Vizsga Feladat

KeNorby

Fejlesztői dokumentáció

Készítette:

Baráth Máté

Vásárhelyi Ágoston

Fésű Gábor

A projekt indításának célja:

Iskolánkban ebben az évben az új vizsgarendszernek köszönhetően a másodéves diákoknak kötelező volt készíteni egy vizsgaremeket, mert ez a vizsga egyik fő része. Amely egy program megvalósításán alapul, mindezt 2 vagy 3 fős csapatokban kellett létrehozniuk.

Számunkra fontos volt, hogy olyan projektet készítsünk amit a későbbiekben, tovább fejlesztve akár értékesíteni is tudunk de ha más nem mindenképpen jól láthatóan szimulálja, hogy hogyan működik egy kerékpár webshop ami tanulás céljából is első osztályú.

A szoftver licence :

A program kizárólag a fejlesztők tulajdona, tovább értékesítés és módosítás nem engedélyezett. Megvásárlás esetén 5 db számítógépen futtatható maximum.

A szoftver kialakításának fő irányelvei :

1. A szoftver szabad felhasználású programokkal készül (Visual Studio Code, Mariadb-8.0.3 php, Netbeans 12.6)
2. A program Windows, Linux és MacOs rendszeren is futtatható
3. A program php-ban, java 16 verzióban és Angular,Laravel keretrendszerekben készült
4. Az adatbázis kezelő rendszer Mariadb
5. A program RUP fejlesztői módszer szerint készül, a fejlesztés interaktív és inkrementális
6. Alapvető funkciók :

Web :

- regisztráció (email cím, felhasználónév, jelszó)
- bejelentkezés (felhasználónév, jelszó)
- kijelentkezés
- új jelszó kérése email címre
- kategória szerinti szűrés (down-hill, bmx, országúti, stb…)
- méret szerinti szűrés (ember magassága)
- két érték közötti szűrés árban (fejlesztés alatt)
- hol található az adott termék (kerületek)
    - hirdetés neve alapján keresés
- rendezés ár szerint növekvő
- rendezés ár szerint csökkenő
- rendezés dátum szerint növekvő
- rendezés dátum szerint csökkenő
- rendezés népszerűség szerint
- üzenet küldése az eladónak
- üzenet fogadása a leendő vásárlótól
- hirdetés feladás
- hirdetés törlése
- saját hirdetések listája
- Beállítások :
    - email cím módosítás
    - felhasználónév módosítás
    - jelszó módosítás
    - telefonszám módosítás
    - cím módosítás (város, irányítószám, utca)

Asztali alkalmazás:

- bejelentkezés
- a webes felhasználók blokkolása/engedélyezése
- a webes felhasználók adatainak (név, email) módosítása
- keresés név szerint
- új admin felvétele

Mobil alkalmazás:

- bejelentkezés
- bicikli számláló
- sebesség mérő

Használat előtt:

1. Indítsa el a xampp alkalmazást.
2. Hozzon létre egy adatbázist a megadott néven és importálja.
3. Csatolja be a webes alkalmazás src mappáját a xampp/htdocs mappába.
4. Csatolja be a xampp/php mappába a php.init filet amit a Kiegészítő fájlokon belül talál.
5. Csatolja be a xampp/sendmail mappába a sendmail.init filet amit a Kiegészítő fájlokon belül talál.
6. Az asztali alkalmazásnál csatolja be a kettő kiegészítő file-t a használathoz és indítsa el az API szervert . (Minden információ a readme.txt-ben)
7. Ha megtette az összes lépést, akkor nyissa meg a böngészőt és írja be a fenti keresőrészbe: localhost/src.
8. A mobil alkalmazás indításhoz minden információt megtalál a mobil mappán belül található readme.txt -ben.

Stage RUP1

Követelmény feltárás:

Nem fukcionális követelmények :

1. A programnak futnia kell minden operációs rendszeren
2. Gépigény:
   Windows 7 vagy újabb verzió, Linux, MacOs
   4096 Mb RAM
   3 GHz processzor
   DirectX 11.
3. Háttérprogramok :
   Java futtatókörnyezet, mariadb adatbázis kezelő, xampp

A fejlesztés Netbeans IDE és Visual Studio Code használatával történik, Java 16 és mariadb 10.7.3 verziója.

Funkcionális követelmények :

1. A program használatához authentikáció szükséges
2. Az adatokat adatbázis tárolja
3. Az adatokat grafikusan kell megjeleníteni
4. Adatok módosítása
5. Adatok törlése
6. Új adatok felvétele

Megszortások :

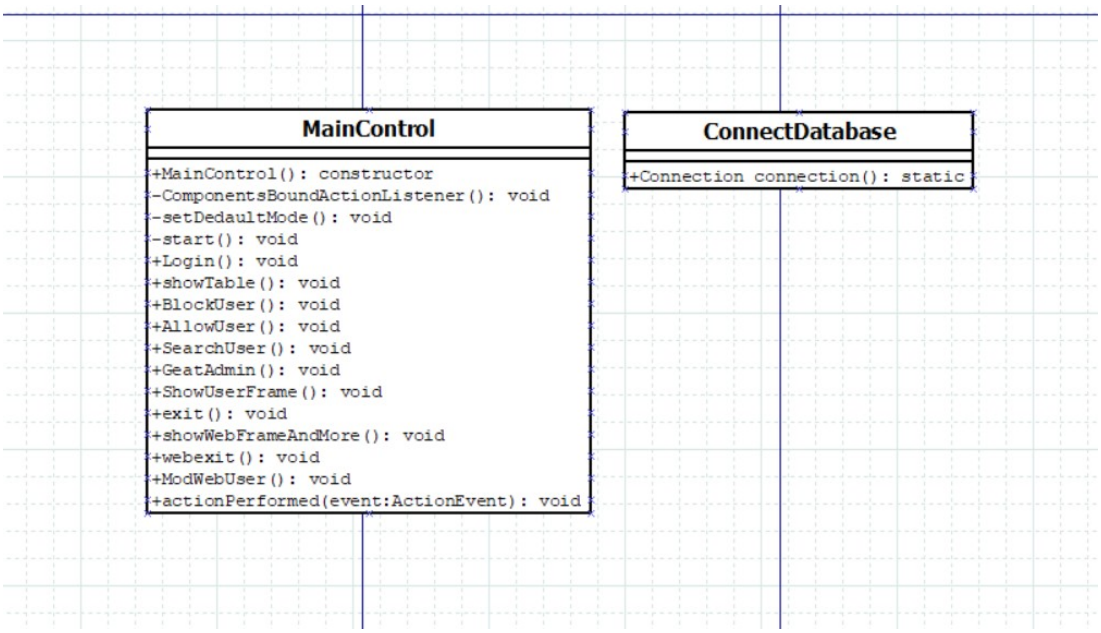
1. Az autentikációhoz szükséges adatokat adatbázisban kell tárolni(asztali alkalmazás)
2. A jelszó és felhasználói az asztali alkalmazáshoz:
   123456789

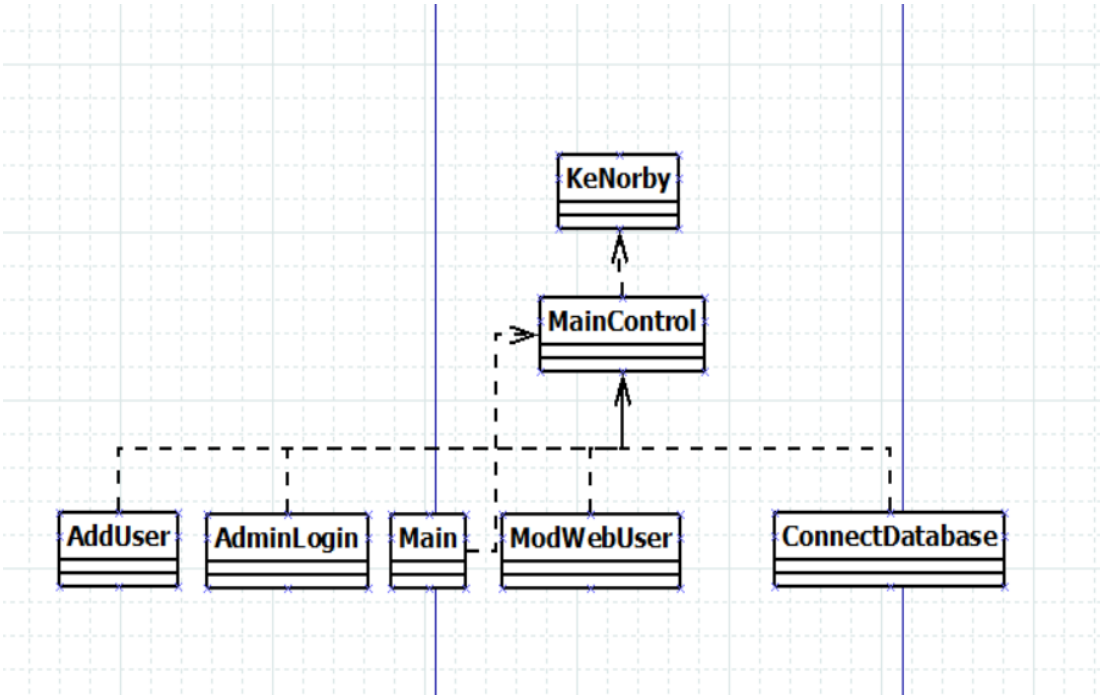
Stage RUP2

Tervezés :

UML tervek :

Asztali alkalmazás:

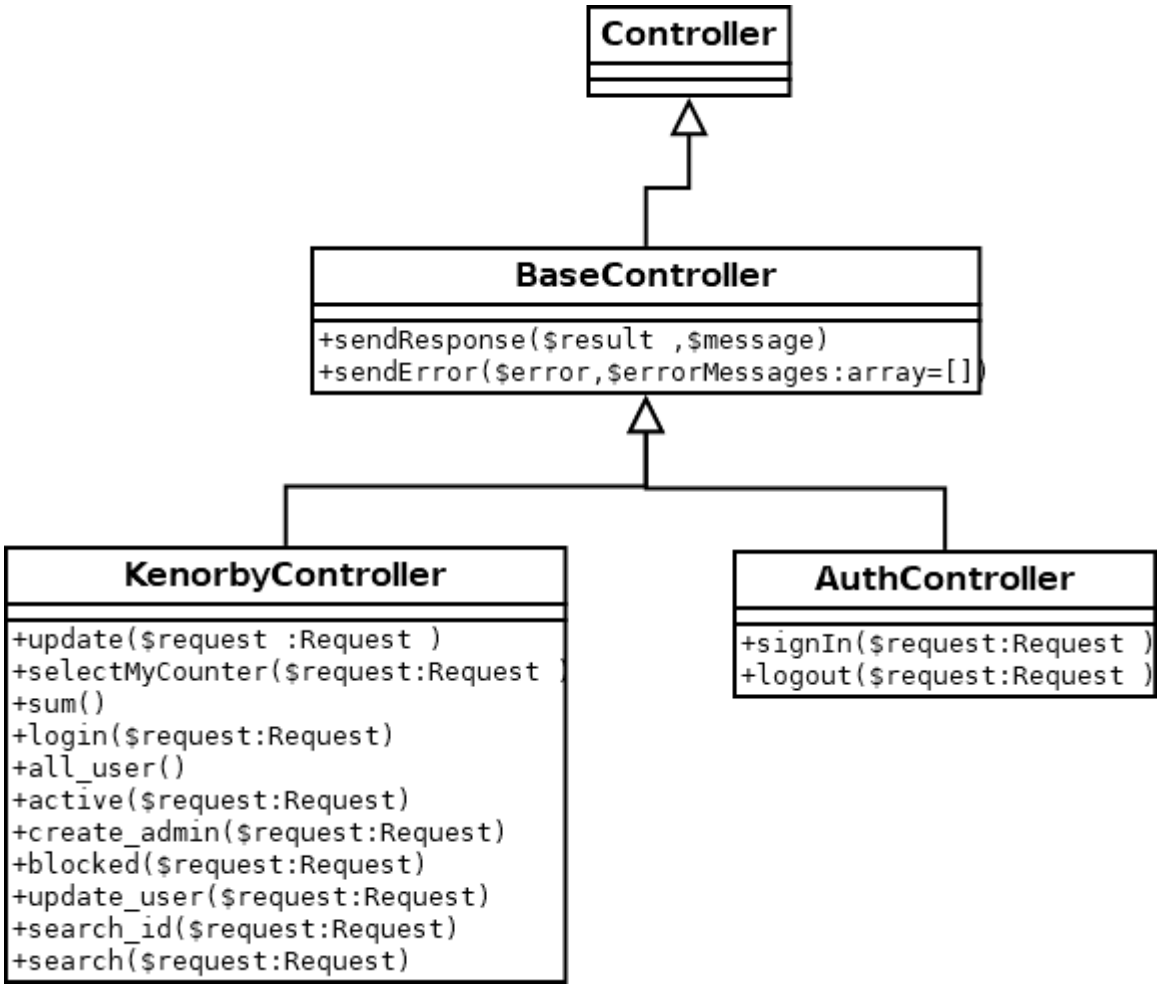




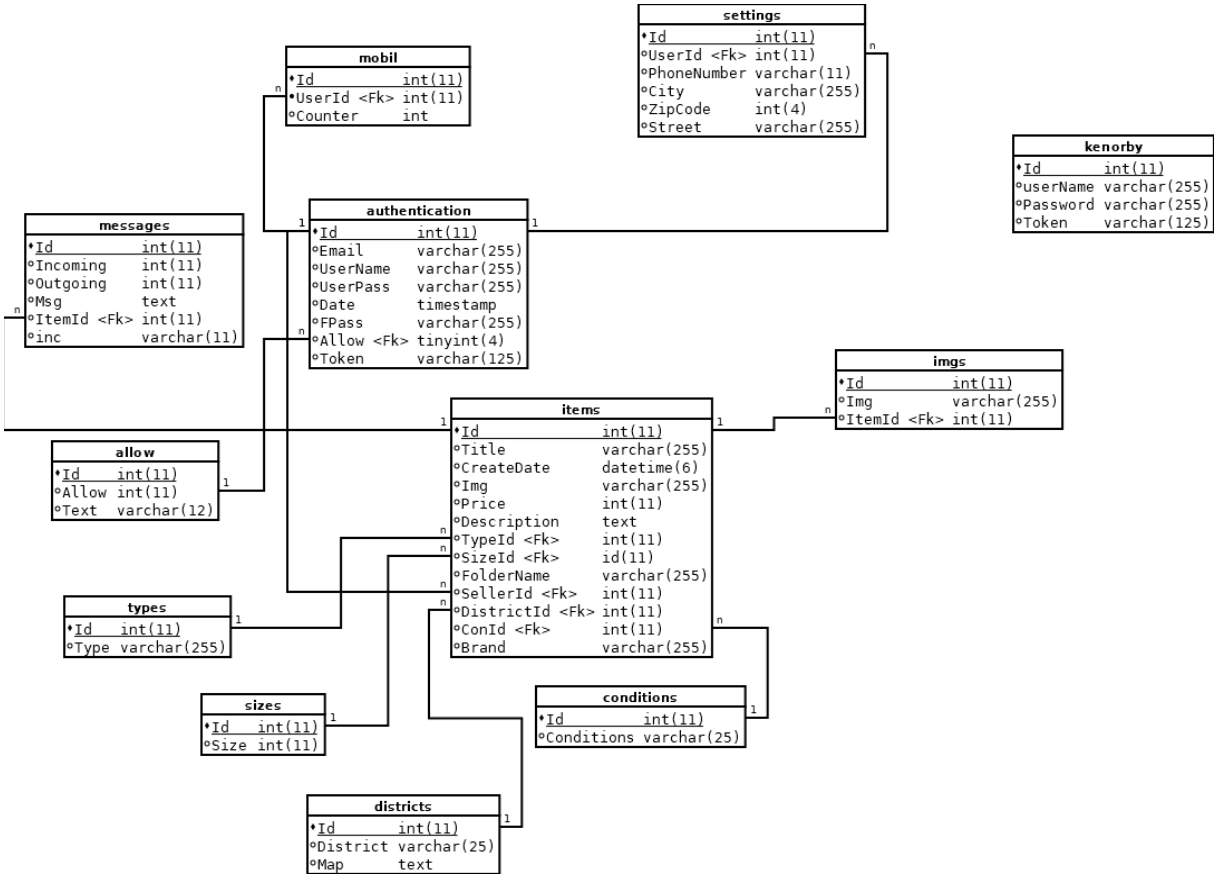
Web:

function
+EmptyInputsRegist($Username,$Password,$Email)
+EmptyInputsLogin($Username,$Password)
+EmptyInput($var)
+InvalidInputUsername($Username)
+InvalidInputPassword($Password)
+InvalidString($String)
+InvalidInt($Number)
+PhoneLength($Number)
+InvalidInputEmail($Email)
+NotLongEnough($Username,$Password)
+NotLongEnoughUsername($Username)
+ZipCodeLength($Code)
+MatchPass($Pass,$NewPassword)
+UserExistLogin($conn,$Username)
+UserExist($conn,$Username,$Email)
+UserExistEmail($conn,$Email)
+UserExistLoginUserName($conn,$Username)
+UserExistForgetPass($conn,$Email)
+createUser($conn, $Username,$Password,$Email)
+addSettings($conn,$Email)
+createSettings($conn,$row)
+loginDate($conn,$name)
+EmailUpdate($conn,$Id,$NewEmail)
+PhoneUpdate($conn,$Name,$Phone)
+CityUpdate($conn,$Name,$City)
+ZipCodeUpdate($conn,$Name,$ZipCode)
+NameUpdate($conn,$Id,$NewName)
+StreetUpdate($conn,$Name,$Street)
+PasswordUpdate($conn,$Username,$Password)
+loginUser($conn, $name, $pass)
+randomString($n)
+sendMail($conn,$Email)
+SendMailReal($Email,$pass,$username)
+SendEmail($pdo,$income,$outgoing,$msg,$id, $inc)
+SelectItem($pdo,$id)
+SelectImgs($pdo,$id)
+SelectDistricts($pdo)
+SelectSizes($pdo)
+SelectTypes($pdo)
+SelectMinPrice($pdo)
+SelectMaxPrice($pdo)
+SelectConditions($pdo)
+SelectWatchItem($pdo,$var,$sorted)
+Sorted($sorted)
+SelectMassage($pdo,$id,$var,$inc)
+SelectItemsMes($pdo,$var)
+NotSelect($var)
+insertIntoItems($pdo,$title,$Path,$price, $desc,$type,$size,$FolderPath, $var,$where,$condition,$brand)
+itemId($pdo)
+InsertImgs($pdo,$imagePaths,$var)
+selItem($pdo,$id)
+delItem($pdo,$id)
+delImg($pdo,$id)
+updateIntoItems($pdo,$title,$Path,$price, $desc,$type,$size,$FolderPath, $var,$where,$condition,$brand, $Id)
+CheckAllow($pdo,$user)

API:


Controller
BaseController
+sendResponse($result ,$message)
+sendError($error,$errorMessages:array=[])
KenorbyController
+update($request :Request)
+selectMyCounter($request:Request)
+sum()
+login($request:Request)
+all_user()
+active($request:Request)
+create_admin($request:Request)
+blocked($request:Request)
+update_user($request:Request)
+search_id($request:Request)
+search($request:Request)
AuthController
+signIn($request:Request)
+logout($request:Request)

Adatbázis :


settings
Id int(11)
UserId <Fk> int(11)
PhoneNumber varchar(11)
City varchar(255)
ZipCode int(4)
Street varchar(255)
mobil
Id int(11)
UserId <Fk> int(11)
Counter int
kenorby
Id int(11)
userName varchar(255)
Password varchar(255)
Token varchar(125)
messages
Id int(11)
Incoming int(11)
Outgoing int(11)
Msg text
ItemId <Fk> int(11)
inc varchar(11)
authentication
Id int(11)
Email varchar(255)
UserName varchar(255)
UserPass varchar(255)
Date timestamp
FPass varchar(255)
Allow <Fk> tinyint(4)
Token varchar(125)
imgs
Id int(11)
Img varchar(255)
ItemId <Fk> int(11)
items
Id int(11)
Title varchar(255)
CreateDate datetime(6)
Img varchar(255)
Price int(11)
Description text
TypeId <Fk> int(11)
SizeId <Fk> id(11)
FolderName varchar(255)
SellerId <Fk> int(11)
DistrictId <Fk> int(11)
ConId <Fk> int(11)
Brand varchar(255)
allow
Id int(11)
Allow int(11)
Text varchar(12)
types
Id int(11)
Type varchar(255)
sizes
Id int(11)
Size int(11)
conditions
Id int(11)
Conditions varchar(25)
districts
Id int(11)
District varchar(25)
Map text

kenorby:

1. userName: felhasználónév
2. Password: jelszó
3. Token: azonosító

items:

1. Title: cím
2. CreateDate: létrehozási dátum
3. Img: elérési útvonal, kép neve és kiterjesztése
4. Price: ár
5. Description: leírás
6. TypeId: külső kulcs
7. SizeId: külső kulcs
8. FolderName: A mappa neve
9. SellerId: külső kulcs
10. DistrictId: külső kulcs
11. ConId: külső kulcs
12. Brand: márka

imgs:

1. Img: elérési útvonal és kiterjesztés
2. ItemId: külső kulcs

conditions:

1. Conditions: állapot

districts:

1. District: kerület
2. Map: térkép

sizes:

1. Size: méret

types:

1. Type: típus

allow:

1. Allow: Állapot
2. Text: engedélyez, blokkol

messages:

1. Incoming: Kitül
2. Outgoing: Kinek
3. Msg Üzenet
4. ItemId: külső kulcs
5. inc: Egyedi azonosító

authentication:

1. Email: email

2. UserName: felhasználó
3. UserPass: jelszó
4. Date: dátum
5. FPass : Random generált jelszó
6. Allow: külső kulcs
7. Token: azonosító

settings:

1. UserId: külső kulcs
2. PhoneNumber: telefonszám
3. City: város
4. ZipCode: irányítószám
5. Street: utca

mobil:

1. Mid: külső kulcs
2. UserId: felhasználó azonósító
3. Counter: számláló

Stage RUP3

(asztali alkalmazás)

Implementáció:

MainControl:

Mezők:

private ALoginfrm (Példányosított AdminLogin objektum)
private Mainfrm (Példányosított Main objektum)
private AddUser (Példányosított Userfrm objektum)
private ModWebUser (Példányosított Webfrm objektum)
Public String Id változó

Konstructor:

nincs bejövő paraméter: ConnectDatabase conn tárolva a conn mezőben
setDefaultMode() metódus hívása;
ComponentsBoundActionListener() metódus hívása;

ComponentsBoundActionListener():

Nincs bejövő paraméter.
A formokon elhelyezett komponensek kötése az ActionListener interfészhez.
A komponensekhez getter setter metódusokkal lehet hozzáférni.

SetDefaultMode():

Nincs bejövő paraméter.
Példányosítja az AdminLogin osztályt és láthatóvá teszi.
Példányosítja a Mainfrm osztályt és nem láthatóvá teszi.

Példányosítja a AddUser osztályt és tárolja az Userfrm mezőben.
Példányosítja a ModWebUser osztályt és tárolja az Webfrm mezőben.

start():

Nincs bejövő paraméter.
Láthatóvá teszi a példányosított Mainfrm osztályt.
AdminLogin-t dispose módba teszi.
Meghívja a ShowTable() metódust.

Login():

Nincs bejövő paraméter.
User, Pass Stringekben tárolja a jTextfieldekből kiolvasott adatokat. SessionId-ban eltárolja a randomGenarator értékét.
Kapcsolódik az api szerverhez elküldi a szükséges adatokat.
A vissza kapott responseCode alapján végez egy feltététel vizsgálgatott ha a code 200-as akkor a start() metódus hívja.
Eltérő adatok esetén („Hibás Adatok ”) üzenet jelenik meg AdminLogin erre a célra kiválasztott jlabelén.
A folyamat során hiba történik akkor egy újabbakban hibaüzenetet jelenit meg.

ShowTable():

Nincs bejövő paraméter.
Kapcsolódik az adatbázisra, sql utasítást lefuttatja és az adatokat feltölti Mainfrm.getTbl() táblázatba. A folyamat során hiba történik akkor egy újablakban hibaüzenetet jelenit meg.

BlockUser():

Nincs bejövő paraméter.

Létrehoz egy int típusú row változót aminek az értéke az éppen kiválasztott sor indexének értéke.
Majd Stringként tárolja a Mainfrm.getTbl() 2. oszlop érteket.
Kapcsolódik az api szerverhez elküldi a szükséges adatokat.
A vissza kapott responseCode alapján végez egy feltététel vizsgálgatott ha a code 200-as akkor ,kiolvassa a beviteli mező tartalmát ha üres akkor meghívja a ShowTable() metódust,.
Ellenkező esetben meghívja a SearchUser() metódust, ha a folyamat közeben hiba történik hibaüzenetet dob "Nincs kiválaszott sor!" Ha a szerverrel akadnak gondok „http válasz” és kódot kapunk.

AllowUser():

Majd Stringként tárolja a Mainfrm.getTbl() 2. oszlop érteket.
Kapcsolódik az api szerverhez elküldi a szükséges adatokat.
A vissza kapott responseCode alapján végez egy feltététel vizsgálgatott ha a code 200-as akkor ,kiolvassa a beviteli mező tartalmát ha üres akkor meghívja a ShowTable() metódust,.
Ellenkező esetben meghívja a SearchUser() metódust, ha a folyamat közeben hiba történik hibaüzenetet dob "Nincs kiválaszott sor!" Ha a szerverrel akadnak gondok „http válasz” és kódot kapunk.

SearchUser():

Nincs bejövő paraméter.
Felveszi az adatot a mezőből, feltétel vizsgálattal ellenzői hogy a mezőben van –e adat, ha nincs beírva semmi akkor lekéri az összes adatot ,ha van akkor csatlakozik az adatbázishoz és lekérni annak a adatait, ha a folyat során hiba történik akkor hibaüzenetet dob.

GetAdmin():

Nincs bejövő paraméter.
Felveszi a mező tartalmát, megvizsgálja a karakter számokat ,ha nem érik el a 9 karakter nem folytatja a folyamatot. Abban az esetben ha a feltétel teljesül Kapcsolódik az api szerverhez elküldi a szükséges adatokat.
A vissza kapott responseCode alapján végez egy feltététel vizsgálgatott, ha a code 200-as akkor sikeres volt a művelet meghívja az exit() metódust.

ShowUserFrame():

Nincs bejövő paraméter.
Láthatóvá teszi a Userfrm-et.

exit():

Nincs bejövő paraméter.
Láthatatlanná teszi a Userfrm-et és kiüríti a mezőket.

showWebFrameAndMore():

Nincs bejövő paraméter.
A kiválasztott mezőből kiolvassa az adatokat, eltárolja a változókban majd egy sql utasítást futtat le, hogy megtudja azokhoz az adatokhoz tartozó id-t, majd a megfelelő mezőkbe beállítja az adatokat, ha a folyamat során elakad hibát dob egy ablakban.

webexit():

Nincs bejövő paraméter.

Láthatatlanná teszi a Webfrm-et.

ModWebUser():

Nincs bejövő paraméter.
A mezők adatait eltárolja változókban, Kapcsolódik az api szerverhez elküldi a szükséges adatokat.
A vissza kapott responseCode alapján végez egy feltététel vizsgálgatott ha a code 200-as akkor leüríti a mezők tartalmát és bezárja a framet és meghívja a függvényt ami frissíti a táblát, ha valahol elakad hibát dob. Szerver hiba esetén egy responceCode jelenít egy ablakban más hibánál magát a hibaüzenetet.

ActionPerformed():

A különböző gombok lenyomásakor meghívja a hozzá tartozó metódusokat.

(web)

EmptyInputsRegist:

Három bejövő paraméter létrehoz result nevű változót le ellenőrzi azoknak a tartamát ha az egyik üres akkor result true állítja ha mindnek van tartalma a false logikai értékre állítja majd vissza küldi result változót adja vissza.

EmptyInputsLogin:

Kettő bejövő paraméter létrehoz result nevű változót le ellenőrzi azoknak a tartamát ha az egyik üres akkor result true állítja ha mindnek van tartalma a false logikai értékre állítja majd vissza küldi result változót adja vissza.

EmptyInput:

Egy bejövő paraméter létrehoz result nevű változót le ellenőrzi a tartamát ha az üres akkor result true állítja ha van tartalma a false logikai értékre állítja majd vissza küldi result változót adja vissza.

InvalidInputUsername:

Egy bejövő paraméter létrehoz result nevű változót le ellenőrzi a tartamát ha a szöveg tartalmaz nem megengedett karaktert akkor result true állítja ha van tartalma a false logikai értékre állítja majd vissza küldi result változót adja vissza.

InvalidInputPassword:

Egy bejövő paraméter létrehoz result nevű változót le ellenőrzi a tartamát ha a szöveg tartalmaz nem megengedett karaktert

akkor result true állítja ha van tartalma a false logikai értékre állítja majd vissza küldi result változót adja vissza.

InvalidString:

Egy bejövő paraméter létrehoz result nevű változót le ellenőrzi a tartamát ha a szöveg tartalmaz nem megengedett karaktert akkor result true állítja ha van tartalma a false logikai értékre állítja majd vissza küldi result változót adja vissza.

InvalidInt:

Egy bejövő paraméter létrehoz result nevű változót le ellenőrzi a tartamát ha a szöveg tartalmaz nem megengedett karaktert akkor result true állítja ha van tartalma a false logikai értékre állítja majd vissza küldi result változót adja vissza.

PhoneLength:

Egy bejövő paraméter létrehoz result nevű változót le ellenőrzi a szöveg hosszát ha nem megfelelő karakterszámból áll akkor result true állítja ha van tartalma a false logikai értékre állítja majd vissza küldi result változót adja vissza.

InvalidInputEmail:

Egy bejövő paraméter létrehoz result nevű változót le ellenőrzi a szöveg valódi email cím formátumú ha nem akkor result true állítja ha van tartalma a false logikai értékre állítja majd vissza küldi result változót adja vissza.

NotLongEnough:

Egy bejövő paraméter létrehoz result nevű változót le ellenőrzi a szöveg hosszát ha nem megfelelő karakterszámból áll akkor

result true állítja ha van tartalma a false logikai értékre állítja majd vissza küldi result változót adja vissza.

NotLongEnoughUsername:

Egy bejövő paraméter létrehoz result nevű változót le ellenőrzi a szöveg hosszát ha nem megfelelő karakterszámból áll akkor result true állítja ha van tartalma a false logikai értékre állítja majd vissza küldi result változót adja vissza.

ZipCodeLength:

Egy bejövő paraméter létrehoz result nevű változót le ellenőrzi a szöveg hosszát ha nem megfelelő karakterszámból áll akkor result true állítja ha van tartalma a false logikai értékre állítja majd vissza küldi result változót adja vissza.

MatchPass:

Kettő bejövő paraméter létrehoz result nevű változót összehasonlítja őket ha nem egyeznek meg akkor result true állítja ha van tartalma a false logikai értékre állítja majd vissza küldi result változót adja vissza.

UserExistLogin:

Kettő bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt felparaméterezi és végre hajtja. Az adatokat eltárolja egy változóban majd feldolgozza az adatokat vissza küldi az adatokat ha nem sikerült false érteket ad vissza. Végül lezárja a kapcsolatot.

UserExist:

Három bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt felparaméterezi és végre hajtja. Az adatokat eltárolja egy változóban majd feldolgozza az adatokat vissza küldi az adatokat ha nem sikerült false érteket ad vissza. Végül lezárja a kapcsolatot.

UserExistEmail:

Kettő bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt felparaméterezi és végre hajtja. Az adatokat eltárolja egy változóban majd feldolgozza az adatokat vissza küldi az adatokat ha nem sikerült false érteket ad vissza. Végül lezárja a kapcsolatot.

UserExistLoginUserName:

Kettő bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt felparaméterezi és végre hajtja. Az adatokat eltárolja egy változóban majd feldolgozza az

adatokat vissza küldi az adatokat ha nem sikerült false érteket ad vissza. Végül lezárja a kapcsolatot.

UserExistForgetPass:

Kettő bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt felparaméterezi és végre hajtja. Az adatokat eltárolja egy változóban majd feldolgozza az adatokat vissza küldi az adatokat ha nem sikerült false érteket ad vissza. Végül lezárja a kapcsolatot.

CreateUser:

Négy bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt felparaméterezi és végre hajtja. Végül lezárja a kapcsolatot. Vissza add az url keresztül egy üzenetet.

AddSettings:

Kettő bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt felparaméterezi és végre

hajtja. Meghívja a createSettings metódust és át ad neki 2 paramétert. Végül lezárja a kapcsolatot.

CreateSettings:

Kettő bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt felparaméterezi és végre hajtja. Végül lezárja a kapcsolatot.

LoginDate:

Kettő bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt felparaméterezi és végre hajtja. Végül lezárja a kapcsolatot.

EmailUpdate:

Három bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt felparaméterezi és végre hajtja. Végül lezárja a kapcsolatot. Session email változójának értékét megváltoztatja.

PhoneUpdate:

Három bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt felparaméterezi és végre hajtja. Végül lezárja a kapcsolatot.

CityUpdate:

Három bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt felparaméterezi és végre hajtja. Végül lezárja a kapcsolatot.

ZipCodeUpdate:

Három bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt felparaméterezi és végre hajtja. Végül lezárja a kapcsolatot.

NameUpdate:

Három bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást

ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt felparaméterezi és végre hajtja. Végül lezárja a kapcsolatot. Session username változójának értékét megváltoztatja.

StreetUpdate:

Három bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt felparaméterezi és végre hajtja. Végül lezárja a kapcsolatot.

PasswordUpdate:

Három bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz létrehoz empty nevű változót ezt követően titkosítja a password nevű paraméter és eltárolja érétkét majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt felparaméterezi és végre hajtja. Végül lezárja a kapcsolatot.

LoginUser:

Három bejövő paraméter Meghívja a UserExistLogin metódust a vissza kapót értéket eltárolja. Ellenőrzi annak értékét ha false akkor átirányítja a megadott url a hiba üzenettel együtt. Különben kiolvas a metódus bizonyos adatait és össze hasonlítja, ellenőrzi. Ha ezek nem egyeznek meg akkor átirányítja a

megadott url a hiba üzenettel együtt. Máskülönben ellenőrzi annak jogát ha nincs joga az oldalt használatba venni akkor átirányítja a megadott url a hiba üzenettel együtt. Abban az esetben ha joga van használni akkor loginData metódust meghívja át adja az értékeket session indít és 4 különböző értékben eltárolja szükséges adatokat.

RandomString:

1 bejövő paraméterek characters eltárol egy rakás karaktert str meg létre hozzavégig iterál megadatott paraméter hosszán és str nevű változóhoz fűzi a random kiválasztott karaktereket a characters változóból és visszatér str nevű változóval.

SendMail:

Kettő bejövő paraméter létre hoz sql utasítást ami sql nev változóban tárol el. Kapcsolódik az adatbázishoz majd mysqli_stmt_prepare()-be helyezi az kapcsolatot és sql utasítást ha ez nem sikerült visszatér a megadott linkre a hibaüzlettel amit a linken keresztül továbbít az oldalra és megszakítja folyamatot. Abban az esetben ha ez sikeres volt meghívja randomString metódust át ad neki egy értéket majd a vissza kapott értéket eltárolja majd ezt tokosítja felparaméterezi és végre hajtja. Végül lezárja a kapcsolatot. Meghívja a UserExistEmail metódust átadja a szükséges értéket a visszakapottat meg eltárolja. Ezt követően meghívja SendMailReal metódust és átadja a szükséges értékeket.

SendMailReal:

Három bejövő paraméter Létre hozza az email küldéshez szükséges formaiságokat és legenerálja az üzenetet és elküldi az

emailt. A sikerességtől függően vissza ad egy url az üzenettel együtt.

SendEmail:

Hat bejövő paraméter. Létrehozza az sql utasítást bindValue felparaméterezi és végül végrehajta.

SelectItem:

Kettő bejövő paraméter. Létrehozza az sql utasítást összekapcsolja a táblákat bindValue felparaméterezi és végül végrehajtja. Majd Egy asszociatív többként visszaküldi.

SelectImgs:

Kettő bejövő paraméter. Létrehozza az sql utasítást bindValue felparaméterezi és végül végrehajtja. Majd Egy asszociatív többként visszaküldi.

SelectDistricts:

1 bejövő paraméter. Létrehozza az sql utasítást és végül végrehajtja. Majd Egy asszociatív többként visszaküldi.

SelectTypes:

1 bejövő paraméter. Létrehozza az sql utasítást és végül végrehajtja. Majd Egy asszociatív többként visszaküldi.

SelectMinPrice:

1 bejövő paraméter. Létrehozza az sql utasítást és végül végrehajtja. Majd Egy asszociatív többként visszaküldi.

SelectMaxPrice:

1 bejövő paraméter. Létrehozza az sql utasítást és végül végrehajtja. Majd Egy asszociatív többként visszaküldi.

SelectConditions:

1 bejövő paraméter. Létrehozza az sql utasítást és végül végrehajtja. Majd Egy asszociatív többként visszaküldi.

SelectWatchItem:

Kettő bejövő paraméter. Létrehozza az sql utasítást összekapcsolja a táblákat bindValue felparaméterezi és végül végrehajtja. Majd Egy asszociatív többként visszaküldi.

Sorted:

1 bejövő paraméter A beérkező érték alapján vissza küld egy értéket.

SelectMassage:

Négy bejövő paraméter. Létrehozza az sql utasítást összekapcsolja a táblákat bindValue felparaméterezi és végül végrehajtja. Majd Egy asszociatív többként visszaküldi.

SelectItemsMes:

Kettő bejövő paraméter. Létrehozza az sql utasítást összekapcsolja a táblákat bindValue felparaméterezi és végül végrehajtja. Majd Egy asszociatív többként visszaküldi.

NotSelect:

1 bejövő paraméter. Étéke alapján visszaküld egy logikai értéket.

InsertIntoItems:

12 bejövő paraméter.
Létrehozza az sql utasítást összekapcsolja a táblákat bindValue felparaméterezi és végül végrehajtja.

ItemId:

1 bejövő paraméter. Létrehozza az sql utasítást és végül végrehajtja. Majd Egy asszociatív többként visszaküldi.

InsertImgs:

Négy bejövő paraméter. Létrehozza az sql utasítást összekapcsolja a táblákat bindValue felparaméterezi és végül végrehajtja. Majd Egy asszociatív többként visszaküldi.

SellItem:

Kettő bejövő paraméter. Létrehozza az sql utasítást bindValue felparaméterezi és végül végrehajtja. Majd Egy asszociatív többként visszaküldi.

DelItem:

Kettő bejövő paraméter. Létrehozza az sql utasítást bindValue felparaméterezi és végül végrehajtja.

DelImg:

Kettő bejövő paraméter. Létrehozza az sql utasítást bindValue felparaméterezi és végül végrehajtja.

UpdateIntoItems:

13 bejövő paraméter. Létrehozza az sql utasítást bindValue felparaméterezi és végül végrehajtja.

CheckAllow:

2 bejövő paraméter. Létrehozza az sql utasítást összekapcsolja a táblákat bindValue felparaméterezi és végül végrehajtja. Majd Egy asszociatív többként visszaküldi.

(laravel)

routes/api.php:

Útvonalak, API végpontok

Végpontok	Metódusok	Middleware	Leírás
/kenorby	Post	Igen	Adatfrissítés
/counter	Post	Igen	Számláló adati megjelenítése
/sum	Post	Igen	Összes számlálói adat megjelenítése
/login	Post	Nem	Belépés a fiókba
/logout	Post	Igen	Kilépés a fiókól
/desktop-login	Post	Nem	Belépés az asztali alkamásba
/desktop-update-user-data	Post	Igen	Asztali alkalmazás felhasználói adat módosítás
desktop-update-status-active	Post	Igen	Állapot módosítás
/desktop-update-status-block	Post	Igen	Állapot módosítás

/desktop-create-user	Post	Igen	Admin felvétele
/desktop-all-user	Post	Igen	Összes felhasználó megjelenítése
/desktop-search	Post	Igen	Keresés
/desktop-select-id	Post	Igen	Id lekérése

BaseController

Két metódust tartalmaz:

SendResponse($result,$message):
Két bejövő paraméter vissza küld egy $response nevű asszociatív tömböt „data” betöltve a $result értékét a „message” meg $message értékét. Majd ezt json formátumba alakítja. Végül visszatér a $response és egy 200 kódot.

sendError($error,$errorMessages=[]):
Két bejövő paraméter $errorMessages alap értéke egy üres tömb. Létrehoz $response nevű asszociatív tömböt „data” betöltve a $errorMessages értékét abban az esetben ha annak értéke nem üres „message” meg $error értékét. Majd ezt json formátumba alakítja. Végül visszatér a $response és egy 404 kódot.

AuthController

Két metódust tartalmaz:

signIn(Request $request):
Egy bejövő paraméter. $data eltárolja a sql lekérdezés eredményét. Majd ellenőrzi a lekérdezett adat és bejövő paraméter Userpass elemével megegyezik ha igen $succes listában eltárolja a szükséges adatokat. Majd BaseController található sendResponse függvényt felparaméterezi és visszatér az értékével. Abban az esetben ha az adat összehasonlítás false értéket adatott akkor a BaseController sendError függvényét paraméterezi fel és tér vissza annak értékével.

logut(Request $request):
Fejlesztés alatt.

KenorbyController

11 metódust tartalmaz:

update(Request $request):
Egy bejövő paraméter. Ha sikerül frissíteni az adatbázis táblájának kiválasztott sorát akkor visszatér BaseController található sendResponse függvénnyel. Felparaméterezi és visszatér az értékével. Ha a folyamat sikertelen volt akkor BaseController sendError függvényét paraméterezi fel és tér vissza annak értékével.

selectMyCounter(Request $request):

Egy bejövő paraméter. Ha sikerül lekérdeznie az adatbázisban található adatok(at) visszatér BaseController található sendResponse függvénnyel. Felparaméterezi és visszatér az értékével. Ha a folyamat sikertelen volt akkor BaseController sendError függvényét paraméterezi fel és tér vissza annak értékével.

sum():

Nincs bejövő paraméter. Ha sikerül lekérdeznie az adatbázisban található adatok(at) visszatér BaseController található sendResponse függvénnyel. Felparaméterezi és visszatér az értékével. Ha a folyamat sikertelen volt akkor BaseController sendError függvényét paraméterezi fel és tér vissza annak értékével.

login():

Bejövő paraméter request a request→UserName-je alapján lekérdezi az adatott az adatbázisból összehasonlítja a az ott található jelszót és a requestben található password-öt az eredmény alapján hívja meg a sendResponse felparaméterezve vagy a sendError felparaméterezve.

all_user():

Bejövő paraméter nincs. Lekérdezi az adatott az adatbázisból majd az eredmény alapján hívja meg a sendResponse felparaméterezve vagy a sendError felparaméterezve.
login():

Bejövő paraméter request a request→UserName-je alapján lekérdezi az adatott az adatbázisból összehasonlítja a az ott található jelszót és a requestben található password-öt az eredmény alapján hívja meg a sendResponse felparaméterezve vagy a sendError felparaméterezve.

create_admin():

Bejövő paraméter request.A request→Password-jét titkosítja majd végre hajtja az sql utasítást eredmény alapján hívja meg a sendResponse felparaméterezve vagy a sendError felparaméterezve.

active():

Bejövő paraméter request.A request→UserName-je alapján frissíti a adatott az adatbázisban majd meghívja a sendResponse felparaméterezve.

blocked():

Bejövő paraméter request.A request→UserName-je alapján frissíti a adatott az adatbázisban majd meghívja a sendResponse felparaméterezve.

update_user():

Bejövő paraméter request. A request alapján végre hajtja a sql utasítást majd vissza tér sendPesponse-val felparaméterezve.

search_id():

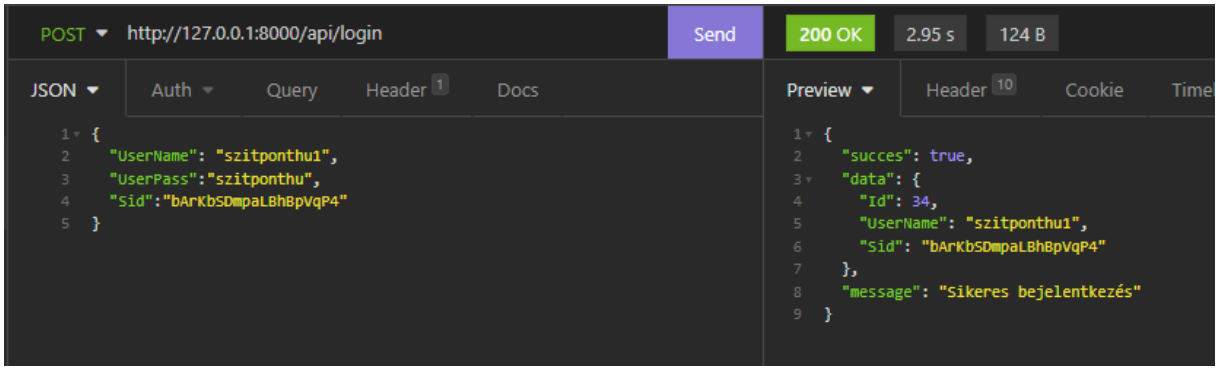
Bejövő paraméter request. A request alapján végre hajtja a sql utasítást majd az eredmény függően vissza tér sendResponse-val felparaméterezve vagy a sendError-ral

search():

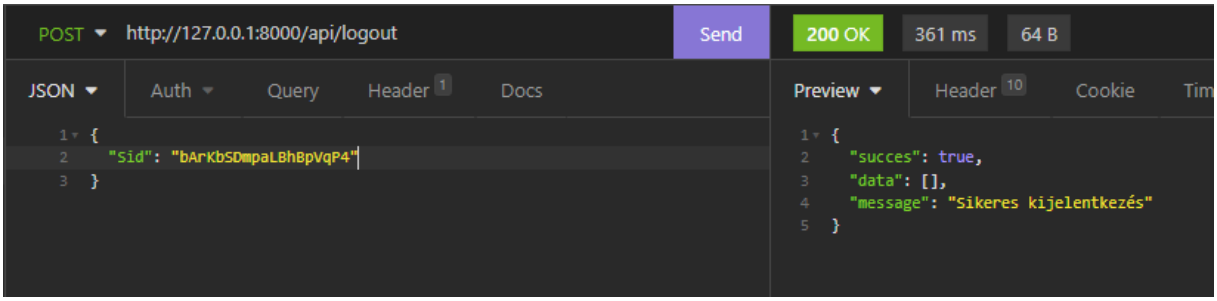
Bejövő paraméter request. A request alapján végre hajtja a sql utasítást majd az eredmény függően vissza tér sendResponse-val felparaméterezve vagy a sendError-ral.

(Tesztelés)

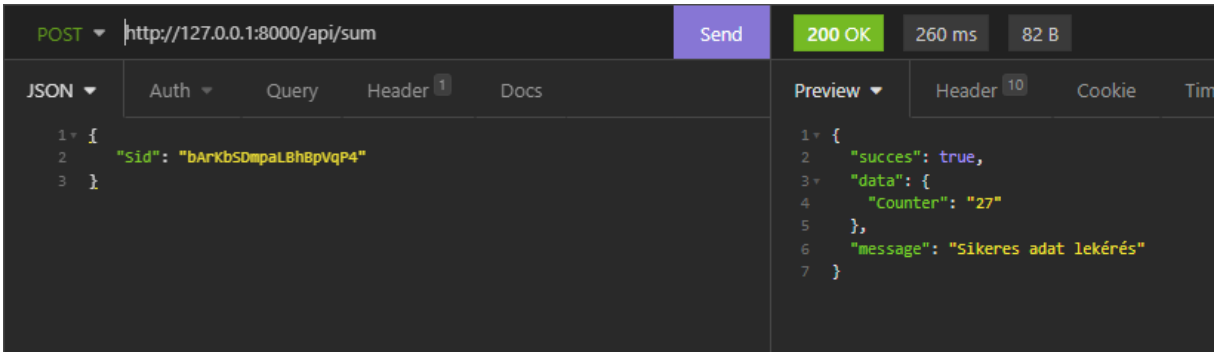
/Login



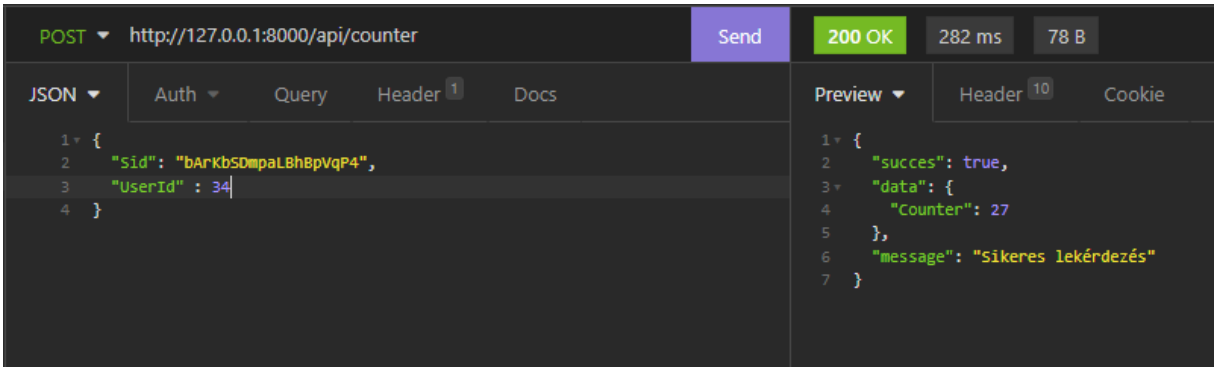
/logout



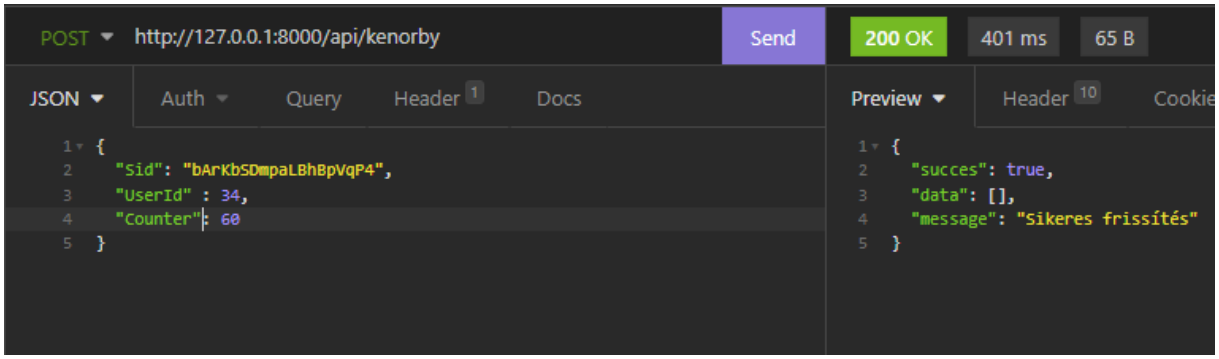
/sum



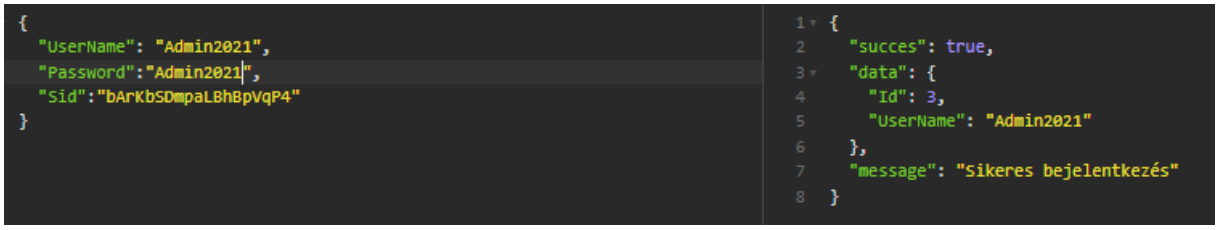
/counter



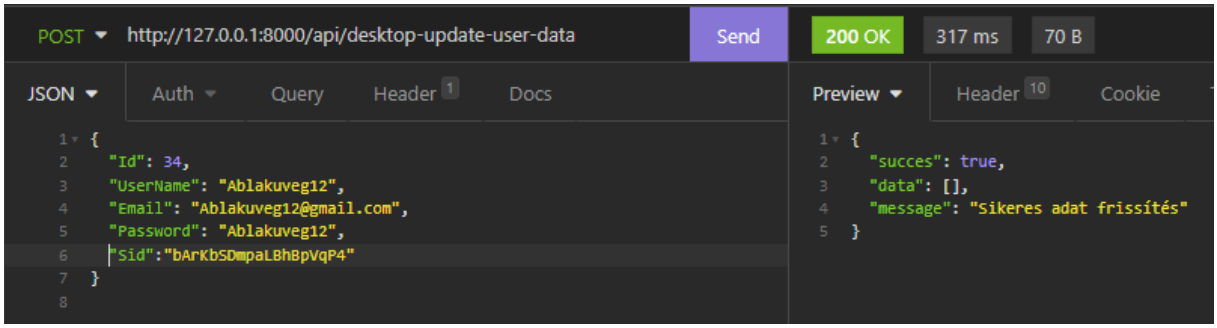
/kenorby



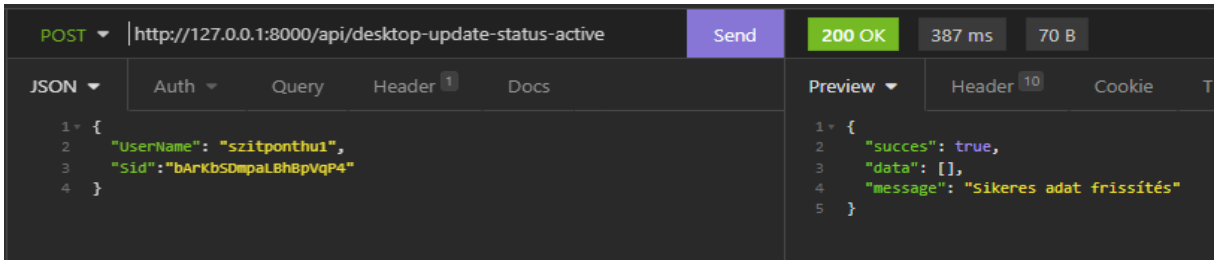
/desktop-login



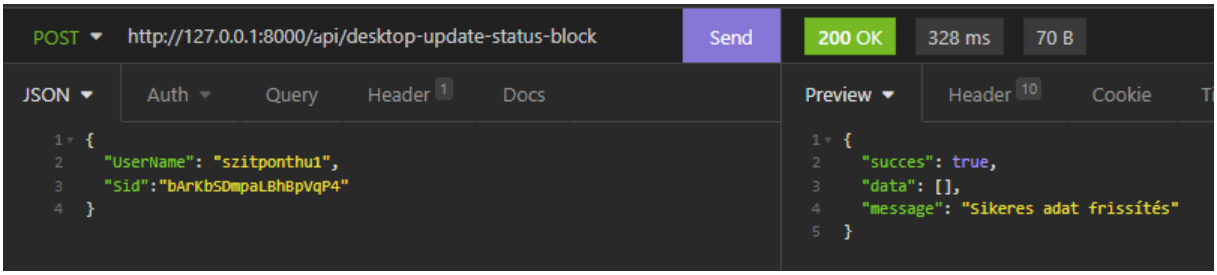
/desktop-update-user-data



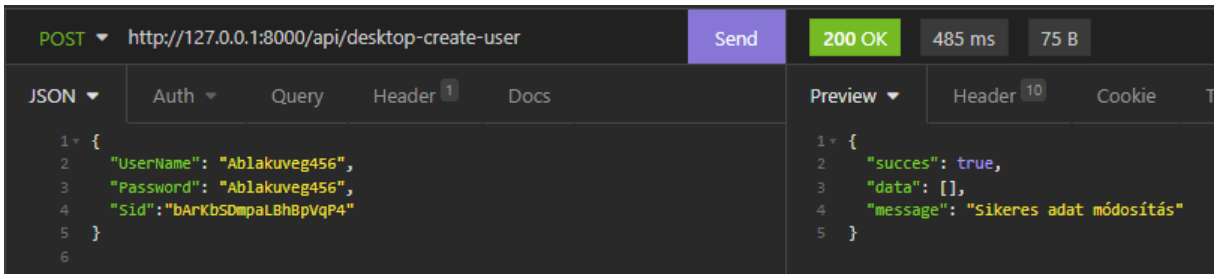
/desktop-update-statues-active



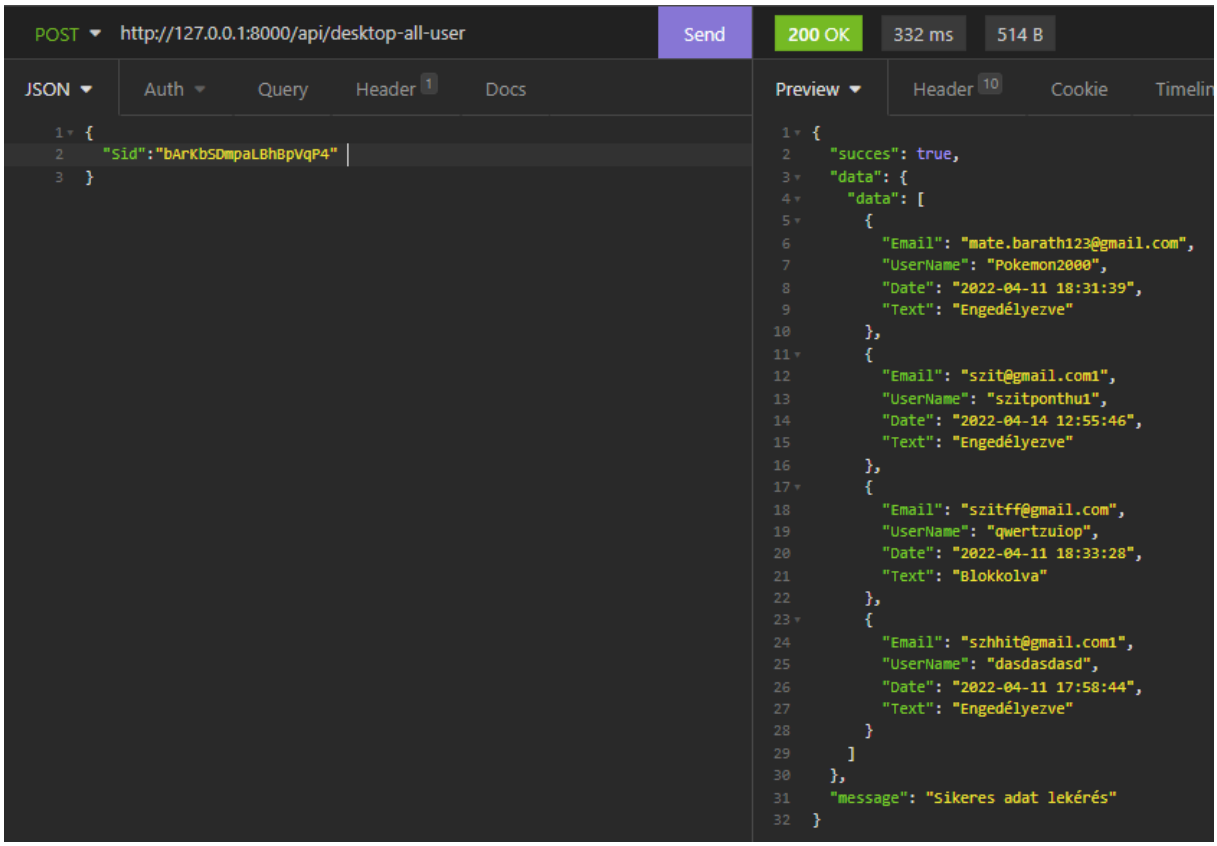
/desktop-update-statues-block



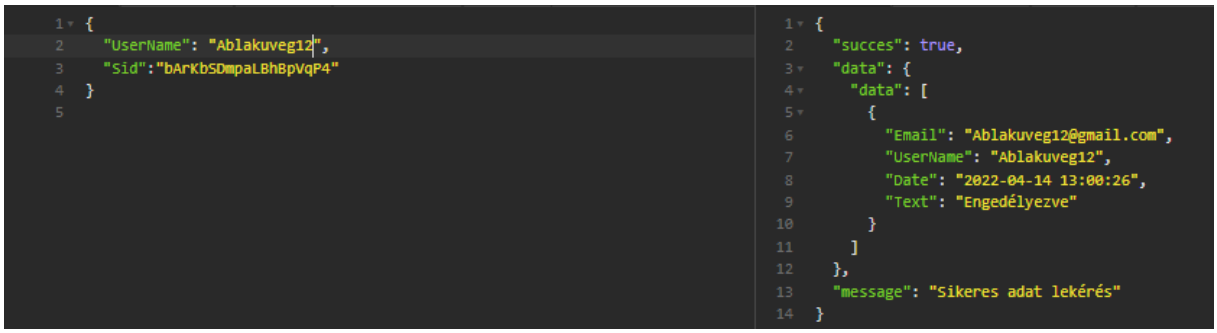
/desktop-create-user



/desktop-all-user



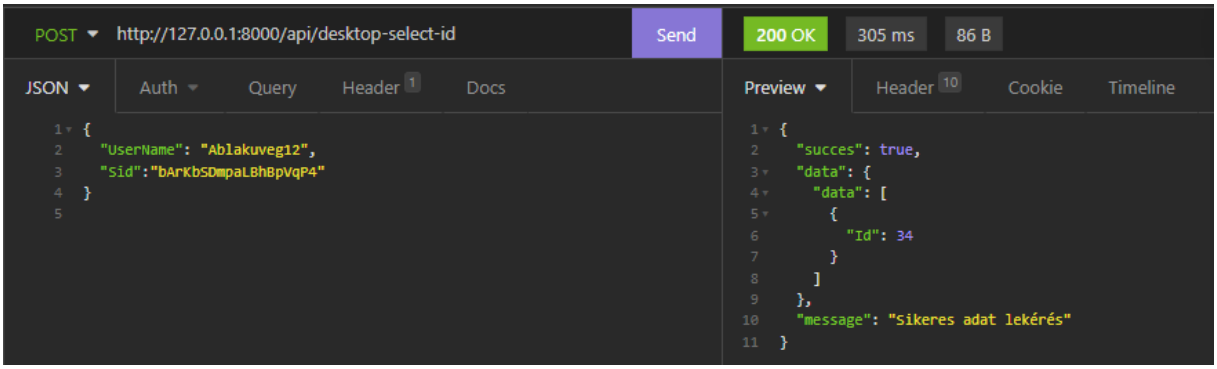
/desktop-search



```
{
  "UserName": "Ablakuveg12",
  "Sid":"bArKbSDmpaLBhBpVqP4"
}
```

```
{
  "succes": true,
  "data": {
    "data": [
      {
        "Email": "Ablakuveg12@gmail.com",
        "UserName": "Ablakuveg12",
        "Date": "2022-04-14 13:00:26",
        "Text": "Engedélyezve"
      }
    ]
  },
  "message": "Sikeres adat lekérés"
}
```

/desktop-select-id




```
{
  "UserName": "Ablakuveg12",
  "Sid":"bArKbSDmpaLBhBpVqP4"
}
```

```
{
  "succes": true,
  "data": {
    "data": [
      {
        "Id": 34
      }
    ]
  },
  "message": "Sikeres adat lekérés"
}
```

Tartalomjegyzék