

API v1 Pre-Release-Definition for PrivateGPT 1.3

Please note: This definition might be subject to change.

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	Bearer {api-token}

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}.

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.

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       "a14e3c4-dfb4-4088-b379-362a3772d3bf",
```

```

7         "fdb75197-aab6-4c10-bd73-1ed1475e00cb"
8     ]
9 },
10    "message": "success",
11    "status": 200
12 }

```

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

Continue an existing chat [↗](#)

Continue an already started chat.

Endpoint	<code>{base_url}/api/v1/chats/{chatId}</code>
Method	POST
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             "aa14e3c4-dfb4-4088-b379-362a3772d3bf",
7             "fdb75197-aab6-4c10-bd73-1ed1475e00cb"
8         ]
9     },
10    "message": "success",
11    "status": 200
12 }

```

Get Informations about an existing Chat [↗](#)

Endpoint	<code>{base_url}/api/v1/chats/{chatId}</code>
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                 "sources": [
14                     "aa14e3c4-dfb4-4088-b379-362a3772d3bf",
15                     "fdb75197-aab6-4c10-bd73-1ed1475e00cb"
16                 ]
17             }
18         ]
19     },
20     "message": "success",
21     "status": 200
22 }
```

Sources [↗](#)

Store a new Source via Markdown [↗](#)

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

Body

```

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

Get informations 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   "title": "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   "message": "success",
4   "status": 200
5 }

```

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

```

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

```

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 [↗](#)

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

Body

```
1 {
2   "name": "Roy Trenneman",
3   "email": "roy@acme.com",
4   "language": "de", // optional - defaults to "en"
5   "timezone": "UTC", // optional - defaults to "Europe/Berlin"
6   "password": "SuperSecret42!",
7   "usePublic": true,
8   "groups": ["Group A", "Group C"],
9   "roles": ["Sources", "system"],
10  "activateFtp": true,
11  "ftpPassword": "myFTP-Password1337"
12 }
```

Response

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

Edit an existing user [↗](#)


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

Body

```

1 {
2   "email": "roy@acme.com", // E-Mail Field is required and must exist
3   "name": "Roy Trenneman",
4   "language": "de",
5   "timezone": "UTC",
6   "password": "SuperSecret42!",
7   "publicUpload": true,
8   "groups": ["Group A", "Group C"],
9   "roles": ["Sources", "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 }

```

