Projektarbeit M165

Inhalt

[1 Auftragsüberblick 1](#_Toc134605657)

[1.1 Rezeptkategorien 2](#_Toc134605658)

[1.2 Projektstruktur 2](#_Toc134605659)

[1.3 Projektmanagement 3](#_Toc134605660)

[1.4 Datenbank 3](#_Toc134605661)

[1.5 Node.js-Anwendung 3](#_Toc134605662)

[2 Abgabe, Termine 4](#_Toc134605663)

[3 Vorgehen 4](#_Toc134605664)

[4 Bewertung 4](#_Toc134605665)

# Auftragsüberblick

Sie erstellen eine Datenbank mit CouchDB zur **Verwaltung von Rezepten**. Dazu stehen 6 Doppellektionen zur Verfügung.

Ein Rezept wird wie folgt modelliert:

* Rezeptname: Text, muss vorhanden sein.
* Rezeptkategorie: Text, muss vorhanden sein.
* Erstelldatum: Datum, muss vorhanden sein.
* Zutaten: muss vorhanden sein, unterscheidet sich von Rezept zu Rezept.
* Anleitung: Text, muss vorhanden sein.
* Bild: Anhang, muss vorhanden sein.
* Bewertung: Zahl zwischen 1 – 10, kann vorhanden sein.
* Kommentar: Text, kann vorhanden sein.
* 2 sinnvolle eigene Attribute, die Sie selbst definieren. Diese können vorhanden sein.

Sie erstellen 6 verschiedene Rezepte:

* Rezept 1 beinhaltet nur die Muss-Attribute
* Rezept 2 beinhaltet alle Muss-Attribute sowie einen Kommentar
* Rezept 3 beinhaltet alle möglichen Attribute inkl. ihre eigenen Attribute
* Rezept 4 und 5 beinhalten alle Muss-Attribute sowie die Bewertung
* Rezept 6 beinhaltet nur den Rezeptnamen und die Rezeptkategorie: Dieses Rezept dient als Test für die Validierung und kann nicht in die Datenbank eingefügt werden.

Sie erstellen eine neue Datenbank und fügen die Rezepte in ihre Datenbank ein.

Mit einer Node.js-Anwendung können Sie diese Rezepte anzeigen und suchen.

## Rezeptkategorien

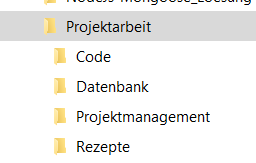
Sie wählen **eine** Rezeptkategorie aus. Alle SuS haben eine andere Rezeptkategorie. Die Lehrperson hat eine Liste und koordiniert die Vergabe am 10.05.2023.

Es stehen folgende Rezeptkategorien zur Verfügung:

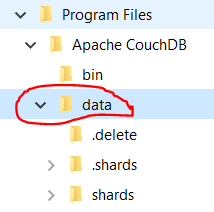
* Kuchen
* Desserts
* Salate
* Gratins
* Grill
* Pizza
* Reisgerichte
* Vorspeisen
* Sandwiches
* Brote
* Coupe
* Apéro
* Omeletten
* Wok
* Burger

## Projektstruktur

Ihre Projektstruktur sieht wie folgt aus:



* Verzeichnis Code: Beinhaltet ihre Node.js-Anwendung gemäss Abschnitt «Node.js-Anwendung»
* Verzeichnis Datenbank: Beinhaltet ihre Datenbank. In dieses Verzeichnis kopieren Sie den **gesamten** Inhalt des **data**-Verzeichnisses, siehe <https://docs.couchdb.org/en/stable/maintenance/backups.html>



* Verzeichnis Projektmanagement: Beinhaltet die Projektplanung sowie die Screenshots gemäss Abschnitt «Projektmanagement»
* Verzeichnis Rezepte: Beinhaltet die 6 Rezepte als json-Files

Sie geben ihr gesamtes Projekt in GitLab ab. Es gibt keine Abgabe per E-Mail / Teams etc.

## Projektmanagement

Eine Planung ihrer Arbeiten und Termine hilft Ihnen, schlaflose Nächte zu vermeiden.

Erstellen Sie folgende Projektplanung, z.B. in Excel:

* Auflistung aller Arbeiten mit dem Aufwand in Stunden.
* Definition von 2-3 Meilensteinen. Ein Meilenstein ist ein Termin mit definierten Ergebnissen.

Sie erstellen eine rudimentäre Dokumentation. Diese besteht aus folgenden Screenshots:

* Aufgesetzte, lauffähige Replikation
* Vollständiges Rezept mit Bild (in Fauxton oder Node.js-Anwendung)
* Fehlermeldung bei Rezept 6 wegen Validierung

## Datenbank

Sie erstellen eine neue Datenbank «Rezepte-Mein-Name», z.B. «Rezepte-Martin-Hager».

Folgende Attribute ihrer Rezepte werden validiert:

* Rezeptname
* Rezeptkategorie
* Anleitung

Sie erstellen einen Index, damit die Volltextsuche nach Zutaten performant funktioniert. Bemerkung: Die Suche funktioniert auch ohne Index, d.h. den Index können Sie erst am Schluss erstellen.

Die modellierten Rezepte fügen Sie als json-Files in ihre Datenbank ein. Sie können cURL oder Fauxton verwenden. Rezept 6 können Sie nicht einfügen, da es nicht valid ist.

Sie setzen eine Replikation ihrer Datenbank auf. Die Replikation findet alle 30 Sek. statt.

## Node.js-Anwendung

Sie erstellen eine einfache Node.js-Anwendung, welche folgende Anforderungen erfüllt:

* Beim Start / Aufruf der Anwendung werden alle Rezepte angezeigt. Das Bild muss nicht sichtbar sein (es kann).
* Rezepte können nach Zutaten gesucht werden: Es wird nach einer einzigen Zutat gesucht. Sie müssen nicht nach mehreren Zutaten gleichzeitig suchen können.
* Es braucht kein Paging: Alle Rezepte können auf einer Seite angezeigt werden.
* Benutzernamen und Passwörter können Sie hart codieren.
* Nach dem Klonen des Repositories wird die Node.js-Anwendung mit den Befehlen «npm install» und «npm run start» gestartet.

Mögliche Node.js-Bibliotheken:

* <https://github.com/apache/couchdb-nano>
* <https://www.npmjs.com/package/node-couchdb>

Sie können auch eine andere Bibliothek verwenden.

# Abgabe, Termine

|  |  |
| --- | --- |
| *Was* | *Termin* |
| Projektplan in GitLab | 17.05.2023, 22 Uhr |
| Rezepte als json-Files in GitLab | 31.05.2023, 22 Uhr |
| Gesamtes Projekt in GitLab | 23.06.2023, 22 Uhr |

# Vorgehen

Ein mögliches Vorgehen ist wie folgt:

* Überblick gewinnen, Projektplan erstellen
* Modellierung Rezepte
* GitLab-Projekt mit definierter Struktur erstellen
* Datenbank erstellen, Rezepte mit Bild einfügen
* Teilvalidierung Schema
* Indexe erstellen
* Replikation aufsetzen
* Node.js-Anwendung erstellen
* Test und Abgabe

# Bewertung

Diese Projektarbeit wird zum ersten Mal durchgeführt. Falls es unvorhergesehene Probleme bei der Arbeit gibt, melden Sie sich frühzeitig bei der Lehrperson. Falls nötig, wird die Bewertung angepasst!

Eine Woche vor Abgabe ist definitiv zu spät für Anpassungen.

Siehe Excel-File, folgt noch.