



# FWE | SS 23 | Hausaufgabe 1

**Bewertung** - Die Hausaufgabe geht zu 15% in die Gesamtnote. Es gibt insgesamt 100 Punkte. Eine Bewertung mit weniger als 50 Punkten gilt führt zum 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**n bearbeitet werden.

## Die Aufgabe

Sushi. Lasagne. Schnitzel. Pizza. Salat. Omelette. Es gibt die verschiedenste 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 ersten Hausaufgabe geht es dabei nur um die Backend API, welche in der zweiten Hausaufgabe dann um ein Frontend erweitert wird.

Es sollen die folgenden Features umgesetzt werden:

- **CRUD System** - Es ist möglich mittels einer HTTP API Rezepte und Zutaten anzuzeigen, anzulegen, zu ändern und zu löschen
  - Ein Rezept beschreibt die Arbeitsschritte um ein bestimmtes Gericht zu kochen.
    - Felder z.B. Name, Beschreibung, Link zu einem Bild, Rezeptschritte, Rating, ...
  - Ein Zutat ist Bestandteil eines Rezepts und wird benötigt um dieses zu kochen
    - Felder z.B. Name, Beschreibung, Link zu einem Bild, ...
  - Ein Rezept kann 0 oder N Zutaten haben
  - Eine Zutat kann 0 oder N Rezepte haben
  - Es muss möglich sein ein oder mehrere Zutaten zu einem Rezept hinzuzufügen und löschen zu können. Hierbei soll die benötigte Menge der Zutat und die Einheit (z.B. ml, l, gr, kg, ...) spezifiziert werden können.
  - Es soll eine Route geben die alle Rezepte in dem eine bestimmte Zutat existiert zurückgeben kann.
  - Es muss möglich sein Rezepte nach ihrem Namen zu durchsuchen
- 

- 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.

- Stelle uns eine Möglichkeit zur Verfügung, das System zu testen. Es muss in der ersten Hausaufgabe noch kein Frontend geschrieben werden. Präferiert sind automatisierte Tests (Bonuspunkte). Es reicht aber auch eine Postman Collection oder curl Aufrufe. Wichtig ist, dass **alle funktionalen** Fälle abgedeckt sind.
- 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

In dieser Hausaufgabe geht es erstmal **nur um das Backend** unserer App.

Verwende dafür **ausnahmslos Node und TypeScript**.

Als Webserver empfehlen wir **Express** - hierbei hast Du aber freie Wahl. Nutze sämtliche Libraries, die Du möchtest und die Dir bei Deiner Aufgabe helfen. Für das Speichern der Daten kannst Du aus den folgenden Varianten wählen

- mySQL (empfohlen)
- mongoDB
- redis
- json files

Bitte hinterlasse uns eine **genaue Anleitung** zum Aufsetzen in der ReadME deines Projekts. Gerne kannst Du auch **Docker Container** verwenden und uns Dein Backend in einem fertigen Container zur Verfügung stellen, welchen Du in die Gitlab Registry oder den Docker hub pushst. Bitte denke auch in diesem Fall an eine **genaue Anleitung**, wie wir Deine Hausaufgabe aufsetzen sollen.

---

**⚠ 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 in Klassen, Variablen und Funktionsnamen (deutsch/englisch)?
- Kommentierter Code an den notwendigen Stellen?

### Kernfunktionalität - (25 Punkte)

---

- Ist das Anlegen, Bearbeiten, Auflisten und Löschen von Rezepten möglich?
- Ist das Anlegen, Bearbeiten, Auflisten und Löschen von Zutaten möglich?
- Ist es möglich Zutaten zu Rezepten hinzuzufügen und sie zu entfernen?
- Können alle Rezepte die zu einer Zutat gehören angezeigt werden?

- Können Rezepte nach Ihrem Namen gesucht werden?

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

---

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

### **Verwendung von HTTP - (5 Punkte)**

---

- Ist Dein API Design nach Rest Best Practices aufgebaut?
- Hast Du korrekte Response Codes?

### **Tests - (5+5 Punkte bei automatisierten Tests)**

---

- Wie testest Du deine Hausaufgabe?
- Beschreibe das genaue Vorgehen und gib uns die Möglichkeit Dein Projekt ebenfalls zu testen  Nur durch die Nutzung von automatisierten Tests ist hier die volle Punktzahl zu erreichen

 **WICHTIG: Beschreibe Freestyle Task#1 & #2 in der readme genau (sonst kann es dafür keine Punkte geben).** 

### **Freestyle task #1 - (10 Punkte)**

---

- Überrasche uns mit einem Feature, welches nicht in der Aufgabenbeschreibung gefordert ist. Wichtig: Es muss zu dem aktuellen Projekt passen. Dieses Feature **darf keine Daten von einer externen API konsumieren** sondern muss die Funktionalität der bestehende App erweitern.

### **Freestyle task #2 - (10 Punkte)**

---

- Überrasche uns mit einem weiteren Feature, welches nicht in der Aufgabenbeschreibung gefordert ist. Wichtig: Es muss zu dem aktuellen Projekt passen. Dieses Feature **muss Daten von einer externen API konsumieren**.