Webprojekt-Bibliothek

Inhalt

[Backend 2](#_Toc184584656)

[Routen (web.php) 2](#_Toc184584657)

[Models 2](#_Toc184584658)

[Book.php 2](#_Toc184584659)

[Lending.php 3](#_Toc184584660)

[Librarian.php 3](#_Toc184584661)

[Migrations 3](#_Toc184584662)

[Books Migration (create\_books\_table.php) 4](#_Toc184584663)

[Librarian Migration (create\_librarians\_table.php) 4](#_Toc184584664)

[Lendings Migration (create\_lendings\_table.php) 5](#_Toc184584665)

[Testdaten einfügen mit Factories und DatabaseSeeder 5](#_Toc184584666)

[BookFactory.php 5](#_Toc184584667)

[LibrarianFactory.php 6](#_Toc184584668)

[DatabaseSeeder.php 6](#_Toc184584669)

[Frontend 7](#_Toc184584670)

[Vue/Inertia Initialisierungsdatei (app.ts) 7](#_Toc184584671)

[Hauptapp (app.blade.php) 7](#_Toc184584672)

[Vue-Pages 7](#_Toc184584673)

[Home.vue 7](#_Toc184584674)

# Backend

## Routen (web.php)

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

Automatisch generierte Beschreibung

In dieser Datei werden die Routen festgelegt, also welche Seite (oder welche Daten generell) bei einer bestimmten URL zurückgegeben wird.

## Models

Laravel nutzt das MVC (Model-View-Controller)- Modell. Laravel verfügt auch über eine Object-Relational Mapper (ORM), welcher die Datensätze aus der verknüpften Tabelle mit PHP Objekten synchronisiert. Dadurch müssen keine dedizierten SQL-Abfragen erstellt werden und auf die Daten in der Datenbank kann einfach durch PHP-Objekte zugegriffen werden. In Laravel heißt der ORM „**Eloquent**“.

* Model – das Objekt, welches mit der Datenbank gemappt wird
* View – in diesem Fall die Haupt-Blade-Date (also Haupt-HTML Seite – wird später noch behandelt)
* Controller – die Controller dienen als Schnittstelle zwischen Model und View. Sie kontrollieren, wie die Daten in den Views angezeigt werden.

In Laravel werden die Beziehung und andere Spezifikationen in der Model Datei konfiguriert.

### Book.php

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

Automatisch generierte Beschreibung

### Lending.php

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

Automatisch generierte Beschreibung

### Librarian.php

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

Automatisch generierte Beschreibung

## Migrations

Im Laravel-Framework wird die Datenbank mit Migrations erstellt. Damit kann dieselbe Datenbankstruktur auf verschieden Systemen mit einem Befehl erstellt werden. In Laravel geht das mit dem Befehl „php artisan migrate**“.** Für jede Tabelle wird eine migration PHP Datei erstellt, in der die Spalten und dazugehörige Datentypen angegeben werden:

### Books Migration (create\_books\_table.php)

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

Automatisch generierte Beschreibung

### Librarian Migration (create\_librarians\_table.php)

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

Automatisch generierte Beschreibung

### Lendings Migration (create\_lendings\_table.php)

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

Automatisch generierte Beschreibung

## Testdaten einfügen mit Factories und DatabaseSeeder

Mit Factories und der **faker-Bibliothek** können für Tabellen Testdatensätze automatisiert erstellt werden.

Der DatabaseSeeder erlaubt es, beim migrieren von der Datenbank direkt die Tabellen programmatisch zu befüllen.

### BookFactory.php

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

Automatisch generierte Beschreibung

### LibrarianFactory.php

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

Automatisch generierte Beschreibung

### DatabaseSeeder.php

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

Automatisch generierte Beschreibung

## Tabellen

### Bücher

Ein Bild, das Text, Software, Multimedia-Software, Grafiksoftware enthält.

Automatisch generierte Beschreibung

### Bibliothekare

Ein Bild, das Multimedia-Software, Software, Grafiksoftware, Screenshot enthält.

Automatisch generierte Beschreibung

### Ausleihen

Ein Bild, das Screenshot, Text, Multimedia-Software, Software enthält.

Automatisch generierte Beschreibung

# Frontend

## Vue/Inertia Initialisierungsdatei (app.ts)

Diese TypeScript-Datei regelt, wie die InertiaApp funktioniert.  
Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

## Hauptapp (app.blade.php)

Weil die Applikation eine SPA ist, gibt es eine Haupt-HTML Seite (oder im Fall von Laravel .blade Seite), auf der alle DOM-Elemente dynamisch ausgetauscht werden.  
Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

## Vue-Pages

Die einzelnen Seiten befinden sich in Form von Vue-Komponenten im ***resources/js/Pages*** Verzeichnis von der Laravel App. Mit dem Inertia-Router kann zwischen diesen Pages navigiert werden.

### Home.vue

Die einzige Seite, die bis jetzt existiert, ist Home.vue. Diese dient nur zum Testen, ob das Frontend überhaupt funktioniert und zeigt noch keine relevanten Information an.

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

Automatisch generierte Beschreibung