HTTP, JavaScript & Backend – Zusammenfassung

Stand: 27.01.2026

HTTP & Kommunikation im Web

Im Web kommunizieren Browser und Server über HTTP. Der Browser schickt eine Anfrage (Request) und der Server antwortet mit Daten (Response). Dabei sind HTTP-Methoden, Statuscodes und JSON-Daten zentrale Bestandteile. Diese Kommunikation bildet die Grundlage moderner Web-Anwendungen.

Fetch API & REST APIs

Mit der Fetch API kann JavaScript Daten von Servern abrufen oder an sie senden. REST APIs stellen strukturierte Endpunkte bereit, über die Daten gelesen oder verändert werden können. Die Antworten werden meist im JSON-Format verarbeitet.

Callback-Funktionen

Callbacks sind Funktionen, die an andere Funktionen übergeben und später ausgeführt werden. Sie werden häufig bei zeitaufwendigen Aufgaben verwendet, können den Code aber bei starker Verschachtelung unübersichtlich machen.

Promises

Promises beschreiben einen Wert, der in der Zukunft verfügbar ist. Sie helfen dabei, asynchronen Code übersichtlich zu strukturieren und ersetzen viele verschachtelte Callbacks.

Async / Await

Async und Await bauen auf Promises auf und ermöglichen es, asynchronen Code fast wie synchronen Code zu schreiben. Dadurch wird der Ablauf verständlicher und Fehler lassen sich klar mit try/catch behandeln.

Node.js & Express.js

Node.js erlaubt die Ausführung von JavaScript auf dem Server. Mit Express.js lassen sich Server und REST APIs einfach umsetzen, indem Routen definiert und JSON-Daten bereitgestellt werden.

Gesamtzusammenfassung: Die Inhalte zeigen, wie Web-Anwendungen über HTTP kommunizieren und wie JavaScript asynchron arbeitet. Aufbauend darauf lassen sich mit Node.js und Express.js einfache Backends und APIs erstellen.