EIA Dokumentation zur Einrichtung

Node js.

Mit Node js. startest du einen lokalen Server. Installiere auf der obersten Ebene deines Repository die neusten Datentypen von Node. Öffne dazu in VSCode ein Terminal und gib den Befehl **npm install @types/node** ein. Es sollte ein Ordner node_modules und eine Datei package.lock.json entstanden sein. Die package.lock.json kannst du löschen. Der Ordner node_modules ist wichtig, denn er beinhaltet die Definitionsdateien für TypeScript. Aufgrund der großen Kapazität des Ordners solltest du es nicht auf dein Github-Repository pushen. Lege stattdessen ein „File“ in deinem Repository mit dem Namen *.gitignore* an. Darin schreibst du einfach *node_modules* und Git wird sich nicht mehr um diesen Ordner kümmern. Um ein Skript mit Node laufen zu lassen musst du in deinem Terminal einfach **node NameDesSkripts.js** eingeben. *NameDesSkripts* steht für den Namen deines tatsächlichen Skripts, das du laufen lassen möchtest.

Heroku Server

Damit du deinen eigenen Heroku-Server nutzen kannst, gehe in den Ordner Main und dort in „Main.ts“. Dort kannst du in Zeile 3, deinen Heroku http-Pfad einfügen.

Mongo Datenbank

Damit du deine Daten an die Datenbanksoftware schicken kannst, musst du Node zunächst um die entsprechenden Module für die Arbeit mit MongoDB erweitern.

Dazu führst du folgende Befehle auf der obersten Ebene deines Projektes aus.

- npm install @types/mongodb
- npm install mongodb

Die Datei package.json sollte nun neue Einträge unter *dependencies* aufweisen, die auf die Module und Versionen von MongoDB verweisen. Damit weiß dann auch z.B. Heroku, mit welchen Modulen das Projekt arbeitet.

Online Service Mongo Atlas

Lege ein Cluster in Mongo Atlas an. Dort kannst du dir auch eine Datenbank und eine Collection anlegen. Gehe im Code in den Ordner „Server“ und öffne dort die Datei „Server.ts“. In Zeile 19 kannst du deine eigenen Connection-String zu deiner Datenbank einfügen.

Viel Spaß bei der Nutzung!