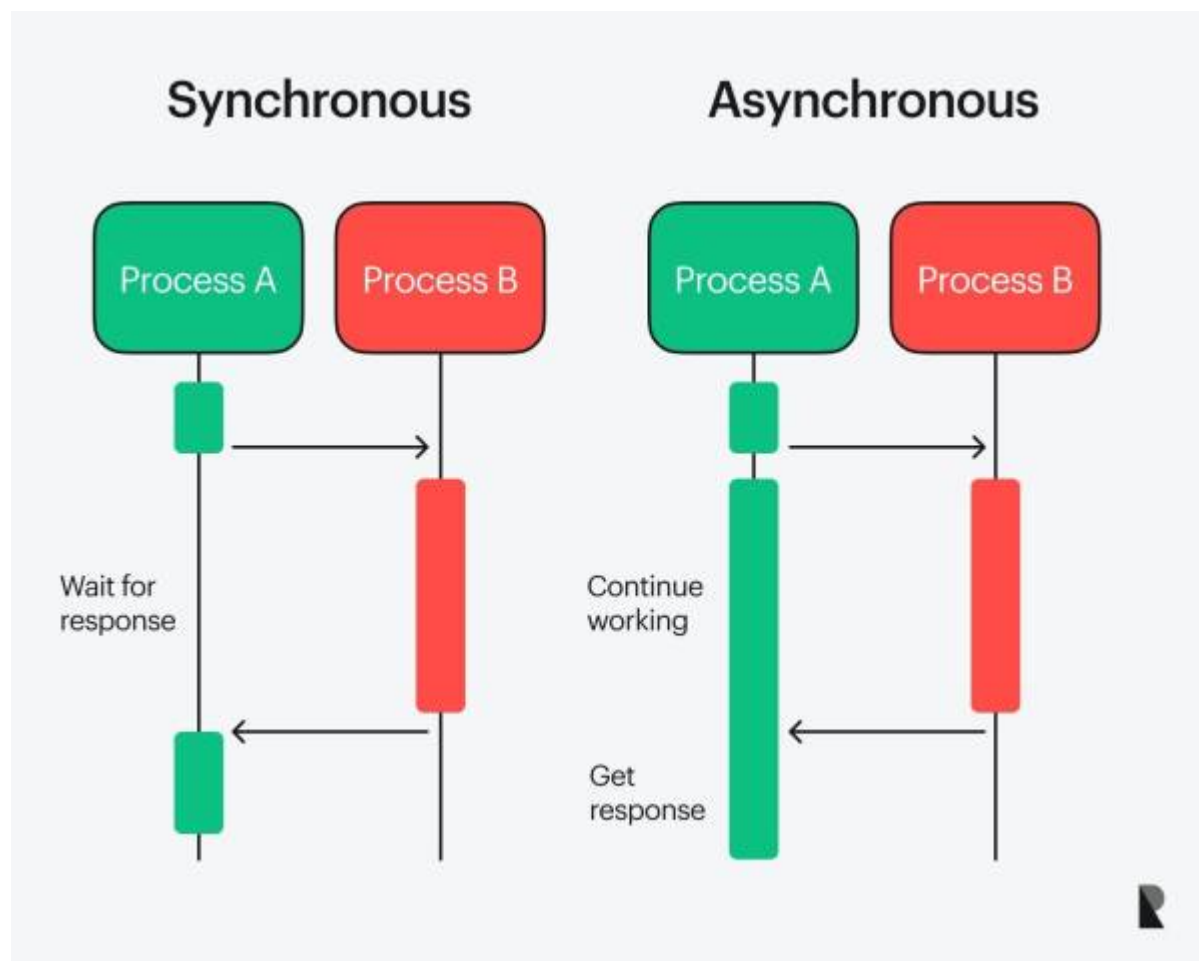


LU01b - Synchrone VS Asynchrone Verarbeitung

Grundsätzlich gibt es zwei Varianten in JS, wie Daten eingelesen werden können:

- Synchron
- Asynchron

Beim synchronen Einlesen wartet JavaScript, bis die Datei komplett gelesen wurde - und macht in der Zeit gar nichts anderes. Beim asynchronen Einlesen liest Node.js die Datei im Hintergrund ein und kann waehrenddessen weiterarbeiten. Das eine blockiert, das andere nicht.



Vergleich Sync VS Async

Die nachfolgende Tabelle zeigt die wesentlichen Unterschiede

Art	Verhalten	Vorteil	Nachteil
synchron (<code>readFileSync`</code>)	Programm pausiert, bis Datei fertig eingelesen ist	einfach, gut fuers schnelle Rumprobieren	blockiert den gesamten Ablauf - bei grossen Dateien oder vielen Zugriffen fatal

Art	Verhalten	Vorteil	Nachteil
asynchron (<code>readFile`</code>)	Datei wird im Hintergrund gelesen, Code kann weiterlaufen	performant, ideal fuer Server/mehrere Zugriffe	Rueckgabewert kommt erst spaeter per Callback/Promise

Beispiel für Synchrones Einlesen

```
const fs = require("fs");  
// Nachfolgend die Zeile für synchrones Einlesen des JSON-Files  
const daten = fs.readFileSync("data.json", "utf8");  
console.log("Weiter gehts erst, wenn das oben fertig ist.");
```

Beispiel für Asynchrones Einlesen

```
const fs = require("fs");  
// Nachfolgend die Zeile für Asynchrones einlesen des JSON-Files  
<color #ed1c24>fs.readFile("data.json", "utf8", (err, daten)</color> => {  
    console.log("Ich komme spaeter – Datei ist jetzt fertig eingelesen.");  
});  
console.log("Dieser Text erscheint direkt danach – ohne warten.");
```

Lernvideo

Lernvideo - 5': What is asynchronous javascript-code?	what_is_asynchronous_javascript_code.mp4
Lernvideo - 8': Asynchronous Vs Synchronous Programming	asynchronous_vs_synchronous_programming.mp4



Volkan Demir

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/de/modul/m307/learningunits/lu01/02?rev=1769668629>

Last update: **2026/01/29 07:37**

