**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 werde. Beginnen wir einmal aus der Sicht des Clients.

**Client (main.ts):**

* Als globale Variable muss die Serveradresse gegen die eigene Heroku-Adresse ausgetauscht werden, an die der Client seine Anfragen versenden kann  
  🡪 vgl. let url: string = "https://fireworkseditor.herokuapp.com";

**Server (server.ts):**

* Was nicht vergessen werden darf, ist die package.json zu erneuern. Das heißt die Codezeile in der auf die Server.js Datei verwiesen wird, um diese dann auch über node auszuführen  
  🡪 vgl. "start": "node ENDABGABE/Server/server.js"
* Für die Verbindung und für weitere Operationen mit dem Server und der Datenbank, werden Erweiterungen wie Http, Url und Mongo importiert, 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";

**Datenbank mit MongoDB:**

* Wie auch bereits für die Verbindung mit Heroku, benötigt man auch für die Verbindung mit Mongo-Atlas einen Connection-string 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 muss in der Funktion connectToDatabase die Mongo.Collection (hier in der Variable rocket) angepasst werden. In unserem Fall muss dafür der Name der Datenbank und der Collection mit den eigenen Bezeichnungen ausgetauscht werden   
  🡪 vgl. rocket = mongoClient.db("fireworks").collection("rockets");