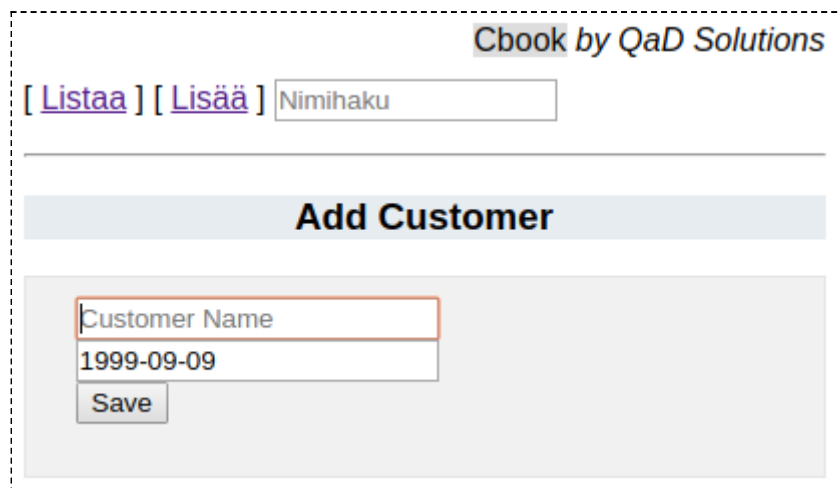


Syötteiden validointi

Syötteiden validointi eli *oikeellisuuden tarkistaminen* on keskeisen tärkeää useissa sovelluksissa. Validoinnilla varmistetaan määriteltyjen *syöttörajoitteiden* toteutuminen. Esim. henkilötunnukseksi ei voida sallia tallennettavaksi kuin täsmälleen oikeanpituinen ja -muotoinen merkkijono.

Tässä esitetään Laravelin suoraviivaisin tapa syötteiden tarkistamiseen suoraan kontrollerissa. Astetta elegantimpaa tapaa voi tutkia vaikkapa [täällä](#).

Tutkitaan syötteiden validointia aiemmin esitetyn asiakastietojen lisäystoiminnon osalta. Lisäyslomakkeessa on syöttökentät asiakkaan nimen ja syntymäpäivän tallentamista varten



The screenshot shows a web interface for adding a customer. At the top, there is a header 'Cbook by QaD Solutions'. Below it, there are two links: '[Listaa]' and '[Lisää]', followed by a search input field labeled 'Nimihaku'. The main section is titled 'Add Customer'. It contains two input fields: the first is labeled 'Customer Name' and the second contains the date '1999-09-09'. Below these fields is a 'Save' button.

Validointi selaimessa ja palvelimella

Syötteiden validointi voidaan tehdä kahdella tasolla

- selaimessa ennen palvelimelle lähettämistä ja
- lopullisena pakollisena tarkistuksena palvelimella

Selaintarkistukset

- antavat käyttäjälle nopeaa palautetta syöttötapauhtuman onnistumisesta, joten ne parantavat käyttäjäkokemusta
- on helppo ohittaa, joten selainpään tarkistukset tulee nähdä vain **esitarkistuksena** lähetettävälle datalle

syötteet on **aina** tarkistettava vähintään palvelimen (tässä
=~Laravel) päässä!

Esitarkistus: validointi selaimessa

HTML5 mahdollistaa useita erilaisia esitarkistusmahdollisuuksia. Esim. `INPUT`-elementin `required`-määritteellä voidaan estää täysin tyhjä syöte seuraavasti

```
1 | <input type="text" required name="name">
```

ja käyttäjän yrittäessä lähettää täyttämätöntä lomaketta nähdään seuraava ilmoitus



HTML5 tarjoaa lisäksi mm. seuraavia syöttörajoitteita:

- `disabled`, `max`, `maxlength`, `min`, `pattern`, `readonly`, `required`, `size`, `step` ja `value`

Esim. **pattern**-rajoitteen säännöllisillä lausekkeilla (regex = regular expressions) voidaan tehdä tarkistuksia lähes kaikkeen kuviteltavaan

Esimerkki: Kevyt luottokortin numeron muototarkistus

```
1 | <label for='lk'>Luottokortin numero:</label><br>
2 | <input type='text' id='lk' name='lk' pattern='[0-9 ]+'
3 |       title='Vain numerot ja välilyönti hyväksytään'>
```

Validointi Laravelin kontrollerissa

- Tehdään **ennen** lomaketietojen tallentamista `validate()`-metodilla. Jos tarkistukset eivät mene läpi, niin palataan takaisin lisäyslomakkeeseen. Kokeile esim. seuraavaa

```
1  ...
2  public function store() {
3
4      request()->validate([
5          'name' => 'required',
6          'birth_date' => 'required|min:10',
7      ]);
8  ...
```

Edellä olevassa on ongelmana se, että käyttäjä ei saa syöttövirheestään minkäänlaista informaatiota.

Laravel tarjoamaan muuhun mittavaan kokoelmaan erilaisia validointisääntöjä voit tutustua seuraavien linkkien avulla

- <https://laravel.com/docs/6.x/validation>
- <https://laravel.com/docs/6.x/validation#available-validation-rules>

Syöttövirheen näyttäminen

Syöttövirhe näytetään `create.blade.php`-näkymässä esim. syöttölomakkeen alapuolella. Kokeile ja selvitä itsellesi ohjelman toiminta

```
...
    </form>
</div> <!-- class="boxi" -->

    @if ($errors->any())

        <div style='background-color: #faa;'>
        <ul>
            @foreach ($errors->all() as $error)
                <li> {{ $error }}</li>
            @endforeach
        </ul>
        </div>
    @endif
```

Edellä olevassa on ongelmana se, että käyttäjä kyllä saa syöttövirheestään informaatiota, mutta se ei suoraan näy siinä kohdassa, missä ongelma esiintyy!

Syöttövirheen näyttäminen kentän yhteydessä

Kokeile ja selvitä itsellesi ohjelman toiminta

```
1 | <input type="text" required name="name" {!! $errors->has('name' |
```

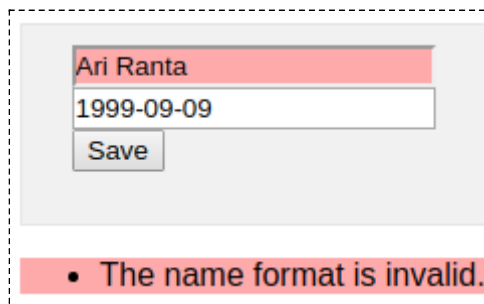
Syöttövirheen validointi säännöllisellä lausekkeella

Laravelin `regex`-tarkistus säännöllisillä lausekkeilla (regular expressions, RE) tarjoaa mahdollisuuden kaikkiin niihin tarkistuksiin, mihin ennalta määrättyt säännöt eivät pysty. RE-osaajalle se voi olla myös nopein tapa päästä halutuun tulokseen. Laravelin `regex`-tarkistus käyttää sisäisesti PHP:n `preg_match()` -funktia eli PERLin RE-syntaksia.

Esim. seuraavalla säännöllä hyväksytään vain merkkejä a-zA-Z0-9äöåÄÖÅ vähintään 3kpl ja enintään 24kpl

```
1 | request()->validate([
2 |     'name' => ['required', 'regex:/^[a-zA-Z0-9äöåÄÖÅ]{3,24}$/'],
3 |     'birth_date' => 'required|min:10',
4 | ]
5 | );
```

Esim. virheellinen välilyönti syötteessä näkyy seuraavasti



Edellä olevassa on pienenä ongelmana vielä se, että virheilmoitus ei ole kovin kuvaava.

Ks. myös

- <https://laravel.com/docs/6.x/validation#rule-regex>

Virheilmoituksen räätälöinti

Kokeile

```
1 request()->validate([  
2   'name' => ['required', 'regex:/^[a-zA-Z0-9äöåÄÖÅ]{3,24}$/',  
3   'birth_date' => 'required|min:10',  
4   ],  
5   ['name.regex' => 'Pitää olla: ^[a-zA-Z0-9äöåÄÖÅ]{3,24}$' ]  
6   );
```

No. Ei se virheilmoitus vieläkään informatiivinen ole, mutta noin sitä voi itse säätää.

Muita tapoja säätää omia virheilmoituksia: [Custom Error Messages](#).

© #AriRantala