### MVC met Laravel

Opdracht: **MVC routing en formulier validatie**

**Leerdoelen**

Herhaling

* Weten wat MVC is.
* Weten wat de voordelen van MVC zijn.
* Views kunnen maken en gebruiken.
* Controllers kunnen maken en gebruiken.

Nieuw

* Routes op kunnen zetten voor zowel POST als GET requests.
* Het verschil tussen POST en GET request begrijpen.
* Formulieren kunnen versturen naar de controllers.
* Laravel gebruiken om formulieren te beschermen tegen Cross-site\_request\_forgery aanvallen.
* Formulier validatie kunnen toepassen op de Laravel manier.

**Werkwijze**

Je maakt de opdrachten zelfstandig. Niet alles is voorgekauwd. Je zult dus zelf informatie op moeten zoeken op het internet. Ook kun je terugkijken naar de informatie in de presentatie.

## Introductie

Tijdens de komende opdrachten ga je een formulier (<form>) maken waarmee nieuwe songs aan de bibliotheek toegevoegd kunnen worden.

Hiervoor ga je een formulier maken, een controller en route opzetten om dit formulier te behandelen, en de ingevoerde data van het formulier valideren.

## Opdracht 2: Het maken van een formulier (<form>).

Je gaat een formulier maken waarmee een nieuwe single aan de muziekbibliotheek toegevoegd kan worden. In opdracht 3 ga je dit formulier functioneel maken.

Maak eerst een nieuwe pagina aan met daarop een formulier om een single toe te voegen. De volgende informatie moet ingevuld kunnen worden:

* Een invoerveld voor de naam van de single.
* Een invoerveld voor de naam van de artiest of band.
* Een invoerveld voor het jaar waarop de single released is.
* Een submit knop waarmee het formulier verstuurd wordt.  
    
    
  *Voorbeeld van het “nieuwe single” formulier*  
  Graphical user interface, text, application

  Description automatically generated

Om het formulier te versturen ga je het volgende nodig hebben:

1. Een store() methode die gebruikt gaat worden om de ingevulde gegevens uit de HTTP POST request te halen.
2. Een nieuwe HTTP post route die gebruikt kan worden om het formulier naar de store() metode te versturen.

De volgende 2 bronnen kunnen je hierbij helpen:

* Het maken van een HTTP POST route:  
  [Routing - Laravel - The PHP Framework For Web Artisans](https://laravel.com/docs/9.x/routing#available-router-methods)
* Informatie uit een HTTP POST request ophalen in een controller:  
  <https://laravel.com/docs/9.x/requests#accessing-the-request>

Vervolgens kun je Laravel Blade gebruiken om het formulier te versturen naar de aangemaakte route via de route() functie. Kijk hiervoor naar het voorbeeld hieronder.

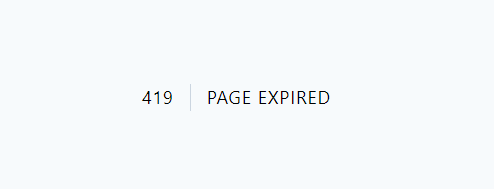
*Voorbeeld van formulieren versturen via Laravel Blade*  
****

Zorg dat je het “nieuwe song” formulier naar de store() functie op de SongController stuurt via een HTTP POST request. **Dit geeft een 419|Page expired error.**

In de volgende opdracht leer je waarom je deze error ziet; en hoe je hem oplost!

## Opdracht 3: Cross-site request forgery aanvallen

Als je het “nieuwe song” formulier verstuurt krijg je momenteel de volgende error:



Deze error krijg je omdat Laravel je forceert om je websites te beschermen tegen Cross-site request forgery aanvallen.

Op de volgende pagina kun je lezen wat deze aanval is; én hoe je met Laravel je website kunt beschermen tegen Cross-site request forgery aanvallen: <https://laravel.com/docs/9.x/csrf#csrf-introduction>

**Doe het volgende:**

1. Beschrijf kort in je eigen woorden wat een Cross-site request forgery is.
2. Zorg dat je Laravel systeem beschermd is tegen deze aanvallen. Zodat het “nieuwe song” formulier verstuurd kan worden.
3. Toon de data die door de gebruiker ingevoerd is op het scherm via een echo() **of** een var\_dump(). Je zult hiervoor moeten werken met het Laravel Request object.   
   Op de volgende pagina kun je vinden wat dit Request object is:  
   [*https://laravel.com/docs/9.x/requests#accessing-the-request*](https://laravel.com/docs/9.x/requests#accessing-the-request)

## Opdracht 4: Valideren van de ingevulde gegevens

Je hebt nu een formulier gemaakt waarmee een single aan het systeem toegevoegd kan worden. En je hebt het formulier beschermd tegen Cross-site\_request\_forgery aanvallen.

Nu ga je de informatie die ingevuld wordt valideren. Ook hiervoor heeft Laravel iets handigs! Namelijk de Validate() methode van het Request object.

**Doe het volgende**

1. Lees de volgende bron door voor informatie over de validatie functie van het request object: <https://laravel.com/docs/9.x/validation#quick-writing-the-validation-logic>
2. Pas de volgende validatieregels toe op het nieuwe song formulier:
   * Alle velden zijn verplicht
   * De naam van single moet minstens 3 karakters, en maximaal 32 karakters lang zijn.
   * De naam van artiest moet minstens 6 karakters, en maximaal 32 karakters lang zijn.
   * Het jaar waarop de single released is moet een nummer zijn.  
       
     Tip: in de volgende bron zijn alle validatieregels te vinden:  
     <https://laravel.com/docs/9.x/validation#available-validation-rules>
3. Toon alle validatiefouten op het scherm. Hiervoor kun je gebruik maken van een Laravel Blade foreach loop. Zie de volgende bron voor meer informatie:  
   <https://laravel.com/docs/9.x/validation#quick-displaying-the-validation-errors>

## Opdracht 5: Inleveren

Lever de volgende bestanden in wanneer je klaar bent met de opdrachten:

1. De web.php file.
2. Alle gemaakte controllers.
3. Alle gemaakt views.
4. De antwoorden op de gestelde vragen

Je werkt tijdens deze opdrachten door in het systeem dat je vorige week gemaakt hebt.

## Opdracht 1: Vragen over HTTP en Routes.

Bekijk eerst de volgende documentatie: <https://laravel.com/docs/9.x/routing#available-router-methods>

Beantwoord daarna de volgende 3 vragen:

1. Welke route methodes kunnen gebruikt worden in het Laravel framework?

GET, POST, PUT, DELETE, PATCH en OPTION

1. Wat is het verschil tussen een HTTP GET request en een HTTP POST request?

GET haalt gegevens op terwijl POST gegevens naar de server stuurt om iets te creeren of bij te werken

1. Welke HTTP request methode wordt gebruikt om een formulier (<form>) te versturen?

Bij altijd post