Tartalomjegyzék

[0., Előfeltétel Composer telepítése 2](#_Toc106127771)

[1., LARAVEL keretrendszer telepítése 2](#_Toc106127772)

[2., ADATBÁZIS létrehozása 2](#_Toc106127773)

[2.1., Az .env fájlban adatbáziselérés beállítása 2](#_Toc106127774)

[3., Migráció elkészítése (adatbázis tábláinak létrehozása) 2](#_Toc106127775)

[4., Modell létrehozása 3](#_Toc106127776)

[5., Controller (vezérlő) 3](#_Toc106127777)

[6., Routing (útvonalválasztás) 4](#_Toc106127778)

[7., View (nézet) létrehozása 5](#_Toc106127779)

# 0., Előfeltétel Composer telepítése

<https://getcomposer.org/download/> oldalról letölteni és telepíteni

# 1., LARAVEL keretrendszer telepítése

Laragon – jobb klikk – „Gyors létrehozása weboldalnak” – Laravel  
VAGY  
composer create-project --prefer-dist laravel/laravel bukizoli-20220307

// A Laravel projekt létrehozásakor szokás használni a **--prefer-dist** kapcsolót. A kapcsoló megmondja, hogy a Laravel és függőségeit a terjesztési archívumból szeretnénk letölteni, a verzióval követett, korábbi verziók nélkül. Ennek használata gyorsabb letöltést eredményez. A **--prefer-dist** az alapértelmezés. Ezzel szemben a **--prefer-source** a forráskódot, a korábbi verziókkal egyben tölti le, mint a git clone, parancs.

VAGY  
Laravel installerrel:  
 composer global require laravel/installer  
 laravel new bukizoli-20220307

Laravel alkalmazás kiszolgálójának futtatása:

cd bukizoli-20220307  
 code . // Visual Studio Code futtatása alkalmazás megnyitásával  
 php artisan serve // Ha szükséges alap 8000-es port módosítása,  
 akkor végére pl.: **php artisan serve –port=8080**

// Ha távolról is szeretnénk elérni, használjuk a --host kapcsolót:

**php artisan serve --host=0.0.0.0 --port=8080**

127.0.0.1:8000 // Böngészőben erről az útról lehet az oldalt megtekinteni

# 2., ADATBÁZIS létrehozása

Pl. phpMyadmin-ban vegyük fel az adatbázist és egy hozzátartozó felhasználót. A táblákat nem szükséges létrehozni, azokat a Laravel migration nevű szolgáltatásával hozzuk létre.

## 2.1., Az .env fájlban adatbáziselérés beállítása

DB\_CONNECTION=mysql  
DB\_HOST=127.0.0.1  
DB\_PORT=3306  
DB\_DATABASE=20220307 // adatbázisunk neve  
DB\_USERNAME=root // adatbázisunk felhasználóneve  
DB\_PASSWORD= // adatbázisunk jelszava

# 3., Migráció elkészítése (adatbázis tábláinak létrehozása)

A migrációk a következő helyen találhatók: **database/migrations**

Megjegyzés: Amelyik táblában idegen kulcs (FK) van, azt később kell létrehozni, amiben nincs idegen kulcs (FK), az az elején kell létrehozni

php artisan make:migration create\_tanulok\_table

Az elkészült fájl neve az aktuális dátummal fog kezdődni, pl:

2022\_06\_14\_174953\_create\_tanulok\_table.php

A fájl tartalmazni fog egy **up()** és egy **down()** függvényt.

Az **up()** függvény átírásával kialakítjuk a táblánkat:

**public** **function** up()

{

Schema::create('tanulok', **function** (Blueprint $table) {

$table->id();

$table->string('nev'); // nev nevű mezőt hoz létre

$table->integer('kor'); // kor nevű mezőt hoz létre

$table->string('nem'); // nem nevű mezőt hoz létre

$table->string('diakigszam'); // diakigszam nevű mezőt hoz létre

$table->timestamps();

});

}

A **Schema::create()** függvény fogja elkészíteni a táblát. Az első paraméter az elkészítendő tábla neve. A következő egy névtelen függvény, amely a Blueprint típusú objektumot vár paraméterként, ami maga a tábla.

A **php artisan migrate** parancs hatására, létrejön az adatbázisban a tábla.

# 4., Modell létrehozása

Az adatbázisok megléte után a kontroller az adatokat egy modelltől fogja megkapni. Egy **Tanulok** nevű modellt kell létrehozni. **A modell neve meg kell egyezzen a tábla nevével, de a modell neve legyen nagybetűs.**

**php artisan make:model Tanulok**

Így létrejön egy **Tanulok.php** fájl.

A modellek a következő helyen találhatók: **app/Http/Models**

**<?php**

**namespace** App\Models;

**use** Illuminate\Database\Eloquent\Factories\HasFactory;

**use** Illuminate\Database\Eloquent\Model;

**class** Tanulok **extends** Model

{

    protected $table = 'tanulok';

    protected $primaryKey = 'id';

    protected $fillable = ['nev', 'kor', 'nem', 'diakigszam'];

}

# 5., Controller (vezérlő)

Alapértelmezés szerint a vezérlők az **app/Http/Controllers** könyvtárban vannak tárolva.

Kontroller létrehozása, pl TanulokController néven:

php artisan make:controller TanuloController --resource

Így a fenti könyvtárban a **TanuloController.php** állomány jön létre. A **TanuloController** osztályon (**class** TanulokController **extends** Controller) belül tudunk függvényt létrehozni, illetve létre is jönnek a --resource miatt, mint:

index(); create(); store(); show(); edit(); update(); destroy():

**public** **function** index() {

$tanulok = Tanulok::all();

return view('tanulok.index')->with('tanulok', $tanulok);

}

**public** **function** create() {

return view('tanulok.create');

}

**public** **function** store(Request $request) {

$input = $request->all();

Tanulok::create($input);

return redirect('tanulok')->with('flash\_message', 'Tanuló hozzáadva!');

}

**public** **function** show($id) {

$tanulok = Tanulok::find($id);

return view('tanulok.show')->with('tanulok', $tanulok);

}

**public** **function** edit($id) {

$tanulok = Tanulok::find($id);

return view('tanulok.edit')->with('tanulok', $tanulok);

}

**public** **function** update(Request $request, $id) {

$tanulok = Tanulok::find($id);

$input = $request->all();

$tanulok->update($input);

return redirect('tanulok')->with('flash\_message', 'Tanuló frissítve!');

}

**public** **function** destroy($id) {

Tanulok::destroy($id);

return redirect('tanulok')-> with('flash\_message', 'Tanuló törölve!');

}

Most be kell jegyeznünk az útválasztóban a kontrollert.

# 6., Routing (útvonalválasztás)

A routok a következő helyen találhatók, mint pl. **routes/web.php** fájl.

Szerkesszük a **routes/web.php** fájlt. Az állomány elején mondjuk meg, hogy szeretnénk használni a TanulokController osztályt:

**use** App\Http\Controllers\TanuloController;

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\TanuloController;

Route::get('/', function () {

return 'Szia Lajos!';

});

Route::resource('/tanulok', TanulokController::class);

Az első paraméter megmondja, ha **/tanulok** hivatkozás érkezik, akkor a második paraméterben megadott **TanulokController** osztályt futtassuk.

# 7., View (nézet) létrehozása

A nézetek a következő helyen találhatók: **resources/views**, elnevezése: **valami.blade.php**

Nézet úgy hozható létre, ha létrehozunk a fenti elérési helyen egy fájlt **.blade.php** kiterjesztéssel.

A **layout.balde.php** fájlban:

Az **index.balde.php** fájlban:

Az **edit.balde.php** fájlban:

A **show.balde.php** fájlban:

A **create.balde.php** fájlban: