

Quicknote AB151-07

Post Detail

Im ersten Schritt habe ich die PostShow-View geschrieben. Dafür habe ich zwei Spalten gemacht. Die Linke für das Bild ist 7/12 breit. Die Rechte für die Kommentare ist 5/12 breit. Die Bilder-Seite konnte ich aus `_posts_list` kopieren und musste diese noch ein wenig anpassen.

Nach dem ich die Änderungen übernommen hatte, konnte ich unter dem Post auf das Datum klicken, um auf die Detail-View zu gelangen.

Danach habe ich in der zweiten Spalte mit einem Loop 100 Kommentare erstellt. Diese waren fix im Code und wurden noch nicht aus der DB gelesen.

Post Delete

In diesem Schritt habe ich die Destroy-Methode im PostController geschrieben. Diese überprüft, ob der User der Besitzer des Posts ist. Wenn dies der Fall ist, wird der Post gelöscht, sonst gibt es eine Fehlermeldung. Um diese Methode aufzurufen, habe ich auf der show und list View je ein Link erstellt, welcher den Post löscht.

Berechtigung Post Delete

In diesem Schritt habe ich das Post Model angepasst. Dazu habe ich ihn mit der Methode `belongs_to?` erweitert. Diese schaut, ob es einen User mit der id des User-Attributs des Posts gibt.

Model: Posts mit Like ergänzen

Hier habe ich ein neues Model namens Like erstellt. Ein Like gehört zu genau einem User und einem Post. Da ein User einen Post nur einmal liken kann, habe ich eine Validierung definiert, welche sicherstellt, dass bei einer `user_id` die `post_id` unique sein muss. Danach habe ich die Migration mit „db:migrate“ ausgeführt. Im Post Model habe ich anschliessend eine Methode implementiert, welche zurückgibt, ob ein Post von einem User (Parameter) geliked ist.

Im User Model habe ich anschliessend definiert, dass ein User mehrere Likes haben kann.

Da Rails die JS-Library turbolinks verwendet, um die Navigation zu beschleunigen und diese unsere Link-Funktion stört musste diese noch aus unserem `application.js` entfernen.

Likes-Controller

In diesem Schritt habe einen neuen Like-Controller erstellt und eine create, eine destroy und eine `like_params` Methode implementiert. Diese können nur von authentifizierten Usern aufgerufen werden. Die create Methode erstellt einen neuen Like die destroy Methode löschen den Like auf welche die ID referenziert.

Post Like

Dieser Schritt war der letzte und aufwändigste. Als erstes habe ich die Post-List angepasst. Damit man einen Post liken kann. Diesen Code habe später teilweise ausgelagert. Insgesamt gehören 4 partial Views zu den Posts.

- `_like_icon`
- `_like_text`
- `_photos`
- `_posts_list`

Damit man die Seite nicht aktualisieren muss, sobald man einen Post geliked hat. Habe ich zwei Ajax-Datei vom Share verwendet. Sobald ich die Views angepasst hatte, musste ich noch einen Fehler beheben. Wenn man einen Post löschen wollte, der bereits Likes hatte, gab es einen Fehler, weil der Like immer noch eine Referenz auf den Post hatte.

Um dies zu beheben, schrieb ich hinter die Beziehung mit den Likes im Post Model „dependent: :destroy“.

Das bedeutet, dass wenn man einen Post löscht auch alle dazugehörigen Likes gelöscht werden.

Selbstreflexion

1. Was habe ich gelernt?

Wie man Sicherheitsmassnahmen einbauen kann um Funktionen vor nichtautorisierten Usern zu schützen.

Wie man ein Model anpasst.

Anwendung von Partial Views.

2. Was hat mich behindert?

Behindert hat mich nichts Technisches. Allerdings war das viele Abschreiben aufwändig.

3. Was habe ich nicht verstanden?

Ich habe alles verstanden.


4. Was kann ich beim Studium besser machen?

Während dem AB habe ich gut gearbeitet. Ich habe während dem lösen des ABs Notizen gemacht und konnte diese bei Schreiben der QN anschauen.


Fazit


Das Arbeitsblatt fand ich sehr aufwändig und gegen den Schluss auch unübersichtlich. Man musste viele Views mehrmals editieren und Teile auslagern. Zudem musste man viel abschreiben. Ansonsten zeigt es sehr gut wie man ein neues Feature in eine App einbauen kann, wie zum Beispiel eine Like-Funktion.



Post-Liste



Luca









Luca Lidl


LESS THAN A MINUTE AGO


Add a comment ...







Luca








Luca likes this

Luca BSC YB

1 MINUTE AGO

Add a comment ...



[OUT US](#)

[SUPPORT](#)

[BLOG](#)

[PRESS](#)

[API](#)

[JOB](#)

[PRIVACY](#)

[TERMS](#)

[DIRECTORY](#)

[LANGUAGE](#)

Detail-View

