

### 3. feladat - Iluska

cmd

```
composer create-project laravel/laravel harmadikfeladat --prefer-dist
cd harmadikfeladat
composer require laravel/breeze --dev
php artisan breeze:install
api
0-pest
```

---

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=harmadikfeladat
DB_USERNAME=root
DB_PASSWORD=
```

---

```
php artisan make:model Foglalas -m
```

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
```

```
class CreateFoglalasokTable extends Migration
{
    /**
     * Futtatja a migrációt: létrehozza a foglalások táblát.
     */
    public function up()
    {
        Schema::create('foglalasok', function (Blueprint $table) {
            $table->id(); // Egyedi azonosító (ID)
            $table->string('targy_neve'); // Foglálás tárgya neve
            $table->string('kategoria'); // Kategória (pl. autó, festmény)
```

```

        $table->decimal('napi_dij', 8, 2); // Napi bérleti díj
        $table->date('kezdes_datum'); // Bérlet kezdő dátuma
        $table->date('vege_datum'); // Bérlet vég dátuma
        $table->decimal('osszes_ar', 10, 2); // Összes ár kiszámítva
        $table->timestamps(); // Létrehozás és módosítás időbélyege
    });
}

/**
 * A migráció visszavonása: törli a foglalások táblát.
 */
public function down()
{
    Schema::dropIfExists('foglalások');
}
}

```

---

php artisan migrate

php artisan make:controller FoglalasController

App/Http/Controllers/FoglalasController.php

```
<?php
```

```

namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Foglalas;

class FoglalasController extends Controller
{
    /**
     * Foglalások listázása (GET /api/foglalások).
     */
    public function index()
    {
        return response()->json(Foglalas::all());
    }
}

```

```

}

/**
 * Új foglalás létrehozása (POST /api/foglalasok).
 */
public function store(Request $request)
{
    $request->validate([
        'targy_neve' => 'required|string|max:255',
        'kategoria' => 'required|string|max:255',
        'napi_dij' => 'required|numeric|min:0',
        'kezdes_datum' => 'required|date',
        'vege_datum' => 'required|date|after:kezdes_datum',
    ]);

    // Összesített ár kiszámítása
    $napok_szama = (strtotime($request->vege_datum) -
    strtotime($request->kezdes_datum)) / (60 * 60 * 24);
    $ossz_ar = $napok_szama * $request->napi_dij;

    // Foglalás mentése
    $foglalas = Foglalas::create([
        'targy_neve' => $request->targy_neve,
        'kategoria' => $request->kategoria,
        'napi_dij' => $request->napi_dij,
        'kezdes_datum' => $request->kezdes_datum,
        'vege_datum' => $request->vege_datum,
        'ossz_ar' => $ossz_ar,
    ]);

    return response()->json($foglalas, 201);
}

/**
 * Egy adott foglalás megjelenítése (GET /api/foglalasok/{id}).
 */
public function show($id)
{
    $foglalas = Foglalas::find($id);

    if (!$foglalas) {
        return response()->json(["message" => "A megadott

```

```

foglalás nem található!"], 404);
    }

    return response()->json($foglalas);
}

/**
 * Foglalás módosítása (PUT /api/foglalasok/{id}).
 */
public function update(Request $request, $id)
{
    $foglalas = Foglalas::find($id);

    if (!$foglalas) {
        return response()->json(["message" => "A megadott
foglalás nem található!"], 404);
    }

    $request->validate([
        'targy_neve' => 'required|string|max:255',
        'kategoria' => 'required|string|max:255',
        'napi_dij' => 'required|numeric|min:0',
        'kezdes_datum' => 'required|date',
        'vege_datum' => 'required|date|after:kezdes_datum',
    ]);

    // Összesített ár frissítése
    $napok_szama = (strtotime($request->vege_datum) -
strtotime($request->kezdes_datum)) / (60 * 60 * 24);
    $ossz_ar = $napok_szama * $request->napi_dij;

    $foglalas->update([
        'targy_neve' => $request->targy_neve,
        'kategoria' => $request->kategoria,
        'napi_dij' => $request->napi_dij,
        'kezdes_datum' => $request->kezdes_datum,
        'vege_datum' => $request->vege_datum,
        'ossz_ar' => $ossz_ar,
    ]);

    return response()->json($foglalas);
}

```

```

/**
 * Foglалás törlése (DELETE /api/foglалasok/{id}).
 */
public function destroy($id)
{
    $foglалas = Foglалas::find($id);

    if (!$foglалas) {
        return response()->json(["message" => "A megadott
foglалás nem található!"], 404);
    }

    $foglалas->delete();
    return response()->json(["message" => "Foglалás sikeresen
törölve"]);
}
}

```

#### **app/Models/Foglалas.php**

```

<?php

namespace App\Models;

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

class Foglалas extends Model
{
    use HasFactory;

    /**
     * Az adatbázis tábla neve.
     *
     * Laravel automatikusan a többes számú alakot használja,
     * de ha pontosítani akarod, akkor megadhatod itt manuálisan:
     */
    protected $table = 'foglалasok';

    /**
     * A kitölthető oszlopok a tömeges hozzárendeléshez.

```

```

        */
        protected $fillable = [
            'targy_neve',
            'kategoria',
            'napi_dij',
            'kezdes_datum',
            'vege_datum',
            'osszes_ar'
        ];
    }

```

### Routes/api.php

```

<?php

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\FoglalasController;

/*
|-----
|
| API Routes
|-----
|
| Itt adhatók meg az API végpontok, amelyeket a frontend vagy külső
| kliensek használhatnak az adatok elérésére.
|
*/

// Foglalások API útvonalai
Route::get('/foglalasok', [FoglalasController::class, 'index']); //
Összes foglalás lekérése
Route::post('/foglalasok', [FoglalasController::class, 'store']); //
Új foglalás rögzítése
Route::get('/foglalasok/{id}', [FoglalasController::class, 'show']);
// Egy adott foglalás megtekintése
Route::put('/foglalasok/{id}', [FoglalasController::class,
'update']); // Foglalás módosítása

```

```
Route::delete('/foglalasok/{id}', [FoglalasController::class,
'destroy']); // Foglálás törlése
```

```
/*
|-----
-----
| Autentikációs és felhasználói műveletek (példa)
|-----
-----
*/
```

```
// Ha szeretnél autentikációt hozzáadni, használhatsz Laravel
Passport vagy Sanctum rendszert.
// Példa útvonalak:
Route::middleware('auth:sanctum')->get('/user', function (Request
$request) {
    return $request->user();
});
```

xampple

```
CREATE TABLE foglalasok ( id INT AUTO_INCREMENT PRIMARY KEY, tárgy_neve
VARCHAR(255) NOT NULL, kategoria VARCHAR(255) NOT NULL, napi_dij DECIMAL(8,2)
NOT NULL, kezdes_datum DATE NOT NULL, vege_datum DATE NOT NULL, osszes_ar
DECIMAL(10,2) NOT NULL, created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP );
```

```
Invoke-WebRequest -Uri "http://127.0.0.1:8000/api/foglalasok" -Method GET
```

```
(Invoke-WebRequest -Uri "http://127.0.0.1:8000/api/foglalasok" -Method GET).Content
```

```
-----
Invoke-WebRequest -Uri "http://127.0.0.1:8000/api/foglalasok" -Method POST -Body
(@{
```

```
    tárgy_neve = "Autó";
```

```
kategoria = "Jármű";
```

```
napi_dij = 5000;
```

```
kezdes_datum = "2025-05-20";
```

```
vege_datum = "2025-05-25"
```

```
} | ConvertTo-Json) -ContentType "application/json"
```

---

```
Invoke-WebRequest -Uri "http://127.0.0.1:8000/api/foglalasok/1" -Method GET
```