Central API Access to LLMs at KIT via the KI-Toolbox

Version: 09.10.2026

What is an API – and when do I need one?

An API (Application Programming Interface) is an interface through which programs communicate automatically with a service. Instead of chatting in the browser, your applications, scripts or tools address the AI models directly.

You normally need an API if you want to automate or integrate AI – for example:

- Integrate AI functions into your own applications or workflows,
- Automate recurring tasks (e.g. evaluating many documents or generating texts in batches),
- Connect external tools such as coding assistants (e.g. Cline) to the AI models.

If you only want to use AI models interactively in the chat, you can use the KI-Toolbox web interface – no API is required for this.

Use modern AI language models easily and securely via a central interface (API). This solution is aimed at all KIT members and facilitates the integration of Large Language Models (LLMs) into your own applications and workflows.

This service is provided by the Scientific Computing Center (SCC) as part of the "GenAI@KIT" strategy project.

Your advantages at a glance:

- **No need for a separate account with providers:** Log in with your KIT account and use a self-generated API key. Separate registrations, e.g. with OpenAI or Microsoft, are not required.
- **Central token procurement:** KIT purchases the necessary usage quotas and makes them available to everyone. In this way, you benefit from better conditions.
- **Improved data protection and security standards:** Thanks to framework agreements, e.g. with Microsoft for the Azure Cloud, KIT has stricter requirements for handling your data than for individual use.

Billing and usage information (fair use principle):

Billing is carried out centrally by KIT. The service is designed to support your daily work and research in an uncomplicated way. You are welcome to use it to the usual extent.

Budgeting: Costs incurred for cloud model tokens are budgeted. There is a limited monthly free budget. You can increase your personal budget at https://my.litellm.scc.kit.edu and also monitor your own usage there. Local models are currently not budgeted.

However, if you are planning larger, automated projects – e.g. processing complete research datasets with many thousands of API calls – please contact the responsible team in advance via the service address ki-toolbox@scc.kit.edu. This allows us to plan resources together and ensure a smooth process for all users.

Special case: Service development and future service accounts

The current resources are intended for individual, official use. Therefore, an agreement is mandatory. Conformity with the EU AI Act and other legal requirements must also be clarified on your own responsibility, e.g. obligations arising from the provider and operator role.

If you plan to develop an automated service, bot, or application that uses the API on a permanent basis, you will need to set up a dedicated service account for it. Please have this service account created by your ITB (IT representative) for each service. The service account must then be activated for use by sending an e-mail request (ki-toolbox@scc.kit.edu) to the KI-Toolbox team.

This procedure ensures a clear separation between personal use and business use.

System overview: How does the service work?

The system consists of two main components:

1. The web interface:
- Available at: https://ki-toolbox.scc.kit.edu/
- Provides an interactive chat environment comparable to that of ChatGPT.
- You can also generate your personal API key here.
2. The API:
- An OpenAI-compatible and a LiteLLM-compatible API endpoint are provided (KI-Toolbox API and LiteLLM API).
- You can find the corresponding base endpoints (Base URLs) of both access routes in the section Two Access Routes.

Through this service, you have access to a selection of different models:

- **Local models**: These are operated entirely on the infrastructure of the SCC at KIT.
- **Cloud models**: Models from external providers such as OpenAI, e.g. provided via the Microsoft Azure cloud at a German location.
- Please **do not** use the two models standard-external and standard-local via the API. The two models have a customized system prompt that can change your own system prompt when used via the API and thus lead to unwanted results.

Two Access Routes: Which API for what?

The models can be used via two central access routes. Both routes are based on the same models but differ in key management and scope of functions:

- **KI-Toolbox API** (OpenAI-compatible): You generate the API key in the KI-Toolbox. This route also gives you access to the KI-Toolbox functions such as the knowledge base (RAG).
- **LiteLLM API**: You manage API keys, budget and validity yourself in the MyLiteLLM portal (https://my.litellm.scc.kit.edu). This route offers pure LLM access only – without RAG, web search and other KI-Toolbox functions – but is more direct and slightly faster.

When to use which access route?

	KI-Toolbox API	LiteLLM API
Endpoint (Base URL)	https://ki-toolbox.scc.kit.edu/api/v1	https://litellm.scc.kit.edu
API key	Generated in the KI-Toolbox (Settings > account)	Created in the MyLiteLLM portal: https://my.litellm.scc.kit.edu/apikeys
Scope of functions	LLM access incl. KI-Toolbox functions (e.g. knowledge base/RAG)	Pure LLM access (LiteLLM protocol)
Key and cost control	One key per person; use via the web interface and via API key is not reported separately in the budgeting	Additional API keys possible – each with its own budget, configurable validity and separate usage statistics
Typical use	Your own applications and scripts that also use KI-Toolbox functions	External tools with LiteLLM connection (e.g. Cline) as well as multiple keys with separate budgets

Budgeting and usage control: Budgeting runs centrally for both access routes via the MyLiteLLM portal – it covers both the use of the KI-Toolbox (web interface and API key) and the use via the LiteLLM API. The access routes differ in usage tracking:

- **KI-Toolbox API:** One key per person. Use of the web interface and use via API key are not differentiated in the budgeting – usage and budget apply jointly per person.
- **LiteLLM API:** You can create additional API keys here, e.g. one per tool or project. Each key has its own budget and separate usage statistics.

Rule of thumb: Start with the KI-Toolbox API – it covers the full scope of functions. Use the LiteLLM API if you need several keys with their own budgets or validity periods, or if you want to connect a tool directly via LiteLLM (see Using the LiteLLM API).

Model choice in both access routes: Both APIs offer model aliases that can be used instead of the model ID. We recommend going in this direction (see Model Aliases).

Information Security, Data Protection and Copyright

The responsible handling of data is of paramount importance when using AI models.

Your responsibility: Copyright and licenses: If you upload documents to the system (e.g. for RAG applications), it is your personal responsibility to ensure that you have the necessary rights of use. Check the license of the document in question and make sure that processing in this context is permitted.

The data flow: Where is my data and when?
The system has a vector database for retrieval-augmented generation (RAG) hosted locally at the SCC. This knowledge base is only available when using the KI-Toolbox (web interface and KI-Toolbox API) – the LiteLLM API does not provide access to it (see Two Access Routes).

- **Data storage (RAG):** When you upload documents, they are processed and stored in the local vector database. In this state, your data does not leave KIT.
- **Data processing (request):** When you make a request, your prompt and the relevant data snippets found by the RAG system are sent to the selected model.
- **IMPORTANT**: If you select a cloud model (e.g. *alias.simple-gpt*), this combined data is transferred to the external provider (Microsoft Azure) at the time of the query. In the case of local models, the data remains within KIT.

Three-step model for classifying data (guidance)
Please strictly adhere to the following classification:

- **Level 1: Public data**
  - **Description:** Data that is publicly available without restrictions.
  - **Classification:** For this type of data, all available models (local and cloud) can usually be used.
- **Level 2: Confidential but semi-public data**
  - **Description**: Data that is not intended for the general public, but whose disclosure does not pose a major risk (e.g. KIT intranet content).
  - **Classification**: Depending on the case, the OpenAI models in the Azure Cloud can be used for this purpose. Contractual regulations often provide an adequate level of protection here. These are classified under Azure in Open WebUI and follow the naming scheme *azure.[…]*.
- **Level 3: Highly confidential or secret data**

- **Description**: Data the disclosure of which could cause significant harm (sensitive research data, personal data with a high need for protection).
- **Classification**: Only the local models may be used for this data. These are classified under KIT in Open WebUI and follow the naming scheme *kit.[…]*.

Responsible use of resources:

Use smaller, cheaper models (e.g. alias.simple-local) first and only use the more powerful and expensive models when necessary.

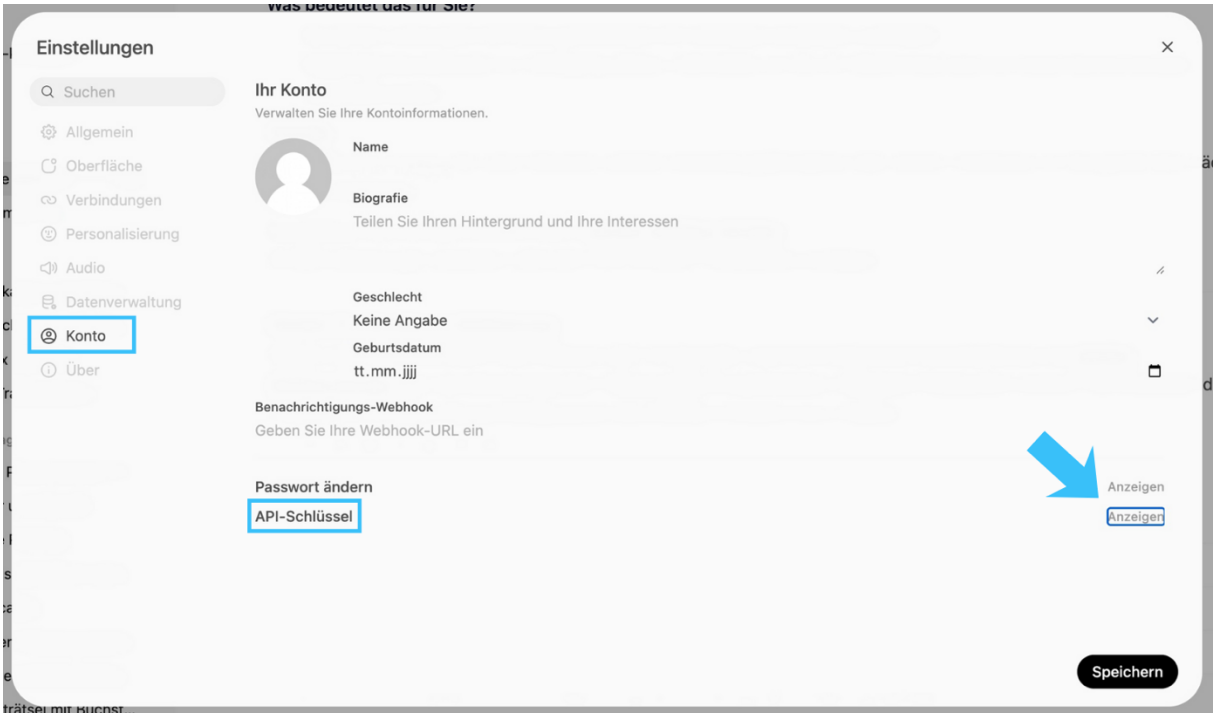
Instructions: Using the KI-Toolbox API (Chat Completion)

Interactive API Documentation for Reference
The most comprehensive and always up-to-date reference for all endpoints and parameters can be found in the interactive API documentation (Swagger UI). There you can also test requests directly.

- Available at: https://ki-toolbox.scc.kit.edu/docs

Step 1: Generate API Key

1. Log in to https://ki-toolbox.scc.kit.edu/.
2. Use the gear icon to navigate to **Settings > account**.
3. Click Create API Key and copy the key. Treat it like a password!



An image containing screenshot, software, text, computer icon. AI-generated content may be incorrect.

Step 2: Use the OpenAI-compatible API endpoint
The base endpoint is https://ki-toolbox.scc.kit.edu/api/v1. The endpoint for chat requests is: https://ki-toolbox.scc.kit.edu/api/v1/chat/completions

Step 3: Choose the right model
Preferably use a model alias (e.g. alias.medium-local or alias.simple-gpt) – see Model Aliases. Alternatively, you can address the concrete model ID; the current list of model IDs can be found at https://www.scc.kit.edu/dienste/ki-toolbox.php.

Step 4: Make an API request (examples)

curl example:

```
curl -X POST https://ki-toolbox.scc.kit.edu/api/v1/chat/completions
\
     -H "Authorization: Bearer YOUR_PERSONAL_API_KEY" \
     -H "Content-Type: application/json" \
     -d '{
          "model": "alias.medium-local",
          "messages": [
              {
                  "role": "system",
                  "content": "You are a helpful assistant at KIT."
              },
              {
                  "role": "user",
                  "content": "Explain the principle of Rayleigh scatt
ering in three sentences."
              }
          ]
     }'
```

Python example with the openai library:
(Prerequisite: pip install openai)

```
from openai import OpenAI

# Configure API key and endpoint
api_key = "YOUR_PERSONAL_API_KEY"
base_url = "https://ki-toolbox.scc.kit.edu/api/v1"

# Initialize Client
client = OpenAI(api_key=api_key, base_url=base_url)

try:
    # Send chat request
    chat_completion = client.chat.completions.create(
        model="alias.medium-local",  # Select the appropriate model
here
        messages=[
            {"role": "system", "content": "You are a helpful assista
nt at KIT."},
```

```
            {"role": "user", "content": "Explain the principle of Ra
yleigh scattering in three sentences."}
        ]
    )

    # Output answer
    print(chat_completion.choices[0].message.content)

except Exception as e:
    print(f"An error has occurred: {e}")
```

Use of the RAG system (knowledge base)

With Retrieval-Augmented Generation (RAG), you can provide the model with your own documents as a knowledge base for its answers.

Upload via the web interface (for easy starting)

For basic use and the upload of individual documents, you can use the web interface.

- **Where?** The upload area can be found under Tools > Knowledge.
- **What?** This method works well for typical file formats such as PDF, TXT, etc.

Upload via API and best practices (for advanced use)

For automation or the upload of larger volumes of documents, the use of the API is more convenient.

- **API endpoints:** The specific endpoints for managing files and knowledge bases can be found in the interactive API documentation (https://ki-toolbox.scc.kit.edu/docs).
- **Data preparation recommendation:** To get the best results with RAG, it's a good idea to prepare your source documents in a clean, structured Markdown format before uploading them. This allows the model to interpret the content more reliably. Tools like Docling can be helpful in this conversion.

Using the LiteLLM API (MyLiteLLM)

The LiteLLM API offers direct LLM access without KI-Toolbox functions (see Two Access Routes).

Step 1: Create an API key in the MyLiteLLM portal

1. Log in at https://my.litellm.scc.kit.edu.
2. Create a new API key at https://my.litellm.scc.kit.edu/apikeys.
3. Set the validity period per key and, if required, a budget; you can also view your usage to date there.

This allows you to use several API keys with different budgets and separate usage tracking.

Step 2: Test authentication

```
curl --fail --header "Authorization: Bearer $LITELLM_API_KEY" "https
://my.litellm.scc.kit.edu/api/v1/status"
```

Step 3: Use in tools (e.g. Cline)

1. Select **LiteLLM** as the API provider.
2. Enter https://litellm.scc.kit.edu as the Base URL.
3. Use the API key from MyLiteLLM and select the model from the model list, e.g. alias.medium-local.

The documentation of the LiteLLM REST API can be found at https://litellm.scc.kit.edu/.

Model Aliases

To simplify handling of the models in the KI-Toolbox and in external tools, there are model aliases that you can use instead of the model ID. This means you do not have to adjust the model ID when a model changes. Please note that the underlying base models (and thus also the costs and behaviour of the models) can change. The aliases for the local models, for GPT and for Claude follow the scheme below:

Scheme	Local	GPT	Claude
simple: Efficient models for simple tasks – low costs, low resource consumption, fast results	`alias.simple-local` (Mistral 4 Small)	`alias.simple-gpt` (GPT-6 Luna)	`alias.simple-claude` (Claude Haiku 4.5)
medium: Balanced model for everyday tasks – moderate costs, resources and speed	`alias.medium-local` (DeepSeek V4 Flash)	`alias.medium-gpt` (GPT-5.6 Terra)	`alias.medium-claude` (Claude Sonnet 5.5)
complex: Powerful model for complex tasks – at the cost of high costs and long execution times	`alias.complex-local` (GLM-5.3 Flash)	`alias.complex-gpt` (GPT-6.1 Sol)	`alias.complex-claude` (Claude Opus 5.5)

Alternatively, you can also address models manually and specifically. The current model IDs can be found at https://www.scc.kit.edu/dienste/ki-toolbox.php.

Info & Contact

License Notice

This manual from the Center for Technology-Enhanced Learning (ZML) at the Karlsruhe Institute of Technology (KIT) is licensed under a Creative Commons Attribution 4.0 International License.

Imprint

Publisher: Karlsruhe Institute of Technology (KIT) Kaiserstraße 12 76131 Karlsruhe

Contact: InformatiKOM Adenauer Ring 12 76131 Karlsruhe Germany Tel.: +49 721 608-48200 E-mail: info@zml.kit.edu

Questions about the KI-Toolbox should be directed to: ki-toolbox@scc.kit.edu