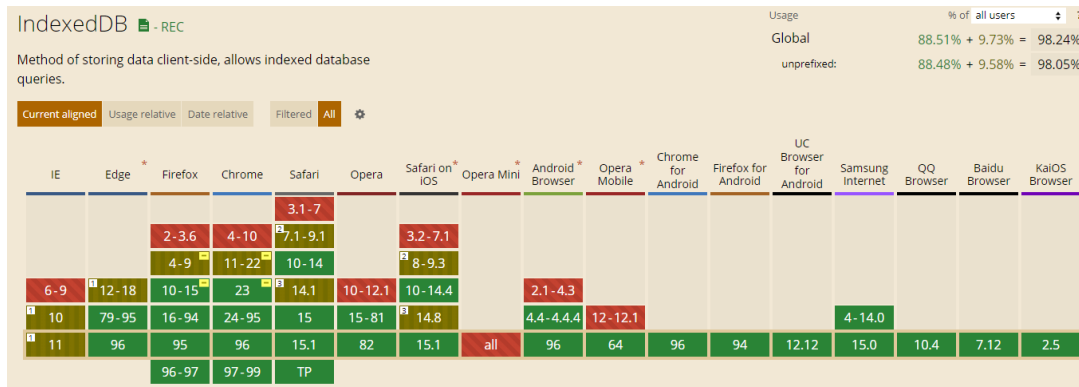


Robert Baumgartner

In einem der vorhergehenden Arbeitsblätter haben wir bei fehlender Internetverbindung den **Delete** Button disabled oder versteckt. Großartig wäre es aber, wenn auch im offline Fall CRUD Operationen möglich wären. Dazu müssten wir die Änderungen der CRUD Operationen lokal speichern und sie dann später, wenn der User wieder online ist, mit dem Server synchronisieren.

Wir benötigen also die Möglichkeit lokal Daten zu lesen und zu speichern. Hier bietet sich **IndexedDB** an, eine NoSQL Datenbank, die in jedem gängigen Browser vorhanden ist.



Aufgabe 1: Arbeite das Arbeitsblatt **Speichern von Daten am Client-IndexedDB.pdf** durch.

Behandlung von Offline CRUD Operationen

Im Folgenden wollen wir das Löschen von Daten bei fehlender Internetverbindung ermöglichen. Wenn du möchtest, kannst du auch die anderen CRUD Operationen als zusätzliche Übung einbauen.

Aufgabe 2: Zeige den **Delete** Button auch im offline Fall an, implementiere - falls nicht schon erfolgt - das Löschen am Server und baue das Caching der **employees** Objekte aus dem Service Worker aus. Die Bilder soll Workbox weiterhin cachen.

```
if (workbox) {
  console.log(`Workbox is loaded`);
  workbox.setConfig({ debug: true });
  workbox.precaching.precacheAndRoute(self.__precacheManifest);

  // workbox.routing.registerRoute(
  //   '/employees',
  //   new workbox.strategies.NetworkFirst({
  //     cacheName: 'roberts-employee-cache',
  //   }),
  // );

  workbox.routing.registerRoute(
    new RegExp('/images/.*.jpg'),
    new workbox.strategies.StaleWhileRevalidate({
      cacheName: 'roberts-image-cache',
    }),
  );
} else {
  console.log(`Workbox didn't load`);
}
```

Robert Baumgartner

Wir werden die **employee** Objekte selbst in der Datenbank **employeesDB** und dem Store **employees** cachen! Füge dazu im **created** Hook den entsprechenden Code ein, um die DB anzulegen.

Aufgabe 3: Holen der Daten vom Server. Der Trick dabei ist das Hinzufügen eines Attributes zu jedem Employee Objekt. Das Attribut **isDeleted**, zeigt an, ob das Objekt lokal gelöscht wurde.

In der Methode **fetchData**,

wenn der User **online** ist, füge beim Holen der Daten das Attribut hinzu:

```
let { data } = await axios.get(`${this.serverAddress}/employees`);
data = data.map((el) => ({ ...el, isDeleted: false }));
```

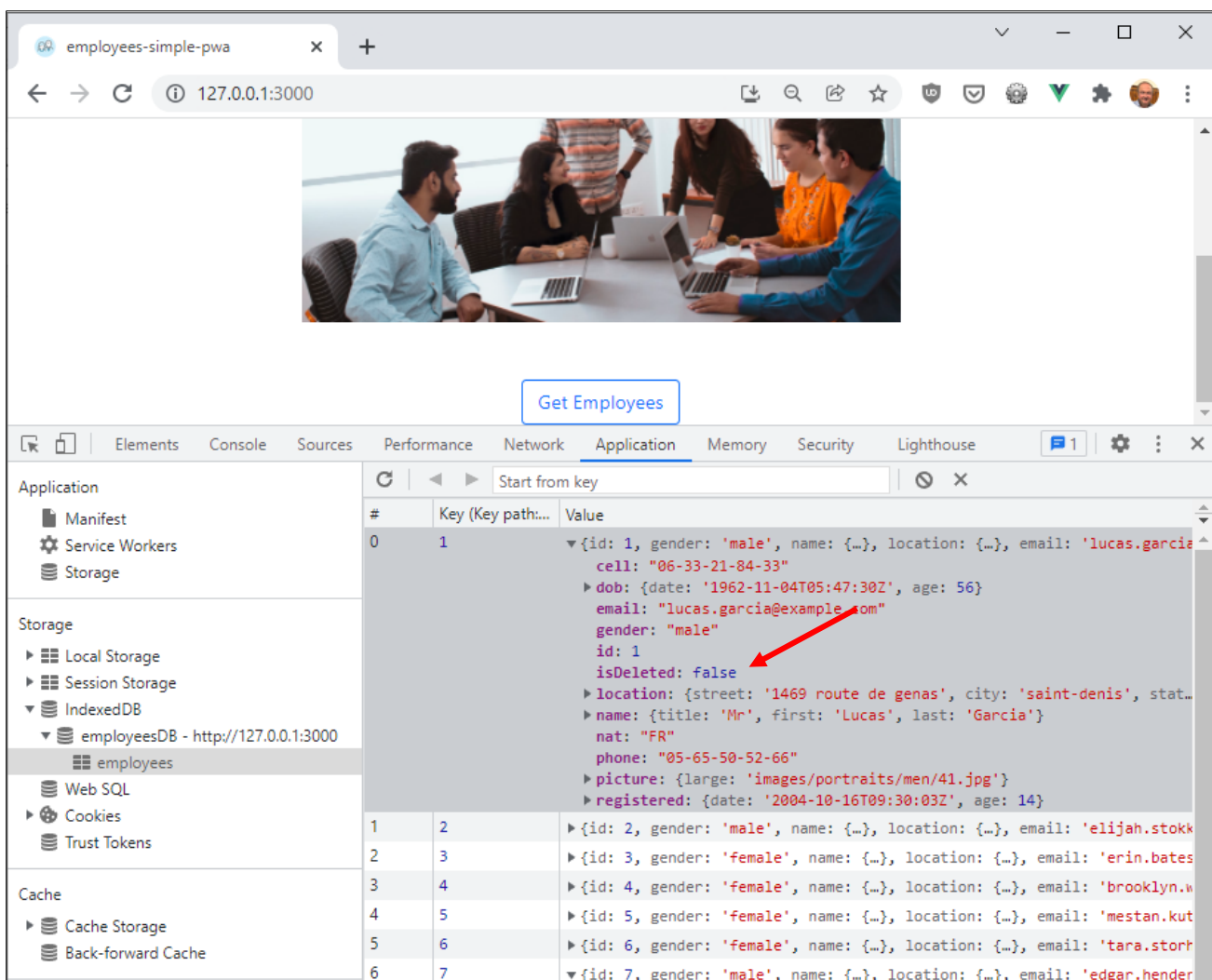
Danach lösche alle **employee** Objekte aus dem Store **employees** (Shortcut **clear**) und füge die soeben vom Server geholten Objekte ein.

wenn der User **offline** ist, hole die **employees** Objekte aus dem Store

```
this.employees = employees.filter((el) => !el.isDeleted);
```

Teste dein Werk!

Online Mode nach **Get Employees**: IndexedDB erstellt, Store befüllt.

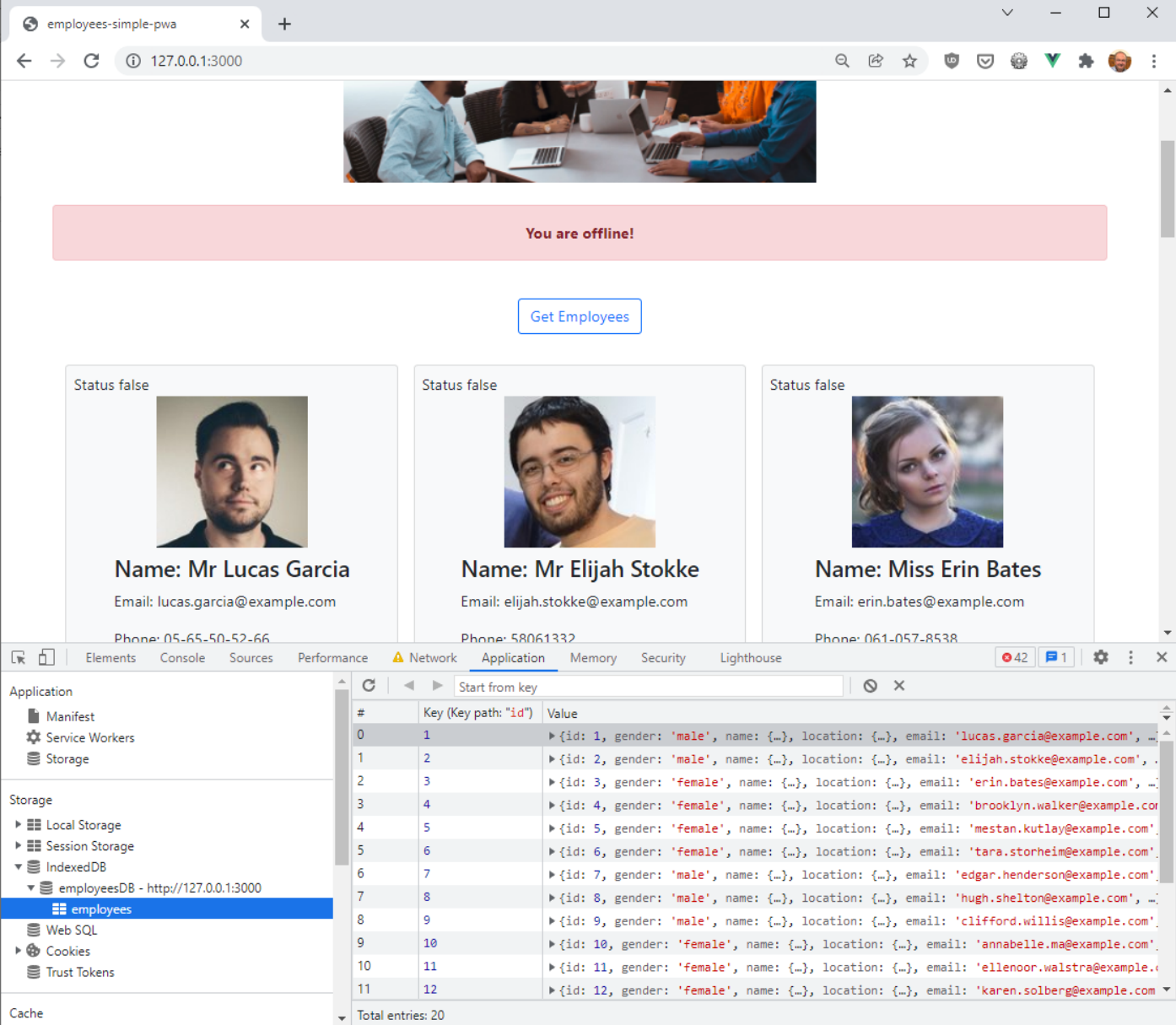


The screenshot shows a web browser with the URL `127.0.0.1:3000` and the application `employees-simple-pwa`. The browser's developer tools are open, and the `Application` tab is selected. In the `Storage` section, the `IndexedDB` database `employeesDB` is expanded, and the `employees` store is selected. The store contains 7 objects. The first object is highlighted, showing its details:

```
{id: 1, gender: 'male', name: {title: 'Mr', first: 'Lucas', last: 'Garcia'}, location: {street: '1469 route de genas', city: 'saint-denis', state: 'FR'}, dob: {date: '1962-11-04T05:47:30Z', age: 56}, email: 'lucas.garcia@example.com', phone: '05-65-50-52-66', picture: {large: 'images/portraits/men/41.jpg'}, registered: {date: '2004-10-16T09:30:03Z', age: 14}, isDeleted: false}
```

A red arrow points to the `isDeleted: false` property in the object.

Robert Baumgartner

Offline Mode nach Refresh und **Get Employees**: Daten IndexedDB. Bilder aus Cache.


The screenshot shows a web browser at 127.0.0.1:3000 displaying an offline Progressive Web App (PWA). The page has a header image of people working, followed by a pink banner saying "You are offline!". Below this is a blue button labeled "Get Employees". Underneath, three employee cards are displayed, each with a placeholder image and the text "Status false". The cards show the following details:

- Mr Lucas Garcia**: Email: lucas.garcia@example.com, Phone: 05-65-50-52-66
- Mr Elijah Stokke**: Email: elijah.stokke@example.com, Phone: 58061332
- Miss Erin Bates**: Email: erin.bates@example.com, Phone: 061-057-8538

The browser's developer tools are open, showing the IndexedDB database. The left sidebar lists the database structure, and the right pane shows a table of 20 entries. The table has columns for #, Key (Key path: "id"), and Value. The values are JSON objects representing employees.

#	Key (Key path: "id")	Value
0	1	{id: 1, gender: 'male', name: {}, location: {}, email: 'lucas.garcia@example.com', ...}
1	2	{id: 2, gender: 'male', name: {}, location: {}, email: 'elijah.stokke@example.com', ...}
2	3	{id: 3, gender: 'female', name: {}, location: {}, email: 'erin.bates@example.com', ...}
3	4	{id: 4, gender: 'female', name: {}, location: {}, email: 'brooklyn.walker@example.com', ...}
4	5	{id: 5, gender: 'female', name: {}, location: {}, email: 'mestan.kutlay@example.com', ...}
5	6	{id: 6, gender: 'female', name: {}, location: {}, email: 'tara.storheim@example.com', ...}
6	7	{id: 7, gender: 'male', name: {}, location: {}, email: 'edgar.henderson@example.com', ...}
7	8	{id: 8, gender: 'male', name: {}, location: {}, email: 'hugh.shelton@example.com', ...}
8	9	{id: 9, gender: 'male', name: {}, location: {}, email: 'clifford.willis@example.com', ...}
9	10	{id: 10, gender: 'female', name: {}, location: {}, email: 'annabelle.ma@example.com', ...}
10	11	{id: 11, gender: 'female', name: {}, location: {}, email: 'ellenoor.walstra@example.com', ...}
11	12	{id: 12, gender: 'female', name: {}, location: {}, email: 'karen.solberg@example.com', ...}

Total entries: 20

Aufgabe 4: Welche Workbox Caching Strategie für **employees** hast du gerade selbst implementiert?**Aufgabe 5:** Löschen eines Employees.In der Methode **delEmployee**,wenn der User **online** ist, lösche den Employee am Server und ändere das GUI und den Store.wenn der User **offline** ist, setze das **isDelete** Property im Store **employees** für das Objekt auf **true** und ändere das GUI.**Aufgabe 6:** Wenn der User online ist (entweder beim Start der App oder wenn der Event **online** kommt) checke den Store auf Objekte, bei denen das **isDelete** Property auf **true** ist und lösche diese vom Server. Ändere den Store entsprechend.

That's it!