



# FWE | SS 23 | Hausaufgabe 2

**Bewertung** - Die Hausaufgabe geht zu 15% in die Gesamtnote. Es gibt insgesamt 100 Punkte. Eine Bewertung mit weniger als 50 Punkten gilt für den Ausschluss von der Lehrveranstaltung.

**Bearbeitungszeit** - Beginn **19.04.2023** - Ende **21.05.2023** (bis 23:59 Uhr)

**⚠️** Code, der **nach** der Bearbeitungszeit eingereicht wird, wird **ausnahmslos nicht** bewertet.

**Modus** - Die Hausaufgabe muss **einzel** bearbeitet werden.

## Die Aufgabe

Sushi. Lasagne. Schnitzel. Pizza. Salat. Omelette. Es gibt die verschiedensten Gerichte und viele Arten wie man sie zubereiten kann. Ist man kein erfahrener Koch, klappt das mit einem guten Rezept am Besten. Doch wo findet man heutzutage schnell und einfach gute Rezepte? Im Internet! Deswegen wollen wir in den beiden Hausaufgaben einen Teil dazu beitragen und uns eine eigene Rezeptverwaltung bauen.

In der zweiten Hausaufgabe geht nun um das Frontend, welche das Backend aus der ersten Hausaufgabe erweitert.

Es sollen die folgenden Features umgesetzt werden:

- Erstelle eine Listenansicht und eine Detailansicht für die Rezepte. Aus den jeweiligen Ansichten sollen sich die Daten anlegen, ändern und ggf. löschen lassen (**CRUD**)
  - **CRUD System** - Es ist möglich mittels der Rest API aus der ersten Hausaufgabe Rezepte und Zutaten anzuzeigen, anzulegen, zu ändern und zu löschen
  - Biete eine Möglichkeit Rezepte nach ihrem Namen zu durchsuchen indem du die entsprechende API Route dazu integrierst
  - Biete eine Möglichkeit alle Rezepte zu einer Zutat anzuzeigen
- 
- Schreibe eine genaue Dokumentation der Applikation in der Readme, welche mindestens
    - die Funktionalitäten beschreibt
    - die Struktur der Routen auflistet
    - beschreibt wie die Applikation aufzusetzen ist
    - beschreibt wie die API getestet werden kann.

 Die Dokumentation **muss** in Markdown verfasst werden.

- Es dürfen beliebig viele Felder und weitere Tabellen hinzugefügt werden, solange diese zur App passen und eine sinnvolle Funktionalität aufweisen.
- Das System ist “Single User”, muss also keine Accounts, Sessions, Zugriffsrechte, Logins oder Ähnliches beinhalten.

## Generelle Informationen

Bitte benutze Deinen Code aus Hausaufgabe 1 als Grundlage für Hausaufgabe 2. Zur Abnahme muss sowohl das Backend als auch das Frontend **aus einem Repository** heraus startbar sein. Bitte nutzt ausschließlich React und Typescript für die Hausaufgabe.

 **node\_modules, environment files und api keys** gehören im Normalfall **nicht** in ein Repository, bitte schaut Euch dazu .gitignore files an und nutzt sie entsprechend.

Eine readme File in einem Projekt ist hauptsächlich für andere Entwickler gedacht, welche Euer Projekt aufsetzen und nutzen möchten und sollte daher die entsprechenden Informationen beinhalten. Einen guten Artikel dazu gibt es hier:  
<https://goo.gl/ZNh6yh>.

Externe APIs für Euren Freestyle Part können z.B. hier gefunden werden:  
<https://github.com/toddmotto/publicapis> und sollten passend zu der Hausaufgabe ausgewählt werden. Ihr dürft euch auch gerne eine eigene zweite API schreiben über die Ihr Daten konsumiert.

## Abgabe der Aufgabe

Der gesamte Code muss in das Gitlab der h\_da gepushed werden:  
<https://code.fbi.h-da.de/>.

Wie Du dabei Git verwendest, ist Dir überlassen. Wichtig ist nur, dass Du Deine Ergebnisse vor dem Abgabetermin in den Master/Main Branch eincheckst. Abgaben als .zip Datei oder per E-Mail sind unzulässig und werden **ausnahmslos nicht bewertet**. In der Vergangenheit kam es hier des Öfteren zu Problemen. Kümmert Euch bitte rechtzeitig darum.

**Der Name der ersten und zweiten Hausaufgabe muss folgendem Schema entsprechen FWE-SS-23-123456** Bitte ersetzt das 123456 durch Eure Matrikelnummer. Beide Hausaufgaben werden **im gleichen Repository** abgegeben.

**WICHTIG: Achtet darauf dass Eure Repository auf PRIVATE gestellt ist und Ihr die folgenden beiden Nutzer eingeladen habt @fbi1483 und @hda10128**

## Bewertungskriterien (Gesamt - 100 Punkte)

### Dokumentation & Funktionalität des Projekts - (20 Punkte)

- 
- Hast Du alle benötigten Informationen über Dein Projekt beschrieben?
  - Funktioniert das Projekt nach dem Aufsetzen korrekt?

### Einheitliche Syntax & moderner modularer Code - (10 Punkte)

---

- Wie sind Deine Funktionen/Variablen benannt?
- Wie lang sind Deine Funktionen?
- Wie ist Dein Projekt strukturiert?
- Wie interagieren die Klassen miteinander?
- Genügt die Syntax einem Codingstandard?

- Gleichmäßiges Spacing?
- Kein Mix von Sprachen (deutsch/englisch)?
- Kommentierter Code an den notwendigen Stellen?

### **Kernfunktionalität - (30 Punkte)**

---

- Rezepte erstellen / bearbeiten / löschen / in Liste darstellen
- Zutaten zu Rezepten hinzufügen & löschen

### **Fehlerbehandlung - (10 Punkte)**

---

- Hast Du an alle möglichen Nebeneffekte gedacht?
- Was passiert, wenn nicht alle Felder gesendet werden?
- Leere Felder gesendet werden?
- Inkorrekte Felder, Script code oder HTML Code gesendet wird?

### **Benutzung von Komponenten, Routern - (10 Punkte)**

---

- verwende `react-router` um zwischen den unterschiedlichen Ansichten zu wechseln
- Listenansicht / Detailansicht von Rezepten

### **Such Funktionalität - (20 Punkte)**

---

- Biete eine Möglichkeit nach Rezepten zu suchen
- Biete eine Möglichkeit alle Rezepte zu einer bestimmten Zutat anzuzeigen