Szerepkörök:

* Weboldalon azonosítatlan felhasználó (vendég):
  + Bejelentkezés
  + Jelszó módosító link kérése e-mailben (“Elfelejtettem a jelszavam”)
  + Könyvtár katalógusának megtekintése
  + Könyvek részleteinek megtekintése
* Weboldalon regisztrált felhasználó:
  + Könyvtár katalógusának megtekintése
  + Könyvek részleteinek megtekintése
  + Könyvek kedvencekhez adása, vagy levétele
  + Könyvek lefoglalása/előjegyzése (a könyv státuszától függően)
  + Felhasználói profil megtekintése
    - Jelszó módosítás
    - Kijelentkezés
    - Lefoglalt, előjegyzett könyveinek megtekintése, és azok törlése
    - Jelenleg a felhasználó által kivet könyvek megtekintése, és meghosszabbítása (állapottól függően maximum a system\_settings adatbázis táblában meghatározott alkalommal)
    - Korábban a felhasználó által kivett könyvek megtekintése
    - Kedvencek megtekintése, és levétel a kedvencekről
* Weboldalon regisztrált felhasználó lejárt taggsággal:
  + Az azonosítatlan felhasználó jogosultságain felül tud:
    - Kedvencekhez adni könyveket
    - Profilján megtekinteni mindent amit érvényes taggsággal, viszont nem tudja meghosszabbítani a kivett könyveit
* Asztali alkalmazásban normál dolgozói profil:
  + Téma váltás
  + Felhasználók regisztrálása
  + Könyvek
    - Adatainak módosítása
    - Kiadása
    - Visszavétele
    - Lefoglalása (adott felhasználó számára)
    - Előjegyzése (adott felhasználó számára)
    - Visszaadása
    - Hozzáadása
    - Raktárba vétele
    - Leselejtezése
  + Felhasználó
    - Adatainak módosítása
    - Tagság meghosszabbítása
    - Felhasználó deaktiválása (törlése)
  + Összes korábban kiadott, jelenleg kiadott, lefoglalt, előjegyzett könyv megtekintése
  + Kiadó
    - Hozzáadása
    - Adatainak módosítása
  + Dolgozói profil
    - Jelszavának módosítása
    - Deaktiválása
    - Kijelentkezés
* Asztali alkalmazásban admin dolgozói profil:
  + A normál dolgozói profilon felül tud:
    - Új dolgozót felvinni
    - Másik dolgozói profilt deaktiválni
    - Rendszerbeállításokat módosítani

API ENDPOINTS:

* Könyv katalógus lekérése (opcionálisan) szűrve:
  + Metódus: GET
  + Paraméterek:
    - Title: a könyv címére szűr
    - Genre: a könyv kategóriákra szűr
    - Author: a könyv szerzőire szűr
    - Release\_date: kiadás dátumát szűri
    - Lang: meghatározza milyen nyelvű könyvek jelenjenek meg
    - Isbn: érték amire szűr az isbn-nél
    - Page\_number: egy szám ami representálja, hogy hanyadik oldalát kéri a találati listának
  + Visszatérési érték:
    - A találati lista html kódja
* Felhasználó bejelentkezés:
  + Metódus: POST
  + Paraméterek:
    - Uname: felhasználónév
    - Pw: jelszó
    - remember\_me : logikai változó, ami megadja, hogy az “emlékezz rám” opciót bepipálta-e bejelentkezéskor a felhasználó
  + Visszatérési érték:
    - String:
      * “not found” ha a felhasználónév nem található
      * “inactive user” ha a felhasználó fiókja deaktiválva (törölve) lett
      * “incorrect” ha a jelszó felhasználónév kombináció helytelen
      * “not registered” ha a felhasználó a könyvtárban történő regisztrálás után még nem adott meg jelszót a webes bejelentkezéshez
      * “success” sikeres bejelentkezéskor
* Jelszó módosítása a profil oldalról:
  + Metódus: POST
  + Paraméterek:
    - CurrentPassword: a jelenlegi jelszó
    - NewPassword: a jelszó amire cserélni szeretné
  + Visszatérési érték:
    - JSON ami “status” értéke megadja, hogy sikeres volt-e a csere, vagy ha nem akkor mi a probléma
* Kivett könyv lejárati dátumának meghosszabbítása:
  + Metódus: POST
  + Paraméterek:
    - Book\_id: a könyv egyedi azonosítója
    - Action=”extend”: megadja, hogy mit kell tenni, jelen esetben meghosszabbítani
  + Visszatérési érték:
    - JSON ami “message” értéke megadja, hogy sikeres volt-e a hosszabbítás
* Felhasználó bejelentkezve van-e?:
  + Metódus: POST
  + Paraméterek:
    - Action=”isLoggedIn”: megadja, hogy mit kell tenni, jelen esetben megmondani, hogy a felhasználó be van-e jelentkezve
  + Visszatérési érték:
    - Felhasználó egyedi azonosítója, vagy “”
* Kijelentkezés:
  + Metódus: POST
  + Paraméterek:
    - Action=”logout”: megadja, hogy mit kell tenni, jelen esetben kijelentkeztetni a felhasználót
  + Visszatérési érték:
    - Nincs
* “Elfelejtettem a jelszavam” link kérés:
  + Metódus: POST
  + Pareméterek:
    - Username: felhasználónév
    - Action=”resetPassword”: megadja, hogy mit kell tenni, jelen esetben “elfelejtettem a jelszavamat” funkcióhoz linket küldeni email-re
  + Visszatérési érték:
    - JSON ami “status” értéke “invalid” ha nem található a felhasználó a megadott felhasználónévvel, vagy “failed\_to\_send\_email”, ha nem sikerült valami miatt kiküldni az emailt, vagy “success” ilyenkor az “email” értéke a felhaszálóhoz kapcsolt email cím lesz, az első négy karakter kivételével az @-ig kicsillagozva (pl.: hort\*\*\*\*\*\*\*@gmail.com)
* Könyv levétele kedvencek közül:
  + Metódus: POST
  + Paraméterek:
    - ISBN\_id: a könyv isbn-jének az egyedi azonosítója
    - Action=”removeFromFavorites”” megadja, hogy mit kell tenni, jelen esetben törölni a megadott könyvet a kedvencek közül
  + Visszatérési érték:
    - Nincs
* Könyv felvétele kedvencekbe:
  + Metódus: POST
  + Paraméterek:
    - ISBN\_id: a könyv isbn-jének az egyedi azonosítója
    - Action=”addToFavorites” megadja, hogy mit kell tenni, jelen esetben törölni a megadott könyvet a kedvencek közül
  + Visszatérési érték:
    - Nincs
* Jelszó módosítása az emaliben kapott linken keresztül (“elfelejtettem a jelszavamat” funkció):
  + Metódus: POST
  + Paraméterek:
    - Reset\_token: a kapott linkben található token
    - Password: jelszó
  + Visszatérési érték:
    - JSON ami “status” értéke “success” ha sikeresen ki lett cserélve a jelszó, “failed” ha valamilyen hiba folytán nem sikerült, “expired” ha a link már nem érvényes
* Alkalmazott bejelentkeztetése:
  + Metódus: POST
  + Paraméterek:
    - Username: felhasználónév
    - Passw: jelszó
  + Visszatérési érték:
    - JSON-ban visszaadja a “loginEmployee” sql eljárás értékét

A további 48 endpointot az asztali alkalmazás használja, a kommunikáció egyszerűsítésének és egységesítésének érdekében ugyanazt a sémát követi az összes. Mindegyik POST metódus, és van egy “type” paramétere, ami meghatározza melyik SQL eljárást fogja meghívni az API. Az endpoint többi paramétere az eljáráshoz szükséges paramétereket tartalmazza ugyanazzal a névvel, ahogy az eljárásban szerepel. A válaszként pedig az eljárás visszatérési értékét küldi vissza JSON formátumban. Ahol nincs visszatérési értéke az eljárásnak ott nem küld vissza semmit. Az egyetlen kivétel az “addUser” “type” paraméterű endpointnál van, ott a fent említettek mellet még egy jelszó változtatást is meghív az API.

Fejlesztői dokumentáció WEB:

Alap tudnivalók a weboldal működéséről:

Minden böngészőből megnyitott link, ami nem egy konkrét fájlra mutat, az átirányítódik az index.php-ra. Ott dől el a beérkező URL alapján, hogy melyik fájl jelenítődik meg. Ennek az az előnye, hogy változtatni tudjuk, hogy mi legyen a linkben az adott oldalakon tartózkodás során, és nem kell hozzá például változtatni a fájlok nevét és helyét a fájlstruktúrában. Ezen felül még nagy előny, hogy minden bejelentkezéssel kapcsolatos ellenőrzésnek elég csak ebben az egy fájlban szerepelnie.

Bejelentkezés és hozzákapcsolódó részek:

Minden session indítás előtt beállítjuk a session cookie tulajdonságait(mivel mindenhol így van ezért, későbbiekben nem kerül említésre), fontosabbak:

* “lifetime”: megadja, hogy mikor jár le, mivel azt szeretnénk, hogy a böngésző bezárásakor jelentkeztesse ki a felhasználót, ezért nulla értékre állítjuk
* “httponly”: igaz értékkor megakadályozza, hogy javascript-ből el lehessen érni a cookie-t
* “samesite”: “Strict” értékre való állításakor védelmet biztosít CSRF (Cross-site request forgery) támadások ellen

Index.php-n történő folyamatok egy oldal megnyitásakor:

* Elindul egy session
* Ellenőrzi a program, hogy mikor volt a legutóbbi aktivitás (API-al történő kommunikációnál, és egy másik oldalra való navigálásnál frissül az ideje), ha régebben volt, mint a megadott idő (jelen esetben 1800 másodperc -> 30 perc), akkor törli a session-t és adatait, majd új session-t indít (ezzel kilépteti a felhasználót, ha be volt jelentkezve)
  + Erre azért van szükség, mert vannak olyan böngészők, amik rendelkeznek olyan (vagy hasonló) funkcióval, hogy “Onnan folytatom, ahol abbahagytam”, ami visszaállítja a cookie-kat is, ezzel azt eredményezve, hogy bejelentkezve marad a felhasználó annak ellenére is, hogy be volt zárva a böngésző (hiába van beállítva a session cookie, élettartama nullára), ami problémákat okozhatna.
* Beállítja a legutóbbi aktivitás idejét a jelen időre
* Ellenőrzi (ha jelenleg nincs bejelentkezve senki), hogy be van-e kapcsolva az “emlékezz rám” funkció, ha be van, akkor ellenőrzi, hogy az adatbázisban megtalálható-e a cookie-ban található token és érvényes is, akkor új session id-t, és “emlékezz rám” tokent generál, majd bejelentkezteti a felhasználót. Ha nincs egyezés az adatbázissal, akkor törli a cookie-t.
* Megvizsgálja a kért linket, és az alapján (helyenként az oldalhoz szükséges feltételek vizsgálata után) eldönti, hogy melyik fájlt töltse be, vagy hova irányítsa át a kérést

Az “elementCreators.php”-ben található metódusok html kódot készítenek a kapott paraméterek függvényében.

Jelszó módosítás működése:

* Profil oldalról:
  + Javascript ellenőrzi, hogy minden feltételnek megfelel-e a megadott új jelszó, és megegyezik-e amikor megvan ismételve, csak ezután küldi el az API-nak, az új és a régi jelszót. A kérés beérkezése után ellenőrésre kerül, hogy be van-e jelentkezve is a felhasználó, ha igen, akkor utána ellenőrzi a megadott jelszavat, ha az érvényes, akkor kicseréli az újra, majd a biztonság kedvéért újra ellenőrizzük, hogy az új jelszóval be lehet-e jelentkezni.
* Emailben kapott “elfelejtettem a jelszavamat” linkkel:
  + Az oldalon megadott jelszavak egyezése esetén, illetve ha teljesül minden jelszóra vonatkozó feltétel, csak akkor megy ki a kérés az API-nak. Ott ellenőrzésre kerül a linkben található token. Ha lejárt, vagy fel lett már használva, akkor érvénytelenként nyílvánítja ki. Ha érvényes, akkor kicseréli a jelszót, majd leellenőrzi, hogy az új jelszóval be lehet-e jelentkezni, ha be lehet, akkor törli a tokent az adatbázisból