Entwicklung mit der KI-Toolbox

Die KI-Toolbox der TU Braunschweig bietet API-Zugriff auf die On-Premise-Modelle. Damit können Beschäftigte der Universität KI-gestützte Software lokal nutzen oder die Softwareentwicklung vereinfachen, indem sie die KI als Assistent direkt in der lokal installierten integrierten Entwicklungsumgebung (IDE) einsetzen.

- IntelliJ Coding Extension
- OpenCode

IntelliJ Coding Extension

Für das Nutzen des IntelliJ AI Chats kann man entweder auf das Icon rechts oben am Rand der IDE klicken oder dieses über Tools in der oberen Menüleiste aufrufen.

Es wird das AI Assistant Plugin benötigt, was einem IntelliJ auch vorschlägt.



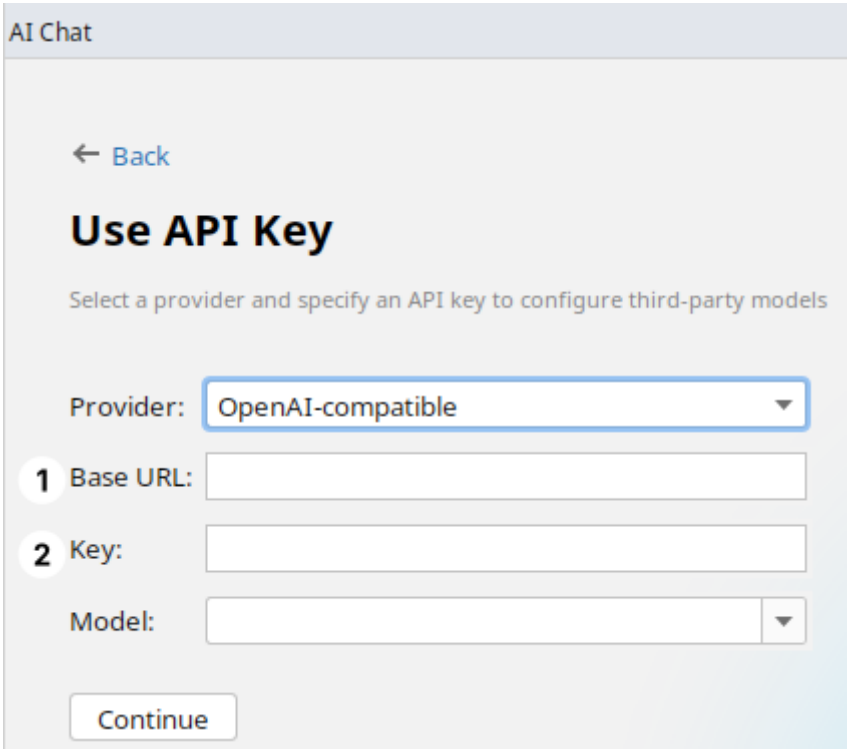
Wir nutzen unsere lokalen Modelle und daher die Option für Third-Party AI providers.

1) Base Url: https://ki-toolbox.tu-braunschweig.de/api/gateway/on-premise/v1

2) Den eigenen API Key findet man unter: https://ki-toolbox.tu-braunschweig.de/authenticationToken/show

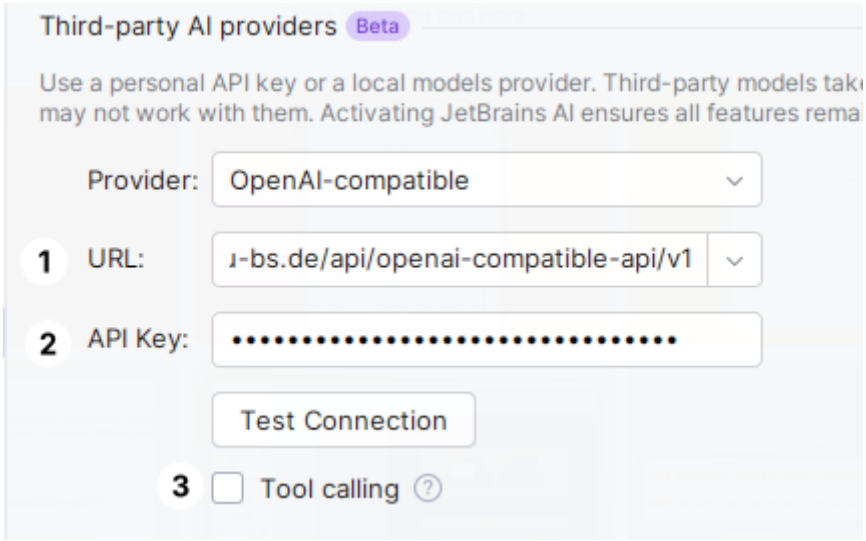
3) In den Settings muss nach der ersten Anmeldung unter Tools > Providers & API keys die Option "Tool calling" deaktiviert werden.

Bei der ersten Nutzung öffnet sich dieses Fenster:

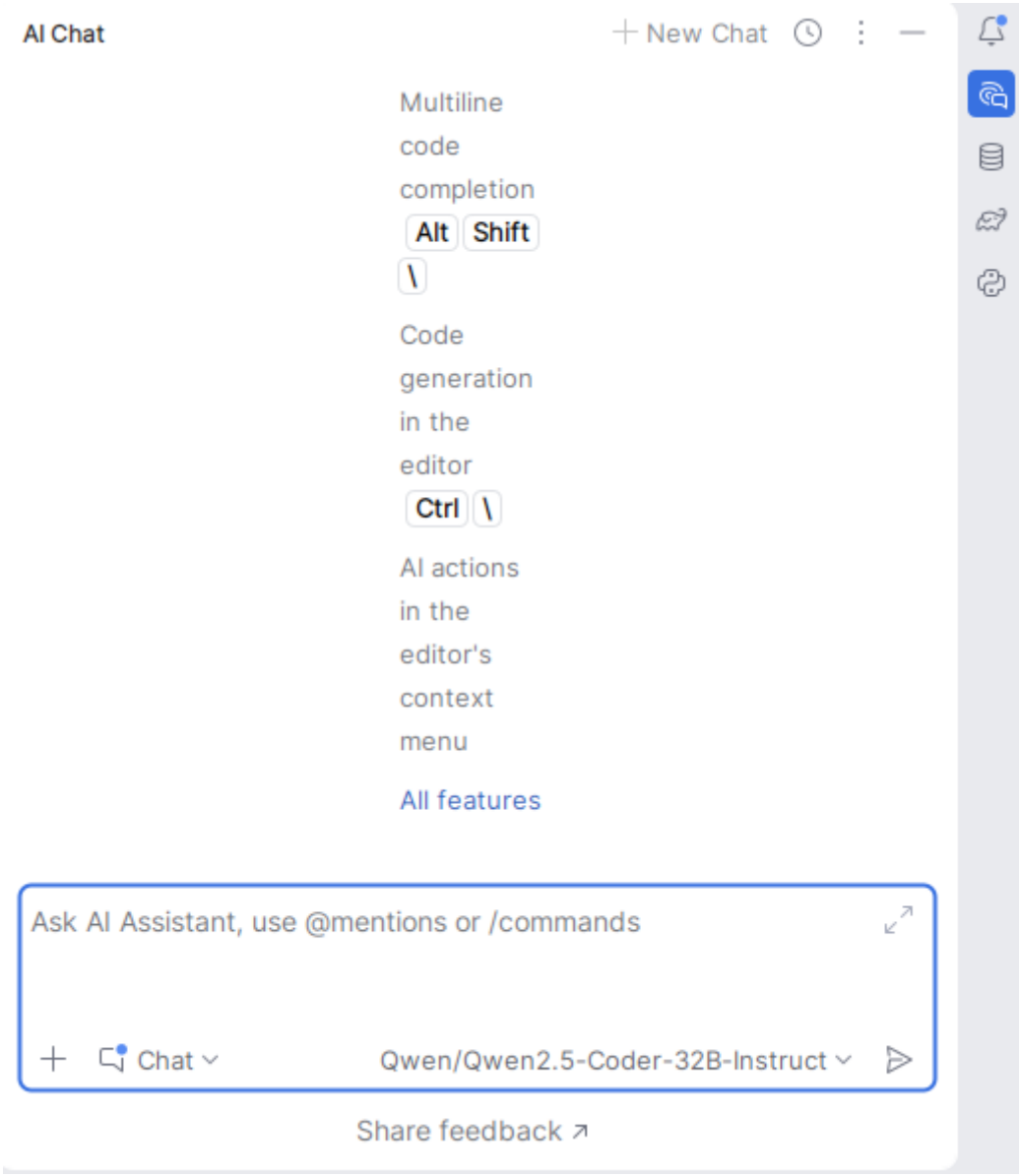


Man kann dann noch das Model Auswählen, welches man nutzen möchte.

Settings > Tools > Providers & API keys



Die Extension kann dann einfach als Chat in IntelliJ genutzt werden:


AI Chat
New Chat
Multiline code completion
Alt Shift \
Code generation in the editor
Ctrl \
AI actions in the editor's context menu
All features
Ask AI Assistant, use @mentions or /commands
Chat
Qwen/Qwen2.5-Coder-32B-Instruct
Share feedback

OpenCode

Um OpenCode zu installieren, bitte folgen Sie den Schritten die in der Dokumentation (https://opencode.ai/docs/) beschrieben sind.

Aus Sicherheitsgründen sollte der Agent in einer eigenen VM oder einem eigenen Container oder mindestens einen anderen Benutzeraccount (z.B. Linux user) ausgeführt werden, der nur für die Projektentwicklung benutzt wird. Es wird nicht empfohlen Agenten mit einem Benutzeraccount auszuführen, der auch Zugriff auf sensible personenbezogene Daten besitzt, z.B. wurden für die manuelle Bearbeitung PDFs in den Downloads Ordner gespeichert, wenn diese PDFs personenbeziehbare Daten beinhalten, darf der Agent nicht auf diese Daten zugreifen. Der Benutzername z.B. wenn die TUBS-ID als Benutzername verwendet wurde, kann bereits personenbeziehbare Daten beinhalten und an externe Anbieter übertragen werden, da der Benutzername einen Teil des absoluten Pfades des Home-Verzeichnisses bildet.

Es gibt verschiedene Optionen OpenCode zu nutzen, z.B. in dem Terminal, in einem Browser oder als IDE-Plugin.

Beispielsweise bei TUI ("Terminal User Interface"), kann man mit dem Kommando */connect* die Zugriffsdaten (API-Key) für das Modell eingeben. Wir nutzen unsere lokalen Modelle und daher die Option für "Other". Den eigenen API Key findet man unter: https://ki-toolbox.tu-braunschweig.de/authenticationToken/show

Weitere Optionen für die Modelle können in eine Datei opencode.json (/home/<user>/.config/opencode/opencode.jsonc) eingetragen werden. Beispiel:

```
{
  "$schema": "https://opencode.ai/config.json",
  "providers": {
    "TU-BS Extern Models": {
      "name": "TU-BS Extern Models",
      "package": "@opencode/ai/providers/openai-compatible-responses",
      "settings": {
        "baseURL": "https://ki-toolbox.tu-braunschweig.de/api/gateway/extern/v1",
      },
      "models": {
          "gpt-5.6-terra": {
            "name": "OpenAI gpt-5.6-terra"
          },
```

```
      "gpt-5.6-luna": {
        "name": "OpenAI gpt-5.6-luna"
      },
      "gpt-5.4": {
        "name": "OpenAI gpt-5.4"
      },
      "gpt-5.4-mini": {
        "name": "OpenAI gpt-5.4-mini"
      },
      "gpt-5.4-nano": {
        "name": "OpenAI gpt-5.4-nano"
      },
      "gpt-5.2": {
        "name": "OpenAI gpt-5.2"
      },
      "gpt-5.1": {
        "name": "OpenAI gpt-5.1"
      },
      "gpt-5": {
        "name": "OpenAI gpt-5"
      },
      "o3": {
        "name": "OpenAI o3"
      },
      "gpt-4.1": {
        "name": "OpenAI gpt-4.1"
      },
      "o4-mini": {
        "name": "OpenAI o4-mini"
      },
      "gpt-4o": {
        "name": "OpenAI gpt-4o"
      },
    }
},
"TU-BS On-Premise Models": {
  "name": "TU-BS On-Premise Models",
  "package": "@opencode/ai/providers/openai-compatible-responses",
  "settings": {
    "baseURL": "https://ki-toolbox.tu-braunschweig.de/api/gateway/on-premise/v1",
```

```
      },
      "models": {
        "Qwen/Qwen3.8-27B": {
          "name": "Qwen/Qwen3.8-27B",
        },
      }
    }
  },
}
```

Dann muss Opencode neu gestartet werden. Der API Key kann über */connect* eingegeben werden.



Danach muss nach "TU-BS" gesucht werden, um "TU-BS On-Premise Models" auszuwählen.



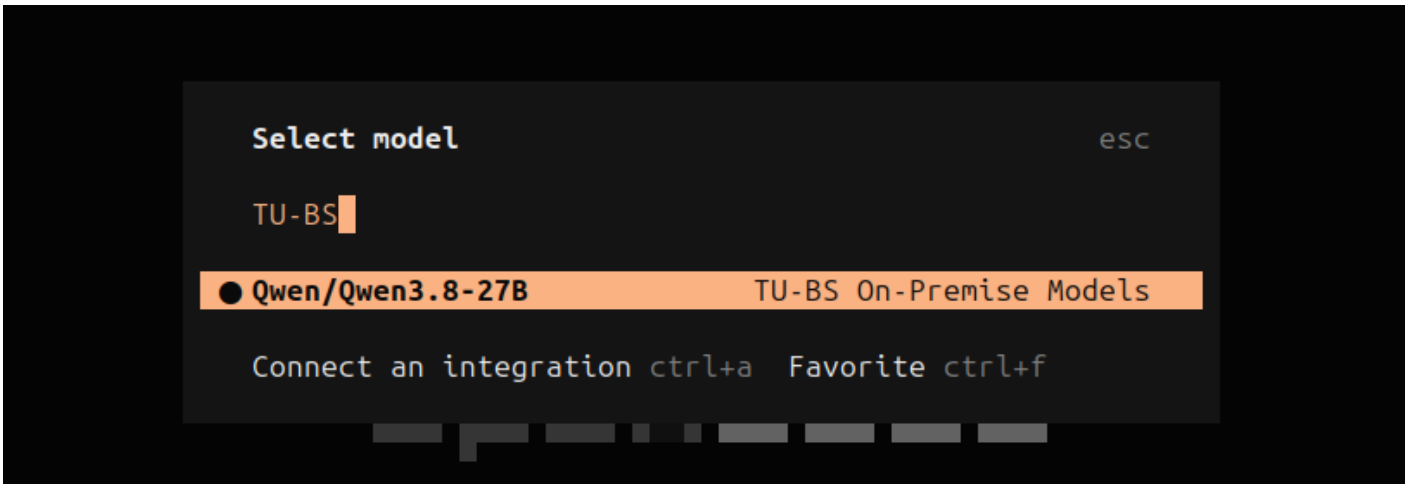
Sobald der Service per Enter ausgewählt wurde, wird der API Key gefordert, der auf https://ki-toolbox.tu-braunschweig.de/authenticationToken/show gefunden werden kann.



Mit dem Kommando */models* können die Modelle gewechselt werden.



In der Suchleiste kann "TU-BS" eingegeben werden, um nach den TU-BS Modellen zu suchen.



Durch die Enter Taste wird die Modellauswahl bestätigt und es kann gearbeitet werden.

