**Feladat leírása**:

A userek hibabejelentés vagy kérés rögzítésének céljából ticketeket (jegyeket) nyithatnak saját-, vagy akár egy kolléga részére is.

A ticketek kétféle típusba sorolhatóak: incidens(incident) vagy kérés (request).

Adott jegy állapotát ill. adatait maga a rögzítő felhasználó vagy bármely másik user módosíthatja. Ugyanez a „szabály” érvényes adott ticket törlésére is.

Minden ticket egyedi azonosítóval van ellátva (id), tartozik hozzá egy rövid leírás (title) és egy leírás (description) mező. Továbbá minden ticketen fel van tüntetve

* a ticket típusa (type 🡪 incident/request)
* a rögzítő személy (created\_by)
* a célszemély (created\_for)
* az aktuális kezelő személy (handled\_by)
* az utolsó módosítást végző user (last\_modified\_by)
* a jegyet lezáró felhasználó (closed\_by) neve
* a rögzítés dátuma (created\_on)
* az utolsó módosítás dátuma (last\_modified\_on)
* a lezárás dátuma (closure\_date)

Minden jegynek 3-féle állapota (status) lehet:

* új (new)
* folyamatban (in\_progress)
* lezárt (closed)

**Táblák**:

|  |
| --- |
| Users |
| id (integer) |
| name (varchar) |
| resolver (boolean) |
| e-mail |
| password |

|  |
| --- |
| Ticket |
| id |
| title |
| status |
| type |
| description |
| created\_on |
| last\_modified\_on |
| closure\_date |
| created\_by |
| created\_for |
| handled\_by |
| last\_modified\_by |
| closed\_by |

**Fejlesztési lehetőségek**:

Csak resolver módosíthasson/törölhessen ticketeket!

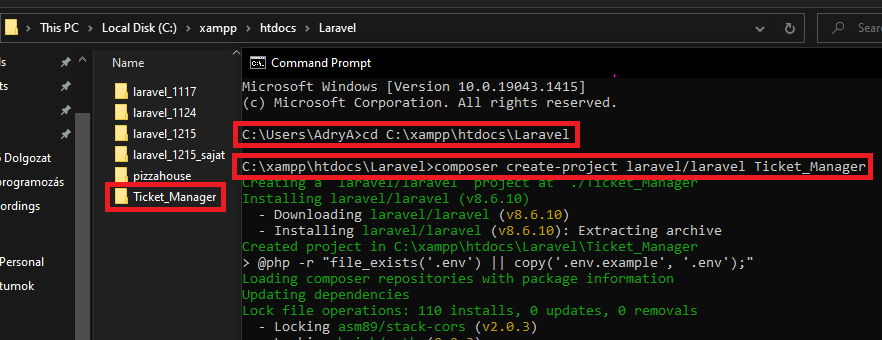
A nyitás lehetősége bármely user számára adott.

**Elkészítés lépései**

Projekt kreálása

C:\Users\AdryA>cd C:\xampp\htdocs\Laravel

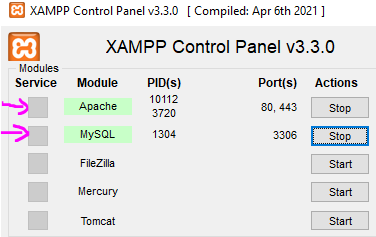
C:\xampp\htdocs\Laravel>composer create-project laravel/laravel *Ticket\_Manager*



Megj.: Nem kell composer install, mivel az már lefut a projekt kreálásakor!

Előkészítés

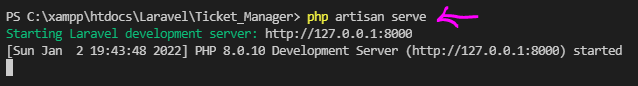
XAMPP elindítása (Apache, MySQL):



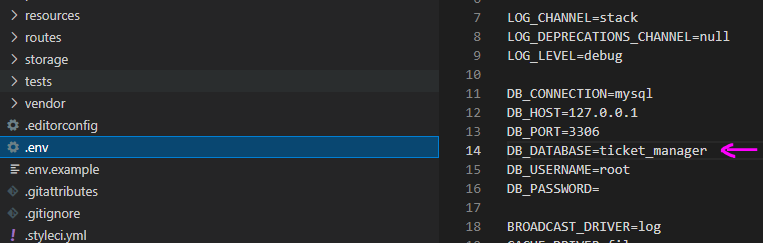
Projekt betöltése VS Code-ba, új terminál nyitása:



php artisan serve



Adatbáziskapcsolat előkészítése:



Az adatbázist a megjelölt néven később hozzuk létre.

Modellek és Controllerek létrehozása

**Modellek helye**: app/Models

**Controllerek helye**: app:/Http/Controllers

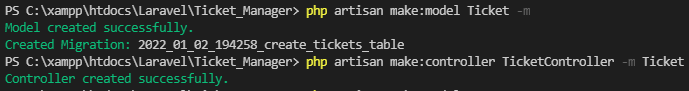
1. Ticket modell

php artisan make:model Ticket -m

// -m jelenti, hogy a modellhez migrációt is létre kívánunk hozni

php artisan make:controller TicketController -m Ticket

// -m Project a modellhez kapcsolódást jelenti

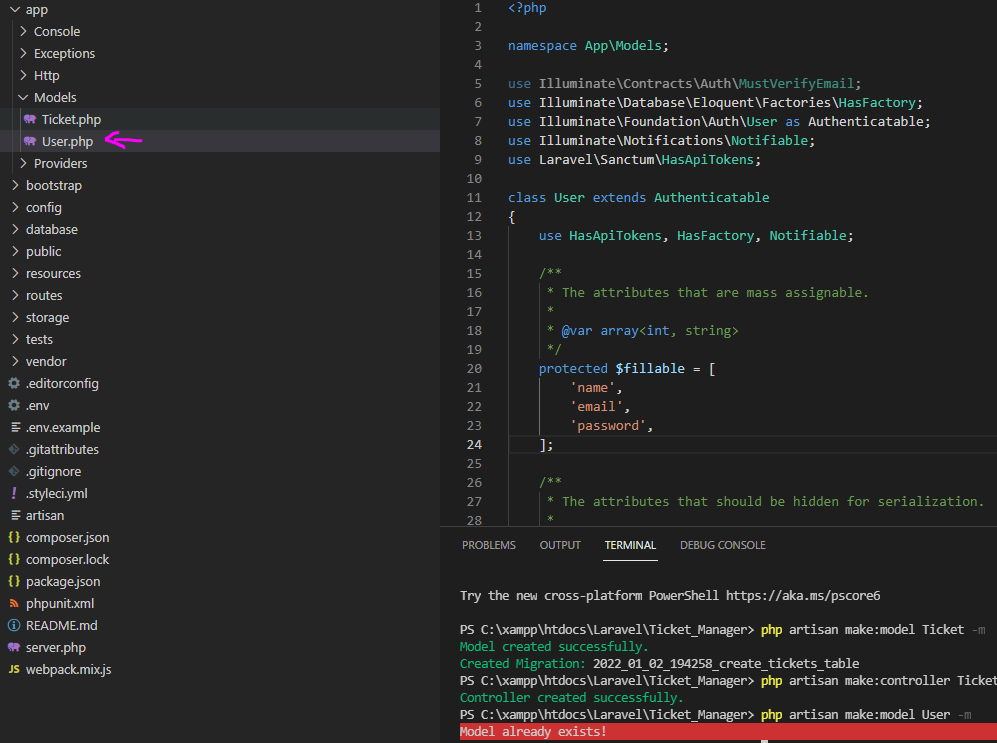


1. User modell

php artisan make:model User -m

// -m jelenti, hogy a modellhez migrációt is létre kívánunk hozni

Megjegyzés: a *User* modell már alapból létezett!



php artisan make:controller UserController -m User

// -m Project a modellhez kapcsolódást jelenti



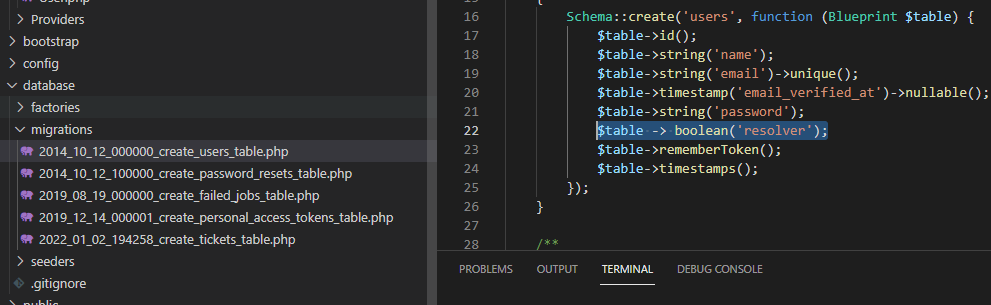
Megjegyzés: jóllehet, a *User* modell már alapból létezett, a UserController viszont nem, így azt külön meg kellett kreálni!

Adatbázis és táblák létrehozása

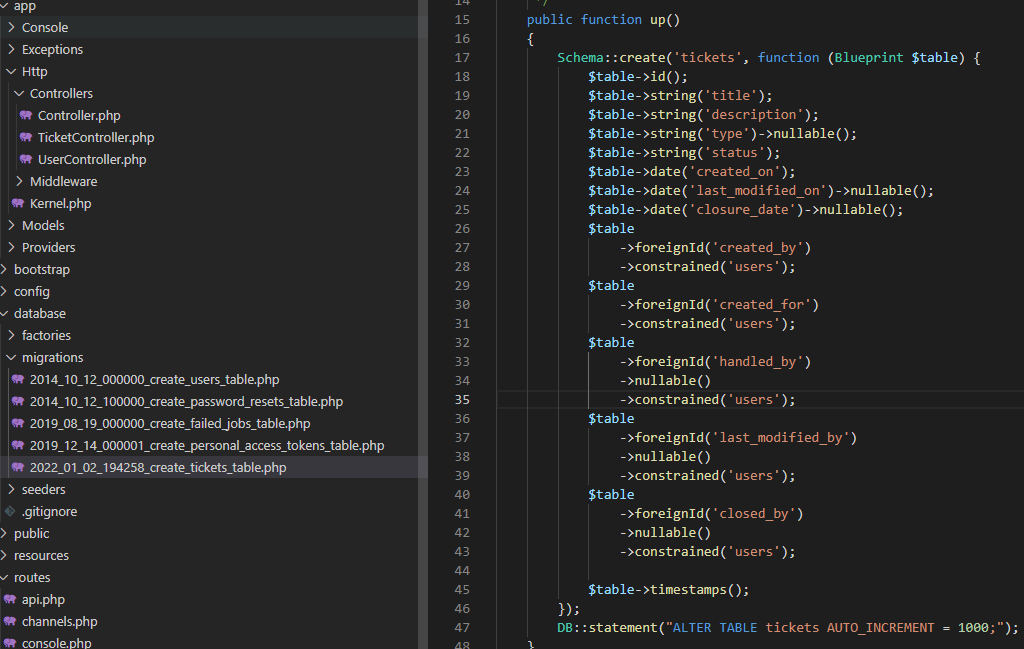
**Adattáblák helye**: database/migrations

1. Users tábla

A default attribútumkészletet kiegészítjük a szükséges („hiányzó”) attribútumokkal (jelen esetben csak a *resolver* nevű, *boolean* típusú oszlop került hozzáadásra):



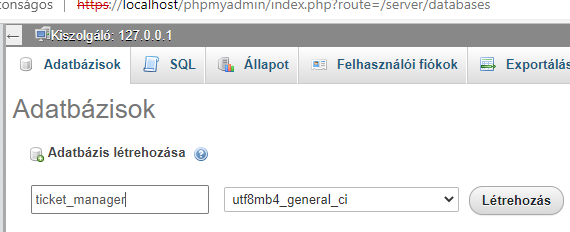
1. Ticket tábla:



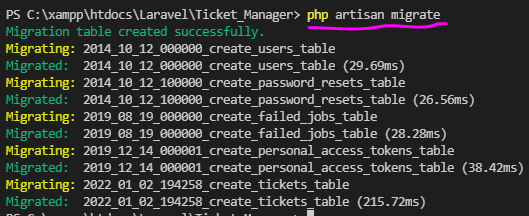
Megjegyzés: mivel egy ticket nem feltétlenül lezárt állapotú, ezért a *closed\_by* és *closure\_date* mezők lehetnek NULL értékűek. Ugyancsak lehet NULL értéke a *type*, *handled\_by*, *last\_modified\_by*, *last\_modified\_on* mezőknek, hiszen érintetlen ticketek esetén ezek nem feltétlenül kitöltöttek.

Ezt reprezentálja a nullable() függvény.

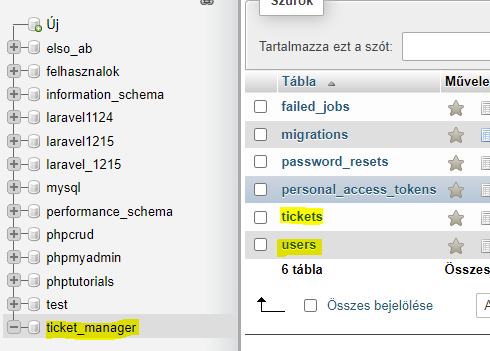
Adatbázis létrehozása PHPMyAdmin-ban:



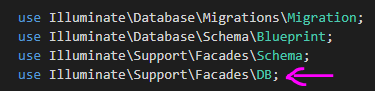
Követhezhet a migráció: php artisan migrate



Létrejöttek a táblák PHPMyAdmin-ban:



Extra: Auto increment funkció beépítése a táblákba:

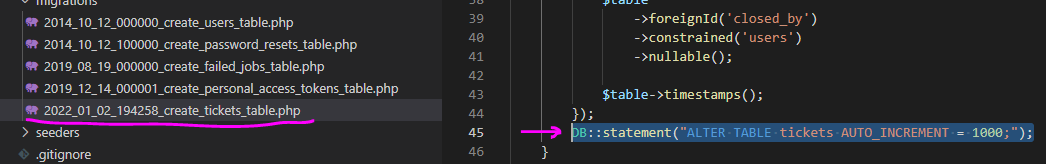


Először is be kell hivatkozni a DB osztályt a migrációs fájl „fejléc” részében:

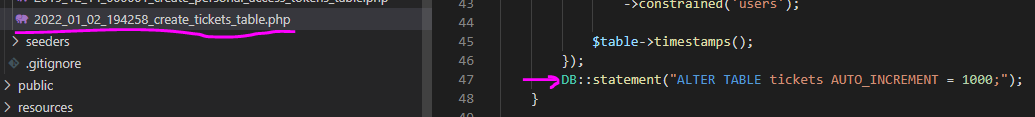
use Illuminate\Support\Facades\DB;

Ezután jöhet az utasítás:

DB::statement("ALTER TABLE tickets AUTO\_INCREMENT = 1000;");



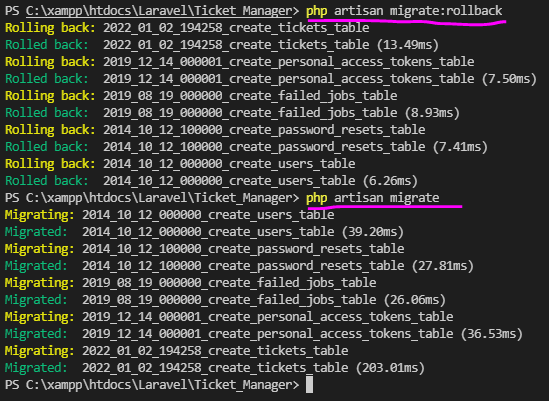
DB::statement("ALTER TABLE users AUTO\_INCREMENT = 100;");



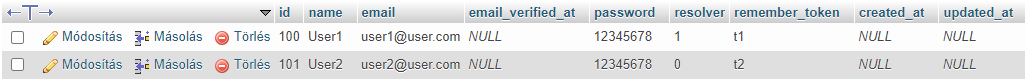
A változtatások „érvényesítéséhez” előbb vissza kell pörgetni az előző migrációt, majd újra végrehajtani:

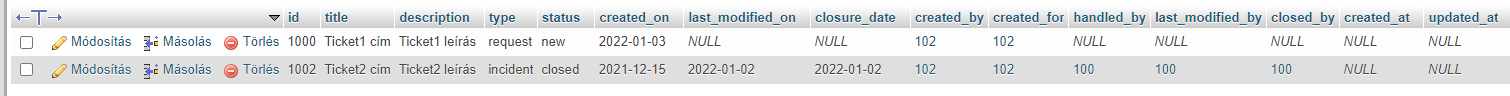
PS C:\xampp\htdocs\Laravel\Ticket\_Manager> php artisan migrate:rollback

PS C:\xampp\htdocs\Laravel\Ticket\_Manager> php artisan migrate

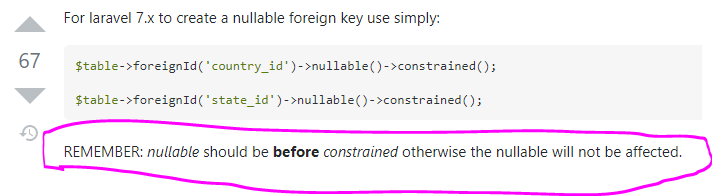


Tesztadatok létrehozása a users táblában PHPMyAdmin-on keresztül:





**Nagyon fontos tanulság**: NULLABLE foreign key helyes szintaxisa:



Forrás: <https://stackoverflow.com/questions/37735055/laravel-database-schema-nullable-foreign>

Különben MyPhpAdmin-ban a következő hibaüzenet fog előjönni:

*„1452 Cannot add or update a child row: a foreign key constraint fails”*

Útvonalak, Controllerek és View-k

**Útvonalak helye:** routes/web.php

**View-k helye:** resources/views/ticket (a ticket foldert manuálisan kreáljuk, ebbe helyezzük a blade-eket)

A használandó útvonalak:

Route::get('/api/tickets', [TicketController::class, 'index']); //összes ticket megjel. (JSON-szerű kimenet)

Route::get('/api/ticket/{id}', [TicketController::class, 'show']); //adott id-jú ticket megjelenítése

Route::post('/api/ticket', [TicketController::class, 'store']); //új rekord létrehozása request alapján

Route::put('/api/ticket/{id}', [TicketController::class, 'update']); //adott id-jú ticket szerkesztése

Route::delete('/api/ticket/{id}', [TicketController::class, 'destroy']); //id alapján töröl

Route::delete('/api/ticket', [TicketController::class, 'destroy2']); //request alapján töröl\*\*\*

Route::get('/ticket/new', [TicketController::class, 'newView']);  //új ticket rögzítése (form)

Route::get('/ticket/edit/{id}', [TicketController::class, 'editView']); //adott id-jú ticket szerkesztése (form)

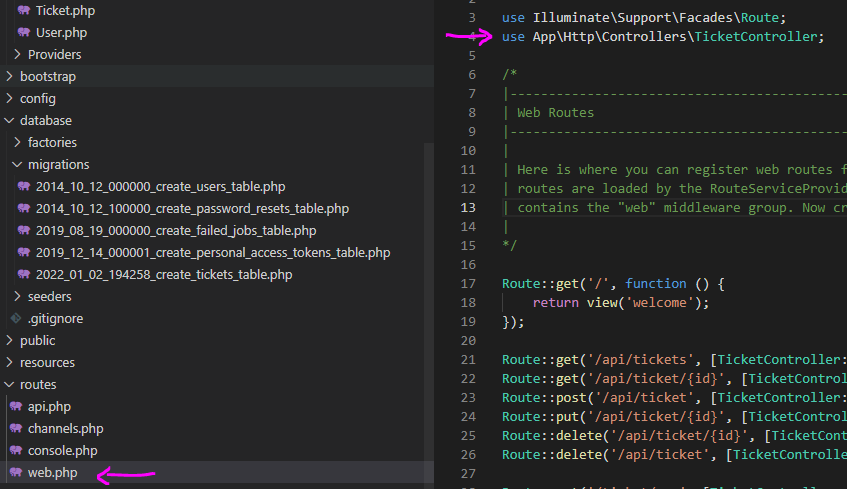
Route::get('/ticket/list', [TicketController::class, 'listView']); //ticketek törlése (form)

Fontos: A web.php-ba be kell hivatkozni a használni kívánt Controllert, különben az alábbihoz hasonló hibaüzenetbe futunk bele:

Illuminate\Contracts\Container\BindingResolutionException

**Target class [TicketController] does not exist.**

use App\Http\Controllers\TicketController;



Összes ticket kilistázása:

JSON-szerű kimenetet kapunk.

Útvonal:

Route::get('/api/tickets', [TicketController::class, 'index']);

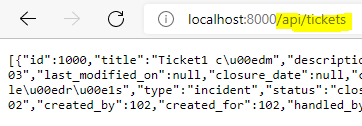
Controller:

   public function index()

    {

        return Ticket::all();

    }



Adott id-jú ticket kiiratása:

Útvonal:

Route::get('/api/ticket/{id}', [TicketController::class, 'show']);

Controller:

 public function show(Ticket $id)

    {

        $ticket = Ticket::find($id);

        return $ticket;

    }

**Authentikáció, felhasználókezelés előkészítése**

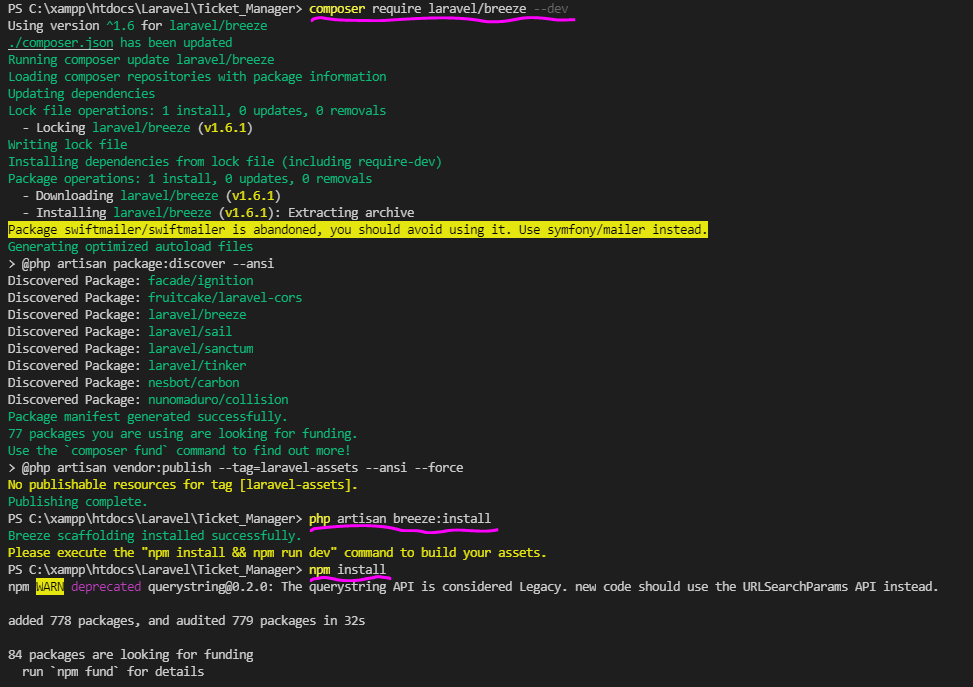
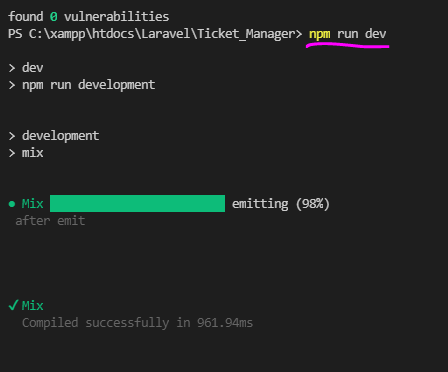
PS C:\xampp\htdocs\Laravel\Ticket\_Manager> composer require laravel/breeze --dev

PS C:\xampp\htdocs\Laravel\Ticket\_Manager> php artisan breeze:install

Breeze scaffolding installed successfully.

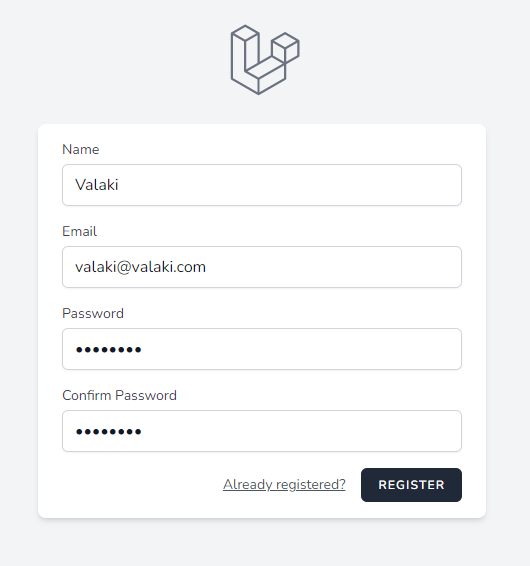
PS C:\xampp\htdocs\Laravel\Ticket\_Manager> npm install

PS C:\xampp\htdocs\Laravel\Ticket\_Manager> npm run dev

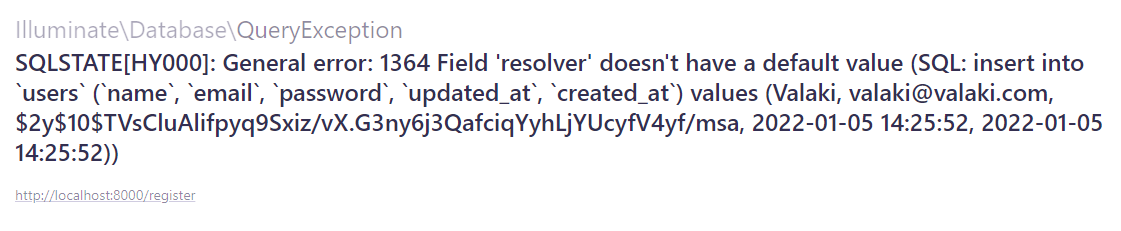


(itt jönne egy php artisan:migrate ill. php artisan serve, ha még nem volt meg)

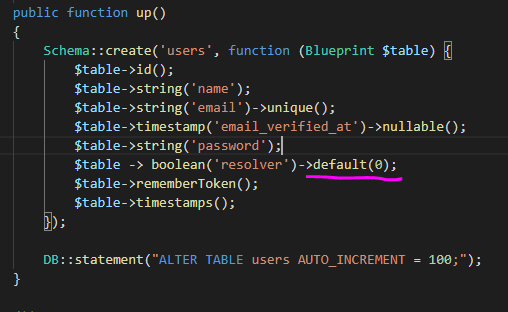
Új regisztráció a <http://localhost:8000/> alatt (<http://localhost:8000/register>):



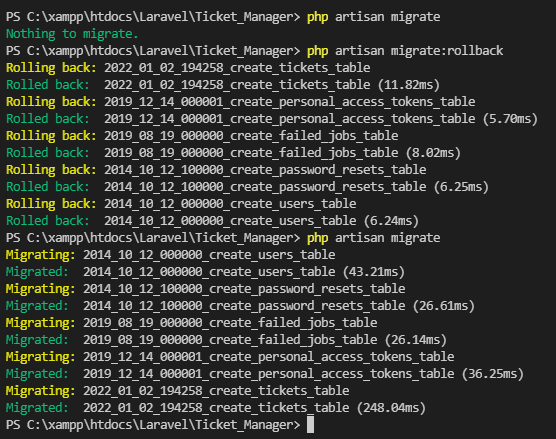
A következő hibaüzenetet kaptam, mivel az eredeti DB-ben vannak olyan mezők is, amelyek a regisztrációs formon nem szerepelnek; ilyen a *resolver* attribútum, amelynek kezdetben nem adtunk default értéket:



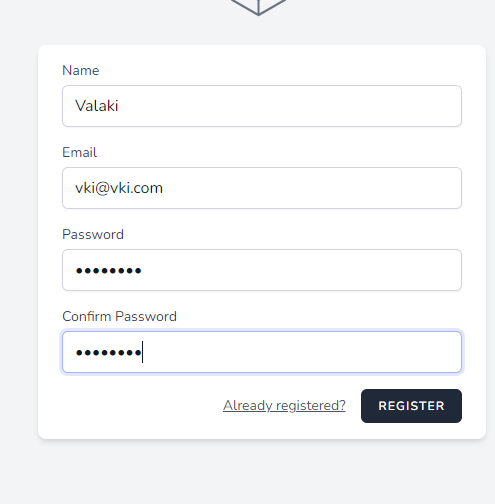
Kiegészítjük tehát a *resolver* mezőt a default értékkel:



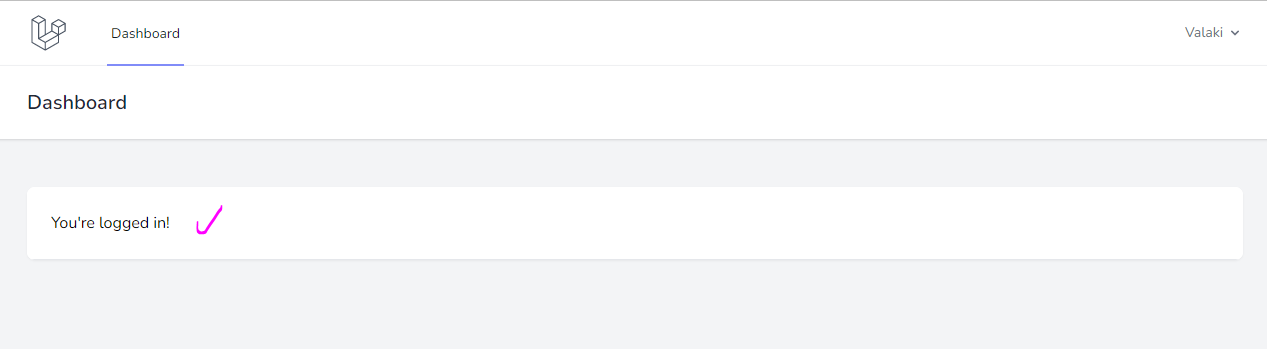
Kell egy php artisan migrate:rollback egy php artisan:migrate

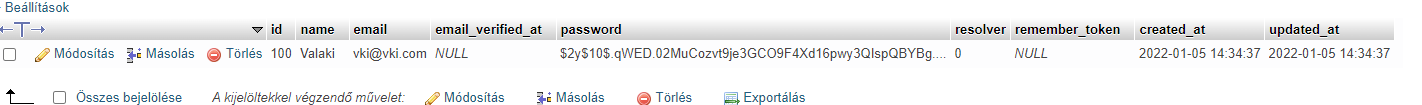


Újra regisztrálunk:



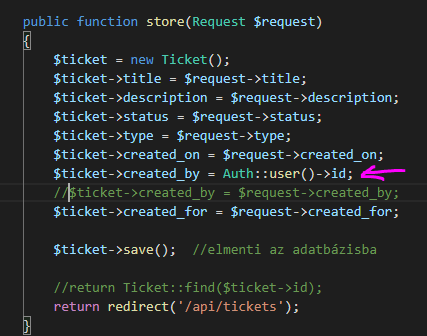
Sikerült:





FIGYELEM! Az előzőleg készített web.php tartalma az authentikáció folyamán eltűnt, és újra fel kellett építeni!

Most már mivel elő van készítve az authentikáció, a TicketController store függvényében beállíthatjuk, hogy csak bejelentkezett felhasználó kreálhasson ticketet:



Middleware-ek

**Middleware-ek helye: app\Http\Middleware**

Létrehozzuk a logikát tartalmazó middleware osztályt:

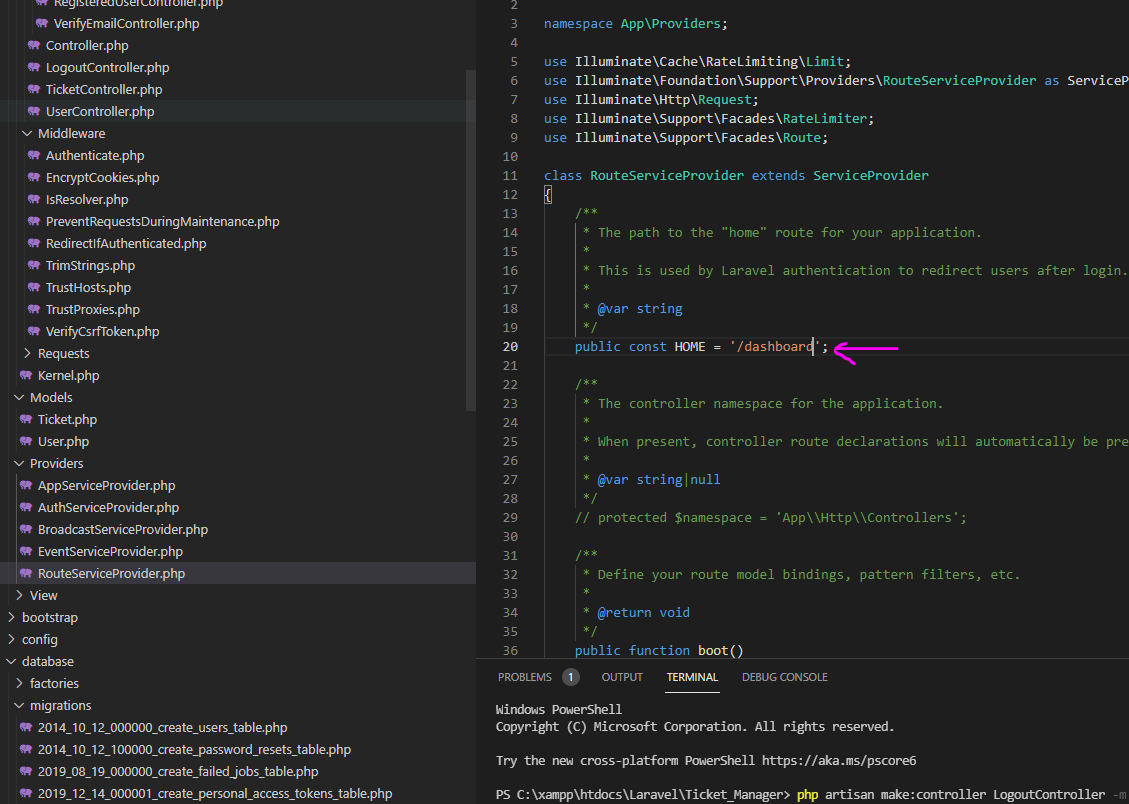
PS C:\xampp\htdocs\Laravel\Ticket\_Manager> php artisan make:middleware IsResolver



Minden bejelentkezés esetén irányítsa át az authentikáció során átengedett usereket az általunk specifikált kezdőoldalra:

<https://stackoverflow.com/questions/60776726/is-there-a-method-to-override-routeserviceproviderhome-in-laravel>

Az app\Providers\RouteServiceProviders.php alatt át kell írni az alapértelmezett útvonalat, amely:



Ebben a feladatban a következőképp történt a korrekció:

 public const HOME = '/api/tickets';