Create

Inleiding

We hebben nu een werkende read; we kunne de inhoud van een tabel op het scherm tonen.

We het model uitbreiden, routes maken en een view maken zodat we een record aan de tabel in de database kunnen toevoegen.

Create Route

We beginnen met een nieuwe route.

```
Route::get('stocks/create', [App\Http\Controllers\StockController::class, 'create']);
```

Controller

Vervolgens maken we een functie create in de StockController file.

```
public function create()
{
    return view('stocks.create'); // -> resources/views/stocks/create.blade.php
}
```

View

We maken even een tijdelijke view.Maak een nieuwe file resources/views/stock/create.blade.php en plaats er even tijdelijk code in, bijvoorbeeld.

```
<h1>Topper!</h1>
```

In de andere view die we al hadden, de file resources/views/stock/index.blade.php plaatsen we de volgende code vlak boven de <tabel>

```
<h1 class="display-3">Stocks</h1>
<div>
  <a href="/stocks/create" class="btn btn-primary mb-3">Add Stock</a>
</div>
```

Testen

Op de stocks pagina (localhost:8000/stocks) staat nu een Create button.

Wat doet deze button?

De button gaat naar /stocks/create.

De route *stocks/create* verwijst naar de *create()* functie in de *StockController* en die verwijst naar de view *resources/views/stock/create.blade.php*

Dus in het kort:

> Create Button -> /stocsk/create -> (route) StockController function create() ->(view) create.blade.php

De routing werkt nu, maar er wordt nog geen nieuwe record aangemaakt.

We gaan nu eerst de create view aanpassen zodat er een form wordt getoond waarin we een nieuw record kunnen aanmaken.

Input form

Plaats in de create.blade.php view file nu de volgende code.

```
<!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">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
    <title>Stocks</title>
```

```
</head>
<body>
  <div class="container" style="margin:40px;">

  <h1 class="display-3">Stocks</h1>

    <form method="post" action="/stocks/store">
        @csrf
        <div class="form-group">
            <label for="stock_name">Stock Name:*</label>
            <input type="text" class="form-control" name="stock_name"/>
        </div>

        <div class="form-group">
            <label for="ticket">Stock Ticket:*</label>
            <input type="text" class="form-control" name="ticket"/>
        </div>

        <div class="form-group">
            <label for="value">Stock Value:</label>
            <input type="text" class="form-control" name="value"/>
        </div>
        <button type="submit" class="btn btn-primary">Add Stock</button>
    </form>

  </div>

</body>
</html>
```

Store Route

Het form doet een post naar /stocks/store. We gaan hiervoor een route toevoegen (in de web.php).

```
Route::get('stocks/store', [App\Http\Controllers\StockController::class, 'store']);
```

Store functie Controller

De store functie in de controller ziet er als volgt uit.

```
public function store(Request $request)
{
    // Validation for required fields (and using some regex to validate our numeric value)
    $request->validate([
        'stock_name'=>'required',
        'ticket'=>'required',
        'value'=>'required|max:10|regex:/^-?[0-9]+(?:\.[0-9]{1,2})?$/'
    ]);
    // Getting values from the blade template form
    $stock = new Stock([
        'stock_name' => $request->get('stock_name'),
        'ticket' => $request->get('ticket'),
        'value' => $request->get('value')
    ]);
    $stock->save();
    return redirect('/stocks')->with('success', 'Stock saved.');    // -> resources/views/stocks/index.blade.php
}
```

Validatie

In regel 4 tot en met 8 worden de velden gevalideerd. Omdat dit in de controller op de server gebeurt is dit een server-side validatie. Je kunt ook via HTML valideren, dat heet cliënt-side validatie en is minder veilig.

Nieuw record

Regel 10 tot en met 14 maakt een nieuw record aan en vult dat met de gevalideerde waarden uit het form.

Regel 15 slaat de record op in de database.

Regel 16 gaat terug naar de /stocks url. Via de route wordt hierdoor de index() functie aangeroepen en worden alle regels getoond.

Het tweede gedeelte van de de regel zorgt ervoor dat er een boodschap op het scherm verschijnt.

--

Revision #7
Created 29 October 2022 15:32:55 by Max
Updated 29 October 2022 21:25:32 by Max