A képen szöveg látható

Automatikusan generált leírás

Web programozás II. beadandó feladat

Holyba Attila Patrik (Beantech0)

FLX1D8

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

*Github:* [*https://github.com/Beantech0/web2*](https://github.com/Beantech0/web2)

Fekete István (Fekete István)

I99I9G

**2024.12.01.**

Tartalom

[Téma 3](#_Toc184065228)

[A weboldal bemutatása 3](#_Toc184065229)

[Feladatok 4](#_Toc184065230)

[Reszponzív téma 4](#_Toc184065231)

[Regisztráció, Bejelentkezés 3 szerepkör 4](#_Toc184065232)

[Soap-szerver 7](#_Toc184065233)

[Soap-kliens 7](#_Toc184065234)

[Soap-MNB 10](#_Toc184065235)

[Restful szerver-kliens menü 12](#_Toc184065236)

[GET 12](#_Toc184065237)

[PUT 13](#_Toc184065238)

[POST 14](#_Toc184065239)

[DELETE 15](#_Toc184065240)

[Publikus weboldal adatai 15](#_Toc184065241)

[Adatok 15](#_Toc184065242)

[Felhasználók 15](#_Toc184065243)

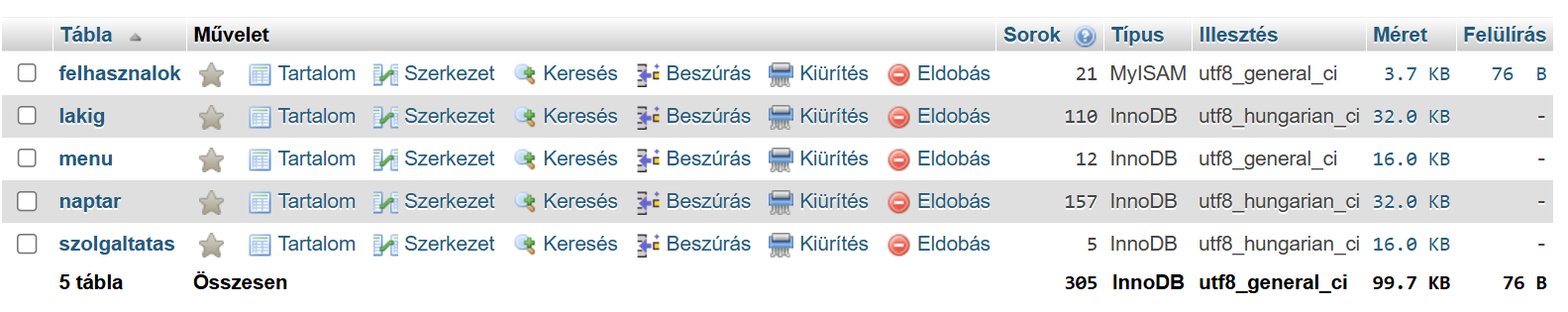
# Téma

A választott adatbázis: Hulladékszállítás.

Ez az adatbázis hulladékszállítással kapcsolatos hulladék típusokat és elszállítási időpontokat tartalmaz.

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

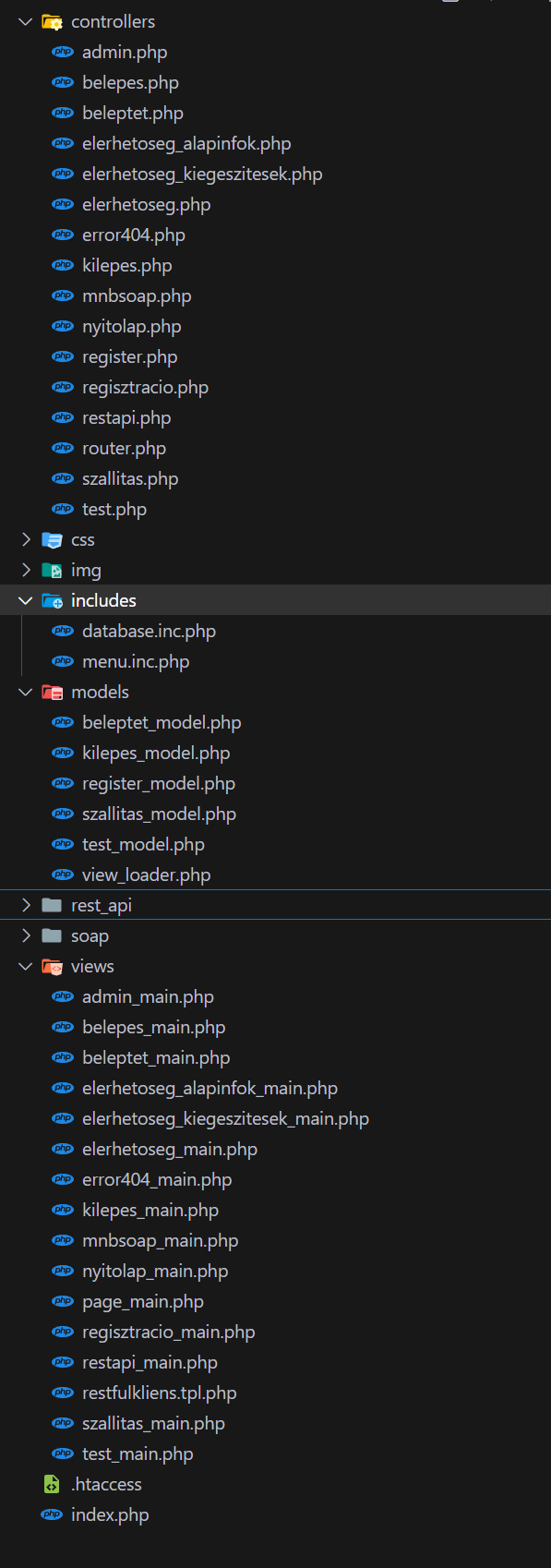
A menü tábla tárolja a megjeleníthető menük adatait és jogköreit.



# A weboldal bemutatása

A feladatot az órán tanult módszerrel és az alapján készítettük el MVC architektúrát használva.

A menüpontok a “menu” táblában vannak kezelve.



# Feladatok

## Reszponzív téma

Boostrapet használtam responsive frontend gyanánt ami jól alkalmazkodik mind asztali mind mobilos nézethez.



## Regisztráció, Bejelentkezés 3 szerepkör

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

A vendég tudja használni az MNB Soap funkciót de a REST API klienst nem tudja használni és a Szállítás Soap modult sem tudja elérni. 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 |
| adminadmin | adminadmin | Admin |
| useruser | useruser | Felhasználó |

**

A jogosultság az órán használt megoldáson alapul ez az adatbázisban a jogosultság mezőben látszódik. A 3 számból az első jelenti a vendéget a 2. a usert, a 3. pedig az admint.

Ha 1-es van akkor elérhető az oldal ha 0 akkor nem érhető el az adott felhasználónak. A szulo jelenti azt hogy ha megvan adva valami akkor nem fő menu hanem almenüje egy másik menünek.

Szerver oldalon ez az includes/menu.inc.php -> getMenu függvényben van kezelve, ami bejelentkezés hatására újból lefut és frissiti a menüsort.

A regisztráció a regisztráció menüpontra kattintva tölt be, ilyenkor egy form jön be ahol meg kell adni minden adatot és ki kell választani a kivánt jogosultsági szintet a könnyebb regisztráció érdekében.

**

A form frontend oldalon is validálva van hogy minden ki legyen töltve, ezután a views átadja Regisztracio.php-ben levő Regisztráló\_Controller átadja a register\_model-nek az adatokat ami ellőrzi hogy nincs-e ilyen felhasználó és ha nincs akkor insertálja a usert az adatbázisba. Amennyiben hiba van akkor is visszatér a regisztráció\_main.php oldalra ahol kiirja hogy vagy sikeres a regisztráció vagy pedig nem, ezután lehet bejelentkezni.

Belépés után a menü változik a jogosultság alapján és jobboldalt felül kiírásra kerül a bejelentkezett felhasználó neve. Továbbá megjeleník egy Kilépés gomb is amivel kilehet jelentkezni az oldalról. Ilyenkor a $\_SESSION-ök törölve lesznek.

**

## Soap-szerver

A weboldal képes soap kéréseket kiszolgálni.

A soap WSDL fájl generálását a hulladekszallitas.php alapján generáltattam le az órán tanult módszerrel.

<?php

    //error\_reporting(0);

    require 'hulladekszallitas.php';

    require 'WSDLDocument/WSDLDocument.php';

    $wsdl = new WSDLDocument('Hulladekszallitas', "http://localhost/web2/soap/szerver/szerver.php", "http://localhost/web2/soap/szerver/");

    $wsdl->formatOutput = true;

    $wsdlfile = $wsdl->saveXML();

    echo $wsdlfile;

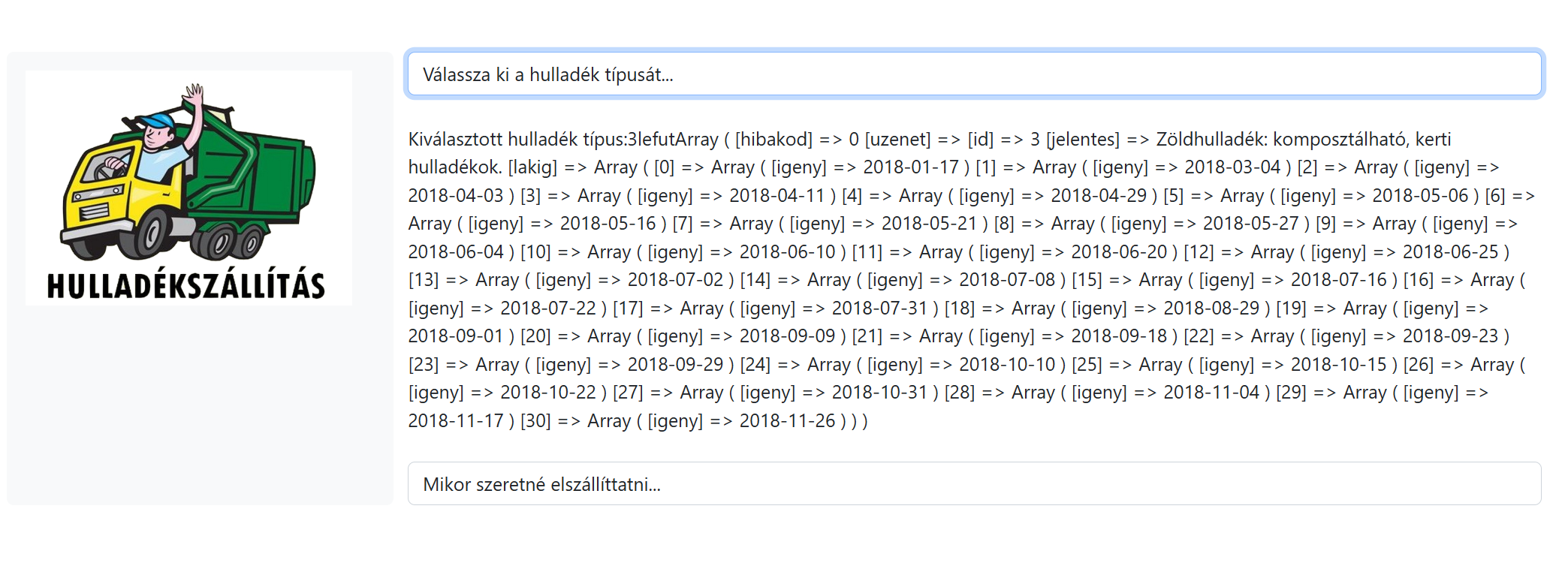
    file\_put\_contents ("hulladekszallitas.wsdl" , $wsdlfile);

?>

## Soap-kliens

A készült egy kliens a tesztelésre az oldalon ez a szállítás menüpont alatt érhető el.

Ahol lenyiló menüből kilehet választani a hulladék típusokat és utána kilistázza a kiválasztott típushoz az elérhető időpontokat a következő lenyiló menübe.



A kliens létrahozása a wsdl fájlból müködik.

$client = new SoapClient('http://localhost/web2/soap/szerver/hulladekszallitas.wsdl');

A szerver teszteléséhez készült egy külön kliens.php is ami listázza metódusokat.

<http://localhost/web2/soap/kliens/teszt.php>

Array

(

[hibakod] => 0

[uzenet] =>

[szolgaltatas] => Array

(

[0] => Array

(

[id] => 1

[jelentes] => Műanyag hulladék: PET palack, kozmetikai flakonok(PP+HDPE), szatyor, zacskó.

)

[1] => Array

(

[id] => 2

[jelentes] => Üveg hulladék: színes és fehér üveg.

)

[2] => Array

(

[id] => 3

[jelentes] => Zöldhulladék: komposztálható, kerti hulladékok.

)

[3] => Array

(

[id] => 4

[jelentes] => Papírhulladékok: újságok, könyvek, kartondobozok.

)

[4] => Array

(

[id] => 5

[jelentes] => Kommunális hulladék: szilárd, a lakókörnyezetünkben található, nem lebomló, nem veszélyes hulladék.

)

)

)

Array

(

[hibakod] => 0

[uzenet] =>

[id] => 1

[jelentes] => Műanyag hulladék: PET palack, kozmetikai flakonok(PP+HDPE), szatyor, zacskó.

[lakig] => Array

(

[0] => Array

(

[igeny] => 2018-01-30

)

[1] => Array

(

[igeny] => 2018-02-27

)

[2] => Array

(

[igeny] => 2018-05-22

)

[3] => Array

(

[igeny] => 2018-06-04

)

[4] => Array

(

[igeny] => 2018-07-18

)

[5] => Array

(

[igeny] => 2018-08-27

)

[6] => Array

(

[igeny] => 2018-09-10

)

[7] => Array

(

[igeny] => 2018-11-05

)

)

)

Array

(

[hibakod] => 0

[uzenet] =>

[id] => 1

[igeny] => 2018-01-30

[naptar] => Array

(

[0] => Array

(

[datum] => 2018-01-31

)

[1] => Array

(

[datum] => 2018-02-14

)

[2] => Array

(

[datum] => 2018-02-28

)

[3] => Array

(

[datum] => 2018-03-14

)

)

)

## Soap-MNB

Az MNB-Soap feladat megoldása az MBN Soap menüpont alatt érhető el, ez a rész mindenkinek elérhető bejelentkezés nélkül is.



A form baloldalán megadható 2 devize és egy datum és ez alapján lekéri a és kiiratja az akkor árfolyamot.

Ennek megoldása az mbnsoap\_main.php fájlban található.

Ami akkor kéri le az adatokat ha a $\_POST[‘maiarfolyam’] létezik.

if (isset($\_POST['maiarfolyam'])) {

    $devizak = $\_POST['egyik'] . "," . $\_POST['masik'];

    $objClient = new SoapClient("http://www.mnb.hu/arfolyamok.asmx?WSDL", array('trace' => true));

    $currrates = $objClient->GetExchangeRates(array('startDate' => $\_POST['datum'], 'endDate' => $\_POST['datum'], 'currencyNames' => $devizak))->GetExchangeRatesResult;

    $dom\_root = new DOMDocument();

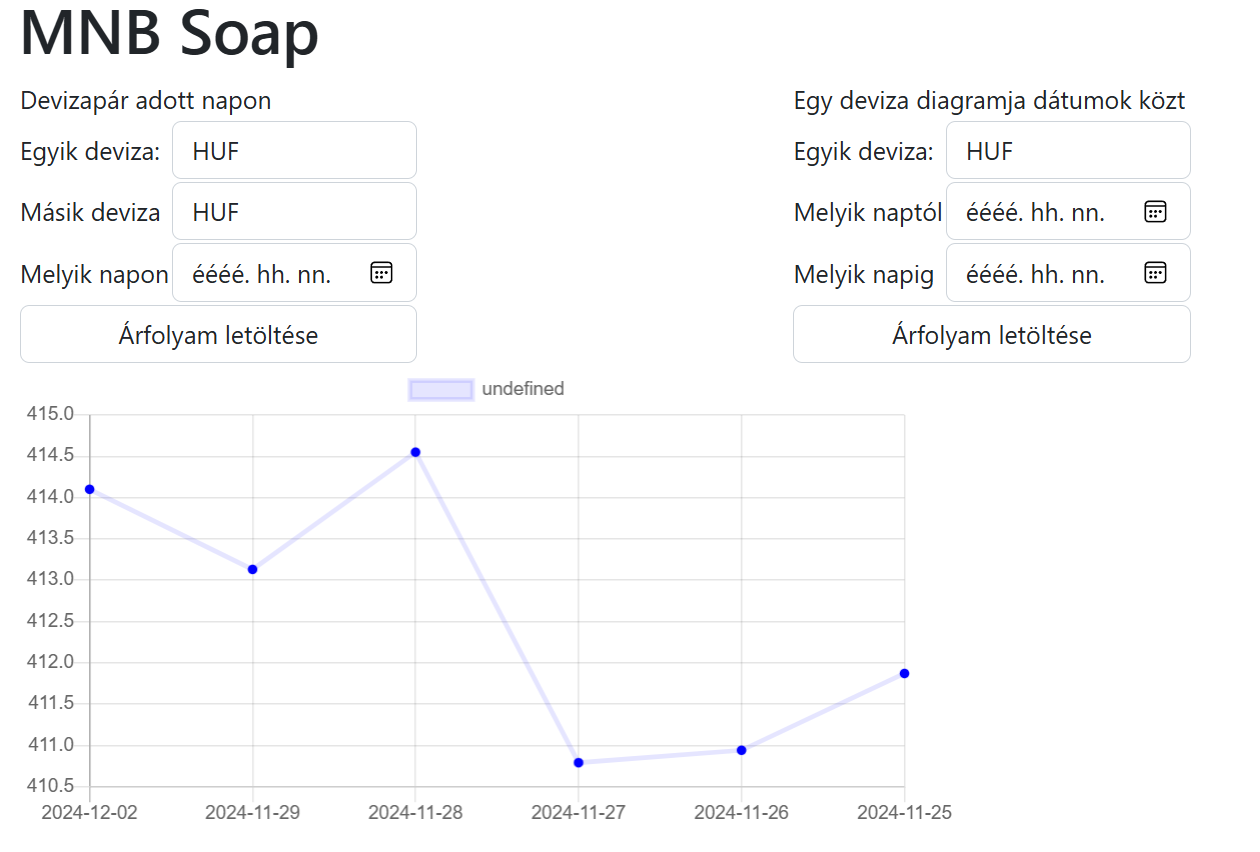
    $dom\_root->loadXML($currrates);

A grafikonhoz a javasolt chart.js libraryt használtam.

<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.4/Chart.js"></script>

Itt a form kitöltése után lekéri SOAP-on keresztül a az adatokat a 2 dátum között.

Ezt pedig egy tutorial alapján a chart.js-el iratjuk ki egy grafikonon.



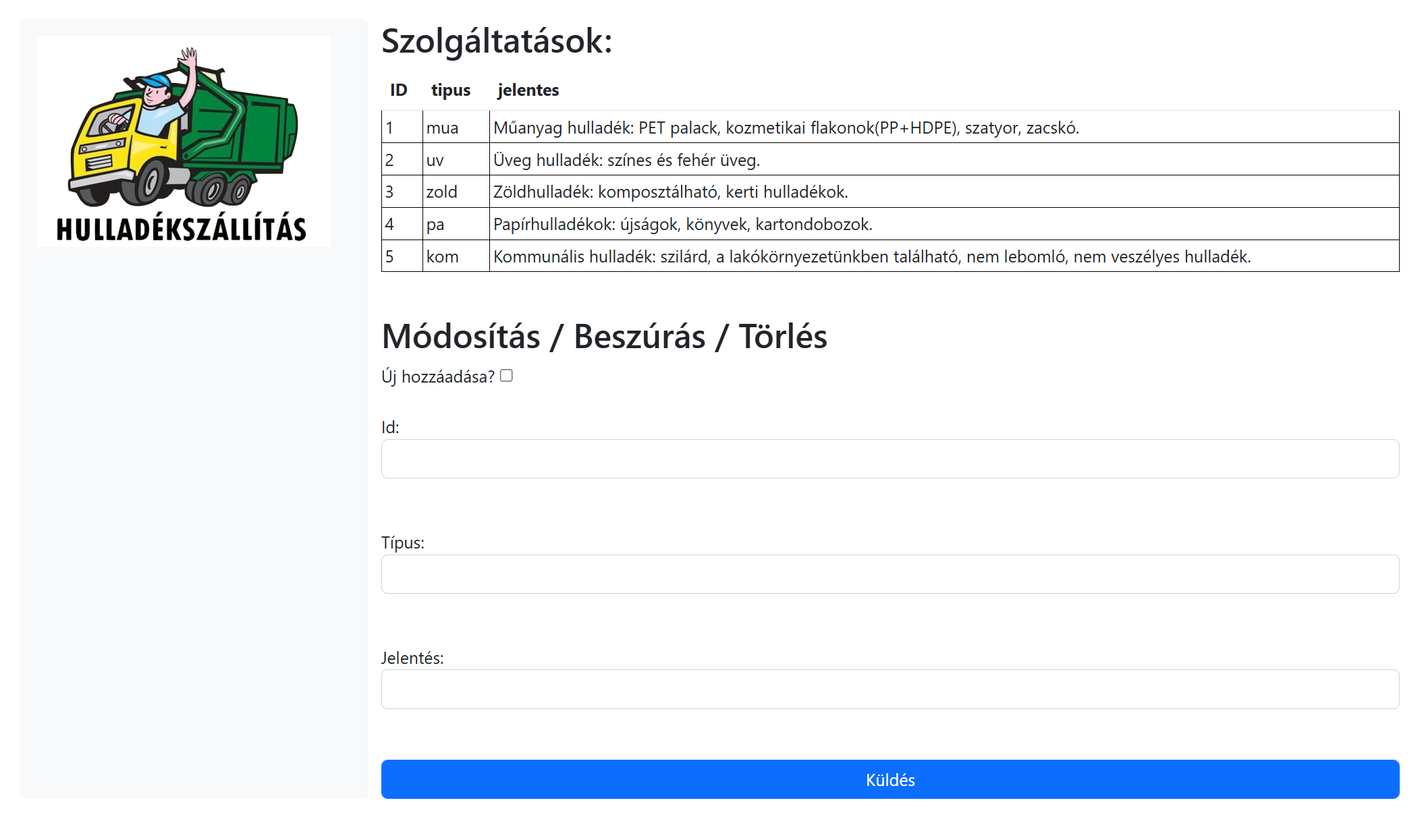
Mind a két formhoz a devizaértékek is Soap lekérdezéssel lettek lekérdezve az MNBtől valós időben.

## Restful szerver-kliens menü

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

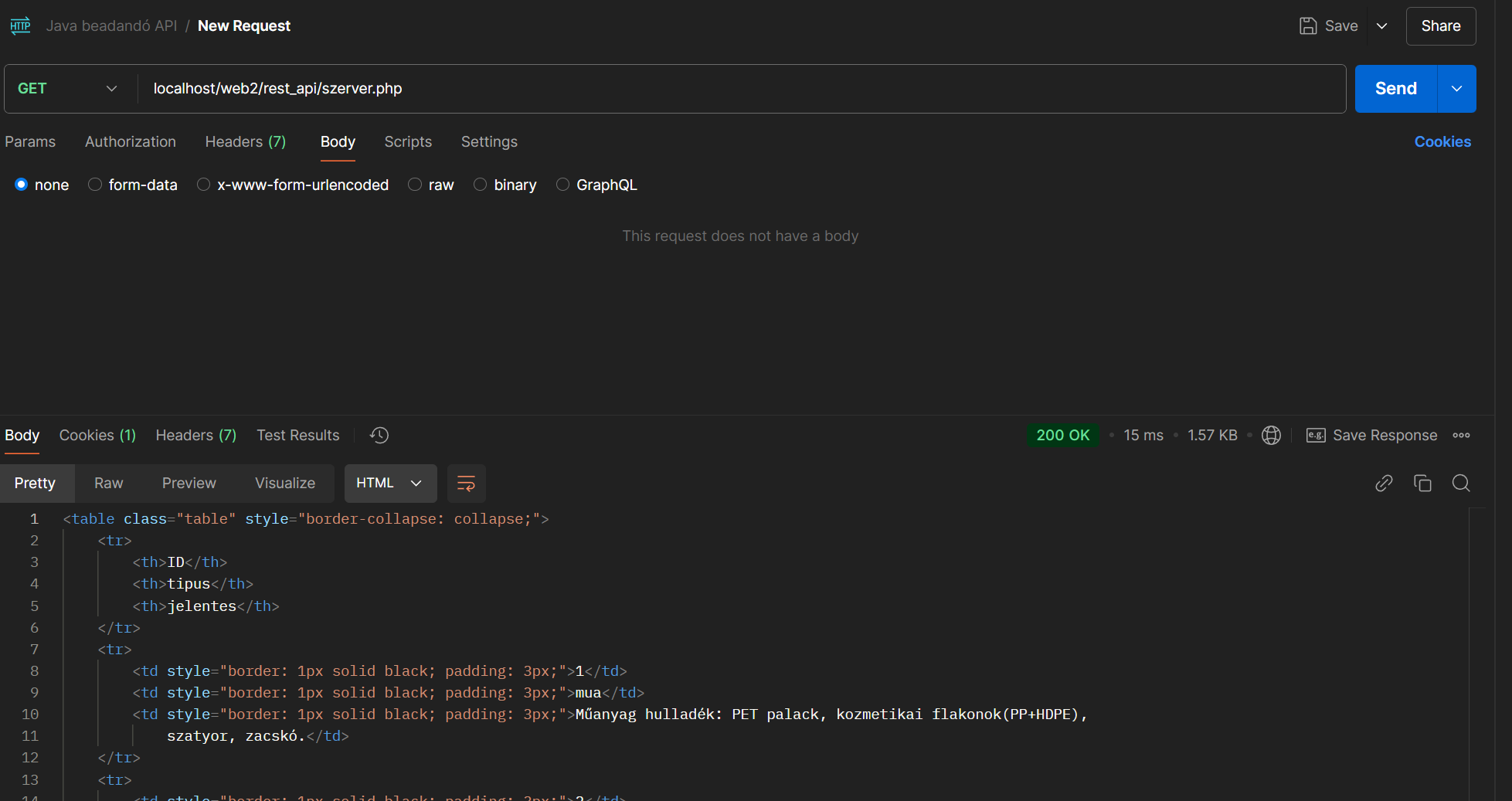
Az alap felépítése az órán mutatott módszer alapján müködik.

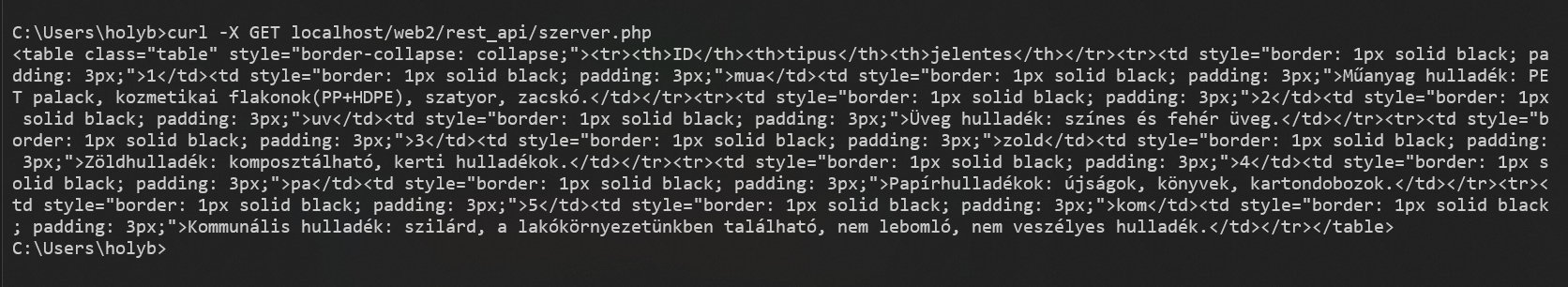
A/rest\_api/szerver.php-n van switch case-el megoldva hogy attól függően hogy GET,POST,PUT,DELETE adat érkezik be a kliensről mi történjen.



## GET

GET method esetén kilistázza a szolgaltatasok táblában levő adatokat html formátumban.





A weboldalon is lehet tesztelni a rest API müködését, ez a REST API kliens menüpont alatt érhető el ami a restapi\_main view-ből indul ki. A GET esetében alapból megjeleniti a listát a tábla tartalmával.



## PUT

Esetében az alábbi rész fut le.

case "PUT":

                $data = array();

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

                parse\_str($incoming, $data);

                $modositando = "id=id"; $params = Array(":id"=>$data["id"]);

                if($data['tipus'] != "") {$modositando .= ", tipus = :tipus"; $params[":tipus"] = $data["tipus"];}

                if($data['jelentes'] != "") {$modositando .= ", jelentes = :jelentes"; $params[":jelentes"] = $data["jelentes"];}

                $sql = "update szolgaltatas set ".$modositando." where id=:id";

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

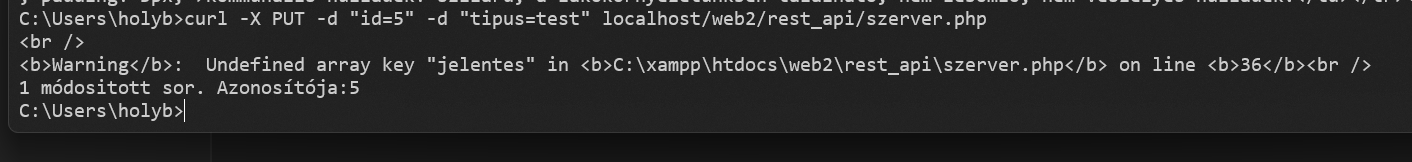
                $count = $sth->execute($params);

                $eredmeny .= $count." módositott sor. Azonosítója:".$data["id"];

Ilyenkor az ID alapján updateli az adott rekordot.

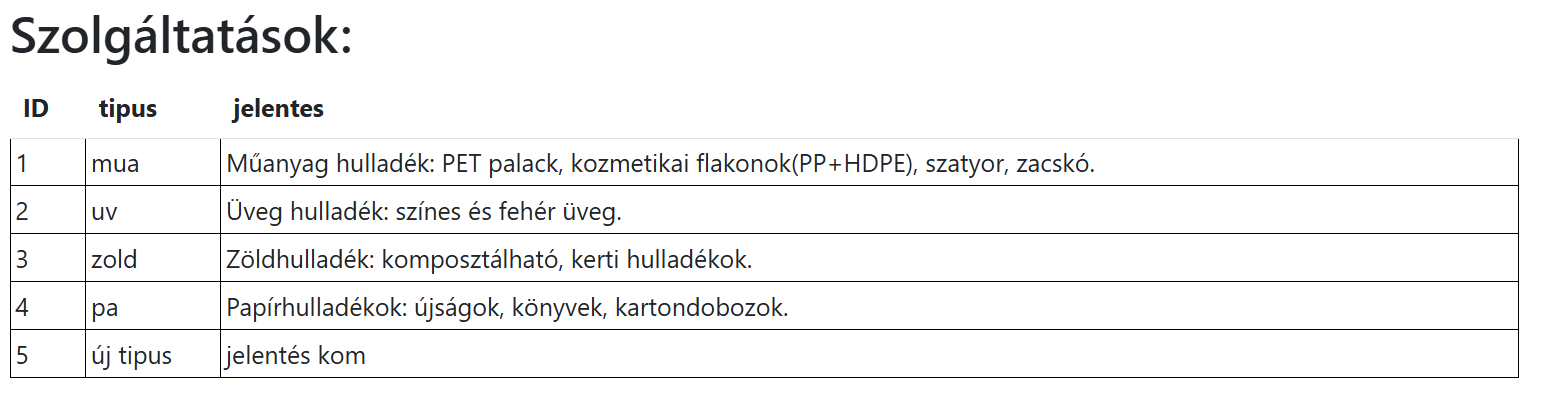
PUT, POST, DELETE esetén sajnos nem müködött Postman-el.

Tesztelésem alapján a Postman mindenképpen ezeknél a metódusoknál JSON-t akar átadni viszont a PHP-ben ez nem tetszik a parse\_str függvénynek, ha JSON\_encode-ot használtam akkor postmanből és CURL-ből jó volt viszont az oldalról nem, így visszaírtam hogy cURL-ből és az oldalról jó legyen, de teszteltem Postmanel is.



Az oldalon levő tesztelés úgy müködik módosítás esetén hogy meg kell adni mindegyik mezőnél az adatot de az “új hozzáadása” checkboxot nem kell bepipálni.

Módosítás után módosult a tábla tartalma.

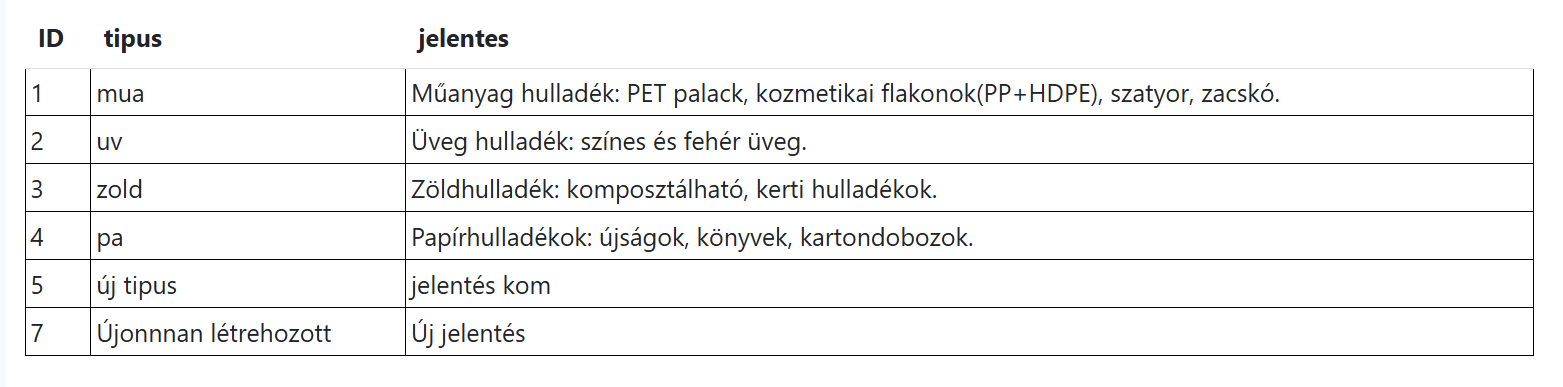


## POST

Akkor fut le minden adat megvan adva, az ID is mert a szolgaltatasok tábla nem auto increment és be van pipálva az “Új hozzáadása” menüpont.

CURL-ben tesztelve az alábbi kód hozz létre új szolgáltatást.





## DELETE

Delete method akkor fut le ha csak az ID mező van megadva és nincs bepipálva az Új hozzáadása menüpont.



# Publikus weboldal adatai

## Adatok

[holybaattila.nhely.hu](http://holybaattila.nhely.hu/)

REST API

[holybaattila.nhely.hu/rest\_api/szerver.php](http://holybaattila.nhely.hu/rest_api/szerver.php)

## Felhasználók

|  |  |  |
| --- | --- | --- |
| Felhasználónév | Jelszó | Jog |
| adminadmin | adminadmin | Admin |
| useruser | useruser | Felhasználó |