

# Customer-Documentation: Genios API

- [Introduction](#)
- [Detailed Technical Documentation](#)
- [JSON and XML Format](#)
- [Permissions](#)
- [Login-Service / Authentication](#)
  - [Example Login Request](#)
  - [Example Login Response](#)
- [Search-Service](#)
  - [Example Search Requests](#)
  - [Example Search Response](#)
- [Document Service](#)
  - [Example Document Requests](#)
  - [Example Document Response](#)
- [Attachment Service](#)
  - [Example Request for Thumbnails](#)

## Introduction

The external Genios API offers searching and document retrieval capabilities for all Genios databases. The exchange format is computer readable by supporting the **REST** standard including the **JSON** and **XML** document-format.

In summary the API support the following services:

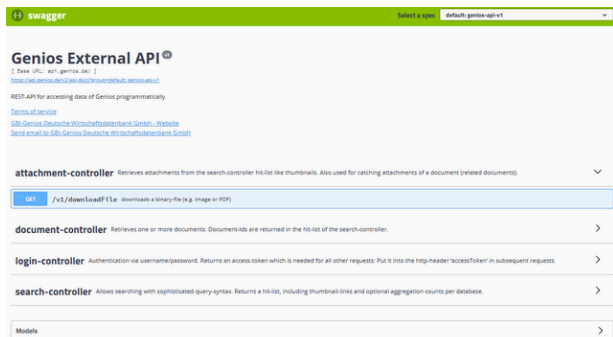
- **Login:** Needs to be used for authentication. Before any other requests can be done, you need to obtain an *accessToken* by providing a valid username and password.
- **Search:** Searches for documents and returns a hit-list. The same sophisticated query-syntax as at *genios.de* is support (Check: Help-Page or Swagger documentation). Optionally an aggregation for hit-count per database is supported.
- **Document:** Retrieves one or more documents by providing document-ids from the hit-list.
- **Attachment:** Allows to read related binary documents. For example: Thumbnail pictures of hit-list-entries; related PDF documents of a genios-document.
- (**SourceInformation:** Reads meta-data about available database and their content. ) *Not available yet*

Please note: For every service you need special permissions. Please double-check with your Genios contact-person if you are allowed to use a specific service.

## Detailed Technical Documentation

For implementation purpose the Genios API is well explained with the REST API documention tool *Swagger*. It can be found at:

<https://api.genios.de/>



The API offers a **versioning** (you see it in the upper right in the Swagger-UI). Please use allows the latest version!

The following chapters will show some example requests. That should be helpful for a quick start. Details can always be looked up in Swagger.

## JSON and XML Format

For every **response** you can decide if it should return JSON or XML format. Use the HTTP header **Accept** to specify **application/xml** or **application/json**

The default would be JSON if you are no specifying anything.

Example JSON response

```
{
  "accessToken": null,
  "success": false,
  "errorMessage": "Wrong credentials. Username: null Path:uri=/v1/login "
```

```
}
```

#### Example XML reponse

```
<loginResponse>
  <accessToken/>
  <success>false</success>
  <errorMessage>Wrong credentials. Username: null Path:uri=/v1/login </errorMessage>
</loginResponse>
```

All following examples will be shown in JSON.

Please note: If you need to send structured data in your **requests** you have also the option to use XML or JSON. You need to set the **HTTP header Content-Type** accordingly. Please check Swagger to see which function support XML/JSON as request-data.

## Permissions

Every API user needs to be assigned some permissions. Genios sets those in the User-Management Backoffice application. The following permissions are provided:

Type	Description
SEARCH	User could do searches via API (and retrieve attachments like thumbnails of results)
DOCUMENT	User could retrieve documents and attachments
LIMIT_SEARCH_REQUEST_5	Allows just 5 searches per day (for testing)
LIMIT_SEARCH_REQUEST_10000	Allows 10.000 searches per day
LIMIT_SEARCH_REQUEST_20000	Allows 20.000 searches per day
COUNTER_5_SUSHI	Is allowed to use the Counter5 Sushi API
FULLTEXT	not implemented yet
ATTACHMENT	not implemented yet (see SEARCH)

## Login-Service / Authentication

Before any other requests can be done, **you need to obtain an accessToken** by providing a valid username and password.

The `accessToken` is just **valid for a short period of time** (currently it's one hour). Whenever you get an unauthorized-error (401 response-code) you need to login again. Sure, you can login as often as you like; meaning you don't need to wait until you get the error.

In subsequent requests (like search or document-retrieval) put the `accessToken` into an **HTTP-header field with the name `accessToken`**.

#### Example Login Request

```
curl -X POST -H 'Content-Type: application/x-www-form-urlencoded' -i 'https://api.genios.de/v1/login' --data 'usern
```

#### Example Login Response

```
{
  "accessToken": "123accessTokenSomething...long-string",
  "success": true,
  "errorMessage": null
}
```

## Search-Service

Searches for documents and returns a hit-list. The same sophisticated query-syntax as at [genios.de](https://api.genios.de) is support (Check: Help-Page or Swagger documentation). Optionally an aggregation for hit-count per database is supported.

Supported **parameters**:

- offset → integer (min=0, max=1.000) → the amount of results that is left out of the resultlist starting from the first result (offset=100 → show documents from result 101)
- size → integer (min=1, max=50) → the amount of results that are shown within a resultlist
- sort → char (BY\_RELEVANCE | BY\_DATE)
  - BY\_DATE → the resultlist will be sorted via publishing date starting from the most recent
  - BY\_RELEVANCE → the resultlist will be sorted via using the GBI-Genios relevance measure (TF-IDF + internal boosting criteria) starting with the most relevant one

In terms of sizing there are some **limitation**. In normal use that should not affect you. But be aware that currently:

- Only the first 10.000 hits can be retrieved via the offset-parameter

2. The maximum size of boolean terms in the search-query is restricted to 1024 clauses
3. The offset can't exceed 1.000.
4. The size can't exceed 50.

## Example Search Requests

Using HTTP Params:

```
curl -X GET -k -H 'accessToken: 123accessTokenSomehting' -i 'https://api.acc.genios.de/v1/search?searchText=TITEL:b
```

Using JSON body (XML would be also possible):

```
curl -X POST -k -H 'Content-Type: application/json' -H 'Accept: application/json' -H 'accessToken :123accessTokenSc
  "searchText": "corona AND bayern",
  "offset": 10,
  "size": 10,
  "countPerDatabase": true,
  "sort": "BY_RELEVANCE"
}
```

## Example Search Response

```
{
  "searchText": "TITEL:biden",
  "offset": 0,
  "totalSize": 22957,
  "sort": "BY_DATE",
  "aggregations": null,
  "hits": [{
    "docId": "NN_e6eb0457137e67c2372f6136dac744f7f31171f4",
    "wordCount": 34,
    "title": "Joe Biden",
    "subTitle": null,
    "textAbstract": "Warum ist der weiße Ex-Vizepräsident Joe Biden bei den Schwarzen in den USA so beliebt? Eine R
    "category": null,
    "timestamp": "2020-03-20T23:59:59.999Z",
    "source": "Nürnberger Nachrichten",
    "database": "NN",
    "geniosWebDocumentUrl": "https://jwww.genios.de/document/NN_e6eb0457137e67c2372f6136dac744f7f31171f4",
    "priceNetInCents": 200,
    "vat": 5,
    "priceGrossInCents": 210,
    "thumbnailData": null,
    "attachments": []
  }], .....
```

## Document Service

Retrieves one or more documents by providing document-ids from the hit-list.

In terms of sizing there are some **limitation**. In normal use that should not affect you. But be aware that currently:

1. The amount of parallel document retrievs is limited to 50 per call. If you need more, you may iterate using only up to 50 documents.

## Example Document Requests

Using HTTP Params:

```
curl -X GET -k -H 'accessToken: 123accessTokenSomething' -i 'https://api.genios.de/v1/documents?docIds=0AZ__0253dea
```

Using JSON body (XML would be also possible; see second example):

```
curl -X POST -H 'Content-Type: application/json' -H 'accessToken: 123accessTokenSomething' -i http://localhost:8080
  "docIds": [ "0AZ__0253deab214e0374cf15596258f3a667a9eb170a", "NN_e6eb0457137e67c2372f6136dac744f7f31171f4" ]
}'
```

```
<?xml version="1.0" encoding="UTF-8"?>
<DocumentRequestUser>
  <docIds>
    <value>WESP__ed8b1fc51cb11c78291173e130490a8948716843</value>
    <value>WRU__579ee1183da2b9115444dd2600d9fe1c16c662ec</value>
  </docIds>
</DocumentRequestUser>
```

#### Example Document Response

```
{
  "items": [{
    "metadata": {
      "source": {
        "sourceId": "OAZ",
        "sourceName": "Oschatzer Allgemeine Zeitung",
        "volume": null,
        "issueNumber": null,
        "date": "2019-01-22",
        "year": 2019,
        "issueTitle": null,
        "issn": {
          "issnPrint": null
        }
      }
    },
    "article": {
      "pageStart": "21",
      "pageEnd": null,
      "articleType": null,
      "pricingOption": null,
      "regionalSection": null,
      "juridicalReference": null
    },
    "authors": [{
      "authorLine": "Petra Philippsen"
    }],
    "itemId": "doc73qn60oe888c2axsfda",
    "techId": "OAZ__0253deab214e0374cf15596258f3a667a9eb170a"
  },
  "content": {
    "titleList": {
      "category": "Sport",
      "headline": null,
      "title": "Totalschaden bei Zverev",
      "subTitle": "„Das musste raus“: Tennisstar lässt seinen Frust über das Aus bei den Australian Open an seine Fans hören",
      "subCat": null,
      "subSubCat": null
    },
    "lead": null,
    "abstractEnglish": null,
    "text": "<paragraph>Fünf. Sechs. Sieben. Und noch ein achttes Mal hackte Alexander Zverev seinen Tennisschläger über dem Netz. Der russische Nummer eins war bei seinem Auftaktmatch gegen den Australier Nick Pietrangeli in der ersten Runde der Australian Open in Melbourne am Samstag gegen den 27-jährigen Amerikaner mit 1:2 Sätzen gescheitert. Zverev hatte sich im ersten Satz mit 6:4 durchgesetzt, verlor aber im zweiten Satz mit 6:1 und im dritten Satz mit 6:4. In der zweiten Runde wird er gegen den 27-jährigen Australier Nick Pietrangeli antreten. Zverev ist der einzige Spieler, der in der ersten Runde der Australian Open in Melbourne am Samstag gegen den 27-jährigen Amerikaner mit 1:2 Sätzen gescheitert ist. Zverev hatte sich im ersten Satz mit 6:4 durchgesetzt, verlor aber im zweiten Satz mit 6:1 und im dritten Satz mit 6:4. In der zweiten Runde wird er gegen den 27-jährigen Australier Nick Pietrangeli antreten. Zverev ist der einzige Spieler, der in der ersten Runde der Australian Open in Melbourne am Samstag gegen den 27-jährigen Amerikaner mit 1:2 Sätzen gescheitert ist."
    "textRaw": null,
    "abstract": null
  }
},
}
```

If you need more information about the structure of those documents please [contact Genios](#).

#### Attachment Service

Allows to read related binary documents. For example: Thumbnail pictures of hit-list-entries; related PDF documents of a genios-document.

#### Example Request for Thumbnails

In the search-response there is the section "thumbnailData". If set you will find a attribute called "attachmentId". Take this ID for the following request:

```
curl -X GET -H 'accessToken: 123accessTokenSomething' -i 'https://api.genios.de/v1/downloadFile?attachmentId=SEfFX2'
```

The **response** will be a binary stream. In the thumbnail case above a PNG image is returned.