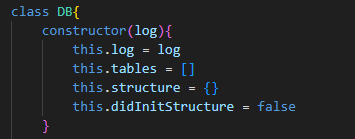
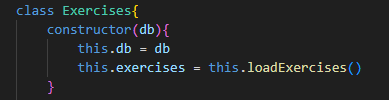
## x.x Az adatbázistól a megjelenítésig

### Inicializálás

Az alkalmazás elindításakor, mielőtt az API endpointok felállításra kerülnének, a program két objektumot is létrehoz, az adatbázis objektumot (*db.js*) és a gyakorlatok objektumot (*exercises.js*). Feladatuk egyszerű:



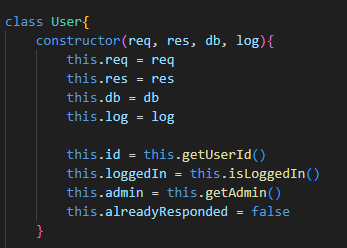
* A DB osztály felel minden adatbázis kérés elküldéséért, és a visszakapott értékek visszaadásáért, majd a kérés naplózásáért. Emellett eltárolja az adatbázis stuktúráját az admin oldal megfelelő működéséért.



* Az Exercises osztály tárolja az adatbázisban lévő gyakorlatokat. Ennek előnye, hogy nem kell minden alkalommal újra lekérni az adatbázisból. Amikor új gyakorlatot hoz létre vagy töröl egy adminisztrátor, a gyakorlatok listája is automatikusan frissül.

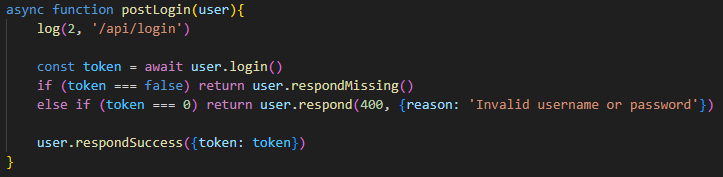
### Kérés feldolgozása

  
Miután a kérés beérkezik a szerver nyitott endpointjára, a kérés átadásra kerül egy *User objektumnak* és egy *kéréskezelő függvénynek*.  
**User osztály**

****

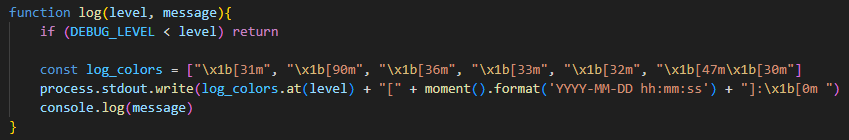
Minden kéréshez egy User objektum tartozik. Ez többek között eltárolja a kérést küldő felhasználó azonosítóját és jogosultságait (amennyiben be van jelentkezve). Minden kérés-specifikus adatbázis műveletet ez az objektum indít, feldolgoz, és ad vissza a szervernek válaszra kész formában.

### Kéréskezelő függvények



A kéréskezelő függvényeknek három feladata van. Előszőr is, naplózza a Bejövő kérés útvonalát. Ezután meghívja a User objektum megfelelő függvényét. Végül a függvény visszatérési értékéből visszaküldi a felhasználó felé a kérésre a választ, vagy éppen a megfelelő hibakódot és leírást.

**Naplózó függvény**



A naplózáshoz egy saját, primitív függvényt használunk. A függvény meghívásakor beállítunk egy szintet, -1-től 4-ig. Amennyiben a *.env* fájlban beállított szint legalább akkora, mint a naplózás szintje, az üzenet (a dátummal és időponttal együtt) ki lesz írva a konzolra.

