You can set the client using the configuration file and start it with appropriate parameters.
- You can use d.velop logfile viewer (d.3 logview) for logging. You can also specify that messages are logged in a separate text file.

Prerequisites:

- The export API is accessible, installed and activated.
- An export directory on which the metadata and the original files can be saved is available. During the export, the documents are saved based on the directory structure (document tree).

Structure of the "Config.json" file

The configuration file includes the following:

- The user information for the source system (on-premises: user name and password, cloud: repository ID and API key)
- URL at which d.velop documents repo exporter can be accessed.
- Filter for defining the documents to be exported.

Note

The base address of the export API is logged in the output at the start.

The configuration file is structured as follows:

```
"baseUrl" : "<http://127.0.0.1:8333>",
"user" : "<exportUser>",
"password" : "<password>",
"repoId" : "<uuid>",
"apiKey" : "<apiKey>",
"timeout" : "<seconds>",
"proxyPort" : "<port>",
"proxyHost" : "<ProxyIp>",
"filter" : {
    "modifiedAfter": "<2017-11-07T13:27:37.643Z>",
    "modifiedBefore": "<2023-01-01T01:01:01.001Z>",
    "numberOfProcesses": <2>,
    "documentTypesByD3Id": [],
    "documentTypesById": [],
    "docIds": [],
    "batchSize": <200>
}
```

Unless otherwise defined, you can also enter all the parameters from the command prompt in the configuration file.

Below you can find descriptions of the parameters in the configuration file:

Parameter	Description		
user	User name of the user with permissions for exports		
	More information about distributing permissions: Configuring user rights for the cloud		
password	Optional: Password of the user exporting the files		
аріКеу	Relevant for the cloud: API key of the user with permissions for exports		
timeout	Optional: Request Timeout		
	Default: 300 seconds		
filter	Restricting filter for the data export		
	More information: Defining the export filter		

Starting the client through a Windows command prompt

You can start d.velop documents repo exporter with the Windows command prompt (**cmd**). Add the following command prompt and adapt the values to your requirements:

export_client.exe --configFile <pathToConfig.json> <optionale Parameter>

The export is performed automatically after the application starts.

Below you can find descriptions of the parameters of the command prompt:

Parameter	Op- tional	Description
configFile	n	Path to the configuration file for the client. Must be transferred when the application is started.
baseUrl	n	Base address of the d.velop documents repo exporter API.
exportPath	n	Path to the target directory for the export (saving the original file and the associated metadata) Must be transferred in the command prompt or configuration file.
repold	-	 Repository ID of the d.velop documents system to be exported. Mandatory when using the export API embedded in d.velop documents. This applies to current d.velop documents systems and exports from the cloud. If you are using the standalone d.velop documents repo export API, you must leave this configuration value out. The d.velop documents system to be exported is defined using the d.velop documents repo exporter API start parameters.
exportWithFile- type	n	This configuration enables exports retaining known file extensions. This mode is not suitable for subsequent migrations or imports to d.velop documents.
exportMetada- taOnly	n	Only the metadata from d.velop documents is exported. This mode can be used to analyze metadata, for example.
exportReposi- toryConfigOnly	n	The client requests only the configuration of the d.velop documents system to be exported and closes after the configuration is saved.
logFile (optional)	n	Path and name of a log file. The file contains the whole export log and is added to on an ongoing basis. The file is not rolling and can therefore take up a large amount of storage space. We recommend producing a backup copy of the file before running d.velop documents repo exporter again.
logLevel (option- al)	n	Log level (default: 6) The values are between 0 and 9 (0: no logging, 9: maximum debug logging).
proxyHost	n	Host name or IP of an HTTP/HTTPS proxy. Authentication on the proxy is not supported.
proxyPort	n	Port of an HTTP/HTTPS proxy
ignoreCertifi- cates	n	Allows the use of invalid and self-signed certificates.
startLink	n	Use this parameter if a previous export failed and you know the URL of the last successful block. This start parameter bypasses any defined filters.
help	n	Displays a help text for the start parameters and closes the application.

Parameter	Op- tional	Description
useSynchIdFor- mat	-	This parameter is used for the export of the metadata with a unique property ID. No: ${\bf n}$, Yes: ${\bf y}$.

How to export from the cloud: Enter the relevant values in the configuration file for the parameters **repold** and **apikey**.

The documents are then saved in the specified folder (**exportPath**) based on the directory structure (document tree), together with the JSON display of the metadata. If you selected the applicable option, the master data is also saved in the specified folder.

Behavior in the event of errors

Information about exported documents is saved in the files **error_log_<number>.json**. The files are numbered sequentially. The files are created after 1,000,000 documents or after 100 errors. After 100,000 documents, a log is written.

Example of an error in "error_log_0.json":

```
"errorBatches": [
      "filter": {
        "docIds": [
          "P00000005",
          "P00000079",
          "P000009725",
          "P000009729",
          "P000009731",
          "P000009733",
          "P000009735",
          "P000009737",
          "P000009739",
          "P000009742"
        ]
      },
      "link": "/repo/export?
startDocId=P000000005&batchNumber=0&docIds=P000000005,P000000079,P000009725,
P000009729, P000009731, P000009733, P000009735, P000009737, P000009739, P000009742
&bS=100"
    }
```

Description:

- filter: When the tool is restarted, you can use this parameter to reexport specific document IDs that were not exported.
- **link**: This parameter is the endpoint that caused the error. You can use the endpoint for debugging with external tools.