Moduleweek 4 – Opdracht 1

Songs opdracht uitbreiden met een Model, migration en seeder

# Migration

In deze opdracht realiseer je de structuur van de database a.d.v. een migration en leg je de koppeling in de .ENV file.

**Stap 1:** ga naar PhpMyAdmin en maak hier de database lv1\_week4 aan.

**Stap 2:** ga naar je .ENV file en pas hier de database settings aan. Gebruik als naam de database van stap 1.

**Stap 3:** Maak een migration aan genaamd “CreateSongsTable”. Zorg dat je in deze migration de volgende velden aanmaakt:

**Tip 1:** <https://laravel.com/docs/9.x/migrations> **Tip 2:** php artisan make:migration naamVdMigration  
**Tip 3:** Wat doet $table->timestamps(); ?

|  |  |  |
| --- | --- | --- |
| **Naam veld** | **Type veld** | **Bijzonderheden** |
| id | Int | Pkey, AI |
| title | String | Max length = 100, not null |
| singer | String | Nullable |
| created\_at | DateTime | Not null |
| updated\_at | DateTime | Not null |

**Stap 4:** run het commando “php artisan migrate” en controleer of de songs tabel met de juiste structuur aangemaakt is in de database.

# Seeder

In deze opdracht gaan we ervoor zorgen dat de tabel “songs” ook daadwerkelijk data bevat a.d.v. een seeder.

**Stap 1:** maak een seeder aan genaamd “SongsTableSeeder”.

**Tip 1:** php artisan make:seeder naamVdSeeder

**Stap 2:** zorg dat je in deze SongsTableSeeder jouw top 5 favoriete nummers op neemt.

**Tip 2:** <https://laravel.com/docs/9.x/seeding>

**Stap 3:** Zorg dat je het commando “php artisan db:seed” uitvoert en controleer of de songs tabel nu ook daadwerkelijk jouw top 5 favoriete nummers bevat.

# Model

In deze opdracht gaan we het “Song” model maken. Dit model is nodig voor de communicatie met de database.

**Stap 1:** maak het “Song” model aan.

**Tip 1:** php artisan make:model naamVhModel

**Stap 2:** configureer het model goed. Controleer hierbij of de table, primary key, primary key type en timestamps goed staan.

**Tip 2:** kijk eens welke klasse jouw model extend en wat daarin gebeurd. Bij PhpStorm kan je middels ctrl + klikken door klikken op een klasse.

# Data ophalen d.m.v. het model

In deze opdracht gaan we de statische array $songs vervangen door de daadwerkelijke data die uit de database komt.

**Stap 1:** Zorg dat de index methode van de Controller de lijst van titels weergeeft uit de database (alleen de titel, niets meer!).

**Tip 1:** in de index methode van de Controller haal je alle songs op d.m.v. modelNaam::all();

**Stap 2:** Zorg dat de show methode van de Controller de details van een song op basis van het meegeven id in de URL weergeeft. Hieronder verstaan we: title en singer.

**Tip 2:** in de show methode van de Controller haal je alle songs op d.m.v. modelNaam::find($id);

**Stap 3:** Zorg ervoor dat je naar de show methode van de controller gaat als je in de index op een titel klikt (de titel waar je op klikt moet ook getoond worden in de show).

# Extra uitdaging:

**Stap 1:** maak een model aan met php artisan make:model andereNaam --migration. Wat is er nu extra aangemaakt?

**Stap 2:** er zijn veel commando’s om je leven als programmeur wat gemakkelijker te maken. Kijk eens naar: <https://laravel.com/docs/9.x/eloquent#generating-model-classes>

**Stap 3:** zorg dat het binnen jouw applicatie mogelijk is om een song aan te maken.

**Tip 1:** kijk welke actie je nodig hebt: [https://laravel.com/docs/9.x/controllers#actions-handled-by-resource-controller](https://laravel.com/docs/8.x/controllers" \l "actions-handled-by-resource-controller)

**Tip 2:** vergeet @csrf niet!