

Validation

In deze les leren we hoe we back-end validation kunnen opzetten met Laravel

Front End Validation

Via het keyword `required` kun je (met Bootstrap) aangeven dat een veld verplicht is. Hiervoor wordt JavaScript gebruikt. Omdat een gebruiker JavaScript in de browser kan uitzetten, kan de Front End Validation worden 'uitgezet'. Het is dus ook van belang dat je back End Validation implementeerd.

Back End Validation

We gaan een request control maken om de back-end validation te implementeren, dat doen we met het volgende CLI commando.

```
php artisan make:request LinkRequest
```

De php file `app/Http/Requests/LinkRequests` is nu aangemaakt. Open deze.

We gaan eerst naar de rules kijken en laten we de velden `title` en `url` verplicht maken. In de function `rules` voeren we dan het volgende in:

```
return [  
    'url' => 'required|url',  
    'title' => 'required'  
];
```

Verder moeten we de return value van de functie `authorize()` op `true` zetten. met deze functie kunnen we logica inbouwen die aangeeft of de gebruiker de update mag uitvoeren of niet.

In de `LinkController` vervangen we de `Request` in de `update()` en in de `store()` in `LinkRequest` en vervolgens zetten we dit bovenaan in de `LinkController`:

```
use App\Http\Requests\LinkRequest;
```

We haalden met de functie `Request` de request values op, nu gebruiken we de `LinkRequest` functie die ervoor zorgt dat we validatie kunnen toepassen.

Test de validatie.

Fout melding

In onze form kunnen we de user nog feedback geven als de validatie een fout geeft. Dat kunnen we doen door de volgende code aan ons vorm toe te voegen.

```
@if ($errors->any())
    <div class="alert alert-danger">
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
@endif
```

(zie ook: <https://laravel.com/docs/5.8/validation>)

Je kunt de error messages aanpassen door in de LinkRequest file een message class te maken, met (bijvoorbeeld) de volgende code.

```
public function messages()
{
    return [
        'title.required' => 'A title is required',
        'url.required' => 'A URL is required',
    ];
}
```

Veilig Programmeren

Je hebt nu een sterke back-end validatie toegepast. In het kader van veilig programmeren is dit de manier om je input te valideren omdat je de validatie hierbij onafhankelijk maakt van de de front end waar je als programmeur weinig controle over hebt.

Samengevat

Voor back-end validation hebben we de volgende stappen uitgevoerd:

1. `php artisan make:request <object>Request`
2. in `app/Http/Request/<object>Request.php` kun je nu je validatie plaatsen
3. Voeg `use App\Http\Requests\<Object>Request` toe in je `<Object>Controller`
4. Vervang de aanroep van `Request` aanroep in `store()` en `Update()` in `<Object>Request`.

-

Revision #5

Created 2 November 2019 19:07:46 by Admin

Updated 8 December 2019 15:36:09 by Admin