Modul 295

**Backend  
entwickeln**

Dokumentation  
Lukas Sutter

Inhalt

[Vorwort 3](#_Toc152589875)

[Projekt M295 4](#_Toc152589876)

[Ziele des ÜKs 4](#_Toc152589877)

[Inhalt des ÜKs 4](#_Toc152589878)

[Datenbankstruktur 5](#_Toc152589879)

[Gebrauch Datenbank 5](#_Toc152589880)

[Datenbankaufstellung 5](#_Toc152589881)

[Tagesablauf 6](#_Toc152589882)

[Tag 1 des ÜKs 6](#_Toc152589883)

[Tag 2 des ÜKs 7](#_Toc152589884)

[Tag 3 des ÜKs (Arbeit Zuhause, weil krank) 8](#_Toc152589885)

[Tag 4 des ÜKs 9](#_Toc152589886)

[Tag 5 des ÜKs 10](#_Toc152589887)

[Inhalt 11](#_Toc152589888)

[Vergleichsmatrix der Evaluation des API Dokumentation Tools 11](#_Toc152589889)

[Punkteverteilung 11](#_Toc152589890)

[Gesamtpunkte Evaluation 12](#_Toc152589891)

[Fazit zu Evaluation 12](#_Toc152589892)

[Beschreibung Vorgehen für Commits ins Repository 13](#_Toc152589893)

[History der Commits 14](#_Toc152589894)

[Link zur Schnittstellen-Dokumentation 15](#_Toc152589895)

[Tests 16](#_Toc152589896)

[Testkonzept 16](#_Toc152589897)

[Testberichte 17](#_Toc152589898)

[Link zur Schnittstellen-Dokumentation 18](#_Toc152589899)

[Schlussfolgerung 19](#_Toc152589900)

[Meine Erkenntnis 19](#_Toc152589901)

[Fazit 19](#_Toc152589902)

[Schlusswort 19](#_Toc152589903)

[Verzeichnisse 20](#_Toc152589904)

[Abbildungsverzeichnis 20](#_Toc152589905)

[Quellenverzeichnis 20](#_Toc152589906)

Vorwort

Im November/Dezember 2023 war ich Teil eines überbetrieblichen Kurses, der sich auf das Modul 295 bezieht – eine Möglichkeit, die Welt der Backend-Entwicklung für Webanwendungen mit PHP zu erkunden. An jedem Montag trafen wir uns, um die vielfältigen Methoden der serverseitigen Programmierung zu erlernen und zu vertiefen. Während dieses Kurses habe ich kontinuierlich an dieser Dokumentation gearbeitet, sie aktualisiert und erweitert, um meinen Lernfortschritt und die gewonnenen Erkenntnisse festzuhalten.

Meine Absicht hinter dieser Dokumentation liegt darin, einen Einblick in meine Vorgehensweise während des Kurses zu geben – wie ich vorgegangen bin, welche Schritte ich unternommen habe und welche wertvollen Lektionen ich dabei gelernt habe. Diese Aufzeichnungen sollen meine Erfahrungen während des überbetrieblichen Kurses veranschaulichen, indem sie die Herausforderungen, Erkenntnisse und Lösungswege, die ich während dieser Zeit erlebt habe, detailliert darlegen.

In dieser Dokumentation werde ich meine Begegnungen mit verschiedenen Problemen, Lösungsstrategien und möglichen Herausforderungen während des Kurses reflektieren. Es wird um die Auseinandersetzung mit komplexen Konzepten der Backend-Entwicklung gehen – von der Verbindung mit Datenbanken über die Erstellung von APIs bis hin zur Bewältigung von Sicherheitsaspekten und der Implementierung von Authentifizierungsmechanismen.

Zudem werde ich meine Entdeckungen, Gedanken und Schritte festhalten, die ich unternommen habe, um bestimmte Hindernisse zu überwinden oder komplexe Situationen zu lösen. Dies kann den Umgang mit Codefehlern, das Verständnis für neue Programmierkonzepte oder den Einsatz verschiedener Tools und Techniken zur Fehlerbehebung und Optimierung umfassen.

Diese Dokumentation wird als Reflexion meiner Lernreise im Rahmen dieses Kurses dienen, indem sie nicht nur meine Fortschritte aufzeigt, sondern auch die evolutionäre Entwicklung meiner Fähigkeiten und meines Verständnisses im Bereich der PHP-basierten Backend-Entwicklung für Webanwendungen verdeutlicht.

# Projekt M295

## Ziele des ÜKs

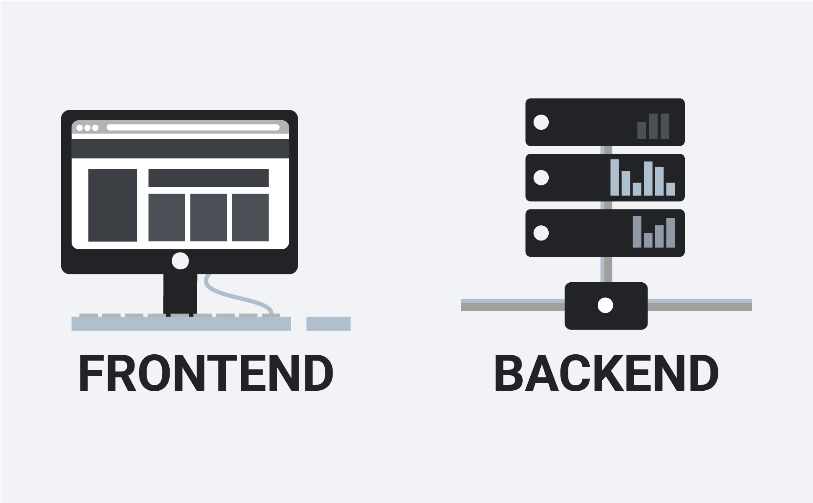
* Richtet die lokale Entwicklungs- und Laufzeitumgebung so ein, dass ein vorgegebenes Projekt entwickelt werden kann.
* Implementiert und dokumentiert mittels vorgegebener Technologie eine effiziente und strukturierte Back-End-Schnittstelle zur Verwaltung einer existierenden Datenquelle.
* Überprüft Zwischenergebnisse mit den Anforderungen und nimmt laufend Korrekturen vor.
* Hält vorgegebene Coderichtlinien ein und überprüft laufend deren Einhaltung.
* Legt Änderungen und Erweiterungen der Implementierung übersichtlich und zuverlässig in einem Softwareverwaltungssystem ab.
* Implementiert im back-End einen aktuellen Authentifizierungsmechanismus und schützt mindestens einen Bereich des Back-Ends vor anonymen Zugriffen.

Abb. 1

## Inhalt des ÜKs

In diesem überbetrieblichen Kurs ging es darum eine Backend Webapplikation zu bauen, wovon wir das Frontend in einem folgenden überbetrieblichen Kurs zusammenbauen. In diesem üK geht es oberflächlich, um das Kreieren von Backend-Schnittstellen. Es gibt auch verschiedene Handlungsziele, die wir am Ende des üKs erfüllen mussten. Darunter gehört eine lokale Entwicklungs- und Laufzeitumgebung einrichten oder auch die Implementation und Dokumentation einer effizienten und strukturierten Back-End-Schnittstelle zur Verwaltung von existierenden Datenquellen. Ebenso gehört ein Authentifizierungsmechanimus für den Schutz eines Back-End Bereichs vor anonymen Zugriffen. Ebenfalls müssen wir zwei verschiedene Arten von Dokumentationen kreieren. Eine davon ist eine physische 10-seitige Dokumentation über den Kurs und das Projekt an sich. Das ist auch diese Dokumentation, die Sie gerade am Lesen sind. Die andere ist eine API-Dokumentation, die mit einem geeigneten Tool zu erstellen ist wie Swagger.io oder Postman. Auch das Verstehen eines Softwareverwaltungsprogramms wie WinSCP gehört dazu und das ganze Projekt wird laufend synchronisiert.

## Datenbankstruktur

### Gebrauch Datenbank

Wir haben zu Beginn als Vorgabe eine Datenbankstruktur bekommen, welche wir in phpMyAdmin umsetzen mussten, sodass eine Datenbank für Kursverwaltung entstand. Das Backend wird nämlich für ein Kursverwaltungstool gebraucht und die Datenbank liefert alle Daten bezüglich Lernenden, Kursleitern oder den Kursen selbst. Uns wurden API (Application Programming Interface) Endpoints gegeben, für welche wir die CRUD-Methoden kreieren mussten. CRUD heisst:

* **C**reate
* **R**ead
* **U**pdate
* **D**elete

Dies mussten wir für all diese Endpoints erstellen:

* /lehrbetriebe/[id]
* /lernende/[id]
* /lehrbetrieb\_lernende/[id]
* /länder/[id]
* /dozenten/[id]
* /kurse/[id]
* /kurse\_lernende/[id]

Dazu gab es noch einen optionalen Endpoint:

* /benutzer/[id] (für Authentifizierung)

### Datenbankaufstellung

Ein Bild, das Text, Reihe, Schrift, Diagramm enthält.

Automatisch generierte Beschreibung

Abb. 2

# Tagesablauf

## Tag 1 des ÜKs

Ein Bild, das Platz enthält.

Automatisch generierte BeschreibungAm ersten Tag des üKs bekamen wir erstmal eine Einführung zum Modul 295 und uns wurde erklärt, um was es in diesem Modul eigentlich geht. Auch die Programmiersprache wurde festgelegt. Undzwar war die Voraussetzung ein PHP-Backend zu erstellen. Das Frontend wird in ReactJS erstellt im nächsten üK, deshalb müssen wir ein funktionierendes Backend für das zukünftige Frontend machen. Wir bekamen ein paar Blätter mit Vorgaben und Aufträgen darauf. Wir mussten eine Datenbank zu einer Kursverwaltungs-Applikation erstellen, mit deren Daten wir dann am Backend arbeiten werden. Ich bereitete mir erstmal meine Datenbank auf und fügte die verschiedenen Tabellen und Spalten ein. Ich achtete stark auf die Benennung der Tabellen und Spalten, damit ich später beim Backend keine grossen CaseSensitive Probleme bekomme. Schlussendlich hatte ich eine Datenbank die bereit war, verwendet zu werden. Ebenfalls begann ich mich besser in PHP einzulesen, da ich noch nicht sehr viele Dinge verstehe oder übersetzen kann. Im Internet konnte ich mir ein paar PHP Snippets ansehen und diese auch ausprobieren. Diese haben mir geholfen einen guten Start zu fassen und ich begann direkt mit der „index.php“ Datei. Dort erstellte ich einen kleinen Grundcode, um die Struktur erstmal zu haben. Eine Datei für die Verbindung zur Datenbank habe ich auch gleich erstellt, dazu auch noch die Datei mit Code für das „Sanitizing“ der verschiedenen Values. Auch die Struktur der Dokumentation habe ich kreiert. Also habe ich am ersten Tag viele Vorbereitungen gemacht, damit ich beim nächsten Tag, nicht mehr alles neu beginnen muss, sondern überall schon ein wenig etwas habe, um nichts zu vergessen bis zum Ende.

Abb. 3

## Tag 2 des ÜKs

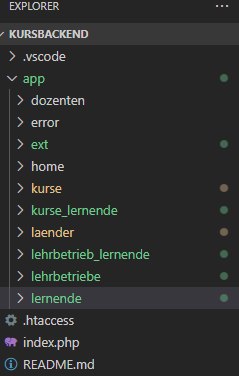
Am zweiten Tag habe ich bereits mit den CRUD-Methoden begonnen. Für die Tabelle „Kurse“ habe ich eine create.php, read.php, update.php und delete.php Datei erstellt. Mit diesen Dateien sollen die CRUD-Methoden möglich gemacht werden, sodass sie auf eine Datenbank angewendet werden können. In der Datei für die Verbindung zur Datenbank, habe ich zuerst mal alle Datenbankelemente, wie Username und Datenbankname, definiert. Somit kann ich anhand von kurzen Variablen, lange und komplizierte Dinge wie das Passwort einfach speichern, ohne alles nochmal abtippen zu müssen. Danach fang ich auch gerade an, die sanitize.php Datei mit Code zu befüllen. Darin habe ich eine Klasse mit zwei Methoden erstellt. Die eine nimmt jeweils ein Stück von „$data“, filtert und wandelt es um, sodass es eine ganze Zahl wird. Die andere Methode nimmt zwei Parameter entgegen, einmal einen Wert, der bereinigt werden soll, und eine PDO-Verbindung, um den Wert für die Verwendung in einer SQL-Abfrage vorzubereiten. Dieses Verfahren hilft dabei, potenziell schädliche Zeichen zu escapen und die Daten vor SQL-Injection-Angriffen zu schützen, indem sie als sichere Zeichenfolge behandelt werden. Nachdem ich diese Datei ebenfalls mit Code gefüllt habe, startete ich eine Struktur für das gesamte Projekt zu erstellen.

Abb. 4

Ich habe für die verschiedenen Tabellen jeweils alleinstehende Ordner gemacht, in der jeweils die CRUD-Methoden zu finden sind. Somit behalte ich einen guten Überblick über mein gesamtes Projekt und kann einfach von Tabelle zu Tabelle navigieren. Am Ende des Tages, schrieb ich etwas an meiner Dokumentation weiter und definierte auch dort bereits eine Vorlage vor. Auch die Aufstellung der Dokumentation habe ich anhand der Vorgaben festgelegt und dazu ein Inhaltsverzeichnis erstellt. Die Titelseite zu der Dokumentation habe ich ebenfalls im gleichen Zug noch gemacht. Anhand von all dem, konnte ich in etwa planen, ob ich die Mindestanforderung von ca. 10 Seiten Dokumentation erfüllen konnte oder nicht.

## Tag 3 des ÜKs (Arbeit Zuhause, weil krank)

## Tag 4 des ÜKs

Am zweitletzten Tag des üKs habe ich mich dazu entschieden ganz viel zeit in die Dokumentation zu investieren. Deshalb fing ich schon am Morgen an, an der Dokumentation zu arbeiten. So habe ich als erstes überlegt, mit welchem Teil der Dokumentation ich starten soll. Ich habe mich dafür entschieden Mal die verschiedenen Tage zu dokumentieren und was ich jeweils gemacht habe. Tag 1 hatte ich bereits fertig und Tag 2 gerade nur begonnen. So versuchte ich mich nochmal daran zu erinnern, was ich zwei Wochen zuvor gemacht habe, und schrieb den Teil für Tag 2. Nachdem Tag 2 fertig geschrieben war, legte ich meinen Fokus auf die Vergleichsmatrix und wie ich diese aufbauen soll. Ich entschied mich für drei Tools, die ich dazu gebrauchen kann. Ich wählte Postman, Swagger und Stoplight. Ich überlegte mir auch, auf welche Punkte ich grossen Wert lag. Dazu dachte ich mir, was wichtig für mich ist bei solchen Tools. So konnte ich gewisse Anforderungen stellen und schrieb diese auch auf. Für die Bewertungsskala habe ich einfach Zahlen von null bis zehn genommen. Wobei null sehr schlecht ist und 10 extrem gut. Dies hielt ich in einer kleinen Excel-Tabelle fest, damit ich die Punkte gut zusammenrechnen konnte. Die Tabelle habe ich noch etwas bunt gemacht, damit man auch von weitem sieht, bei welchem Tool ich gut dabei bin und bei welchem nicht. Eine Tabelle allein reicht für mich persönlich nicht, um Unterschiede zu analysieren. Deshalb habe ich mir noch vorgenommen jeweils ein Diagramm zu erstellen für die Punkteverteilung und die Gesamtpunkte. Das Punkteverteilungsdiagramm zeigt jede Anforderung und die jeweilige Punktzahl der einzelnen Tools anhand von Säulen. So erkennt man gut, welches der Spitzenreiter in der jeweiligen Anforderung ist. Die Säulen habe ich dazu auch in den jeweiligen Hauptfarben des Tools dargestellt. Die Gesamtpunkte habe ich neben ausgerechnet, auch noch in einem Diagramm dargestellt, welches die Gesamtpunkte der einzelnen Tools illustrieren soll. Dazu habe ich ebenfalls Säulen in der Hauptfarbe des Tools benutzt. So konnte ich gut für mich persönlich einen Vergleich zwischen den Tools machen und mir auch den Abstand gut anschauen.

Ein Bild, das Text, Screenshot, Zahl, Diagramm enthält.

Automatisch generierte Beschreibung

Abb. 5

## Tag 5 des ÜKs

# Inhalt

## Vergleichsmatrix der Evaluation des API Dokumentation Tools

### Punkteverteilung

Ich nahm drei Dokumentationstools zum Vergleich und erstellte eine Vergleichsmatrix zu den Tools und bewertete nach gewissen Vorgaben und Wünschen. Aufgeteilt wurden die Vorgaben in sechs verschiedene Abschnitte:

* **Benutzerfreundlichkeit:** Wie einfach steigt man als neuer Benutzer in das Programm ein?
* **Vorwissen:** Wieviel weiss ich über das Tool? / Habe ich schonmal damit gearbeitet?
* **Persönliche Bevorzugung:** Spricht mich etwas persönlich am Tool an?
* **Performance:** Wie gut läuft das Tool? / Gibt es veraltete Versionen? / Gibt es Bugs?
* **Aussehen:** Wie sieht das Tool aus? / Ist es gut übersichtlich?
* **Verständlichkeit:** Verstehe ich das Interface? / Ist alles ziemlich logisch aufgebaut?

Dazu habe ich eine Tabelle, genauso wie ein Diagramm geführt, welches die Punkteverteilung noch besser aufzeigen soll:

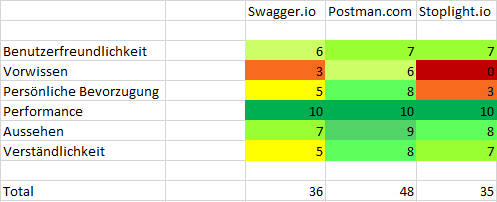


Abb. 6

Abb. 7

### Gesamtpunkte Evaluation

Für die Gesamtpunkte der Vergleichsmatrix, habe ich alle verteilten Punkte bei jedem Dokumentationstool zusammengezählt. Anhand dieser Punkte konnte ich genau sehen, welches Dokumentationstool das Beste für mich ist. Die Gesamtpunkte habe ich ebenfalls in einem Diagramm festgehalten.

* **Swagger.io:** 36 Punkte / 60% der Punkte
* **Postman.com:** 48 Punkte / 80% der Punkte
* **Stoplight.io:** 35 Punkte / ~ 58.33% der Punkte

Abb. 8

**35**

**48**

**36**

Fazit zu Evaluation  
Schlussendlich habe ich mich für das stärkste Dokumentationstool entschieden, also hier in diesem Fall Postman. Postman war für mich schon zu Beginn das Bekannteste der drei Tools. Zugleich gefällt mir das Design fast am besten bei Postman und für mich ist es auch das Verständlichste der drei Tools. Das Interface für den Gebrauch ist bei Postman meiner Meinung nach am besten, da es viele Abschnitte anzeigt, aber dennoch ein relatives offenes und freies Design hat.

Swagger ist mir auch bekannt, jedoch habe ich es nur ein- oder zweimal bis jetzt gebraucht, also bin ich noch nicht sehr vertraut mit dem ganzen Tool. Zudem finde ich es nicht so leicht zu verstehen wie Postman, da es ein stärker abgeändertes Interface hat wie Postman oder Stoplight. Was dafür hier etwas besser ist als bei Postman, ist dass man so ziemlich alles, was man braucht auf einem Fleck hat. Die Funktionen wurden alle in einen Teil der Navbar gesteckt, während es bei Postman und Stoplight oben und links eine Navbar hat, was einen verwirren könnte.

Stoplight teilt ein ähnliches Interface wie Postman, aber von diesem Tool habe ich zuvor noch nie gehört und deshalb auch nie gebraucht. Das User-Interface ähnelt dem von Postman und ist ebenfalls relativ offen und frei gestaltet. Da ich es jedoch noch nie gebraucht habe, würde ich mir nicht zutrauen dies gleich für ein Projekt zu verwenden. Vielleicht ist es besser als gedacht, aber dazu müsste ich mir das Tool ganz genau anschauen und austesten.

## Beschreibung Vorgehen für Commits ins Repository

## History der Commits

## Link zur Schnittstellen-Dokumentation

# Tests

## Testkonzept

## Testberichte

## Link zur Schnittstellen-Dokumentation

# Schlussfolgerung

## Meine Erkenntnis

## Fazit

## Schlusswort

## Quellenverzeichnis

Hier werden die Bilder und Abbildungen mit einer Beschreibung und entsprechender Quelle:

[Abb. 1 4](file:///C:\Users\lukas.sutter\Desktop\Dokumentation.docx#_Toc152590641)

"Zeichen für Frontend-Entwicklung und Backend-Entwicklung"  
Quelle: [Как устроен frontend и backend — на примере создания сайта / Skillbox Media](https://skillbox.ru/media/code/frontend_i_backend/)

[Abb. 2 5](file:///C:\Users\lukas.sutter\Desktop\Dokumentation.docx#_Toc152590642)

"Datenbankstruktur nach Erstellung auf PHPMyAdmin"  
Quelle: eigene Kreation mit Datenbank-Designer auf PHPMyAdmin

[Abb. 3 6](file:///C:\Users\lukas.sutter\Desktop\Dokumentation.docx#_Toc152590643)

"Abbildung von Backend zu React Frontend durch JSON"  
Quelle: Modul 295 - Backend für Applikationen realisieren (Arbeitsblatt)

[Abb. 4 7](file:///C:\Users\lukas.sutter\Desktop\Dokumentation.docx#_Toc152590644)

"Ordnerstruktur innerhalb des Projekts mit allen Tabellen"  
Quelle: eigene Kreation innerhalb VS Code

[Abb. 5 9](file:///C:\Users\lukas.sutter\Desktop\Dokumentation.docx#_Toc152590645)  
"Kreation von Tabelle und Diagramme für Dokumentationstools Evaluation"

Quelle: eigene Kreation innerhalb MS Excel

[Abb. 6 11](file:///C:\Users\lukas.sutter\Desktop\Dokumentation.docx#_Toc152590646)

"Tabelle für Punktzahlberechnung für Evaluation Dokumentationstools"  
Quelle: eigene Kreation innerhalb MS Excel

[Abb. 7 11](file:///C:\Users\lukas.sutter\Desktop\Dokumentation.docx#_Toc152590647)

"Punkteverteilung auf Diagramm gezeigt"  
Quelle: eigene Kreation innerhalb MS Excel

[Abb. 8 12](file:///C:\Users\lukas.sutter\Desktop\Dokumentation.docx#_Toc152590648)

"Gesamtpunktzahlen der Dokumentationstools in Diagramm gezeigt"  
Quelle: eigene Kreation innerhalb MS Excel