

# Laravel projektfeladat: Asztalfoglalás REST API (authentikáció nélkül)

Készíts egy Laravel REST API-t, amely lehetővé teszi a „Kisújj Étterem” asztalfoglalásának online kezelését. A rendszer nem igényel bejelentkezést, tehát **nyilvános API**.

## I. Felvonás: Alapok és adatbázis-előkészítés

Ez a fázis az infrastruktúra beállítását és az adatmodell elkészítését tartalmazza.

- 1. Környezet és projektlétrehozás:** Elindítjuk a webkiszolgálót (XAMPP Apache | php artisan serve) és a MySQL szerveret (XAMPP MySQL) illetve létrehozuk a Laravel projektet (reservationKisuj).
- 2. API alapok beállítása:** Létrehozuk API útvonalakat tartalmazó fájlt, valamint beállítunk egy teszt végpontot (/hello) az útvonal működésének azonnali ellenőrzésére (pl: POSTMAN).
- 3. Adatmodell és migráció létrehozása:** Létrehozuk a Reservation nevű modellt és a hozzá tartozó migrációs fájlt, amit később meg is szerkesztünk.
- 4. Adatbázis konfiguráció:** Létrehozuk a szükséges adatbázist (pl. reservationkisuj), és beállítjuk a kapcsolódási adatokat a környezeti fájlban (.env).
- 5. Migráció szerkesztése és futtatása:** Meghatározzuk az reservations tábla mezőit (például name, email, reservation\_time, guests, note).
- 6. Modell tulajdonságainak beállítása:** Kitöltjük a Reservation modellben a tömeges hozzárendeléshez engedélyezett mezők listáját (\$fillable).
- 7. Tesztadatok feltöltése (Seeding):** A ReservationFactory és ReservationSeeder segítségével életszerű, tesztadatokat (10 rekordot) generáltatunk és töltünk fel az adatbázisba (phpmyadmin-ban ellenőrizzük az adatbázist).

## II. Felvonás: Controller és endpointok implementálása

Ebben a fázisban a tényleges üzleti logika és az API végpontok kialakítása történik meg.

- 1. Controller létrehozása:** Létrehozuk az API logikát tartalmazó ReservationController-t a Http\Controllers\Api könyvtárban belül.
- 2. CRUD logika implementálása:** Megírjuk a Controller metódusait:
  - index() és show(): Foglalások lekérdezése.
  - store(): Új foglalás létrehozása, a bemenő adatok kötelező validálásával.
  - update(): Foglалás módosítása (PUT/PATCH), a részleges frissítéshez szükséges feltételes validációval (sometimes).
  - destroy(): Foglалás törlése.
- 3. Státusz kódok kezelése:** Biztosítjuk a megfelelő HTTP státusz kódok visszaküldését (például 200 OK a sikeres lekérdezéshez és módosításhoz, 201 Created az új rekord létrehozásához, és 404 Not Found a nem létező erőforrás esetén).
- 4. Útvonalak definiálása:** Összekötjük a Controller metódusait a nyilvános API végpontokkal az api.php fájlban (egyenként definiálva | Laravel beépített apiResource parancsával).
- 5. Záró beállítások:** Beállítjuk az időzónát (pl. Europe/Budapest)
- 6. Tesztelés:** POSTMAN segítségével teszteljük az összes végpont (GET, POST, PUT, PATCH, DELETE) működését.
- 7. API dokumentációt készítünk**