Test cases description

KVStoreMetaDataTest (new for MS3)

Test Case	Description
KVStoreMetaDataTest.testMarshallKVStoreMetaData	• testing the marshall logic for KVStoreMetaData
KVStoreMetaDataTest.testUnMarshallKVStoreMetaData	• testing unmarshall logic for KVStoreMetaData for correct input string
KVStoreMetaDataTest.testUnMarshallKVStoreMetaDataThrowsException	• testing unmarshall logic throws proper exception for incorrect input string

ServerDataTest (new for MS3)

Test Case	Description
ServerDataTest.testMarshallServerData	• testing the marshall logic for ServerData
ServerDataTest.testUnMarshallServerData	• testing unmarshall logic for ServerData for correct input string
ServerDataTest.testUnMarshallServerDataThrowsException	• testing unmarshall logic throws proper exception for incorrect input string

KVMarshallerTest

Test Case	Description
KVMarshallerTest.testMarshall	• testing the marshall logic for general values
KVMarshallerTest.testMarshallSpecialCharacters	• testing marshall logic when the marshall values has special characters that are part of the message protocol

KVUnmarshallerTest

KVUnmarshallerTest.testUnmarshall	• testing the unmarshall logic for general values
KVUnmarshallerTest.testUnmarshallScpecialCharacters1	• testing unmarshall logic when the unmarshall values has special characters that are part of the message protocol
KVUnmarshallerTest.testUnmarshallScpecialCharacters2	• testing unmarshall logic when the unmarshall values has special characters that are part of the message protocol
KVUnmarshallerTest.testUnmarshallScpecialCharacters3	• testing unmarshall logic when the unmarshall values has special characters that are part of the message protocol
KVUnmarshallerTest.testUnmarshallScpecialCharacters4	• testing unmarshall logic when the unmarshall values has special characters that are part of the message protocol
KVUnmarshallerTest.testUnmarshallScpecialCharacters5	• testing unmarshall logic when the unmarshall values has special characters that are part of the message protocol
KVUnmarshallerTest.testUnmarshallScpecialCharacters6	• testing unmarshall logic when the unmarshall values has special characters that are part of the message protocol
KVUnmarshallerTest.testInvalidMessage	• testing unmarshall logic when message type is wrong
KVUnmarshallerTest.testEmptyMessage	• testing unmarshall logic when the key value is empty
KVUnmarshallerTest.testMessageWithSpaces	• testing unmarshall logic when the unmarshall values has spaces

FifoCacheTest

FifoCacheTest.addToCacheTest	• test the logic of adding a key value pair to the cache
FifoCacheTest.getFromCacheForExistingKeyTest	• test the logic when getting the value for an existing key
FifoCacheTest.getFromCacheForNonExistingKeyTest	• test the logic when trying to get the value for non existing key
FifoCacheTest.addToCacheReplaceTest	• test the cache replace logic when the cache is full

LFUCacheTest

LFUCacheTest.addToCacheTest	• test the logic of adding a key value pair to the cache
LFUCacheTest.getFromCacheForExistingKeyTest	• test the logic when getting the value for an existing key
LFUCacheTest.getFromCacheForNonExistingKeyTest	• test the logic when trying to get the value for non existing key
LFUCacheTest.addToCacheReplaceTest	• test the cache replace logic when the cache is full

LRUCacheTest

LRUCacheTest.addToCacheTest	• test the logic of adding a key value pair to the cache
LRUCacheTest.getFromCacheForExistingKeyTest	• test the logic when getting the value for an existing key
LRUCacheTest.getFromCacheForNonExistingKeyTest	• test the logic when trying to get the value for non existing key

LRUCacheTest.addToCacheReplaceLogicTest	• test the cache replace logic when the cache is full

SimpleKeyValueStoreTest

SimpleKeyValueStoreTest.shouldGetValue	• test get value logic from the database for a given key
SimpleKeyValueStoreTest.shouldGetMultipleValues	• test multiple consecutive retrievals for keys works correctly
SimpleKeyValueStoreTest.shouldReturnCorrectValuesForHasKey	• test whether the hasKey() method works properly
SimpleKeyValueStoreTest.shouldThrowKeyNotFound	• test the proper exception is thrown when trying to retrieve non existent key
SimpleKeyValueStoreTest.shouldWriteValues	• test the logic of writing to the database works properly for a given key value pair
SimpleKeyValueStoreTest.readWrites	• test that the written key value pairs can be read after
SimpleKeyValueStoreTest.readMultipleWrites	• test that the written multiple key value pairs can be read after

RandomAccessKeyValueStoreTest

RandomAccessKeyValueStoreTest.shouldGetValue	• test get value logic from the database for a given key
RandomAccessKeyValueStoreTest.shouldGetMultipleValues	• test multiple consecutive retrievals for keys works correctly
RandomAccessKeyValueStoreTest.shouldReturnCorrectValuesForHasKey	• test whether the hasKey() method works properly
RandomAccessKeyValueStoreTest.shouldThrowKeyNotFound	• test the proper exception is thrown when trying to retrieve non existent key

RandomAccessKeyValueStoreTest.shouldWriteFile	• test the logic of writing to the database works properly for a given key value pair
RandomAccessKeyValueStoreTest.shouldDeleteValue	• test the logic of deleting a key value pair works properly
RandomAccessKeyValueStoreTest.shouldDeleteValueAndReadOthers	• test that when there are multiple key value pairs in the database and when delete one, others can be read without error
RandomAccessKeyValueStoreTest.shouldUpdateValue	• test that the value is updated properly for an existing key

Connection

Connections.testConnectionSuccess	• test that the connect is success when the host and port are valid
Connections.testUnknownHost	• test that proper exception is thrown when trying to connect to a invalid host
Connections.testIllegalPort	• test that proper exception is thrown when the port is invalid

Interactions

Interactions.testPut	• test that the put command is working correctly for valid key, value pair
Interactions.testPutDisconnected	• test that put is not success when the client is disconnected and a proper exception is thrown
Interactions.testUpdate	• test that the when put command is issued with an existing key, the existing value in the database is updated
Interactions.testDelete	• test that delete key scenario is working properly
Interactions.testGet	• test that the get command is working correctly for valid key

Interactions.testGetUnsetValue	• test that the get command returns proper error when the key is not present in the database