

# LU02d - JSON-Datei verändern und speichern

Nachdem wir JSON-Dateien ins JS eingelesen und im Browser oder in der Konsole ausgeben konnte, wird es Zeit JSON-Inhalte zu verändern und diese wieder in JSON-Dateien zu schreiben. Auch hier benötigen wir unser *fs-Modul*.

## Geaenderte Daten wieder in eine JSON-Datei schreiben

Das nachfolgende Codebeispiel zeigt, wie Daten aus dem JS in eine lokale JSON-Datei geschrieben werden.

### Codebeispiel

```
const fs = require("fs");
//
// Daten aendern
daten.ist_aktiv = false;
daten.hobbys.push("Programmieren");
//
// Objekt wieder als JSON-Text schreiben
const output = JSON.stringify(daten, null, 4);
//
fs.writeFile("person_aktualisiert.json", output, "utf8", (err) => {
  if (err) {
    console.error("Fehler beim Schreiben der Datei:", err.message);
    return;
  }
  console.log("Datei 'person_aktualisiert.json' wurde gespeichert.");
});
```

## JSON im Browser einlesen (z. B. von einem Server)

Wenn Daten zwischen Client und Server oder zwischen Servern ausgetauscht werden müssen, sieht die Sache etwas anders aus. Hier müssen die Daten *HTTP ge-feched* (hergeholt) werden.

### Codebeispiel mit fetch (moderner)

```
fetch("person.json")
  .then((response) => {
    if (!response.ok) {
      throw new Error("HTTP-Fehler: " + response.status);
    }
    return response.json(); // parst JSON automatisch
  })
  .then((daten) => {
    console.log("JSON aus fetch:", daten);
  });
```

```
    console.log("Name:", daten.name);
  })
  .catch((error) => {
    console.error("Fehler beim Laden:", error.message);
  });
```

### Codebeispiel mit async/await (lesbarer)

```
async function ladePerson() {
  try {
    const response = await fetch("person.json");
    if (!response.ok) {
      throw new Error("HTTP-Fehler: " + response.status);
    }
    const daten = await response.json();
    console.log("Person:", daten);
  } catch (err) {
    console.error("Fehler:", err.message);
  }
}
ladePerson();
```

## Fehlerquellen und Handling

Typische Probleme:

- Datei existiert nicht / falscher Pfad (Node.js: ENOENT)
- JSON ist syntaktisch kaputt (fehlendes Komma, falsche Anführungszeichen)
- Im Browser: CORS-Probleme oder HTTP-Fehler

### Test-Script für Node.js

```
const fs = require("fs");
//
try {
  const text = fs.readFileSync("person.json", "utf8");
  const daten = JSON.parse(text);
  console.log("Erfolgreich eingelesen:");
  console.log(JSON.stringify(daten, null, 4));
} catch (err) {
  if (err.code === "ENOENT") {
    console.error("Datei 'person.json' nicht gefunden.");
  } else if (err.name === "SyntaxError") {
    console.error("Fehler im JSON-Format:", err.message);
  } else {
    console.error("Unerwarteter Fehler:", err.message);
  }
}
```



Volkan Demir

From:

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

Permanent link:

<https://wiki.bzz.ch/de/modul/m307/learningunits/lu02/04>

Last update: **2026/02/09 13:41**

