Create en Edit

We gaan de exit en create controls en bijbehorende templates maken voor onze links applicatie.

We beginnen met de edit en die is lastig. De create hadden we al gemaakt en lijkt qua form heel erg op de edit. Dus als we de edit hebben dan is de create ook bijna vanzelf af.

Aan het eind van deze les heb je een werkende CRUD applicatie.

Create

Voor de create functie gaan we de routing, control en view opzetten.

Routing

Voor de routing verwijzen we naar de view create in de links directory, we plaatsen dus de code:

```
return view('links.create');
```

in de create control.

Wrapper

De view wordt weer een beetje complex, we maken namelijk één form die we gaan gebruiken voor create en voor edit. We maken voor create en voor edit een eigen template maar roepen vanuit deze template deze ene form-template aan. Dit wordt ook wel een *wrapper* genoemd. We verpakken de form template als het ware door een andere template. De verpakking (wrapper) bepaald wat het wordt (edit of create) maar de inhoud van de verpakking (wrapper) is gelijk omdat eht form voor edit en create namelijk hetzelfde zijn.

Edit

We plaatsten in de edit template de volgende code (dit is de wrapper voor edit).

```
@extends('layouts.app')

@section('content')

<div class="container">
```

```
    <form action="{{ url('/links/' . $link->id)  }}" method="POST" enctype="multipart/form-
data">
        {{ csrf_field() }}
        {{ method_field('PATCH') }}
        @include('links.form', ['mode' => 'edit'])
    </form>
</div>

@endsection
```

Belangrijk hier is de @include, daar wordt de template links/form.blade.php included en er wordt een parameter (mode=edit) meegegven zodat het form links/form.blade.php 'weet' dat het aangeroepen is vanuit de create.

In het form.blade.php file beginnen we met:

```
{{ $mode == 'create' ? 'Create Link' : 'Modify Link' }} <br>
```

Dit is gewoon een soort if statement, je kunt dit herschrijven als volgt:

```
if ($mode=='create') {
echo "Create Link";
}else{
echo "Modify Link";
}
```

Onder de eerste regel in form.blade.php plaatsen we de voglende code:

```
<div class="form-group">
    <label for="title" class="control-label">{{'Title'}}</label>
    <input type="text" class="form-control" name="title" id="title" value="{{ isset($link-
>title) ? $link->title : '' }}">
</div>

<div class="form-group">
    <label for="url" class="control-label">{{'URL'}}</label>
    <input type="text" class="form-control" name="url" id="url" value="{{ isset($link->url) ?
$link->url : '' }}">
</div>

<div class="form-group">
```

```
    <label for="description" class="control-label">{{'Description'}}</label>
    <textarea type="text" class="form-control" name="description" id="description">{{
isset($link->description) ? $link->description : '' }}</textarea>
</div>

<input type="submit" class="btn btn-success" value="{{ $mode == 'Create' ? 'Create' : 'Update'
}}">

<a class="btn btn-primary" href="{{ url('links') }}">Cancel</a>
```

> Vraag 1, de php functie isset wordt een paar gebruikt in deze template, wat doet deze en waarom is deze nodig?

> Vraag 2, de button tekst (regel 16) is variabel. Wat kan de button tekst worden en waar hangt dat van af?

Nu gaan we naar de control kijken die wordt aangeroepen als we de edit knop indrukken. In de tabel voor alle routing kun je zien dat de standaard url voor een edit is /link/ID/edit. Zoek in de tabel weer op welke control er wordt aangeroepen. Ga naar deze control en plaats daar de volgende code:

```
$link = Link::findOrFail($id);
// echo print_r(compact('link'));
 return view('links.edit', compact('link'));
```

$link is een object en de view heeft een assiociatieve array nodig met één element link dat verwijst naar het object. Dit is wat de compact functie doet (https://www.quora.com/What-does-compact-do-in-Laravel).

Je kunt met de echo die in commentaar staat zien hoe dat werkt.

We hebben nu het juiste form en kunnen de rows van de link wijzigen, maar wat gebeurt er als je op Update drukt? In het form (de wrapper) staat iets bijzonders namelijk {{ method_field('PATCH') }} Dit zorgt ervoor dat de action van het form geen POSTof GET is maar PATCH. Zoek in de tabel op welke method daarbij hoort in de controller.

Plaats dan deze code bij deze method:

```
//dd($request);
$modifiedLink = request()->except(['_token','_method']);
```

```
//dd($modifiedLink);
Link::where('id', '=', $id)->update($modifiedLink);
return redirect('links');
```

Met de regels die in commantaar staan kan je zelf uitzoeken wat er precies gebeurt.

Als het goed is heb je nu een werkende edit functie.

De Edit Samengevat

Samengevat, de edit in vijf stappent:

Stap	Vanaf hier	Actie	Eindigt hier
1	index.blade.php	/links/{ID}/edit (via knop)	linksController@edit
2	linksController@edit	edit.blade.php	wrapper voor form.blade.php
3	form	/links/{ID} (method PATCH)	linksController@update
4	linksController@update	doet **update** en gaat naar /links	linksController@index
5	linksController@index	index.blade.php	

Create

De create code hadden we al eens gemaakt en zag er als voglt uit:

```
$link = new Link;
$link->title = $request->title;
$link->url = $request->url;
$link->description = $request->description;
$link->save();
```

Zoek in de tabel op waar deze code moet in de controller moet staan en plaats deze code. Let op dat als de code klaar is dat je weer terug wilt naar je overzichtpagina. Plaats hiervoor de juiste code op regel 6, dus na de sav().

We hoeven alleen nog maar een wrapper voor de create te maken, deze lijkt op de wrapper voor de edit, maar is net iets anders:

```
@extends('layouts.app')
```

```
@section('content')

<div class="container">
    <form action="{{ url('/links') }}" class="form-horizontal" method="post"
enctype="multipart/form-data">
        {{csrf_field()}}
        @include('links.form', ['mode' => 'create'])
    </form>
</div>

@endsection
```

De verschillen met de wrapper van de edit zijn:

1. de form action is anders want er wordt geen ID meegegeven; een nieuwe entry heeft namelijk nog geen ID.
2. de 'variabele' mode is anders zodat het form 'weet' dat het een create form en geen edit form is.

Opdracht

In het create form staat een button en de tekst is 'update'. Dit was niet de bedoeling, de tekst moest afhankelijk zijn van de wrapper die het form aanroept: bij een update zou de tekst *Update* moeten zijn en bij create zou de tekst *Create* moeten zijn. Kijk wat er fout is en los dit bugje op.

Revision #13
Created 2019-10-29 19:28:44 UTC by Admin
Updated 2019-12-08 19:59:58 UTC by Admin