

# Laravel Kurzreferenz

## Installation

Composer muss installiert sein. Im Kommandozeilen-Tool (mit Admin-Rechten) zum Ordner, in welchem das Projekt angelegt werden soll wechseln und folgenden Code eingeben:

```
composer create-project laravel/laravel: example-app
```

wobei `example-app` der Name des Projektes (= Ordnername) sein soll.

Für eine bestimmte Laravel-Version muss die Versionsnummer angegeben werden.

```
composer create-project laravel/laravel:^9.0 example-app
```

## Routen definieren

Routen werden in der Datei `routes/web.php` definiert

Die Syntax ist:

```
Route::get('[URL]', function () {  
    return [View oder Ausgabe];  
});
```

z.B.

```
Route::get('/', function () {  
    return view('welcome');  
});
```

Die Route für REST ist:

```
Route::resource('[URL]', [Custom]Controller::class);
```

Soll eine bestimmte Methode der Controller-Klasse aufgerufen werden, so ist der optionale Parameter `[Methode]` mit anzugeben.

```
Route::get('[URL]', [Custom]Controller::class, '[Methode]');
```

z.B.

```
Route::resource('/songs', SongsController::class);
```

mit Methoden-Aufruf

```
Route::get('/songs', SongsController::class, 'show');
```

## Controller anlegen

```
php artisan make:controller [NameDesControllers] [--resource]
```

z.B.

```
php artisan make:controller InfosController --resource
```

Wird die Option `--resource` hinzugefügt, werden im Controller die Methoden für die Administration z.B. mit Datenbanken hinzugefügt. Ohne diese Option wird „nur“ ein leerer Controller angelegt.

## Migrations anlegen

```
php artisan make:migration create_[name_der_tabelle]_table
```

z.B.

```
php artisan make:migration create_songs_table
```

**Der Tabellen-Name muss immer im Plural angelegt sein, damit die Zugehörigkeit zum Model gewährleistet ist!**

## Model anlegen

```
php artisan make:model [Model-Name]
```

z.B.

```
php artisan make:model Song
```

**Der Model-Name muss immer im Singular angelegt sein, damit die Zugehörigkeit zur Datenbank-Tabelle gewährleistet ist.**

## Model und Migration gleichzeitig anlegen

```
php artisan make:model [Model-Name] -m
```

z.B.

```
php artisan make:model Song -m
```

## Migration durchführen

```
php artisan migrate
```

## Migration zurücksetzen

```
php artisan migrate:rollback
```

## Factory anlegen

```
php artisan make:factory [Factory-Name] --model=[Model-Name]
```

z.B.

```
php artisan make:factory SongFactory --model=Song
```

## Seeder anlegen

```
php artisan make:seeder [Seeder-Name]
```

z.B.

```
php artisan make:seeder SongSeeder
```

Datenbank Resetten und Testdaten gleichzeitig anlegen

```
php artisan migrate:fresh --seed
```