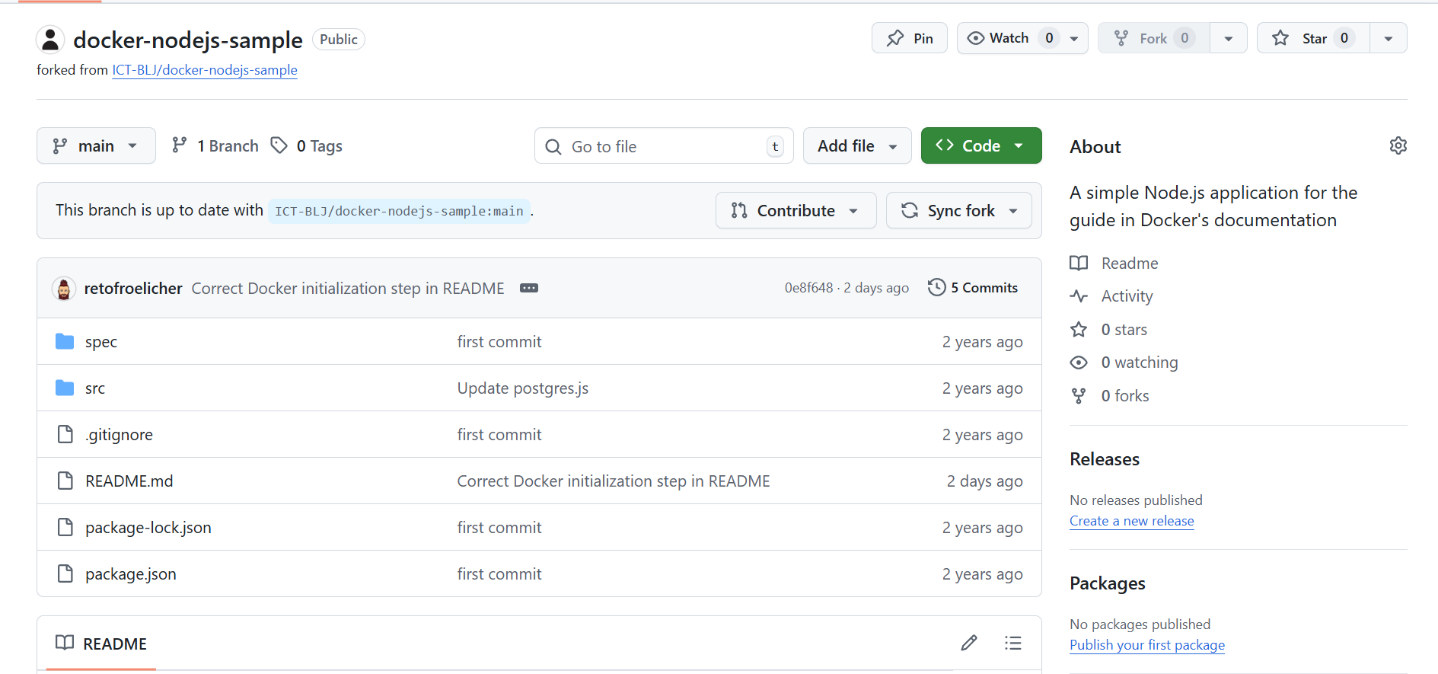
Abschlussaufgabe Development Fundamentals

# Klonen des Repositories

Damit ich das Sample auf meinem lokalen Rechner bearbeiten kann, musste ich auf den GitHub Repository von Reto und dann auf Fork drücken.

In meinem Files Explorer habe ich einen Ordner unter meinem ZLI Ordner erstellt. Darunter werde ich dann die Repo, denn ich geforked habe rüber clonen. Dafür musst ich in Git Bash folgende Befehle eingeben:

* Cd C:/ZLI/Projekte/Dev\_Fundamentals

Damit gehe ich in meinem Ordner, den ich für den Repo erstellt habe rein.

* git clone <https://github.com/esseymulugieta/docker-nodejs-sample>

Das cloned den Repo denn ich geforked habe in meinem Directory.

# README-Datei in Markdown

# Containerize your app

Um das Projekt in einem Docker Container lauffähig zu machen, musste ich folgende Schritte machen:

1. Powershell öffnen
2. Directory wechseln (cd Dateipfad des Sample)
3. Docker init eingegeben
4. Mein Terminal hat mir Fragen gestellt🤔. (Die Antworten habe ich per <https://docs.docker.com/guides/nodejs/containerize/#initialize-docker-assets> beantwortet.
5. Jetzt habe ich mein Container fertiggestellt und musste ihn nur noch runnen mit: docker compose up –build

# Datenbank einrichten

* Im Projektordner einen Unterordner db erstellen.
* Darin eine Datei password.txt anlegen und ein Passwort für die Datenbank eintragen.
* Das Passwort muss **in einer einzigen Zeile stehen**, keine Leerzeilen oder zusätzliche Zeichen.
* Hinweis: Bei der aktuellen npm-Version müssen die Daten in diesem Unterordner liegen, sonst funktioniert es nicht.

# Docker-Dateien anpassen

* docker-compose.yml so bearbeiten, dass die App und die Datenbank korrekt starten.
* Dockerfile in mehrere Abschnitte (Multi-Stage) aufteilen:
  + Entwicklungsbereich: installiert alle benötigten Pakete und startet die App mit npm run dev.
  + Produktionsbereich: nur die notwendigen Pakete, startet App mit node src/index.js.
* Bind-Mount einrichten, damit Änderungen am Code sofort im Container sichtbar sind.

# App testen

* Container neu starten:
* docker compose up --build
* Todo-Items hinzufügen, anzeigen und löschen, um sicherzugehen, dass alles funktioniert.
* Änderungen am Frontend testen, sie erscheinen direkt im Container.
* Daten bleiben auch nach Neustart des Containers erhalten.

# GitHub Workflow einrichten (optional)

* Repository auf GitHub erstellen.
* Persönlichen Access Token erzeugen und als Variable ins Repository einfügen.