

ILUSKA

E meghajtót kellett eddig használni, de lehet vizsgán másikat kell, hogy működjön minden.

1 feladat

Laravel telepítése

```
composer create-project laravel/laravel elsofeladat --prefer-dist
```

```
cd elsofeladat
```

```
composer require laravel/breeze --dev  
php artisan breeze:install  
api
```

0-pest

```
code .
```

Nyissuk meg a .env fájlt, és frissítsük az adatbázis beállításokat:

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

Ezután hozzuk létre az adatbázist MySQL-ben / elsofeladatadatbazisa

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

database/migrations/{timestamp} create bookings table.php

```
<?php
```

```

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

return new class extends Migration {
    public function up()
    {
        Schema::create('bookings', function (Blueprint $table) {
            $table->id();
            $table->integer('user_id');
            $table->integer('item_id');
            $table->date('start_date');
            $table->date('end_date');
            $table->integer('daily_price');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('bookings');
    }
};

```

php artisan migrate

php artisan make:seeder BookingSeeder

database/seeder/BookingSeeder.php

```
<?php
```

```
namespace Database\Seeders;
```

```
use Illuminate\Database\Seeder;
```

```
use Illuminate\Support\Facades\DB;
```

```
class BookingSeeder extends Seeder
```

```
{
```

```
    public function run()
```

```
    {
```

```
        DB::table('bookings')->insert([
```

```
            [
```

```
                'user_id' => 1,
```

```
                'item_id' => 101,
```

```
                'start_date' => '2025-05-15',
```

```
                'end_date' => '2025-05-20',
```

```
                'daily_price' => 5000,
```

```
            ],
```

```
            [
```

```
                'user_id' => 2,
```

```
                'item_id' => 102,
```

```
                'start_date' => '2025-06-01',
```

```
                'end_date' => '2025-06-10',
```

```
                'daily_price' => 7500,
```

```
    ]  
  });  
}  
}
```

```
php artisan db:seed --class=BookingSeeder
```

```
php artisan make:controller BookingController
```

app/Http/Controllers/BookingController.php

```
<?php  
  
namespace App\Http\Controllers;  
  
use App\Models\Booking;  
use Illuminate\Http\Request;  
  
class BookingController extends Controller  
{  
    public function index()  
    {  
        return response()->json(Booking::all());  
    }  
  
    public function store(Request $request)  
    {  
        $request->validate([  
            'user_id' => 'required|integer',  
            'item_id' => 'required|integer',  
            'start_date' => 'required|date',  
            'end_date' => 'required|date|after:start_date',  
            'daily_price' => 'required|integer',  
        ]);  
  
        $booking = Booking::create($request->all());  
        return response()->json($booking, 201);  
    }  
}
```

```
}  
}
```

php artisan make:controller ReserveController

app/Http/Controllers/ReserveController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
class ReserveController extends Controller
```

```
{
```

```
    /**
```

```
     * CSV fájl beolvasása és adatok visszaadása
```

```
     */
```

```
    public function loadCsvData($filename)
```

```
    {
```

```
        $path = base_path("{ $filename }");
```

```
        if (!file_exists($path)) {
```

```
            return response()->json(['error' => 'CSV fájl nem található!'], 404);
```

```
        }
```

```
        $file = fopen($path, 'r');
```

```
        $header = fgetcsv($file);
```

```
        $data = [];
```

```
while ($row = fgetcsv($file)) {  
    $data[] = array_combine($header, $row);  
}  
  
fclose($file);  
return response()->json($data);  
}
```

```
/**  
 * Foglalások listázása (1.csv)  
 */  
public function listReserves()  
{  
    return $this->loadCsvData('1.csv');  
}
```

```
/**  
 * Tárgyak listázása (2.csv)  
 */  
public function listObjects()  
{  
    return $this->loadCsvData('2.csv');  
}
```

```
/**  
 * Új foglalás mentése a CSV fájlba  
 */
```

```

public function store(Request $request)
{
    $request->validate([
        'customer_id' => 'required|integer',
        'object_id' => 'required|integer',
        'start_period' => 'required|date',
        'end_period' => 'required|date|after:start_period',
        'rate_per_day' => 'required|integer',
    ]);

    $filename = base_path("1.csv");
    $file = fopen($filename, 'a');

    if ($file) {
        fputcsv($file, [
            $request->customer_id,
            $request->object_id,
            $request->start_period,
            $request->end_period,
            $request->rate_per_day
        ]);
        fclose($file);
    } else {
        return response()->json(['error' => 'Nem sikerült megnyitni a CSV fájlt!'], 500);
    }

    return response()->json(['message' => 'Foglalás sikeresen hozzáadva a CSV fájlhoz!'], 201);
}

```

```
}  
}
```

routes/api.php

```
<?php
```

```
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\Route;  
use App\Http\Controllers\ReserveController;
```

```
// Foglalások és tárgyak lekérdezése
```

```
Route::get('/reserves', [ReserveController::class, 'listReserves']);
```

```
Route::get('/objects', [ReserveController::class, 'listObjects']);
```

```
// Új foglalás létrehozása
```

```
Route::post('/reserves', [ReserveController::class, 'store']);
```

7. Teszteljük az API végpontokat

```
php artisan serve
```

Teszteljük az API-t Postman vagy cURL segítségével:

Lekérdezés az összes foglalásra


```
Invoke-WebRequest -Uri http://127.0.0.1:8000/api/reserves -Method  
Get
```

```
Invoke-RestMethod -Uri http://127.0.0.1:8000/api/bookings -Method  
Get
```

Új foglalás rögzítése

```
Invoke-WebRequest -Uri http://127.0.0.1:8000/api/reserves -Method  
Post -Body '{"customer_id":6, "object_id":106, "start_period":"2025-  
09-10", "end_period":"2025-09-15", "rate_per_day":8500}' -  
ContentType "application/json"
```