

Name: Connor Jacobs  
Email: [conjacobs@chapman.edu](mailto:conjacobs@chapman.edu)  
Due Date: Dec. 6th, 2023

## Solution Contracts

1. testLedgerName()
  - a. Ensure the ledger's name retrieval is correct
  - b. Check Ledger instance getName() and compare it with the name to ensure expected value
2. testGetNameAndSetName()
  - a. Confirm setting and getting the same of the ledger
  - b. Use getName() in order to check its original name. Then setName() and verify it using getName() again
3. testGetSeedAndSetSeed()
  - a. Confirm the setting and getting of the ledgers seed
  - b. Use getSeed() in order to check its original seed. Then setSeed() and verify it using getSeed() again
4. testSetNote()
  - a. Confirm setting the note of a transaction
  - b. Use setNote() and verify the note of the transaction using getNote()
5. testEqualsDifferentAccount()
  - a. Confirm that when two different accounts are compared they are considered not equal
  - b. Create two unique accounts. Then compare them using the overridden .equals(). It should return False.

End Point	API Method	Request Parameters	Response Parameters	Response Code	JIRA
/accounts	createAccount(Account A)	Account object(request body)	Account object Error 500	200 (success) 500 (internal server error)	<a href="https://conjacobs.atlassian.net/browse/KAN-11">https://conjacobs.atlassian.net/browse/KAN-11</a>
/accounts/{address}	getAccount(String address)	Path Variable: Address	Account Object Error 204 Error 500	200 (success) 204 (no content) 500 (internal server error)	<a href="https://conjacobs.atlassian.net/browse/KAN-12">https://conjacobs.atlassian.net/browse/KAN-12</a>
/transactions/{transactionID}	getTransaction(String ID)	Path Variable: transaction ID	Transaction object	200 (success)	<a href="https://conjacobs.atlassian.net/browse/KAN-13">https://conjacobs.atlassian.net/browse/KAN-13</a>
/transactions	processTransaction(transaction t)	Transaction object (request body)	Success String LedgerException	200 (success) 500 (internal server error)	<a href="https://conjacobs.atlassian.net/browse/KAN-14">https://conjacobs.atlassian.net/browse/KAN-14</a>

#	Test Description	Assumptions and preconditions	Test data	Steps to be executed	Expected result	jira
1	Retrieving the ledger name	The ledger is initialized correctly	String ledgerName "test ledger 2023"	Access the ledgers name using Ledger.getName()	The ledger name matches expected value	<a href="https://conjacobs.atlassian.net/browse/KAN-15">https://conjacobs.atlassian.net/browse/KAN-15</a>
2	Process Transaction with invalid amount	Transaction amount less than or equal 0	Ledger Exception message "Invalid transaction amount"	Retrieve transaction and make the value invalid  Process the transaction with the invalid num	Expects a ledger exception	<a href="https://conjacobs.atlassian.net/browse/KAN-16">https://conjacobs.atlassian.net/browse/KAN-16</a>
3	Process Transaction with max integer value	Transaction amount is Integer.Max_Value	maxIntValueTransaction	Retrieve transaction. Then process a transaction with the max integer value	Expect a ledger exception	<a href="https://conjacobs.atlassian.net/browse/KAN-17">https://conjacobs.atlassian.net/browse/KAN-17</a>
4	Get number of Block	Ledger with blocks created	Ledger block map	Retrieve ledger and then run getNumBlocks(). Then check to see if it is expected	Number of blocks matches expected value	<a href="https://conjacobs.atlassian.net/browse/KAN-18">https://conjacobs.atlassian.net/browse/KAN-18</a>
5	Set transaction ID	There is a transaction to set the ID of	String newTransactionID	Retrieve a transaction, change the id and then check if it matches expected	New transaction ID matches expected transaction ID	<a href="https://conjacobs.atlassian.net/browse/KAN-19">https://conjacobs.atlassian.net/browse/KAN-19</a>