## **Fejlesztői dokumentáció**

Ez a dokumentáció egy raktárkezelő és értékesítési rendszer backendjét mutatja be. A rendszer egy RESTful API-n keresztül érhető el, ami biztonságos és strukturált módon teszi lehetővé a felhasználók, partnerek, termékek, megrendelések és készletmozgások kezelését.

**Programismertető**

## **A rendszer célja**

Egy **belső raktárkezelő és értékesítési modul**, amely:

* nyilvántartja a termékeket és a készletmozgásokat,
* lehetővé teszi **megrendelések rögzítését** (partnerhez kapcsoltan),
* **számlát generál** a megrendelés alapján (PDF formátumban),
* partnereket kezel, és mindent naplóz.

## **Rendszermodulok**

### **1. Termékkatalógus**

* Kategóriák és alkategóriák szerint rendszerezett
* Minden termékhez elérhető:  
  + név, cikkszám, ár, elérhető mennyiség, mértékegység

### **2. Készletmozgás**

* Értékesítő rögzíthet:  
  + **bevét** (pl. új szállítmány érkezett)
  + **kivét** (megrendeléshez kapcsoltan)
  + **belső áthelyezés** (pl. belső felhasználás)

### **3. Partnerkezelés**

* Új partner felvitele egyszerű űrlapon (név, adószám, cím)
* Megrendelés/számla esetén **automatikus kitöltés, ha már létezik**

### **4. Megrendelés**

* Új rendelés partnerhez, termékekkel
* Státuszok: új, teljesítve, stb.
* Rögzített készletkivétellel együtt kezelve

### **5. Számlázás**

* Rendelés alapján **egy számla** hozható létre
* Automatikus számlasorszám (pl. SZ-2025-00123)
* PDF generálás a számla adataiból (tárolható + letölthető)

### **6. Naplózás**

* Minden adatváltozás automatikusan naplózódik:  
  + Ki, mikor, mit csinált
  + Admin számára visszakereshető

**Technológiák:**

* **Keretrendszer:** Express.js
* **Adatbázis ORM:** Sequelize
* **Adatbázis:** MySQL
* **Biztonság:** JWT (JSON Web Tokens) a hitelesítéshez és bcrypt a jelszavak titkosításához
* **Tesztelési keretrendszerek:** jest a teszt futtatására, supertest az Express.js végpontok tesztelésére.
* **Verziókezelés:** nodemon a fejlesztéshez

A backend architektúra moduláris, a funkcionalitások külön router fájlokra vannak osztva, ami könnyű karbantarthatóságot és bővíthetőséget biztosít. Az API végpontok védelme a authenticateJWT és authorizeRole middleware-eken keresztül valósul meg.

## **Futási Környezet Leírása**

A backend alkalmazás futtatásához szükséges minimális szerverinfrastruktúra és erőforrások leírása.

|  |  |  |
| --- | --- | --- |
| **Paraméter** | **Szükséges Érték / Leírás** | **Magyarázat** |
| Számítógép / Szerver | Bármely modern szerver infrastruktúra (fizikai szerver, virtuális gép, konténer, felhőszolgáltatás). | A futási környezet **nem asztali PC**, hanem egy szerver. |
| Operációs Rendszer | Linux (pl. Ubuntu/Debian), Windows Server, vagy bármely OS, amely támogatja a **Node.js** futtatását. | Javasolt a Linux stabilitása és teljesítménye miatt. |
| Memóriaméret | **Minimálisan 512 MB**, javasolt **2 GB** (a forgalomtól függően). | A Node.js alkalmazás futtatásához és az adatbázis (MySQL) processzeinek kezeléséhez szükséges. |
| Processzor (CPU) | Legalább 1 vCPU (Virtual CPU). | A Node.js egy szálon fut, de az I/O műveletek és az adatbázis párhuzamosítást igényelnek. |
| Speciális Perifériaigény | **Nincs.** | Mivel szerveroldali alkalmazás, nem igényel monitort, egeret vagy speciális hardvert. |
| Grafikus Kártya / Felbontás | **Nem releváns.** | A backend alkalmazás szöveges adatokat (JSON) dolgoz fel, grafikus felületet nem jelenít meg. |
| Adatbázis Szerver | **MySQL** vagy **MariaDB** (a választott ORM-nek megfelelően). | Dedikált adatbázis szerver szükséges a perzisztens adatok tárolásához. |

## **Installálási Vázlat**

**1. Backend:**

1. **Előkészületek:**

* **Node.js letöltése/ telepítése:**

1. Letöltés: <https://nodejs.org/en>
2. Telepítés: **npm init-y**

* **MySQL szerver létrehozása:**
  1. Szerver elindítása: **XAMPP -> Apache (Start) -> MySQL (Start) -> MySQL (Admin)**
  2. Adatbázis létrehozása: **a GitHub – Backend-ben található Project.sql fájl importálása a phpMyAdminba.**

1. **Kód Betöltése:**
   * Klónozza le a Backend repository-t.
2. **Függőségek Telepítése:**

* Navigáljon a gyökérkönyvtárba, és futtassa a parancsot: **npm install**

1. **Konfiguráció (.env fájl):**
   * Hozza létre a .env fájlt. Konfigurálja a kritikus változókat:

**PORT=3000**

**SECRET=asd**

1. **Adminisztrátor:**
   * [**admin@admin.hu**](mailto:admin@admin.hu)
   * **jelszó: admin**
2. **Éles Indítás:**
   * Indítsa el az alkalmazást: **npm run dev**

**2. Frontend (React):**

1. **Kód Betöltése:**
   * Klónozza le a Frontend repository-t.
2. **Függőségek Telepítése:**

* Navigáljon a gyökérkönyvtárba, és futtassa a parancsot: **npm install**

1. **Éles Indítás:**
   * Indítsa el az alkalmazást: **npm start**

Github elérhetőség:[**https://github.com/xLogiCorex**](https://github.com/xLogiCorex)

## **Hitelesítés (Authentication) és Jogosultság (Authorization)**

### **A. Felhasználói szerepkörök**

A rendszer két fő szerepkörrel dolgozik:

* **admin**: Teljes hozzáféréssel rendelkezik a rendszerhez, beleértve a felhasználók és partnerek kezelését, valamint a készlet- és megrendeléskezelés minden aspektusát.
* **sales**: Hozzáfér a megrendelések és termékek kezeléséhez, valamint saját készletmozgásaihoz, de korlátozott jogosultságai vannak (pl. nem kezelhet felhasználókat vagy partnereket).

### **B. API végpontok**

**Felhasználókezelés (users.js)**

Ez a router a felhasználók kezeléséért felelős.

|  |  |  |  |
| --- | --- | --- | --- |
| **Végpont** | **HTTP Metódus** | **Leírás** | **Szükséges jogosultság** |
| /login | POST | Bejelentkezés | admin, sales |
| /register | POST | Új felhasználó rögzítése | admin |
| /users | GET | Felhasználók listázása | admin |
| /users/:id/status | PUT | Felhasználó aktiválása/inaktiválása | admin |

* **POST /login**:
  + **Jogosultság**: Az admin és a sales is hozzáférhet.
  + **Hitelesítés**: A felhasználó email címével és jelszavával hitelesít, majd sikeres bejelentkezés esetén egy **JWT tokent** kap. A token tartalmazza az azonosítót, az email címet és a szerepkört (role), és 1 órára (1h) szóló érvényességgel rendelkezik.
  + **Biztonság**: A jelszó ellenőrzése **bcrypt.compare** használatával, hash formátum ellenében történik.
  + **Auditálás**: Minden **sikeres (LOGIN\_SUCCESS)** vagy **sikertelen (LOGIN\_FAIL)** bejelentkezési kísérlet naplózásra kerül (logAction), beleértve az IP-címet és a User-Agent-et is.
* **POST /register**:
  + **Jogosultság**: Szigorúan **admin** jogosultság lehetséges. Ez egy belső rendszer, nem nyilvános regisztrációs felület.
  + **Validáció**: Erőteljes bemeneti ellenőrzés történik a név, email és jelszó hosszúságára és érvényes formátumára (regex) vonatkozóan.
  + **Egyediség**: Ellenőrzi, hogy a megadott email cím vagy felhasználónév **nem létezik-e már** az adatbázisban (409 Conflict).
  + **Titkosítás**: A jelszót a mentés előtt **bcrypt** segítségével titkosítja (hash-eli) a biztonság érdekében.
  + **Naplózás**: A sikeres regisztráció **USER\_REGISTER** műveletként naplózásra kerül.
* **GET /users**:
  + **Jogosultság**: Csak **admin** férhet hozzá a végponthoz.
  + **Adatbiztonság**: Csak a szükséges adatokat (id, név, email, szerepkör, isActive) adja vissza, a jelszót kihagyja.
* **PUT /users/:id/status**:
  + **Jogosultság**: Csak **admin** jogosultság lehetséges.
  + **Funkció**: Lehetővé teszi egy felhasználó **aktiválását** vagy **inaktiválását** az isActive boolean érték módosításával. A művelet **USER\_STATUS\_CHANGE** néven naplózásra kerül.

## **Termék- és Készletkezelés**

### **A. Termékek (products.js)**

Ez a router a termékek kezeléséért felelős.

|  |  |  |  |
| --- | --- | --- | --- |
| **Végpont** | **HTTP Metódus** | **Leírás** | **Szükséges jogosultság** |
| /products | GET | Termékek listázása | admin, sales |
| /products | POST | Új termék létrehozása | admin |

* **GET /products**:
* **Jogosultság**: **admin** és **sales** felhasználók számára is engedélyezett.
* **Keresés (Filter)**: A végpont támogatja a **q** (query) lekérdezési paramétert. Ez lehetővé teszi a termékek keresését az **SKU** (cikkszám) vagy a **név** alapján.
* **Korlátozás**: A találatok maximális száma **100**-ra van korlátozva, elkerülve a nagyméretű adatbázis-lekérdezések miatti teljesítményproblémákat.
* **POST /products**:
* **Jogosultság:** Kizárólag **admin** jogosultság lehetséges.
* **Bemeneti Adatok**: Kötelezően tartalmaznia kell az SKU-t, a nevet és a termék alapvető adatait (pl. ár, egység, kategória azonosító).
* **Validáció**:
* Ellenőrzi, hogy **minden kötelező mező** ki van-e töltve.
* Ellenőrzi az **egyediséget**: megakadályozza a már létező **SKU** vagy **név** használatát, ezzel garantálva az adatok integritását.
* **Naplózás**: A sikeres terméklétrehozási eseményt a rendszer **naplózza** (logAction).

### **B. Készletmozgások (stockMovements.js)**

Ez a router a készletkövetést kezeli.

Minden készletmozgási művelet **adatbázis tranzakcióban** történik, ami garantálja, hogy a termék- és készletmozgás táblák adatai mindig konzisztensek maradnak. A tranzakció hibája esetén a rendszer visszagörget, és nem hajtja végre a műveletet.

|  |  |  |  |
| --- | --- | --- | --- |
| **Végpont** | **HTTP Metódus** | **Leírás** | **Szükséges jogosultság** |
| /stockMovements | GET | Készletmozgások listázása | admin, sales |
| /stockMovements/in | POST | Készlet bevételezés rögzítése | admin, sales |
| /stockMovements/out | POST | Készlet kivezetése rögzítése | admin, sales |
| /stockMovements/transfer | POST | Készlet áthelyezés rögzítése | admin, sales |

* **GET /stockMovements**:
* **Admin jogosultság**: Hozzáfér az **összes** készletmozgási rekordhoz, és szűrheti azokat mozgás **típus** (in, out, transfer), **dátumtartomány**, vagy **termékazonosító** (productId) alapján.
* **Sales jogosultság**: Biztonsági okokból csak azokat a mozgásokat láthatja, amelyeket **ő maga rögzített** (userId szűrés).
* **POST /stockMovements/in**:
* **Folyamat**:

1. Tranzakció indítása.
2. Bemeneti adatok validálása (pl. pozitív mennyiség, létező termék).
3. A productTable-ben frissül a termék **stockQuantity** és **availableStock** mezője.
4. Új bejegyzés rögzítése a stockMovementTable-ben.
5. Egyedi mozgásszám generálása: **KT-YYYY-00001** formátumban.
6. Tranzakció véglegesítése (COMMIT).
7. Sikeres válasz (201 Created).

* **POST /stockMovements/out**:
* **Készletellenőrzés**: Ez a kulcsfontosságú lépés a tranzakción belül történik. A rendszer megvizsgálja, hogy a termék **availableStock** (rendelkezésre álló készlet) mezője elegendő-e a kivezetni kívánt mennyiséghez. Ha nincs, a tranzakció **megszakításra kerül** (ROLLBACK), és a művelet visszautasításra kerül.
* **Folyamat**: Ha a készlet elegendő, a rendszer csökkenti a termék készletszintjét, rögzíti a mozgást, és véglegesíti a tranzakciót.
* **POST /stockMovements/transfer**:
* **Készletellenőrzés**: Hasonlóan a kivezetéshez, itt is ellenőrzés történik a kiinduló készlet rendelkezésre állására.
* **Különbség**: Az áthelyezés **nem befolyásolja** a termék teljes **stockQuantity** mezőjét (hiszen a készlet még mindig a cég tulajdonában van, csak máshol). A mozgás rögzítése elsősorban auditálási és belső logisztikai célokat szolgál.

## **Partnerek és Megrendelések**

### **A. Partnerek (partners.js)**

Ez a router a partnerekkel kapcsolatos végpontokat kezeli.

|  |  |  |  |
| --- | --- | --- | --- |
| **Végpont** | **HTTP Metódus** | **Leírás** | **Szükséges jogosultság** |
| /partners | GET | Partnerek listázása | admin, sales |
| /partners | POST | Új partner rögzítése | admin |

* **GET /partners**:
* **Hozzáférés**: Engedélyezett **admin** és **sales** szerepkörök számára is, mivel a partnerek listája elengedhetetlen a megrendelések felvételéhez.
* **Funkció**: Lekéri az összes rögzített partner adatait az adatbázisból.
* **POST /partners**:
* **Hozzáférés**: Kizárólag **admin** jogosultság lehetséges. A partneradatbázis tisztaságáért és integritásáért az adminisztrátorok felelnek.
* **Bemeneti Validáció**: A végpont rendkívül szigorú ellenőrzéseket hajt végre a bemeneti adatokon:
* **Hiányzó Mezők**: Ellenőrzi, hogy a kötelező adatok (pl. név, adószám, cím, email) ki vannak-e töltve.
* **Hosszúság**: Érvényesíti a nevek és egyéb szöveges mezők minimális és maximális hosszúságát.
* **Adószám Formátum**: Speciális reguláris kifejezésekkel ellenőrzi a **magyar adószám formátumának** helyességét (pl. 12345678-1-12), ezzel minimalizálva az adatrögzítési hibákat.
* **Email Formátum**: Ellenőrzi az email cím érvényességét.
* **Egyediség Ellenőrzése**: Megakadályozza, hogy **már létező adószámmal** partner kerüljön rögzítésre. Ha az adószám már szerepel a rendszerben, **409 Conflict** státuszkóddal tér vissza, jelezve a duplikációt. Az adószám a partner egyedi kulcsa a rendszerben.
* **Naplózás**: A sikeres új partner rögzítését a rendszer **naplózza** (logAction) auditálási célból.

### **B. Megrendelések (orders.js)**

Az orders.js router kezeli az ügyfélmegrendelésekkel kapcsolatos összes műveletet, és a rendszer üzleti logikájának központi része.

|  |  |  |  |
| --- | --- | --- | --- |
| **Végpont** | **HTTP Metódus** | **Leírás** | **Szükséges jogosultság** |
| /orders | GET | Megrendelések listázása | admin, sales |
| /orders/my | GET | Saját megrendelések | admin, sales |
| /orders/:id | GET | Egy adott megrendelés részletes adatainak lekérése | admin, sales |
| /orders | POST | Új megrendelés létrehozása | admin, sales |
| /orders/:id/status | PUT | Megrendelés státuszának módosítása | admin |

* **POST /orders**:
* **Hozzáférés**: **admin** és **sales** felhasználók is rögzíthetnek megrendeléseket.
* **Bemeneti Adatok**: Kötelezően tartalmaznia kell a **partnerId**-t, valamint az **items** tömböt, amelyben minden tételhez a **productId** és a rendelt **quantity** (mennyiség) szerepel.
* **Azonosító Generálás**: A rendszer automatikusan generál egy egyedi megrendelésszámot **ORD-YYYY-00001** formátumban.
* **Árszámítás**: A megrendelés rögzítésekor a rendszer kiszámítja a teljes **nettó árat** a megrendelt tételek és a termékek adatbázisban szereplő árai alapján.
* **Adatbázis Műveletek**: Új rekord jön létre az **orderTable**-ben és minden tételhez egy bejegyzés az **orderItemTable**-ben (rögzítve az eladási árat és a mennyiséget).
* **GET /orders, GET /orders/my**:
* **GET /orders**:
* **admin**: Minden megrendelést lekérdez.
* **sales**: Kizárólag a saját maga által rögzített megrendeléseket látja.
* **GET /orders/my**: Ez egy felhasználóbarát kényelmi végpont. Mindkét szerepkör (főleg a sales felhasználók) a saját megrendeléseit tudja lekérdezni, ami megerősíti a GET /orders jogosultsági szűrését.
* **GET /orders/:id**
* **Hozzáférés**: **admin** és **sales** felhasználók számára engedélyezett
* **Adatbázis Összekapcsolás**: A backend nem csak az alap megrendelés adatokat adja vissza, hanem kézi illesztéseket hajt végre:
* **Megrendelt Tételek**: Az orderItem táblából lekéri, hogy milyen termékek szerepelnek a megrendelésben (termék név, ár, mennyiség).
* **Partner Adatok**: Lekéri a kapcsolódó partner nevét és alapvető elérhetőségi adatait.
* **PUT /orders/:id/status**
* **Jogosultság**: Szigorúan **admin** szerepkör szükséges a megrendelések állapotának megváltoztatásához.
* **Státusz Validáció**: A végpont ellenőrzi, hogy a beérkező **newStatus** egy előre definiált, érvényes állapot-lista tagja-e (new, confirmed, processing, completed, cancelled, on\_hold).
* **Automatikus Számlázás:**
* Amikor a státusz **completed** értékre változik, a rendszer meghívja a belső **createInvoiceForOrder** függvényt.
* Ez a funkció ellenőrzi a duplikációt, kiszámítja a **nettó/bruttó/ÁFA** összegeket, generálja az egyedi számlaszámot (**SZ-YYYY-00001**) és rögzíti az új számlát az adatbázisban.
* Ez a szoros integráció biztosítja, hogy a teljesített megrendelések azonnal és automatikusan számlát kapjanak.

## **Kategóriák**

### **A. Alkategóriák (subcategories.js)**

Kezeli az alkategóriák létrehozását.

|  |  |  |  |
| --- | --- | --- | --- |
| **Végpont** | **HTTP Metódus** | **Leírás** | **Szükséges jogosultság** |
| /subcategories | GET | Alkategóriák listázása | admin, sales |
| /subcategories | POST | Alkategória létrehozása | admin |

* **GET /subcategories**:
  + **Hozzáférés**: **admin** és **sales** felhasználók számára is elérhető, a terméklistázás és szűrés támogatásához.
  + **Funkció**: Lekéri az összes rögzített alkategóriát.
* **POST /subcategories**:
  + **Hozzáférés**: Kizárólag **admin** jogosultság szükséges.
  + **Validáció**:
* Ellenőrzi a szükséges bemeneti mezőket (alkategória neve, kapcsolódó kategória azonosítója).
* A duplikáció ellenőrzése szigorúbb: **megakadályozza, hogy egyazon kategórián belül azonos nevű alkategória létezzen**. Például az "Okostelefon" alkategória nem hozható létre kétszer az "Elektronika" kategórián belül.
  + **Naplózás**: A művelet **naplózásra** kerül.

### **B. Kategóriák (categories.js)**

Kezeli a kategóriák létrehozását.

|  |  |  |  |
| --- | --- | --- | --- |
| **Végpont** | **HTTP Metódus** | **Leírás** | **Szükséges jogosultság** |
| /categories | GET | Kategóriák listázása | admin, sales |
| /categories | POST | Kategória létrehozása | admin |

* **GET /categories**:
  + **Hozzáférés**: Engedélyezett **admin** és **sales** felhasználók számára is. A kategória lista szükséges mind a termékkereséshez, mind a megrendelések felvételéhez.
  + **Funkció**: Lekéri az összes rögzített kategóriát az adatbázisból.
* **POST /categories**:
  + **Hozzáférés**: Kizárólag **admin** szerepkörrel végezhető művelet, biztosítva a terméktörzs integritását.
  + **Validáció**:
* Ellenőrzi, hogy a **newName** mező ne legyen üres.
* Megakadályozza a **duplikált kategórianevek** rögzítését, elkerülve az adatkonfliktust (409 Conflict).
  + **Naplózás**: A sikeres létrehozás **naplózásra** kerül (logAction).

## **Számlázás**

|  |  |  |  |
| --- | --- | --- | --- |
| **Végpont** | **HTTP Metódus** | **Leírás** | **Szükséges jogosultság** |
| /invoices | GET | Számlák listázása | admin, sales |
| /invoices/:id | GET | Lekérdezi egy adott számla részleteit | admin, sales |
| /invoices/:id/pdf | GET | Számla megnyitása PDF-ben, id alapján | admin, sales |
| /invoices | POST | Új számla létrehozása | admin, sales |

* **GET /invoices**:
* **admin jogosultság**: Hozzáfér a **teljes** számlalistához, ami elengedhetetlen a pénzügyi ellenőrzéshez és riportoláshoz.
* **sales jogosultság**: Csak a **saját maga által létrehozott** (hozzárendelt) számlákat láthatja. Ez a hozzáférés-korlátozás védi a bizalmas pénzügyi adatokat.
* **GET /invoices/:id**:
* **Hozzáférési Ellenőrzés**: Bár **admin** és **sales** felhasználók is hívhatják, a **sales** felhasználó esetében a rendszer ellenőrzi, hogy a számla az ő nevéhez kapcsolódik-e.
* **Jogosultság hiánya**: Ha egy **sales** felhasználó egy másik felhasználó által létrehozott számlát próbál lekérni, **403 Forbidden** jogosultsági hibát kap.
* **GET /invoices/:id/pdf**:
* **Technológia**: A végpont a **pdfkit** könyvtárat használja a PDF dinamikus, valós idejű generálásához.
* **Adatkomplexitás**: A generált PDF nem csak az alapadatokat tartalmazza, hanem strukturáltan jeleníti meg a **számlatételeket táblázatos formában**, valamint a teljes **partnerinformációt** is.
* **POST /invoices**:
* **Jogosultság**: **admin** vagy **sales** szerepkör szükséges.
* **Duplikáció Ellenőrzés**: A legfontosabb üzleti logika, hogy a funkció ellenőrzi, létezik-e már számla a kérésben megadott **orderId**-hez (megrendelés azonosítóhoz). Ezzel elkerüli a megrendelések duplikált számlázását.
* **Azonosító Generálás**: Egyedi számlaszámot generál: **SZ-YYYY-00001** formátumban.
* **Auditálás**: A sikeres számlalétrehozás **naplózásra** kerül (logAction).

## **Naplózás (Logging)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Végpont** | **HTTP Metódus** | **Leírás** | **Szükséges jogosultság** |
| /logs | GET | Napló megtekintése | admin |

* **GET /logs**:
* **Hozzáférés**: Kizárólag **admin** jogosultság lehetséges. Ez a korlátozás kulcsfontosságú, mivel a napló bizalmas információkat (pl. sikertelen bejelentkezési kísérletek, felhasználói IP-címek) tartalmaz, és csak a rendszergazdák számára engedélyezett a megtekintése.
* **Funkció**: Lekérdezi a **logTable** táblában rögzített összes bejegyzést.
* **Tartalom**: A naplóbejegyzések a következő adatokat tartalmazzák a logAction funkción keresztül rögzítve:
* **userId**: Ki hajtotta végre a műveletet.
* **actionType**: A művelet típusa (pl. LOGIN\_SUCCESS, ORDER\_CREATE, STOCK\_IN, USER\_REGISTER).
* **targetType**: A megcélzott entitás (pl. User, Order, Product).
* **payload (JSON)**: További részletek a műveletről (pl. a beállított új státusz, vagy a sikertelen bejelentkezési kísérlet oka és IP-címe).

### **Adatbázis séma (dbHandler.js)**

Az adatbázis jól strukturált és normalizált, így képes kezelni a különféle üzleti műveleteket. Definiálja a Sequelize modelleket, amelyek az adatbázis tábláit reprezentálják, és az összes adatbázis-interakció alapját képezik.

* **userTable**: Tárolja a felhasználók adatait (egyedi UUID id, név, email, hash-elt jelszó, szerepkör és aktív állapot).
* **productTable**: Kezeli a termékadatokat (egyedi sku, név, ár, készletinformáció). Kapcsolódik a categoryTable és subcategoryTable táblákhoz.
* **categoryTable és subcategoryTable**: Kategória hierarchiát definiál.
* **partnerTable**: Tárolja a partnerek/ügyfelek adatait (név, adószám, cím, kapcsolattartó).
* **stockMovementTable**: Rögzíti az összes készletváltozást (in, out, transfer).
* **orderTable**: Kezeli a megrendeléseket egyedi megrendelésszámmal, partnerhez és felhasználóhoz kapcsolódva.
* **orderItemTable**: Kapcsolótábla a termékek és megrendelések között.
* **invoiceTable**: Tárolja a számla adatait (egyedi invoiceNumber, nettó/bruttó érték, tételek JSON formátumban).
* **logTable**: Rögzíti a felhasználói tevékenységeket, ami alapvető fontosságú az auditáláshoz

**Tesztesetek**

1. **Kategória (categories.test.js)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Teszteset Azonosító** | **Bemeneti Adatok** | **Státuszkód** | **Magyarázat** |
| CAT-GET-001 | Nincs | 200 OK | Sikeres lekérdezés. |
| CAT-POST-002 | {} (Üres objektum) | 400 Bad Request | Kötelező mező hiánya. |
| CAT-POST-003 | Már létező kategória | 409 Conflict | Egyediségi korlátozás (Duplikált név). |
| CAT-POST-004 | Új Kategória | 201 Created | A categoryTable.create sikeresen lefutott. |
| CAT-POST-005 | Hibás Kategória | 500 Internal Server Error | Belső szerveroldali hiba. |

1. **Számla (invoice.test.js)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Teszteset Azonosító** | **Bemeneti Adatok** | **Státuszkód** | **Magyarázat** |
| INV-GET-001 | Sales | 200 OK | Sikeres lekérdezés, a biztonsági szűrés megtörtént. |
| INV-GET-002 | Admin | 200 OK | Az adminisztrátor teljes hozzáférése garantált. |
| INV-GET-003 | Nincs token | 401 Unauthorized | A token hiánya miatt nincs hozzáférés a védett végponthoz. |

1. **Megrendelés (orders.test.js)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Teszteset Azonosító** | **Bemeneti Adatok** | **Státuszkód** | **Magyarázat** |
| ORD-GET-001 | Sales | 200 OK | Sikeres lekérdezés, szerepkör szerinti szűrés. |
| ORD-GET-002 | Admin | 200 OK | Az adminisztrátor teljes hozzáférése garantált. |
| ORD-GET-003 | Nincs token | 401 Unauthorized /  403 Forbidden | Hitelesítés vagy jogosultság hiánya. |
| ORD-GET-004 | Lejárt JWT token | 401 Unauthorized | Érvénytelen, manipulált vagy lejárt token. |
| ORD-GET-005 | Hiányzó Authorization header | 401 Unauthorized | A fejléc hiányzik, a szerver nem tudja hitelesíteni a kérést. |
| ORD-GET-006 | Rossz formátumú Authorization header | 401 Unauthorized | A fejléc nem a várt Bearer [token] formátumot követi. |
| ORD-GET-007 | Hibás Bearer token | 401 Unauthorized | A token formátuma nem érvényes. |
| ORD-GET-008 | Érvénytelen JWT token | 401 Unauthorized | A token tartalma manipulált, vagy a titkosító kulcs (SECRET) nem egyezik. |
| ORD-POST-001 | Nincs token | 401 Unauthorized / 403 Forbidden | A megrendelés rögzítése védett végpont, hitelesítés nélkül sikertelen. |
| ORD-POST-002 | Hiányzó kötelező mező | 400 Bad Request | A bemeneti adatok validációja meghiúsul. |
| ORD-POST-003 | Duplikált orderszám | 409 Conflict | A generált/megadott megrendelésszám már létezik az adatbázisban. |
| ORD-POST-004 | Sales token, sikeres | 201 Created | Sikeres rögzítés. |
| ORD-POST-005 | Admin token, adatbázishiba | 500 Internal Server Error | A kérés feldolgozása közben belső hiba lép fel. |

1. **Partner (partner.test.js)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Teszteset Azonosító** | **Bemeneti Adatok** | **Státuszkód** | **Magyarázat** |
| **PART-GET-001** | Sales / Admin | 200 OK | A végpont **sales** és **admin** jogosultsággal is elérhető a megrendelés rögzítéshez. |
| **PART-POST-001** | Üres objektum {} | 400 Bad Request | Alapvető validációs hiba. |
| **PART-POST-002** | Túl rövid newName | 401 Unauthorized | Túl rövid név lett megadva. |
| **PART-POST-003** | Érvénytelen adószám | 400 Bad Request | A magyar adószám formátumra vonatkozó **regex** ellenőrzés meghiúsul. |
| **PART-POST-004** | Érvénytelen e-mail cím | 400 Bad Request | E-mail formátum validációs hiba. |
| **PART-POST-005** | Már létező adószám | 409 Conflict | Duplikált **adószám,** már szerepel az adatbázisban. |
| **PART-POST-006** | Minden mező érvényes | 201 Created | Sikeres létrehozás. |
| **PART-POST-007** | Minden mező érvényes, de DB hiba lép fel | 500 Internal Server Error | Belső szerverhiba. |

1. **Termék (products.test.js)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Teszteset Azonosító** | **Bemeneti Adatok** | **Státuszkód** | **Magyarázat** |
| **PROD-GET-001** | Nincs | 200 OK | Sikeres lekérdezés. |
| **PROD-POST-001** | Hiányzó kötelező mezők | 400 Bad Request | Alapvető validációs hiba. |
| **PROD-POST-002** | Túl rövid newSku | 400 Bad Request | Formátum / hosszúság validációs hiba. |
| **PROD-POST-003** | Túl rövid newName | 400 Bad Request | Hosszúság validációs hiba. |
| **PROD-POST-004** | Létező SKU | 409 Conflict | Egyediségi korlátozás hiba (SKU). |
| **PROD-POST-005** | Létező név | 409 Conflict | Egyediségi korlátozás hiba (Terméknév). |
| **PROD-POST-006** | Sikeres létrehozás | 201 Created | Sikeres létrehozás. |
| **PROD-POST-007** | DB hiba | 500 Internal Server Error | Belső szerverhiba. |

1. **Készletmozgás (stockMovement.test.js)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Teszteset Azonosító** | **Bemeneti Adatok** | **Státuszkód** | **Magyarázat** |
| SM-IN-001 | Hiányzó productId vagy quantity | 400 Bad Request | Alapvető validációs hiba. |
| SM-IN-002 | Negatív quantity | 400 Bad Request | Bevételezni csak pozitív mennyiséget lehet. |
| SM-IN-003 | Nem létező productId | 404 Not Found | A művelet nem hajtható végre nem létező terméken. |
| SM-IN-004 | Érvényes adatok | 201 Created | Sikeres létrehozás. |
| SM-OUT-001 | Túl nagy quantity | 400 Bad Request | A rendelkezésre álló készlet mínuszba nem mehet. |
| SM-OUT-002 | Érvényes adatok | 201 Created | Sikeres létrehozás |
| SM-TRANS-001 | Túl nagy quantity | 400 Bad Request | A forrásraktár készletének fedeznie kell a mennyiséget. |
| SM-TRANS-002 | Érvényes adatok | 201 Created | Sikeres létrehozás |

1. **Alkategória (subcategories.test.js)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Teszteset Azonosító** | **Bemeneti Adatok** | **Státuszkód** | **Magyarázat** |
| SUBCAT-GET-001 | Nincs | 200 OK | Sikeres lekérdezés. |
| SUBCAT-POST-001 | Hiányzó kötelező mezők | 400 Bad Request | Alapvető validációs hiba. |
| SUBCAT-POST-002 | Már létező | 409 Conflict | Egyediségi korlátozás (Duplikált). |
| SUBCAT-POST-003 | Érvényes adatok | 201 Created | Sikeres létrehozás. |
| SUBCAT-POST-004 | DB hiba | 500 Internal Server Error | Belső szerverhiba. |

1. **Felhasználó lekérés (usersGet.test.js)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Teszteset Azonosító** | **Bemeneti Adatok** | **Státuszkód** | **Magyarázat** |
| **USER-GET-001** | Nincs Token | 401 Unauthorized / 403 Forbidden | Hitelesítés vagy jogosultság hiánya. |
| **USER-GET-002** | Hibás Token | 401 Unauthorized / 403 Forbidden | Érvénytelen, hibás formátumú token. |
| **USER-GET-003** | Sales Token | 403 Forbidden | Csak az **admin** szerepkör férhet hozzá a listához. |
| **USER-GET-004** | Admin Token | 200 OK | Sikeres lekérdezés. |

1. **Felhasználó belépés (usersLogin.test.js)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Teszteset Azonosító** | **Bemeneti Adatok** | **Státuszkód** | **Magyarázat** |
| **AUTH-L-001** | Hiányzó kötelező mezők. | 400 Bad Request / 401 Unauthorized | Alapvető validációs hiba. |
| **AUTH-L-002** | Nem létező email | 401 Unauthorized | Hitelesítési hiba. |
| **AUTH-L-003** | Helyes email, helytelen password | 401 Unauthorized | Helytelen jelszó. |
| **AUTH-L-004** | Helyes email és password | 200 OK | Sikeres bejelentkezés. |
| **AUTH-L-005** | DB hiba | 500 Internal Server Error | Adatbázishiba. |
| **AUTH-L-006** | Tokenben lévő adatok | 200 OK | Megfelelő token adatok. |

1. **Felhasználó regisztrálás (userRegister.test.js)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Teszteset Azonosító** | **Bemeneti Adatok** | **Státuszkód** | **Magyarázat** |
| **USER-REG-001** | Nincs Token | 401 Unauthorized / 403 Forbidden | A kérés visszautasításra kerül, mert nincs token. |
| **USER-REG-002** | Hibás Token | 401 Unauthorized / 403 Forbidden | A kérés visszautasításra kerül az érvénytelen token miatt. |
| **USER-REG-003** | Sales Token | 403 Forbidden | Csak az **admin** szerepkör jogosult regisztrációra. |
| **USER-REG-004** | Hiányzó kötelező mezők | 400 Bad Request | Alapvető validációs hiba. |
| **USER-REG-005** | Túl rövid newName | 400 Bad Request | Név minimális karakterszámának megsértése. |
| **USER-REG-006** | Hibás newEmail formátum | 400 Bad Request | A megadott e-mail cím nem felel meg a formátumnak. |
| **USER-REG-007** | Már létező email | 409 Conflict | Az e-mail cím már használatban van. |
| **USER-REG-008** | Érvényes adatok | 201 Created | Sikeres létrehozás |
| **USER-REG-009** | DB hiba | 500 Internal Server Error | Adatbázishiba. |
| **USER-POST-001** | Nincs Token | 401 Unauthorized / 403 Forbidden | A kérés visszautasításra kerül, mert nincs token. |
| **USER-POST-002** | Sales Token | 403 Forbidden | Csak az **admin** jogosult új felhasználókat létrehozni. |
| **USER-POST-003** | Hiányzó kötelező mezők | 400 Bad Request | Alapvető validációs hiba. |
| **USER-POST-004** | Rossz role | 400 Bad Request | Ez a végpont kifejezetten a **sales** felhasználók rögzítésére szolgálhat. |
| **USER-POST-005** | Már létező name **vagy** email | 409 Conflict | A felhasználónév vagy az e-mail cím már használatban van. |
| **USER-POST-006** | Érvényes adatok, role: 'sales' | 201 Created | Sikeres létrehozás. |
| **USER-POST-007** | DB hiba | 500 Internal Server Error | Adatbázis műveleti hiba. |
|  |  |  |  |