

# Laravel - Tietokannan sisältöä web-sivulle

Tässä luvussa esitetään kuinka edellisessä luvussa luodun `laraveldb`-tietokannan `customers`-taulun sisältö näytetään web-sivulla reittitiedoston, kontrollerin, mallin ja näkymän avulla.

## 28.01 Luodaan reitti ja kontrolleri

Luodaan `routes/web.php`-tiedostoon reitti tietueiden listaamiseksi

```
Route::get('/customers', 'CustomerController@index');
```

ja lisätään reittiin määritelty kontrolleri `artisan`-skriptillä

```
php artisan make:controller CustomerController
```

tämän tuloksena syntyy

tiedosto `app/Http/Controllers/CustomerController.php`, jonka sisältö on luonnin jälkeen

```
1  <?php
2  namespace App\Http\Controllers;
3  use Illuminate\Http\Request;
4
5  class CustomerController extends Controller
6  {
7      //
8  }
```

Lisätään luotuun kontrolleriin reitissä määritelty `index()`-metodi seuraavasti

```
1  class CustomerController extends Controller
2  {
```

```

3     public function index() {
4         return 'Coming soon: list all customers';
5     }
6 }

```

ja testataan selaimessa osoitteessa

<http://192.168.1.126/~testi/projekti01/public/customers> reitin ja kontrollerin toimivuus. Ohjelman tulisi tulostaa selaimen teksti

*Coming soon: list all customers*

## 28.02 customers-taulun sisältö näkymään

Käytetään edellisessä luvussa esitettyä Eloquentin `all()` -metodia tietuiden hakemiseen ja muotoillaan ne näkymän avulla selaimen näytettäväksi HTML-muodossa.

Testataan aluksi, että tietokantayhteys ja Customer-malli toimii datan hakemissa. Kirjoitetaan `index()` -metodi seuraavassa muodossa ja lisätään rivin 8 use-määrittäminen

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  // Tuodaan Customer-luokka nimiavaruuksiin käyttöön
8  use App\Customer;
9
10 class CustomerController extends Controller
11 {
12     public function index() {
13         $customers = Customer::all();
14         return $customers;
15     }
16 }

```

ja katsotaan selaimessa <http://192.168.1.126/~testi/projekti01/public/customers> lopputulosta. Laravel palauttaa tuloksen JSON-formaatissa ja tuloksen tulisi näyttää sisällöltään oheisen kuvan mukaiselta. Jos selaimesi ei muotoile JSON-

dataa yhtä siistiksi kuin kuvassa, niin asenna selaimeesi halutessasi jokin JSONView- tai vastaava plugin JSON-datan muotoilemiseksi luettavammaksi

```
[
  {
    id: 1,
    name: "Guru Ken",
    birth_date: "2011-11-11",
    created_at: "2019-10-05 08:15:00",
    updated_at: "2019-10-05 08:15:00",
  },
  {
    id: 2,
    name: "Vainio Vilja",
    birth_date: "2012-12-12",
    created_at: "2019-10-05 13:04:40",
    updated_at: "2019-10-05 13:26:26",
  },
  {
    id: 4,
    name: "Alainen Kim",
    birth_date: "2010-10-10",
    created_at: "2019-10-06 05:08:28",
    updated_at: "2019-10-06 05:08:28",
  }
]
```

Muutetaan index-metodia siten, että edellä onnistuneesti haettu data annetaan näkymälle `customers/index` muotoiltavaksi.

```
1 public function index() {
2     $customers = Customer::all();
3     return view('customers/index')->with('customers', $customers);
4 }
```

Luodaan siis `resources/views/customers`-kansio ja sinne tiedosto `index.blade.php`. `with()`-metodin toisena parametrina annettu `$customers`-muuttuja (taulukko) välitetään näkymälle ensimmäisenä parametrina annetun nimiselle `customers` muuttujalle.

`resources/views/customers/index.blade.php`

```
1 <!DOCTYPE html>
2 <html>
3
4     <head>
5         <title>Customers</title>
6         <style>
7             table, th, td {
8                 border:1px solid black;border-collapse:collapse;
```

```

 9      }
10    </style>
11  </head>
12
13  <body>
14    <h1>Customers</h1>
15    <table >
16      <tr><th>Name</th><th>Birth date</th></tr>
17      @foreach ($customers as $customer)
18        <tr>
19          <td>{{ $customer->name }}</td>
20          <td>{{ $customer->birth_date }}</td>
21        </tr>
22      @endforeach
23    </table>
24  </body>
25
26 </html>

```

Huomaa riveillä 17-22 Laravelin **Blade template**-järjestelmän `foreach`-rakenne. Lisäksi tulee kiinnittää huomiota siihen, että tässä `foreach`-rakenne **ei** toteuta sovelluslogiikkaa vaan sen tehtävänä on vain muotoilla taulukossa oleva data näkyville käyttäliittymään

Selaimessa <http://192.168.1.126/~testi/projekti01/public/customers> lopputulos näyttää suunnilleen seuraavalta

Customers	
Name	Birth date
Guru Ken	2011-11-11
Vainio Vilja	2012-12-12
Alainen Kim	2010-10-10

## 28.03 Yhden asiakkaan näyttäminen

### A. Luodaan ensin reitti

```
Route::get('/customers/{id}', 'CustomerController@show');
```

B. ja `show($id)` -metodi, jonka parametrille `$id` välitetään URLissa reitin `{id}`-merkintää vastaavaa osa. Jos

- URL: `/customers/1` -> `$id: 1`
- URL: `/customers/2` -> `$id: 2`
- URL: `/customers/peto` -> `$id: peto`

```
1 public function show($id) {
2     $customer = Customer::find($id);
3     // Kokeile myös usein parempaa
4     //$customer = Customer::findOrFail($id);
5     return $customer;
6 }
```

Esim. URLilla <http://192.168.1.126/~testi/projekti01/public/customers/1> tulisi selaimeen palautua JSON-formaatissa sen asiakkaan tiedot, jonka id-kenttän arvo on 1.

```
1 {
2     "id": 1,
3     "name": "Guru Ken",
4     "birth_date": "2011-11-11",
5     "created_at": "2019-10-05 08:15:00",
6     "updated_at": "2019-10-05 08:15:00"
7 }
```

C. Välitetään yhden asiakkaan tiedot luotavalle näkymälle

`resources/views/customers/show.blade.php` näytettäväksi. Muutetaan `show()` -metodia hieman

```
1 public function show($id) {
2     $customer = Customer::find($id);
3     return view('customers/show')->with('customer', $customer);
4 }
```

ja kirjoitetaan näkymä

`resources/views/customers/show.blade.php`

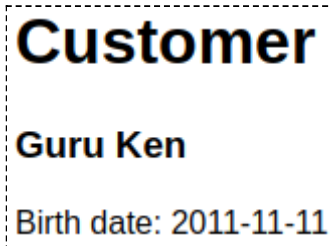
```
1 <!DOCTYPE html>
2 <html>
3
```

```

4      <head>
5          <title>Customers</title>
6          <style>
7              table, th, td {
8                  border:1px solid black;border-collapse:collapse;
9              }
10         </style>
11     </head>
12
13     <body>
14         <h1>Customer</h1>
15         <h3>{{ $customer->name }}</h3>
16         <p>Birth date: {{ $customer->birth_date }}</p>
17     </tr>
18 </body>
19
20 </html>

```

URLilla <http://192.168.1.126/~testi/projekti01/public/customers/1> tulisi selaimeen tulostua näkymä



**Customer**

**Guru Ken**

Birth date: 2011-11-11

## 28.04 Yhden asiakkaan valitseminen listasta

Edellä esitettiin kuinka yksittäisen asiakkaan tiedot voidaan näyttää käyttöliittymässä kirjoittamalla osoiteriville URL ja "*arvaten*" samalla asiakkaan id-numeron. Tehdään haluttujen asiakkaiden valitseminen näytettäväksi helpommaksi tekemällä asiakaslistaan asiakkaista linkki yksittäisen asiakkaan näyttämiseksi.

Tämä tapahtuu muokkaamalla pelkästään listausnäkyä `foreach`-rakenteen osalta oheisesti. Asiakkaan nimestä tehdään hyperlinkki reittiin, joka näyttää juuri tällä rivillä näkyvillä olevan asiakkaan tiedot. Käytetystä web-palvelin-konfiguraatiosta johtuen hyperlinkki alkaa tässä ilman kauttaviivaa, jolloin viittaus on suhteessa käyttämämme public-kansion juureen <http://192.168.1.126/~testi/projekti01/public/resources/views/customers/index.blade.php>

```
1  @foreach ($customers as $customer)
2      <tr>
3          <td>
4              <a href='customers/{{ $customer->id }}'>
5                  {{ $customer->name }}
6              </a>
7          </td>
8          <td>{{ $customer->birth_date }}</td>
9      </tr>
10 @endforeach
```

Laravelin Blade template enginen tarjoamaan toiminnallisuuteen voi tutustua laajemmin mm. osoitteessa  
<https://laravel.com/docs/6.x/blade>