

Wir sind eine zertifizierte B Corporation.

HOME ► WISSENS-HUB ► BLOG

// GenAI für Full Stack EntwicklerInnen: Aller Anfang ist... lokal? (Teil 1)

LLM

Künstliche Intelligenz



Robin Schlenker

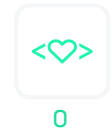
14.6.2024 | 7 Minuten Lesezeit

Als Full Stack EntwicklerIn gibt es heutzutage wohl genug Themenkomplexe zur Einarbeitung. Ob das nächste Frontend Framework des Jahres, die neue Backend-Technologie, einen weiteren Security-Scanner oder doch nur eine weitere Cloud-Integration. Die Auswahl ist schwer, am Ball zu bleiben fast unmöglich. Doch während wir versuchen, irgendwie eine Balance zwischen Wissenstiefe und Schweizer Taschenmesser zu finden, ploppen immer wieder neue Trends auf, die uns unsere Kompetenz in Frage stellen lassen. Generative künstliche Intelligenz, oder GenAI, war für mich ein solches Thema. Seit Jahren purzelte es irgendwo im Internet herum, mal in den Schlagzeilen, mal nur auf der vergilbten Overheadfolie eines Tübinger Uni-Professoren. Doch spätestens seit ChatGPT 2022 veröffentlicht wurde ist der Hype-Train völlig aus dem Ruder gelaufen. An jeder Ecke sprießen neue, bessere Chatbots hervor und kaum ein Tool von Rang und Namen kann es sich mittlerweile noch leisten, nicht wenigstens irgendetwas mit KI zu machen. Bislang hatte ich es gekonnt

Beitrag teilen



Diese Seite ist eine Vorschau. [Klicken Sie hier](#), um den Vorschaumodus zu verlassen.



bröckeln. Immer mehr Kunden suchten nach KI-EntwicklerInnen und möchten intelligentere Services bauen.

Da habe ich mich als Full Stack Entwickler gefragt wie lange wir diesem Trend noch entgehen können? Was wäre denn eine entsprechende Ergänzung meines Portfolios um auch noch in 2 Jahren wenigstens mitreden zu können? Wo liegt die Grenze zwischen ausreichender Wissenstiefe und dem KI-Experten der denkt, Vue sei das französische Wort für Augenlicht. In dieser Blogartikelserie möchte ich diesen Fragen nachgehen. Ich möchte herausfinden was ich als Full Stack Entwickler tun kann, um der "neuen" KI-Welt angemessen begegnen zu können. Was sind die ersten Themen die ich mir aneignen sollte? Wie kann ich von GenAI profitieren und lohnt sich das überhaupt? Werde ich in ein paar Wochen das Schreiben an den Nagel hängen können und mein selbst-trainiertes KI-Modell macht die ganze Arbeit? Und mal ganz unter uns: Wie schwer kann es denn eigentlich sein nachdem wir es mittlerweile geschafft haben <div>-Elemente ordentlich zu zentrieren!

Working on my machine

Als ich vor ein paar Tagen blauäugig an dieses Thema herangetreten bin war meine Vorstellung von GenAI die Folgende: "Große Firmen haben riesige Teams an Wissenschaftlern über Jahre hinweg in einen Keller gesperrt, ein paar hundert Mainframes gesponsert und am Ende ward Licht (Irgendwo im Rechenzentrum). Und die Menschheit sah, dass es gut war." In meiner Wahrnehmung waren LLMs (Large Language Models) das Ergebnis riesiger Rechenleistungen. Sie benötigten unzählige Grafikkarten, Terabytes an Festplattenspeicher und die Ressourcen einer Cloud, um brauchbare Ergebnisse zu liefern. Sicher, gerade die Giganten der Branche wie OpenAI, Meta und Google machen das auch so, doch für mich als kleinen Full Stacker war diese große, hochskalierte Welt zunächst vor allem Eines: Einschüchternd. Als ich dann von einigen unserer KI-Experten lernen durfte, dass die Magie auch schon im Kleinen passiert war ich sofort Feuer und Flamme. Denn wann immer ich in der Lage

was auf meiner eigenen Maschine laufen zu lassen wäre der

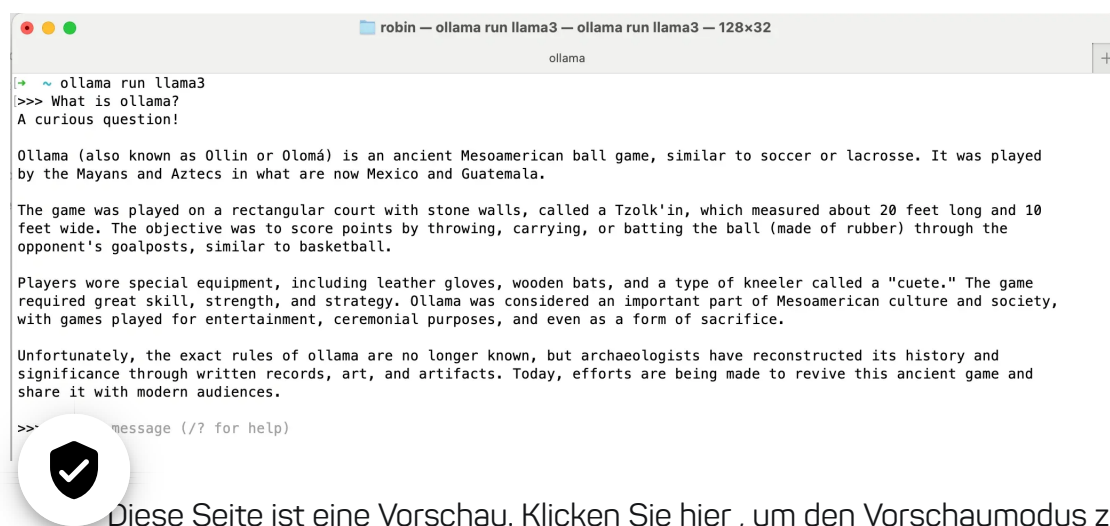
Diese Seite ist eine Vorschau. [Klicken Sie hier](#), um den Vorschaumodus zu verlassen.



Kontrolle über das was da passiert und müsste mich zudem auch nicht um Lizenzen, Abrechnung und ähnliche Themen kümmern. In den nächsten Absätzen möchte ich euch also vorstellen, mit welchen Tools ein erster, entspannter Einstieg in das Thema GenAI möglich ist, komplett ohne eine Internetanbindung.

ollama - Ein kleiner Schritt für meine Festplatte, ein großer Schritt für's Selbstbewusstsein

4.7 Gigabyte. Mit dieser heutzutage lächerlich kleinen Anzahl an Bytes verspricht llama3, eines der bekanntesten LLMs der Szene, das beste frei verfügbare KI-Modell zu sein. Fair enough, llama3 hat auch noch viel größere Modelle, doch dank Komprimierung (Quantization) und kleineren Datensätzen haben die KollegInnen von Meta es geschafft, ihre Chat-KI auf diese wunderbar handliche Größe zu schrumpfen. Fragt man llama3 wäre Ollama zwar ein mesoamerikanisches Ballspiel, für uns ist es vor allem eine Website und ein CLI-Tool, um ebensolche Opensource KI-Modelle verfügbar zu machen. Nachdem man sich das [CLI-Tool](#) heruntergeladen und installiert hat ist der erste lokale Chatbot nicht mehr fern. Ein einfaches ollama run llama3 lädt das Modell auf den Laptop und startet direkt einen Chat. Auch wenn llama3 noch keine Kenntnisse von ollama hat ist die Webseite an sich eine Goldgrube. Dort gibt es allerlei KI-Modelle jedweder Couleur, gut nutzbare Anleitungen und allein das süße Llama des Logos ist einen Besuch wert!



Diese Seite ist eine Vorschau. [Klicken Sie hier](#), um den Vorschaumodus zu verlassen.

Ein kleiner Schmankerl obendrauf ist, dass alle mit ollama ausgeführten Modelle auch über eine REST-API angesprochen werden können.

```
curl http://localhost:11434/api/generate -d '{
  "model": "llama3",
  "prompt": "Why is the sky blue?",
  "stream": false
}'
```

So wäre es schon in diesem Minimal Setup möglich, einen ersten kleinen Chatbot verfügbar zu machen und das ganz ohne aufwändige Cloud-KI Integration.

Die Antworten des Bots übertreffen meine Erwartungen zwar um Längen, können sich aber natürlich nicht vollständig mit ChatGPT und Co messen. Da es bei ollama auch viel spezialisiertere Modelle gibt lassen sich aber schnell bessere Ergebnisse produzieren, je nach Use Case.

Open WebUI - Let it shine!

So schön ein Terminal auch sein mag, richtig nutzbar ist unser lokaler Chatbot so wohl noch kaum. Das Open WebUI Projekt kann uns hier schnell Abhilfe leisten. Mit einem stark an ChatGPT erinnernden Interface lassen sich damit schnell alle ollama-Modelle zu auch von Laien nutzbaren KI-Partnern umfunktionieren. Am einfachsten geht das mit Docker.

```
docker run -p 3000:8080 -e WEBUI_AUTH=False -v open-webu
```

Hier gibt es noch ein paar Details zu beachten:

- **Je nach Setup brauchst du noch eine Verbindung zum Host-System. Bei meinem MacOS-Setup mit Rancher Desktop reicht der obige Befehl, bei Docker Desktop brauchst du noch: `--add-host=host.docker.internal:host-gateway` damit das WebUI deine lokale ollama Instanz findet. Mehr Infos dazu findest du in**

dem Ticket.

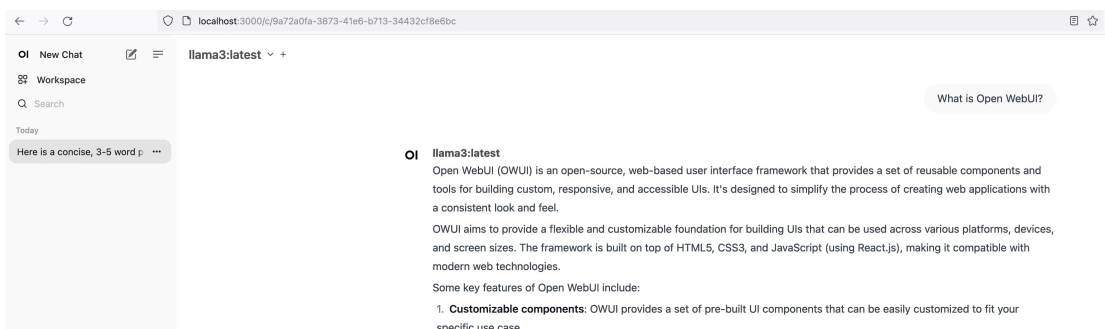


Diese Seite ist eine Vorschau. [Klicken Sie hier](#), um den Vorschaumodus zu verlassen.

System erst einmal nur lokal hosten. Für komplexere Setups ist das natürlich nicht geeignet.

- **Das Interface wird auf Port 3000 erreichbar sein**

Ist das Setup abgeschlossen, finden wir unter <http://localhost:3000> unser eigenes kleines "ChatGPT".



Das Open WebUI interface erinnert doch schon stark an OpenAI's ChatGPT

Continue - Code Completion

Das dritte Tool für die ersten Schritte in die Welt der generativen KI schließt die Brücke zum Full Stack wieder. Daher habe ich mir angeschaut welche Code-Completion es in diesem lokalen Setup gibt. Zuerst benötigen wir für die lokale Codinghilfe ein KI-Modell, welches zumindest etwas spezialisierter ist als das llama3. Nach einem kurzen Blick auf ollama habe ich mich für das 1.7GB große [StarCoder2](#) entschieden. Neben anderen, deutlich größeren und potenteren Alternativen wie zum Beispiel Mistral's neuem [codestral](#), ist der Overhead an Rechenpower bei StarCoder einfach geringer. Nach der gewohnten Installation mit ollama fehlt dann nur noch eine Integration für die IDE. Hier kommt das Continue Projekt ins Spiel, das z.B. bei IntelliJ über den Plugin Mechanismus installiert werden kann. Dann muss noch kurz das richtige Modell in Continue eingestellt werden und fertig ist die KI-basierte Coding Unterstützung. Meiner Erfahrung nach sind viele KI-generierte Codeblöcke eher hinderlich als hilfreich, aber mit ein bisschen Übung lernt man die Diamanten herauszufiltern.

Beim Coolers cool an Continue ist auch das Erklär-Feature, das mit einer einfachen



Diese Seite ist eine Vorschau. [Klicken Sie hier](#), um den Vorschaumodus zu verlassen.

Grundsatzdiskussion über Kommas am Ende der Zeile mit dem Chatbot einfach kein Ende findet.

Continue

TS schema.ts (14-27)

```
export const schema = createSchema({
  typeDefs: file,
  resolvers: {
    DateTime: DateTimeResolver,
    Query: {
      timeslots: async (_, args) => {
        return await timeslotLoader.loadAll(args)
      },
      locations: (): Location[] => {
        return timeslotLoader.getLocations()
      },
    },
  },
})
```

What is this code doing?

🔔

🌀

🗄️

🔍

This TypeScript code defines a GraphQL schema using the `createSchema` function from a library such as Apollo Server, which is commonly used to create and manage GraphQL schemas in a Node.js environment. Here's an explanation of what each part does:

1. The `schema` constant is being assigned the result of calling `createSchema`. This function takes an object that defines both the type definitions (`typeDefs`) and resolvers for the schema.
2. `file` refers to a GraphQL schema definition, which could include queries, mutations, subscriptions, and various data types such as scalars, enums, and interfaces. This file is likely defined using the Schema Definition Language (SDL). However, its content is not shown in this code snippet.
3. `DateTimeResolver`: This is a custom resolver that handles converting DateTime values to and from JavaScript's built-in Date object. The exact implementation of this resolver is not shown here.

Eine solch brauchbare Antwort hätte ich eigentlich nicht erwartet, Continue ist eine tolle IDE-Integration

Was kommt als nächstes?

Mit diesem Setup ist der erste Schritt in die spannende Welt der generativen KI



Diese Seite ist eine Vorschau. [Klicken Sie hier](#), um den Vorschaumodus zu verlassen.

die Technik und Terminologien im GenAI Umfeld ohne sich selbst die Hände schmutzig zu machen. Im nächsten Artikel werde ich über einen ersten konkreten Usecase schreiben. Denn so nett die Chatbots auch sind, von ollama und meinen eigenen Dokumenten scheinen sie nicht viel zu verstehen. Um das zu ändern schauen wir uns sogenannte RAG-Systeme (Retrieval-Augmented Generation) an. Eine Technik, die euch als Full Stack EntwicklerInnen höchstwahrscheinlich in den nächsten Jahren über den Weg laufen wird.

War dieser Beitrag hilfreich?

Ja



Blog-Autor*in



Robin Schlenker

Full Stack Consultant

Du hast noch Fragen zu diesem Thema?
Dann sprich mich einfach an.

Kontakt aufnehmen



Diese Seite ist eine Vorschau. [Klicken Sie hier](#), um den Vorschaumodus zu verlassen.

// Dein Job bei codecentric?

Jobs

Agile Developer und Consultant (w/d/m)

📍 Alle Standorte

Backend

Frontend

Fullstack

Zur Stellenanzeige ▶

// Weitere Artikel in diesem Themenbereich




Diese Seite ist eine Vorschau. [Klicken Sie hier](#), um den Vorschaumodus zu verlassen.

Eine Einführung in das Thema künstliche Intelligenz für Schülerinnen und...

Die Bedeutung von künstlicher Intelligenz wächst in der heutigen Welt. Doch wie funktioniert KI? Es hat zumindest nichts mit Magie zu tun – auch wenn KI gerne damit assoziiert wird. Normalerweise beantworte ich diese Frage ausführlich bei uns im IT ...

Künstliche Intelligenz

 30.1.2024 | 3 Minuten Lesezeit



Meike Wocken

Ersetzt KI die Softwareentwicklung?

In meinem letzten Blogbeitrag habe ich geschrieben, was KI-Tools schon leisten können, wie sie fachliche Anwendungen in Programmcode zu übersetzen und dass die Ergebnisse zwe...

Künstliche Intelligenz

 11.9.2023 | 6 Minuten Lesezeit

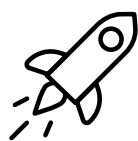


Goetz Markgraf

// Gemeinsam bessere Projekte umsetzen.



Diese Seite ist eine Vorschau. [Klicken Sie hier](#), um den Vorschaumodus zu verlassen.



Wir helfen deinem Unternehmen.

Du stehst vor einer großen IT-Herausforderung? Wir sorgen für eine maßgeschneiderte Unterstützung. Informiere dich jetzt.

Unsere Leistungen ▶



Hilf uns, noch besser zu werden.

Wir sind immer auf der Suche nach neuen Talenten. Auch für dich ist die passende Stelle dabei.

Zu den Jobangeboten ▶

@codecentric



Diese Seite ist eine Vorschau. [Klicken Sie hier](#), um den Vorschaumodus zu verlassen.

Unternehmen ▼

codecentric für dich ▼

Sitemap ▼

Rechtliches ▼



Diese Seite ist eine Vorschau. [Klicken Sie hier](#), um den Vorschaumodus zu verlassen.