7.4-1 | Grundgerüst eines Minimal API

1. Installation .NET

Prüfen Sie mit folgendem Command, ob .NET 8 bereits installiert ist.

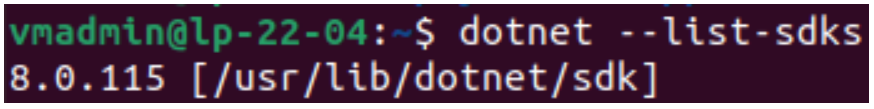
```
dotnet --list-sdks
```

Falls .NET 8 nicht aufgeführt ist, installieren Sie es wie folgt:

```
# Install dotnet 8 sdk
sudo apt-get update
sudo apt-get install -y dotnet-sdk-8.0
```

mit folgendem Command überprüfen Sie, ob .NET8 erfolgreich installiert ist.

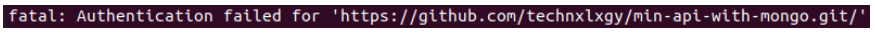
```
dotnet --list-sdks
```



2. Git Repository erstellen

Erstellen Sie ein leeres GIT-Repository “min-api-with-mongo” für Ihr Projekt. Fügen Sie dem Projekt ein README.md hinzu.

→ Beim Erstellen des Repositories muss die Visibility auf **Public** gesetzt sein weil sonst der Error auftaucht:



3. Grundgerüst erstellen

Klonen Sie das frisch angelegte Projekt mit git clone auf Ihre VM in ein beliebiges Verzeichnis (z.B. ~/Documents).

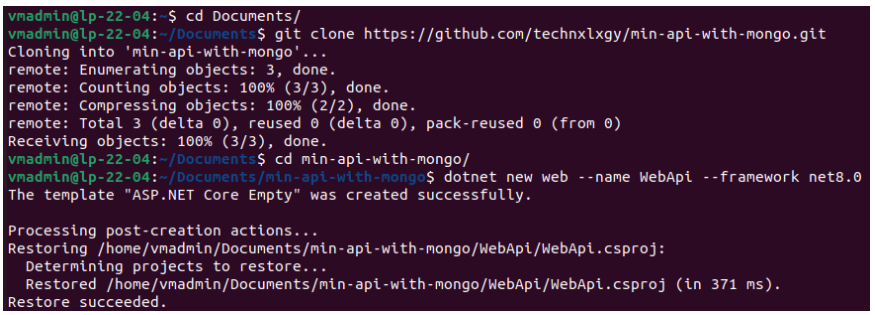
```
mkdir min-api
cd min-api
git clone [HTTPS von GitHub]
```

Navigieren Sie in das Projektverzeichnis min-api-with-mongo.

```
cd min-api-with-mongo
```

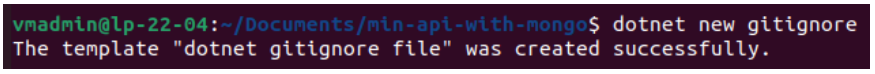
Erstellen Sie ein .NET Projekt WebApi mit Template web:

```
dotnet new web --name WebApi --framework net8.0
```

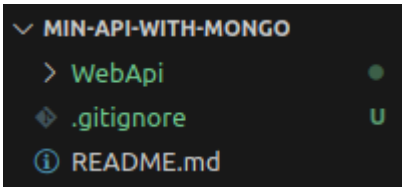


Erstellen Sie im gleichen Verzeichnis ein .gitignore. Es sorgt dafür, dass nur relevanter Source-Code ins GIT-Repository übertragen wird (ohne Binaries, etc.).

```
dotnet new gitignore
```

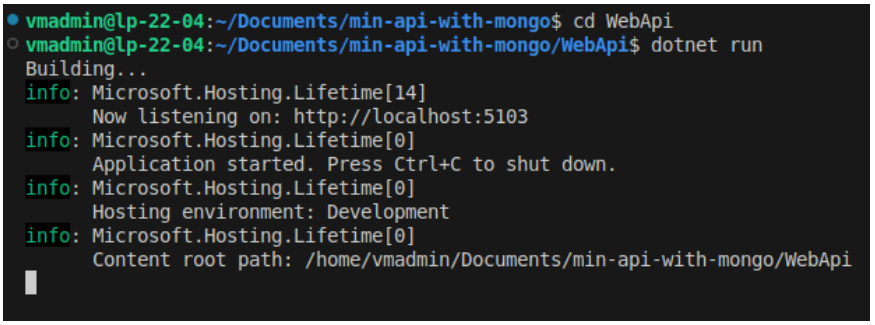


Öffnen Sie Visual Studio Code (VS Code) und öffnen Sie das Projektverzeichnis min-api-with-mongo (File → Open Folder). Ihre Projektstruktur müsste jetzt mit folgendem Bild übereinstimmen:

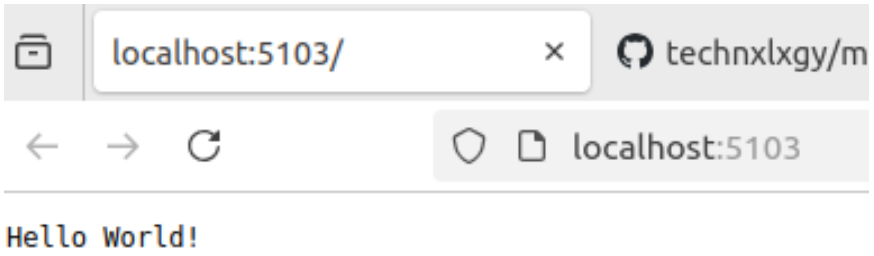


Öffnen Sie in VS Code ein Terminal (Terminal → New Terminal) Navigieren Sie in den Folder WebApi und starten Sie die Anwendung:

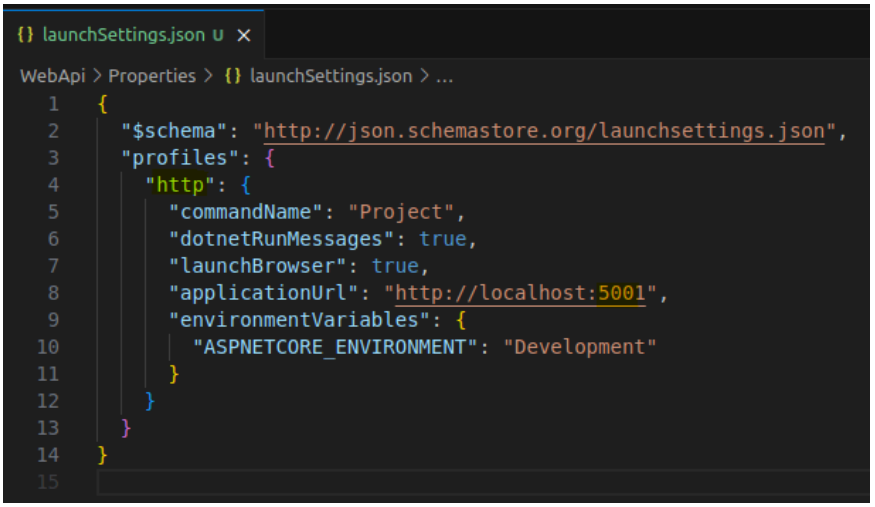
```
dotnet run
```



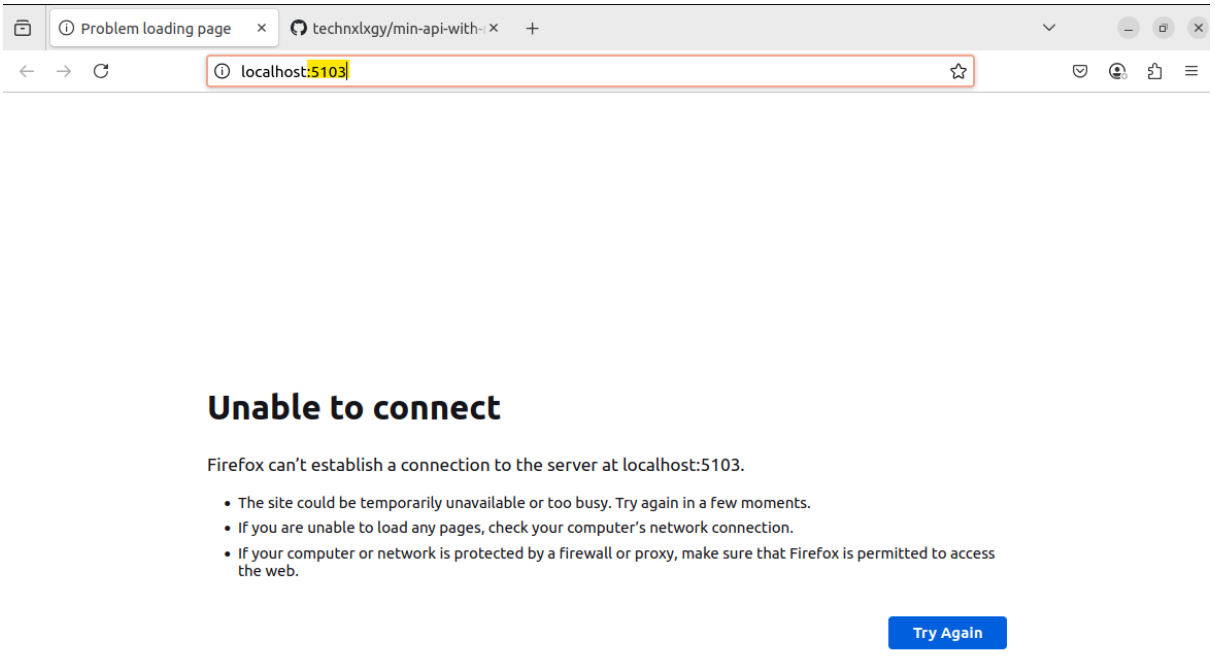
Öffnen Sie die in der Konsole ausgegebene URL und vergewissern Sie sich, dass im Browser Hello World angezeigt wird.



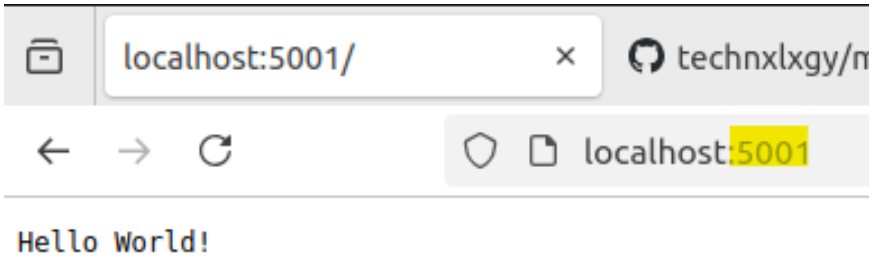
In der Datei Properties/launchSettings.json ist definiert, wie die Anwendung gestartet wird. Per Default sind verschiedene Profile hinterlegt. Passen Sie das http-Profil so an, dass die Anwendung über http://localhost:5001 (http-Profil) erreichbar ist. Löschen Sie die nicht benötigte Section iisSettings und die beiden Profile https und IIS Express und starten Sie die Anwendung neu. Sie müsste nun über http://localhost:5001 erreichbar sein.



alter Port:



neuer Port:



4. Dockerfile

Erstellen Sie im Verzeichnis WebApi ein Dockerfile für ein Multistage Image. Die Anwendung soll mit

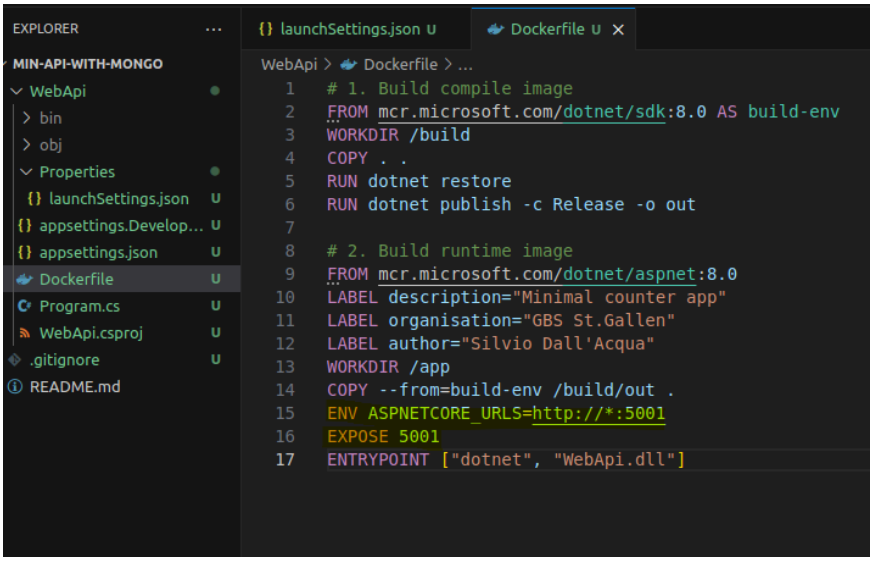
- Image mcr.microsoft.com/dotnet/sdk:8.0 restored und published werden
- Image mcr.microsoft.com/dotnet/aspnet:8.0 ausgeführt werden.

Die Anwendung soll wie bei lokaler Ausführung auch über http://localhost:5001 erreichbar sein.

Dockerfile:

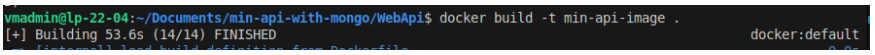
```
# 1. Build compile image
FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build-env
WORKDIR /build
COPY . .
RUN dotnet restore
RUN dotnet publish -c Release -o out

# 2. Build runtime image
FROM mcr.microsoft.com/dotnet/aspnet:8.0
LABEL description="Minimal counter app"
LABEL organisation="GBS St.Gallen"
LABEL author="Silvio Dall'Acqua"
WORKDIR /app
COPY --from=build-env /build/out .
ENV ASPNETCORE_URLS=http://*:5001
EXPOSE 5001
ENTRYPOINT ["dotnet", "WebApi.dll"]
```

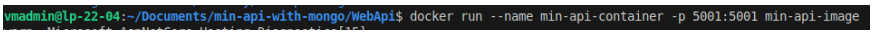


Das gelb markierte ist neu hinzugekommen (spezifisch für Webanwendungen)

```
docker build -t min-api-image .
```

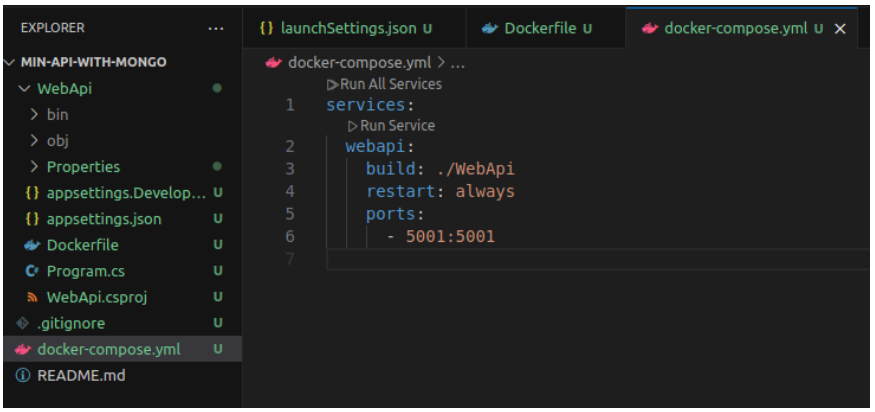



```
docker run --name min-api-container -p 5001:5001 min-api-container
```

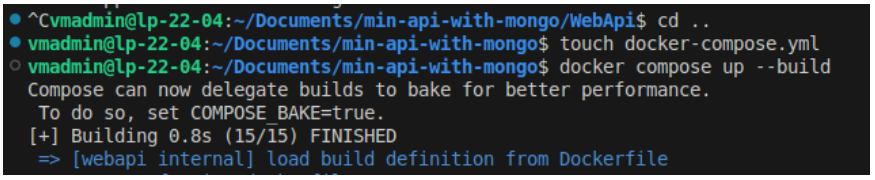


5. Docker-compose.yml

Erstellen Sie direkt im übergeordneten Projektverzeichnis min-api-with-mongo ein docker-compose.yml. Mit docker compose up soll die Anwendung mit Hilfe des Dockerfiles erzeugt und gestartet werden.



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



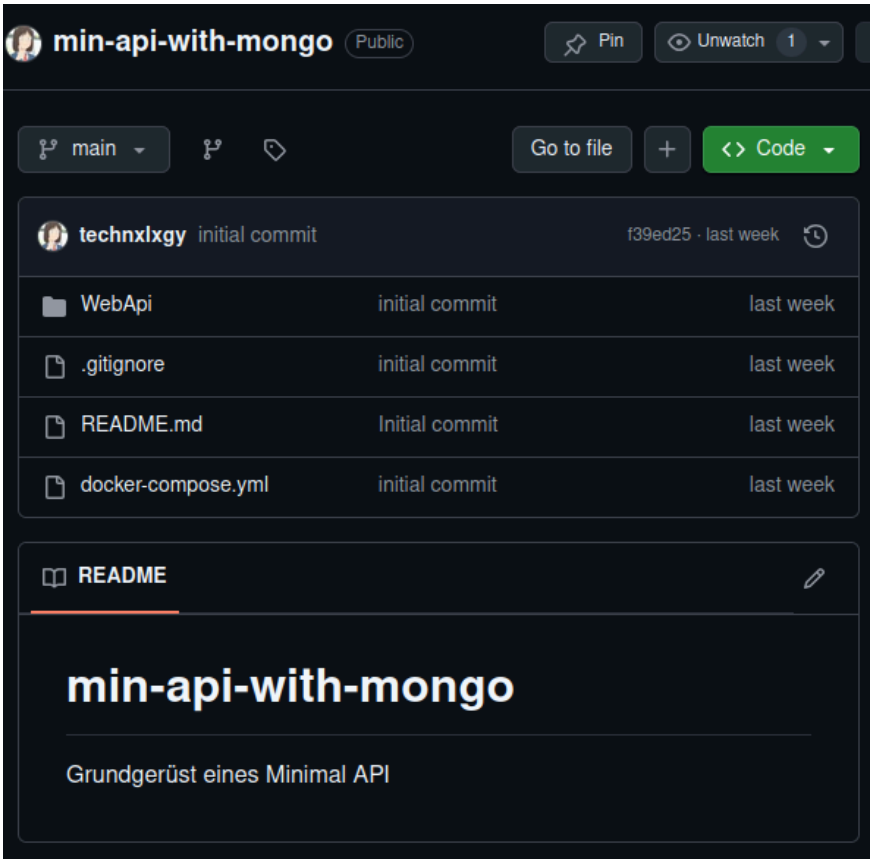
6. Commit und Push

Committen Sie Ihre Änderungen und Pushen Sie Ihren ersten Projektstand in Ihr Git-Repo. VS Code unterstützt Sie dabei mit der integrierten Source Control. Falls User und Email in Ihrem GIT-Client noch nicht gesetzt sind, machen Sie das wie folgt:

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

Hinweis Pushen Sie Ihre Änderungen nach jedem Schritt in Ihr GIT-Repo. So können Sie bei Bedarf jederzeit auf einem definierten Stand aufsetzen.

```
git add .
git commit -m "commit message"
git push
```


min-api-with-mongo
Public
Pin
Unwatch 1
main
Go to file
Code
technxlxgy initial commit
f39ed25 · last week
WebApi initial commit last week
.gitignore initial commit last week
README.md Initial commit last week
docker-compose.yml initial commit last week
README
min-api-with-mongo
Grundgerüst eines Minimal API