

Telefonok - tutorial

Laravel: view

Rostagni Csaba

Telefonok

Előkészületek

1. Hozzon létre egy új projektet a **README.md** leírása alapján, majd indítsa el a szerveret üzemeltető környezetet.
2. Másolja be a **PhoneController** fájlt az **app\Http\Controllers** mappába.
3. A képeket, fájlokat nem minden esetben tároljuk együtt a kóddal. Ezek gyakran a felhasználó által feltöltött tartalom.
 - Ezeket a fájlokat a **storage/app/images** mappában tároljuk ehhez a feladathoz
 - Módosítsa a **config/filesystems.php** fájlt úgy, hogy a publikus mappa legyen **img**, míg a storage mappa legyen **app/images**
 - A **php artisan storage:link** parancs létrehoz egy szimbolikus linket a projektmappában lévő **public** mappába
 - <https://laravel.com/docs/11.x/filesystem#the-public-disk>

Layout

A HTML oldal alap szerkezetét csak 1 helyen szeretnénk karbantartani. Erre szolgál a layout.

A nézetfájlok csak az oldal tartalmát tartalmazzák, vagy annak is csak egy meghatározott részét. Így ezek a fájlok újra felhasználhatóak, amennyiben egy másik oldalon is ugyanazt az elrendezést szeretnénk megjeleníteni.

4. Előfordulhat, hogy több alap elrendezés is létezik, így ezeknek hozzon létre egy **layouts** nevű mappát a **resources/views/** mappában.
5. Az imént létrehozott mappában hozzon létre egy **main.blade.php** fájlt, ami tartalmazza egy HTML oldal alapszerkezetét, illetve itt legyen belinkelve a Bootstrap is, külső hivatkozással.
6. A layout fájlban lesznek bizonyos blokkok, amik helyére valamilyen tartalom kerül. Ezt a blade-nek a **@yield** függvényével jelöljük. Ilyen az oldal címe és a tartalom is.

```
<title>@yield("title")</title>
```

```
<body>@yield("content")</body>
```

7. Készítsen pillanatképet **layout created** megjegyzéssel

Telefonok listázása

8. Készítse el a megfelelő route-ot az alábbi leírás alapján.

| Method | URI | Name | Controller | Action |
|--------|-----|------|-----------------|--------|
| GET | / | home | PhoneController | index |

- A feladat megoldásához a `routes/web.php`-t kell kiegészíteni.
- Mivel a metódus a táblázat szerint **GET**, így a **Route** osztály statikus `get()` metódusát kell meghívni.

```
Route::get();
```

- Az első paraméter az útvonal lesz, ami jelen esetben a `"/`.

```
Route::get('/');
```

- A második paramétere a controllert és az actiont határozza meg. Ezt egy két elemű tömbként lehet megadni.

- A `class` kulcsszó megadja, hogy mi az **osztály teljes neve**, beleértve a névteret is.
- <https://www.php.net/manual/en/language.oop5.basic.php#language.oop5.basic.class.class>

```
use App\Http\Controllers\PhoneController;
...
PhoneController::class
```

- Meg lehetne adni egyszerű szöveggként is az osztály teljes nevét, de ezzel a módszerrel elgépelés esetén hibát kapunk, ami könnyebben javítható.
- Az action a táblázat szerint az `index` lesz.

```
Route::get("/", [PhoneController::class, "index"]);
```

9. A `PhoneController`-hez tartozó nézetfájlokat a `resources/views` mappán belül, a controller után `phone-`nak elnevezett mappában célszerű tárolni, ehhez hozza létre a szükséges mappaszerkezetet.
10. Mivel az `index` action tartalmi részének a megírása a cél, így hozzon létre egy nézetfájlt `index.blade.php` néven az imént létrehozott mappába.
11. A metódus 2 változót adjon át a nézetnek:
 - `$title`, az oldal címe
 - `$phones`, a telefonokat tartalmazó tömb. Ezen belül minden telefon egy-egy asszociatív tömb lesz az alábbi adatokkal:
 - `id`, a telefon azonosítója
 - `manufacturer`, a telefon gyártója (pl.: Nokia)
 - `model`, a telefon modellnek a megnevezése (pl.: 3310)
 - `storage`, a telefon tárhelyének a mérete gigabyte mértékegységben. Amennyiben a telefon nem rendelkezik tárhellyel, úgy az értéke `null`.
 - `color`, a telefon színe
 - `price`, a telefon ára forintban
 - `description`, a telefon leírása
12. Ahhoz, hogy a korábban létrehozott layout fájlban jelenjen meg, úgy meg kell határozni az "ősét". Állítsa be a `layouts.main` fájl, mint a nézet "szülője".

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

13. A layoutban két szekció jelenik meg. Az egyik az oldal címe, a másik pedig a tartalom. Állítsa be, hogy az oldalon megjelenő cím legyen a controller által átadott `$title` változó értéke.

```
@section("title", $title)
```

14. Helyezze el a listázó oldal tartalmát egy `content` nevű szakaszban.

```
@section
...
@endsection
```

15. A teljes tartalmat egy fix szélességű Bootstrap tárolóelemben helyezze el.

```
<div class="container"></div>
```

16. A controllertől kapott címet jelenítse meg címsor 1-es kiemeléssel.

```
<h1>{{ $title }}</h1>
```

17. Az egyes telefonokat BS rácsrendszerben jelenítse meg.

- Alapértelmezetten minden mobil egymás alatt jelenjen meg (mobile first)
- Közepes nézettől kezdve 2 telefon jelenjen meg egymás mellett.
- Nagy nézettől kezdve 4 telefon jelenjen meg egymás mellett
- Ahhoz, hogy az elemek egyforma magasak legyenek használja ki, hogy a BS rácsrendszer sora egy flexbox.
- Minden oszlop kapjon közepes belső margót az y tengelyen, hogy a tartalom ne tapadjon egymásra.

```
<div class="row align-items-stretch">
  <div class="col-md-6 col-lg-3 py-3 ">
    </div>
</div>
```

18. A controllertől kapott `$phones` tömböt járja be megfelelő ciklussal. Ügyeljen rá, hogy a megfelelő HTML elem kerüljön ismétlésre és a hozzá tartozó záró tag-ek is helyesen jelenjenek meg.

```
<div class="row align-items-stretch">
  @foreach($phones as $phone)
    <div class="col-md-6 col-lg-3 py-3 ">
    </div>
  @endforeach
</div>
```

19. Minden telefon adatát 1-1 card komponensben jelenítse meg.

- Állítsa be, hogy a kártya magassága legyen 100%

```
<div class="card h-100">
...
</div>
```

20. Jelenítse meg a telefonhoz tartozó képet.

- A nézetfájlokban az `asset()` függvény generálja le a megfelelő útvonalat. A paramétereként a `public` mappától kezdve kell megadni az útvonalat.
- A képek a `public/img` mappában találhatóak, amennyiben a beállítások helyesek
- A telefon azonosítója a fájlnev, amit a `.jpg` fájlkiterjesztés követ.
- A képet lássa el a `card-img-top` osztállyal.
- Amennyiben a kép nem jelenne meg, úgy a telefon gyártója és a modell neve jelenjen meg egy szóközzel elválasztva.

```

```

21. Alakítsa ki a kártya tartalmi részét az alábbiak szerint:

- A kártya címe legyen a gyártó és a modell egy szóközzel elválasztva.
- A kártya szövege legyen a telefon leírása.

```
<div class="card-body">
  <h5 class="card-title">{{ $phone['manufacturer']}} {{ $phone['model'] }}</h5>
  <p class="card-text">{{ $phone['description'] }}</p>
</div>
```

22. A kártya lábléce legyen egy két oszlopos rácsrendszer, ami 8/12 és 4/12 részben osztja fel a tartalmat.

- A bal oldalon egy elsődleges BS gomb szerepeljen, ami az 1 telefon megjelenítésére szolgáló oldalra mutat.
- A gomb szövege legyen a gyártó és a modell egy szóközzel elválasztva
- A gomb teljes szélességében tölts ki a rendelkezésére álló területet.
- A jobb oldalon az ár jelenjen meg a minta szerint formázva, a pénznem feltüntetésével együtt.

```
<div class="card-footer">
  <div class="row">
    <div class="col-md-8 d-grid">
      <a
        href="{{route('phones.show', ['id' => $phone['id']])}}"
        class="btn btn-primary">
          {{ $phone['manufacturer']}} {{ $phone['model'] }}
      </a>
    </div>
    <div class="col-md-4">
      <span>{{number_format($phone['price'], 0, ",", " ")}} Ft</span>
    </div>
  </div>
</div>
```

23. Készítsen pillanatképet “phone list created” megjegyzéssel

Egy telefon adatainak megjelenítése

24. Készítse el a megfelelő route-ot az alábbi leírás alapján.

| Method | URI | Name | Controller | Action |
|--------|--------------|-------------|-----------------|--------|
| GET | /phones/{id} | phones.show | PhoneController | show |

- A feladat megoldásához a `routes/web.php`-t kell kiegészíteni.
- Mivel a metódus a táblázat szerint **GET**, így a **Route** osztály statikus `get()` metódusát kell meghívni.

```
Route::get();
```

- Az első paraméter az útvonal lesz, ami jelen esetben a `"/phones/{id}"`.

```
Route::get('/phones/{id}');
```

- A második paramétere a controllert és az actiont határozza meg. Ezt egy két elemű tömbként lehet megadni.
 - A `class` kulcsszó megadja, hogy mi az **osztály teljes neve**, beleértve a névteret is.
 - <https://www.php.net/manual/en/language.oop5.basic.php#language.oop5.basic.class.class>

```
use App\Http\Controllers\PhoneController;
...
PhoneController::class
```

- Meg lehetne adni egyszerű szöveggént is az osztály teljes nevét, de ezzel a módszerrel elgépelés esetén hibát kapunk, ami könnyebben javítható.
- Az action a táblázat szerint a **show** lesz.

```
Route::get("/", [PhoneController::class, "show"]);
```

25. A `PhoneController`-hez tartozó nézetfájlokat a `resources/views` mappán belül, a controller után `phone`-nak elnevezett mappában célszerű tárolni ezt a nézetet is.

26. Mivel a **show** action tartalmi részének a megírása a cél, így hozzon létre egy nézetfájlt `show.blade.php` néven a `phone` mappába.

27. A controller 2 változót ad át a nézetnek:

- `$title`, az oldal címe
- `$phone`, egy telefon adatai, egy asszociatív tömb lesz az alábbi adatokkal:
 - `id`, a telefon azonosítója
 - `manufacturer`, a telefon gyártója (pl.: Nokia)
 - `model`, a telefon modellnek a megnevezése (pl.: 3310)
 - `storage`, a telefon tárhelyének a mérete gigabyte mértékegységben. Amennyiben a telefon nem rendelkezik tárhellyel, úgy az értéke `null`.
 - `color`, a telefon színe
 - `price`, a telefon ára forintban
 - `description`, a telefon leírása

28. Ahhoz, hogy a korábban létrehozott layout fájlban jelenjen meg, úgy meg kell határozni az "ősét". Állítsa be a `layouts.main` fájlt, mint a nézet "szülője".

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

29. A layoutban két szekció jelenik meg. Az egyik az oldal címe, a másik pedig a tartalom. Állítsa be, hogy az oldalon megjelenő cím legyen a controller által átadott `$title` változó értéke.

```
@section("title", $title)
```

30. Helyezze el a listázó oldal tartalmát egy `content` nevű szakaszban.

```
@section
...
@endsection
```

31. A teljes tartalmat egy fix szélességű BS tárolóelemben helyezze el.

```
<div class="container"></div>
```

32. A controllertől kapott címet jelenítse meg címsor 1-es kiemeléssel.

```
<h1>{{$title}}</h1>
```

33. A telefon adatait egy két oszlopos rácsrendszerben jelenítse meg. A két oszlop 4/12 és 8/12 részre osztva jelenjen meg.

```
<div class="row">
  <div class="col-md-4">
  </div>
  <div class="col-md-8">
  </div>
</div>
```

34. Az első oszlopban a telefon képe jelenjen meg!

- A nézetfájlokban az `asset()` függvény generálja meg a megfelelő útvonalat. A paramétereként a `public` mappától kezdve kell megadni az útvonalat.
- A képek a `public/img` mappában találhatóak, amennyiben a beállítások helyesek
- A telefon azonosítója a fájlnev, amit a `.jpg` fájlkiterjesztés követ.
- A képet lássa el a `img-fluid` osztállyal.
- Amennyiben a kép nem jelenne meg, úgy a telefon gyártója és a modell neve jelenjen meg egy szóközzel elválasztva.

```

```

35. A jobb oldali oszlopban felül jelenjen meg egy felsorolás a telefon adataival.

```
<ul>
  <li>Gyártó: {{ $phone["manufacturer"] }}</li>
  <li>Típus: {{ $phone["model"] }}</li>
  <li>Szín: {{ $phone["color"] }}</li>
</ul>
```

36. A tárhely egyes modelleknél nincs kitöltve. Ezeknél a *“nincs”* jelenjen meg, míg ahol ki van töltve ott jelenjen meg a mértékegység is, ami a *“GB”*.

```
<ul>
  <li>
    Tárhely:
    @if(is_null($phone["storage"]))
      nincs
    @else
      {{$phone["storage"]}} GB
    @endif
  </li>
</ul>
```

37. Az ár itt is formázottan jelenjen meg! Az árak egész számként jelenjenek meg, használjon tizedes vesszőt, az ezres elválasztás karaktere pedig legyen szóköz.

```
<ul>
  <li>Ár: {{number_format($phone["price"], 0, ",", " ")}} Ft</li>
</ul>
```

38. A felsorolás alatt jelenjen meg a leírás egy bekezdésben.

```
<p>{{$phone["description"]}}</p>
```

39. Végül helyezzen el egy linket a főoldalra.

```
<a href="{{route("home")}}">Vissza a főoldalra</a>
```

Menü

40. A menü egy nagyobb egység, így külön fájlban lesz a helye. Hozzon létre egy `menu.blade.php` fájlt a `resources/views/layouts` mappába.
41. A menü csak akkor jelenik meg az oldalon, ha a meg van hívva a layout fájlban. Mivel ez egy külső fájl, így a `@yield` helyett a `@include`-ot helyezze el közvetlen a nyitó `<body>` után.

```
@include('layouts.menu')
```

42. Hozzon létre egy tömböt *“iPhone”*, *“Pixel”*, *“3310”*, *“Moto G54”* értékekkel 1-től indexelve.

```
@php
$phones = [
  1 => "iPhone",
  2 => "Pixel",
  3 => "3310",
  4 => "Moto G54"
];
@endphp
```

43. Másolja be a Navbar BS komponens kódját, majd törölje ki a fölösleges elemeket.
44. A navigációs sáv legyen sötét, és állítsa be a háttérét `#233049` színkódúra.

```
<nav class="navbar" data-bs-theme = "dark" style="background-color: #233049;">
```

45. Állítsa be, hogy a közepes mérettől jelenjen meg minden menüpont.

```
<nav class="navbar navbar-expand-md" data-bs-theme = "dark" style="background-color: #233049;">
```

46. Állítsa be, hogy a brand felirat legyen “Telefonok”, és a listázó oldalra mutasson.

```
<a class="navbar-brand" href="{{route('home')}}">Telefonok</a>
```

47. Kétféle link létezik. Az egyik az aktív oldalt mutatja, míg a másik az éppen nem aktív oldalakra hivatkozik.

```
<a class="nav-link active" aria-current="page" href="#">Home</a>
```

```
<a class="nav-link" href="#">Link</a>
```

48. Hozzon létre egy linket “Főoldal” névvel, ami a telefonokat listázó oldalra mutat. Amennyiben ez lenne az aktív link, úgy jelölje ezt megfelelően.

```
<li class="nav-item">
    @php
        $url = route('home');
    @endphp

    @if(request()->routeIs('home'))
        <a class="nav-link active" aria-current="page" href="{{ $url }}">Főoldal</a>
    @else
        <a class="nav-link" href="{{ $url }}">Főoldal</a>
    @endif
</li>
```

49. Jelenítse meg a négy telefont a menüben. Mindegyik a saját részletes adatait megjelenítő oldalra mutasson.

```
@foreach($phones as $id => $name)
<li class="nav-item">
    @php
        $url = route('phones.show', ['id' => $id]);
    @endphp
    @if(request()->routeIs('phones.show') && request()->route('id') == $id)
        <a class="nav-link active" aria-current="page" href="{{ $url }}">{{ $name }}</a>
    @else
        <a class="nav-link" href="{{ $url }}">{{ $name }}</a>
    @endif
</li>
@endforeach
```