

Aktueller Stand & Arbeitsweise für die Datenbank

Dieses Dokument beschreibt unseren aktuellen Entwicklungsstand und die geplante Vorgehensweise für die Datenbanknutzung. Bitte lest es sorgfältig durch, damit wir alle auf dem gleichen Stand sind.

1. Aktueller Stand: Die lokale Docker-Sandbox

Momentan verwenden wir für die Entwicklung und zum Testen ausschließlich einen lokalen "Sandbox"-Ansatz mit Docker.

Was bedeutet das?

- Jedes Teammitglied kann mit Docker eine private, lokale Postgres-Datenbank auf seinem eigenen Rechner starten
- Diese Datenbank wird aus den SQL-Skripten in unserem Projekt erstellt, sodass jeder die exakt gleiche Tabellenstruktur hat

Warum ist das ein guter Start?

- Konsistenz: Jeder hat eine identische Umgebung, was "Bei mir geht's aber nicht"-Fehler vermeidet
- Isolation: Wir können unabhängig voneinander arbeiten, ohne die Arbeit der anderen zu stören
- Sicherheit: Jeder kann seine lokale Datenbank nach Belieben testen, füllen und zurücksetzen (docker compose down -v), ohne Konsequenzen für das Team

2. Anleitung: Setup & Arbeiten mit Docker & Postgres

Voraussetzungen:

Docker Desktop installieren: Ladet euch Docker Desktop von der offiziellen Docker-Webseite herunter und installiert es. Es enthält sowohl Docker Engine als auch Docker Compose. Startet es nach der Installation.

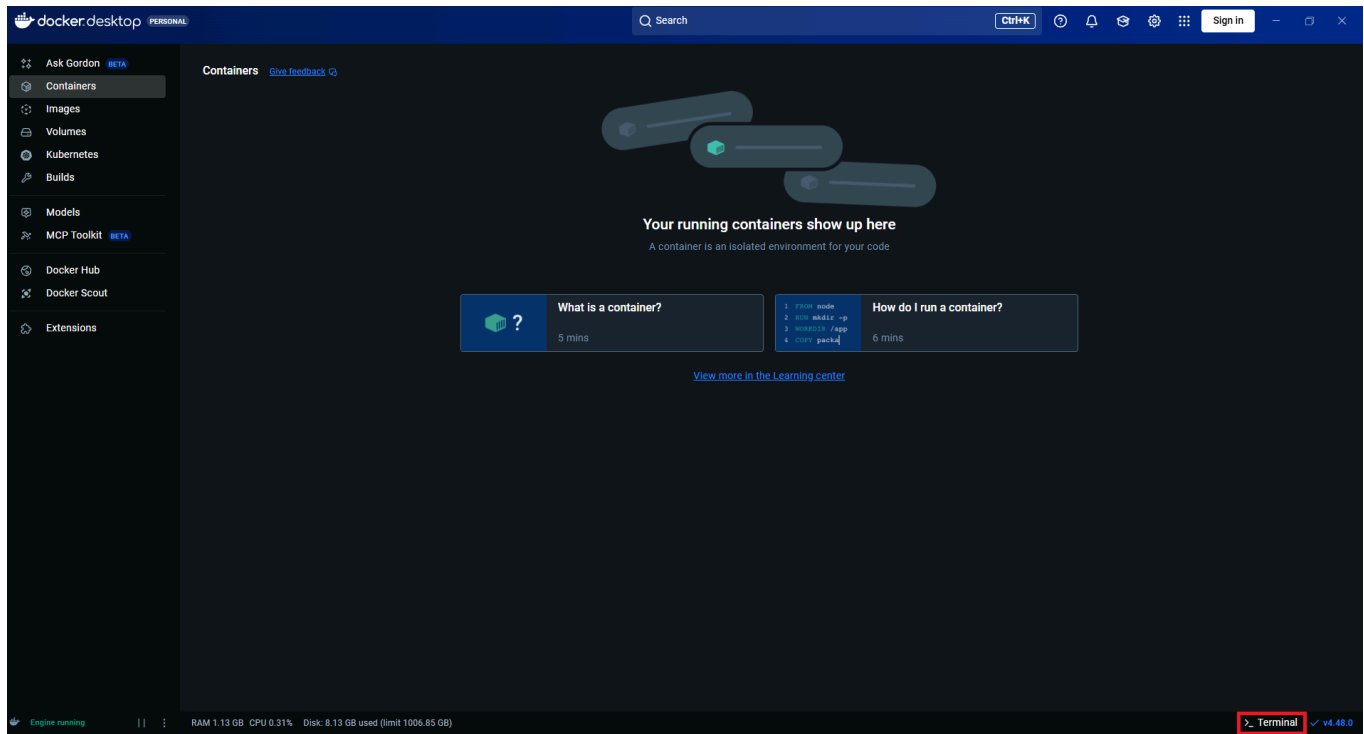
Schritte:

2.1 Auschecken des Repositorys

```
git clone https://github.com/username/projektname.git
```

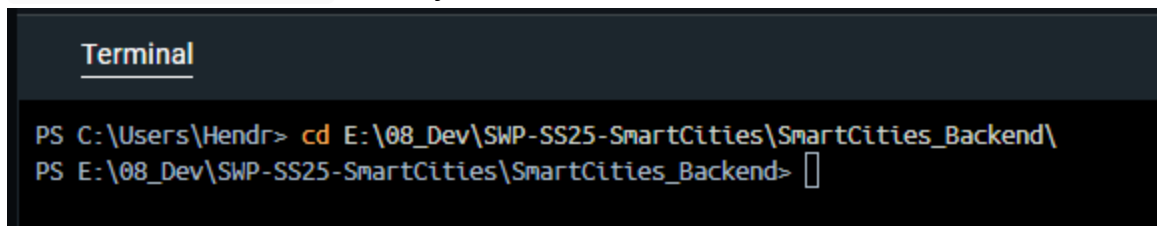
2.2 Starten des Docker-Stacks

Startet Docker-Desktop und startet mit einem Klick auf **Terminal** unten rechts in der Ecke eine Kommandozeile.



2.3 Starten des Docker-Stacks

Navigiere innerhalb der Console mit dem Befehl `$cd` in das Backend-Verzeichnis `SmartCities_Backend` im Projekt.



2.4 Starten des Docker-Stacks

Starte mit folgendem Befehl den Docker-Stack:

```
docker-compose up --build -d
```

Im Anschluss sollte unter dem Punkt `Containers` der `smartcities_backend` -Stack zusehen sein.

Ask Gordon BETA

Containers

Images

Volumes

Kubernetes

Builds

Models

MCP Toolkit BETA

Docker Hub

Docker Scout

Extensions

Containers [Give feedback](#)

Container CPU usage

0.50% / 1600% (16 CPUs available)

Container memory usage

374.51MB / 15.22GB

Show charts

Q Search

☐

Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	smartcities_backend	-	-	-	0.04%	50 seconds ago	<input type="checkbox"/> ⋮ <input type="trash"/>
<input type="checkbox"/>	postgres_db	cd9005fa810b	postgres	5432:5432	0.02%	50 seconds ago	<input type="checkbox"/> ⋮ <input type="trash"/>
<input type="checkbox"/>	node_api	c37bbc63c111	smartcities_backend-api	3000:3000	0%	50 seconds ago	<input type="checkbox"/> ⋮ <input type="trash"/>
<input type="checkbox"/>	postgres_admin	dfb326e8bc6c	doage/pgadmin4	8080:80	0.02%	50 seconds ago	<input type="checkbox"/> ⋮ <input type="trash"/>

2.5 Starten des Docker-Stacks

Zum Stoppen des Container unter Beibehalt aller Volumes und Container:

```
docker-compose down
```

Zum Stoppen der Container und löschen der Volumes: (Nach Updates vom Backend notwendig)

```
docker-compose down -v
```

⚠ Achtung: docker-compose down -v löscht auch alle gespeicherten Daten!

Verwende diesen Befehl, wenn z. B. das Datenbank-Schema geändert wurde und du die DB neu aufbauen willst.