Teszt Dokumentáció

- RepositoryTests
  - Mindegyik file elején van egy function, ami létrehozza a test file-t, és minden elindításnál egy új, random file nevet generál
  - BookingRepositoryTest
    - LoadAsync_FileDoesNotExist_ReturnsEmptyList
      - Ez a teszt hivatkozik egy olyan repo névre, ami nem létezik, azonban a LoadAsync függvény, „amivel ezt megteszi” nem dob erre hibát, hanem létrehoz egy új üres listát. Azt nézi meg, hogy true-t adott-e vissza a LoadAsync és, hogy az újonnan létrehozott lista üres-e.
    - AddAsync_AddsBooking_AndPersistsToFile
      - A teszt létrehozza a repo-t, ahhoz hozzáad egy foglalást, majd a file-, amibe ez el lett mentve, beolvassa a repo2-be, majd ellenőrzi, hogy benne van-e az a booking, amit ez előbb elmentettünk.
    - DeleteAsync_DeletesBooking_AndPersistsToFile
      - Elment 2 Bookingot, és megnézi, hogy mindkettő el lett-e mentve, majd az elsőt kitörli. Ezután szintén egy új repoba beolvassa ugyanazt a file-t, és megnézi a tartalmát, hogy 1 dolog van-e csak benne, és hogy annak mi az Id-ja.
  - PlacesRepositoryTest
    - LoadAsync_FileDoesNotExist_ReturnsEmptyList
      - Ez a teszt hivatkozik egy olyan repo névre, ami nem létezik, azonban a LoadAsync függvény, „amivel ezt megteszi” nem dob erre hibát, hanem létrehoz egy új üres listát. Azt nézi meg, hogy true-t adott-e vissza a LoadAsync és, hogy az újonnan létrehozott lista üres-e.
    - AddAsync_AddsPlace_AndPersistsToFile
      - A teszt létrehozza a repo-t, ahhoz hozzáad egy place-t, majd a file-t amibe ez el lett mentve, beolvassa a repo2-be, majd ellenőrzi, hogy benne van-e az a place amit ez előbb elmentettünk.
    - DeleteAsync_DeletesPlace_AndPersistsToFile
      - Elment 3 Place-t, és megnézi, hogy mindhárom el lett-e mentve, majd az elsőt kitörli. Ezután szintén egy új repoba beolvassa ugyanazt a file-t, és megnézi a tartalmát, hogy 2 dolog van-e csak benne, és hogy annak mi az Id-ja.
  - UserRepositoryTest
    - LoadAsync_FileDoesNotExist_ReturnsEmptyLis
      - Ez a teszt hivatkozik egy olyan repo névre, ami nem létezik, azonban a LoadAsync függvény, „amivel ezt megteszi” nem dob erre hibát, hanem létrehoz egy új üres listát. Azt nézi meg, hogy true-t adott-e vissza a LoadAsync és, hogy az újonnan létrehozott lista üres-e.

        - AddAsync_AddsUser_AndPersistsToFile
            - A teszt létrehozza a repo-t, ahhoz hozzáad egy user-t. Majd a file-t amibe ez el lett mentve, beolvassa a repo2-be, majd ellenőrzi, hogy benne van-e az a place amit ez előbb elmentettünk.
        - DeleteAsync_DeletesUser_AndPersistsToFile
            - Elment 3 user-t, és megnézi, hogy mindhárom el lett-e mentve, majd az elsőt kitörli. Ezután szintén egy új repo-ba beolvassa ugyanazt a file-t, és megnézi a tartalmát, hogy 2 dolog van-e csak benne, és hogy annak mi az Id-ja.
- InMemoryRepository
    - Ugyanazt csinálja, mint a korábban megírt *Repository osztályok, azonban itt nem file-ba, hanem csak listába mentenek, és csak ramban vannak meg, nem mentődnek el sehova.
- ServireTests
    - AuthServiceTest
        - Setup
            - Inicializál egy RAM-ban létező InMemoryUserRepositoryt, amivel dolgozik a test file és hozzáad egy user-t.
        - AuthenticateAsync_ReturnsUser_WhenPasswordCorrect
            - megnézi, hogy be lehet-e lépni az AuthenticateAsyn függvénnyel, és visszaadja-e a felhasználót.
        - AuthenticateAsync_ReturnsNull_WhenPasswordWrong
            - megnézi, hogy be lehet-e lépni az AuthenticateAsyn függvénnyel, és visszaadja-e a felhasználót. Azonban itt null-t kell visszaadnia, mert rossz a jelszó.
        - RegisterGuestAsync_Throws_WhenUserNameAlreadyExists
            - megpróbál használót regisztráltatni, mivel azonban már létezik ilyen felhasználónév, így hibát dob ki.
        - ChangePasswordAsync_ChangesPassword_WhenOldMatches
            - megnézi, hogy be tud-e lépni a felhasználó fiókjába, hogy tud-e jelszót változtatni, és utána az új jelszóval szintén be tud-e lépni.
        - ChangePasswordAsync_Fails_WhenOldWrong
            - megnézi, hogy a régi jelszóval be tud-e lépni. Aztán megpróbál jelszót változtatni egy hibás jelszóval, azonban így false-ot dob vissza, mert hibás a régi jelszó.
    - BookingServiceTest
        - Setup
            - ez inicializálja a user, place és booking repo-t, és beletesz 1 usert és 2 place-t
        - CreateBookingForPlaceAsync_Throws_WhenOverlapping
            - Ez a teszt elment egy booking-ot, és utána megpróbálja szinte ugyanarra az időszakra lefoglalni ugyanazt a helyet, és erre hibát kell dobnia, mert nem lehet átfedés.

        - CreateBookingForPlaceAsync_Throws_WhenGuestCountExceedsCapacity
            - Ez a teszt azt nézi meg, hogy foglaláskor ellenőrizze az alkalmazás, hogy a helynek amit le szeretnének foglalni, van-e guest limitje es ha igen, akkor túllépik-e.
    - PlacesServiceTest
        - Setup
            - Létrehoz kettő place-t és inicializálja a repokat
        - DeletePlaceAsync_Fails_IfActiveBookingExists
            - Ez a teszt azt nézi meg hogy a már lefoglalt place-t lehet-e törölni, és mivel igen, false értékkel tér vissza.
        - DeletePlaceAsync_Succeeds_IfNoActiveBooking
            - Ez a teszt kitörli a place-t és true-val tér vissza, valamint megnézi, hogy megtalálható-e még az előbb törölt place, és oda false értéket vár.