Projektarbeit M165

Inhalt

[1 Auftragsüberblick 1](#_Toc134625280)

[1.1 Rezeptkategorien 2](#_Toc134625281)

[1.2 Projektstruktur 2](#_Toc134625282)

[1.3 Projektmanagement 3](#_Toc134625283)

[1.4 Datenbank 3](#_Toc134625284)

[1.5 Web-Anwendung 3](#_Toc134625285)

[2 Abgabe, Termine 4](#_Toc134625286)

[3 Vorgehen 4](#_Toc134625287)

[4 Bewertung 4](#_Toc134625288)

# Auftragsüberblick

Sie erstellen eine Datenbank mit CouchDB zur **Verwaltung von Rezepten**. Dazu stehen 4 Doppellektionen zur Verfügung. Sie arbeiten zu zweit.

Ein Rezept wird wie folgt modelliert:

* Rezeptname: Text, muss vorhanden sein.
* Rezeptkategorie: Text, muss vorhanden sein. Ein Rezept kann zu verschiedenen Rezeptkategorien gehören.
* 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 11 verschiedene Rezepte:

* Rezepte 1 und 2 beinhalten nur die Muss-Attribute, Rezeptkategorie ist verschieden
* Rezepte 3 und 4 beinhalten alle Muss-Attribute sowie einen Kommentar, Rezeptkategorie ist verschieden
* Rezepte 5 und 6 beinhalten alle Muss-Attribute sowie mehrere Kommentare, Rezeptkategorie ist verschieden
* Rezept 7 und 8 beinhalten alle möglichen Attribute inkl. ihre eigenen Attribute, Rezeptkategorie ist verschieden
* Rezept 9 und 10 beinhalten alle Muss-Attribute sowie die Bewertung, Rezeptkategorie ist verschieden
* Rezept 11 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 Web-Anwendung können Sie diese Rezepte anzeigen und suchen.

## Rezeptkategorien

Es stehen folgende Rezeptkategorien zur Verfügung:

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

Sie können auch andere Kategorien verwenden.

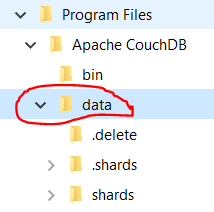
## Projektstruktur

Ihre Projektstruktur sieht wie folgt aus:

Ein Bild, das Text, Screenshot, Schrift, Design enthält.

Automatisch generierte Beschreibung

* Verzeichnis Code: Beinhaltet ihre Web-Anwendung gemäss Abschnitt «Web-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 Doku: Beinhaltet die Screenshots gemäss Abschnitt «Dokumentation»
* Verzeichnis Rezepte: Beinhaltet alle Rezepte als json-Files

Sie geben ihr gesamtes Projekt in GitLab oder GitHub ab.

## Dokumentation

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

* Aufgesetzte, lauffähige Replikation
* Vollständiges Rezept mit Bild in der Datenbank (in Fauxton oder Web-Anwendung)
* Fehlermeldung bei Rezept 11 wegen Validierung

## Datenbank

Sie erstellen eine neue Datenbank «Rezepte».

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 11 können Sie nicht einfügen, da es nicht valid ist.

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

## Web-Anwendung

<https://www.javatpoint.com/nodejs-couchdb>

Sie erstellen eine einfache Web-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.

Sie können die Web-Anwendung mit Node.js erstellen:

* 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.

Sie können auch eine andere Web-Technologie als Node.js verwenden. Die Datenbank kann auch in der Cloud sein.

# Abgabe, Termine

|  |  |
| --- | --- |
| *Was* | *Termin* |
| Gesamtes Projekt in Git | 21.6.23 22:00 Uhr |

# Vorgehen

Ein mögliches Vorgehen ist wie folgt:

* Überblick gewinnen
* Modellierung Rezepte
* Git-Projekt mit definierter Struktur erstellen
* Datenbank erstellen, Rezepte mit Bild einfügen
* Teilvalidierung Schema
* Indexe erstellen
* Replikation aufsetzen
* Web-Anwendung erstellen
* Test und Abgabe

# Bewertung

Siehe Excel-File.