

WBA2 SoSe 2017

Kickoff Aufgabe

Grundlegende Informationen

Node.js hat das Modul **fs** integriert, welches den Zugriff auf Dateien ermöglicht.

Es wird wie andere Module mittels **require()** eingebunden:

```
var fs = require('fs');
```

Die Funktion

```
fs.readFile(__dirname+"/dateiname", function(err, data) { ... });
```

ermöglicht das asynchrone Auslesen von Dateien.

Die Variable **__dirname** enthält den Namen des Verzeichnisses, in dem das aktuelle Programm liegt. Er wird in diesem Beispiel vor den Pfad angehängt, um eine vollständige Pfadangabe zu vermeiden. Der Parameter **data** ist normalerweise ein Buffer mit Binärdaten, der mittels **data.toString()** in einen String umgewandelt werden kann.

Die Funktion

```
fs.writeFile(__dirname+"dateiname", data, function(err) { ... });
```

ermöglicht das asynchrone Schreiben von Dateien.

Weitere Informationen zum fs-Modul von node.js finden Sie in der [node.js-Dokumentation](#).

Mittels **JSON.stringify(...)** können Objekte in einen JSON-String umgewandelt werden und mittels **JSON.parse(...)** können JSON-Strings in ein Objekt umgewandelt werden.

Wichtig: Aus Ihrer Git-Historie sollte hervorgehen, dass die Aufgaben in der hier vorgegebenen Reihenfolge gelöst wurden. Erstellen Sie also pro Aufgabe mindestens einen separaten Commit.

Aufgabe 1

Schreiben Sie ein Programm in Node.js, das die Datei `staedte.json` ausliest und die darin enthaltene Liste von Städten in folgender Form auf der Konsole ausgibt:

```
name: Baoding, Hebei,  
country: China,  
population: 110.550.000  
-----  
name: Shanghai  
country: China  
population: 14.010.000  
-----  
...
```

Verwenden Sie zum Auslesen der Datei die oben genannten **asynchronen** Funktionen des `fs`-Moduls.

Aufgabe 2

Erweitern Sie das Programm aus Aufgabe 1 so, dass die Namen, Länder und Einwohnerzahlen jeweils **farbig** ausgegeben werden. Hierzu können Sie das Modul **chalk** verwenden, welches auch im Workshop vorgestellt wurde. Mehr zum Chalk-Modul finden Sie [hier](#). Erstellen Sie zu ihrem Programm auch eine **package.json**-Datei, welche unter anderem die Modulabhängigkeiten definiert. Weitere Informationen zur **package.json**-Datei finden Sie in der [NPM-Dokumentation](#).

Aufgabe 3

Erweitern Sie das Programm aus Aufgabe 2 so, dass es die Städte nach Einwohnerzahlen sortiert, die sortierte Liste in der neuen Datei **staedte_sortiert.json** speichert und sie erst danach auf der Konsole ausgibt.

Informationen zum Sortieren von Arrays in Javascript finden Sie beispielsweise [hier](#).