

An isometric illustration featuring the Node.js logo in the center. Surrounding it are various geometric shapes and icons representing development and deployment: a person at a computer, a server rack, a database cylinder, a cloud, and a network diagram. The background is a mix of green and blue geometric patterns.

L3: Node.js laborgyakorlatok

Első rész: node.js projekt

- A jelen laborfeladattól kezdődően minden repóban egy darab építkező projektet szeretnénk látni. Ennek értelmében **ne bontsuk többet külön alfolderekbe** a feladatokat. Csak egy projekt legyen, s ez induljon a repó gyökeréből.
- Ellenőrizzük, hogy a tárolónk gyökere tartalmazzon egy `package.json` állományt. Ez lesz a projektünk telepítés-deszkriptora.
- A kliensoldali kódunkat továbbra is tároljuk kizárólag a `public` almappában, ahonnan `express` segítségével szolgáljuk a kliensnek. Az első 2 laborfeladat állományait törölhetjük innen, mivel a git úgyis megjegyzi visszamenőleg.
- **Minden használt szerveroldali függőséget adjunk meg a `package.json`-ünkben** (használjuk az `npm i --save <packagename>` vagy `npm i --save-dev <packagename>` parancsokat az automatikus frissítésért).
- Továbbra is feltétel, hogy ne legyenek `ESLint/Prettier` warningok/errorok a kódban (ld. korábbi laborfeladatok leírása), s hogy zöld legyen az utolsó pipeline kimenete.

Második rész: formfeldolgozás

- Ebben a feladatban kientstől érkezett információkat szükséges feldolgozni és validálni egy `express` szerver segítségével. A kliensoldal legyen egy vagy több `express public`-kal felszolgált HTML/CSS oldal, amely tartalmazza az alábbiak alapján megadott `formokat`.
- **Az adatok validálását végezzük el frontenden és backenden is**, mivel Postmanből bárki leadhat validálatlan adatot. Hibás adatok esetén küldjünk vissza érthető magyarázatot és **megfelelő HTTP státuszkódot**. Használjunk a formok leadásakor a helyzetnek **megfelelő HTTP metódust** (`GET` vagy `POST`).
- Állományfeltöltés esetén tároljuk az állományokat egy új almappában (ezt helyezzük `.gitignore`-ba, hogy a verziókövetőnk ne kövesse). Más feltöltött adatot tárolhatunk memóriában **logikusan strukturált objektumok formájában** vagy JSON állományban. Loggoljunk minden lépést megfelelően. Adatbázis használata megengedett, de nem kötelező (ha mégis használnánk, minden szükséges felépítési szkript legyen a repón a reprodukálhatóság érdekében).

3.1. Lakáshirdetések

- Készítsünk egy formot, amellyel felhasználók lakáshirdetéseket tölthetnek fel a rendszerbe. Egy hirdetés tartalmazza a lakás címét (város, negyed), felszínterületét, árát, szobák számát, feltöltés dátumát, stb. A hívás térítsen vissza egy generált egyedi ID-t, amellyel utólag visszakereshető a bejegyzés.
- Készítsünk egy formot, amellyel egy felhasználó képeket tölthet fel egy hirdetéshez, az egyedi ID-ját alkalmazva. Ha megadott ID-jú hirdetés nem létezik, adjunk megfelelő hibaüzenetet.
- Készítsünk egy formot, amellyel egy vendég kereshet lakáshirdetések között. Megadhat egy várost, negyedet, minimum árat és maximum árat. A visszatérített információ nem szükséges HTML/CSS-sel formázva legyen.

3.2. Asztalfoglaló vendéglőknek

- Készítsünk egy formot, amellyel egy tulajdonos bevezethet egy vendéglőt a rendszerbe. Egy vendéglő tartalmazza a címét (város, utca, szám külön), telefonszámot, nyitvatartást, stb. A hívás térítsen vissza egy generált egyedi ID-t, amellyel utólag visszakereshető a bejegyzés.
- Készítsünk egy formot, amellyel egy tulajdonos képeket tölthet fel egy vendéglőhöz, az egyedi ID-ját alkalmazva. Ha megadott ID-jú vendéglő nem létezik, adjunk megfelelő hibaüzenetet.
- Készítsünk egy formot, amellyel egy vendég asztalt foglalhat egy vendéglőnél, megadva a nevét, a vendéglő egyedi ID-ját és a foglalás időpontját. Kezeljük a helyzetet amikor nyitvatartáson kívül próbál valaki foglalni. Tároljuk a foglalásokat.

3.3. Online napló

- Készítsünk egy formot, amellyel egy tanár (még nem szükséges validálni a szerepkörét) létrehozhat új tantárgyakat. Egy tantárgyat körülír az egyedi kódja, leírása, feladatok listája (eleinte üres).
- Készítsünk egy formot, amellyel egy tanár feladatot hozhat létre egy már létező tantárgyhoz. Szükséges információk: a tantárgy egyedi azonosítója (ha még nem létezik ilyen tantárgy, jelezzünk hibát), a feladat rövid leírása, határideje, illetve egy részletes leírás PDF formátumú állomány formájában (csak PDF-eket engedjünk feltölteni).
- Készítsünk egy formot, amellyel egy tanár letörölhet egy tantárgyat az egyedi kódja alapján. Törlés esetén törlődjön a tantárgyhoz tartozó összes feladat is.

3.4. Használt autó kereskedés

- Készítsünk egy formot, amellyel felhasználók használt autókat tölthetnek fel a rendszerbe. Egy hirdetés tartalmazza az autó márkáját, városát, árát, feltöltés dátumát, stb. A hívás térítsen vissza egy generált egyedi ID-t, amellyel utólag visszakereshető a bejegyzés.
- Készítsünk egy formot, amellyel egy felhasználó képeket tölthet fel egy hirdetéshez, az egyedi ID-ját alkalmazva. Ha megadott ID-jú hirdetés nem létezik, adjunk megfelelő hibaüzenetet.
- Készítsünk egy formot, amellyel egy vendég kereshet hirdetések között. Megadhat egy márkát, várost, minimum árat és maximum árat. A visszatérített információ nem szükséges HTML/CSS-sel formázva legyen.

3.5. Online filmgyűjtemény

- Készítsünk egy formot, amellyel egy admin (még nem szükséges validálni a szerepkörét) filmet vezethet be a rendszerbe. Szükséges információk: cím, megjelenési év, leírás, zsáner, illetve egy borítókép. A hívás térítsen vissza egy generált egyedi ID-t, amellyel utólag visszakereshető a bejegyzés.
- Készítsünk egy formot, amellyel egy felhasználó egy filmnek visszajelzést (review) adhat, a film egyedi ID-ját alkalmazva. A visszajelzés áll egy pontszámból és egy szöveges véleményezésből. Ha megadott ID-jú film nem létezik, adjunk megfelelő hibaüzenetet.
- Készítsünk egy formot, amellyel egy felhasználó kereshet/szűrhet filmek között. Megadhat egy részleges címet, zsánert, minimum évszámot és maximum évszámot. A visszatérített információ nem szükséges HTML/CSS-sel formázva legyen.

3.6. Sportközpont pályafoglaló

- Készítsünk egy formot, amellyel egy admin (még nem szükséges validálni a szerepkörét) bevezethet egy sportpályát a rendszerbe. Egy bejegyzés tartalmazza a pálya típusát (kosárlabda, foci, stb.), órabérárát, címét, leírását. A hívás térítsen vissza egy generált egyedi ID-t, amellyel utólag visszakereshető a bejegyzés.
- Készítsünk egy formot, amellyel egy admin képeket tölthet fel egy sportpályához, az egyedi ID-ját alkalmazva. Ha megfelelő ID-jú pálya nem létezik, adjunk megfelelő hibaüzenetet.
- Készítsünk egy formot, amellyel egy felhasználó kereshet/szűrhet a sportpályák között. Választhat pályatípust legördülő listából, illetve megadhat minimum és maximum órabérárát. A visszatérített információ nem szükséges HTML/CSS-sel formázva legyen.

3.7. Órarendkészítő

- Készítsünk egy formot, amellyel adminok (még nem szükséges validálni a szerepkörét) felvihetnek új tantárgyakat. Egy tantárgyat körülír az egyedi kódja, neve, hogy melyik évfolyamnak szól, hány óra kurzus, szeminárium és labor jár vele. Ha egy egyedi kóddal már létezik tantárgy, küldjünk megfelelő hibát.
- Készítsünk egy formot, amellyel egy tanár erőforrás-állományokat (pl. PDF slide-ok) tölthet fel egy tantárgyhoz, az egyedi ID-ját alkalmazva (Vigyázat: több állomány tartozhat egy tantárgyhoz). Ha megfelelő ID-jú tantárgy nem létezik, adjunk megfelelő hibaüzenetet.
- Készítsünk egy formot, amellyel egy diák csatlakozhat vagy kiléphet a tantárgyhoz/-ból, megadva a tantárgy és a felhasználó ID-ját. Hogyha a megadott ID-jú tantárgy nem létezik, vagy a diák státusza nem megfelelő (próbál csatlakozni egy tárgyhoz, de már csatlakozott korábban, vagy próbál kilépni, de nem része), térítsünk vissza megfelelő hibaüzenetet.

3.8. Online könyvtár

- Készítsünk egy formot, amellyel adminok (még nem szükséges validálni a szerepkörét) bevezethetnek új könyveket a rendszerbe. Egy könyv egyedi azonosítója az ISBN száma, tartalmaz még címet, szerzőt, kiadási évet, rövid összefoglalót, borítóképet (állomány) és példányok számát.
- Készítsünk egy formot, amellyel egy felhasználó, megadva a nevét és egy könyv ISBN számát, kölcsönözhet/visszaszolgáltathat egy könyvet. Ha nincs több példány a könyvtárban kölcsönzéskor, vagy

próbál visszaadni egy könyvet, amit nem vett kölcsön, kapjon hibaüzenetet. Siker esetén a példányok számát frissítsük megfelelően.

- Készítsünk egy formot, amellyel egy felhasználó kereshet/szűrhet a könyvek között. Kereshet részleges cím vagy szerző alapján, illetve szűrhet minimum és maximum év alapján, továbbá megjelölheti, ha csak olyan könyveket szeretne látni, melyből legalább egy példány van készleten.

3.9. Rendezvényszervező

- Készítsünk egy formot, amellyel adminok (még nem szükséges validálni a szerepkörét) rendezvényeket vezethetnek be a rendszerbe. Egy rendezvényhez tartozik név, időintervallum, helyszín, szervező személyek listája (alapértelmezetten üres). A hívás térítsen vissza egy generált egyedi ID-t, amellyel utólag visszakereshető a bejegyzés.
- Készítsünk egy formot, amellyel egy szervező csatlakozhat egy rendezvényhez a neve és a rendezvény egyedi ID-ja alapján. Ugyanezen formon legyen lehetséges visszalépni a szervezési listából. Kezeljük megfelelően a hibákat (pl. ha csatlakozna de már csatlakozott korábban, stb.).
- Készítsünk egy formot, amellyel egy szervező egy rendezvényhez fényképeket tölthet fel. Ehhez meg kell adja a saját nevét és a rendezvény egyedi ID-ját. Ha feltöltéskor a felhasználó még nem szervező a bizonyos rendezvényen, jelezzünk megfelelő hibát.

3.10. Vonattársaság

- Készítsünk egy formot, amellyel adminok (még nem szükséges validálni a szerepkörét) vonatjáratokat vezethetnek be a rendszerbe. A járatok tartalmazzák a következőket: honnan, hova, a hét melyik napján, órájában, jegy ára, vonat típusa – gyors, regionális, stb. A hívás térítsen vissza egy generált egyedi ID-t, amellyel utólag visszakereshető a bejegyzés.
- Készítsünk egy formot, amellyel egy felhasználó helyet foglalhat egy bizonyos járaton, ennek egyedi ID-ját alkalmazva. Ha megadott ID-jú járat nem létezik, adjunk megfelelő hibaüzenetet.
- Készítsünk egy formot, amellyel egy vendég kereshet járatok között. Megadhat egy kiindulópontot, célpontot, minimum árat és maximum árat. A visszatérített információ nem szükséges HTML/CSS-sel formázva legyen.