

Web-Entwicklung: Übungsblatt 4

In den bisherigen Übungen haben Sie mit Mock-Datensätzen gearbeitet. In dieser Übung sollen Sie zum ersten Mal eine echte externe API, nämlich [SWAPI - The Star Wars API](#), ansprechen.

Aufgabe 1: API-Dokumentation

Setzen Sie sich zunächst mit der [Dokumentation der API](#) auseinander.

Hinweis: Üblicherweise finden Sie für andere APIs oder npm-Module ähnliche Dokumentation. Die Qualität dieser Dokumentationen ist sehr unterschiedlich und zumindest ein Indikator für die Reife eines Moduls. Achten Sie also bei der Auswahl von Modulen für Ihre eigenen Projekte auch darauf.

Aufgabe 2: Asynchrone Kommunikation via Callbacks

Die asynchrone Kommunikation mit der HTTP-API setzt einen HTTP-Client voraus. Das Node.js-Standardmodul `http` bietet entsprechende Funktionalitäten an. Dabei kommt - wie in den meisten npm-Modulen - das Callback-Entwurfsmuster zum Einsatz.

Erstellen Sie eine `GET`-Anfrage für eine SWAPI-Ressource Ihrer Wahl und inspizieren Sie von der API zurückgelieferte Objekt. In einem Callback soll der Body der HTTP-Antwort ausgegeben werden.

Testen Sie auch, ob Ihr Callback sinnvoll auf Fehler reagiert. Dazu können Sie den Dienst `httpstatus` einsetzen, der ebenfalls über eine HTTP-API HTTP-Antworten mit dem gewünschten HTTP-Fehlercode ausliefert. Eine weitere Fehlerart, die Sie behandeln sollten, sind Anfragen an gar nicht existierende URLs.

Hinweis: Gerne dürfen Sie die in der letzten Vorlesung kennen gelernten ES6-Sprachkonstrukte einsetzen, z.B. die Schlüsselwörter `let` und `const`, Template Strings oder Arrow Functions.

Aufgabe 3: Asynchrone Kommunikation via Promises

Wie Sie in der Vorlesung gesehen haben, ist Promise ein ES6-Standardobjekt zur syntaktischen Vereinfachung asynchroner Funktionalitäten.

Wandeln Sie Ihre Lösung von Aufgabe 2 so um, dass die asynchrone Funktionalität Ihres Callbacks von einem Promise-Objekt übernommen wird.