clausematch

Public API Overview

Contents

Introduction

ClauseMatch is an award-winning London based regulatory technology company. The product is a unique software platform that allows people in Compliance, Finance, Legal and Risk to collaborate on documents faster and more efficiently while solving multiple organisational workflow problems.

It works as a browser-based collaborative document editor whereby all comments, approvals and changes made in a document can be tracked with a full audit trail, allowing unprecedented reporting capabilities.

All communications are performed via REST based APIs.

It is important to notice here that ClauseMatch does not store documents as files on a file system, instead ClauseMatch document content is represented as machine-readable data and stored as database records. The key difference between ClauseMatch and other document management solutions is that in ClauseMatch "paragraph" is a fundamental concept of utmost importance. It is a building block for documents, the following points summarise relevant considerations on this matter:

- ClauseMatch documents have a well-defined and rigid structure.
- Paragraphs are key elements of the document structure.
- Every document section and every document heading need to be a separate paragraph.
- Paragraphs are organized in levels thereby defining document structure and numbering.
- In ClauseMatch, a paragraph is not just a piece of text, but a separate entity with its own timeline of changes, comments and approvals.
- User permissions and approvals can be configured not only at a document level but also for each individual paragraph.

1. Common API

1.1. Authentication

JWT authentication is used to access ClauseMatch Public API.

2. Content API

2.1. Get a list of categories

2.1.1. Description

Call this endpoint to retrieve the set of existing categories and use them to search for documents. Categories in ClauseMatch acts similar to folders in a file system. The difference is that a document can belong to more than one category at the same time. Lists of categories form tree-like structures with categories and sub-categories similar to folders and subfolders.

2.1.2. Request

/api/v1/categories

2.1.3. Request Method

HTTP GET

2.1.4. Request Parameters

Name	Type	Description
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.

2.1.5. Response example

```
Status: 200 OK
{
  "current_page": 1,
  "total_pages": 1,
  "requested_page_size": 50,
  "actual_page_size": 3,
  "results": [
    {
      "id": "5a8a1283-aa12-4f57-ac59-3c68bba8526b",
       "name": "Risks"
    },
    {
      "id": "62e35964-788b-4ca3-b865-ec50d1b01727",
      "parent_category_id": "5a8a1283-aa12-4f57-ac59-3c68bba8526b",
      "name": "Operational Risks"
    },
    {
      "id": "c76168ea-11a2-4dd5-98b5-6706735756a4",
      "parent_category_id": "62e35964-788b-4ca3-b865-ec50d1b01727",
      "name": "People Risks"
    }
  ]
}
```

2.2. Get a list of metadata

2.2.1. Description

Call this endpoint to retrieve the list of existing metadata and use it to search for documents. Metadata is information about the document. Metadata is a set of typed key-value pairs associated with a document. "Next review date", "dispensation date", "document type", "policy owner" are some examples of document metadata could be used for.

Currently supported types of metadata values are:

- "Text field"
- "Number"
- "Tags"
- "Link field"
- "Checkbox"
- "Date field"
- "Field with options".

When searching for documents you can provide any values for "Text field", "Number", "Tags", "Link field" and "Date field". "Checkbox" supports 2 options "true" and "false". "Field with options" supports values from a predefined list returned for this metadata.

2.2.2. Request URL

/api/v1/metadata

2.2.3. Request Method

HTTP GET

2.2.4. Request Parameters

Name	Type	Description
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.

2.2.5. Response example

Status: 200 OK

```
{
   "current_page": 1,
   "total_pages": 1,
   "requested_page_size": 50,
   "actual_page_size": 3,
   "results": [
    {
      "id": "ed6f21a3-2c76-4735-8d41-08cf7e1d02b2",
      "name": "Impact",
      "type": "field_with_options",
      "allowed_values": [ "Low", "Medium", "High" ]
    },
    {
      "id": "aec8614f-eb83-4c4a-9901-c11a6df55221",
      "name": "Summary",
      "type": "text_field"
    },
    {
      "id": "480f9687-458c-4288-a2ac-9d388f722f25",
      "name": "Review Date",
      "type": "date_field"
    }
  ]
}
```

2.3. Get a list of tags

2.3.1. Description

Call this endpoint to retrieve a list of existing tags and use them to search for documents.

Tags help organize information and make it easier to browse for a specific content. Users can add tags to documents, paragraph and data connections thereby giving themselves ability to search and filter the corresponding elements by tags.

Tags in ClauseMatch are similar to tags in other systems. Each tag has a name, colour and is associated with a tag group. Tag groups are usually used for grouping of logically related tags.

2.3.2. Request URL

/api/v1/tags

2.3.3. Request Method

HTTP GET

2.3.4. Request Parameters

Name	Type	Description
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.

2.3.5. Response example

```
Status: 200 OK
{
  "current_page": 1,
  "total_pages": 1,
  "requested_page_size": 50,
  "actual_page_size": 3,
  "results": [
    {
      "id": "31e2a143-9470-4fa2-9fa9-1f7bbf3bb29d",
      "name": "Minor",
      "group_name": "Priority",
      "colour": "#00ff00"
    },
    {
      "id": "296ab290-fdd1-46ee-b6b6-6d5bb75d8941",
      "name": "Major",
      "group_name": "Priority",
      "colour": "#ff0000"
    },
    {
      "id": "5fd96484-02e7-45af-8957-63c42a5c3e06",
      "name": "TODO",
      "group_name": "Markers"
    }
  ]
}
```

2.4. Get a list of users

2.4.1. Description

Call this endpoint to retrieve a list of existing users. Users in ClauseMatch are associated with individuals who have access to the system. Each action or change in ClauseMatch is always associated with a user.

System utilises the concept of user to:

- specify individuals who have access to documents.
- associate each change, comment, approval, etc. with its author and record in the audit trail.
- request document- and paragraph-level approvals.
- specify granular visibility for paragraphs and comments.

2.4.2. Request URL

/api/v1/users

2.4.3. Request Method

HTTP GET

2.4.4. Request Parameters

Name	Type	Description
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.
include_disabled	boolean	Specifies if disabled users should be included in results or not. **Optional.** Defaults to "false".

2.4.5. Response example

```
Status: 200 OK
{
   "current_page": 1,
   "total_pages": 1,
   "requested_page_size": 50,
   "actual_page_size": 2,
   "results": [
      {
         "id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
         "email": "john.doe@clausematch.com",
         "first_name": "John",
         "last_name": "Doe",
         "enabled": true
      },
      {
         "id": "d0badc5a-1079-4766-aec7-b0677ff60269",
         "email": "jane.doe@clausematch.com",
         "first_name": "Jane",
         "last_name": "Doe",
         "enabled": false
      }
   ]
}
```

2.5. Get a list of groups

2.5.1. Description

Call this endpoint to retrieve a list of existing user groups. Each user group has a name and contains a set of different users. In ClauseMatch a user can belong to multiple groups simultaneously.

Similar to users, groups can be used to:

- specify who has access to a document.
- request document- and paragraph-level approvals.
- specify granular visibility for paragraphs and comments.

2.5.2. Request URL

/api/v1/groups

2.5.3. Request Method

HTTP GET

2.5.4. Request Parameters

Name	Type	Description
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.

2.5.5. Response example

Status: 200 OK

```
{
  "current_page": 1,
  "total_pages": 1,
  "requested_page_size": 50,
  "actual_page_size": 1,
  "results": [
    {
      "id": "624507d7-2dd5-442d-8dd8-6090a6c7eb72",
      "name": "Security Department",
      "users": [
        {
          "id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
          "email": "john.doe@clausematch.com",
          "first_name": "John",
          "last_name": "Doe"
        }
      ]
    }
  ]
}
```

2.6. Documents search

2.6.1. Description

Documents are an essential part of ClauseMatch. Use this endpoint to search for them. ClauseMatch supports a very granular and powerful full-text, google-like search. Any information associated with documents can be used as a search term.

The following is the list of attributes which can be used in the search:

- The query for a full-text search in document titles or content.
- Categories.
- Tags.
- Metadata and their values.
- Document owners.
- Document stages.

2.6.2. Request URL

/api/v1/documents/search

2.6.3. Request Method

HTTP POST

2.6.4. Request Parameters

Name	Type	Description
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.
query *	string	Specifies query that should match resulting document title or paragraphs content. **Optional.** Defaults to <undefined>.
category_id *	[string]	Specifies category ID that should be associated with resulting document. **Repeatable. Optional.** Defaults to <undefined>.
metadata_id *	[string]	Specifies metadata ID that should be associated with resulting document. **Repeatable. Optional.** Defaults to <undefined>.
tag_id *	[string]	Specifies tag ID that should be associated with resulting document. **Repeatable. Optional.** Defaults to <undefined>.
owner_id *	[string]	Specifies owner (user or group) ID that should be associated with resulting document. **Repeatable. Optional.** Defaults to <undefined>.

* At least one of "query", "category_id", "metadata_id", "tag_id" or "owner_id" parameter should be specified in request.

2.6.5. Response example

```
Status: 200 OK
{
  "current_page": 1,
  "total_pages": 1,
  "requested_page_size": 50,
  "actual_page_size": 1,
  "results": [
    {
      "id": "05ec9a73-494a-444e-9f11-397532c561ed",
      "type": "regular_document",
      "latest_version": "1.0.14",
      "latest_title": "Metrics & Alerting",
      "latest_stage": "Release",
      "latest_categories": [
        {
          "category_id": "c76168ea-11a2-4dd5-98b5-6706735756a4",
          "category_name": "People Risks"
        }
      ]
      "created": {
        "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
        "user_email": "john.doe@clausematch.com",
        "user_first_name": "John",
        "user_last_name": "Doe",
        "date": "2019-02-15T08:42:14.123"
      },
      "last_modified": {
        "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
        "user_email": "john.doe@clausematch.com",
        "user_first_name": "John",
        "user_last_name": "Doe",
        "date": "2020-04-30T18:50:24.866"
      }
    }
  ]
}
```

2.7. Get a list of versions for a document

2.7.1. Description

Call this endpoint to retrieve a list of existing versions for a document. Each document always has at least one version. The document id and version together create a unique reference. The latest version represents the current state of the document.

In ClauseMatch each document is usually represented with many versions. Think of it as if a document is a VCS branch and a version is a VCS commit. Multiple versions are joined under a single document object with a unique id providing users with the ability to always refer to a single document object despite any changes.

Usually, each version is associated with changes in the document content or in attributes like metadata, tags, data connections, etc. In ClauseMatch versions are represented by two integer numbers separated with a dot. The first number is the document's major version and the second one is the minor version. E.g.: 1.7, where 1 stands for the major version and 7 for the minor.

Please keep in mind that different versions of the same document can have different titles.

2.7.2. Request URL

/api/v1/documents/{document_id}/versions

2.7.3. Request Method

HTTP GET

2.7.4. Request Parameters

Name	Type	Description
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.

2.7.5. Response example

```
Status: 200 OK
{
  "current_page": 1,
  "total_pages": 1,
  "requested_page_size": 50,
  "actual_page_size": 1,
  "results": [
    {
      "id": "05ec9a73-494a-444e-9f11-397532c561ed",
      "version": "1.0.14",
      "title": "Metrics & Alerting",
      "created": {
        "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
        "user_email": "john.doe@clausematch.com",
        "user_first_name": "John",
        "user_last_name": "Doe",
        "date": "2019-02-15T08:42:14.123"
      },
      "last_modified": {
        "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
        "user_email": "john.doe@clausematch.com",
        "user_first_name": "John",
        "user_last_name": "Doe",
        "date": "2020-04-30T18:50:24.866"
      }
    }
  ]
}
```

2.8. Get basic attributes for a document version

2.8.1. Description

Call this endpoint to get basic information about a document version.

Here are some of the attributes associated with each document version:

- unique document id and version number.
- title.
- dates and users related to the initial version creation and the last modification.
- document stage.
- id of the template which was used to create this document.
- languages used in the document version.
- categories where the document sits.

2.8.2. Request URL

/api/v1/documents/{document_id}/versions/{document_version}

2.8.3. Request Method

HTTP GET

2.8.4. Response example

Status: 200 OK

```
{
  "id": "05ec9a73-494a-444e-9f11-397532c561ed",
  "type": "regular_document",
  "version": "1.0.14",
  "title": "Metrics & Alerting",
  "stage": "Final",
  "language_settings": [ "en_US", "it_IT" ],
  "categories": [
    {
      "category_id": "c76168ea-11a2-4dd5-98b5-6706735756a4",
      "category_name": "People Risks"
    }
  ],
  "created": {
    "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
    "user_email": "john.doe@clausematch.com",
    "user_first_name": "John",
    "user_last_name": "Doe",
    "date": "2019-02-15T08:42:14.123"
  },
  "last_modified": {
    "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
    "user_email": "john.doe@clausematch.com",
    "user_first_name": "John",
    "user_last_name": "Doe",
    "date": "2020-04-30T18:50:24.866"
  }
}
```

2.9. Get content for a document version

2.9.1. Description

Call this endpoint to get document content. ClauseMatch represents a document as a tree-like set of paragraphs. Each top-level paragraph can have as many children as required. However, the depth of the paragraph tree is limited to 7 levels.

The following information is essential to understand how ClauseMatch stores document content.

Similarly to documents, each paragraph has unique id and version. Combination of these attributes can be used in a URL which would point to a particular version of a paragraph.

Paragraph content is a normalised HTML string with embedded styles. It can be rendered in the same way as any other HTML content.

Paragraphs are returned as an ordered list which is equivalent to their order in a document.

A level identifies the number of right indents required to be applied to a paragraph. These indents are similar to "Indent right" in any other text editor. So if the level equals 3 then all paragraph content should be indented to the right on 3 x [default right indent width]. Default right indent width should be chosen based on your preferences. We recommend to set it to 20 px.

Numbering properties define if the paragraph has a number and its value. Numbering should always be displayed in front of the paragraph content.

The paragraph type can have one of the following values:

- header - a paragraph that represents a document header for each page
- title page - a paragraph that represents the title page of a document
- headings from 1 to 5 - document headings which are used to create logical sections and a table of content
- normal - usual paragraph
- snippet - immutable paragraph derived from another document
- footnote - footnote paragraph which content is associated with some other paragraph in the document
- page brake - simple page brake indicating that the next paragraph should start from the new page
- TOC - table of content
- footer - a paragraph that represents a document footer for each page

2.9.2. Request URL

/api/v1/documents/{document_id}/{document_version}/paragraphs

2.9.3. Request Method

HTTP GET

2.9.4. Request Parameters

Name	**Type**	**Description**
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.

2.9.5. Response example

Status: 200 OK

```
{
  "current_page": 1,
  "total_pages": 1,
  "requested_page_size": 50,
  "actual_page_size": 1,
  "results": [
    {
      "id": "1692bc74-bc3b-4a46-8dce-00347b070d4f",
      "type": "normal_paragraph",
      "level": 1,
      "numbering": "1.",
      "content": "<b>Lorem ipsum</b> dolor sit amet, consectetur adipiscing elit. Etiam eget orci risus.
Praesent sollicitudin felis ut lectus ultrices, ut aliquam elit suscipit. Suspendisse gravida risus vel neque
euismod, finibus ullamcorper leo cursus. Nunc vitae consectetur ipsum. Nunc hendrerit eget ante congue
pulvinar. Vestibulum tincidunt mattis elit at maximus. Maecenas faucibus, ex sit amet venenatis feugiat, felis
velit pellentesque mi, id mollis justo libero ac leo. Sed vitae dolor nec enim sodales porttitor. Etiam iaculis
arcu at vestibulum tempor. Nulla est ipsum, feugiat ac justo tristique, feugiat commodo est.",
            "created": {
               "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
               "user_email": "john.doe@clausematch.com",
               "user_first_name": "John",
               "user_last_name": "Doe",
               "date": "2019-02-15T08:42:14.123",
            },
            "last_modified": {
               "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
               "user_email": "john.doe@clausematch.com",
               "user_first_name": "John",
               "user_last_name": "Doe",
               "date": "2020-04-30T18:50:24.866"
            }
         }
      ]
   }
```

2.10. Get metadata for a document version

2.10.1. Description

Call this endpoint to retrieve a list of metadata and their values associated with the specified document version. Metadata is a set of typed key-value pairs associated with a document.

Currently supported types are:

- "Text field"
- "Number"
- "Tags"
- "Link field"
- "Checkbox"
- "Date field"
- "Field with options".

2.10.2. Request URL

/api/v1/documents/{document_id}/versions/{document_version}/metadata

2.10.3. Request Method

HTTP GET

2.10.4. Request Parameters

Name	Type	Description
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.

2.10.5. Response example

```
Status: 200 OK
{
  "current_page": 1,
  "total_pages": 1,
  "requested_page_size": 50,
  "actual_page_size": 1,
  "results": [
    {
      "metadata_id": "ed6f21a3-2c76-4735-8d41-08cf7e1d02b2",
      "metadata_name": "Impact",
      "metadata_type": "field_with_options",
      "metadata_value": "Low"
    },
    {
      "metadata_id": "aec8614f-eb83-4c4a-9901-c11a6df55221",
      "metadata_name": "Summary",
      "metadata_type": "text_field",
      "metadata_value": "Document about metrics and alerting"
    }
  ]
}
```

2.11. Get tags for a document version

2.11.1. Description

Tags help organize information and make it easier to browse for a specific content. Users can add tags to documents, paragraph to have ability to search and filter content by tags.

Call this endpoint to retrieve a list of document- and paragraph-level tags associated with the specified document version.

2.11.2. Request URL

/api/v1/documents/{document_id}/versions/{document_version}/tags

2.11.3. Request Method

HTTP GET

2.11.4. Request Parameters

Name	Type	Description
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.
scope	string	Specifies scope that should be associated with resulting tag. **Optional.** Defaults to <any>. Allowed values: "document", "paragraph".
paragraph_id	string	Specifies paragraph ID that should be associated with resulting tag. **Optional.** Defaults to <undefined>. Not read unless [scope=paragraph] is specified.

2.11.5. Response example

Status: 200 OK

```
{
  "current_page": 1,
  "total_pages": 1,
  "requested_page_size": 50,
  "actual_page_size": 2,
  "results": [
    {
      "scope": "document",
      "tag_id": "31e2a143-9470-4fa2-9fa9-1f7bbf3bb29d",
      "tag_name": "Minor",
      "tag_group_name": "Priority",
      "tag_colour": "#00ff00"
    },
    {
      "scope": "paragraph",
      "paragraph_id": "1692bc74-bc3b-4a46-8dce-00347b070d4f",
      "tag_id": "5fd96484-02e7-45af-8957-63c42a5c3e06",
      "tag_name": "TODO",
      "tag_group_name": "Markers"
    }
  ]
}
```

2.12. Get data connections for a document version

2.12.1. Description

Data Connection functionality allows users to create mapping between related paragraphs and/or documents across the platform.

2.12.2. Request URL

/api/v1/documents/{document_id}/versions/{document_version}/connected

2.12.3. Request Method

HTTP GET

2.12.4. Request Parameters

Name	Type	Description
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.
scope	string	Specifies scope that should be associated with resulting connection. **Optional.** Defaults to <any>. Allowed values: "document", "paragraph".
paragraph_id	string	Specifies paragraph ID that should be associated with resulting connection. **Optional.** Defaults to <undefined>. Not read unless [scope=paragraph] is specified.

2.12.5. Response example

```
Status: 200 OK
{
  "current_page": 1,
  "total_pages": 1,
  "requested_page_size": 50,
  "actual_page_size": 2,
  "results": [
    {
      "scope": "document",
      "connected_document_id": "24c819e8-bade-4ef1-aec8-a0eac0738bcf",
      "connected_document_title": "Other Metrics",
      "connected_document_version": "4.2"
    },
    {
      "scope": "paragraph",
      "paragraph_id": "1692bc74-bc3b-4a46-8dce-00347b070d4f",
      "connected_document_id": "41d66035-5f7c-4f96-a48e-c3c88806dbf7",
      "connected_document_title": "Other Metrics",
      "connected_document_version": "42",
      "connected_paragraph_id": "650cfd47-dc3c-4e84-8272-cd31afbab25f"
    }
  ]
}
```

3. Documents export API

3.1. Export a document revision to PDF

3.1.1. Description

This endpoint should be used to provide capabilities for end-users to print the documents or to save them locally on their computer. Documents can be exported as PDF or DOCX files.

3.1.2. Request URL

/api/v1/documents/{document_id}/versions/{document_version}/pdf

3.1.3. Request Method

HTTP GET

3.1.4. Request Parameters

None

3.1.5. Response example

Status: 303 See Other
Location: https://mycompany.clausematch.com/api/v1/files/1637f172230948518c7157854a6c008e

3.2. Export a document revision to DOCX

3.2.1. Description

This endpoint should be used to provide capabilities for end-users to print the documents or to save them locally on their computer. Documents can be exported as PDF or DOCX files.

3.2.2. Request URL

/api/v1/documents/{document_id}/versions/{document_version}/pdf

3.2.3. Request Method

HTTP GET

3.2.4. Request Parameters

None

3.2.5. Response example

Status: 303 See Other
Location: https://mycompany.clausematch.com/api/v1/files/3e1161d0a0fb45dc9d9a76c54c5cb00c

4. Policy Management API

4.1. Get a list of stages for a document

4.1.1. Description

Document stages define document workflow, help track its progress and trigger notifications for the upcoming stage deadlines (e.g. publication deadline).

As the work on a document progresses, documents owner(s) move the document to the next stage.

Stages are linear and have a few important properties:

- "Name" - the title of the stage.
- "Expiration date" - stage deadline.
- "Requirements" which specifies if any mandatory document level approvals must be acquired before moving to the next stage.
- "Is document editable" which specifies if users are allowed to modify document content in this stage.
- "Is current stage" - attribute that identifies if this stage represents the current stage of a document.

4.1.2. Request URL

/api/v1/documents/{id}/policy-management/stages

4.1.3. Request Method

HTTP GET

4.1.4. Request Parameters

Name	Type	Description
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.

4.1.5. Response example

```
Status: 200 OK
{
  "current_page": 1,
  "total_pages": 1,
  "requested_page_size": 50,
  "actual_page_size": 4,
  "results": [
    {
      "stage_idx": 1,
      "stage_name": "Draft",
      "stage_is_mandatory": false,
      "stage_allows_content_editing": true,
      "stage_is_final": false
    },
    {
      "stage_idx": 2,
      "stage_name": "Review",
      "stage_is_mandatory": true,
      "stage_allows_content_editing": true,
      "stage_is_final": false
    },
    {
      "stage_idx": 3,
      "stage_name": "Pre-Release",
      "stage_is_mandatory": false,
      "stage_allows_content_editing": false,
      "stage_is_final": false
    },
    {
      "stage_idx": 4,
      "stage_name": "Pre-Release",
      "stage_is_mandatory": true,
      "stage_allows_content_editing": false,
      "stage_is_final": true
    }
  ]
}
```

4.2. Get Policy Management access rights (permissions) for a document

4.2.1. Description

Document permissions define users who have access to a document.

The following roles are supported:

- Owner - have full access and control over every document aspect.
- Editor - Editor's role mainly is limited to editing. As an Editor the user can read and edit a document, delete and restore paragraphs, upload and download file attachments, view document settings, metadata and user permissions, read and write comments, review a document, compare versions and view approvals etc.
- Reviewer - as a viewer the user can view a document, read and write comments, track changes, compare versions and view the timeline and approvals etc.
- Read Published - a Read Published user can only view the document/template after it has been published. A Read Published user cannot make any changes to a document.

4.2.2. Request URL

/api/v1/documents/{id}/policy-management/permissions

4.2.3. Request Method

HTTP GET

4.2.4. Request Parameters

Name	Type	Description
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.

4.2.5. Response example

Status: 200 OK

```
{
  "current_page": 1,
  "total_pages": 1,
  "requested_page_size": 50,
  "actual_page_size": 2,
  "results": [
    {
      "type": "user",
      "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
      "user_email": "john.doe@clausematch.com",
            "user_first_name": "John",
            "user_last_name": "Doe",
            "role": "owner"
        },
        {
            "type": "group",
            "group_id": "624507d7-2dd5-442d-8dd8-6090a6c7eb72",
            "group_name": "Security Department",
            "role": "reviewer"
        }
    ]
}
```

4.3. Get document and paragraph level approvals for a document revision

4.3.1. Description

If the document version is to be approved by stakeholders before publication, the approval process can be set up in ClauseMatch.

Use this endpoint to retrieve document- and paragraph-level approvals for a document version.

The following information is important for understanding ClauseMatch approvals:

- Approvals can be assigned to individual users or user groups
- Each approval can be associated with a paragraph or a whole document
- Each approval can have one of the following states: "pending", "approved", "rejected"
- A simple note (a string of text) can present for approvals in the "approved" or "rejected" state

4.3.2. Request URL

/api/v1/documents/{document_id}/versions/{document_version}/policy-management/approvals

4.3.3. Request Method

HTTP GET

4.3.4. Request Parameters

Name	Type	Description
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.
scope	string	Specifies scope that should be associated with resulting approval. **Optional.** Defaults to <any>. Allowed values: "document", "paragraph".
paragraph_id	string	Specifies paragraph ID that should be associated with resulting approval. **Optional.** Defaults to <undefined>. Not read unless [scope=paragraph] is specified.

4.3.5. Response example

```
Status: 200 OK
{
  "current_page": 1,
  "total_pages": 1,
  "requested_page_size": 50,
  "actual_page_size": 2,
  "results": [
    {
      "scope": "document",
      "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
      "user_email": "john.doe@clausematch.com",
      "user_first_name": "John",
      "user_last_name": "Doe",
      "status": "pending"
    },
    {
      "scope": "paragraph",
      "paragraph_id": "1692bc74-bc3b-4a46-8dce-00347b070d4f",
      "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
      "user_email": "john.doe@clausematch.com",
      "user_first_name": "John",
      "user_last_name": "Doe",
      "status": "accepted",
      "note": "Good enough."
    }
  ]
}
```

4.4. Get document and paragraph level comments with replies for a document version

4.4.1. Description

During document drafting and review, owners, reviewers and editors can leave new comments and replies to existing comments.

Use this endpoint to access their comments.

Each comment can be associated with a particular paragraph or be related to the document itself. A comment can have multiple replies posted by different users.

4.4.2. Request URL

/api/v1/documents/{document_id}/versions/{document_version}/policy-management/comments

4.4.3. Request Method

HTTP GET

4.4.4. Request Parameters

Name	Type	Description
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.
scope	string	Specifies scope that should be associated with resulting comment. **Optional.** Defaults to <any>. Allowed values: "document", "paragraph".
paragraph_id	string	Specifies paragraph ID that should be associated with resulting comment. **Optional.** Defaults to <undefined>. Not read unless [scope=paragraph] is specified.
replies_size	integer	Amount of replies to show. **Optional.** Defaults to 15.

4.4.5. Response example

Status: 200 OK

```
{
   "current_page": 1,
   "total_pages": 1,
   "requested_page_size": 50,
   "actual_page_size": 2,
   "results": [
    {
      "scope": "document",
      "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
      "user_email": "john.doe@clausematch.com",
      "user_first_name": "John",
      "user_last_name": "Doe",
      "content": "Fusce sem augue, egestas at auctor sit amet, efficitur sed eros. Cras ac dignissim lectus.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam iaculis, tellus sed molestie ultrices, dolor lacus
feugiat est, et interdum metus dolor at odio.",
      "replies": {
        "requested_size": 50,
        "actual_size": 1,
        "results": [
          {
            "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
            "user_email": "john.doe@clausematch.com",
            "user_first_name": "John",
            "user_last_name": "Doe",
            "content": "Interdum et malesuada fames ac ante ipsum primis in faucibus. In
a augue sed nisi volutpat finibus. Nullam nisl metus, vehicula in arcu ac, vehicula fringilla neque. Sed bibend
um convallis fermentum. Cras sem velit, elementum sed interdum et, maximus vitae justo."
          }
        ]
      }
    },
    {
      "scope": "paragraph",
      "paragraph_id": "1692bc74-bc3b-4a46-8dce-00347b070d4f",
      "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
      "user_email": "john.doe@clausematch.com",
      "user_first_name": "John",
      "user_last_name": "Doe",
      "content": "Cras commodo augue eget sapien interdum finibus. Sed turpis felis, dapibus nec libero et,
condimentum tristique urna. Aliquam bibendum nisl vitae felis pharetra convallis. Etiam in tincidunt ipsum.
Maecenas blandit magna at venenatis suscipit. Pellentesque posuere augue interdum aliquet placerat.",
      "replies": {
        "requested_size": 50,
        "actual_size": 0,
        "results": []
      }
    }
  ]
}
```

5. Policy Portal API

5.1. Get basic Policy Portal info for a document version

5.1.1. Description

Call this endpoint to get basic information about the state of a document version on the Policy Portal.

This endpoint provides information which answers the following questions:

- Is the document version published on the Policy Portal.
- When was it published and who did it.
- Who are the key contacts for this version.
- How many times was this version viewed by the users.

5.1.2. Request URL

/api/v1/documents/{document_id}/versions/{document_version}/policy-portal/info

5.1.3. Request Method

HTTP GET

5.1.4. Request Parameters

None

5.1.5. Response example

Status: 200 OK

```
{
  "id": "05ec9a73-494a-444e-9f11-397532c561ed",
  "latest_published_version": "1.0.13",
  "latest_published_title": "Metrics & Alerting v1",
  "latest_categories": [
    {
      "category_id": "c76168ea-11a2-4dd5-98b5-6706735756a4",
      "category_name": "People Risks"
    }
  ],
  "key_contacts": [
    {
      "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
      "user_email": "john.doe@clausematch.com",
      "user_first_name": "John",
      "user_last_name": "Doe",
    }
  ],
    "created": {
      "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
      "user_email": "john.doe@clausematch.com",
      "user_first_name": "John",
      "user_last_name": "Doe",
      "date": "2019-02-15T08:42:14.123"
    },
    "last_modified": {
      "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
      "user_email": "john.doe@clausematch.com",
      "user_first_name": "John",
      "user_last_name": "Doe",
      "date": "2020-04-30T18:50:24.866"
    }
  }
```

5.2. Get document and paragraph level questions with replies for a document version

5.2.1. Description

After the document was published on the Policy Portal, portal users can leave their questions and comments on the published document for the document owner to answer. Use this endpoint to retrieve these questions.

Each question can be associated with a particular paragraph or be related to the document itself. A question can have multiple answers posted by different users.

5.2.2. Request URL

/api/v1/documents/{document_id}/versions/{document_version}/policy-portal/questions

5.2.3. Request Method

HTTP GET

5.2.4. Request Parameters

Name	Type	Description
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.
scope	string	Specifies scope that should be associated with resulting questions. **Optional.** Defaults to <any>. Allowed values: "document", "paragraph".
paragraph_id	string	Specifies paragraph ID that should be associated with resulting question. **Optional.** Defaults to <undefined>. Not read unless [scope=paragraph] is specified.
replies_size	integer	Amount of replies to show. **Optional.** Defaults to 15.

5.2.5. Response example

```
Status: 200 OK
{
   "current_page": 1,
   "total_pages": 1,
   "requested_page_size": 50,
   "actual_page_size": 2,
   "results": [
      {
         "scope": "document",
         "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
         "user_email": "john.doe@clausematch.com",
         "user_first_name": "John",
         "user_last_name": "Doe",
         "content":
"Fusce sem augue, egestas at auctor sit amet, efficitur sed eros. Cras ac dignissim lectus. Lorem ipsum dolor
 sit amet, consectetur adipiscing elit. Nam iaculis, tellus sed molestie ultrices, dolor lacus feugiat est,
et interdum metus dolor at odio.",
         "replies": {
            "requested_size": 50,
            "actual_size": 1,
            "results": [
               {
                  "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
                  "user_email": "john.doe@clausematch.com",
                  "user_first_name": "John",
                  "user_last_name": "Doe",
                  "content": "Interdum et malesuada fames ac ante ipsumprimis in faucibus. In
a auguesednisivolutpatfinibus. Nullamnislmetus, vehicula in arcuac, vehiculafringillaneque. Sedbibendumco
nvallisfermentum. Crassemvelit, elementumsedinterdum et, maximus vitae justo."
```

```
          }
        ]
      }
    },
    {
      "scope": "paragraph",
      "paragraph_id": "1692bc74-bc3b-4a46-8dce-00347b070d4f",
      "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
      "user_email": "john.doe@clausematch.com",
      "user_first_name": "John",
      "user_last_name": "Doe",
      "content": "Cras commodo augue eget sapien interdum finibus. Sed turpis felis, dapibus nec libero et, condimentum tristique urna. Aliquam bibendum nisl vitae felis pharetra convallis. Etiam in tincidunt ipsum. Maecenas blandit magna at venenatis suscipit.",
      "replies": {
        "requested_size": 50,
        "actual_size": 0,
        "results": []
      }
    }
  ]
}
```

5.3. Get Policy Portal access rights for a document version

5.3.1. Description

Use this endpoint to get information about the users and groups who have access to the published document. Multiple groups of users and individual users can have access to the same document at the same time.

The special "Access to all" option denotes document versions which can be viewed by all users who have access to the Policy Portal.

5.3.2. Request URL

/api/v1/documents/{document_id}/versions/{document_version}/policy-portal/access-rights

5.3.3. Request Method

HTTP GET

5.3.4. Request Parameters

Name	Type	Description
page_number	integer	Results page number to show. **Optional.** Defaults to 1.
page_size	integer	Amount of results on page. **Optional.** Defaults to 15.

5.3.5. Response example

```
Status: 200 OK
{
   "current_page": 1,
   "total_pages": 1,
   "requested_page_size": 50,
   "actual_page_size": 2,
   "results": [
      {
         "type": "user",
         "user_id": "41d8dc0c-fc18-472b-8936-c06a02c7f879",
         "user_email": "john.doe@clausematch.com",
         "user_first_name": "John",
         "user_last_name": "Doe"
      },
      {
         "type": "group",
         "group_id": "624507d7-2dd5-442d-8dd8-6090a6c7eb72",
         "group_name": "Security Department"
      }
   ]
}
```