

Anleitung Client/Server/Datenbank:

Damit eine externe Person die Anwendung „Firework-Editor“ bei sich installieren kann, um daran weiter zu feilen sind ein paar Schritte nötig, die ich unten auflisten werden. Beginnen wir einmal aus der Sicht des Clients.

Client:

- Als globale Variable muss die Serveradresse eingefügt werden, auf dem die Anwendung laufen soll
→ vgl. `let url: string = "https://fireworkseditor.herokuapp.com";`
- Diese kann ganz einfach über Heroku selbst erstellt werden. Dafür muss im rechten, oberen Bildschirmrand auf „New / Create new app“ geklickt werden. Schon öffnen sich erste Angaben Felder, wie App-Name, Region, etc. Sind alle Schritte abgeschlossen, kann links oben über „Personal“ auf die App zugegriffen werden.
- Steht Client seitig alles soweit, kann der Server über „Deploy“ und dann ganz unten „Deploy branch“, gestartet werden (natürlich erst wenn Server-seitig alles steht). Über „view app“ findet man dann den Link (url)
- Folglich wären die Konsolenausgaben ganz einfach rechts oben über „More / View logs“ zu finden

Server:

- Was nicht vergessen werden darf, ist die `package.json` zu erneuern. Das heißt die Codezeile in der auf die `Server.js` Datei verwiesen wird und letztlich also den Pfad, wo diese auf dem Rechner liegt, widerspiegelt
→ vgl. `"start": "node ENDABGABE/Server/server.js"`
- Bevor dann ein namespace aufgebaut werden kann, müssen Erweiterungen eingebaut werden, die bereits bestimmte Funktionen zur Vereinfachung mit sich bringen. Diese stehen ganz oben (die 3. Codezeile benötigen wir erst für die Datenbank) → vgl.

```
import * as Http from "http";  
import * as Url from "url";  
import * as Mongo from "mongodb";
```

- Damit der Server seine gewünschten Aufgaben nach sich ziehen kann, brauchen wir eine Funktion, die den Server startet
→ vgl. `function startServer(): void`
- Darin wird der Server erstellt. Falls der port nicht explizit festgelegt wird, steht hier der port 5001 zur Verfügung, worauf der Server im Anschluss „horcht“.
- Zu guter Letzt wird ein request-Listener auf den Server installiert, welcher eine weitere Funktion aufruft (hier: `handleRequest`)

Datenbank mit MongoDB:

- Nach erfolgreicher Registrierung muss rechts oben über „New project“ ein Projekt angelegt werden. Dieses setzt sich dann über „Create a new Cluster“ aus einem Datenbank-, und einem Collection-Namen zusammen
- Code: Ist dies erledigt, würde ich die `Mongo.Collection` in einer globalen Variablen speichern → vgl. `let rocket: Mongo.Collection;`
- Direkt darunter brauchen wir wie auch bereits Server seitig, die url zur Datenbank → vgl.

```
let databaseUrl: string = "mongodb+srv://helenareimold:hallo@cluster0.eivgu.mongodb.net/fireworks?retryWrites=true&w=majority"
```

(!Wichtig: <passwort> muss mit dem festgelegten Passwort ersetzt werden (hier: hallo), <dbname> ebenso (hier: fireworks))

- Diesen Link findet man über das Projekt, unter dem Punkt „Connect“ → „connect you application“
- Im letzten Schritt wird diese databaseUrl dann der Funktion, welche die Verbindung zur Datenbank herstellen soll, mitgegeben
→ vgl. `async function connectToDatabase(_url: string): Promise<void>`