32 PASSED

100% passed
32 executed

29 seconds ago
last run

10 seconds
duration

GitHub Actions
Git Ref 76634ea
Job 18680447840

Linux

OpenJDK 64-Bit Server VM 17.0.16+8

cucumber-jvm 7.19.0

Search with text or @tags

You can search with plain text or Cucumber Tag Expressions to filter the output

✓ classpath:features/accountsDelete.feature

Feature: Check BOOLEANS before deleting Accounts by accountId

Background:

✓ **Given** the system is ready and i log with email "admin@admin.com" and password "1234"

@NoCleanup
Scenario: Delete blocked account

✓ **Given** a customer has an account with id 170
✓ **When** the customer tries to delete the account
✓ **Then** The customer gets a 400 status response and message: "Error: Account with id 170 is blocked"

@NoCleanup
Scenario: Delete account in debt

✓ **Given** a customer has an account with id 171
✓ **When** the customer tries to delete the account
✓ **Then** The customer gets a 400 status response and message: "Error: Account with id 171 is in debt"

@NoCleanup
Scenario: Delete account in debt and blocked

✓ **Given** a customer has an account with id 172
✓ **When** the customer tries to delete the account
✓ **Then** The customer gets a 400 status response and message: "Error: Account with id 172 is in debt and blocked"

✓ classpath:features/accountsLogin.feature

Feature: Accounts login endpoints.

Background:

✓ **Given** the system is ready and i log with email "john.doe@example.com" and password "password123"

Scenario: Logged account information

- ✓ **When** i request this users account information
- ✓ **Then** i should receive the code 200

Scenario: Logged account amount

- ✓ **When** i request this users account amount
- ✓ **Then** i should receive the amount

✓ classpath:features/createCards.feature

Feature: Testing the cards creation registering a new user with random parameters.

Scenario Outline: A customer registers, creates N Accounts and M cards associated to those accounts.

- ✓ **Given** The customer registers with <naccounts> accounts, <ncards> cards and an initial amount of <amount>
- ✓ **When** The customer logs in with their register credentials
- ✓ **And** The customer checks their cards
- ✓ **Then** The customer gets a <status> status response

Examples:

	naccounts	ncards	amount	status
✓	1	1	140	200
✓	1	0	20	200
✓	1	4	130	200
✓	2	2	231	200

Scenario: Creating a card to your own account

- ✓ **Given** I have logged in with email "john.doe@example.com" and password "password123"
- ✓ **When** I create a card to the account: 1
- ✓ **Then** The customer gets a 200 status response and message: "Card created successfully"

Scenario: Creating a card to other one account

- ✓ **Given** I have logged in with email "john.doe@example.com" and password "password123"
- ✓ **When** I create a card to the account: 2
- ✓ **Then** The customer gets a 400 status response and message: "Error creating card: you can only create card to your account"

✓ classpath:features/customerRegistrationUpdate.feature

Feature: Customer Registration and Update

Scenario: A customer registers with a new account, updates their name and credentials, and verifies the changes

- ✓ **Given** The customer registers with random email, name, surname and password
- ✓ **And** The customer logs in with their register credentials
- ✓ **When** The customer updates their name to "Rafa" and surname "Laguna"

- ✓ **And** The customer updates their email to "rafa.laguna6@example.com" and password to "newPassword123"
- ✓ **Then** The customer’s name, surname, email and password have been updated "successfully"

✓ classpath:features/deleteCustomer.feature

Feature: Delete a customer

Background:

- ✓ **Given** the system is ready for user authentication

@NoCleanup

Scenario: Register a new customer with a new account and delete his customer registration by id

- ✓ **Given** I have registered with name "paula", surname "Calvente", email "paula@example.com" and password "password"
- ✓ **And** I login with email "admin@admin.com" and password "1234"
- ✓ **When** The customer deletes his customer registration by id
- ✓ **Then** The customer gets a 200 status response

✓ classpath:features/feature1.feature

Feature: User authentication

As a user, I want to be able to register, login, and logout in the application.

Background:

- ✓ **Given** the system is ready for user authentication

Scenario: Register a new customer

- ✓ **When** I register with name "John", surname "Doe", email "probando5@example.com" and password "password123"
- ✓ **Then** I should receive a message "You have registered successfully."

Scenario: Login with valid credentials

- ✓ **Given** I have registered with name "John", surname "Doe", email "probando6@example.com" and password "password123"
- ✓ **When** I login with email "probando6@example.com" and password "password123"
- ✓ **Then** I should receive a message "Correct authentication"

Scenario: Login with invalid credentials

- ✓ **When** I login with email "invalid@example.com" and password "wrongpassword"
- ✓ **Then** I should receive a message "Invalid credentials"

Scenario: Logout after logging in

- ✓ **Given** I have logged in with email "probando3@example.com" and password "password123"
- ✓ **When** I log out
- ✓ **Then** I should receive a message "Logged out successfully. Cookies cleared."

✓ classpath:features/getTransfers.feature

Feature: Get Transfers History given an account

Background:

- ✓ **Given** the system is ready and i log with email "john.doe@example.com" and password "password123"

Scenario: Logged customer accounts

- ✓ **When** i request this users account information
- ✓ **Then** i should receive the code 200

Scenario Outline: All logged customer main account Transfers

- ✓ **When** i request all transfers with accountId <accId>
- ✓ **Then** i should receive the code <code>
- ✓ **And** i should receive all of the transfers or an error message

Examples:

	accId	code
✓	1	200
✓	2	403
✓	99999	400

✓ classpath:features/lessThan1kTransfer.feature

Feature: Check Transfer Not Enough Money in Sender User

Scenario Outline: A customer with an account with less than 1000€ does a transfer to another user of 1500€

- ✓ **Given** The customer registers with <numberOfAccounts> accounts, <ncards> cards and an initial amount of <money>
- ✓ **And** The customer logs in with their register credentials
- ✓ **When** The customer make a transfer with their main account and transferAmount <transferAmount> to an account with id <receiverAccountId>
- ✓ **Then** The customer gets a <status> status response and message: <message>

Examples:

	numberOfAccounts	ncards	money	receiverAccountId	transferAmount	status	message
✓	1	0	800	2	1500	400	"Transfer cannot be completed. Not enough money or blocked receiver."

✓ classpath:features/loginTest.feature

Feature: LoginTest with GenericSteps and AuthenticationTestSteps

Scenario Outline: The customer logins with a user to test GenericSteps

- ✓ **When** The customer logins with email "<email>" and password "<password>"
- ✓ **Then** The customer gets a <status> status response and message: "<message>"

Examples:

	email	password	status	message
✓	john.doe@example.com	password123	200	Correct authentication
✓	john.doe@example.com	password13	400	Invalid credentials

✓ classpath:features/rolesCustomer.feature

Feature: Test endpoints filtered by role

@NoCleanup

Scenario Outline: Obtain Customer list

- ✓ **Given** the system is ready and i log with email "<email>" and password "<pwd>"
- ✓ **When** i request all Customers list
- ✓ **Then** i should receive the code <code>
- ✓ **And** if the response is successful, i should receive the customers list

Examples:

	email	pwd	code
✓	john.doe@example.com	password123	403
✓	admin@admin.com	1234	200

@createFakeAccountFirst @NoCleanup

Scenario Outline: Delete Customer's account

- ✓ **Given** the system is ready and i log with email "<email>" and password "<pwd>"
- ✓ **When** i try to delete an another customer's account
- ✓ **Then** i should receive the code <code> and a status message

Examples:

	email	pwd	code
✓	john.doe@example.com	password123	403
✓	admin@admin.com	1234	200

✓ classpath:features/withdraw.feature

Feature: Withdraws tests

Background:

- ✓ **Given** The customer registers with 1 accounts, 1 cards and an initial amount of 2000.0
- ✓ **And** The customer logs in with their register credentials

- ✓ **And** I set the test cardId to my first registered card

Scenario: Create a withdrawal within the daily and monthly limit

- ✓ **When** I update the daily limit to 1000.0 for the current card
- ✓ **And** I update the monthly limit to 5000.0 for the current card
- ✓ **And** I create a withdraw of 50.0 EUR with the current card
- ✓ **Then** The customer gets a 200 status response

Scenario: Surpass the daily limit

- ✓ **When** I update the daily limit to 300.0 for the current card
- ✓ **And** I create a withdraw of 250.0 EUR with the current card
- ✓ **Then** The customer gets a 200 status response
- ✓ **When** I create a withdraw of 100.0 EUR with the current card
- ✓ **Then** The customer gets a 400 status response and message: "Saldo diario insuficiente, saldo restante diario: 50.0"

Scenario: Surpass the monthly limit

- ✓ **When** I update the daily limit to 10000.0 for the current card
- ✓ **And** I update the monthly limit to 1500.0 for the current card
- ✓ **And** I repeatedly create withdraws summing to 1600.0 EUR with the current card
- ✓ **Then** The customer gets a 400 status response and message: "Saldo mensual insuficiente, saldo restante mensual: 100.0"

Scenario: Update limits and try again

- ✓ **When** I update the daily limit to 500.0 for the current card
- ✓ **And** I update the monthly limit to 2000.0 for the current card
- ✓ **And** I create a withdraw of 400.0 EUR with the current card
- ✓ **Then** The customer gets a 200 status response