

# F1 Garage Marketplace – feladatleírás

Készítette: Szomor Levente SZFT5

Dátum: 2026-02-23

Fullstack (Laravel + Vue)

## Áttekintés

Készítsen egy egyszerű piactér rendszert, ahol a Formula 1 paddockban elérhető tételek (listings) kezelhetők, és a tételekhez opcionálisan beszállító (vendor) rendelhető. A backend REST API-t biztosít, a frontend pedig státusz-oszlopos nézetben jeleníti meg a hirdetéseket.

API base URL: `http://backend.vm1.test/`

## Backend programozás (Laravel)

A feladat megoldásához hozzon létre migrációkat, modelleket, FormRequest-eket és controllereket az alábbi specifikáció szerint.

Tábla	Mezők	Megjegyzés
vendors	id (AI, PK), name (varchar 50), paddock_zone (varchar 4	0), discount (int), timestamps, A beszállító neve, helye a paddockban
listings	id (AI, PK), title (varchar 100), status (varchar 20, default c quanti	Státusz: open, negotiating, sold, archived.

Kapcsolatok:

- Vendor: hasMany Listing
- Listing: belongsTo Vendor (nullable) Routing (azonosító csak szám lehet):

Metódus	Végpont	Controller@action
GET	/api/listings	ListingController@index
GET	/api/listings/{id}	ListingController@show
POST	/api/listings	ListingController@store
PUT	/api/listings/{id}	ListingController@update
DELETE	/api/listings/{id}	ListingController@destroy
GET	/api/vendors	VendorController@index
GET	/api/vendors/{id}	VendorController@show
POST	/api/vendors	VendorController@store
PUT	/api/vendors/{id}	VendorController@update

DELETE	/api/vendors/{id}	VendorController@destroy
--------	-------------------	--------------------------

Validálás – Listing:

- title kötelező, max 100 karakter
- status nem kötelező (DB default: open), csak: open|negotiating|sold|archived
- vendor\_id nem kötelező; ha meg van adva, csak létező Vendor azonosító lehet; üres érték esetén NULL
- price kötelező, egész szám, minimum 0
- quantity kötelező, egész szám, minimum 0 • description kötelező, min 10, max 2000 karakter

Vendor:

- name kötelező, max 50 karakter
- paddock\_zone kötelező, max 40 karakter
- discount kötelező, egész szám, 0–30 Válaszformátum:
- Minden sikeres válasz JSON legyen, egységesen `data` kulccsal (pl. `{ "data": {...} }`).
- Sikeres törlés esetén 204 No Content választ adjon.

Seedelés:

- A mellékelt `VendorSeeder\_F1.php` és `ListingSeeder\_F1.php` alapján töltse fel az adatbázist.
- Állítsa be, hogy a seederek megfelelő sorrendben fussanak le.

## Swagger / OpenAPI

Az API leírását a mellékelt `openapi\_f1\_garage\_marketplace.yaml` tartalmazza. A dokumentáció ennek megfelelően legyen elérhető.

## Frontend programozás (Vue 3 + Pinia)

Készítse el az alábbi útvonalakat (a tanult routing módszerek egyikével):

Útvonal neve	URL
home	/
create-listing	/listings/create
edit-listing	/listings/:id
vendors	/vendors
vendor	/vendors/:id

Állítsa be, hogy a router betöltődése előtt az oldal címe legyen az adott oldal meta.title értéke.

Tárolók (Pinia):

- ListingStore: tickets mintájára listings állapot + getListings/getListing/createListing/updateListing/deleteListing műveletek.
- VendorStore: vendors állapot + getVendors/getVendor műveletek.
- ListingStore statuses getter: open, negotiating, sold, archived -> magyar címkékkel: Nyitott, Tárgyalás alatt, Eladva, Archivált.

Oldalak:

- Főoldal: 4 oszlop státusz szerint (kis nézeten egymás alatt, nagy nézettől 4 oszlopos grid).
- BaseCard: listing title + vendor neve (ha van). Kattintás: szerkesztő oldal.
- Új listing: űrlap (title, status select, vendor select, price, quantity, description). Mentés után vissza a főoldalra.
- Szerkesztés: előtöltés, update + delete (confirm). Siker után vissza a főoldalra.
- Vendors oldal: BaseTable (név + hozzá rendelt listingek száma).
- Vendor részletek: státusz-oszlopos nézetben csak az adott vendor listingjei.

## Tesztelés

Tesztelje az API-t Swagger UI / Thunder Client / Bruno segítségével. Ellenőrizze a 422 validációs hibákat és a 204 törlést.

Tesztelje a frontend adatfolyamot: betöltéskor a store-ok kérjék le az adatokat, a státusz-oszlopok helyesen rendezzenek.