

Übungsblatt 13: Express

Vorstellung in den Tutorien am 7. - 9. Februar 2023

13.1 Express: Flugsuchmaschine 2

Letzte Woche haben wir mithilfe von nginx und Node.js eine Flugsuchmaschine entwickelt, die Suchanfragen von Clients an mehrere Anbieter weiterleitet und die Ergebnisse aggregiert an die Clients übergibt.

Die Geschäftsleitung von *AxelAirways*, einem der o.g. Anbieter, war so begeistert von unserer Flugsuchmaschine, dass sie uns darum gebeten haben, eine vergleichbare Anwendung speziell für ihre Airline zu entwickeln. Aus technischen Gründen soll diese Anwendung allerdings ausschließlich mit Express, einem Web Framework für Node.js, entwickelt werden.

Für diese Aufgabe benötigen Sie zwei Node-Module: `express` und `cookie-parser`. Falls Sie diese Module noch nicht installiert haben, können Sie dies mit folgenden Befehlen tun:

```
npm install express
```

```
npm install cookie-parser
```

In der Vorgabe finden Sie eine fertig implementierte *index.html*, eine kleine Flugdatenbank (*flights/db.js*), sowie zwei unvollständige .js-Dateien (*server.js* und *flights/controller.js*), in denen Sie die unten beschriebene Funktionalität als Express-Middleware-Funktionen implementieren sollen. Ausführen können Sie die Anwendung mit dem Befehl `node server.js`.

1. Sämtliche HTTP-Anfragen werden mittels einer Logger-Middleware auf der Konsole ausgegeben.
2. Ruft ein Nutzer den Standardpfad / der Anwendung auf, wird die vorgefertigte *index.html* zurückgegeben.
3. Bei GET-Anfragen an den Pfad /*flights* wird ein Array mit sämtlichen Flügen der Airline zurückgegeben.
4. Bei GET-Anfragen an den Pfad /*flights/flightnumber/*<flightnumber> wird der Flug mit der entsprechenden Flugnummer zurückgegeben (als ein Array mit nur einem Eintrag).
5. Bei GET-Anfragen an den Pfad /*flights/*<departure> werden alle Flüge mit dem angegebenen Startflughafen zurückgegeben. Diesen Anfragen kann der Query-Parameter *destination* angehängt werden, um auch einen Zielflughafen zu spezifizieren.
6. Falls keine Flüge für eine Suchanfrage gefunden werden, wird der HTTP-Statuscode 404 zurückgegeben.
7. Die IT-Abteilung von *AxelAirways* bittet Sie außerdem darum, für Debugging-Zwecke einen PATCH-Endpoint unter dem Pfad /*flights* einzurichten, der im Request-Body einen im Format `x-www-urlencoded` kodierten Flug akzeptiert, dessen Flugnummer in der Konsole ausgibt und anschließend den HTTP-Statuscode 418 zurückgibt.

Die Marketing-Abteilung von *AxelAirways* hat durch Nutzerstudien herausgefunden, dass Nutzer, die öfter nach Flügen suchen, höheres Interesse an den Flügen haben und daher bereit sind, mehr zu zahlen. Die Geschäftsleitung möchte daher einen Mechanismus in die Anwendung einbauen, der die Flugpreise erhöht, je öfter ein Nutzer Suchanfragen sendet.

8. Implementieren Sie diese diabolische Funktionalität mithilfe von Cookies, mit denen die Suchanfragen von Nutzern gezählt werden! Für jede weitere Anfrage nach der ersten sollen die übergebenen Flugpreise um 10€ erhöht werden.