

Tesztelési dokumentáció

Unit tesztek

1. UserServices tesztek

1.1 UserExistsInRepo

Leírás:

Ellenőrzi, hogy a **UserService.AddUser()** függvény megfelelően eltárolja-e a megadott felhasználót, majd hogy a **UserService.ContainsEmail()** képes-e megtalálni az email-címet a tárolt adatok között.

1.2 SetLoggedInUser

Leírás:

Felhasználót ad hozzá a rendszerhez, majd beállítja őt **LoggedInUser**-ként. Ellenőrzi, hogy ez a property megfelelően frissült-e.

1.3 ClearLoggedInUser

Leírás:

Egy beállított LoggedInUser-t töröl a **UserService.ClearLoggedInUser()** segítségével, majd ellenőrzi, hogy a **LoggedInUser** értéke valóban **null**-ra váltott-e.

2. InputValidation tesztek

2.1 EmailEmpty

Leírás:

Ellenőrzi, hogy a rendszer megfelelően kezeli-e az **üres email**-eket.

2.2 EmailWithoutAtSign

Leírás:

Ellenőrzi, hogy a rendszer megfelelően kezeli-e azokat az **email**-eket, amelyek **nem tartalmazznak "@" szimbólumot**.

2.3 EmailWithoutDot

Leírás:

Ellenőrzi, hogy a rendszer megfelelően kezeli-e azokat az **email**-eket, amelyek **nem tartalmazznak "." szimbólumot**.

2.4 EmailWithOnlyOneTLDChr

Leírás:

Ellenőrzi, hogy a rendszer megfelelően kezeli-e azokat az **email**-eket, amelyek **Top-Level-Domain-je csak egy karaktert** tartalmaz.

2.5 EmailIsCorrect

Leírás:

Ellenőrzi, hogy a rendszer megfelelően kezeli-e a **helyesen megadott email**-eket.

2.6 PasswordLengthLessThanFiveChrs

Leírás:

Ellenőrzi, hogy a rendszer megfelelően kezeli-e azokat a **jelszavakat**, amelyek **kevesebb, mint 5 karakterből állnak**.

2.7 PasswordContainsSymbol

Leírás:

Ellenőrzi, hogy a rendszer megfelelően kezeli-e azokat a **jelszavakat**, amelyek **tartalmaznak szimbólumot**.

2.8 PasswordContainsOnlyLowercaseChrs

Leírás:

Ellenőrzi, hogy a rendszer megfelelően kezeli-e azokat a **jelszavakat**, amelyek **csak kisbetűket tartalmaznak**.

2.9 PasswordWithNoNumbers

Leírás:

Ellenőrzi, hogy a rendszer megfelelően kezeli-e azokat a **jelszavakat**, amelyek **nem tartalmaznak számot**.

2.10 PasswordIsCorrect

Leírás:

Ellenőrzi, hogy a rendszer megfelelően kezeli-e azokat a **jelszavakat**, amelyek **minden feltételt teljesítenek**.

Terheléses teszt

Leírás:

A terheléses teszt (Load Test) során világossá vált a Dictionary mekkora előnyt nyújt más adatszerkezetekhez képest, $O(1)$ lookup, azaz konstans elérési időt nyújt az adatok eléréséhez. A `UserService.AddUser()` metódusa lassúnak bizonyul akár 100 felhasználó esetén is, a jövőben ezen érdemes lenne javítani (például nem json fájlformátumot használva)!