

NBA Garage Marketplace - feladatleírás

Készítette: Szomor Levente SZFT5

Dátum: 2026-02-27

Készítsen egy egyszerű NBA reliktia piacteret, ahol hirdetéseket lehet felvenni, módosítani, törleni, és eladókhöz (vendors) rendelni.

A feladat két részből áll: backend (Laravel API) és frontend. A kommunikáció JSON formátumban történjen.

Backend programozás

Adatbázis és modellek

1. Hozzon létre migrációkat, amelyek létrehozzák a vendors és listings táblákat. Mindkét táblában legyenek időbényegek (created_at, updated_at).

vendors tábla mezők:

- id – egész [AI, PK]
- name – szöveg (max 60)
- city – szöveg (max 40)
- rating – egész (1..5)
- timestamps

listings tábla mezők:

- id – egész [AI, PK]
- vendor_id – egész [FK] (kötelező)
- category – szöveg (max 20) (engedélyezett: jersey, shoe, card, relic)
- title – szöveg (max 80)
- team – szöveg (max 50)
- price – egész (min 1)
- stock – egész (min 0)
- is_active – логикai (default: true)
- listed_from – dátum
- listed_to – dátum vagy NULL
- timestamps

2. Állítsa be a kapcsolatot: listings.vendor_id → vendors.id (ON DELETE CASCADE, ON UPDATE CASCADE).

3. Hozza létre a Vendor és Listing modelleket, valamint a kapcsolatokat:

- Vendor: hasMany(listings)
 - Listing: belongsTo(vendor)
4. Seedelés: használja a mellékelt VendorSeeder_NBA.php és ListingSeeder_NBA.php fájlokat.
Gondoskodjon róla, hogy a seederek megfelelő sorrendben fussanak le. Futtassa a migrációt és a seedelést is.

Routing

5. Valósítsa meg az alábbi végpontokat. Ügyeljen rá, hogy az azonosító csak szám lehet (route constraint)!

Vendors

- GET /api/vendors - VendorController@index
- GET /api/vendors/{vendor} - VendorController@show
- POST /api/vendors - VendorController@store
- PUT /api/vendors/{vendor} - VendorController@update
- DELETE /api/vendors/{vendor} - VendorController@destroy

Listings

- GET /api/listings - ListingController@index
- GET /api/listings/{listing} - ListingController@show
- POST /api/listings - ListingController@store
- PUT /api/listings/{listing} - ListingController@update
- DELETE /api/listings/{listing} - ListingController@destroy

Validálás

6. Jogosultságot nem kell ellenőrizni. Hibára 422 választ adjon validációs üzenetekkel.
7. Vendor validálási szabályok:

- name: kötelező, max 60 karakter
- city: kötelező, max 40 karakter
- rating: kötelező, egész, 1..5

8. Listing validálási szabályok:

- vendor_id: kötelező, létező vendor azonosító
- category: kötelező, értéke csak: jersey, shoe, card, relic
- title: kötelező, 3..80 karakter

- team: kötelező, 3..50 karakter
- price: kötelező, egész, min 1
- stock: kötelező, egész, min 0
- listed_from: kötelező, date
- listed_to: opcionális, date, de ha meg van adva: listed_to \geq listed_from
- is_active: opcionális boolean (ha nincs: marad default true)

Controllerek

9. VendorController metódusok:

- index: adja vissza az összes vendort; mindenikhez tartozzanak a hirdetések is (eager load)
- show: adja vissza a megadott vendort a hirdetéseivel
- store: validálás után hozza létre az új vendort, és adja vissza
- update: módosítsa a megadott vendort, és adja vissza
- destroy: törölje a vendort (204 No Content)

10. ListingController metódusok:

- index: adja vissza a hirdetéseket; támogassa a szűrést query paraméterekkel:
 - category (pontos egyezés)
 - team (részleges egyezés, case-insensitive LIKE)
 - active (true/false) \rightarrow is_active alapján
- show: adja vissza a hirdetést a vendor adataival együtt
- store/update: validálás után mentés, majd visszaadás
- destroy: törlés (204 No Content)

API dokumentáció (Swagger)

11. Az API leírását a Swagger tartalmazza. A mellékelt openapi_nba_garage_marketplace.yaml alapján legyen elérhető a dokumentáció.

Frontend programozás

Útvonalak

1. Készítse el az alábbi útvonalakat (bármely tanult megoldással):

- / – hirdetések (lista + szűrők)
- /listings/create – új hirdetés
- /listings/:id – hirdetés szerkesztése
- /vendors – eladók lista
- /vendors/:id – eladó részletek (hirdetések oszlopokban)

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

Tárolók (Store-ok)

3. Készítsen ListingStore tárolót: state: listings.

- getListings(filters)
- getListing(id)
- createListing(listing)
- updateListing(id, listing)
- deleteListing(id)
- getter: categories() – jersey/shoe/card/relic magyar nevekkel (Mez, Cipő, Kártya, Relikvia)

4. Készítsen VendorStore tárolót: state: vendors.

- getVendors()
- getVendor(id)
- getter: vendorOptions() – id→name párok; első elem legyen üres kulcs/érték, ha nincs vendor kiválasztva

5. Betöltés: a gyökérkomponens betöltésekor kérje le a szükséges adatokat (vendors + listings).

Oldalak

6. Főoldal (Hirdetések):

- Szűrők: kategória, active (checkbox), team (szöveges keresés)
- Kártyákon jelenjen meg: title, team, category (magyar), price, stock, vendor neve
- Kártyára kattintva navigáljon a szerkesztés oldalra

7. Új hirdetés oldal:

- Śrlap mezők: vendor (select), category (select), title, team, price, stock, listed_from, listed_to (opcionális), is_active (checkbox)

- Mentés után visszairányítás a főoldalra

8. Hirdetés szerkesztése oldal:

- Betöltéskor kérje le a hirdetést id alapján; amíg tölt, jelenjen meg betöltés jelzés

- Gombok: Mentés, Törlés (confirm után)

- Mentés/törlés után vissza a főoldalra

9. Eladók oldal:

- Táblázat: név, város, értékelés, hirdetések száma

- Sorra kattintva navigáljon az eladó részletek oldalra

10. Eladó részletek:

- Főcím: eladó neve

- Hirdetések kategória szerint 4 oszlopban (Mez/Cipő/Kártya/Relikvia), kártyás megjelenítéssel

Tesztelés

11. Tesztelje az API-t böngészőben, Thunder Clientben vagy Bruno-val; ellenőrizze a 404/422 eseteket és a szűrések működését.