

# Grundgerüst eines Minimal API

---

## Installation .NET 8

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
```

---

## GIT-Repository erstellen

1. Ein neues Projekt im GitHub erstellen (public → funktioniert auf Linux nicht mit private und clone), mit README.md
2. Lokal Befehl ausführen:

```
git clone <repositoryURL>
```

---

## Grundgerüst erstellen

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

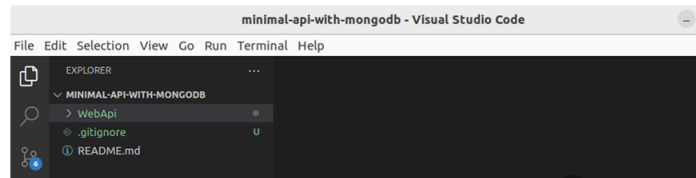
Erstellen Sie ein .NET Projekt WebApi mit Template web:

```
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
```

## Projektstruktur



**Abbildung 1:** Projektstruktur

Öffnen Sie in VSCode ein Terminal (Terminal → NewTerminal)

Navigieren Sie in den Folder WebApi und starten Sie die Anwendung:

```
cd WebApi
```

```
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.

```
{
  "$schema": "http://json.schemastore.org/launchsettings.json",
  "profiles": {
    "http": {
      "commandName": "Project",
      "dotnetRunMessages": true,
      "launchBrowser": true,
      "applicationUrl": "http://localhost:5001",
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    }
  }
}
```

## 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="Nataliia Yurchenko"
WORKDIR /app
COPY --from=build-env /build/out .
ENV ASPNETCORE_URLS=http://*:5001
EXPOSE 5001
ENTRYPOINT ["dotnet", "WebApi.dll"]
```

```
docker build -t mini-web-image .
docker run -p 5001:5001 mini-web-image
```

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

---

## 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.

```
services:
  webapi:
    build: ./WebApi
    restart: always
    ports:
      - 5001:5001
```

## Commit und Push

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



mit linux ubuntu bestehen probleme mit autorisierung auf git mit cli → GitHub Credential Manager installieren

<https://medium.com/@techworldthink/setting-up-github-credential-manager-on-ubuntu-013210e02652>