Entwicklung von Back-End-Apps mit Node.js und Express

Modul 1 Spickzettel: Einführung in serverseitiges JavaScript

Paket/Methode	Beschreibung	Codebeispiel

http/createServer	Das http-Paket wird verwendet, um remote Verbindungen zu einem Server herzustellen oder um einen Server zu erstellen, der auf Anfragen von Clients hört.
	
	createServer - Nimmt einen requestListener, eine Funktion, die die Parameter request und response entgegennimmt, wobei request der Handle für die Anfrage vom Client und response der Handle ist, der an den Client gesendet wird.
		`const http = require('http');`
		`const requestListener = function(req, res) {`
		`  res.writeHead(200);`
		`  res.end('Hallo, Welt!');`
		`}`
		`const port = 8080;`
		`const server = http.createServer(requestListener);`
		`console.log('Server hört auf Port: '+ port);`
		`server.listen(port);`

new Date()	Die Methode new Date() gibt das aktuelle Datum als Objekt zurück. Sie können Methoden auf dem Datumsobjekt aufrufen, um es zu formatieren oder die Zeitzone zu ändern.	`module.exports.getDate = function getDate() {`
		`  let aestTime = new Date().toLocaleString("de-DE", {timeZone: "Australien/Brisbane"});`
		`  return aestTime;`
		`}`

import()	Die Import-Anweisung wird verwendet, um Module zu importieren, die von einem anderen Modul exportiert wurden. Eine Datei, die wiederverwendbaren Code enthält, wird als Modul bezeichnet.	`// addTwoNos.mjs`
		`function addTwo(num) {`
		`  return num + 4;`
		`}`
		`export { addTwo };`
		`// app.js`
		`import { addTwo } from './addTwoNos.mjs';`
		`// Gibt aus: 8`
		`console.log(addTwo(4));`

require()	Die eingebaute NodeJS-Methode require() wird verwendet, um externe Module zu integrieren, die in verschiedenen Dateien enthalten sind. Die require()-Anweisung liest und führt im Wesentlichen eine JavaScript-Datei aus, bevor sie das Exportobjekt zurückgibt.	`module.exports = 'Hallo Programmierer';`
		`let msg = require('./messages.js');`
		`console.log(msg);`


Skills Network