A képen szöveg látható

Automatikusan generált leírás

Web programozás II. beadandó feladat

Badinszky Dániel

HLISZU

*A beadandót egyedül késízettem, de a feladat kiírásnak megfelelően 2 git accounttal.*

*Webhely: http://hliszu.nhely.hu/*

*Github: https://github.com/badinszky97/web2Soap*

**2024.10.26.**

Tartalom

[Téma 3](#_Toc180314972)

[A weboldal bemutatása 4](#_Toc180314973)

[Feladatok 4](#_Toc180314974)

[Az első oldalon mutassa be a céget egy látványos weboldalon 4](#_Toc180314975)

[Legyen Regisztráció, Bejelentkezési lehetőség 4](#_Toc180314976)

[Legalább 3 felhasználói szerepet különböztessen meg 5](#_Toc180314977)

[Menük elérési mátrixa 5](#_Toc180314978)

[Létező felhasználók az adatbázisban 5](#_Toc180314979)

[Legyen egy oldal, ahol a választott adatbázisból jelenít meg adatokat Ehhez 3 tábla adatait használja fel az adatbázisból. 6](#_Toc180314980)

[Az egyik oldalon legyen egy kapcsolat űrlap, amelynek segítségével üzenetet lehet küldeni az oldal tulajdonosa számára. 6](#_Toc180314981)

[Tegye lehetővé megtekinteni egy hatodik oldalon táblázatban az előző pontban elküldött üzeneteket az adatbázisból fordított időrend szerint 7](#_Toc180314982)

[Valósítson meg az alkalmazásban egy RESTful API-t 7](#_Toc180314983)

[cURL 7](#_Toc180314984)

[Postman 9](#_Toc180314985)

[Használják a GitHub (github.com) verziókövető rendszert. 12](#_Toc180314986)

[Előkészületek a JAR fájlhoz 13](#_Toc180314987)

# Téma

A választott adatbázis: Forgalomkorlátozások.

Ez az adatbázis magyarországi utakat tartalmaz és azok lezárásainak adatait. Összesen 3 tábla található benne. Az adatok szöveges formátumban voltak elérhetőek, így az adatbázis szerkezetet kézzel hoztam létre, majd illesztettem bele az adatokat.

A feladat kiírásának megfelelően létrehoztam egy “felhasznalok” nevű táblát, amiben a regisztrált felhasználók adatai vannak tárolva.

A menü tábla tárolja a megjeleníthető menük adatait és jogköreit.

A korlátozás tábla tartalmazza a megjelenítendő adatok főbb részét.

A megnevezés és mérték táblák ehhez vannak több az egyhez kapcsolatban kapcsolva.



# A weboldal bemutatása

A felhasználókezelést a Web programozás I-en tanult módszerrel valósítottam meg. Frontend controller biztosítja a dinamikus bővítési lehetőséget. A menükben megjelenő opciók adatbázisban vannak tárolva.

A PHP kód minden oldalbeolvasáskor összeveti a jogkört az adatbázisban eltárolt jogokkal és az alapján jeleníti meg a menüt.

$sql = "SELECT \* FROM menu WHERE vendeg=1";

if(isset($\_SESSION['login']) && $\_SESSION['admin']=0)

{

// felhasznalo

$sql = "SELECT \* FROM menu WHERE felhasznalo=1";

}

if(isset($\_SESSION['login']) && $\_SESSION['admin']=1)

{

// admin

$sql = "SELECT \* FROM menu WHERE felhasznalo=1";

}

$stmt = $dbh->prepare($sql, array(PDO::ATTR\_CURSOR => PDO::CURSOR\_SCROLL));

$stmt->execute();

while ($row = $stmt->fetch(PDO::FETCH\_ASSOC, PDO::FETCH\_ORI\_NEXT)) {

echo "<li>";

if($row['url'] == "/")

{

$row['url'] = ".";

}

else

{

$row['url'] = "?oldal=" . $row['url'];

}

echo "<a href=" . $row['url'] . ">" . $row['szoveg'] . "</a>";

echo "</li>";

}



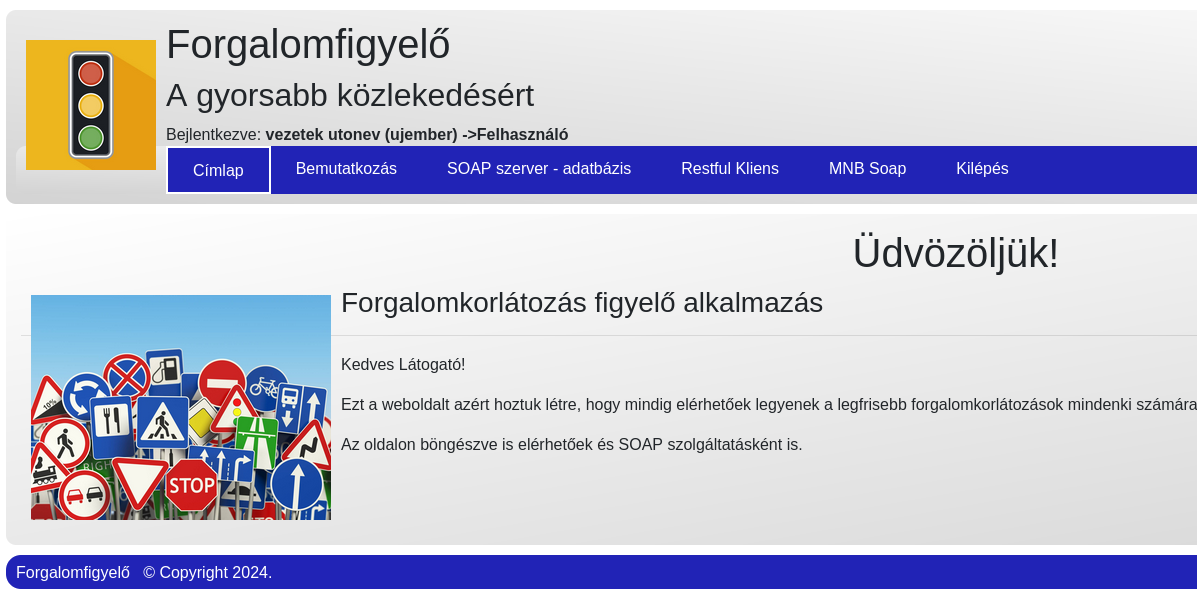
# Feladatok

## Reszponzív téma

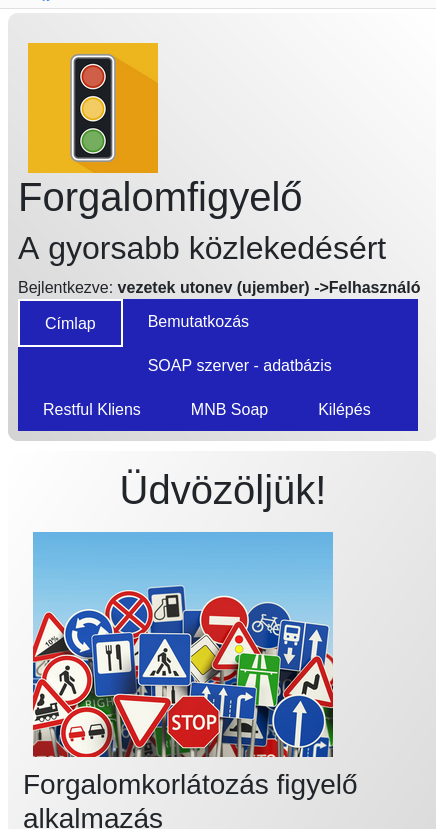
Bootstrapet alkalmaztam a weboldalon, hogy reszponzív felületet tudjak kialakítani. Az internetről töltöttem le a borítóképet és egy rövid leírást tettem ki a főoldalra. A hozzá tartozó fájlállományok nincsenek a szerverre másolva, így biztosítva a gyorsabb oldal betöltődést.

A bootstrapen belül is több téma alakítható ki. Én a bootswatch oldalon található Zephyr témát választottam. (https://bootswatch.com/zephyr/)

Asztali nézet:



Mobilos nézet:



## Legyen Regisztráció, Bejelentkezési lehetőség

Alapvető funkciója a felhasználókezelés. 3 szintet különböztet meg: Vendég, Felhasználó, Admin.

Egy vendég nem lát semmit a weboldalon a bemutatkozó oldalon kívül. Egy bejelentkezett felhasználó eléri a SOAP és Restful alkalmazások kezelőfelületeit. Egy admin ugyan azt látja, mint egy bejelentkezett felhasználó, csak egy admin oldallal többet.

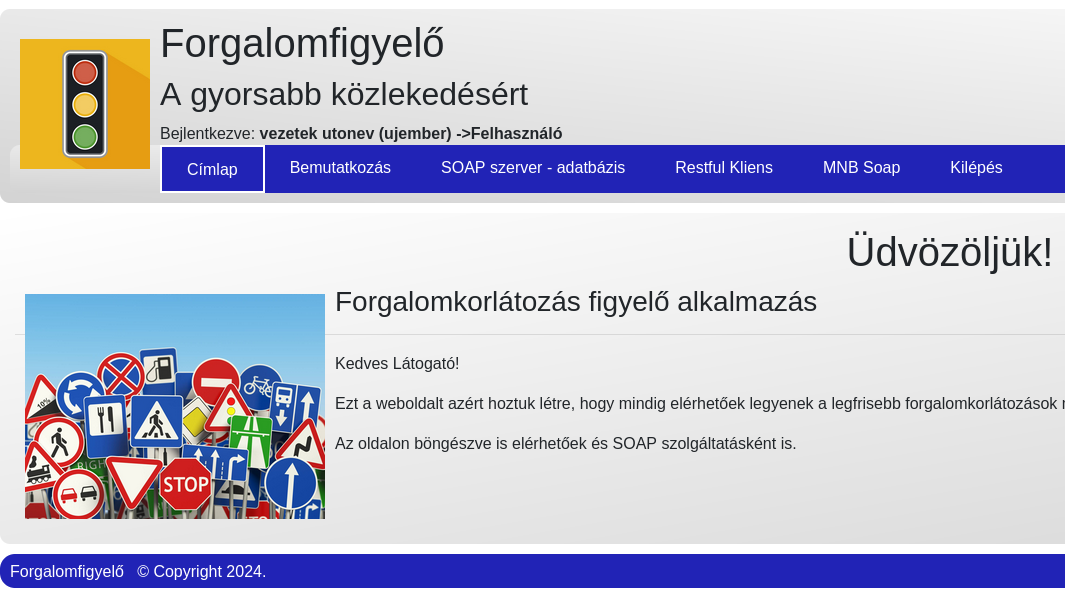
*A jelszó sha1-es hashkódként vannak tárolva az adatbázisban. A bejelentkezési adatok SESSION adatként vannak tárolva szerver oldalon.*

|  |  |  |
| --- | --- | --- |
| Felhasználónév | Jelszó | Jog |
| dani | dani | Admin |
| ujember | ujember | Felhasználó |

*Menük elérésének mátrixa jogkörönként oldalakra vetítve:*

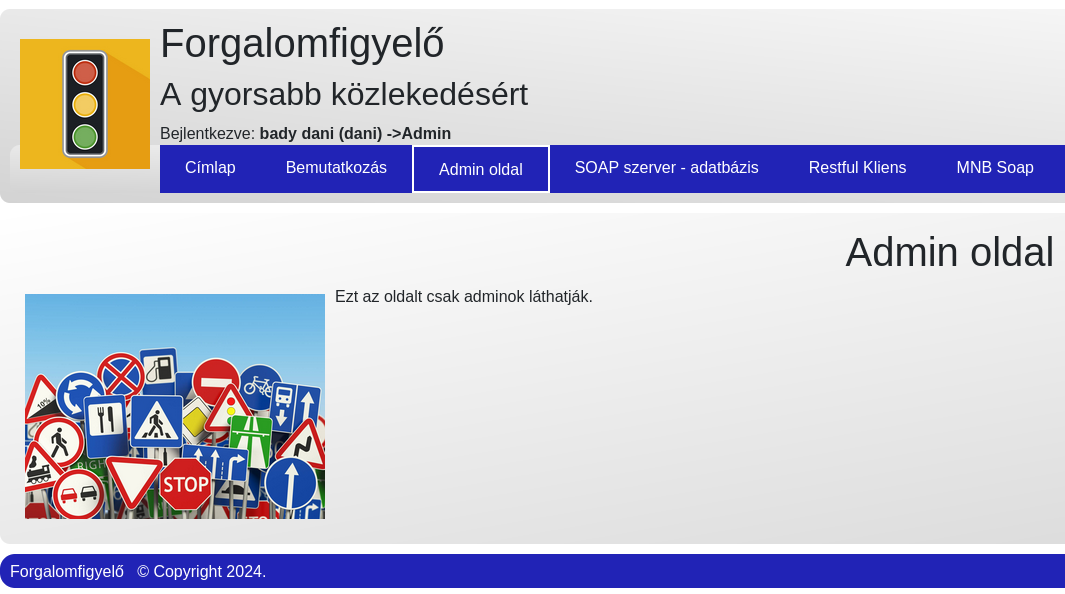
|  |  |  |  |
| --- | --- | --- | --- |
|  | Vendég | Felhasznalo | Admin |
| / |  |  |  |
| bemutatkozas |  |  |  |
| belepes |  |  |  |
| Admin |  |  |  |
| kilepes |  |  |  |
| belep |  |  |  |
| regisztral |  |  |  |
| korlatozasok |  |  |  |
| restfulkliens |  |  |  |
| mnbsoap |  |  |  |

*A regisztráció menüponttal egy felhasználó kerül kétrehozásra. Admin jogkört kizárólag adatbázisból lehet adni. Regisztráció után be is lehet jelentkezni és a fejlécben megjelenik a bejelentkezett felhasználó neve és jogköre.*

**

*Ilyenkor a kilépés gomb aktívvá válik és a bejelentkezés/regisztráció ablak nem elérhető.*

Adminként lehetőség van egy olyan oldalt is megnézni amit sima felhasználó nem lát:



Ezzel lett megvalósítva a 3 megkülönböztetett szint.

FELADAT:

Soap szerver

A weboldal képes soap kéréseket kiszolgálni. A WSDL fájl publikus címe:

http://hliszu.nhely.hu/szerver/lezarasok.wsdl

Kettő darab publikus funkció lett megvalósítva:

getlezarasok()

Ez a funkció összefűzi a 3 adattáblát és egyként a felhasználónak átadja. A megnevezés és mérték valójában külső táblában vannak tárolva.

getlezaras(utszam, telepulesnev)

Ez a funkció leszűkíti a visszaadandó találatokat útszám ÉS településnév-re. A feltételeknek együtt kell teljesülniük. Ezzel biztosítjuk, hogy végül csak egy rekordot kapjunk vissza.

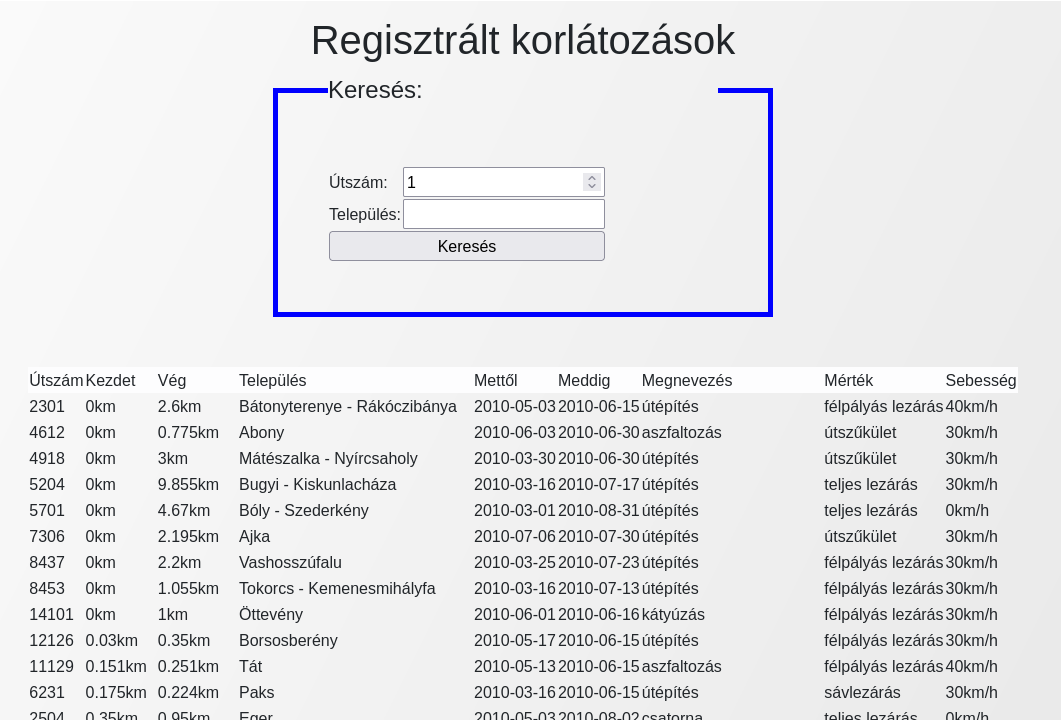
Az ehhez tartozó kód a *szerver*/lezarasok.php fájlban találhatóak.

Ezzel a két funkcióval megvalósítottam a paraméteres és paraméter nélüli soap kérést is.

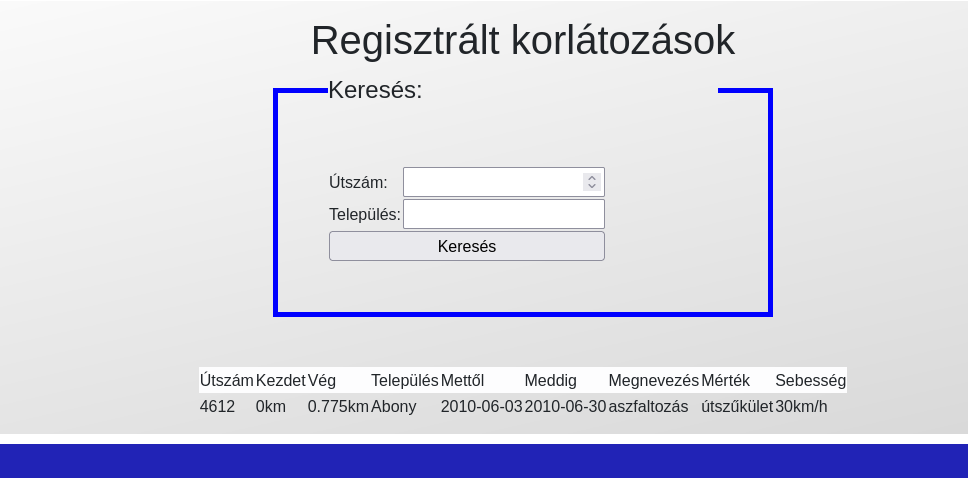
FELADAT

Soap kliens

A Soap szerver teszteléséhez készítettem egy klienst a weboldalra ami localhost-on keresztül a saját szerveréhez címez kérést. Ennek eredménye látható a “SOAP szerver – adatbázis” oldalon.



Hogyha nem keresünk rá semmire, vagy úgy keresünk, hogy a település mező üres akkor paraméter nélüli kereséséként az összes találatot fel fogja adni. Amennyiben mind a két mezőt kitöltjük paraméteres keresésként csak az egyező rekord jelenik meg:

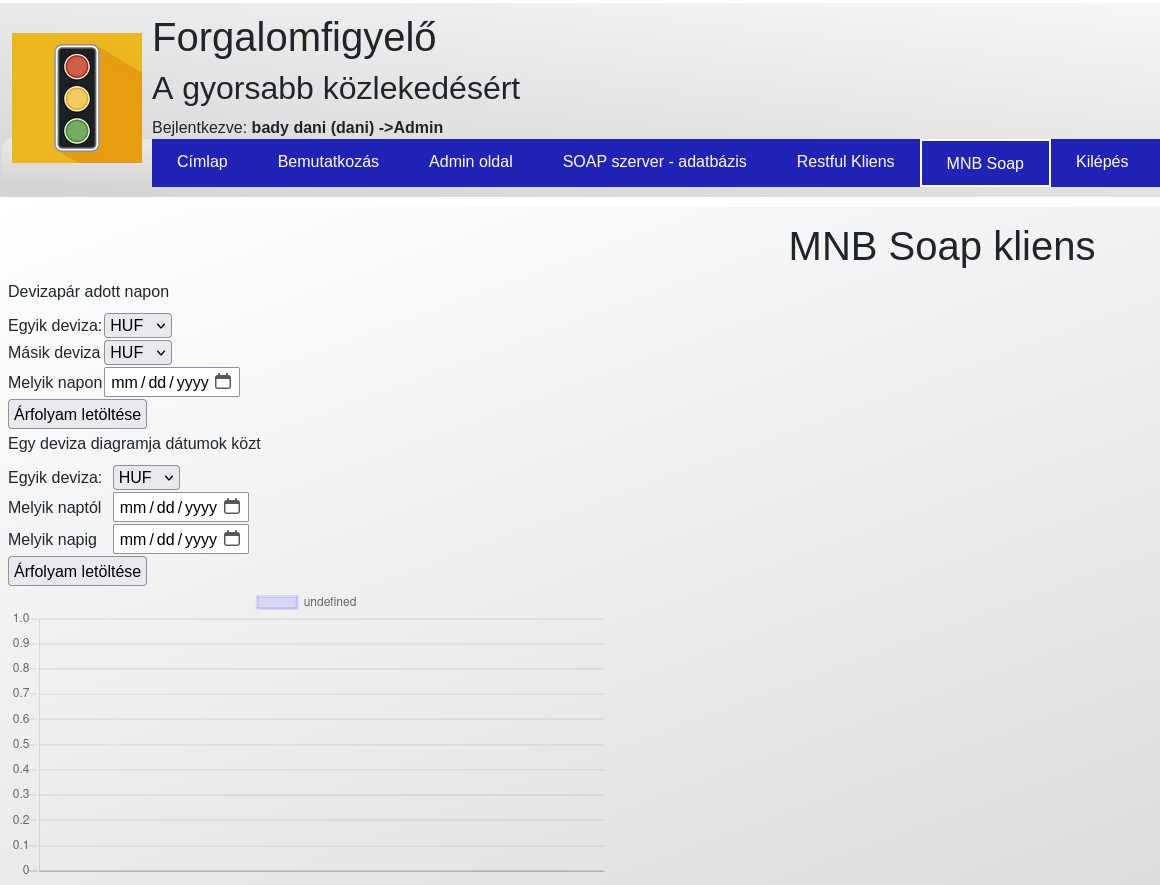


FELADAT

Soap MNB

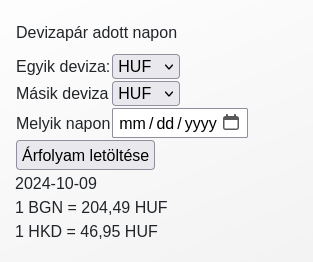
Készítettem egy klienst ami az MNB publikus Soap szerverét címzi és kér tőle adatokat.

Ez az ‘MNB Soap’ menüpont alatt érhető el.

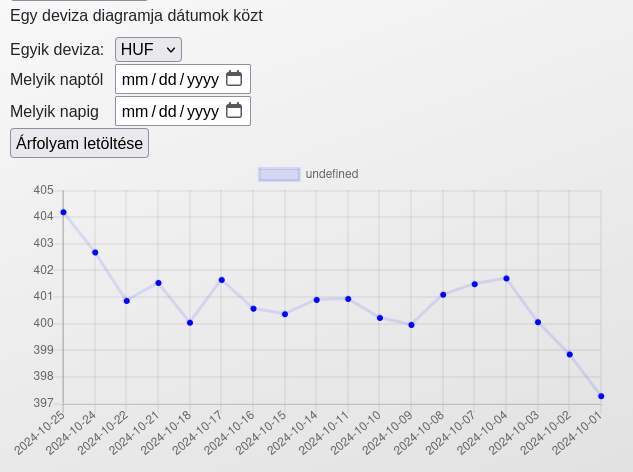


Az oldalt megnyitva 2db üres formot találunk egymás alatt. Ez a két feladatot valósítja meg a beadandó kiírásában.

A “Devizapár adott napon” funkció képes két deviza árát lekérdezni egy adott napon. Ennek kijelzése a form alatt jelenik meg a felső “Árfolyam letöltése” gomb megnyomása után.



Az “Egy deviza diagramja dátumok közt” form képes egy deviza árfolyamát két adott időpont közt diagrammon megjeleníteni. A devizát kiválasztva és a dátumok megadása után az “Árfolyam letöltése” gomb megrajzolja a diagrammot.



Mind a két formhoz a devizaértékek is Soap lekérdezéssel lettek lekérdezve az MNBtől valós időben.

FELADAT

Restful szerver – kliens

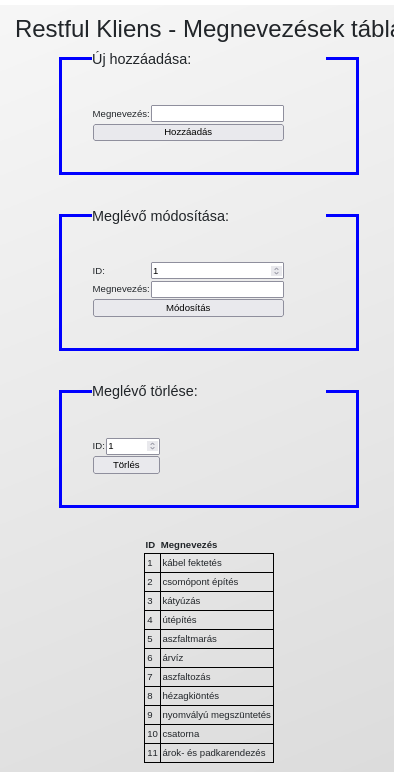
A weboldal képes Restful kéréseket kiszolgálni. Ennek URL-je:

http://hliszu.nhely.hu/szerver/restfulszerver.php

Megvalósítja a GET,POST,PUT,DELETE típusokat is.

Alapesetben a formon 3 form és egy táblázat jelenik meg alul.

GET

Az alsó táblázat mutatja az adatbázistábla aktuális adatait minden oldal betöltésekor. Ezt GET kéréssel a restful szerver küldi. Csak a Restful szerver kapcsolódik ebben az esetben ehhez az adatbázis táblához.

A három form megvalósítja a POST,PUT,DELETE funkciókat.

Restful szerver:

$dbh = new PDO('mysql:host=localhost;dbname=hliszu','root', '',

array(PDO::ATTR\_ERRMODE=>PDO::ERRMODE\_EXCEPTION));

$dbh->query('SET NAMES utf8 COLLATE utf8\_hungarian\_ci');

switch($\_SERVER['REQUEST\_METHOD']) {

case "GET":

$sql = "SELECT \* FROM megnevezes";

$sth = $dbh->query($sql);

$eredmeny .= "<table style=\"border-collapse: collapse;\"><tr><th>ID</th><th>Megnevezés</th></tr>";

while($row = $sth->fetch(PDO::FETCH\_ASSOC)) {

$eredmeny .= "<tr>";

foreach($row as $column)

$eredmeny .= "<td style=\"border: 1px solid black; padding: 3px;\">". $column."</td>";

$eredmeny .= "</tr>";

}

$eredmeny .= "</table>";

break;

case "POST":

$sql = "INSERT INTO megnevezes (nev) VALUES ('" . $\_POST['megnevezes'] ."')";

$sth = $dbh->query($sql);

$eredmeny = $sth;

break;

case "PUT":

$incoming = file\_get\_contents("php://input");

parse\_str($incoming, $data);

$sql = "UPDATE megnevezes SET nev =\"" . $data['megnevezesnev'] ."\" WHERE id=". $data['id'];

$sth = $dbh->query($sql);

$eredmeny = $sth;

break;

case "DELETE":

$incoming = file\_get\_contents("php://input");

parse\_str($incoming, $data);

$sql = "DELETE FROM megnevezes WHERE id=". $data['id'];

$sth = $dbh->query($sql);

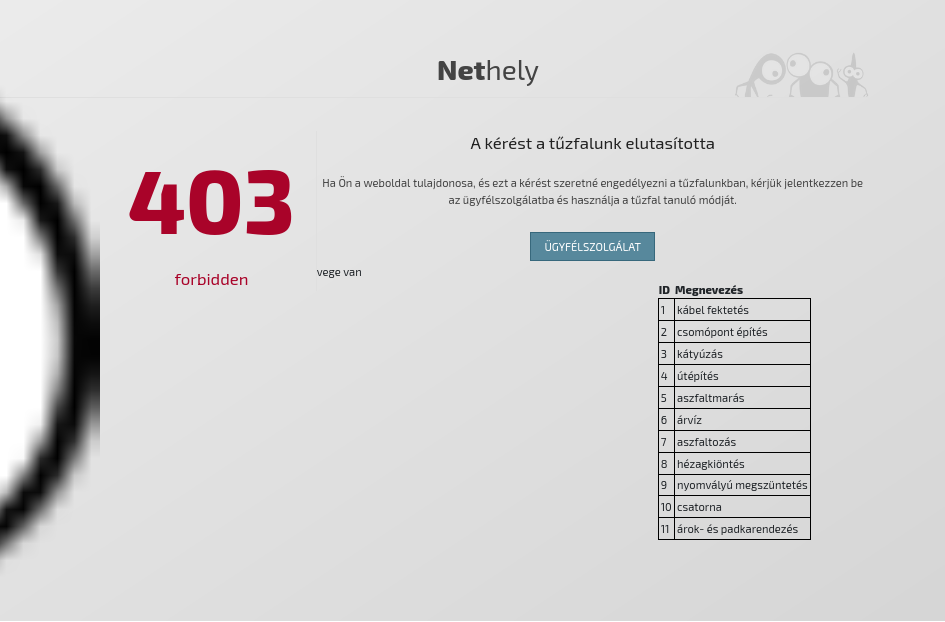
$eredmeny = $sth;

break;

}

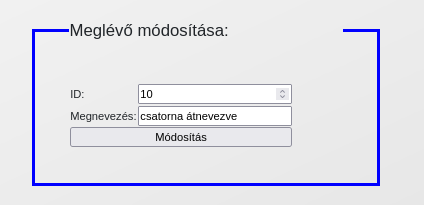
POST

A legfelső form képes hozzáadni a táblához egy új elemet. ID megadása nem szükséges. Ez a funkció otthoni xampp szerveren tökéletesen működik. Sajnos a nethelyre feltöltve a POST kéréseim blkkolva vannak a tűzfaluk által.



A tűzfal tanuló módba állítása és utána teljes kikapcsolása sem odlotta meg a problámát. Ezután felvettem a kapcsolatot az oldal üzemeltetőjével, hogy miért nem engedélyezett a post küldés ebben az esetben, mik jelennek meg a logfájlban de sajnos nem válaszoltak. Ez a fajta működés felettéb furcsának hat, mert a GET,PUT és DELETE funkciók tökéletesen működnek ugyan azzal a típusú kóddal. Mivel cURLből tökéletesen működik a funkció, így a többszöri POST küldések sorozata lehet a probléma a tűzfalnak, de a funkcionalitás működőképes a háttérben.

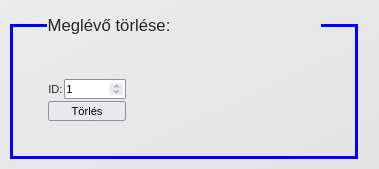
PUT

A put funkció képes módosítani egy megnevezést a táblában. Az ID mező adja meg, hogy melyik rekord lesz módosítva és a Megnevezés mező adja meg az új értékét. A példában a 10-es számú rekord értékét módosítom “csatorna átnevezve” értékre.



DELETE

A delete form képes egy elemet kitörölni a táblából. Itt elég megadni az elem Idjét és a törlés gombra az oldal újratöltődésekor már nem lesz a listában.



FELADAT

Restful tesztelése külső appal:

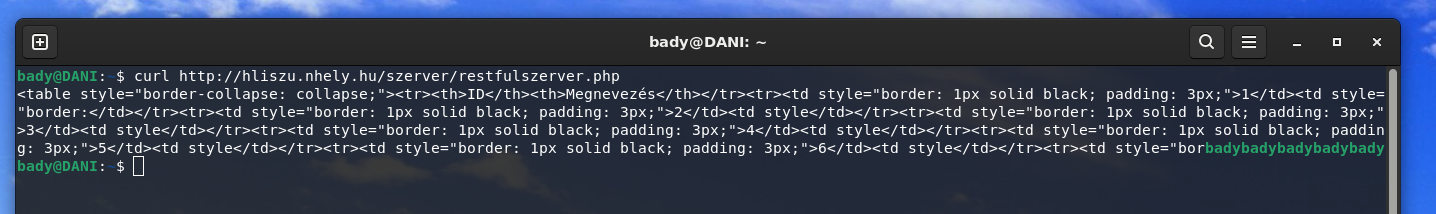
Az ehhez használt url:

http://hliszu.nhely.hu/szerver/restfulszerver.php

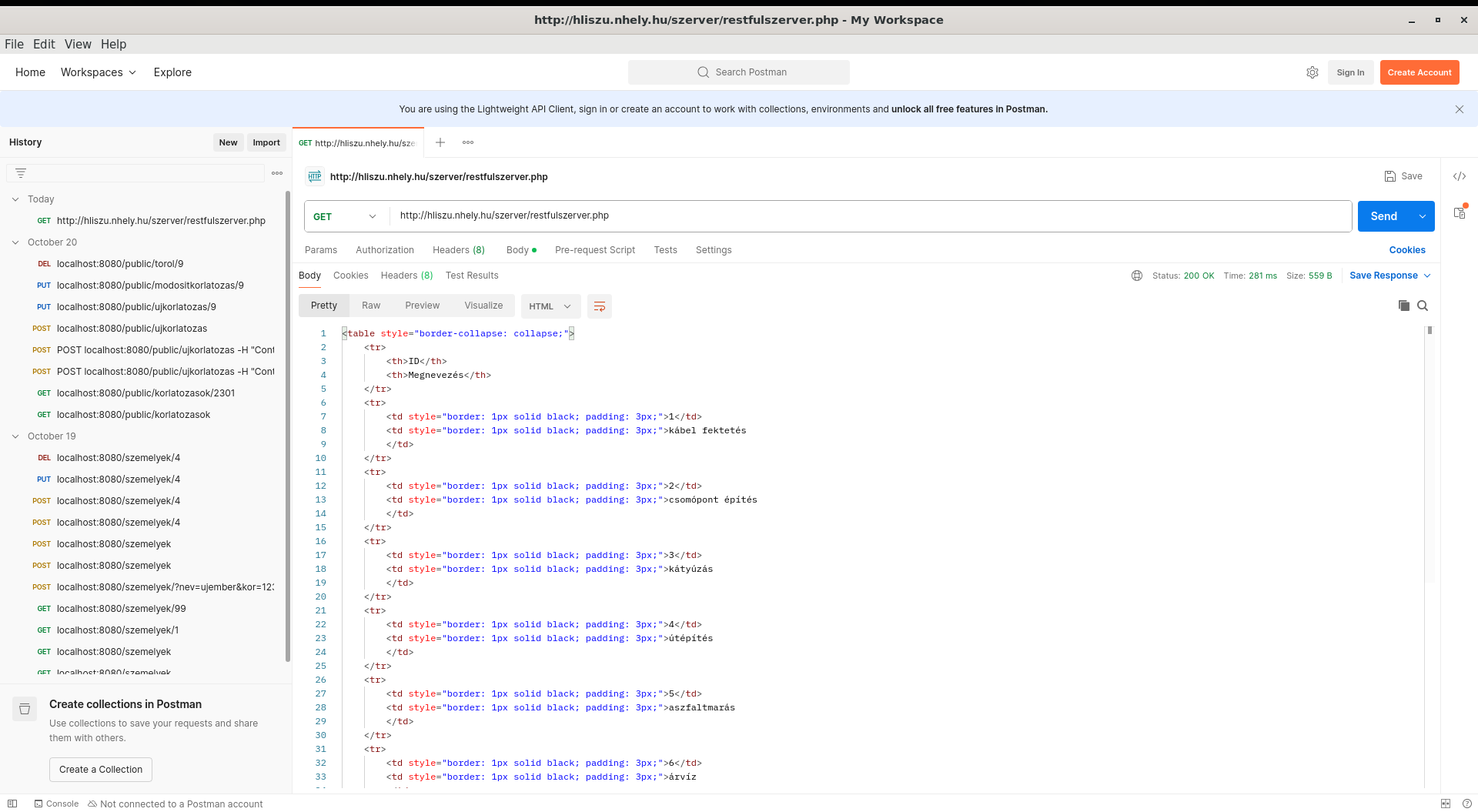
GET

A get HTML formátumban adja vissza az adatokat.

CURL: curl http://hliszu.nhely.hu/szerver/restfulszerver.php



Postman:

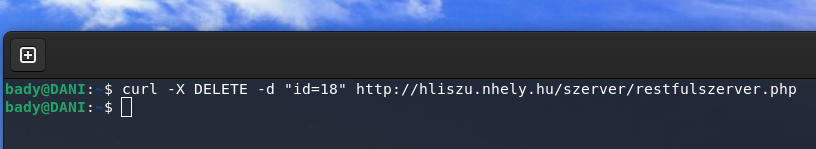


DELETE:

Használt cURL parancs:

curl -X DELETE -d "id=19" http://localhost/web2/szerver/restfulszerver.php

Ezzel egy próbaként létrehozott 19-es Id-jű megnevezés rekord törölve lett.

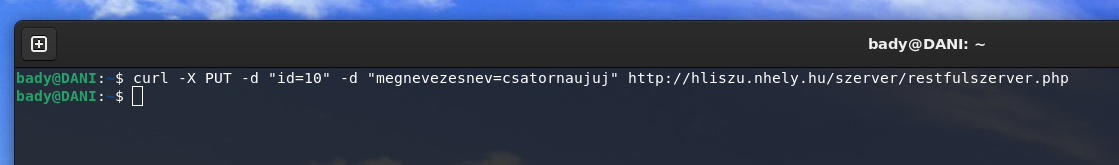


PUT:

Használt cURL parancs:

curl -X PUT -d "id=10" -d "megnevezesnev=csatornaujuj" http://hliszu.nhely.hu/szerver/restfulszerver.php

Ezzel a 10es ID-jű elemet nevezzük át.



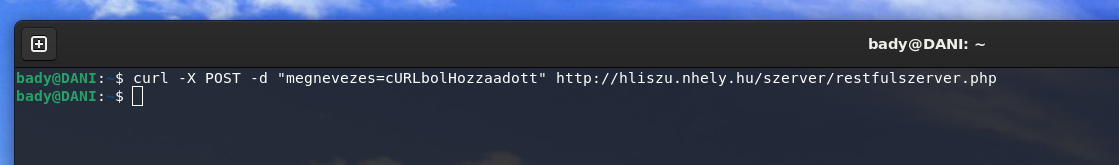
Utána a weboldalon is ellenőrizve:



POST

Használt parancs:

curl -X POST -d "megnevezes=cURLbolHozzaadott" http://hliszu.nhely.hu/szerver/restfulszerver.php



A weboldalon ellenőrizve az elem hozzáadásra került.

