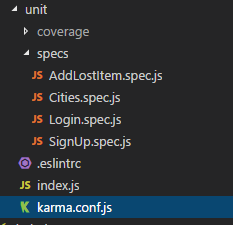
# TDD – exemplificare în cadrul unui instrument

Proiectul fiind bazat pe TDD necesita implementarea testelor unitare inaintea realizarii task-urilor propriu-zise. Pentru a realiza testele unitare este nevoie de un tool special pentru aceasta directie.

Fiind vorba de un proiect realizat in framework-ul Vue.js, testele sunt realizate in JavaScript. Biblioteca utilizata pentru realizarea acestor teste este karma.js (karma-mocha). Acest pachet este instalat cu ajutorul npm (node package manager).

Biblioteca karma.js consta in scrierea mai multor fisiere ce reprezinta firecare o suita de teste unitare, avand extensia .spec.js. Acesta are un fisier de configurare, de unde se pot selecta fisierele ce urmeaza a fi rulate.



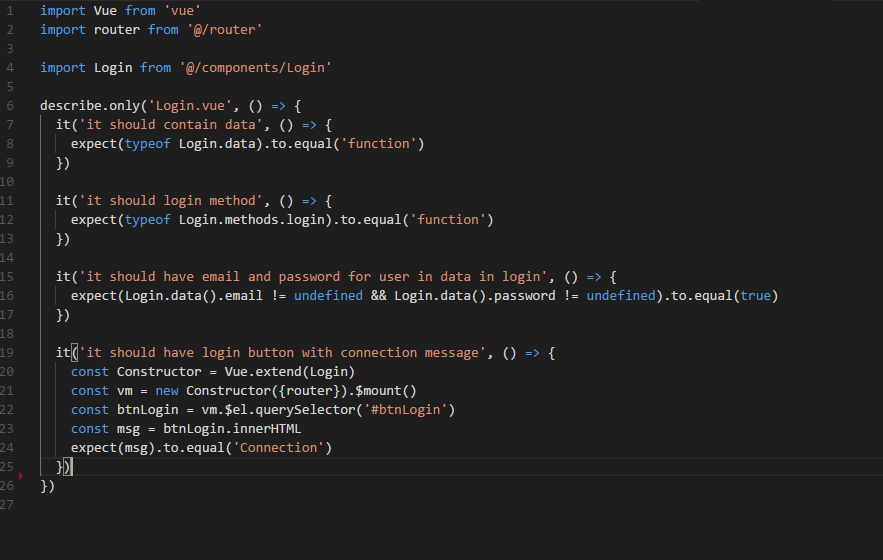
Comanda la care aceste teste se ruleaza este: *npm run unit*, comanda ce verifica mai intai fisierul de configurare: karma.conf.js, iar in continuare este rulat fisierul *index.js*, unde este creat mediul de test, un fisier similar celui de baza din aplicatia principala, aplicatia client.

In fisierul index.js se specifica ce teste sunt rulate, cele unitare de verificare (pot fi selectate ce fisiere sa se ruleze) sau/si testele de *coverage*, ce analizeaza cat la suta din cod este atins.

**Scrierea unui test**

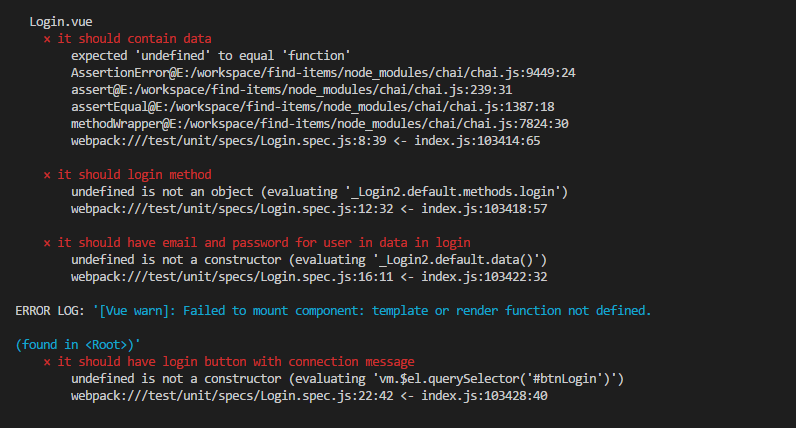
O suita de teste este scrisa in fisierul cu extensia .spec.js, spre exemplu: Login.spec.js. In acest fisier trebuie importate componentele necesare testului pentru a putea rula si componenta ce va fi testata. Framework-ul folosit prezinta metoda ***describe***ce primeste 2 parametrii: numele fisierului ce va fi testat si o functie ce va contine testele propriu zise pentru componenta respectiva. Apelul in plus al metodei *only* ne permite rularea doar acestei suite de teste din cele 4 (AddLostItem, Cities, Login si SignUp).

Metoda ***it*** din cadrul framework-ului folosit ne permite scrierea unui test pentru o anumita functionalitate, primind ca parametrii: mesajul testului si o functie ce contine testul propriu-zis.

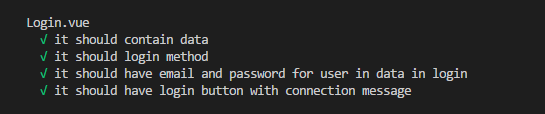


Metoda ***expect*** continuata de **to.equal** verifica conditia pusa, iar in caz de eroare testul pica. De exemplu: expect(typeof Login.data).to.equal(‘function’) -> verifica daca tipul Login.data este o functie, in caz de esec testul pica, iar in caz contrar testul este validat cu succes.

La inceput, componenta *Login.vue* nefiind implementata, testele vor esua.

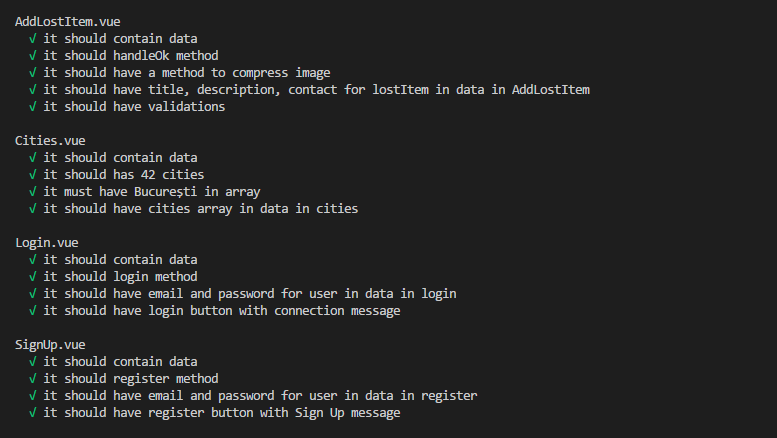


In urma implementarii componentei Login.vue si a functionalitatilor necesare testele au fost rulate din nou.



Se observa astfel ca functionalitatile testate in suita de mai sus au fost implementate cu succes.

In mod similar au fost realizate testele si pentru componentele: AddLostItem.vue, Cities.vue si SignUp.vue.



Rezultatele testelor ne arata ca functionalitatile au fost implementate cu succes pentru componentele prezentate. Bineinteles, testele nu ne garanteaza lipsa posibilelor erori in utilizare, dar ne ajuta in evitarea existentei acestora intr-un numar mare.