Backend

**Konzulens:**

* Kaczur Sándor (it-tanfolyam.hu)

**Szükséges szoftverek:**

* xampp v3.3.0
* Composer v2.7.2
* Laravel v10

**Beüzemelés:**

A projekt clone-ozása után a xampp felületén importáljuk be a *ISABIKEDB1.0.sql* állományt

ez legenerálja nekünk az adatbázist a hozzá tartozó felhasználókat és a tárolt eljárásokat.

Ezt követően lépjünk be ebbe a könyvtárba *TMTK\_enjoyer\_vizsgaremek\Backend\api\isabikeAPI*,

és adjuk ki a következő parancsot *composer install*.

Miután ez a művelet le futót készítsünk másolatot a *.env.example* fájlból és nevezzük el *.env* -nek.

Ezután fútassuk le a projekt könyvtárában *TMTK\_enjoyer\_vizsgaremek\Backend\api\isabikeAPI* a

*php artisan key:generate* parancsot. Majd a server indításához adjuk ki a *php artisan serve* parancsot

ez localhost [*http://127.0.0.1:8000*](http://127.0.0.1:8000)fogja elindítani alapértelmezetten

**Táblák és kapcsolatok:**

*Táblák:*

* Felhasznalok: Felhasználók adatait tárolja.
* Viszaigazolo\_kod: A regisztrációhoz szükséges visszaigazoló kódokat tárolja.
* Tokenek: A bejelentkezéshez szükséges tokeneket tárolja.
* Ertekelesek: A termékek értékeléseit tárolja.
* Velemenyek: A termékekről írt véleményeket tárolja.
* Kedvezmenyek: A webáruházban elérhető kedvezményeket tárolja.
* Szalitok: A szállítókat tárolja.
* Fizetes\_opciok: A fizetési opciókat tárolja.
* Rendelesek: A rendelések adatait tárolja.
* Gyartok: A gyártókat tárolja.
* Kategoriak: A termékkategóriákat tárolja.
* Tomeg\_tulajdonsagai: A tömeg mértékegységeit tárolja.
* Termekek: A termékek adatait tárolja.
* Termek\_kepek: A termékekhez tartozó képeket tárolja.
* Ar\_valtozasok: A termékek árváltozásait tárolja.
* Kosarazot\_termekek: A kosárba tett termékeket tárolja.
* Rendelt\_termekek: A rendelt termékeket tárolja.

*Kapcsolatok:*

* A Velemenyek tábla felhasznalo\_id oszlopa a Felhasznalok tábla felhasznalo\_id oszlopára hivatkozik.
* A Velemenyek tábla ertekeles\_id oszlopa az Ertekelesek tábla ertekeles\_id oszlopára hivatkozik.
* A Rendelesek tábla kedvezmeny\_id oszlopa a Kedvezmenyek tábla kedvezmeny\_id oszlopára hivatkozik.
* A Rendelesek tábla szalito\_id oszlopa a Szalitok tábla szalito\_id oszlopára hivatkozik.
* A Rendelesek tábla fizetes\_opcio\_id oszlopa a Fizetes\_opciok tábla fizetes\_opcio\_id oszlopára hivatkozik.
* A Termekek tábla gyarto\_id oszlopa a Gyartok tábla gyarto\_id oszlopára hivatkozik.
* A Termekek tábla kategoria\_id oszlopa a Kategoriak tábla kategoria\_id oszlopára hivatkozik.
* A Termekek tábla tomeg\_tulajdonsaga\_id oszlopa a Tomeg\_tulajdonsagai tábla tomeg\_tulajdonsaga\_id oszlopára hivatkozik.
* A Termek\_kepek tábla termek\_id oszlopa a Termekek tábla termek\_id oszlopára hivatkozik.
* A Ar\_valtozasok tábla termek\_id oszlopa a Termekek tábla termek\_id oszlopára hivatkozik.
* A Kosarazot\_termekek tábla felhasznalo\_id oszlopa a Felhasznalok tábla felhasznalo\_id oszlopára hivatkozik.
* A Kosarazot\_termekek tábla termekek\_id oszlopa a Termekek tábla termek\_id oszlopára hivatkozik.
* A Rendelt\_termekek tábla rendeles\_id oszlopa a Rendelesek tábla rendeles\_id oszlopára hivatkozik.
* A Rendelt\_termekek tábla termek\_id oszlopa a Termekek tábla termek\_id oszlopára hivatkozik.

*Megjegyzések:*

* Az adatbázis UTF-8 karakterkód

**Tárolt eljárások:**

**get\_kategoriak\_procedure:**

*Leírás:*

A get\_kategoriak\_procedure tárolt eljárás lekéri az ÖSSZES rekordot a kategoriak táblából.

*Bemeneti paraméterek:*

A tárolt eljárásnak nincs bemeneti paramétere.

*Kimeneti paraméterek:*

A tárolt eljárás a kategoriak tábla minden rekordját visszaadja. A visszaadott rekordok a következő oszlopokat tartalmazzák:

* + kategoria\_id (integer): A kategória azonosítója
  + kategoria\_nev (string): A kategória neve

**check\_if\_felhasznalo\_exist\_procedure:**

*Leírás:*

A check\_if\_felhasznalo\_exist\_procedure tárolt eljárás ellenőrzi, hogy létezik-e a Felhasznalok táblában a megadott felhasznalo\_nev vagy email cím alapján.

*Bemeneti paraméterek:*

* felhasznalo\_nev\_p (varchar(20)): A felhasználónév
* email\_p (varchar(40)): Az email cím

*Kimeneti paraméterek:*

* result (integer): A Felhasznalok táblában található rekordok száma, amelyek megfelelnek a megadott feltételeknek.

**add\_felhasznalo\_procedure:**

*Leírás:*

Az add\_felhasznalo\_procedure tárolt eljárás új rekordot hoz létre a Felhasznalok táblában, és hozzárendel egy visszaigazoló kódot a Viszaigazolo\_kod táblában. A tárolt eljárás nem ad vissza kimeneti paramétert.

*Bemeneti paraméterek:*

* felhasznalo\_nev\_p (varchar(20)): A felhasználónév
* vezetek\_nev\_p (varchar(30)): A vezetéknév
* kereszt\_nev\_p (varchar(30)): A keresztnév
* email\_p (varchar(40)): Az email cím
* jelszo\_p (varchar(200)): A jelszó
* viszaigazolo\_kod\_p (varchar(5)): A visszaigazoló kód

Kimeneti paraméterek:

* felhasznalo\_id (integer): A Felhasznalok táblában újonnan létrehozott rekord azonosítója
* felhasznalo\_nev (string): A felhasználónév
* vezetek\_nev (string): A vezetéknév
* kereszt\_nev (string): A keresztnév
* vasarlo\_telefonszama (string): A vásárló telefonszáma (lehet null)
* email (string): Az email cím
* jelszo (string): A jelszó (titkosítva)
* szalitasi\_cime (string): A szállítási cím (lehet null)
* jogosultsag (integer): A jogosultsági szint
* regisztracio\_ideje (string): A regisztráció időpontja
* viszaigazolas\_statusz (integer): A visszaigazolási státusz
* aktiv (integer): Aktív-e a felhasználó
* viszaigazolo\_kod\_id (integer): A Viszaigazolo\_kod táblában azonosító
* viszaigazolo\_kod (string): A visszaigazoló kód

**verify\_felhasznalo\_procedure:**

*Leírás:*

A verify\_felhasznalo\_procedure tárolt eljárás ellenőrzi a viszaigazolo\_kod paraméterben megadott visszaigazoló kódot, és ha érvényes, akkor aktiválja a felhasználót a Felhasznalok táblában. A tárolt eljárás lekéri a Felhasznalok táblából az aktivált felhasználó adatait.

*Bemeneti paraméterek:*

* viszaigazolo\_kod\_p (varchar(5)): A visszaigazoló kód

*Kimeneti paraméterek:*

A tárolt eljárás a Felhasznalok táblából lekéri az aktivált felhasználó adatait.

* felhasznalo\_id (integer): A felhasználó azonosítója
* felhasznalo\_nev (string): A felhasználónév
* vezetek\_nev (string): A vezetéknév
* kereszt\_nev (string): A keresztnév
* vasarlo\_telefonszama (string): A vásárló telefonszáma (lehet null)
* email (string): Az email cím
* jelszo (string): A jelszó (titkosítva)
* szalitasi\_cime (string): A szállítási cím (lehet null)
* jogosultsag (integer): A jogosultsági szint
* regisztracio\_ideje (string): A regisztráció időpontja
* viszaigazolas\_statusz (integer): A visszaigazolási státusz
* aktiv (integer): Aktív-e a felhasználó
* viszaigazolo\_kod\_id (integer): A Viszaigazolo\_kod táblában azonosító
* viszaigazolo\_kod (string): A visszaigazoló kód

**logIn\_felhasznalo\_procedure:**

*Leírás:*

A logIn\_felhasznalo\_procedure tárolt eljárás ellenőrzi a beírt jelszót és email címet a Felhasznalok táblában. Ha a beírt adatok helyesek, és a felhasználó aktiválva van, a tárolt eljárás bejegyzést hoz létre a tokenek táblában, és aktiválja a felhasználót. A tárolt eljárás lekéri a Felhasznalok és a tokenek táblákból a bejelentkezett felhasználó adatait.

*Bemeneti paraméterek:*

* felhasznalo\_jelszo\_p (varchar(200)): A beírt jelszó
* email\_p (varchar(40)): A beírt email cím
* token\_p (varchar(120)): A token

*Kimeneti paraméterek:*

A tárolt eljárás vissza adja a Felhasznalok és a tokenek táblákból a bejelentkezett felhasználó adatait és egy „true” értéket result ként

* result (boolean): Az művelet sikeresége
* felhasznalo\_id (integer): A felhasználó azonosítója
* felhasznalo\_nev (string): A felhasználónév
* vezetek\_nev (string): A vezetéknév
* kereszt\_nev (string): A keresztnév
* vasarlo\_telefonszama (string): A vásárló telefonszáma (lehet null)
* email (string): Az email cím
* jelszo (string): A jelszó (titkosítva)
* szalitasi\_cime (string): A szállítási cím (lehet null)
* jogosultsag (integer): A jogosultsági szint
* regisztracio\_ideje (string): A regisztráció időpontja
* viszaigazolas\_statusz (integer): A visszaigazolási státusz
* aktiv (integer): Aktív-e a felhasználó
* token\_id (integer): A tokenek táblában azonosító
* token (string): A token

**set\_token\_procedure:**

*Leírás:*

A set\_token\_procedure tárolt eljárás bejegyzést hoz létre a tokenek táblában a megadott felhasznalo\_id\_p azonosítóhoz és token\_p tokenhez. Ezzel egyidejűleg aktiválja a felhasználót a Felhasznalok táblában.

*Bemeneti paraméterek:*

* felhasznalo\_id\_p (mediumint): A felhasználó azonosítója
* token\_p (varchar(120)): A token

*Kimeneti paraméterek:*

A tárolt eljárás nem ad vissza kimeneti paramétereket.

**logout\_procedure:**

*Leírás:*

A logout\_procedure tárolt eljárás inaktiválja a felhasználót a Felhasznalok táblában, és törli a bejegyzést a tokenek táblában a megadott token\_p token alapján.

*Bemeneti paraméterek:*

* token\_p (varchar(120)): A token

Kimeneti paraméterek:

A tárolt eljárás nem ad vissza kimeneti paramétereket.

**get\_felhasznalok\_procedure:**

*Leírás:*

Ez az eljárás lekéri a felhasználók adatait. A token alapján csak adminisztrátorok (jogosultság = 3) számára elérhető ez a procedure.

*Bemeneti paraméterek:*

* token\_p (varchar(120)): A felhasználó tokenje

*Kimeneti paraméterek:*

result (boolean): true és az összes felhasználó adata, ha a felhasználó adminisztrátor, false ha nem

* felhasznalo\_id (integer): A felhasználó azonosítója
* felhasznalo\_nev (string): A felhasználónév
* vezetek\_nev (string): A vezetéknév
* kereszt\_nev (string): A keresztnév
* vasarlo\_telefonszama (string): A vásárló telefonszáma (lehet null)
* email (string): Az email cím
* jelszo (string): A jelszó (titkosítva)
* szalitasi\_cime (string): A szállítási cím (lehet null)
* jogosultsag (integer): A jogosultsági szint
* regisztracio\_ideje (string): A regisztráció időpontja
* viszaigazolas\_statusz (integer): A visszaigazolási státusz
* aktiv (integer): Aktív-e a felhasználó

**get\_one\_felhasznalo\_procedure:**

*Leírás:*

Ez az eljárás lekéri egy felhasználó adatait a token alapján.

*Bemeneti paraméterek:*

token\_p (varchar(120)): A felhasználó tokenje

*Kimeneti paraméterek:*

result (boolean): true, ha a token érvényes, false ha nem

A felhasználó adatai (ha a token érvényes)

* felhasznalo\_id (integer): A felhasználó azonosítója
* felhasznalo\_nev (string): A felhasználónév
* vezetek\_nev (string): A vezetéknév
* kereszt\_nev (string): A keresztnév
* vasarlo\_telefonszama (string): A vásárló telefonszáma (lehet null)
* email (string): Az email cím
* jelszo (string): A jelszó (titkosítva)
* szalitasi\_cime (string): A szállítási cím (lehet null)
* jogosultsag (integer): A jogosultsági szint
* regisztracio\_ideje (string): A regisztráció időpontja
* viszaigazolas\_statusz (integer): A visszaigazolási státusz
* aktiv (integer): Aktív-e a felhasználó

**update\_felhasznalok\_procedure**

*Leírás:*

Ez az eljárás frissíti egy felhasználó adatait felhasználó ID alapján. Csak adminisztrátorok (jogosultság = 3) számára elérhető ezt a vizsgálatot token alapján végzi

*Bemeneti paraméterek:*

* token\_p (varchar(120)): A felhasználó tokenje
* felhasznalo\_id\_p (mediumint): A felhasználó ID-ja
* felhasznalo\_nev\_p (varchar(20)): A felhasználó neve
* vezetek\_nev\_p (varchar(30)): A vezetéknév
* kereszt\_nev\_p (varchar(30)): A keresztnév
* vasarlo\_telefonszama\_p (varchar(20)): A telefonszám
* email\_p (varchar(40)): Az email cím
* szalitasi\_cime\_p (varchar(40)): A szállítási cím
* jogosultsag\_p (tinyint): A jogosultsági szint (1 - felhasználó, 3 - adminisztrátor)

*Kimeneti paraméterek:*

result (boolean): true, ha a frissítés sikeres volt, false ha nem

A frissített felhasználó adatai (ha a frissítés sikeres volt)

* felhasznalo\_id (integer): A felhasználó azonosítója
* felhasznalo\_nev (string): A felhasználónév
* vezetek\_nev (string): A vezetéknév
* kereszt\_nev (string): A keresztnév
* vasarlo\_telefonszama (string): A vásárló telefonszáma (lehet null)
* email (string): Az email cím
* jelszo (string): A jelszó (titkosítva)
* szalitasi\_cime (string): A szállítási cím (lehet null)
* jogosultsag (integer): A jogosultsági szint
* regisztracio\_ideje (string): A regisztráció időpontja
* viszaigazolas\_statusz (integer): A visszaigazolási státusz
* aktiv (integer): Aktív-e a felhasználó

**update\_one\_felhasznalo\_procedure**

*Leírás:*

Ez az eljárás frissíti egy felhasználó adatait a token alapján.

*Bemeneti paraméterek:*

* token\_p (varchar(120)): A felhasználó tokenje
* felhasznalo\_nev\_p (varchar(20)): A felhasználó neve
* vezetek\_nev\_p (varchar(30)): A vezetéknév
* kereszt\_nev\_p (varchar(30)): A keresztnév
* vasarlo\_telefonszama\_p (varchar(20)): A telefonszám
* email\_p (varchar(40)): Az email cím
* szalitasi\_cime\_p (varchar(40)): A szállítási cím

*Kimeneti paraméterek:*

result (boolean): true, ha a frissítés sikeres volt, false ha nem

A frissített felhasználó adatai (ha a frissítés sikeres volt)

* felhasznalo\_id (integer): A felhasználó azonosítója
* felhasznalo\_nev (string): A felhasználónév
* vezetek\_nev (string): A vezetéknév
* kereszt\_nev (string): A keresztnév
* vasarlo\_telefonszama (string): A vásárló telefonszáma (lehet null)
* email (string): Az email cím
* jelszo (string): A jelszó (titkosítva)
* szalitasi\_cime (string): A szállítási cím (lehet null)
* jogosultsag (integer): A jogosultsági szint
* regisztracio\_ideje (string): A regisztráció időpontja
* viszaigazolas\_statusz (integer): A visszaigazolási státusz
* aktiv (integer): Aktív-e a felhasználó

**add\_termek\_procedure**

*Leírás:*

Ez az eljárás új terméket ad hozzá a Termekek táblához. Csak adminisztrátorok (jogosultság = 3) számára elérhető.

*Bemeneti paraméterek:*

* token\_p (varchar(120)): A felhasználó tokenje
* kategoria\_id\_p (mediumint): A termék kategóriájának ID-ja
* termek\_nev\_p (varchar(50)): A termék neve
* gyarto\_id\_p (tinyint): A gyártó ID-ja
* raktarondb\_p (bigint): A termék raktárkészlete
* tomeg\_tulajdonsaga\_id\_p (tinyint): A tömegegység ID-ja
* tomeg\_erteke\_p (double): A termék tömege
* szine\_p (varchar(50)): A termék színe
* leiras\_p (varchar(500)): A termék leírása
* egyseg\_ar\_p (mediumint): A termék ára

*Kimeneti paraméterek:*

result (boolean): true, ha a termék hozzáadása sikeres volt, false ha nem

A hozzáadott termék adatai (ha a termék hozzáadása sikeres volt)

* termek\_id (mediumint)
* kategoria\_id (mediumint)
* termek\_nev (varchar(50) )
* gyarto\_id (tinyint )
* raktarondb (bigint )
* tomeg\_tulajdonsaga\_id (tinyint)
* tomeg\_erteke (double)
* szine (varchar(50) )
* leiras (varchar(500) )
* egyseg\_ar (mediumint)
* elerheto (boolean )

**update\_termek\_procedure**

*Leírás:*

Ez az eljárás frissíti egy termék adatait a Termekek táblában. Csak adminisztrátorok (jogosultság = 3) számára elérhető.

*Bemeneti paraméterek:*

* token\_p (varchar(120)): A felhasználó tokenje
* termek\_id\_p (mediumint): A termék ID-ja
* kategoria\_id\_p (mediumint): A termék kategóriájának ID-ja
* termek\_nev\_p (varchar(50)): A termék neve
* gyarto\_id\_p (tinyint): A gyártó ID-ja
* raktarondb\_p (bigint): A termék raktárkészlete
* tomeg\_tulajdonsaga\_id\_p (tinyint): A tömegegység ID-ja
* tomeg\_erteke\_p (double): A termék tömege
* szine\_p (varchar(50)): A termék színe
* leiras\_p (varchar(500)): A termék leírása
* egyseg\_ar\_p (mediumint): A termék ára

*Kimeneti paraméterek:*

result (boolean): true, ha a frissítés sikeres volt, false ha nem

A frissített termék adatai (ha a frissítés sikeres volt)

* termek\_id (mediumint)
* kategoria\_id (mediumint)
* termek\_nev (varchar(50) )
* gyarto\_id (tinyint )
* raktarondb (bigint )
* tomeg\_tulajdonsaga\_id (tinyint)
* tomeg\_erteke (double)
* szine (varchar(50) )
* leiras (varchar(500) )
* egyseg\_ar (mediumint)
* elerheto (boolean )

**update\_termekek\_elerheto\_procedure**

*Leírás:*

Ez az eljárás frissíti egy termék elérhetőségét a Termekek táblában. Csak adminisztrátorok (jogosultság = 3) számára elérhető.

*Bemeneti paraméterek*

token\_p (varchar(120)): A felhasználó tokenje

termek\_id\_p (MEDIUMINT): A termék ID-ja

*Kimeneti paraméterek:*

result (boolean): true, ha a frissítés sikeres volt, false ha nem

**get\_termekek\_procedure**

*Leírás:*

* Az eljárás lekéri a Termekek tábla adatait.
* Az eljárás becsatlakozik a Kategoria táblához a kategoria\_id alapján.
* Az eljárás becsatlakozik a Gyartok táblához a gyarto\_id alapján.
* Az eljárás becsatlakozik a Tomeg\_tulajdonsagai táblához a tomeg\_tulajdonsaga\_id alapján.
* Az eljárás becsatlakozik a Termek\_kepek táblához a termek\_id alapján.
* Az eljárás csoportosítja az eredményeket a termek\_id alapján.
* Az eljárás kiszámítja a legkisebb kep\_helye értéket minden termékhez.
* Az eljárás limit\_p számú eredményt ad vissza.

Bemeneti paraméterek:

* limit\_p (integer): A lekérdezendő termékek maximális száma

Kimeneti paraméterek:

A Termekek tábla adatai, kiegészítve a kategória nevével, a gyártó nevével, a mértékegységgel és a termékfotók URL-jével.

* termek\_id (mediumint)
* kategoria\_id (mediumint)
* kategoria\_neve (varchar(50))
* termek\_nev (varchar(50) )
* gyarto\_id (tinyint )
* gyarto\_neve (varchar(50))
* telefonszama (varchar(50))
* webhely (varchar(50))
* raktarondb (bigint )
* tomeg\_tulajdonsaga\_id (tinyint)
* mertek\_egysege (varchar(50))
* tomeg\_erteke (double)
* szine (varchar(50) )
* leiras (varchar(500) )
* egyseg\_ar (mediumint)
* elerheto (boolean )
* kep\_helye (text)

**add\_gyartok\_procedure**

*Leírás:*

Ez az eljárás új gyártót ad hozzá a Gyartok táblához. Csak adminisztrátorok (jogosultság = 3) számára elérhető.

*Bemeneti paraméterek:*

* token\_p (varchar(120)): A felhasználó tokenje
* gyarto\_neve\_p (varchar(20)): A gyártó neve
* telefonszama\_p (varchar(25)): A gyártó telefonszáma
* webhely\_p (varchar(50)): A gyártó webhelye

*Kimeneti paraméterek:*

* result (boolean): true, ha a gyártó hozzáadása sikeres volt, false ha nem

**get\_gyartok\_procedure**

*Leírás:*

Ez az eljárás lekéri az összes gyártót a Gyartok táblából.

*Bemeneti paraméterek:*

Nincs

*Kimeneti paraméterek:*

* gyarto\_id (tinyint)
* gyarto\_neve (varchar(20))
* telefonszama (varchar(25))
* webhely (varchar(50))

**update\_gyartok\_procedure**

*Leírás:*

Ez az eljárás frissíti egy gyártó adatait a Gyartok táblában. Csak adminisztrátorok (jogosultság = 3) számára elérhető.

*Bemeneti paraméterek:*

* token\_p (varchar(120)): A felhasználó tokenje
* gyarto\_id\_p (tinyint): A gyártó ID-ja
* gyarto\_neve\_p (varchar(20)): A gyártó neve
* telefonszama\_p (varchar(25)): A gyártó telefonszáma
* webhely\_p (varchar(50)): A gyártó webhelye

*Kimeneti paraméterek:*

* result (boolean): true, ha a frissítés sikeres volt, false ha nem

**get\_ertekelesek\_procedure**

*Leírás:*

Ez az eljárás lekéri az összes értékelést az ertekelesek táblából.

*Bemeneti paraméterek:*

Nincs

*Kimeneti paraméterek:*

* ertekeles\_id (tinyint )
* ertekeles (varchar(20))

**get\_tomeg\_tulajdonsagok\_procedure**

*Leírás:*

Ez az eljárás lekéri az összes tömegtulajdonságot a tomeg\_tulajdonsagai táblából.

*Bemeneti paraméterek:*

Nincs

*Kimeneti paraméterek:*

A tomeg\_tulajdonsagai tábla összes adata

* tomeg\_tulajdonsaga\_id (tinyint)
* mertek\_egysege (varchar(20))

**add\_tomeg\_tulajdonsagok\_procedure**

*Leírás:*

Ez az eljárás új tömegtulajdonságot ad hozzá a tomeg\_tulajdonsagai táblához. Csak adminisztrátorok (jogosultság = 3) számára elérhető.

*Bemeneti paraméterek:*

Token\_p (varchar(100)): A felhasználó tokenje

mertek\_egysege (varchar(20)): A tömegtulajdonság mértékegysége

*Kimeneti paraméterek:*

result (boolean): true, ha a tömegtulajdonság hozzáadása sikeres volt, false ha nem

A hozzáadott tömegtulajdonság adatai (ha a hozzáadás sikeres volt)

* tomeg\_tulajdonsaga\_id (tinyint)
* mertek\_egysege (varchar(20))

**add\_velemenyek\_procedure**

*Leírás:*

Ez az eljárás új véleményt ad hozzá a velemenyek táblához.

*Bemeneti paraméterek:*

* Token\_p (varchar(100)): A felhasználó tokenje
* ertekeles\_id\_p (tinyint): Az értékelés ID-ja, amihez a vélemény kapcsolódik
* velemeny\_p (varchar(500)): A vélemény szövege

*Kimeneti paraméterek:*

result (boolean): true, ha a vélemény hozzáadása sikeres volt, false ha nem

**get\_velemenyek\_procedure**

*Leírás:*

Ez az eljárás lekéri az összes véleményt a velemenyek táblából.

*Bemeneti paraméterek:*

Nincs

*Kimeneti paraméterek:*

A velemenyek tábla összes adata

* velemeny\_id smallint
* felhasznalo\_id mediumint
* ertekeles\_id tinyint
* velemeny varchar(500)
* lathato boolean

**change\_velemenyek\_lathato\_procedure**

*Leírás:*

Ez az eljárás megváltoztatja egy vélemény láthatóságát a velemenyek táblában. Csak adminisztrátorok (jogosultság = 3) számára elérhető.

*Bemeneti paraméterek:*

* Token\_p (varchar(100)): A felhasználó tokenje
* velemeny\_id\_p (smallint): A vélemény ID-ja

*Kimeneti paraméterek:*

result (boolean): true, ha a láthatóság módosítása sikeres volt, false ha nem

**add\_termek\_kep\_procedure**

*Leírás:*

Ez az eljárás új termékfotót ad hozzá a termek\_kepek táblához. Csak adminisztrátorok (jogosultság = 3) számára elérhető.

*Bemeneti paraméterek:*

* Token\_p (varchar(100)): A felhasználó tokenje
* termek\_id\_p (mediumint): A termék ID-ja
* kep\_helye\_p (varchar(100)): A termékfotó fájlneve és elérési útja a szerveren

*Kimeneti paraméterek:*

result (boolean): true, ha a termékfotó hozzáadása sikeres volt, false ha nem

**get\_termek\_kepek\_procedure**

*Leírás:*

Ez az eljárás lekéri az összes termékfotót egy adott termékhez a termek\_kepek táblából.

*Bemeneti paraméterek:*

* termek\_id\_p (mediumint): A termék ID-ja

*Kimeneti paraméterek:*

A termek\_kepek tábla összes adata a megadott termékhez

* termek\_kep\_id mediumint
* termek\_id mediumint
* kep\_helye text

**delete\_termek\_kep\_procedure**

*Leírás:*

Ez az eljárás törli az összes termékfotót egy adott termékhez a termek\_kepek táblából. Csak adminisztrátorok (jogosultság = 3) számára elérhető.

*Bemeneti paraméterek:*

* Token\_p (varchar(100)): A felhasználó tokenje
* termek\_id\_p (mediumint): A termék ID-ja

*Kimeneti paraméterek:*

result (boolean): true, ha a termékfotók törlése sikeres volt, false ha nem

**add\_kosarazot\_termek\_procedure**

*Leírás:*

Ez az eljárás hozzáad egy terméket a kosárhoz a Kosarazot\_termekek táblában.

*Bemeneti paraméterek:*

* Token\_p (varchar(100)): A felhasználó tokenje
* termekek\_id\_p (mediumint): A termék ID-ja
* darabszam\_p (tinyint): A termék darabszáma

*Kimeneti paraméterek:*

* result (tinyint):
  + 0: Hibás token
  + 2: Nincs elegendő raktárkészlet
  + 3: Sikeres hozzáadás

**get\_one\_kosarazot\_termek\_procedure**

*Leírás:*

Ez az eljárás lekéri a kosárban lévő termékeket a Kosarazot\_termekek táblából egy adott felhasználó számára.

*Bemeneti paraméterek:*

* Token\_p (varchar(100)): A felhasználó tokenje

*Kimeneti paraméterek:*

result (boolean): true, ha a lekérés sikeres volt, false ha nem

A Kosarazot\_termekek tábla összes adata a bejelentkezett felhasználóhoz

* felhasznalo\_id mediumint
* termekek\_id mediumint
* darabszam tinyint
* kosarazas\_ideje DATETIME

**update\_kosarazot\_termeke\_darabszam\_procedure**

*Leírás:*

Ez az eljárás frissíti a kosárban lévő termék darabszámát a Kosarazot\_termekek táblában.

*Bemeneti paraméterek:*

* Token\_p (varchar(100)): A felhasználó tokenje
* termekek\_id\_p (mediumint): A termék ID-ja
* darabszam\_p (tinyint): A termék új darabszáma

*Kimeneti paraméterek:*

* result (tinyint):
  + 0: Hibás token
  + 1: Nincs elegendő raktárkészlet
  + 2: Sikeres frissítés

**delete\_kosarazot\_termekek\_procedure**

*Leírás:*

Ez az eljárás töröl egy terméket a kosárból a Kosarazot\_termekek táblában.

*Bemeneti paraméterek:*

* Token\_p (varchar(100)): A felhasználó tokenje
* termekek\_id\_p (mediumint): A termék ID-ja

*Kimeneti paraméterek:*

result (boolean): true, ha a termék törlése sikeres volt, false ha nem

**get\_one\_termek\_procedure**

*Leírás:*

eljárás lekér egy terméket a termekek táblából a termék ID-je alapján.

*Bemeneti paraméterek:*

* termek\_id\_p (mediumint): A termék ID-ja

*Kimeneti paraméterek:*

A termekek tábla összes adata a megadott termékhez, kiegészítve a kategória nevével, a gyártó nevével, a tömegegység mértékegységével, a termék elérhetőségével és a termékfotó URL-jével.

* termek\_id (mediumint)
* kategoria\_id (mediumint)
* kategoria\_neve (varchar(50))
* termek\_nev (varchar(50) )
* gyarto\_id (tinyint )
* gyarto\_neve (varchar(50))
* telefonszama (varchar(50))
* webhely (varchar(50))
* raktarondb (bigint )
* tomeg\_tulajdonsaga\_id (tinyint)
* mertek\_egysege (varchar(50))
* tomeg\_erteke (double)
* szine (varchar(50) )
* leiras (varchar(500) )
* egyseg\_ar (mediumint)
* elerheto (boolean )
* kep\_helye (text)

**get\_kategoriak\_procedure**

*Leírás:*

Ez az eljárás lekéri az összes kategóriát a kategoriak táblából.

*Bemeneti paraméterek:*

Nincs

*Kimeneti paraméterek:*

A kategoriak tábla összes adata

* kategoria\_id mediumint
* kategoria\_neve varchar(50)

**get\_one\_rendeles\_termekek\_procedure**

*Leírás:*

Ez az eljárás lekéri egy adott rendeléshez tartozó termékeket a rendelt\_termekek és a termekek táblákból.

*Bemeneti paraméterek:*

* rendeles\_id (mediumint): A rendelés ID-ja

*Kimeneti paraméterek:*

* termék\_nev (varchar(255)): A termék neve
* darabszam (tinyint): A termék darabszáma
* egyseg\_ar (decimal(10,2)): A termék egységára

**get\_rendelt\_termekek\_procedure**

*Leírás:*

Ez az eljárás lekéri az összes rendelt terméket a rendelt\_termekek táblából.

*Bemeneti paraméterek:*

* Token\_p (varchar(100)): A felhasználó tokenje

*Kimeneti paraméterek:*

result (boolean): true, ha a lekérés sikeres volt, false ha nem

A rendelt\_termekek tábla összes adata

* rendeles\_id mediumint
* termek\_id mediumint
* darabszam tinyint
* kezbesitve boolean
* kezbesites\_ideje DATETIME,

**get\_all\_rendeles\_procedure**

*Leírás:*

Ez az eljárás lekéri az összes rendelést a rendelesek táblából.

*Bemeneti paraméterek:*

* Token\_p (varchar(100)): A felhasználó tokenje

*Kimeneti paraméterek:*

result (boolean): true, ha a lekérés sikeres volt, false ha nem

A rendelesek tábla összes adata

* rendeles\_id mediumint
* felhasznalo\_id mediumint,
* rendeles\_ideje DATETIME default NOW(),
* megjegyzes varchar(200),
* szalito\_id tinyint,
* fizetes\_opcio\_id TINYINT,
* kedvezmeny\_id tinyint,

**get\_one\_rendeles\_procedure**

*Leírás:*

Ez az eljárás lekéri egy adott felhasználó egy rendelését a rendelesek táblából.

*Bemeneti paraméterek:*

* Token\_p (varchar(100)): A felhasználó tokenje

*Kimeneti paraméterek:*

result (boolean): true, ha a lekérés sikeres volt, false ha nem

A rendelesek tábla összes adata a megadott felhasználóhoz tartozó egyetlen rendeléshez

* rendeles\_id mediumint
* felhasznalo\_id mediumint,
* rendeles\_ideje DATETIME default NOW(),
* megjegyzes varchar(200),
* szalito\_id tinyint,
* fizetes\_opcio\_id TINYINT,
* kedvezmeny\_id tinyint,

**add\_rendeles\_procedure**

*Leírás:*

Ez az eljárás új rendelést hoz létre a rendelesek táblában, frissíti a felhasználó szállítási adatait a felhasznalok táblában, Át teszi a kosarazot\_termekek tábla felhasználóhoz kötőt tartalmát a rendelt termékek táblába és termekek táblában a termékek darabszámát csökkenti annyival amennyit rendeltek belőle

*Bemeneti paraméterek:*

* Token\_p (varchar(100)): A felhasználó tokenje
* vasarlo\_telefonszama\_p (varchar(20)): A vásárló telefonszáma
* szalitasi\_cime\_p (varchar(40)): A szállítási cím
* megjegyzes\_p (varchar(200)): Megjegyzés a rendeléshez
* szalito\_id\_p (tinyint): A szállító ID-je
* fizetes\_opcio\_id\_p (tinyint): A fizetési opció ID-je
* kedvezmeny\_id\_p (tinyint): A kedvezmény ID-je
* felhasznalo\_id (integer): A Felhasznalok táblában újonnan létrehozott rekord azonosítója
* felhasznalo\_nev (string): A felhasználónév
* vezetek\_nev (string): A vezetéknév
* kereszt\_nev (string): A keresztnév
* vasarlo\_telefonszama (string): A vásárló telefonszáma (lehet null)
* email (string): Az email cím
* jelszo (string): A jelszó (titkosítva)
* szalitasi\_cime (string): A szállítási cím (lehet null)
* jogosultsag (integer): A jogosultsági szint
* regisztracio\_ideje (string): A regisztráció időpontja
* viszaigazolas\_statusz (integer): A visszaigazolási státusz
* aktiv (integer): Aktív-e a felhasználó
* viszaigazolo\_kod\_id (integer): A Viszaigazolo\_kod táblában azonosító
* viszaigazolo\_kod (string): A visszaigazoló kód

*Kimeneti paraméterek:*

result (boolean): true, ha a lekérés sikeres volt, false ha nem

A frissen létrehozót rendelés adatait és a hozzá tartozó felhasználó adatait

* rendeles\_id mediumint
* felhasznalo\_id mediumint,
* rendeles\_ideje DATETIME default NOW(),
* megjegyzes varchar(200),
* szalito\_id tinyint,
* fizetes\_opcio\_id TINYINT,
* kedvezmeny\_id tinyint,

*Működési leírás:*

* Az eljárás lekéri a felhasználó ID-t a token alapján a Tokenek táblából.
* Ha a felhasználó nem létezik, az eljárás false értéket ad vissza.
* Ha a felhasználó létezik, az eljárás a következőket teszi:
* Új rendelést hoz létre a rendelesek táblában a megadott adatokkal.
* Lekéri a frissen létrehozott rendelés ID-jét.
* Beilleszti a kosárban lévő termékeket a rendelt\_termekek táblába a frissen létrehozott rendelés ID-jével.
* Frissíti a termék raktárkészletét a rendelt termékek mennyiségével.
* Frissíti a felhasználó szállítási adatait a felhasznalok táblában.
* Törli a kosár tartalmát a kosarazot\_termekek táblából.
* Az eljárás true értéket, a frissen létrehozott rendelés adatait és a felhasználó e-mail címét adja vissza.

**update\_rendelt\_termek**

*Leírás:*

Ez az eljárás megjelöli egy adott rendelt terméket kézbesítettként a rendelt\_termekek táblában.

*Bemeneti paraméterek:*

* Token\_p (varchar(100)): A felhasználó tokenje
* rendelt\_termek\_id\_p (mediumint): A rendelt termék ID-je

*Kimeneti paraméterek:*

* result (boolean): true, ha a frissítés sikeres volt, false ha nem

**Controllerek:**

**TermekContoller**

**A Termék Controller osztály a vizsga projektben a termékekkel kapcsolatos műveleteket kezeli. Felelős a termékek adatainak lekérdezéséért, létrehozásáért, módosításáért és törléséért.**

*getTermekek:*

*A getTermekek metódus /termekek/{limit} végpont meghivásával indul el az adatbázis Termekek tábla rekordjait adja vissza paraméternek egy limitet(int) vár ami azt határozza meg hogy hány darab terméket kérjen le a táblából a get\_termekek\_procedure(int) eljárás meghívásával*

*visszatérési adatok:*

* *termek\_id (int)*
* *kategoria\_id (int)*
* *kategoria\_neve (str)*
* *termek\_nev (str)*
* *gyarto\_id (int)*
* *gyarto\_neve (str)*
* *telefonszama (str)*
* *webhely (str)*
* *raktarondb (int)*
* *tomeg\_tulajdonsaga\_id (int)*
* *mertek\_egysege (str)*
* *tomeg\_erteke (float)*
* *szine (str)*
* *leiras (str)*
* *egyseg\_ar (int)*
* *elerheto (boolean)*
* *kep\_helye(str)*

*getOneTermekek:*

*A getOneTermekek metódus /onetermekek/{id} végpont meghivásával indul el az adatbázis Termekek tábla egy rekordjait adja termek\_id alapján paraméternek egy id(int) vár ami azt határozza meg hogy melyik terméket kérje le a táblából a get\_one\_termek\_procedure eljárás meghívásával*

*visszatérési adatok:*

* *termek\_id (int)*
* *kategoria\_id (int)*
* *kategoria\_neve (str)*
* *termek\_nev (str)*
* *gyarto\_id (int)*
* *gyarto\_neve (str)*
* *telefonszama (str)*
* *webhely (str)*
* *raktarondb (int)*
* *tomeg\_tulajdonsaga\_id (int)*
* *mertek\_egysege (str)*
* *tomeg\_erteke (float)*
* *szine (str)*
* *leiras (str)*
* *egyseg\_ar (int)*
* *elerheto (boolean)*
* *kep\_helye(str)*

*addTermek:*

*A addTermek metódus /* *addtermek végpont meghivásával indul el az adatbázis Termekek tábla egy uj rekordot adhozzá paraméternek egy Request adatait várj ami alapján létrehozza a terméket add\_termek\_procedure eljárás meghívásával.*

*Request adatok:*

* *token (str)*
* *termek\_kateg (int)*
* *termek\_nev (str)*
* *gyarto\_id (int)*
* *raktarondb (int)*
* *tomeg\_tulajdonsaga\_id (int)*
* *tomeg\_erteke*
* *szine (str)*
* *leiras (str)*
* *egyseg\_ar (int)*

*visszatérési adatok:*

* *token (str)*
* *termek\_kateg (int)*
* *termek\_nev (str)*
* *gyarto\_id (int)*
* *raktarondb (int)*
* *tomeg\_tulajdonsaga\_id (int)*
* *tomeg\_erteke*
* *szine (str)*
* *leiras (str)*
* *egyseg\_ar (int)*

*updateTermek:*

*A updateTermek metódus /* *updatetermek* *végpont meghivásával indul el az adatbázis Termekek tábla egy meglévö terméket változtat paraméternek egy Request adatait várj ami alapján megváltoztassa a terméket update\_termek\_procedure eljárás meghívásával.*

*Request adatok:*

* *token (str)*
* *termek\_kateg (int)*
* *termek\_nev (str)*
* *gyarto\_id (int)*
* *raktarondb (int)*
* *tomeg\_tulajdonsaga\_id (int)*
* *tomeg\_erteke*
* *szine (str)*
* *leiras (str)*
* *egyseg\_ar (int)*

*visszatérési adatok:*

* *token (str)*
* *termek\_kateg (int)*
* *termek\_nev (str)*
* *gyarto\_id (int)*
* *raktarondb (int)*
* *tomeg\_tulajdonsaga\_id (int)*
* *tomeg\_erteke*
* *szine (str)*
* *leiras (str)*
* *egyseg\_ar (int)*

*deleteTermek:*

*A deleteTermekmetódus /* *deletetermek végpont meghívásával indul el az adatbázis Termekek tábla egy meglevő termék állapotát változtatja hogy elérhető vagy nem paraméternek egy Request adatait várj ami alapján megváltoztassa a terméket update\_termekek\_elerheto\_procedure eljárás meghívásával.*

*Request adatok:*

* *BearerToken (str)*
* *termek\_id (int)*

*visszatérési adatok:*

* *result*

*getKategoriak:*

*A getKategoriak /* *kategoriak* *végpont meghívásával indul el az adatbázis Kategoriak tábla rekordjait adja vissza get\_kategoriak\_procedure* *eljárás meghívásával.*

*visszatérési adatok:*

* *kategoria\_id (int)*
* *kategoria\_neve (str)*

**FelhasznaloController**

*registerUser:*

*A registerUser metodus a /* *registration* *végpont meghívásával indul el az adatbázis Felhasznalok tábla hoz létre egy új felhasználót paraméternek egy Request-et vár amiben a felhasználó létrehozásához szükséges adatok vannak check\_if\_felhasznalo\_exist\_procedure és a* *add\_felhasznalo\_procedure* *eljárás meghívásával.*

*Request adatok:*

* *felhasznalo\_nev (str)*
* *kereszt\_nev (str)*
* *vezetek\_nev (str)*
* *email (str)*
* *password (str)*

*visszatérési adatok:*

* *felhasznalo\_id (int)*
* *felhasznalo\_nev (str)*
* *vezetek\_nev (str)*
* *kereszt\_nev (str)*
* *vasarlo\_telefonszama (str)*
* *email (str)*
* *jelszo (str)*
* *szalitasi\_cime (str)*
* *jogosultsag (int)*
* *regisztracio\_ideje*
* *viszaigazolas\_statusz*
* *aktiv*
* *viszaigazolo\_kod\_id (int)*
* *viszaigazolo\_kod (str)*

*verifyUser:*

*A verifyUser metodus /* *verify végpont meghívásával indul el az adatbázis felhasznalok tábla rekordjait változtassa meg hogy a felhasználó vissza igazolt státusza true értéket vegyen fel egy Request-et vár el paraméterként verify\_felhasznalo\_procedure eljárás meghívásával.*

*Request adatok:*

* *code (str)*

*visszatérési adatok:*

* *felhasznalo\_id (int)*
* *felhasznalo\_nev (str)*
* *vezetek\_nev (str)*
* *kereszt\_nev (str)*
* *vasarlo\_telefonszama (str)*
* *email (str)*
* *jelszo (str)*
* *szalitasi\_cime (str)*
* *jogosultsag (int)*
* *regisztracio\_ideje*
* *viszaigazolas\_statusz*
* *aktiv*
* *viszaigazolo\_kod\_id (int)*
* *viszaigazolo\_kod (str)*

*logIn*

*A logIn metodus /* *login végpont meghívásával indul el az adatbázis felhasznalok táblában változtassa meg hogy a felhasználó akti státusza true értéket vegyen fel ha helyes email jelszó párossal próbál meg bejelentkezni, sikeres bejelentkezéskor egy generált tokent add vissza Request-et vár el paraméterként logIn\_felhasznalo\_procedure eljárás meghívásával.*

*Request adatok:*

* *password (str)*
* *email (str)*

*visszatérési adatok:*

* *data (str)*

*logOut*

*A logOut metodus /* *logout* *végpont meghívásával indul el Request-et vár el paraméterként ahol ha helyes tokent adunk meg akkor token alapján kijelentkezteti a felhasználót és törli a toként logIn\_felhasznalo\_procedure eljárás meghívásával.*

*Request adatok:*

* *token (str)*

*visszatérési adatok:*

* *nincs*

*getFelhasznalok*

*A getFelhasznalok metodus /* *getfelhasznalok végpont meghívásával indul el Request-et vár el paraméterként ahol ha helyes admin tokent adunk meg akkor lekéri az összes felhasználó adatát a felhasznalok táblából get\_felhasznalok\_procedure eljárás meghívásával.*

*Request adatok:*

* *Bearer Token (str)*

*visszatérési adatok:*

* *felhasznalo\_id (int)*
* *felhasznalo\_nev (str)*
* *vezetek\_nev (str)*
* *kereszt\_nev (str)*
* *vasarlo\_telefonszama (str)*
* *email (str)*
* *jelszo (str)*
* *szalitasi\_cime (str)*
* *jogosultsag (int)*
* *regisztracio\_ideje*
* *viszaigazolas\_statusz*
* *aktiv*
* *viszaigazolo\_kod\_id (int)*
* *viszaigazolo\_kod (str)*

*updateFelhasznalo*

*A updateFelhasznalo metodus /* *updatefelhasznalo végpont meghívásával indul el Request-et vár el paraméterként ahol ha helyes admin tokent adunk meg akkor felhasználo id alapján frissíti egy felhasználó adatát a felhasznalok táblában update\_felhasznalok\_procedure eljárás meghívásával.*

*Request adatok:*

* *token*
* *felhasznalo\_id*
* *felhasznalo\_nev*
* *vezetek\_nev*
* *kereszt\_nev*
* *vasarlo\_telefonszama*
* *email*
* *szalitasi\_cime*
* *jogosultsag*

*visszatérési adatok:*

* *felhasznalo\_id (int)*
* *felhasznalo\_nev (str)*
* *vezetek\_nev (str)*
* *kereszt\_nev (str)*
* *vasarlo\_telefonszama (str)*
* *email (str)*
* *jelszo (str)*
* *szalitasi\_cime (str)*
* *jogosultsag (int)*
* *regisztracio\_ideje*
* *viszaigazolas\_statusz*
* *aktiv*
* *viszaigazolo\_kod\_id (int)*
* *viszaigazolo\_kod (str)*

*updateOneFelhasznalo*

*A updateOneFelhasznalo metodus /* *updateonefelhasznalo végpont meghívásával indul el Request-et vár el paraméterként ahol token alapján frissíti a felhasználó adatát a felhasznalok táblában update\_one\_felhasznalo\_procedure eljárás meghívásával.*

*Request adatok:*

* *token*
* *felhasznalo\_nev*
* *vezetek\_nev*
* *kereszt\_nev*
* *vasarlo\_telefonszama*
* *email*
* *szalitasi\_cime*
* *jogosultsag*

*visszatérési adatok:*

* *felhasznalo\_id (int)*
* *felhasznalo\_nev (str)*
* *vezetek\_nev (str)*
* *kereszt\_nev (str)*
* *vasarlo\_telefonszama (str)*
* *email (str)*
* *szalitasi\_cime (str)*

*getOneFelhasznalo*

*A getOneFelhasznalo metodus /* *getonefelhasznalo* *végpont meghívásával indul el Request-et vár el paraméterként ahol token alapján lekéri egy felhasználó adatát a felhasznalok táblábol get\_one\_felhasznalo\_procedure eljárás meghívásával.*

*Request adatok:*

* *token*

*visszatérési adatok:*

* *felhasznalo\_id (int)*
* *felhasznalo\_nev (str)*
* *vezetek\_nev (str)*
* *kereszt\_nev (str)*
* *vasarlo\_telefonszama (str)*
* *email (str)*
* *jelszo (str)*
* *szalitasi\_cime (str)*
* *jogosultsag (int)*
* *regisztracio\_ideje*
* *viszaigazolas\_statusz*
* *aktiv*
* *viszaigazolo\_kod\_id (int)*
* *viszaigazolo\_kod (str)*

*genToken*

A genToken metodus a Random osztály használatával egy 60 karakteres tokent generál amit tartalmazhat számokat kis és nagy betűt

*visszatérési adatok:*

* token (str)

*hashPass*

A hashPass metodus a hash\_hmac metódus segítségével le kódolja a felhasználó jelszavát

SHA256 kodlolást használ és „salt” ként a saláta szöveg van neki megadva

*bemeneti paraméter:*

* jelszó (str)

*visszatérési adatok:*

* kódolt jelszó (str)

*genEmailCode*

A genEmailCode metodus a Random osztály segítségével egy 5 karakter hosszú számot ad vissza

*visszatérési adatok:*

* kód (str)

**ErtekelesekController**

*getErtekelesek*

*A getErtekelesek metódus /* *ertekelesek végpont meghivásával indul el az adatbázis Kategoriak tábla rekordjait adja vissza get\_ertekelesek\_procedure eljárás meghívásával*

*visszatérési adatok:*

* ertekeles\_id (int)
* ertekeles (str)

**GyartokController**

*addGyarto*

*A addGyarto metódus /* *addgyarto végpont meghivásával indul el egy Request-et vár el paraméternek amiból egy új gyártót add hozzá a gyartok táblához, ha a token admin tokenje add\_gyartok\_procedure* *eljárás meghívásával*

*Request adatok:*

* *token (str)*
* *gyarto\_neve (str)*
* *telefonszama (str)*
* *webhely (str)*

*visszatérési adatok:*

* result

*getGyartok*

*A getGyartok metódus /* *gyartok végpont meghivásával indul el az adatbázis Gyartok tábla összes rekordját agya vissza get\_gyartok\_procedure eljárás meghívásával*

*visszatérési adatok:*

* *gyarto\_id (int)*
* *gyarto\_neve (str)*
* *telefonszama (str)*
* *webhely (str)*

*updateGyarto*

*A updateGyarto metódus /* *updategyarto végpont meghívásával indul el egy Request-et vár el paraméternek amiból egy gyártót változtat meg id alapján ha a token admin tokenje update\_gyartok\_procedure eljárás meghívásával*

*Request adatok:*

* *token (str)*
* *gyarto\_id (int)*
* *gyarto\_neve (str)*
* *telefonszama (str)*
* *webhely (str)*

*visszatérési adatok:*

* result

**KosarazotTermekekController**

*getOneKosar*

*getOneKosarOriginal*

*addToKosar*

*updateKosar*

*deleteKosar*

*deleteKosarOriginal*

**RendelesekController**

*addRendeles*

*getOneRendeles*

*getOneRendelesOriginal*

*getAllRendeles*

*getAllRendeltTermek*

*getOneRendelesTermekek*

*updateRendeltTermek*

**TermekKepController**

*addTermekKep*

*A addTermekKep metódus /* *addtermekkep végpont meghívásával indul el egy Request-et vár el paraméternek amiból egy új képet add hozzá a termek\_kepek táblához ha a token egy admin tokenje add\_termek\_kep\_procedure eljárás meghívásával*

*Request adatok:*

* *token (str)*
* *termek\_id (int)*
* *kep\_helye (str)*

*visszatérési adatok:*

* *result (int)*

*deleteTermekKep*

*A deleteTermekKep metódus /* *deletetermekkep végpont meghívásával indul el egy Request-et vár el paraméternek amiból egy képet töröl ki a termek\_kepek táblábol ha a token egy admin tokenje delete\_termek\_kep\_procedure eljárás meghívásával*

*Request adatok:*

* *token (str)*
* *termek\_id (int)*

*visszatérési adatok:*

* *result (int)*

**TomegTulajdonsagokController**

*getTomegTulajdonsagok*

*A getTomegTulajdonsagok metódus /* *tomegtulajdonsagok végpont meghívásával indul el*

*adatbázis Tomeg\_tulajdonsagai tábla összes rekordját agya vissza get\_tomeg\_tulajdonsagok\_procedure eljárás meghívásával*

*visszatérési adatok:*

* *tomeg\_tulajdonsaga\_id (int)*
* *mertek\_egysege (str)*

*addTomegTulajdonsagok*

*A addTomegTulajdonsagok metódus /* *addtomegtulajdonsagok végpont meghívásával indul el egy Request-et vár el paraméternek amiból egy új tömeg tulajdonságot add hozzá a Tomeg\_tulajdonsagai táblához add\_tomeg\_tulajdonsagok\_procedure eljárás meghívásával*

*Request adatok:*

* *token (str)*
* *mertek\_egysege (str)*

*visszatérési adatok:*

* tomeg\_tulajdonsaga\_id (int)
* *mertek\_egysege (str)*
* *result (int)*

**VelemenyekController**

*getVelemenyek*

*A getVelemenyek metódus /* *getvelemenyek végpont meghívásával indul el*

*adatbázis velemenyek tábla összes rekordját adja vissza get\_velemenyek\_procedure eljárás meghívásával*

*visszatérési adatok:*

* *velemeny\_id (int)*
* *felhasznalo\_id (int)*
* *ertekeles\_id (int)*
* *velemeny (str)*
* *lathato (int)*

*addVelemenyek*

*A addVelemenyek metódus /* *addvelemenyek végpont meghívásával indul el egy Request-et vár el paraméternek amiból egy új véleményt add hozzá a velemenyek táblához add\_velemenyek\_procedure eljárás meghívásával*

*Request adatok:*

* *token (str)*
* *ertekeles\_id (int)*
* *velemeny (int)*

*visszatérési adatok:*

* *result (int)*

*deleteVelemenyek*

*A deleteVelemenyek metódus /* *deletevelemenyek végpont meghívásával indul el egy Request-et vár el paraméternek amiból egy véleménynek a láthatósági státuszát változtatja a velemenyek táblába ha a token egy admin tokenje delete\_termek\_kep\_procedure eljárás meghívásával*

*Request adatok:*

* *velemeny\_id*

*visszatérési adatok:*

* *result (int)*

**Response controllerek**

A response controlerek felelösek a végpontok által hivot metodusok viszatérési értékének a formázására

* *FelhasznalokResponseController*
* *GyartokResponseController*
* *KosarazotTermekekResponseController*
* *RendelesekResponseController*
* *TermekekResponseController*
* *TermekKepResponseController*
* *TomegTulajdonsagokResponseController*
* *VelemenyekResponseController*