

Backend Update und Delete

Lernziele

- ☐ Verstehen des Update- und Delete-Teils von CRUD-Operationen
 - ☐ In der Lage sein, **UPDATE** und **DELETE** API-Routen zu implementieren
-

Update

Um einen Eintrag in deiner Datenbank zu aktualisieren, musst du zwei Dinge tun:

- Definiere eine **PUT** API-Route und
- Verbinde den Submit-Handler eines Editierformulars mit dieser API-Route.

Update mit Mongoose

Zuerst definierst du eine **PUT** API-Route:

```
// /api/jokes/[id].js
if (request.method === "PUT") {
  await Joke.findByIdAndUpdate(id, {
    $set: request.body,
  });
  // Finde den Joke anhand seiner ID und aktualisiere den Inhalt, der Teil
  // des Request-Bodys ist!
  response.status(200).json({ status: `Joke ${id} updated!` });
  // Wenn erfolgreich, erhältst du einen OK-Statuscode.
}
```

PUT mit **fetch**

Zweitens sagst du dem Submit-Handler deines Editierformulars

- dass er die **PUT** API-Route verwenden soll (d.h. eine Anfrage an die Datenbank senden, um einen Eintrag zu bearbeiten),
- auf die Antwort der Datenbank warten und gegebenenfalls das UI aktualisieren oder
- den Benutzer über **push()** auf eine andere Seite navigieren.

💡 Hinweis: **PUT** und **PATCH** sind semantisch unterschiedlich. Laut Konvention verwenden wir **PUT**, um ein ganzes Dokument zu aktualisieren, und **PATCH**, um einzelne Felder zu aktualisieren. In unserem Beispiel verwenden wir **PUT**, da wir nur *ein* Feld zum Aktualisieren haben.

Gehe zu der **page** oder **component**, in der du den Submit-Handler deines Editierformulars schreiben möchtest. Wir benötigen die Methode **mutate**, um die Joke-Komponente nach einem erfolgreichen Update zu aktualisieren.

```
// /components/Joke/index.js
export default function Joke() {
  //...
  const { data, isLoading, mutate } = useSWR(`/api/jokes/${id}`);

  async function handleEdit(event) {
    event.preventDefault();
    const formData = new FormData(event.target);
    const jokeData = Object.fromEntries(formData);

    const response = await fetch(`/api/jokes/${id}`, {
      method: "PUT",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify(jokeData),
    });

    if (response.ok) {
      mutate();
    }
  }
  return; //...
}
```

Delete

Um einen Eintrag in deiner Datenbank zu löschen, musst du zwei Dinge tun:

- Definiere eine **DELETE** API-Route und
- Verbinde eine Handler-Funktion, die diese API-Route verwendet.

Delete mit Mongoose

Zuerst definierst du eine DELETE API-Route:

```
if (request.method === "DELETE") {
  await Joke.findByIdAndDelete(id);
  // Deklariere jokeToDelete als den Joke, der durch seine ID
  // identifiziert wird, und lösche ihn.
  // Diese Zeile übernimmt den gesamten Löschvorgang.
  response.status(200).json({ status: `Joke ${id} successfully deleted.`
});
}
```

DELETE mit fetch

Wir schreiben eine Handler-Funktion, die `fetch()` mit den entsprechenden Argumenten aufruft, und übergeben sie einem Löschbutton:

```
async function handleDelete() {
  await fetch(`/api/jokes/${id}`, {
    method: "DELETE",
  });
  // Du übergibst den durch seine ID identifizierten Joke an unsere
  DELETE-Request-Methode.
  // Dies ist der gesamte Code, der dafür benötigt wird.
  router.push("/");
  // Nachdem der Joke gelöscht wurde, routest du zurück auf unsere Index-
  Seite.
}

return (
  <button type="button" onClick={handleDelete}>
    Delete
  </button>
);
```

Wir wollen `mutate` nicht mit "DELETE"-Anfragen verwenden, da die gelöschten Daten nicht neu abgerufen werden können und eine erneute Validierung zu einem Fehler führen würde.

Resources

- [useSWRMutation \(SWR Docs\)](#)