

# Web-Entwicklung: Übungsblatt 2

---

In Web-Anwendungen kommen Sie oft in Kontakt mit externen [APIs](#). Diese stellen Ihnen auf Anfrage Daten zur Verfügung, welche Sie weiterverarbeiten oder darstellen können. Häufig kommt zur Kommunikation mit externen APIs das HTTP-Protokoll zum Einsatz. Wie Sie aus einer clientseitigen Web-Anwendung HTTP-Anfragen stellen und die dazugehörigen Antworten auswerten können, lernen Sie schon bald.

Beispielhaft wird Ihnen für diese Übungsaufgaben ein einfacher Datensatz (wie er z.B. auch im *body* einer HTTP-Antwort stehen könnte) als statische Datei `dataset.json` zur Verfügung gestellt.

Hinweis: Der Datensatz wurde mit dem Dienst [mockaroo - realistic data generator](#) erstellt.

## Aufgabe 1: Analyse des Datensatzes

Setzen Sie sich mit der o.g. Datei auseinander, um die Struktur des Datensatzes zu verstehen. Dies wird nötig sein, um die folgenden Aufgaben lösen zu können.

Dabei könnte auch das dazugehörige JSON-Schema (die Datei mit der Endung `.schema`) hilfreich sein. Dieses Dokument beschreibt auf einer Meta-Ebene, wie eine gültige JSON-Datei, die diesem Schema genügt, aufgebaut sein muss.

Hinweis: In Visual Studio Code wird beim Öffnen der JSON-Datei das dazugehörige JSON-Schema automatisch ausgewertet. Wenn Sie mit der Maus über ein Element der JSON-Datei fahren, wird die im Schema hinterlegte Erläuterung des Elements angezeigt.

## Aufgabe 2: Auswertung des Datensatzes

Um den Datensatz in eine JavaScript-Anwendung zu importieren, können Sie den Aufruf der globalen Methode `require('./dataset.json')` verwenden. Liegen beide Dateien nicht im gleichen Verzeichnis, muss der Pfad entsprechend angepasst werden. `require()` liefert Ihnen den Inhalt von JSON-Dateien als JavaScript-Objekt zurück (siehe [Node.js-Dokumentation: require\(\)](#)).

Versuchen Sie nun, mithilfe von JavaScript die Antworten auf die folgenden Fragen zum Datensatz zu ermitteln und auszugeben:

1. Wie viele Einträge hat der Datensatz?
2. Wieviele *Subscriptions* von Frauen sind im Datensatz enthalten?
3. Wieviele aktive Nutzer sind insgesamt erfasst?
4. Wieviele derzeit aktive *Subscriptions* sind seit dem 01.01.2017 hinzugekommen?
5. Wieviele *Subscriptions* wurden bis zum 01.01.2017 (exklusive) wieder gekündigt?
6. **(Bonus)** Welche Themen (**subscribed**) haben die noch aktiven Nutzer mit dem frühesten Erstellungsdatum im Datensatz?

Hinweise:

- [MDN: Date](#)
  - Es wird empfohlen, die Variante des Date-Konstruktors mit der expliziten Angabe der Datumskomponenten Jahr, Monat und Tag zu verwenden.
  - Date-Objekte kann man vergleichen (`<`, `>` usw.), addieren sowie substrahieren.
- Die Methode `Object.keys()` liefert die Schlüssel eines assoziativen Arrays als Array zurück.

## Aufgabe 3: Elemente des Datensatzes löschen bzw. entfernen

Beobachten Sie im Debugger den Effekt der folgenden Operationen auf die Elemente des `data`-Arrays. Lassen Sie die Array-Länge stets vor und nach einer Operation ausgeben.

### a) Löschen

Nutzen Sie den `delete`-Operator, um ein beliebiges Element zu löschen.

### b) Entfernen

Entfernen Sie jeweils ein Element vom Anfang, aus der Mitte und vom Ende des Datensatzes mithilfe dazu geeigneter Array-Methoden.

Was ist der wesentliche Unterschied zwischen dem Löschen von Array-Elementen mithilfe von `delete` und dem Entfernen?

## Bonus-Aufgabe 4: Korrektur eines fehlerhaften Datensatzes

In der Datei `dataset-unfixed.json` finden Sie einen weiteren Datensatz. Dieser Datensatz hat fehlerhafte Einträge: Das Datum `unsubscribed_at` ist bei einigen Einträgen kleiner als das Datum `created_at`, d.h. die *Subscription* wurde scheinbar gekündigt bevor sie überhaupt erstellt wurde.

Dieser Fehler soll behoben werden, indem das Datum `unsubscribed_at` auf den Tag nach `created_at` gesetzt wird. Der korrigierte Datensatz soll in eine neue Datei geschrieben werden.

Hinweise:

- Mit der Date-Methode `getTime()` können Sie sich die Zeit in Millisekunden seit dem 01.01.1970 ausgeben lassen. Mit `setTime()` kann man diese wiederum setzen.
- Die Methode `JSON.stringify()` wandelt ein als Argument übergebenes Objekt in einen String um.
- Zur Umleitung der Ausgaben in eine Datei rufen Sie ihr Programm wie folgt auf: `node FixDataset.js > ./dataset-fixed.json`
- Testen Sie die vorherigen Aufgaben mit dem korrigierten Datensatz. Die Ergebnisse sollten gleich bleiben.