API definition for Private GPT 1.4

> ℹ️ The Private GPT software is not designed and not intended for any high-risk use as specified in the EU Artificial Intelligence Act (Art. 6 and Annex III). As such the Private GPT software may only be used, where it does not pose a significant risk of harm to the health, safety or fundamental rights of natural persons, including by not materially influencing the outcome of decision making. It is in the sole responsibility of the user to assess and handle such risk potentially arising from the user's specific use of the Private # GPT software.

Publication date	27. März 2025
Released in version	1.4
Version compatibility	`1.3` `1.4`

This document defines the endpoints for the PrivateGPT API version 1.
All data in bodies and responses are formatted in JSON.

Changes

➕ Delete an existing chat

➕ Delete all existing chats from the authorized user

➕ Added sources to conversations

➕ Reactivate deleted user

Fixes

Security Fix: Enforcement of Permissions on API Resources

Previous Behavior:

Users without the necessary permissions were able to view, modify, and delete resources such as documents, groups, and users via the API. It was even possible to grant a user all permissions through API calls, regardless of their actual access rights.

Current Behavior:

Access control is now correctly enforced on all relevant API endpoints. Only users with the appropriate permissions can:

- View, update, or delete documents, groups, or user accounts
- Assign or modify permissions via the API

Unauthorized access attempts will now be blocked with appropriate error responses.

Correct Document Status Reporting

The endpoint

```
GET {base_url}/api/v1/sources/{sourceId}
```

now correctly returns the actual status of a document.

Previously, the `status` field always showed `"vectorized"` even if the document was in an error state. This has been fixed to reflect the true processing state (e.g., `"error"`, `"processing"`, etc.).

Contents

Login

Register for API

Get an access token for authentication in other API calls.

The user requesting a token needs to be existing beforehand within PrivateGPT.

Endpoint	{base-url}/api/v1/login
Method	POST
Headers	
Accept	application/json
Authentication	No Auth

Body

```
1  {
2      "email":"{user-email}",
3      "password":"{user-password}"
4  }
```

The `{user-email}` and `{user-password}` correspond to the credentials of an existing user in PrivateGPT.

Response

```
1  {
2      "data": {
3          "token": "1|h2Y8cWpKwsGI8iaKl8WUo4bz2PusBR4M6wmgvIDe1d98dac8"
4      },
5      "message": "success",
6      "status": 200
7  }
```

This `token` will be used in the following calls as {api-token}. It has no expiration date.

Logout

Invalidate the API-Token

Endpoint	{base-url}/api/v1/logout
Method	DELETE
Headers	
Accept	application/json
Authentication	Bearer {api-token}

Response:

```
1  {
2      "data": {},
3      "message": "success",
4      "status": 200
5  }
```

Chats (Conversations)

New chat

Start a new chat. To use the LLMs 'document knowledge' (RAG), one or more available groups must be provided, or usePublic needs to be true. If no groups are provided (or the list is empty) and usePublic is false the LLMs 'general knowledge' will be used instead.

ℹ️ usePublic means public documents that can be used in the chat by all users.

Please find the available languages below under **List of options**.

Endpoint	{base_url}/api/v1/chats
Method	POST
Headers	
Accept	application/json

Authentication	Bearer {api-token}

Body

```
1  {
2      "language": "en",
3      "question": "Hello World?",
4      "usePublic": true,
5      "groups": [] // optional
6  }
```

ℹ If "usePublic" is set to false **and** no groups are provided, RAG will be deactivated for this chat.

Response

```
1  {
2      "data": {
3          "chatId": "9d29bfb6-m763-92ne-4f10-c9c28e4d94fa",
4          "answer": "Hello! How can I assist you today?",
5          "sources": []
6      },
7      "message": "success",
8      "status": 200,
9      "notice": "PrivateGPT provides automated responses and can make mistakes. Verify critical
10 }
```

This `chatId` will be used in the continuing of a chat as {chatId}.

ℹ **Note on Document Results in Chats (since v1.4)**

Starting with version 1.4, chats that include document results will always return a page number of `0` . This is due to the introduction of a new context-finding service, which currently does not provide specific page numbers.

In the frontend, the absence of a page number is handled by simply omitting its display. However, in the API response, the `page` parameter remains present in the result object for consistency, but its value will always be `0` .

Continue an existing chat

Continue an already started chat.

Endpoint	{base_url}/api/v1/chats/{chatId}
Method	PATCH
Headers	
Accept	application/json
Authentication	Bearer {api-token}

Body

```
1  {
2      "question": "Hello World?"
3  }
```

Response

```
1  {
2      "data": {
3          "chatId": "9d29bfb6-m763-92ne-4f10-c9c28e4d94fa",
4          "answer": "Hello! How can I assist you today?",
5          "sources": []
6      },
7      "message": "success",
8      "status": 200,
9      "notice": "PrivateGPT provides automated responses and can make mistakes. Verify critical
10 }
```

Get Information about an existing Chat

Endpoint	{base_url}/api/v1/chats/{chatId}
Method	GET
Headers	
Accept	application/json
Authentication	Bearer {api-token}

Response

```
1  {
2
3      "data": {
```

```
 4            "chatId": "chat-uuid",
 5            "title": "My Chat",
 6            "language": "en",
 7            "groups": ["Group A", "Internals"],
 8            "messages": [
 9                {
10                    "question": "What is PrivateGPT?",
11                    "answer": "PrivateGPT is mindblowing!",
12                    "language": "en"
13                }
14            ]
15        },
16        "message": "success",
17        "status": 200
18  }
```

> ⓘ **Limitations on Accessing Single Documents Chats and Sources via API**
>
> Documents uploaded during a chat session (new single documents chat feature) are not yet accessible via the API.
>
> Chats that involve uploaded documents directly in the chat are not available through the API (`/chats` endpoint).
>
> Documents associated with single documents chats are also not accessible via the API (`/sources` endpoint).

Delete an existing Chat

Endpoint	{base-url}/api/v1/chats/{chatId}
Method	DELETE
Headers	
Accept	application/json
Authentication	Bearer {api-token}

Response:

```
1  {
2      "data": {},
3      "message": "success",
4      "status": 200
5  }
```

Delete all existing Chats from the authorized User

Endpoint	{base-url}/api/v1/chats/flush
Method	DELETE
Headers	
Accept	application/json
Authentication	Bearer {api-token}

Response:

```
1  {
2      "data": {},
3      "message": "success",
4      "status": 200
5  }
```

Sources

Store a new Source via Markdown

> ℹ Limitation: It is currently not possible to upload sources other than Markdown files via API.

The user needs to have the group added to their user profile if they want to upload to a certain group. This is different from the WebUI behavior, where admins can upload to any groups. To upload via API, the groups have to be connected to the user.

Endpoint	{base_url}/api/v1/sources
Method	POST
Headers	
Accept	application/json
Authentication	Bearer {api-token}

Body

```
1  {
2      "name": "My new Source",
3      "groups": ["Group A"] // optional, can be omitted/empty if source should be public
4      "content": "This is my content in Markdown format"
5  }
```

Response

```
1  {
2      "data": {
3          "documentId": "9d29bfb6-m763-92ne-4f10-c9c28e4d94fa"
4      },
5      "message": "success",
6      "status": 200
7  }
```

Get information about an existing source

Endpoint	{base_url}/api/v1/sources/{sourceId}
Method	GET
Headers	
Accept	application/json
Authentication	Bearer {api-token}

Response

```
1   {
2       "data": {
3           "sourceId": "446f198f-9d45-423b-95d1-624c4cdbfd1c",
4           "title": "Another amazing source",
5           "groups": ["Internals"],
6           "isPublic": false,
7           "state": "vectorized",
8           "creator": {
9               "email": "roy@acme.com",
10              "name": "Roy Trenneman"
11          },
12          "createdAt": "2024-10-01T13:37:42Z",
13          "updatedAt": "2024-10-26T09:11:13Z",
14      },
15      "message": "success",
16      "status": 200
17  }
```

Edit an existing Source

Endpoint	{base_url}/api/v1/sources/{sourceId}
Method	PATCH
Headers	
Accept	application/json
Authentication	Bearer {api-token}

Body

```
1  {
2      "name": "My amazing source",
3      "groups": ["Group A"], // optional, list may be empty if source should be public
4      "content": "This is my content in Markdown format"
5  }
```

📋 All Fields are optional and remain unchanged if not provided.

Response

```
1  {
2      "data": {
3          "documentId": "9d29bfb6-m763-92ne-4f10-c9c28e4d94fa"
4      },
5      "message": "success",
6      "status": 200
7  }
```

Get all sources for the provided group

Endpoint	{base_url}/api/v1/sources/groups
Method	POST
Headers	
Accept	application/json
Authentication	Bearer {api-token}

Body

```
1  {
2      "groupName": "Group A"
3  }
```

Response

```json
1  {
2      "data": {
3          "sources": [
4                  "9d29bfb6-m763-92ne-4f10-c9c28e4d94fa",
5                  "fe8a1336-510b-4baf-8d2b-47f20c33e7d6",
6                  "bcd69bc4-e3fc-4771-8526-e7d894574f13",
7                  "c2df51f3-e4fe-4311-8b27-c37df49e047c",
8                  "5fa13006-4aa9-401f-899c-c2aa995fb7a6"
9              ]
10         },
11     "message": "success",
12     "status": 200
13 }
```

Delete an existing source

Endpoint	{base_url}/api/v1/sources/{sourceId}
Method	DELETE
Headers	
Accept	application/json
Authentication	Bearer {api-token}

Response

```json
1  {
2      "data": [],
3      "message": "success",
4      "status": 200
5  }
```

Groups

Available groups

Get all available groups.

personalGroups: all groups the current user can access. Usable for chat.

assignableGroups: all available groups. Usable for creating and editing a user and sources.

Endpoint	{base_url}/api/v1/groups
Method	GET
Headers	
Accept	application/json
Authentication	Bearer {api-token}

Response:

```
1  {
2      "data": {
3          "personalGroups": ["alpha", "beta"],
4          "assignableGroups": ["alpha", "beta", "internal", "finance"]
5      },
6      "message": "success",
7      "status": 200
8  }
```

Store a new group

Endpoint	{base_url}/api/v1/groups
Method	POST
Headers	
Accept	application/json
Authentication	Bearer {api-token}

Body

```
1  {
2      "groupName": "Group A"
3  }
```

Response:

```
1  {
2      "data": {},
3      "message": "success",
```

```
4        "status": 200
5    }
```

Delete an existing group

Endpoint	{base_url}/api/v1/groups
Method	DELETE
Headers	
Accept	application/json
Authentication	Bearer {api-token}

Body

```
1    {
2        "groupName": "Group A"
3    }
```

Response:

```
1    {
2        "data": {},
3        "message": "success",
4        "status": 200
5    }
```

Users

Store a new User

The identification of a user is in relation to the email address. Please find the different available roles below under <u>List of options</u>.

Endpoint	{base_url}/api/v1/users
Method	POST

Headers	
Accept	application/json
Authentication	Bearer {api-token}

Body

```json
{
    "name": "John Doe",
    "email": "john.doe@pgpt.local",
    "language": "de", // optional - defaults to "en"
    "timezone": "UTC", // optional - defaults to "Europe/Berlin"
    "password": "SuperSecret42!",
    "usePublic": true,
    "groups": ["Group A"],
    "roles": ["system"],
    "activateFtp": true,
    "ftpPassword": "myFTP-Password1337"
}
```

Response

```json
{
    "data": {},
    "message": "success",
    "status": 200
}
```

Edit an existing user

Endpoint	{base_url}/api/v1/users
Method	PATCH
Headers	
Accept	application/json
Authentication	Bearer {api-token}

Body

```json
{
    "email": "roy@acme.com", // E-Mail Field is required and must exist
    "name": "Roy Trenneman",
    "language": "de",
```

```
 5        "timezone": "UTC",
 6        "password": "SuperSecret42!",
 7        "publicUpload": true,
 8        "groups": ["Group A"],
 9        "roles": ["system"],
10        "activateFtp": true,
11        "ftpPassword": "myFTP-Password1337"
12  }
```

> 🗒 All other fields than e-mail are optional and remain unchanged if not provided.

Response

```
1  {
2      "data": {},
3      "message": "success",
4      "status": 200
5  }
```

Delete an existing user

Endpoint	{base_url}/api/v1/users
Method	DELETE
Headers	
Accept	application/json
Authentication	Bearer {api-token}

Body

```
1  {
2      "email": "roy@acme.com"
3  }
```

Response

```
1  {
2      "data": {},
3      "message": "success",
4      "status": 200
5  }
```

Reactivate an existing user

Endpoint	{base_url}/api/v1/users/reactivate
Method	POST
Headers	
Accept	application/json
Authentication	Bearer {api-token}

Body

```
1  {
2      "email": "roy@acme.com"
3  }
```

Response

```
1  {
2      "data": {},
3      "message": "success",
4      "status": 200
5  }
```

List of options

Languages

The languages directly relate to those available in the chat of the web application.

Parameter	Language
en	English
de	Deutsch
it	Italiano
fr	Français

es	Español
ar	العربية
bg	Български
cs	Čeština
eo	Esperanto
et	Eesti
el	Ελληνικά
fi	Suomi
fa	فارسی
he	עברית
hi	हिन्दी
hu	Magyar
id	Bahasa Indonesia
ja	日本語
lt	Lietuvių
lv	Latviešu
nl	Nederlands
no	Norsk
pl	Polski
pt	Português
pt-br	Português (Brasil)
ro	Română
ru	Русский
sk	Slovenčina
sv	Svenska

tr	Türkçe
uk	Українська
zh	中文

Roles

The roles are directly relating to those of the web application.

Parameter	Function
ad	Active Directory Settings
analytics	Analytics
documents	Manage sources & upload files
confluence-documents	Import data from confluence
smtp	SMTP settings
system	System logs and certificates
users	User administration