**MyTodo**

IDE: Visual Studio 2022 Community

Backend: C# .Net 6.0 és ASP.Net 6

Forntend: Angular 14 és Material

DB: MS SQL Server 2017, EF Core 6

Authentikáció: JWT Bearer

Authorizáció: Permission alapú

Teszt: Unit tesztek választott osztályhoz, Integrációs tesztek választott API-hoz

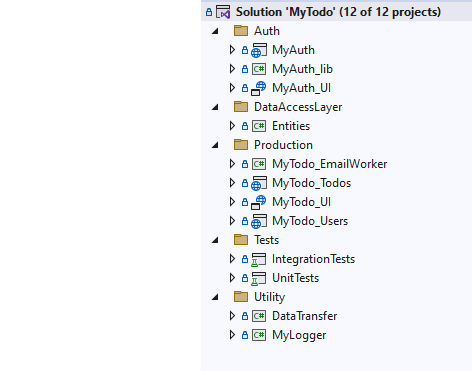
Log: Serilog alkalmazása saját absztrakcióval

DTO (Data transfer object): Automapper alkalmazása Entitás <-> Dto előállításra

Microservice: Két fő adathalmazhoz (User, Todo) külön API

Email: Lejáró Todo-k esetén Email küldés Windows service applikációval

Github repo: <https://github.com/salathcs/MyTodo>



**Solution projektek**

* Auth – authentikáció és authorizáció
  + MyAuth – Auth server API: Login, Registration és Validation
  + MyAuth\_lib – Class library: Auth szerver és Auth kliens logika kiemelve, tartalmazza a policy-ket is
  + MyAuth\_UI – Angular projekt: bejelentkezés és regisztráció
* DataAccessLayer – entitások
  + Entities – Class library: entitások és MyTodoContext (EF DbContext)
* Production – a termék
  + MyTodo\_EmailWorker – Windows service, konfigurálható késleltetéssel lekéri Todo-kat, melyek konfigurálható paraméter alapján lejárnak, és ezek esetén email küldésének megkísérlése a felhasználója számára
  + MyTodo\_Todos – Todo API: Crud metódusokkal, és manager controllerrel az egyéb logikákhoz
  + MyTodo\_UI – Angular projekt: bejelentkezett felhasználó Todo lista, vagy az összes felhasználó, Admin Permission esetén megtekinthető más felhasználóhoz tartozó Todo lista
  + MyTodo\_Users - User API: Crud metódusokkal, és manager controllerrel az egyéb logikákhoz
* Tests – Tesztek
  + IntegrationTests – Integrációs tesztek MyTodo\_Todos Crud API-hoz admin és általános jogkörökkel
  + UnitTests – Unit tesztek MyTodo\_Todos CrudService osztályhoz
* Utility – Segéd projektek
  + DataTransfer – Class library: DTO modellek és Automapper profilok
  + MyLogger – Class library: Serilog feletti absztrakció

**Konfiguráció**

Konfiguráció a különálló egységeknek backend oldalon appsettings.json fájlban:

**Általános:**

* **Serilog** – logger konfiguráció, egységesen alkalmaztam fájl és konzol logolást, ahol csak a fájlnév tér el a komponensek között.
* **ConnectionString** – DefaultConnection használatára van szükség, én a saját gépemen található MS SQL szervert használtam Trusted connection-nel.
* **AllowedHosts** – ASP.Net kulcs.

**MyAuth:**

* **TokenExpiration –** Login-kor előálló Bearer token lejárati ideje. Ezután új autentikációra van szükség.

**MyTodo\_EmailWorker:**

* **TimeDelayMinutes –** Todo lekérdezések közötti idő percekben.
* **TodoBeforeExpireMinutes –** Todo lejárati dátumának intervalluma, az aktuális időtől kiindulva percekben.
* **LoginName –** API login neve.
* **Password –** API jelszava.
* **LoginUrl –** Auth API Login endpoint-tal, bejelentkezéshez.
* **TodosUrl –** Todos API Manager GetByExpiration endpoint-tal, Todo lekérdezéshez.
* **UpdateEmailSentUrl –** Todos API Manager UpdateEmailSentOn endpoint-tal, Todo frissítéshez, hogy következő iterációban ne kérdezze le.
* **Mail\_From –** Küldött email feladója.
* **Mail\_Host –** Mail szerver.
* **Mail\_LoginName –** Mail szerver login név.
* **Mail\_Password –** Mail szerver login jelszó.

**MyTodo\_Todos / MyTodo\_Users:**

* **AuthServerValidationUrl –** Auth API Validate endpoint-tal, beérkező kérések validálásához.
* **AuthClientRequestValidCacheExpiration –** Validált kérésekről cache készül ezzel a lejárattal, így csökkentve a terhet a Validate endpoint-ról.

Konfiguráció frontend oldalon environment.ts / environment.prod.ts fájlokban.

**MyAuth\_UI:**

* **appUIUrl –** Login után átirányít a megadott URL-re, aminek a MyTodo\_UI applikáció címének kell lennie.

**MyTodo\_UI:**

* **authUIUrl –** Kijelentkezéssel, vagy token lejárat, esetleges hiba esetén error page-ről van lehetőség visszairányítást kérni újbóli Login-re. Ilyenkor az itt konfigurált címre irányít, aminek a MyAuth\_UI címének kell lennie.

**Futtatás**

A három API-t VS segítségével futtattam, ezek a Properties / launchSettings.json alapján vannak konfigurálva, hogy megtalálják egymást. Mindhárom Https konfigurált. MyTodo\_Todos, és MyTodo\_Users projektek működéséhez futtatni kell a MyAuth API-t.

A két frontend projektet külön futtattam konzolról. MyAuth\_UI 4200-as alapértelmezett porton indul, a MyTodo\_UI módosítva lett az angular.json-ben 4201-es portra. Ezt a konfiguráció is tükrözi. MyTodo\_UI működéséhez az összes itt felsorolt alkalmazás futtatására szükség van.

Az EmailWorker szintén futtatható VS segítségével. Működéséhez a MyTodo\_Todos API-ra van szükség.

**Adatbázis**

A képen szöveg látható

Automatikusan generált leírás

* EFMigrationsHistory – EF által a migrációk követésére.
* identities – Felhasználó identitása, bejelentkezéshez szükséges adatokat, felhasználónév és jelszót tartalmazza. Új felhasználó esetén létre kell hozni egy rekordot, törlése esetén az identitás rekordot is törölni kell.
* permissions – Jogosultságok tábla. Jelenleg egy rekordot az Admin jogot tartalmazza.
* userPermissions – Felhasználók és jogosultságok kapcsoló táblája, így valósítva meg a több-több kapcsolatot.
* users – alapvető felhasználói adatokat tartalmaz, pl: email.

EF Tools segítségével migrációkat hoztam létre, és frissítettem azokkal az adatbázist.

Az entitás kapcsolatokat inverz tulajdonságokkal képeztem le, és Lazy loading-ot használtam.

* Alapvető admin felhasználó seed-ből:
  + username: admin
  + pw: admin

**Folyamatok**

**MyAuth\_UI**

A felhasználó bejelentkezést, vagy regisztrációt hajthat végre. Bejelentkezés után rövidesen átirányításra kerül MyTodo\_UI-ra, az URL-hez hozzá csatolja a megfelelő adatokat, amiket a Login hívással kapott meg.

**MyTodo\_UI**

A bejelentkezett felhasználót a saját Todo listája fogadja. Itt használhatja a CRUD funkciókat. Lehetősége van a felhasználók megtekintésére is. CRUD funkciók csak részlegesen érhetők el itt. Új felhasználó felvételéhez a regisztrációs felületet kell használnunk.

Admin jogosultság esetén adhatjuk, vagy vehetjük el másik felhasználótól az admin jogot. Szintén admin jog esetén más felhasználók Todo listáit is megtekinthetjük, és szerkeszthetjük is.

**MyAuth**

Sikeres Login esetén AuthResult típusú választ ad:

A képen szöveg látható

Automatikusan generált leírás

A kéréseknél a token-t cookie-ban, vagy Authorization Header-ben várja.

A userId-re a frontend épít logikát: pl saját Todo lista lekérdezése.

Validate endpoint-ot a többi API hívja a saját beérkezett kéréseik validálásához. Ilyenkor a szerver validálja a token-t (authentikáció) és utána validálja a policy-t is a felhasználóra nézve. Siker esetén visszaadja a felhasználói Claim-eket.

Kliens oldalon sikeres validáció mentésre kerül cache-be, felhasználó és policy kulccsal, és a felhasználóhoz tartozó claim-ekkel.

Auth result swagger teszteléshez endless token-nel (10 év lejárattal), admin jogosultsággal:

{

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJNeVRvZG9KV1RTZXJ2aWNlVG9rZW4iLCJqdGkiOiI0Njk0NDRiMC00ODIyLTQ0MTgtOTIwZS04YmYzMDRjNDBiMTciLCJpYXQiOjE2NTg1NjMzNTEsInVuaXF1ZV9uYW1lIjoiYWRtaW4iLCJJZGVudGlmaWVyIjoiMSIsIkFkbWluUGVybWlzc2lvbiI6IiIsIm5iZiI6MTY1ODU2MzM1MSwiZXhwIjoxOTc0MTgyNTUxLCJpc3MiOiJNeVRvZG9KV1RTZXJ2aWNlU2VydmVyIiwiYXVkIjoiTXlUb2RvSldUU2VydmljZUNsaWVudCJ9.s78c8w8HahHjCiPLA\_Ak2AcVXW6AaY8fE1TfGUf2dEo",

"expiration": "2032-07-23T08:02:31.8367938Z",

"name": "MyAdmin",

"userId": 1

}

**Továbbfejlesztési ötletek**

* Password titkosítása, mind db, mind http oldalon.
* Frontend:
  + Nyelvválasztás
  + Spinner
  + Grid funkcióinak kiegészítése, pl: szűrés
  + Komponens kiemelés
* Tesztek:
  + Unit tesztek kiterjesztése a többi projektre
  + Integrációs tesztek folytatása MyTodo\_Todos API-ra és implementálása MyTodos\_Users-re.