

Modul 294 - Aufgabenblatt 10

von Lukas Meier

Unterricht vom 27.02.2026

1 Vorbereitung

Nutzen Sie den aktuellen Projektstand mit funktionierendem CRUD (GET, POST, PATCH/PUT, DELETE).

Legen Sie eine zentrale Datenvariable an, z.B. `allNotes`, die alle vom Backend geladenen Notizen enthält.

2 UI-Elemente für Suche, Filter und Sortierung

Ergänzen Sie im HTML mindestens folgende Elemente:

- Suchfeld (Textinput)
- Filter (Select oder Buttons, z. B. alle / offen / erledigt)
- Sortierung (Select, z. B. neueste / älteste / Titel A-Z)

3 UI-State definieren

Erstellen Sie ein State-Objekt, das die aktuelle Auswahl speichert, z. B.:

- `searchTerm`
- `status`
- `sortBy`

4 Suche implementieren

Erstellen Sie eine Funktion `applySearch(notes, term)`.

Anforderungen:

- Suche über Titel und Inhalt
- Gross-/Kleinschreibung ignorieren
- Leerer Suchbegriff liefert alle Notizen zurück

5 Filter implementieren

Erstellen Sie eine Funktion `applyFilter(notes, status)`.

Nutzen Sie den Status Ihrer Datenstruktur (z.B. `done`, `completed` oder `archived`).

Der Filter soll mindestens 3 Zustände unterstützen:

- alle
- offen
- erledigt

6 Sortierung implementieren

Erstellen Sie eine Funktion `applySort(notes, sortBy)`.

Unterstützen Sie mindestens:

- neueste zuerst (Datum absteigend)
- älteste zuerst (Datum aufsteigend)
- Titel alphabetisch

Wichtig: Verändern Sie nicht direkt das Original-Array, sondern arbeiten Sie mit einer Kopie.

7 Pipeline kombinieren

Erstellen Sie eine Funktion `getVisibleNotes()` mit folgendem Ablauf:

- Suche anwenden

- Filter anwenden
- Sortierung anwenden

Das Ergebnis soll an `renderNotes(...)` übergeben werden.

8 Events anbinden

Binden Sie Events an alle UI-Controls:

- `input` beim Suchfeld
- `change` bei Filter und Sortierung

Bei jeder Änderung:

- State aktualisieren
- Ansicht neu rendern

9 Integration mit bestehendem CRUD

Nach Create, Update und Delete muss die Liste weiterhin korrekt angezeigt werden.

Vorgehen:

- Daten neu laden oder lokal aktualisieren
- danach `getVisibleNotes()` und `renderNotes()` ausführen

10 Testfälle

Testen Sie mindestens folgende Fälle:

- Suche nach Begriffen, die in mehreren Notizen vorkommen
- Suche nach Begriff, der nicht existiert
- Kombination aus Suche + Filter
- Kombination aus Suche + Filter + Sortierung

Dokumentieren Sie kurz, welche Kombinationen Sie getestet haben.

11 Optional: Usability

Implementieren Sie eine oder mehrere Erweiterungen:

- “Keine Treffer”-Hinweis
- “Filter zurücksetzen”-Button
- Debounce für die Suche (z. B. 200ms)

12 Abschluss

Die Lektion ist abgeschlossen, wenn:

- Suche, Filter und Sortierung einzeln funktionieren
- alle drei Funktionen kombiniert stabil laufen
- CRUD weiterhin korrekt funktioniert

Erstellen Sie danach einen Commit, z. B. `feat: add note search, filter and sorting controls`.