

Web-programozás II. beadandó

Dokumentáció

Github:

https://github.com/Zsotti44/web2_beadando_forint

Web:

<http://katonazsoltweb2.testhosting.hu/>

Készítették	Feladatkör
Katona Zsolt Krisztián	Fejlesztés
Sándor Adrián	Fejlesztés

Lektorálta
-

Dokumentum azonosító:	WEBII-D-1
Verzió:	0.01
Dátum:	2024.11.26

© Minden jog fenntartva!

Tartalomjegyzék

1. Bevezetés.....	5
1.1. Dokumentum célja	5
1.2. Választott projekt leírása.....	5
1.3. Feladatok	5
2. Teszt felhasználói adatok	7
3. Fejlesztési környezet	7
3.1. Verziókövetés: GitHub.....	7
4. Megvalósítás.....	9
Adatbázis	9
Adatbázis táblák.....	9
anyag tábla:	9
tervezo tábla:.....	10
erme tábla:.....	10
akod tábla:	10
tkod tábla:.....	11
REST végpontok	12
Végpontok	12
/api/anyag GET végpont	13
/api/tervezo GET végpont	13
/api/akod GET végpont	13
/api/tkod GET végpont	13
/api/erme végpontok	14
MNB SOAP kliens	16
Design.....	17
Felhasználók kezelése	18
Menük	18
Oldalak	20
Főoldal.....	20
SOAP Szolgáltatás	21
MNB Adatszolgáltatás	22
MNB Adatszolgáltatás	22
Napi árfolyam	22
Devizák havi árfolyamai.....	23

Devizapárok napi árfolyama	24
Authentikációs oldalak	24
Rest szolgáltatás	25
Admin / Menü.....	26
PDF	27

1. Bevezetés

1.1. Dokumentum célja

Jelen leírás a(z) „L-EA01 WEB-programozás II. [L-K-GINFBAL-WEBPROG2-1-EA01] [CLS 2024/25/1]” elnevezésű kurzus, Sándor Adrián és Katona Zsolt által fejlesztés alatt álló számítástechnikai projekt, weboldal fejlesztői környezetét, koncepcióját, design elemeinek tervezett megvalósítását és fejlesztési tevékenységeit ismerteti.

1.2. Választott projekt leírása

A weboldal a magyar forint érméinek részletes adatait mutatja be, különös tekintettel a címletekre, a kiadott darabszámokra, a tervezőkre és az érme összetételére. Az oldal célja, hogy rendszerezett formában nyújtson információkat a forint érmékről, segítve ezzel a numizmatika iránt érdeklődőket, a gyűjtőket vagy akár a szakmai kutatókat. Az adatokat átlátható módon, könnyen navigálható felületen keresztül érhetik el a látogatók.

A menürendszer hierarchikus felépítése és a dinamikus funkciók biztosítják, hogy a felhasználók egyszerűen hozzáférhessenek az érmeikkel kapcsolatos specifikus információkhoz, például a kibocsátási dátumokhoz vagy az érme tervezőjéhez. A tervezés során kiemelt figyelmet kaptak az érme gyűjtés szempontjából releváns adatok, például az érme anyagának összetétele (arany, ezüst stb.) és az érme történeti háttere.

Ez a weboldal nemcsak az érme műszaki és történeti adatait rendszerezi, hanem betekintést nyújt a magyar forint fejlődéstörténetébe is, hozzájárulva ezzel a pénztörténet iránti érdeklődés mélyítéséhez. Az egyetemi beadandó szempontjából remek példát nyújt arra, hogyan lehet adatbázis-alapú információkat vizuálisan és strukturáltan megjeleníteni egy modern webes környezetben.

1.3. Feladatok

1. feladat (Kötelező elem!)

a, a menüpontok neveit és a menüpontokhoz tartozó oldalak azonosítóját az adatbázisban tárolja, Megoldás: [Menük](#)

b, többszintes menürendszert megvalósít Megoldás: [Menük](#)

c, Regisztráció, Bejelentkezés: legalább „látogató”, „regisztrált látogató” és „admin” szerepkör el van különítve. Megoldás: [Felhasználók kezelése](#)

d, Az első oldalon mutassa be a témát. Megoldás: [Főoldal](#)

2. Keressen és alkalmazzon egy **ingyenes responzív témát** az oldalaihoz. A dokumentációban írja le, hogy melyik témát választotta.

Ötleteket talál a dokumentum végén.

Megoldás: [Téma \(Stocker\)](#), [Design](#)

3. **SOAP-szerver menü:** Készítsen egy Restful webszolgáltatást, Az adatbázis minden tábláját használja fel a feladathoz. Megoldás: [Rest szolgáltatás](#)

4. **SOAP-kliens menü:** Készítsen a SOAP webszolgáltatáshoz egy klienst, amivel tudja tesztelni a szolgáltatást Megoldás: [SOAP Szolgáltatás](#)

5. **SOAP-MNB:** Egy oldalon használja fel a Magyar Nemzeti Bank SOAP adatszolgáltatását, és valósítson meg több lekérdezési lehetőséget is az adatokból.

Lehessen keresni az alábbiakra:

a, Egy adott devizapár (pl. Eur-Huf, Eur-Usd, ...) adott napján mennyi volt az árfolyam?

Megoldás: [Devizapárok napi árfolyama](#)

b, Egy adott devizapár egy adott hónapjában minden napra mennyi volt az árfolyam táblázatban kiírva. Valamint az adatok megjelenítése grafikonon is. Grafikonhoz javasolt a <https://www.chartjs.org/> használata. Minta: <https://www.mnb.hu/arfolyam-lekerdezes>

Megoldás: [Devizák havi árfolyamai](#)

6. **Restful-szerver menü:** Készítsen egy Restful webszolgáltatást az adatbázisának egyik táblájához. Valósítsa meg a GET, POST, PUT, DELETE funkciókat. Tesztelje a webszolgáltatást cURL-el és Postman-el. A dokumentációban mutassa be a tesztelés lépéseit.
Megoldás: [REST végpontok](#)
7. **Restful-kliens menü:** Készítsen a Restful webszolgáltatáshoz egy klienst, amivel tudja tesztelni a szolgáltatást (GET, POST, PUT, DELETE funkciók). Megoldás: [Rest szolgáltatás](#)
8. **PDF menü:** Készítsen egy PDF készítő szolgáltatást a **TCPDF** segítségével. A felhasználó 3 szöveges beviteli mezőben, vagy lenyíló listában megad adatokat és a rendszer az adatbázisból olvasva készít egy letölthető PDF fájlt. Az adatbázis három tábláját használja fel.
Megoldás: [PDF](#)
9. Alkalmazását tölts fel és valósítsa meg Internetes tárhelyen is.
(Kötelező elem! A működés ez alapján lesz javítva) Bármelyik tárhely-szolgáltatót használhatja. **URL címben szerepeljen a csoport egyik tagjának a neve.**
Megoldás: <http://katonazsoltweb2.testhosting.hu/>
10. Használják a GitHub (github.com) verziókövető rendszert.
(Kötelező elem! A forrás ez alapján lesz javítva)
Ne csak a kész alkalmazást töltsék fel egy lépésben, hanem a részállapotokat is még legalább 5 lépésben személyenként.
A GIT-en saját nevet válasszanak, ami alapján be lehet azonosítani, hogy ki mit töltött fel.
Megoldás: https://github.com/Zsolti44/web2_beadando_forint
11. A **GitHub-on** a **projektmunka módszert** alkalmazzák
12. Készítsen egy legalább 15 oldalas dokumentációt képernyőképekkel (**Kötelező elem!**), amiben bemutatja alkalmazását és leírja, hogy az előző pontok feladatait hogyan valósította meg.
Fontos, hogy ez utóbbit leírja a dokumentációban, mert a feladat ez alapján lesz javítva. A dokumentációban adja meg a tárhely és a GitHub projektjének URL címét is és az Internetes tárhelyen az **admin és egy másik felhasználó belépési adatait.**

1.4. Feladatok felosztása

A feladatok felosztása nehéz feladat, mind a ketten kivettük a részünket a legtöbb feladatból.

Katona Zsolt:

- Projekt alapjainak létrehozása
- Design
- Téma bemutatása
- Authentikáció / felhasználókezelés
- Menürendszer
- PDF Generálás
- Havi deviza grafikon
- MNB SOAP alapok
- Deploy

Sándor Adrián:

- SOAP Kliens és szerver létrehozása
- MNB SOAP Specifikus feladatok megvalósítása
- REST Szolgáltatások teljes kivitelezése
- Adatbázis
- Modellek megvalósítása
- Routeok megvalósítása

2. Teszt felhasználói adatok

Regisztrált vendég:

Felhasználónév: user

Jelszó: user

Adminisztrátor:

Felhasználónév: admin

Jelszó: admin

3. Fejlesztési környezet

3.1. Verziókövetés: GitHub

A GitHub, mint vezető szoftverfejlesztési platform, lehetővé teszi a csapat számára, hogy hatékonyan kezelje a kódbázist, nyomon követhesse a változásokat, és elősegítse a kollaboratív munkát azáltal, hogy egy központosított helyen tárolja a projekt fájljait és dokumentációját. Az integrált verziókezelési rendszer előnyei közé tartozik a fejlesztési folyamatok nagyfokú automatizálása, a visszaállíthatóság, valamint a különböző fejlesztési ágak (branches) kezelése, ami lehetővé teszi a csapat számára, hogy párhuzamosan dolgozzon különböző funkciókon. A GitHub használatával a fejlesztőcsapat kihasználhatja a kódmegosztás és a közösségi hozzájárulás előnyeit is, így gyorsítva a fejlesztési ciklusokat és javítva a projekt átláthatóságát és nyomon követhetőségét.

3.2. Felhasznált technológiák

3.2.1. Bootstrap: Egy ingyenes és nyílt forrású front-end keretrendszer, amely elősegíti a gyors és egyszerű reszponzív webdesign kialakítását. Készen álló HTML és CSS sablonokat, valamint JavaScript komponenseket kínál, amelyekkel könnyedén lehet készíteni felhasználóbarát és esztétikailag vonzó weboldalakat.

3.2.2. Reszponzív webdesign: Az a tervezési és fejlesztési módszertan, amelynek célja, hogy a weboldalak minden eszközön - asztali számítógéptől kezdve a mobiltelefonokig - jól nézzenek ki és jól működjenek. A Bootstrap keretrendszer kiválóan támogatja a reszponzív webdesign kialakítását.

3.2.3. Laravel:

A Laravel egy modern PHP keretrendszer, amely az egyszerűsége és az eleganciára épít. MVC (Model-View-Controller) architektúrát követ, ami megkönnyíti az alkalmazások fejlesztését és karbantartását. Számos beépített funkcióval rendelkezik, például routing, ORM (Eloquent), beépített felhasználói autentikáció, csomagkezelés (Composer), valamint Blade sablonmotor. A Laravel népszerűsége főként annak köszönhető, hogy gyorsan lehet vele skálázható és biztonságos webes alkalmazásokat fejleszteni.

3.2.4.Laravel Auth

A Laravel beépített autentikációs rendszere egyszerűsíti a felhasználók hitelesítését és azonosítását. Az artisan parancssor segítségével gyorsan generálható login, regisztráció, jelszóemlékeztető és e-mail-verifikáció funkció. A Laravel Auth támogatja a session-alapú hitelesítést, az API tokeneket (pl. Passport vagy Sanctum segítségével), és testreszabható middleware-k segítségével biztosítja a jogosultságkezelést. A Laravel 10-től kezdve a Breeze és a Fortify csomagok is elérhetők, amelyek egyszerű és testreszabható megoldásokat kínálnak.

3.2.5.Livewire

A Livewire egy Laravel-specifikus front-end keretrendszer, amely lehetővé teszi, hogy dinamikus felhasználói felületeket hozz létre JavaScript írása nélkül. A Livewire mögötti alapötlet az, hogy a komponenseket PHP-ben írod meg, és a háttérben AJAX hívásokkal frissíti az oldal egyes részeit. Ez különösen hasznos, ha interaktív elemeket, például űrlapokat vagy valós idejű adatmódosításokat szeretnél létrehozni. A Livewire egyszerűen integrálható a Blade sablonmotorral, és remekül működik együtt más Laravel-funkciókkal.

3.3. Szoftverek és futtatókörnyezet

3.3.1.XAMPP

A XAMPP egy ingyenes, multiplatform szervercsomag, amely az Apache webszervert, a MySQL vagy MariaDB adatbázist, valamint a PHP és Perl nyelvek futtatókörnyezetét tartalmazza. A XAMPP célja, hogy egyszerűsítse a fejlesztési környezet beállítását, így gyorsan tesztelheted és fejlesztheted webalkalmazásaidat. Könnyen telepíthető, és tartalmaz egy vezérlőpanelt, ahol a szervereket (pl. Apache, MySQL) egyszerűen elindíthatod vagy leállíthatod.

3.3.2.phpMyAdmin

A phpMyAdmin egy nyílt forráskódú webes felület, amely lehetővé teszi a MySQL és MariaDB adatbázisok kezelését. Ezzel az eszközzel könnyen létrehozhatasz, módosíthatsz vagy törölhetsz adatbázisokat, táblákat, mezőket, valamint futathatsz SQL lekérdezéseket. A phpMyAdmin egyszerűsíti az adatbázis-kezelést a grafikus felhasználói felülete révén, így nem szükséges mély SQL-ismeret az alapműveletek elvégzéséhez.

3.3.3.IntelliJ PHPStorm

A PHPStorm egy professzionális fejlesztőkörnyezet (IDE), amely a JetBrains által lett kifejlesztve, és kifejezetten PHP fejlesztésre optimalizálták. Számos fejlett funkcióval

rendelkezik, például kódkitöltés, hibakeresés (debugging), verziókövető rendszerek (pl. Git) integráció, valamint beépített támogatás a modern keretrendszerekhez (pl. Laravel, Symfony). A PHPStorm további előnye, hogy kiválóan együttműködik a front-end technológiákkal (HTML, CSS, JavaScript), valamint a PHPUnit teszteléssel.

4. Megvalósítás

Adatbázis

Az Adatbázis a weboldalunk alapvető és kritikus eleme, amely két fő célra használatos. Elsődlegesen szolgál a felhasználók hitelesítésére, ugyanis itt tároljuk azokat az adatokat, amelyek az azonosításhoz és az egyedi felhasználói profilok létrehozásához szükségesek, mint a felhasználónevek, jelszavak, jogosultságok és id. Másrészt az adatbázis tartalmazza a feladat eredetileg meghatározott tábláit és az importálást követően a letöltött adatokat.


A felhasználóknak lehetőségük van üzenetet küldeni az üzemeltetőnek. Ezek az üzenetek szintén az adatbázisban vannak tárolva. Egy-egy rekordon, a felhasználó azonosítójával megállapítható ki küldte az üzenetet.

Miután bejelentkezés nélkül is van lehetőség üzenetküldésre, ilyenkor pedig nincs azonosítható felhasználó, null értékkel kerülnek rögzítésre az üzenetek, megjelenítéskor ezek a rekordok a „Vendég” megnevezést kapják.

Adatbázis táblák


anyag tábla:

A tábla a korlátozott darabszámban kiadott magyar forint érmékhez használt nyersanyagokat tartalmazza. Minden rekord egy konkrét fémet reprezentál. A femid mező, a tábla elsődleges kulcsa, mely azonosítja a fémeket.

Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett
femid 	int(4)			Nem	Nincs
femnev	varchar(16)	utf8_general_ci		Igen	NULL


tervezo tábla:

A tervező tábla a forint érték tervezőinek adatait tárolja. Minden sor egy konkrét tervezőt reprezentál, beleértve a tervező egyedi azonosítóját és nevét. Ez a tábla a tervezők azonosítására szolgál, amely lehetővé teszi az egyes érték készítőinek nyomon követését.

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett
1	tid 	int(1)			Nem	Nincs
2	nev	varchar(32)	utf8_general_ci		Igen	NULL

erme tábla:

Az erme tábla a limitált darabszámban kiadott magyar forint érték alapvető adatait tárolja. Minden sor egy érmeadatot reprezentál, annak címletével, tömegével, kiadott darabszámával, valamint a kibocsátás és bevonás dátumával. Ez a tábla az érték részleteinek, információinak tárolására szolgál.

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett
1	ermeid 	int(4)			Nem	Nincs
2	cimlet	int(6)			Igen	NULL
3	tomeg	decimal(4,2)			Igen	NULL
4	darab	int(11)			Igen	NULL
5	kiadas	date			Igen	NULL
6	bevonas	date			Igen	NULL



akod tábla:

Az akod tábla egy kapcsolótábla. Az alapanyagoknak használt fémek, azaz anyag tábla és az érték adatait tartalmazó erme adatbázistábla adatainak kapcsolatait tartalmazza. A kapcsolótáblára azért volt szükség, mert az anyagok és érték kapcsolata több a többhöz, egy érmehez több alapanyagot is használhattak, valamint egy alapanyagot több érme anyagaként is felhasználtak.

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett
1	ermeid 	int(4)			Igen	NULL
2	femid 	int(4)			Igen	NULL

tkod tábla:

A tkod tábla szintén egy kapcsolótábla, ami a tervezők és érmék kapcsolatait reprezentálja. Az érmék és tervezők közt szintén több a többhöz a kapcsolat, több tervező is közreműködhetett egy érme megalkotásában, valamint egy-egy tervező több érménél is szerepet játszott.

#	Név	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett
1	ermeid 	int(4)			Igen	NULL
2	tervezoid 	int(4)			Igen	NULL

Az adatbázis elkészülte után a Laravelben a megfelelő modelleket valósítottuk meg, amelyek közvetlen kapcsolatot biztosítanak az adatbázis és az alkalmazás logikája között. Ezek segítségével könnyen kezelhetők az adatbázis műveletek, mint például az adatok lekérdezése, beszúrása, frissítése vagy törlése. A modellek reprezentálják az egyes táblákat, könnyítik az adatokkal való munkát, támogatják a kapcsolatok kezelését és ezek az üzleti logika központjai.

Az adatbázis tartalmaz a fent felsorol táblákon kívül több adattáblát is, mint például a sessions, cache táblák. Ezek nem szerepeltek az importálandó állományban, viszont a Laravel automatikusan létrehozta ezeket a megfelelő működés miatt. Ezek a táblák az alkalmazás működéséhez szükségesek és bizonyos komponensek alapértelmezett konfigurációjával együtt kerülnek létrehozásra.

REST végpontok

A REST hívások HTTP protokollon keresztül történnek. Általában négy fő műveletet használunk, GET az adatok lekérésére, POST új adatok rögzítéséhez, PUT az adatok frissítésére, módosítására szolgál, végül a DELETE végpontok adatok törlésére használtak.

A kérések törzs adata (amennyiben van), többnyire xml vagy json formában kerül küldésre, valamint jellemzően ugyan így kapja vissza a választ a kliens, kiegészülve még egy lehetőséggel, a html-el.

A REST végpontokhoz Basic autentikáció szükséges. Az oldalon regisztrációkor megadott e-mail cím és jelszó használható a végpontok elérésére.

A routes/api.php fájlban definiáltuk a Laravel alkalmazás API végpontjait, amelyek RESTful kéréseket fogadnak. Ez a fájl külön van választva a webes útvonalaktól, hogy elkülönítse az API-funkcionalitást. Itt határozzuk meg, hogy milyen HTTP metódusokat használhatnak a kliensek, és melyik controller kezelje ezeket a kéréseket.

Az API útvonalak automatikusan a /api prefixel vannak ellátva, például egy érme adatainak lekérése a GET /api/erme végponton történhet. Ez a fájl tehát az API-központú forgalom irányításának egyik legfőbb része.

Fejlesztés során Postmannel teszteltük a végpontokat, ezen kérésekről, egy importálható Postman collection elérhető a GIT repositoryban a documents mappában, a Forint_postman_collection néven, minden a következőkben felsorolt végponthoz.

Az elkészült végpontokat cUrl-el is teszteltük, itt például a 25-ös azonosítójú érme törlése látható.

```
C:\Users\Adrian>curl --location --request DELETE "http://127.0.0.1:8000/api/erme/25" ^
More? --header "Authorization: Basic YWRyaWFuQGFKcmllhbi5odToxMjM0NTY3OA=="
{"message":"\u00c9rme sikeresen t\u00f6r\u00f6lt."}
C:\Users\Adrian>
```

Végpontok

Az Adatbázis mind az 5, a feladat által meghatározott adattáblája lekérhető GET metódussal, valamint az érme táblához kapcsolódóan megvalósításra kerültek POST, PUT és DELETE végpontok is.

/api/anyag GET végpont

Az anyag tábla az érmék alapanyagaival elérhető, http GET kérést indítva a következő címen:

<http://katonazsoltweb2.testhosting.hu/api/anyag> .

Az adathalmaz szűkítésére femid query paraméter használható. A képen a válasz json formátuma látható.

```
1  [
2      {
3          "femid": 1,
4          "femnev": "Ezüst"
5      },
6      {
7          "femid": 2,
8          "femnev": "Réz"
9      },
10     ]
```

/api/tervezo GET végpont

A tervezők adatainak eléréséhez http GET kérést szükséges indítani a következő címre:

<http://katonazsoltweb2.testhosting.hu/api/tervezo> Az

adathalmaz szűkítésére a tid query paraméter használható. A képen a válasz json formátuma látható.

```
1  [
2      {
3          "tid": 1,
4          "nev": "Bognár György"
5      },
6      {
7          "tid": 2,
8          "nev": "Kósa István"
9      },
10     ]
```

/api/akod GET végpont

Az anyag-érme táblák kapcsolótáblája a <http://katonazsoltweb2.testhosting.hu/api/akod> címre indított http GET kéréssel érhető el. Az adathalmaz szűkítésére query paraméterként a femid megadható. A képen a válasz json formátuma látható.

```
1  [
2      {
3          "ermeid": 1,
4          "femid": 5
5      },
6      {
7          "ermeid": 2,
8          "femid": 5
9      },
10     ]
```

/api/tkod GET végpont

Az érme-tervező kapcsolótábla adatai szintén elérhetők, ehhez http GET kérést kell indítani a <http://katonazsoltweb2.testhosting.hu/api/tkod> címre. A tervezoid query paraméterrel szűrhető az adathalmaz. A képen a válasz json formátuma látható.

```
1  [
2      {
3          "ermeid": 1,
4          "tervezoid": 5
5      },
6      {
7          "ermeid": 2,
8          "tervezoid": 5
9      },
10     ]
```

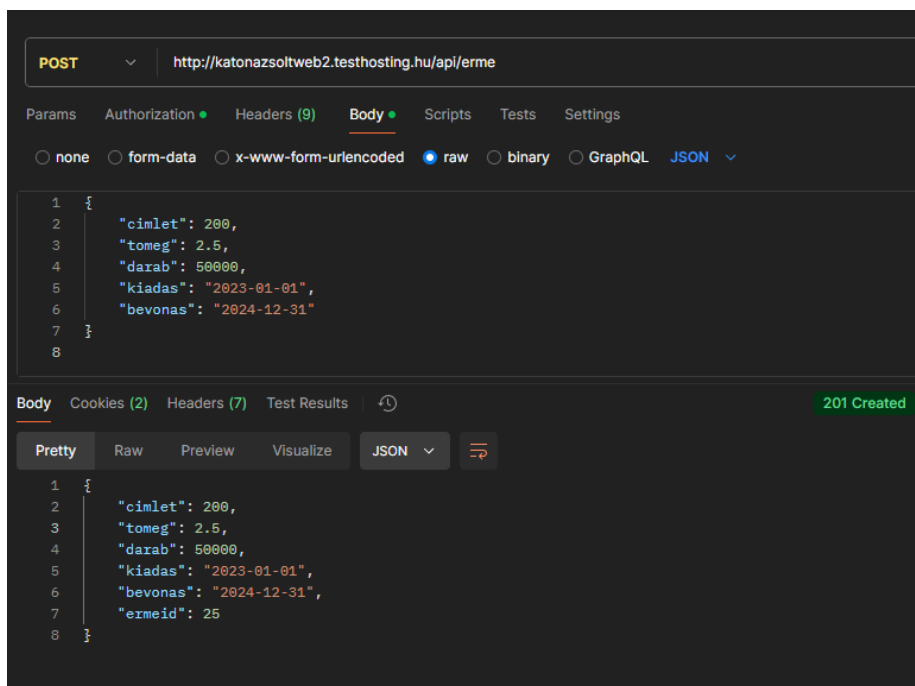
/api/erme végpontok

A program lehetőséget nyújt új érme felvételére, meglévő módosítására és akár egy érme törlésére is az adatbázisból.

Érme adatok lekérése az előző végpontokhoz hasonlóan történik. GET http kérést kell indítani a <http://katonazsoltweb2.testhosting.hu/api/erme> címre, melyet az ermeid query paraméterrel szűrhetünk. A képen a válasz json formátuma látható.

```
1 {  
2   "ermeid": 1,  
3   "cimlet": 1,  
4   "tomeg": "1.50",  
5   "darab": 227158000,  
6   "kiadas": "1946-08-01",  
7   "bevonas": "1995-06-30"  
8 }
```

Új adat rögzítéskor a használt Url nem különbözik a GET kérésnél használttól, csupán annak metódusát szükséges módosítani.



Egy ilyen kérésnél a küldött adatok vizsgálata elhagyhatatlan, így az ErmeControllerhez kapcsolódóan létrehoztunk egy ErmeRequest osztályt, mely az adatok validációs feladatait látja el. Ez ellenőrzi a bejövő kérést, megvizsgálja a mezők értékét, típusát, kötelezőségét és ha minden rendben talál, visszaadja a vezérlést az ErmeControllernek és folytatódhat az adatrögzítés, módosítás. Ellenkező esetben, a probléma okával tér vissza.

Már meglévő rekord módosításakor az url-ben szerepelnie kell az érme azonosítójának a kérést törzsében pedig a többi adatnak a POST kéréshez hasonlóan. Módosításkor az url-ben megadott id-val rendelkező érme adatait módosítjuk.

Törlés szintén lehetséges, erről korábban cUrl-t is csatoltunk. Egy érme törléséhez http DELETE kérést kell indítani a /api/erme végpontra. Amennyiben a megadott azonosító nem létezik, a kérést elutasítjuk 404-es státuszkóddal.

```
1  {  
2  |    "message": "Érme nem található."  
3  |  }
```

MNB SOAP kliens

A Laravel nem tartalmaz alapértelmezetten SOAP implementációt, miután ez a REST-nél egy régebbi, mára kevésbé népszerű protokoll. A feladat elvégzésében a Laminas keretrendszert használtuk, ami a számos összetevője közt a SOAP szerverek kezelésében is segítséget nyújt.

Az MNBService osztály egy szolgáltatásosztály, mely az MNBController-t szolgálja ki és SOAP kliens segítségével kommunikál a Magyar Nemzeti Bank (MNB) hivatalos árfolyamadatokat nyújtó webszolgáltatásával. Az osztály felelős az MNB SOAP végpontjainak eléréséért, az árfolyam adatok lekérdezéséért és azok feldolgozásáért.

A getWsd() metódus a WSDL fájl tartalmát tölti le, mely alapján létrehozható volt a többi metódus olyan célokra mint az elérhető devizanemek letöltése, a napi árfolyamok letöltése, havi árfolyamok letöltése.

Az MNBService egy robusztus SOAP kliens, amely egyszerűsíti az MNB webszolgáltatásaival való integrációt.

Az MNBController osztály a Magyar Nemzeti Bank SOAP szolgáltatásához készült kliensfunkciókat biztosítja az alkalmazás nézeti számára.

A devizanemeket több lenyíló lista is megjeleníti, melyeket adatbázisban nem tárolunk külön, de az oldal megfelelő sebessége érdekében ezeket gyorsítótárazzuk. A devizanemeket a rendszer a Laravel Cache rendszerében tárolja.

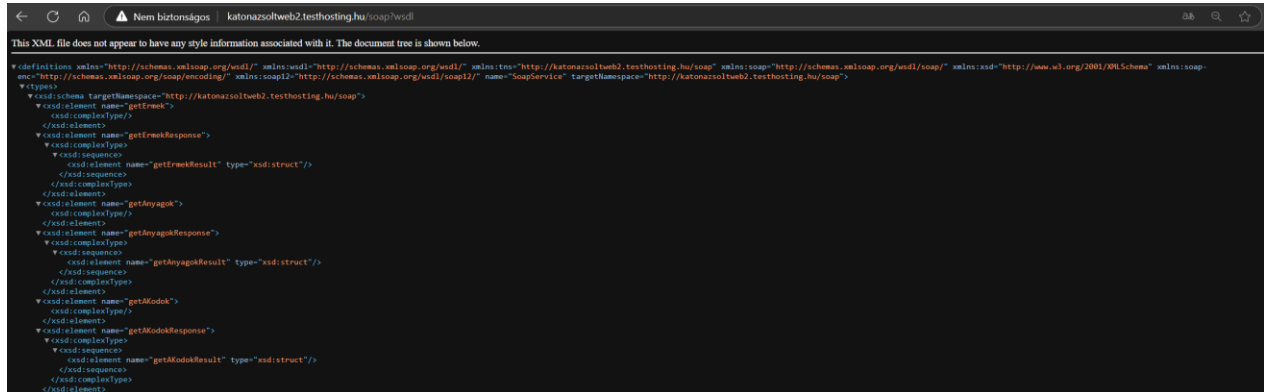
Az MNB szolgáltatásokhoz kapcsolódó útvonalak, a web.php-ban kerültek meghatározásra. Az összes végpont egy /mnb prefixel van ellátva, így külön csoportosítva az alkalmazás egyéb végpontjaitól.

A nézeteket, vagyis a weboldalon történő megjelenítést a resources/views/MNB és a resources/view/livewire mappában található php fájlok valósítják meg, utóbbi a dinamikus tartalom megjelenítésben nyújt segítséget.

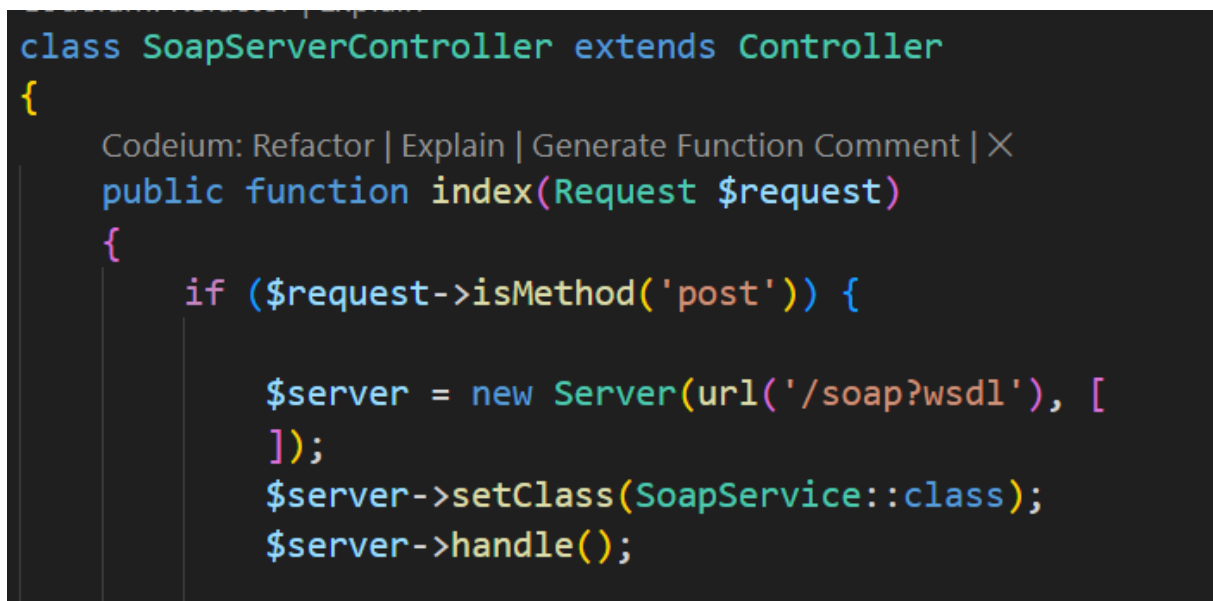
Saját SOAP Server

Az alkalmazás saját SOAP Szerverrel rendelkezik, mely a felhasznált táblák teljes tartalmát képes visszaadni.

WSDL: <http://katonaszoltweb2.testhosting.hu/soap?wsdl>



A SoapServerController kezeli.



Design

A weboldal a feladatnak megfelelően egy ingyenesen elérhető CSS+HTML témát használ, némi bootstrap kiegészítéssel.

Téma alapja: <https://themewagon.com/themes/stocker/>

A téma tökéletesen illik az általunk választott témához, mely a Magyar Forint pénzürméire fókuszál.

Felhasználók kezelése

A felhasználók kezelésében a Laravel Auth van segítségünkre. A Laravel Auth legenerálta a kellő kontrollereket, middlewareket és a User modellt.

A User modell került bővítésre egy „role” mezővel, amely a felhasználó jogosultságaiért felel.

```
protected $fillable = [  
    'name',  
    'email',  
    'password',  
    'role',  
];  
  
protected function casts(): array  
{  
    return [  
        'email_verified_at' => 'datetime',  
        'password' => 'hashed',  
        'role' => 'string',  
    ];  
}
```

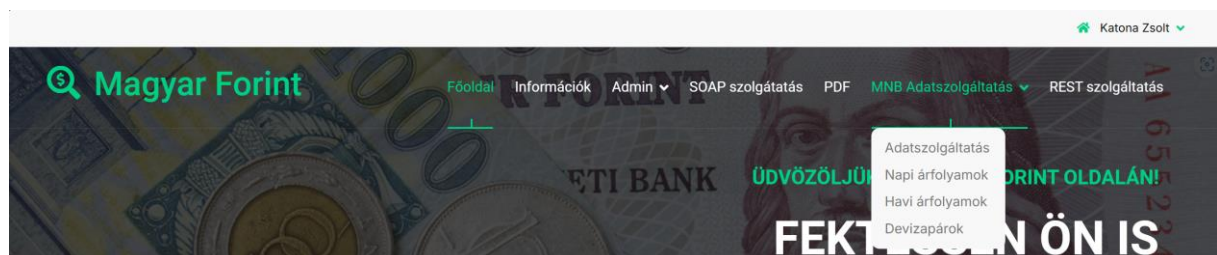
Jogosultságok a feladatot követően a következők:
„látogató”, „regisztrált látogató” és „admin”

Middleware:
IsAdmin

```
public function handle($request, Closure $next)  
{  
    if (auth()->check() && auth()->user()->role === 'admin') {  
        return $next($request);  
    }  
  
    return redirect('/');  
}
```

Menük

A feladatnak megfelelően a menük az adatbázisból töltődnek be és többszintes menürendszert valósítanak meg.



A menük az Admin menüpont „Menük” almenüpontja alatt szerkeszthetőek.

Az adminmenü kontroller egy Livewire kontroller így az oldal frissítése nélkül mennek végbe a feladatok.

Új Menü								
Menü Neve	View	Menü Cím	Szerepkör	Szülő	Menü Sorrend	Aktív	Műveletek	
Főoldal	/	Főoldal	none	0	1	Igen	Szerkesztés	Törlés
Információk		Információk	user	0	2	Igen	Szerkesztés	Törlés
Admin	admin	Adminisztráció	admin	0	3	Igen	Szerkesztés	Törlés
SOAP szolgáltatás	soapData	SOAP szolgáltatás	none	0	4	Igen	Szerkesztés	Törlés
Menük	/admin/menuk	Admin- Menük	none	3	1	Igen	Szerkesztés	Törlés
PDF	pdf	PDF Generátor	none	0	6	Igen	Szerkesztés	Törlés
MNB Adatszolgáltatás	mnb	MNB SOAP	none	0	7	Igen	Szerkesztés	Törlés
Adatszolgáltatás	mnb/mnb	MNB SOAP service	none	8	8	Igen	Szerkesztés	Törlés
Napi árfolyamok	mnb/exchangeRate	Árfolyam lekérdezés	none	8	9	Igen	Szerkesztés	Törlés
Havi árfolyamok	mnb/monthlyExchangeRate	Árfolyam lekérdezés	none	8	10	Igen	Szerkesztés	Törlés
Devizapárok	mnb/showCurrencyPair	Devizapárok lekérése	none	8	11	Igen	Szerkesztés	Törlés
REST szolgáltatás	/rest/erme	REST szolgáltatás	user	0	12	Igen	Szerkesztés	Törlés

Minden menühöz megadható:

- Menü megjelenített neve
- Linkje (View)
- Menü címe
- Láthatósága (Szerepkör: none,user,admin)
- Szülő (ha van)
- Sorrend
- Státusz (Aktív?)

Megjelenítés:

A menüket az AppServiceProvider osztja meg a nézetekkel.

```
public function boot(): void
{
    $menuitems = Menu::where('active',1)->orderBy('menu_order')->get();
    view()->share('menuitems', $menuitems);
}
```

A menü layout a \resources\views\layouts\menu.blade.php -ben kapott helyet.

Oldalak

Főoldal

A **Főoldal** a magyar forint érméit és bankjegyeit mutatja be. Az oldal tetején egy szembetűnő üdvözlő szöveg hívja fel a figyelmet a forint fontosságára és történetére. Az oldal középső részén egy táblázat jelenik meg, amely részletes adatokat tartalmaz az érmékről, például címlet, súly, kiadás és bevonás dátuma, összetételük, valamint a tervezőik nevei.

A táblázat alatt vizuális elemekkel illusztrált bankjegyek és érme láthatók, mindegyikhez rövid leírás tartozik, amely ismerteti az adott címlet érdekességeit és hátterét. Az oldal kiemeli a forgalomban lévő bankjegyeket és érméket, külön figyelmet fordítva azok dizájnjára és történeti értékére. A megjelenés modern és reszponzív, felhasználóbarát navigációval, így könnyen kezelhető bármilyen eszközön. Az oldal célja az ismeretterjesztés és az érme, valamint bankjegyek történetének bemutatása egy esztétikus és informatív platformon.

A táblázat lapozható és a fent helyet kapott bevitelimezővel szűrhető is. (Keresés)

Az oldal alján helyet kapott egy a csalásokra figyelmeztető szöveg.



Fedezze fel a Magyar Pénztermék Különleges Fémösszetételét!

A magyar pénzeszközök nem csupán a mindennapi életben, hanem a történelem és a kultúra színpadán is fontos szerepet töltenek be. A Magyar Pénztermék Különleges Fémösszetételét a Magyar Nemzeti Bank gondoskodik róla, hogy a pénzeszközök ne csak a pénzügyi, hanem a történelmi és kulturális értékek hordozói is legyenek.

Különleges Fémösszetétel
A pénzeszközök fémösszetételét a Magyar Nemzeti Bank gondoskodik róla, hogy a pénzeszközök ne csak a pénzügyi, hanem a történelmi és kulturális értékek hordozói is legyenek.

Történelmi és Esztétikai Érték
A pénzeszközök nem csupán a mindennapi életben, hanem a történelem és a kultúra színpadán is fontos szerepet töltenek be. A Magyar Pénztermék Különleges Fémösszetételét a Magyar Nemzeti Bank gondoskodik róla, hogy a pénzeszközök ne csak a pénzügyi, hanem a történelmi és kulturális értékek hordozói is legyenek.


Fenntartás és Visszatérítési Követelmények
A pénzeszközök fémösszetételét a Magyar Nemzeti Bank gondoskodik róla, hogy a pénzeszközök ne csak a pénzügyi, hanem a történelmi és kulturális értékek hordozói is legyenek.

Visszatérítési és Visszatérítési Követelmények
A pénzeszközök fémösszetételét a Magyar Nemzeti Bank gondoskodik róla, hogy a pénzeszközök ne csak a pénzügyi, hanem a történelmi és kulturális értékek hordozói is legyenek.

Mindenkori érme legérdekesebb tulajdonságai									
ID	Címlet	Érték [g]	Dátum	Kiadás dátuma	Bevonás dátuma	Anyag	Tervező		
1	150	227 156 000	1546-08-31	1556-06-30		Alumínium	Székelyi István		
2	1	421 892 123	1867-03-12	1889-06-30		Alumínium	Székelyi István		
3	2,35	465 571 975	1885-03-16	1908-03-28		Székelyi István, Székelyi István	Székelyi István		
4	2	13 000 000	1846-08-31	1901-06-30		Alumínium	Székelyi István		
5	5,100	8 700 000	1885-03-16	1901-06-30		Székelyi István	Székelyi István		
6	2	303 208 000	1873-07-01	1901-06-30		Székelyi István	Székelyi István		
7	2	467 772 000	1867-03-12	1901-06-30		Székelyi István	Székelyi István		
8	22,500	30 000	1846-08-31	1901-06-30		Székelyi István	Székelyi István		
9	12,000	10 000 000	1846-08-31	1901-06-30		Székelyi István	Székelyi István		
10	2,100	10 000 000	1846-08-31	1901-06-30		Székelyi István	Székelyi István		

Jelenleg forgalomban lévő bankjegyek és pénzérme


Bankjegyek



500 forintos bankjegy

Az 500 forintos bankjegy a Magyar Nemzeti Bank által kiadott legkisebb címletű bankjegy, amely a pénzügyi életben a leggyakrabban használt bankjegy.


[Tudjon meg többet](#)



1000 forintos bankjegy

Az 1000 forintos bankjegy a Magyar Nemzeti Bank által kiadott bankjegy, amely a pénzügyi életben a leggyakrabban használt bankjegy.


[Tudjon meg többet](#)



2000 forintos bankjegy

Az 2000 forintos bankjegy a Magyar Nemzeti Bank által kiadott bankjegy, amely a pénzügyi életben a leggyakrabban használt bankjegy.


[Tudjon meg többet](#)



5000 forintos bankjegy

Az 5000 forintos bankjegy a Magyar Nemzeti Bank által kiadott bankjegy, amely a pénzügyi életben a leggyakrabban használt bankjegy.


[Tudjon meg többet](#)



10 000 forintos bankjegy

Az 10 000 forintos bankjegy a Magyar Nemzeti Bank által kiadott bankjegy, amely a pénzügyi életben a leggyakrabban használt bankjegy.

[Tudjon meg többet](#)




20 000 forintos bankjegy

Az 20 000 forintos bankjegy a Magyar Nemzeti Bank által kiadott bankjegy, amely a pénzügyi életben a leggyakrabban használt bankjegy.

[Tudjon meg többet](#)


Érme



5 forintos érme

A 5 forintos érme a Magyar Nemzeti Bank által kiadott legkisebb címletű érme, amely a pénzügyi életben a leggyakrabban használt érme.


[Tudjon meg többet](#)



10 forintos érme

A 10 forintos érme a Magyar Nemzeti Bank által kiadott érme, amely a pénzügyi életben a leggyakrabban használt érme.


[Tudjon meg többet](#)



20 forintos érme

A 20 forintos érme a Magyar Nemzeti Bank által kiadott érme, amely a pénzügyi életben a leggyakrabban használt érme.


[Tudjon meg többet](#)



50 forintos érme

A 50 forintos érme a Magyar Nemzeti Bank által kiadott érme, amely a pénzügyi életben a leggyakrabban használt érme.


[Tudjon meg többet](#)



100 forintos érme

A 100 forintos érme a Magyar Nemzeti Bank által kiadott érme, amely a pénzügyi életben a leggyakrabban használt érme.

[Tudjon meg többet](#)

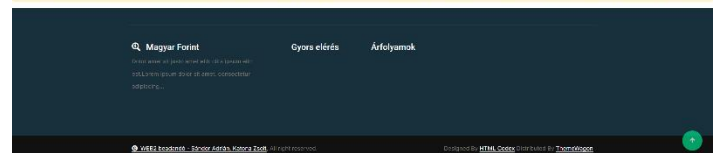


200 forintos érme

A 200 forintos érme a Magyar Nemzeti Bank által kiadott érme, amely a pénzügyi életben a leggyakrabban használt érme.

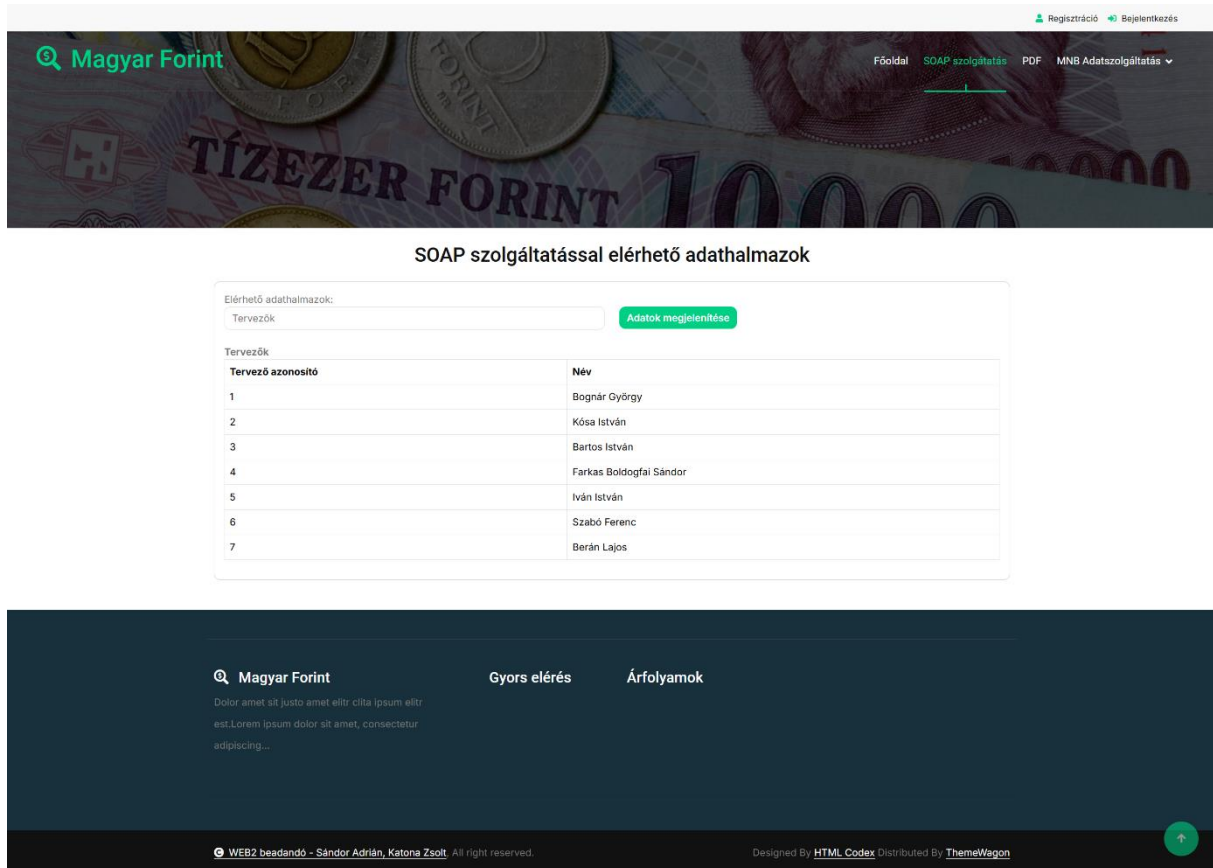
[Tudjon meg többet](#)

Figyelem! Csak az érvényes érmék, bankjegyek és pénzeszközök használata javasolt. A pénzeszközök használata során a pénzeszközök érvényességét mindig ellenőrizze.



SOAP Szolgáltatás

Az oldalon a weboldal saját SOAP szolgáltatása tesztelhető egy modern, de letisztult környezetben. A lenyíló menüben kiválaszthatóak az adatbázis táblái majd az adatok lekérdezése gomb megnyomására táblázatba rendezve jelennek meg a kapott adatok.



Az oldalt a SoapClientController vezérli. A kontroller a SoapClientService-t hívja meg.

```
Codeium: Refactor | Explain | Generate Function Comment | X
public function getErmek() {
    return response()->json($this->soapClient->getErmek());
}
Codeium: Refactor | Explain | Generate Function Comment | X
public function getTervezok() {
    return response()->json($this->soapClient->getTervezok());
}
Codeium: Refactor | Explain | Generate Function Comment | X
public function getTKodok() {
    return response()->json($this->soapClient->getTKodok());
}
Codeium: Refactor | Explain | Generate Function Comment | X
public function getAKodok() {
    return response()->json($this->soapClient->getAKodok());
}
Codeium: Refactor | Explain | Generate Function Comment | X
public function getAnyagok() {
    return response()->json($this->soapClient->getAnyagok());
}
Codeium: Refactor | Explain | Generate Function Comment | X
public function getErmekWithAllInfo() {
    return response()->json($this->soapClient->getErmekWithAllInfo());
}
```

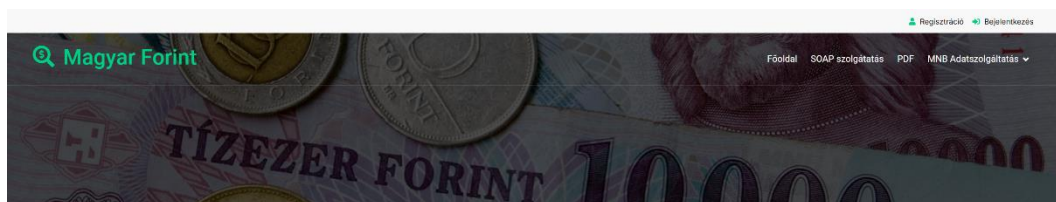
SoapClientService közvetlenül a SOAP Servert szólítja meg és a kapott értékekkel tér vissza.

A SoapClientController a kapott XML-t JSON formátummá alakítja, majd a kapott JSON-t adja tovább a Blade-nek.

MNB Adatszolgáltatás

MNB Adatszolgáltatás

Rövid ismertető az MNB Soap adatszolgáltatásról.



Magyar Nemzeti Bank adatszolgáltatása

A Magyar Nemzeti Bank (MNB) adatszolgáltatása kulcsfontosságú szerepet játszik a pénzügyi piacon és a gazdasági döntéshozatalban. Az MNB különböző adatokat és statisztikákat kínál, amelyek segíthetnek a felhasználóknak a pénzügyi intézményekről, vállalkozásokról vagy pénzügyi intézményekről. A Magyar Nemzeti Bank webservice szolgáltatásának köszönhetően folyamatosan élő árfolyam adatokkal dolgozhatunk.

Miért jó az MNB adatszolgáltatása?

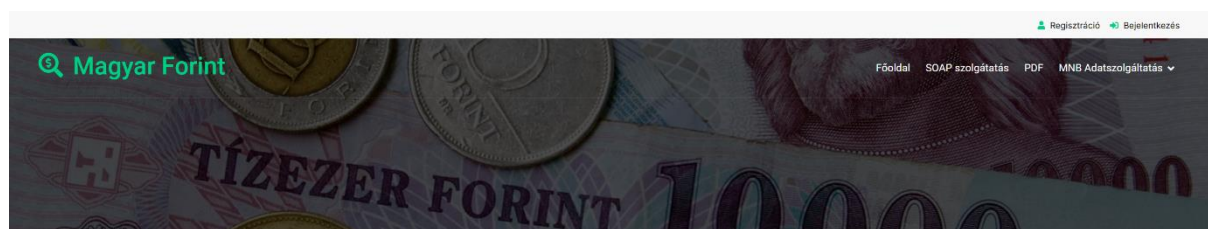
- Áttáthatóság és megbízhatóság:**
Az MNB által nyújtott adatok hiteles forrást jelentenek.
- Valós idejű információk:**
Az MNB folyamatosan frissíti adatbázisát, amely lehetővé teszi, hogy a legfrissebb árfolyamokat és statisztikákat dolgozhassunk. Ez különösen fontos a devizapiacokon, ahol a kis eltérések is jelentős pénzügyi következményekkel járhatnak.

Mire használható az MNB adatszolgáltatása?

- Deviza árfolyamok nyomon követése:**
Az adatszolgáltatás segítségével a különböző devizák árfolyamait.
- Gazdasági elemzések:**
A kutatók és elemzők az MNB adatainak felhasználásával mélyreható elemzéseket készíthetnek a magyar gazdaság állapotáról, valamint a pénzügyi piacok alakulásáról.
- Befektetési döntések támogatása:**
A befektetők számára az MNB által nyújtott információk segíthetnek a pénzügyi eszközök helyes kiválasztásában.

Napi árfolyam

Egy valuta adott napi árfolyamának lekérdezése.

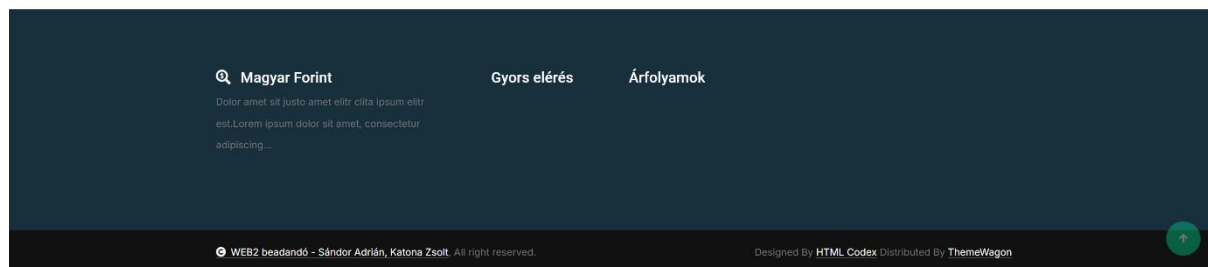


Napi árfolyam

Figyelem! A szolgáltatás csak hétköznapokra tud árfolyam adatokat megjeleníteni. Kérjük, válasszon hétköznapi dátumot.

Devizák: Dátum:

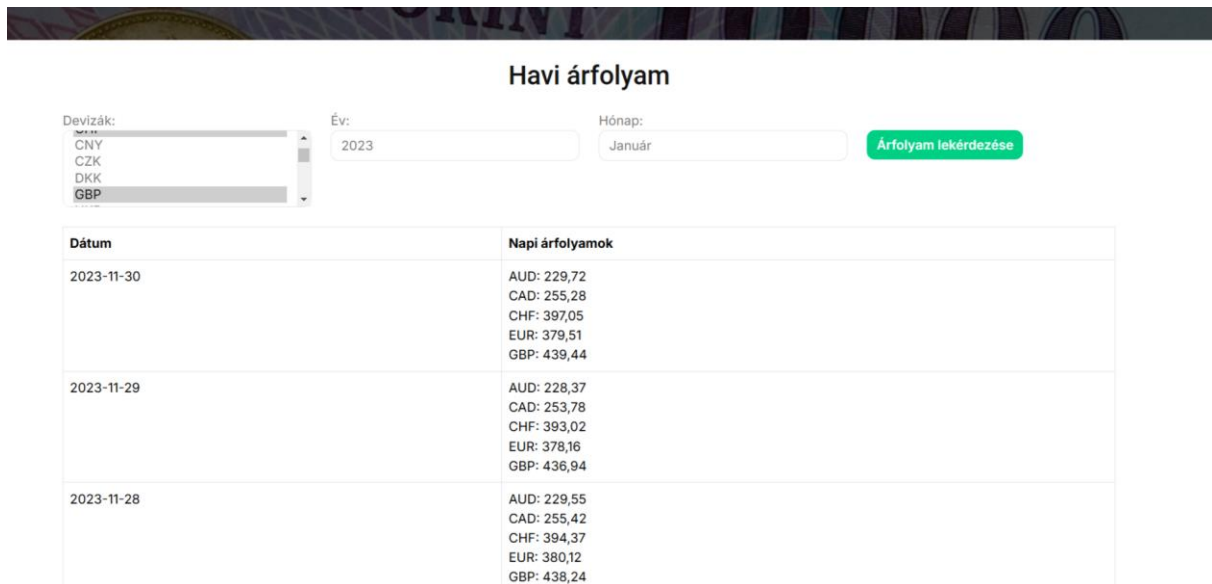
A 2024-11-25 napi GBP árfolyam: 493,69Ft



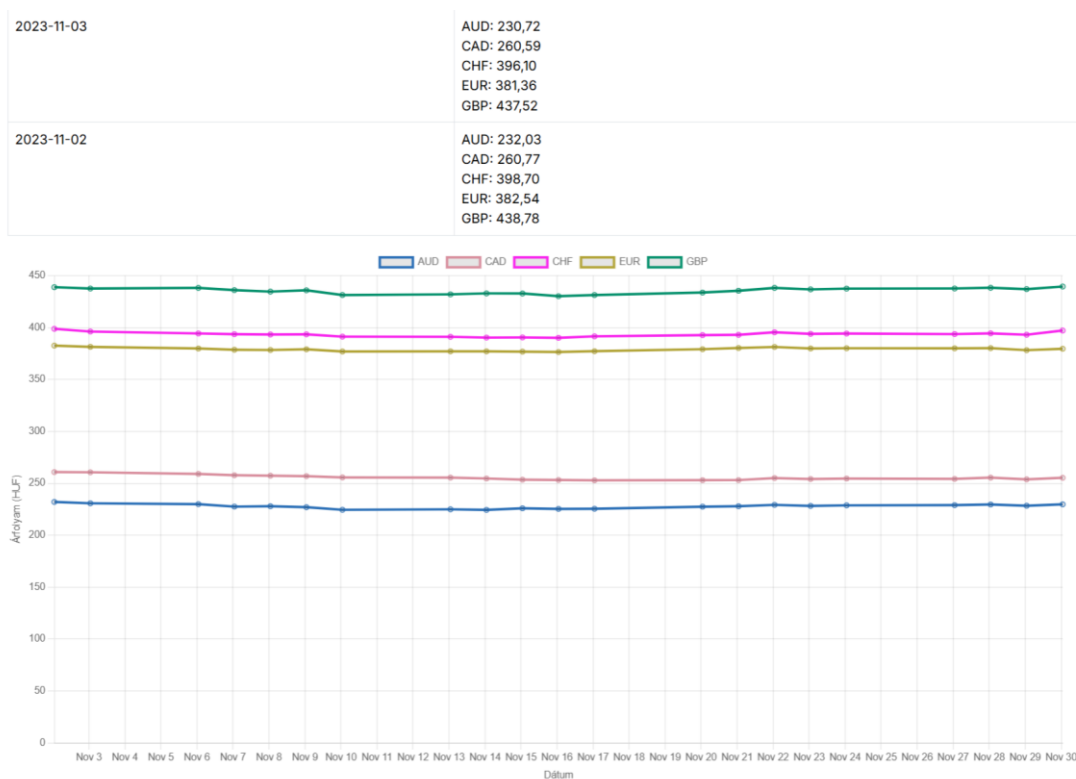
Devizák havi árfolyamai

A felhasználó több - akár az összes felsorolt – valuta árfolyamát kéri le adott hónapra.

Felhasználónak lehetősége van akár az összes felsorolt valuta kiválasztására.



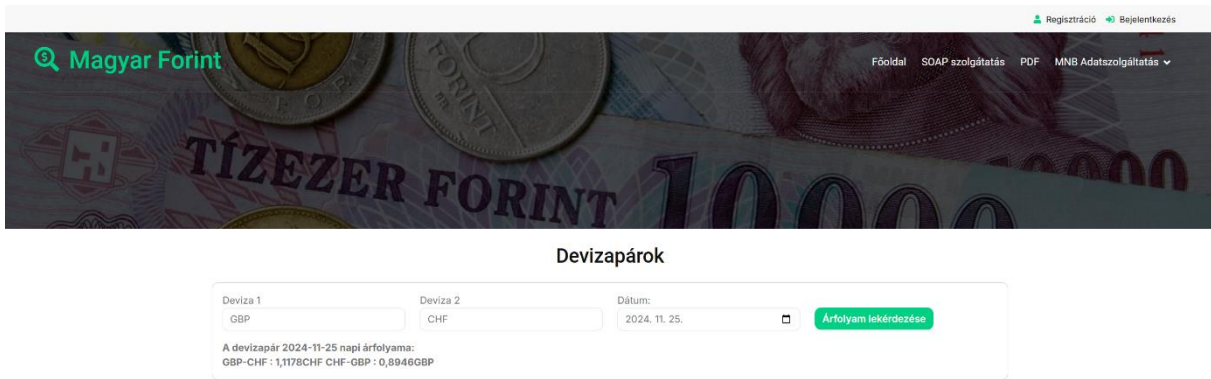
A kapott táblázat alatt egy dinamikusan frissülő diagramm látható, mely a Chart.js felhasználásával kerül létrehozásra.



Az oldalt az MNBController vezérli amely az MNBService-ből gyűjti az adatokat. Az MNBService közvetlenül az MNB Soap szervertől kéri le az aktuális adatokat.

Devizapárok napi árfolyama

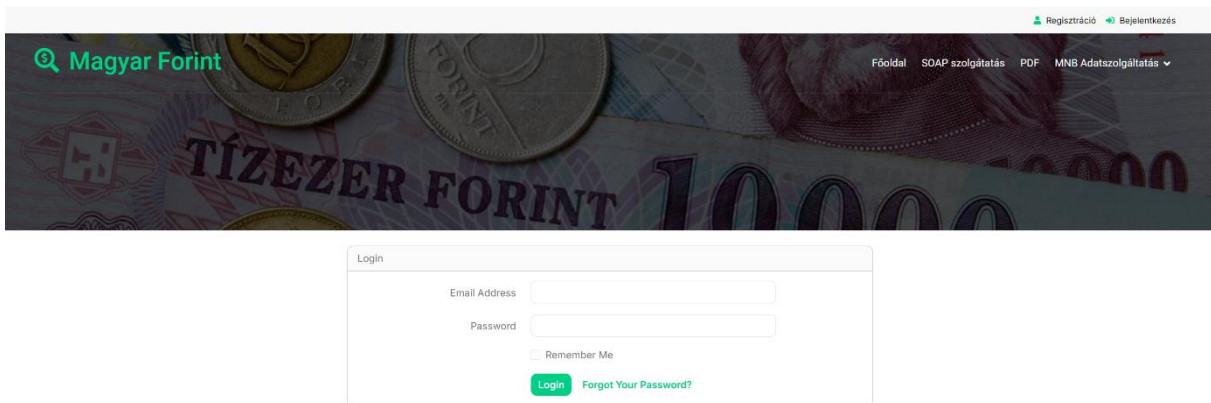
A felhasználó 2 valuta árfolyamát kéri le adott napra.



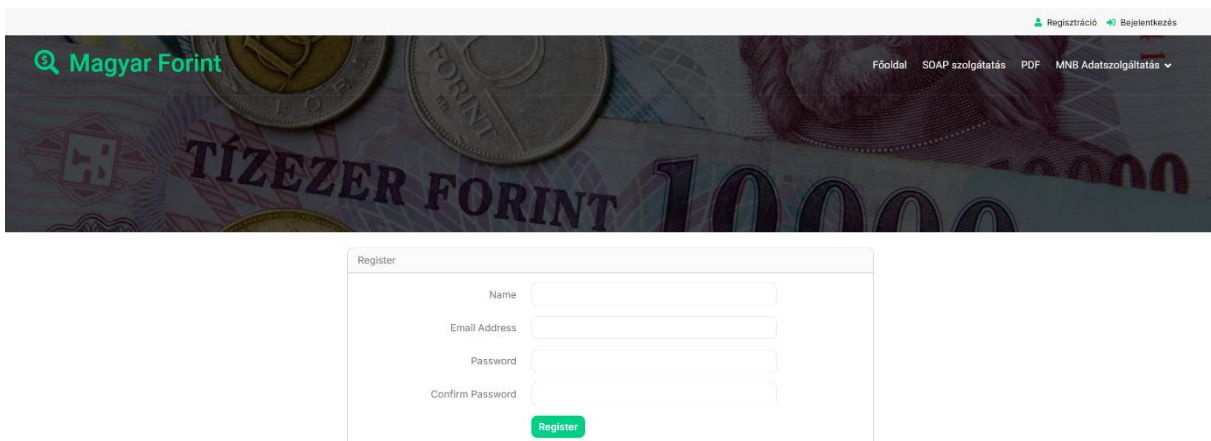
Az oldalt az MNBController vezérli amely az MNBService-ből gyűjti az adatokat. Az MNBService közvetlenül az MNB Soap szervertől kéri le az aktuális adatokat.

Authentikációs oldalak

Bejelentkezés



Regisztráció



Rest szolgáltatás

Bejelentkezett felhasználó az „ermek” táblázatban tud ezzel az interfésszel módosításokat végezni. A végpontok [ITT](#) láthatóak.

Karona Zsolt

Magyar Forint

Főoldal Információk Admin SOAP szolgáltatás PDF MNB Adatszolgáltatás REST szolgáltatás

Érme adatok kezelése

ID	Címlet	Tömeg (g)	Darabszám	Kiadás Dátuma	Bevonás Dátuma
1	1	1.50	227158000	1946-08-01	1995-06-30
2	1	1.40	431890120	1967-05-12	1995-06-30
3	1	2.05	483371015	1993-03-29	2008-02-28
4	2	2.85	13500000	1946-08-01	1951-11-30
5	2	5.00	57528000	1950-01-20	1971-06-30
6	2	4.44	303208159	1970-07-01	1995-06-30
7	2	3.10	467772105	1993-03-29	2008-02-28
8	5	20.00	39802	1946-08-01	1977-06-30
10	5	12.00	10004252	1947-05-19	1977-06-30
11	5	7.40	20029200	1967-05-12	1972-06-30
12	5	5.73	58284387	1971-08-02	1987-03-31
13	5	5.00	109668035	1983-04-18	1995-06-30
14	5	4.20	197772300	1993-06-21	
15	10	8.83	66130376	1971-06-01	1987-03-31
16	10	6.10	108330025	1983-04-18	1995-06-30
17	10	6.10	158688505	1993-06-21	
18	20	7.06	108792015	1983-04-18	1995-06-30
19	20	6.90	171477005	1993-03-29	
20	50	7.70	65406505	1993-03-29	
21	100	9.40	42573505	1993-06-21	1998-12-31
22	100	8.00	155073000	1996-10-21	
23	200	12.00	11250014	1992-12-01	1998-04-03

Adatok módosítása

Módosítás Törés Rögzítés

Azonosító

1

Címlet

1

Tömeg

1,50

Darab

227158000

Kiadás dátuma

1946. 08. 01.

Bevona dátuma

1995. 06. 30.

Módosítás

Érme adatok kezelése

ID	Címlet	Tömeg (g)	Darabszám	Kiadás Dátuma	Bevonás Dátuma
1	1	1.50	227158000	1946-08-01	1995-06-30
2	1	1.40	431890120	1967-05-12	1995-06-30
3	1	2.05	483371015	1993-03-29	2008-02-28
4	2	2.85	13500000	1946-08-01	1951-11-30
5	2	5.00	57528000	1950-01-20	1971-06-30
6	2	4.44	303208159	1970-07-01	1995-06-30
7	2	3.10	467772105	1993-03-29	2008-02-28
8	5	20.00	39802	1946-08-01	1977-06-30
10	5	12.00	10004252	1947-05-19	1977-06-30
11	5	7.40	20029200	1967-05-12	1972-06-30
12	5	5.73	58284387	1971-08-02	1987-03-31
13	5	5.00	109668035	1983-04-18	1995-06-30
14	5	4.20	197772300	1993-06-21	
15	10	8.83	66130376	1971-06-01	1987-03-31
16	10	6.10	108330025	1983-04-18	1995-06-30
17	10	6.10	158688505	1993-06-21	
18	20	7.06	108792015	1983-04-18	1995-06-30
19	20	6.90	171477005	1993-03-29	
20	50	7.70	65406505	1993-03-29	
21	100	9.40	42573505	1993-06-21	1998-12-31
22	100	8.00	155073000	1996-10-21	
23	200	12.00	11250014	1992-12-01	1998-04-03

Adatok módosítása

Módosítás Törés Rögzítés

Azonosító

1

Törés

Érme adatok kezelése

ID	Cimlet	Tömeg (g)	Darabszám	Kiadás Dátuma	Bevonás Dátuma
1	1	1.50	227158000	1946-08-01	1995-06-30
2	1	1.40	431890120	1967-05-12	1995-06-30
3	1	2.05	483371015	1993-03-29	2008-02-28
4	2	2.85	13500000	1946-08-01	1951-11-30
5	2	5.00	57528000	1950-01-20	1971-06-30
6	2	4.44	303208159	1970-07-01	1995-06-30
7	2	3.10	467772105	1993-03-29	2008-02-28
8	5	20.00	39802	1946-08-01	1977-06-30
10	5	12.00	10004252	1947-05-19	1977-06-30
11	5	7.40	20029200	1967-05-12	1972-06-30
12	5	5.73	58284387	1971-08-02	1987-03-31
13	5	5.00	109688035	1983-04-18	1995-06-30
14	5	4.20	197772300	1993-06-21	
15	10	8.83	66130376	1971-06-01	1987-03-31
16	10	6.10	108330025	1983-04-18	1995-06-30
17	10	6.10	158688505	1993-06-21	
18	20	7.06	108792015	1983-04-18	1995-06-30
19	20	6.90	171477005	1993-03-29	
20	50	7.70	65406505	1993-03-29	
21	100	9.40	42573505	1993-06-21	1998-12-31
22	100	8.00	155073000	1996-10-21	
23	200	12.00	11250014	1992-12-01	1998-04-03

Adatok módosítása

Módosítás Törés Rögzítés

Cimlet

2

Tömeg

3,10

Darab

467772105

Kiadás dátuma

1993. 03. 29.

Bevona dátuma

2008. 02. 28.

Rögzítés

Az oldal funkcióit a RestClientController vezérli.

Admin / Menü

Admin felhasználó ezen az oldal tudja a menüket kezelni. Menükről részletesen [ITT](#).

Magyar Forint

Főoldal Információk Admin SOAP szolgáltatás PDF MNB Adatszolgáltatás REST szolgáltatás

Új Menü							
Menü Neve	View	Menü Cím	Szerepkör	Szülő	Menü Sorrend	Aktív	Műveletek
Főoldal	/	Főoldal	none	0	1	Igen	Szerkesztés Törés
Információk		Információk	user	0	2	Igen	Szerkesztés Törés
Admin	admin	Adminisztráció	admin	0	3	Igen	Szerkesztés Törés
SOAP szolgáltatás	soapData	SOAP szolgáltatás	none	0	4	Igen	Szerkesztés Törés
Menük	/admin/menuk	Admin- Menük	none	3	1	Igen	Szerkesztés Törés
PDF	pdf	PDF Generátor	none	0	6	Igen	Szerkesztés Törés
MNB Adatszolgáltatás	mnb	MNB SOAP	none	0	7	Igen	Szerkesztés Törés
Adatszolgáltatás	mnb/mnb	MNB SOAP service	none	8	8	Igen	Szerkesztés Törés
Napi árfolyamok	mnb/exchangeRate	Árfolyam lekérdezés	none	8	9	Igen	Szerkesztés Törés
Havi árfolyamok	mnb/monthlyExchangeRate	Árfolyam lekérdezés	none	8	10	Igen	Szerkesztés Törés
Devizapárok	mnb/showCurrencyPair	Devizapárok lekérése	none	8	11	Igen	Szerkesztés Törés
REST szolgáltatás	/rest/erme	REST szolgáltatás	user	0	12	Igen	Szerkesztés Törés

PDF

A felhasználó megadott választható értékekből PDF-et tud generálni a TCPDF segítségével. Az oldalt a PDFController vezérli.

Érme Információk Lekérdezése

Érme Címletek:

Válasszon címetet

Anyag Típusok:

Válasszon anyagot

Tervezők:

Válasszon tervezőt

PDF Generálása