Testy BE

Wyniki testów

Pomyślne: T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15 T16 T17 T18 T19 T20

Scenariusze testowe

T1 - Użytkownik tworzy unikalne konto

1. Wywołaj PUT /auth z argumentami login: test5, password: test5, password_confirmation: test5
2. System wykonuje 302 Redirect i wywołuje GET /users
3. System zwraca 200 Success.

T2 - Użytkownik tworzy konto o istniejącym loginie

1. Wywołaj PUT /auth z argumentami login: test5, password: test5, password_confirmation: test5
2. System zwraca 409 Conflict.

T3 - Użytkownik tworzy konto o słabym haśle

1. Wywołaj PUT /auth z argumentami login: test5, password: admin, password_confirmation: admin
2. System zwraca 400 Bad Request z "password": is too simple.

T4 - Użytkownik tworzy konto o krótkim haśle

1. Wywołaj PUT /auth z argumentami login: test5, password: a, password_confirmation: a
2. System zwraca 400 Bad Request z "password" with value "a" fails to match the required pattern: /^[a-zA-Z0-9]{3,40}$/

T5 - Użytkownik tworzy konto o długim haśle

1. Wywołaj PUT /auth z argumentami login: test5, password: 123456789012345678912345678912345678912345678, password_confirmation: 123456789012345678912345678912345678912345678 "password" with value "123456789012345678912345678912345678912345678" fails to match the required pattern: /^[a-zA-Z0-9]{3,40}$/
2. System zwraca 400 Bad Request z "password" with value "a" fails to match the required pattern: /^[a-zA-Z0-9]{3,40}$/

T6 - Użytkownik podaje różne hasła

1. Wywołaj PUT /auth z argumentami login: test5, password: admin, password_confirmation: admin5
2. System zwraca 400 Bad Request z "password_confirmation": passwords do not match.

T7 - Użytkownik nie podaje potwierdzenia hasła

1. Wywołaj PUT /auth z argumentami login: test5, password: admin, password_confirmation: admin5
2. System zwraca 400 Bad Request z "password_confirmation": is required.

T8 - Użytkownik loguje się niepoprawnymi danymi

1. Wywołaj POST /auth z argumentami login: test5, password: admin
2. System zwraca 401 Unauthorized.

T9 - Użytkownik wylogowuje się

1. Wywołaj DELETE /auth
2. System zwraca 200 Success.

T10 - Użytkownik loguje się poprawnymi danymi

1. Wywołaj POST /auth z argumentami login: test5, password: test5
2. System zwraca 200 Success.

T11 - Właściciel przesyła zaszyfrowany dokument

1. Przygotuj zakodowany aes-256-ctr dokument.
2. Wywołaj PUT /documents z argumentami name: test, content: zakodowana treść dokumentu.
3. System zwraca 200 Success.

T12 - Właściciel sprawdza uprawnienia do dokumentu

1. Wywołaj GET /permissions
2. System zwraca 200 Success i obiekt permissions.

T13 - Właściciel sprawdza uprawnienia do zaszyfrowanego dokumentu

1. Wywołaj GET /permissions/1
2. System zwraca 200 Success i obiekt permissions.

T14 - Właściciel odszyfrowuje dokument

1. Wywołaj GET /documents/1
2. System zwraca 200 Success i obiekt document.

T15 - Właściciel udostępnia dokument

1. Wywołaj PUT /shares z argumentami: id: 1, login: test
2. System zwraca 200 Success i obiekt shares.

T16 - Właściciel listuje swoje udostępnienia

1. Wywołaj GET /shares.
2. System zwraca 200 Success i obiekt shares.

T17 - Właściciel wysyła swój klucz publiczny

1. Wywołaj POST /shares/1/0 z argumentem publicKey: XXX
2. System zwraca 200 Success.

T18 - Właściciel wysyła zaszyfrowane hasło

1. Wywołaj POST /shares/1/2 z argumentami: publicKey: XXX, crypted: XXX
2. System zwraca 200 Success.

T19 - Partner pobiera zaszyfrowane hasło

1. Wywołaj POST /shares/1/3 z argumentem publicKey: XXX
2. System zwraca 200 Success.

T20 - Właściciel ustawia timer dla partnera

1. Wywołaj PUT /timer/permissions z argumentami: sec: 3, id: 1
2. System zwraca 200 Success.