

# Hands-on-Workshop: Eigene KI-Agenten mit LangChain.js & LangGraph.js

## Dein Einstieg in die Welt der Retrieval-Augmented Generation

Liebe Teilnehmerin, lieber Teilnehmer,

bist du bereit, deine KI-Skills auf das nächste Level zu bringen? Dann willkommen zum Hands-On Workshop.

Bei diesem intensiven Tag tauchen wir tief in die Welt der KI ein und ich zeige dir, wie du **Sprachmodelle mit deinen eigenen Daten fütterst**, um wirklich smarte und anpassbare Lösungen zu bauen. Gemeinsam werden wir **RAG-Architekturen verstehen, mit LangChain.js arbeiten, Vektordatenbanken einbinden und deine KI-Workflows optimieren**.

Egal, ob du gerade erst in die KI-Welt einsteigst oder schon Erfahrung hast - dieser Workshop bringt dich garantiert weiter! Freue dich auf **spannende Hands-On-Sessions, Geek-Talks und praxisnahe Projekte, die du direkt umsetzen kannst**.

### Was ihr vorbereiten solltet:

Damit wir direkt durchstarten können, bitte ich euch, folgende Software auf euren Maschinen zu installieren. Keine Sorge, alles läuft plattformübergreifend auf Windows, Mac und Linux:

#### 1. Node.js:

- **Version:** Ich nutze die LTS-Version 22.x.
- **Wichtig:** Stellt sicher, dass npm funktioniert, insbesondere wenn ihr hinter einem Proxy sitzt.

#### 2. Burp Suite Community Edition:

- **Hinweis:** Standardmäßig läuft es auf Port 8080, kann aber bei Bedarf angepasst werden.

#### 3. Docker Desktop:

- **Anforderung:** Wir müssen in der Lage sein, Images zu laden. Siehe dazu die Anleitung zur Installation von Open WebUI weiter unten.

## Zugriff auf Sprachmodelle:

Wir werden sowohl lokale als auch Cloud-basierte Sprachmodelle verwenden. Optimal wäre es, wenn ihr beide Varianten einrichten könnt, um alle Szenarien kennenzulernen. Hauptsache, eine Variante funktioniert bei euch.

In der Cloud:

- **OpenAI:**
  - **Anforderung:** Ein Account mit hinterlegter Kreditkarte und aufgeladenem Guthaben.
- **Azure AI Foundry / Azure OpenAI:**
  - **Anforderung:** Ein Account mit hinterlegter Kreditkarte und den entsprechenden Berechtigungen. (Das werde ich verwenden.)

Lokal auf eurem Rechner:

- **LM-Studio:**
  - **Beschreibung:** Eine kostenlose plattformübergreifende Desktop-App. (Werde ich einsetzen.)
- **Ollama:**
  - **Hinweis:** Läuft über Docker.

Benötigte Sprachmodelle für lokale Lösungen:

- Qwen2.5 7B Instruct (*ca. 4,36 GB*)
- multilingual-e5-base-gguf (*ca. 300 MB - Embedding-Modell*)

Installation von Open WebUI via Docker:

Open WebUI ermöglicht uns eine benutzerfreundliche Oberfläche für unsere KI-Modelle.

Docker-Befehl zum Starten:

```
docker run -d -p 8181:8080 --add-host=host.docker.internal:host-gateway --name open-webui ghcr.io/open-webui/open-webui:main
```

**Hinweis:** Dieser Befehl startet Open WebUI und macht es lokal unter <http://localhost:8181> zugänglich.

Zur Info: Ich werde **Visual Studio Code** als IDE und **GitHub Copilot** nutzen - ihr könnt aber selbstverständlich mit euren bevorzugten Tools arbeiten.

Falls du Probleme bei der Einrichtung hast, melde dich gerne vorher bei mir. Falls dein System nicht die nötige Leistung mitbringt oder du durch Berechtigungen eingeschränkt bist, kann ich dir für den Workshop-Tag eine virtuelle Maschine bereitstellen.

Ich freue mich darauf, dich kennenzulernen und gemeinsam mit dir in die faszinierende Welt von RAG & KI-Optimierung einzutauchen. Ein Tag voller **Technik, Praxis und jeder Menge Nerd-Talk** erwarten dich!

Falls du Fragen hast, zögere nicht, mich zu kontaktieren: [gregor.biswanger@web-enliven.de](mailto:gregor.biswanger@web-enliven.de)

Lass uns durchstarten - ich kann's kaum erwarten! 🚀

Beste Grüße aus Nürnberg,  
Gregor Biswanger

