1. 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.

* 1. Adatbázis táblák
     1. gp tábla:

A gp tábla a nagydíjak (Grand Prix) adatait tárolja, amelyek a versenysorozat különböző helyszínein zajlottak. Minden rekord egy nagydíjat reprezentál, amely tartalmazza a verseny dátumát, nevét és helyszínét. Ez a tábla tehát a versenyek alapvető információit rögzíti, amellyel összekapcsolhatók a versenyzők eredményei az eredmeny táblából.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

* + 1. eredmeny tábla:

Az eredmeny tábla a motorsport versenyek eredményeit tárolja. Minden sor egy adott versenyző (pilóta) egy konkrét versenyen elért eredményét tartalmazza. A tábla különböző oszlopokban tárolja az eredménnyel kapcsolatos adatokat, beleértve a verseny dátumát, ami egy idegenkulcs a gp táblára, továbbá a pilóta azonosítóját, ami szintén egy idegen kulcs, a pilota táblára, a helyezést, a verseny során előforduló hibákat, valamint az autó típusát és motorját.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

* + 1. pilota tábla:

A pilota tábla a versenyzők, azaz a pilóták adatait tárolja. Minden sor egy versenyzőt reprezentál, beleértve a nevüket, nemüket, születési dátumukat és nemzetiségüket. Ez a tábla a motorsport versenyzői adatainak tárolására szolgál, és lehetővé teszi az egyes pilóták azonosítását az eredményeik alapján.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

* + 1. uzenet tábla:

Az **uzenet** tábla egy üzenetek tárolására szolgáló tábla, amelyben a felhasználók által létrehozott üzenetek adatai kerülnek mentésre. Minden rekord egy adott üzenetet tartalmaz, amit egy felhasználó küldött, beleértve az üzenet szövegét, a felhasználó azonosítóját, valamint a rögzítés időpontját.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

* + 1. felhasznalo tábla:

A felhasznalo tábla a felhasználók adatait tárolja, akik a rendszerbe beléphetnek. Minden rekord egy felhasználót reprezentál, amely tartalmazza a felhasználó nevét, jelszavát és jogosultsági szintjét. Ez a tábla kulcsfontosságú a felhasználói hitelesítéshez és jogosultságkezeléshez a rendszerben.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

A Githubon található, importálható adatok.sql a teljes adatbázist tartalmazza. A query a feladat nevű adatbázist is létrehozza importáláskor.

A felhasználók tábla tartalmaz két alap felhasználót, az admin-t és a user-t, ezek jelszavai megegyeznek a felhasználónevekkel, viszont jogosultsági szintjeik eltérőek.

1. 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 mostani projektben ezen négy, leggyakrabban használt metódus közül csupán két típus használata volt szükséges.

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.

Fejlesztés során, Postmannel teszteltük a végpontokat, ezen kérésekről, egy importálható Postman collection elérhető a GIT Repositoryban, minden a következőkben felsorolt végponthoz.

* 1. Diagnosztikai végpontok: /diag

REST API-hoz, sok esetben szerencsés egy olyan végpont kialakítása, amelyet meghívva megbizonyosodhat a kérést indítója, hogy a szolgáltatás elérhető, fut.

Ezek a végpontok jellemzően nem végeznek bonyolult műveleteket, mivel a legkevesebb hibalehetőségre kell törekedni. Célja, hogy egy 200-as üzenetben jelezze, elérhető a szolgáltatás.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Automatikusan generált leírásA projektünkben ez a végpont az available végpont. A válasz, a <http://localhost:8080/diag/available> címre indított http GET kérésre, egy egyszerű html, mely h1-es címsorban a „Service is available” szöveget adja vissza.

* 1. Üzenetek GET és POST végpontok */api/message*

A Message végpontok az üzenetkezelési végpontokat tartalmazza, ahol a felhasználók üzeneteket küldhetnek, illetve lekérdezhetik a már mentett üzeneteket.

* + 1. /getall GET végpont

Az üzeneteket az UzenetRepository.findAll() metódus segítségével kérdezi le a rendszer, amely a Uzenet entitásokat tartalmazza. A válasz json formátumban kerül átadásra. A <http://localhost:8080/api/message/getall> GET kérésre a következő formátumú json-t tartalmazza a response.

[

    {

        "id": 3,

        "felhasznalo\_id": null,

        "uzenet": "Tesztelünk felhasznalo nelkul",

        "rogzites": "2024-10-05T18:07:11.000+00:00"

    },

    {

        "id": 4,

        "felhasznalo\_id": 4,

        "uzenet": "Hello, szeretnék érdeklődni a semmiről.",

        "rogzites": "2024-10-07T16:14:45.000+00:00"

    }

]

* + 1. /post POST végpont

Új üzenet rögzítésére nyújt lehetőséget. A <http://localhost:8080/api/message/post> url-re indított POST kéréssel érhető el a végpont.

REST kérésen keresztül, nem rögzíthető felhasználó nélküli üzenet, vagyis „Vendég” felhasználói rekordok nem jöhetnek létre a végponton. Példa request body:

{

    "felhasznalo\_id" : 10,

    "uzenet" : "Itt egy példa üzenet."

}

A válasz sikeres rögzítés esetén 200, egyéb esetben 400-as státusszal és a probléma okával tér vissza, mint a „Message must be filled”.

* 1. Statisztikai végpontok */api/stat*

Az importált adatbázis adatokból, néhány kisebb statisztikát készítettünk, ezek érhetők itt el.

* + 1. /winner GET végpont

Ez a végpont visszaadja az összes rekordot a nyertes view-ból. A végpont a nyertesRepository.findAll() metódus segítségével kérdezi le az adatokat.

A képen szöveg, képernyőkép, Betűtípus, képernyő látható

Automatikusan generált leírásA <http://localhost:8080/api/stat/winner> url-re indított GET kéréssel a képen látható json érhető el.

* + 1. A képen szöveg, képernyőkép látható

       Automatikusan generált leírás/statistics GET végpont

Ezen a végponton az oldalon látható statisztikák közül kerül átadásra a versenyek száma, a pilóták száma, valamint egy json arrayben a nemzetekre összesített nyertesek száma.

A <http://localhost:8080/api/stat/statistics> url-re indított GET kérés response json-je látható a képen.

* 1. Hitelesítő végpont

Az alkalmazás hitelesítését Spring Security-vel oldottuk meg. Az oldalon a bejelentkezés a beépített módon történik, viszont egy REST végponton keresztül lehetőséget biztosítunk regisztrációhoz. A <http://localhost:8080/auth/registration> url-re küldött http POST kéréssel érhető el a regisztrációs végpont.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Automatikusan generált leírás

A Postman lehetőséget biztosít a kérés cUrl-ként történő másolására. A képen látható regisztrációt a következő cUrl-el lehet megvalósítani:

curl --location 'http://localhost:8080/auth/registration' \

--header 'Content-Type: application/json' \

--header 'Authorization: Basic dXNlcjph' \

--data-raw '{

"username" : "tesztmail@mailcim.hu",

"password" : "admin1234"

}'