TESTARE UNITARĂ ÎN JAVASCRIPT

Membrii echipei:

1. Mihai Alexandru-Mario
2. Ojoc Diana-Cristiana
3. Cîrstea Ruxandra-Gabriela

AGENDA

- OBIECTIV
- FUNCȚIA TESTATĂ
- FRAMEWORK-URI DE TESTARE
- SCENARII TESTATE
- COMPARAȚIE: TEST MANUAL VS AI
- MUTATION TESTING CU STRYKER

OBIECTIV

- Testarea unitară este o etapă esențială în procesul de dezvoltare software, având ca scop verificarea comportamentului izolat al fiecărei componente din cod.

- Funcția **removeUserFromConversations** este responsabilă de gestionarea eliminării unui utilizator din toate conversațiile sale, având diverse ramificații logice în funcție de opțiunile primite ca parametru.

FUNCȚIA TESTATĂ

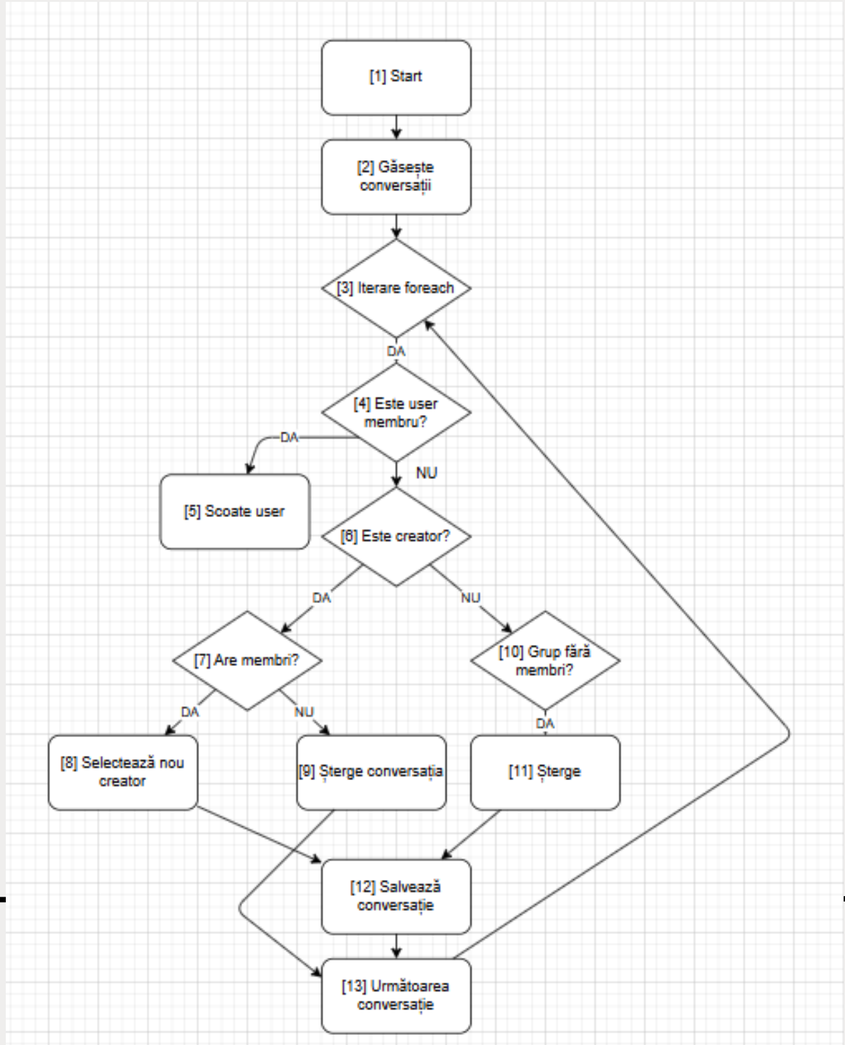
removeUserFromConversations(userId, options = { removeEmptyConversations: true, newCreatorSelection: "random" }, logger = console)

Elimină un utilizator din toate conversațiile

Schimbă creatorul dacă e necesar

Șterge conversațiile fără membri

Emite loguri și tratează erori (ex. ID invalid)



FRAMEWORK-URI DE TESTARE COMPARATE

Criteriu	Jest	Mocha	Jasmine
Popularitate	Foarte popular (Facebook, Meta)	Popular – folosit în multe proiecte open-source	Mai puțin popular
Configurare	Minimală – funcționează out-of-the-box cu zero configurare	Medie – necesită configurare manuală pentru assert/mock	Medie – include multe funcții dar poate deveni complexă
Mocking integrat	Da	Nu – trebuie integrat cu alt tool	Parțial – include funcționalitate de mocking de bază
Raport Coverage	Integrat	Cu plugin	Cu plugin
Async/Await	Suport complet – testare nativă async/await	Suport complet – dar necesită gestionarea explicită a obiectelor de tip Promise	Suport parțial – mai vechi, poate necesita workaround-uri

JEST

Avantaje:

Setup rapid, configurare minimă, mocking și snapshot inclus

Dezavantaje:

Poate fi mai lent pe proiecte mari, consumă mai multă memorie

1. Inițializare proiect node:

```
npm init -y
```

2. Instalare module necesare:

```
npm install jest mongodb-memory-server @types/jest mongoose uuidv4
```

3. Configurare Jest în fișierul `package.json`:

```
{
  "name": "proiect_tss",
  // [configurație parțial omisă]
  "scripts": {
      "test": "jest --runInBand"
   },
  "jest": {
      "verbose": true,
      "testEnvironment": "node",
      "clearMocks": true,
      "moduleFileExtensions": [
         "js",
         "json"
      ],
      "testMatch": [
         "**/tests/**/*.test.js",
         "**/?(*.)+(spec|test).js"
      ],
      "collectCoverage": true,
      "coverageDirectory": "coverage",
      "coveragePathIgnorePatterns": [
         "/node_modules/",
         "/tests/"
      ]
   }
}
```

SCENARII TESTATE

Eliminare simplă a unui utilizator

- Pornind de la 2 membri, îl şterge pe unul şi verifică să mai fie un singur membru rămas.

Eliminare din toate conversaţiile

- Un utilizator prezent în 3 conversaţii; se verifică că nu mai apare în niciuna.

Alegere creator (admin) nou Opțiunea “first”

- Dacă se șterge creatorul și mai rămân membri, noul creator devine primul din listă.

Edge cases - ID invalid

- Apelarea cu un ID invalid trebuie să emită logger.error.

COMPARAȚIE: TEST MANUAL VS AI

Aspect	Fișier inițial	Fișier generat cu AI
Acoperire teste	Mai larg, include mai multe cazuri limită specifice, testarea opțiunilor implicite și gestionarea conversațiilor goale în diverse condiții, testarea logging-ului.	Funcționalități de bază, mai puține cazuri limită explicite, fără testarea opțiunilor implicite detaliată sau a logging-ului.
Modularitate test funcțional	Are `createTestConversation()` cu valori implicite (utilizează valoarea creator furnizată sau primul membru din `members` ca fallback) și flexibilitate (permite specificarea numelui grupului la apelare).	Are `createConversation()` simplificat (utilizează doar valoarea creator furnizată), fără opțiuni suplimentare (numele grupului este fix).
Structură teste	Organizată pe categorii: funcționalități de bază, realocare creator, edge cases etc.	Toate testele într-un singur `describe`, mai puțin organizate.
Cazuri limită (edge cases)	ID invalid, creator care nu e membru, user fără conversații.	Doar unul: creator care nu e membru.
Variabile	Clară și contextuală (ex: `userId`, `otherUser1`, `mockLogger`)	Mai simplă și uneori mai vagă (`uid`, `target`, `creatorOnly`)
Complexitate	Mare – ideal pentru validare completă	Redusă – potrivit pentru faze inițiale ale dezvoltării sau debugging rapid

COMPARAȚIE: TEST MANUAL VS AI

```
PASS tests/removeUserFromConversations.test.js
  √ [Test 1] conectare la baza de date MongoDB in memorie (10 ms)
  Functii de baza
    √ [Test 2] elimina utilizatorul din conversatii (32 ms)
    √ [Test 3] elimina utilizatorul conectat din toate conversatiile (44 ms)
    √ [Test 4] gestionare conversatie cu multi membri (15 ms)
  Alegere creator (admin) nou
    √ [Test 5] realocare creator cand utilizatorul era creator si raman membri (selecteaza primul) (14 ms)
    √ [Test 6] selecteaza aleatoriu noul creator cand utilizatorul era creator (14 ms)
    √ [Test 7] verifica comportamentul implicit al optiunilor (12 ms)
    √ [Test 8] pastreaza conversatia goala daca removeEmptyConversations este false si userul nu era creator (12 ms)
    √ [Test 9] pastreaza conversatia goala daca removeEmptyConversations este true si userul nu era creator (12 ms)
  Gestionare conversatii fara membri
    √ [Test 10] sterge conversatia cand nu mai raman membri si removeEmptyConversations e true (11 ms)
    √ [Test 11] sterge conversatia cand nu mai raman membri si removeEmptyConversations e false (11 ms)
  Edge cases
    √ [Test 12] gestionare cand utilizatorul este creator dar nu este membru (12 ms)
    √ [Test 13] gestionare cand utilizatorul nu este in nicio conversatie (3 ms)
    √ [Test 14] logheaza eroare cand se ofera un id invalid (1 ms)

-----------------|---------|----------|---------|---------|-------------------
File             | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----------------|---------|----------|---------|---------|-------------------
All files        |     100 |      100 |     100 |     100 |
 models          |     100 |      100 |     100 |     100 |
  Conversation.js |     100 |      100 |     100 |     100 |
 src             |     100 |      100 |     100 |     100 |
  userCleanup.js |     100 |      100 |     100 |     100 |
-----------------|---------|----------|---------|---------|-------------------
Test Suites: 1 passed, 1 total
Tests:       14 passed, 14 total
Snapshots:   0 total
Time:        2.261 s
Ran all test suites.
```

```
PASS tests/removeUserFromConversationsAi.test.js
  removeUserFromConversations logic
    √ should exclude user from member list (38 ms)
    √ should delete conversation if user was only member and creator (17 ms)
    √ should assign new admin using first member (14 ms)
    √ should randomly pick new creator if not specified (15 ms)
    √ removes user from multiple conversations (22 ms)
    √ handles when user is creator but not a member (13 ms)

-----------------|---------|----------|---------|---------|-------------------
File             | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----------------|---------|----------|---------|---------|-------------------
All files        |   97.14 |    93.75 |     100 |   96.96 |
 models          |     100 |      100 |     100 |     100 |
  Conversation.js |     100 |      100 |     100 |     100 |
 src             |   96.66 |    93.75 |     100 |   96.42 |
  userCleanup.js |   96.66 |    93.75 |     100 |   96.42 | 81
-----------------|---------|----------|---------|---------|-------------------
Test Suites: 1 passed, 1 total
Tests:       6 passed, 6 total
Snapshots:   0 total
Time:        1.133 s, estimated 2 s
Ran all test suites.
```

MUTATION TESTING CU STRYKER

Testarea prin mutanți ajută la identificarea testelor slabe și la îmbunătățirea acoperirii, chiar și atunci când acoperirea liniilor de cod pare suficientă.

Configurare Stryker mutation tester în fișierul `stryker.conf.js`:

```
/**
* @type {import('@stryker-mutator/api/core').StrykerOptions}
*/
module.exports = {
mutate: ["src/**/*.js"],
testRunner: "jest",
reporters: ["html", "clear-text", "progress"],
coverageAnalysis: "off",
jest: {
 projectType: "custom",
},
};
```