

2024

Dokumentation der Vorgehensweise

Elion Baftiri

Klein Computer System AG

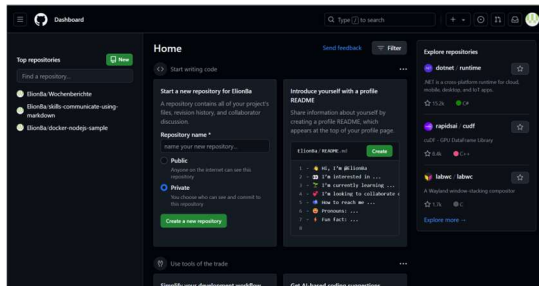
30.10.2024

Inhalt

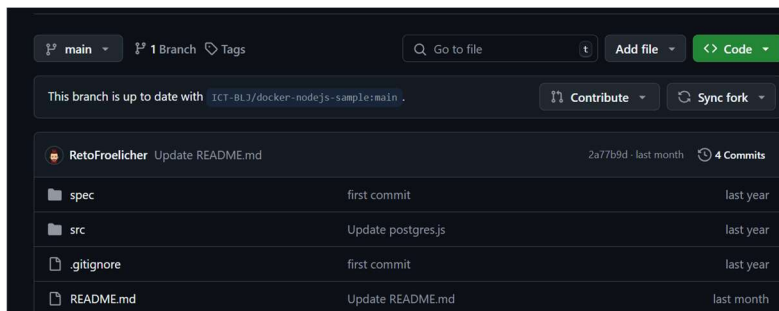
1	Fehler! Textmarke nicht definiert.
2	Fehler! Textmarke nicht definiert.
2	Fehler! Textmarke nicht definiert.
1	Fehler! Textmarke nicht definiert.
2	Fehler! Textmarke nicht definiert.
2	Fehler! Textmarke nicht definiert.
1	Fehler! Textmarke nicht definiert.
2	Fehler! Textmarke nicht definiert.
2	Fehler! Textmarke nicht definiert.
2	Fehler! Textmarke nicht definiert.
2	Fehler! Textmarke nicht definiert.
2	Fehler! Textmarke nicht definiert.
2	Fehler! Textmarke nicht definiert.

Klonen des Repositories

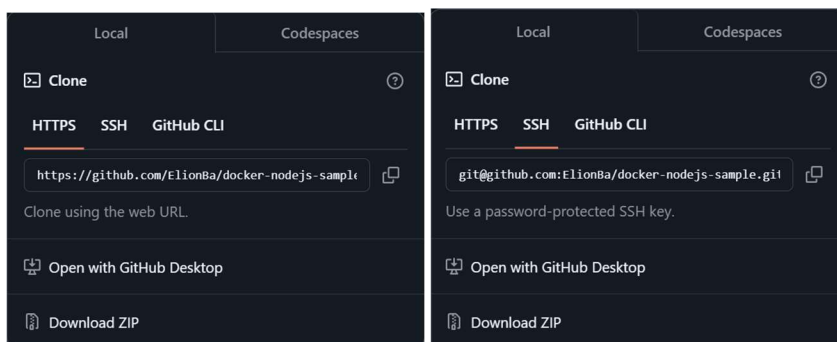
1. Öffne GitHub
2. Gehe auf die Startseite
3. Wähle eine Repository aus



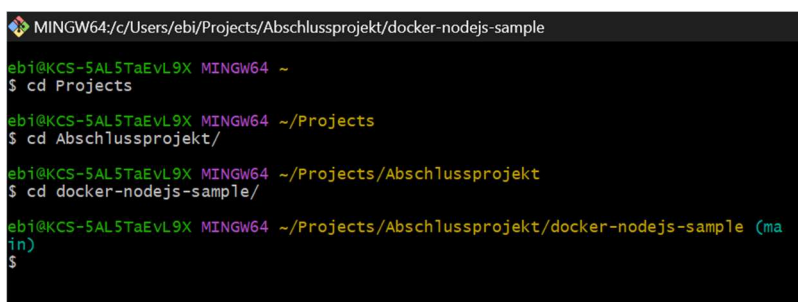
4. Klicke auf den grünen "Code"-Button:



5. Entscheide ob HTTPS oder SSH und kopiere den Link



6. Öffne das Terminal oder die Git-Bash
7. Wechsle mit cd in den Ordner wo du es klonen möchtest



8. Schreibe git clone (HTTPS oder SSH LINK Einfügen)

9. Jetzt wurde es geclont

Einrichtung der Entwicklungsumgebung

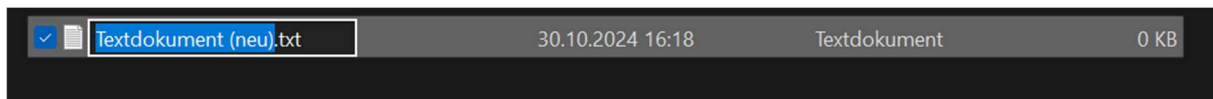
Um mit allem zu beginnen, muss man Git und Docker installieren

Git= <https://git-scm.com/downloads/win>

Docker= <https://docs.docker.com/desktop/install/windows-install/>

Erstellung der README.md

1. Gehe zum Explorer
2. mache einen Rechtsklick
3. gehe auf Neu
4. erstelle ein neues Textdokument



5. bennene das Textdokument als README und ändere nach dem Punkt das txt zu md
6. Schon ist es eine README.md

Verwendung von Git (Commit, Push)

1. Man öffnet die Git bash app
2. Man sucht mit dem cd command das gewünschte Verzeichnis
3. Wenn man in dem gewünschten Verzeichnis drin ist, dann überprüft man den Status mit git status

```
eb19KCS-5AL5TaEvL9X MINGW64 ~/Projects/Abschlussprojekt/docker-nodejs-sample (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .dockerignore
    Dockerfile
    README.Docker.md
    markdown-cheat-sheet.md

no changes added to commit (use "git add" and/or "git commit -a")
eb19KCS-5AL5TaEvL9X MINGW64 ~/Projects/Abschlussprojekt/docker-nodejs-sample (main)
$ |
```

4. Die Änderungen müssen hinzugefügt werden mit dem Command git add .

```
ebi@KCS-5AL5TaEvL9X MINGW64 ~/Projects/Abschlussprojekt/docker-nodejs-sample (main)
$ git add .
warning: in the working copy of '.dockerignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Dockerfile', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'README.Docker.md', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'markdown-cheat-sheet.md', LF will be replaced by CRLF the next time Git touches it

ebi@KCS-5AL5TaEvL9X MINGW64 ~/Projects/Abschlussprojekt/docker-nodejs-sample (main)
$ |
```

5. Jetzt muss man mit dem Befehl git commit -m "beispiel name" die Änderungen commiten

```
ebi@KCS-5AL5TaEvL9X MINGW64 ~/Projects/Abschlussprojekt/docker-nodejs-sample (main)
$ git commit -m "Eingefügt"
[main e4a51c5] Eingefügt
6 files changed, 275 insertions(+), 48 deletions(-)
create mode 100644 .dockerignore
create mode 100644 Abschlussprojekt-Dokumentation.pdf
create mode 100644 Dockerfile
create mode 100644 README.Docker.md
create mode 100644 markdown-cheat-sheet.md
```

6. Am Schluss muss man mit dem git push befehl alles auf GitHub pushen

Erstellung und Nutzung von Docker-Containern

1. Man öffnet git bash
2. Man sucht mit dem cd command das gewünschte Verzeichnis
3. Man macht den Befehl docker init und beantwortet alle Fragen die da gestellt wurden wie ich hier im beispiel

```
ebi@KCS-5AL5TaEvL9X MINGW64 ~/Projects/Abschlussprojekt/docker-nodejs-sample (main)
$ docker init

Welcome to the Docker Init CLI!

This utility will walk you through creating the following files with sensible defaults for your project:
- .dockerignore
- Dockerfile
- compose.yaml
- README.Docker.md

Let's get started!

Warning - The following Docker files already exist in this directory:
- .dockerignore
- Dockerfile
- compose.yaml
- README.Docker.md
? Do you want to overwrite them? Yes
? What application platform does your project use? Node
? What version of Node do you want to use? 18.0.0
? What version of Node do you want to use? 18.0.0
? Which package manager do you want to use? npm
? What command do you want to use to start the app? [tab for suggestions] (node index.js)
? What command do you want to use to start the app? node index.js
? What port does your server listen on? 3000
? What port does your server listen on? 3000

Created - .dockerignore
Created - Dockerfile
Created - compose.yaml
Created - README.Docker.md

Your Docker files are ready!
Review your Docker files and tailor them to your application.
Consult README.Docker.md for information about using the generated files.

What's next?
Start your application by running - docker compose up --build
Your application will be available at http://localhost:3000
```

4. Am schluss muss man den Container mit dem Befehl `docker compose up --build` starten
5. Der Container sollte gehostet sein um das zu testen muss man auf der Website `localhost:3000` eingeben