

Refactoring

Inleiding

We hebben een hele eenvoudige Read gemaakt van de producten.

We gaan nu een nette output maken, maar voordat we dat gaan doen gaan we eerst onze code netjes opdelen in controller en view (=refactoring). Nu hebben we (om te testen) even een hele eenvoudige output in de routing file gezet. Leuk om te testen, maar als we alle code zo zouden maken dan krijgen we een hele grote overzichtelijke web.php

We gaan dus eerst de code die we nu hebben op de juiste plaats zetten. Daarna gaan we onze output fraaier maken.

Controller

We maken eerst een controller aan voor product.

```
php artisan make:controller ProductController
```

In de controller zetten we de volgende functie.

```
public function read() {  
    $products = \App\Models\Product::all();  
    return view('product-read', ['products' => $products]);  
}
```

De functie verwijst naar de `product-view` en geeft de variabele `$products` mee.

View

We maken in de views folder een nieuw bestand `product-read.blade.php`

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```

    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
    <?php dd($products); ?>
</body>
</html>

```

Met de dd() functie zien we de inhoud van \$products. Dat ziet er ongeveer zo uit. Maar voordat je dat ziet zal je eerst de route moeten aanpassen.

```

Illuminate\Database\Eloquent\Collection {#303 ▼ // resources\views\product-read.blade.php
  #items: array:10 [▼
    0 => App\Models\Product {#305 ▼
      #connection: "mysql"
      #table: "product"
      #primaryKey: "id"
      #keyType: "int"
      +incrementing: true
      #with: []
      #withCount: []
      +preventsLazyLoading: false
      #perPage: 15
      +exists: true
      +wasRecentlyCreated: false
      #escapeWhenCastingToString: false
      #attributes: array:6 [▼
        "id" => 1
        "name" => "Logitech M90 muis met kabel"
        "description" => "De Logitech M90 is een bekabelde, optische muis met een goede
reactiesnelheid (1000 dpi). Door de gebruiksvriendelijke, symmetrische vorm van de muis
bedient u ►"
        "amount" => "7.50"
        "created_at" => null
        "updated_at" => null
      ]
      .....
      .....

```

Route

In web.php zet je de volgende route

```
// plaats dit bovenaan in de routing file (web.php)
use App\Http\Controllers/ProductController

//plaats dit bij de andere routes in de routing file
Route::get('/products', [ProductController::class, 'read']);
```

Hiermee koppel je de url */products* aan de controller. En de controller gaat naar de view en de view laat de dd() zien.

Zorg er voor dat de oude route naar */products* verwijderd wordt want je kunt geen twee dezelfde routes hebben want dan weet Laravel niet waar je heen wil.

In de volgende les gaan we de view goed maken.

Inleveren

1. Screenshot van je browser waarin je de dd() laat zien. Zoals in het voorbeeld hierboven bij de view.
Zorg dat jouw eigen data uit jouw database te zien is.

Voorbeeld:

[image-1666306821592.png](#)

--

Revision #8

Created 2022-10-20 22:23:48 UTC by Max

Updated 2022-10-28 19:06:16 UTC by Max