

BankingApp-JPA-Auth0

BankingApp Java MVC JPA + Auth0 service. Integration + Unit tests using JUnit. 2 Java running configurations - MySQL (prod) + H2 (test)

Aplicatia BankingApp este o aplicatie bancara cu rolul de a gestiona conturile clientilor. In acest sens, platforma dispune de mai multe banci la care clientii isi pot deschide conturi prin semnarea de contracte. Odata deschis un cont, acestuia ii este asignat un card. Clientul are la dispozitie alegerea unei oferte din lista creditelor oferite de banci, cu conditia ca acesta sa aiba deja un cont deschis la acea banca. Clientul poate initia plati din conturile sale pentru plata creditelor sau poate initia depuneri in conturi. Administratorul platformei este responsabil de adaugarea bancilor si ofertelor de credite. Tot el este responsabil de moderarea continutului de pe platforma.

Caracteristici principale ale proiectului:

- Persistenta datelor folosind JPA
- Arhitectura MVC
- Tratarea exceptiilor si a datelor din formulare
- 2 profiluri de rulare, fiecare profil avand baza sa de date. MySQL este baza de date folosita pentru mediul principal de lucru, iar cea H2 pentru mediul de test
- Folosirea serviciului Auth0 pentru autentificare/autorizare
- Thymeleaf drept engine pentru SSR
- Logging pentru teste
- Coverage cu JaCoCo
- Paginare si sortare pentru plati si depuneri
- Relatii intre tabele de tipurile OneToOne, ManyToOne, ManyToMany
- Operatii CRUD: Create, Read, Update, Delete
- Teste de integrare cu JUnit pentru serviciul auth0
- Teste unitare cu JUnit pentru controllere si servicii

Euristici:

- La fiecare request se interogheaza serviciul auth0 pentru obtinerea datelor despre utilizatorul care a facut requestul

- Se verifica daca userul care a facut requestul este logat sau nu, daca nu e logat va fi redirectionat catre homepage si se va cere logarea sa
- Paginile se randeaza dinamic in functie de rolul utilizatorului si a informatiilor din baza de date injectate de controllere
- Pe pagina de profil se pot actualiza datele despre utilizator, rolul utilizatorului poate fi schimbat doar din dashboardul auth0
- Verificarea datelor din formulare + clase pentru tratarea exceptiilor (ex: PhoneNumberValidator)
- Clientul isi poate crea 2 tipuri de conturi: cont curent si cont de depozit
- Un client poate avea maxim 2 conturi deschise la o banca (unul din fiecare tip)
- La crearea unui cont, se verifica daca intre client si banca exista deja un contract. Daca nu exista un contract, se creeaza contractul si noul cont, altfel se adauga contul la contractul existent
- La crearea unui cont se asigneaza automat un card de credit. Se foloseste un generator de carduri care genereaza coduri IBAN si CVV valide
- Stergerea se face cascadata, daca exista un singur cont pe un contract si contul este sters, se sterge si contractul si cardul asignat contului. La fel si daca este sters cardul asignat unicului cont de pe contract. La stergerea contractului sunt sterse toate conturile deschise in baza contractului
- Daca o banca este stearsa, vor fi sterse toate contractele (implicit toate conturile si cardurile) deschise la acea banca
- Paginile pentru plati si depuneri au cate 5 intrari/pagina si dispun de 4 filtre: sortare asc + desc dupa valoarea platii/depunerii si sortare asc + desc dupa numele bancii