

LU14 - Probeprüfung: Wetter-App

Zeit: 60 Minuten

Hilfsmittel: Unterlagen auf Moodle, Internet, persönliche Notizen – keine KI, keine Lösungen anderer Lernenden

Rahmenbedingungen

Heute wird konzentriert und selbstständig gearbeitet. Wer sich vorbereiten oder üben möchte, soll das ungestört tun können.

- Jede/r arbeitet für sich – keine Gespräche, keine Ablenkungen
- Keine sozialen Medien, kein Shopping, kein Chatten
- Bei Fragen: Hand heben und die Lehrperson fragen

<note important> **Hinweis:** Die Daten von MockAPI stimmen unter Umständen nicht 1:1 mit dem Design überein. Das ist in der Praxis normal: Design arbeitet mit Dummy-Daten, die Programmierung mit echten Daten. </note>

Erste Schritte

1. Starter-Projekt von Moodle herunterladen (*LU14 - Probeprüfung*)
2. Projektordner im Code-Editor öffnen (*VS Code oder WebStorm*) und Figma-File analysieren
3. Zeit für Fragen im Plenum
4. Abhängigkeiten installieren
5. Entwicklungsserver starten
6. Projekt im Browser öffnen

Danach die Schritte unten gemäss den Kommentaren im Code ausführen.

Aufgaben

Container.vue

<script>-Bereich

| Schritt | Aufgabe |
|---------|--|
| 1 | API-Daten von MockAPI laden |
| 2 | Antwort der API auslesen und der Variable zuweisen |

<template>-Bereich

| Schritt | Aufgabe |
|---------|--|
| 3 | Über die geladenen Daten iterieren und jedes Element an die Day-Komponente übergeben |

Day.vue

<script>-Bereich

| Schritt | Aufgabe |
|---------|--|
| 4 | Erwartete Prop definieren (Name und Datentyp) |
| 5 | Reaktive Variable für den Öffnungs-Zustand anlegen |

<template>-Bereich

| Schritt | Aufgabe |
|---------|--|
| 6 | Klick-Interaktion auf dem Button einrichten |
| 7 | Wochentag, Maximal- und Minimaltemperatur anzeigen |
| 8 | Klasse für den geöffneten Zustand dynamisch setzen |
| 9 | Beschreibung, Luftfeuchtigkeit, UV-Index und Wind anzeigen |

Die API gibt ein Array (Liste) mit Objekten zurück. Ein einzelnes Objekt (= einzelner Listeneintrag) hat diese Struktur:

```
{
  "id": 1,
  "wochentag": "Montag",
  "datum": "6. Juli",
  "icon": "https://storage.googleapis.com/m291-lb02-cases/weather/sun.png",
  "tempMax": 31,
  "tempMin": 18,
  "beschreibung": "Der Tag startet strahlend und die Temperatur wird bis zum
Nachmittag auf bis zu 31 Gradklettern. Kein Wölkchen trübt den Himmel – ein
perfekter Hochsommertag.",
  "feuchtigkeit": 38,
  "uvIndex": 8,
  "wind": "12 km/h W",
  "ratschlag": "Sonnencreme LSF 50 nicht vergessen!"
}
```

Day.vue - CSS

<style>-Bereich

| Schritt | Aufgabe |
|---------|---|
| 10 | Versetzten Einblende-Effekt (Treppeneffekt) pro Kachel umsetzen |

| Schritt | Aufgabe |
|---------|---|
| 11 | ☐ BONUS: Aktiven Zustand der Kachel gestalten (machen Sie das erst zum Schluss) |
| 12 | Darstellung des Datums aus Figma umsetzen |
| 13 | Darstellung der Beschreibung aus Figma umsetzen |
| 14 | Übergangsanimation für das Auf- und Zuklappen einrichten |
| 15 | Geöffneten Zustand der Detailansicht definieren |
| 16 | Layout der Chips (Feuchtigkeit, UV, Wind) umsetzen |

Abgabe

1. node_modules-Ordner **nicht** im ZIP einschliessen
2. Projektordner als .zip komprimieren
3. ZIP-Datei in Moodle hochladen (LU14 - Probeprüfung → Abgabe Wetter-App)

Fertig

- Vergleichen Sie Ihre Lösung mit der Musterlösung (wird auf Moodle freigeschaltet, sobald Sie abgegeben haben): Was stimmt überein? Was nicht?
- Wiederholen Sie die Theorie auf Moodle zu den Punkten, wo Sie unsicher waren
- In Container.vue wird der Ladezustand noch nicht dem User angezeigt. Ergänzen Sie die Komponente mit einer isLoading-Variable. Hat sie den Wert true, zeigen Sie den Text „Daten werden geladen...“ an - ist sie falsch, zeigen Sie das div mit der Klasse forecast an. Sie brauchen dazu die Direktiven v-if und v-else.

From:

<https://wiki.bzz.ch/> - BZZ - Modulwiki

Permanent link:

https://wiki.bzz.ch/de/modul/m291/learningunits/lu14/aufgaben/a_instructions?rev=1781605541

Last update: 2026/06/16 12:25

