CMTAT Test Framework

Table of Contents

Guideline

It is important that the tests can easily be improved and understood by others.

For each test file, the list of tests must be present.

How write a test ?

The test must follow the pattern AAA for the documentation and the structure.

First, read this excellent document by Microsoft.

Here a little resume :

Term	Definition
Arrange	Arrange your objects, create and set them up as necessary.
Arrange - Assert	Assertion to check your arrange
Act	The tested function
Assert	All check to verify the result obtained by the call of the function(s) in the Act part.

New test file

- Create a new tab with a new Id [A,B, C…..]

- Create a new tab in the section checklist

New test

For each new test : add an entry after the previous ones in the corresponding table

Example : you create a new test called *testCanTransferIsTrue* in the file RuleWhitelist.t.sol. You add then an entry in the corresponding table. After that, add the test in the checklist too.

Below is an example of an entry in the table

id	Test function	Truffle/ Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
[...previous test]									
25	testCanTransferIsTrue	[both, Foundry, Truffle]	The tested function	What is the result supposed to be returned by the function ???	[yes, no, -] “no” means “events are not checked” “-” means “there are no events to check”	Test with Truffle [As expected] or[Not as exepected + the result]	Test with Foundry [As expected] or[Not as exepected + the result]	[Ok, Not Ok]	Possible improvement for the test

Checklist

The checklist allows you to quickly check that all the functions are tested as well as to find the corresponding test.

The abbreviation OZ means that the tested function comes from the Library OpenZeppelin.

PauseModule (A)

File : **PauseModule.sol**

Functions	Test id
pause	A1, A2, A3, A7, A8
unpause	A4, A5, A6

MintModule (B)

File : **MintModule.sol**

Functions	Test id
mint	B1, B2, B3

BurnModule (C)

File : **BurnModule.sol**

Functions	Test id
burnFrom	C1,C2,C3,C4
forceBurn	C1, C2, C3b, C4

ValidationModule (D)

File : **ValidationModule.sol**

Functions	Test id
setRuleEngine	D1, D2
detectTransferRestriction	D3, D5
messageForTransferRestriction	D4, D6
Transfer (OZ)	D7, D8
mint	-

EnforcementModule (E)

File: EnforcementModule.sol

Functions	Test id
freeze	E1, E2, E5
unfreeze	E3, E4, E6

AuthorizationModule (F)

File: AuthorizationModule.sol, AccessControlUpgradeable.sol (OpenZeppelin)

Functions	Test id
GrantRole (OZ)	F1, F3
RevokeRole (OZ)	F2, F4

SnapshotModule(G)

Functions	Test id
scheduleSnapshot	1
scheduleSnapshotNotOptimized	2
rescheduleSnapshot	3, 4
unscheduleLastSnapshot	5, 6
unscheduleSnapshotNotOptimized	7, 8

BaseModule (H)

File: BaseModule.sol

Functions	Test id
tokenId	1
terms	2
setTokenId	3, 4
setTerms	5, 6
setInformation	7, 8

setFlag	9, 10
kill	12, 13

ERC20BaseModule (I)

- CMTAT

Functions	Test id
decimals	
transferFrom	
Approve	

- OpenZeppelin

Functions	Test id
name	1
symbol	2
approve	3, 4
increaseAllowance	5, 6
decreaseAllowance	

Proxy (Z)

Functions	Test id
Kill	Z1/1, Z2/1, Z2/2, Z2/3
UpgradeProxy (Truffle Plugin function)	Z3/1

Test list

Test Z – Proxy

Kill Implementation

We use a different version of the CMTAT where we have removed the check of access control on the kill function

The goal is to verify if the modifier *onlyDelegateCall* works as intended

Test Z1

Target File : CMTAT.sol

Test files: KillImplementation.test.js (Truffle)

id	Test function	Truffle/ Foundry	Target function	Expected result	Event check	Actual result	Conclusion	Improvement
1	testCannotKillTheImplementationContract	Truffle	kill	The contract is not killed	Yes	As expected	Ok	

Test Z2

Target File : CMTAT.sol

Test files: Proxy.test.js (Truffle)

id	Test function	Truffle/	Target function	Expected result	Event	Actual result	Concl	Improvement

		Foundry			check		usion	
1	testCannotBeTakenControlByAttacker1	Truffle	kill	-The attacker can not take control of the implementation contract. -It can not execute the function kill, an error is generated.	-	As expected	Ok	
2	testCannotBeTakenControlByAttacker2	Truffle	kill	Same result than testCannotBeTakenControlByAttacker1	-	As expected	Ok	
3	testCannotKillTheImplementationContractByAdmin	Truffle	kill	The admin can not execute the function kill, an error is generated.	-	As expected	Ok	

Test Z3

Target File : CMTAT.sol

Test files: UpgradeProxy.test.js (Truffle)

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Actual result	Conclusion	Improvement
1	testKeepStorageForTokens	Truffle	upgradeProxy	The proxy is upgraded with the new implementation and keeps its storage for the tokens balance.	-	As expected	Ok	

Test A - PauseModule

Target File: PauseModule.sol

Test files: PauseModuleCommon.js (Truffle), PauseModule.t.sol (Foundry)

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Actual result	conclusion	Improvement
1	testCanBePausedByAdmin	Both	pause	The contract is in pause	Yes	As expected	Ok	
2	testCanBePausedBy	Both	pause	The contract is in pause	Yes	As expected	Ok	

	ANewPauser							
3	testCannotBePaused ByNonPauser	both	pause	Revert because the sender has not the right role.	-	As expected	Ok	
4	testCanBeUnpaused ByAdmin	both	unpause	A contract in pause can get out from this state with a call to the unpause function by the admin	Yes	As expected	Ok	
5	TestCanBeUnpaused ByANewPauser	both	unpause	A contract in pause can get out from this state with a call to the unpause function by an address with the right role (PAUSER_ROLE)	Yes	As expected	OK	
6	testCannotBeUnpau sedByNonPauser	both	unpause	Revert because the sender has not the right role.	-	As expected	Ok	
7	testCannotTransferT okenWhenPaused_A	both	pause	The transfer is reverted because the contract is in pause	-	As expected	Ok	
8	testCannotTransferT okenWhenPaused_B	both	pause	The transfer is reverted because the contract is in pause	-	As expected	Ok	

Test B - MintModule

Target File : MintModule.sol

Test files: MintModuleCommon.js (Truffle), MintModule.t.sol (Foundry)

id	Test function	Truffle/ Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	testCanBeMintedByAdmin	Both	mint	The tokens are minted	Yes	As expected	As expected	Ok	
2	testCanBeMintedByANewMinter	Both	mint	The tokens are minted	Yes	As expected	As expected	Ok	
3	testCannotIssuingByNonMinter	Both	mint	Revert because the sender has not the right role.	-	As expected	As expected	OK	

Test C - BurnModule

Target File : **BurnModule.sol**

Target File : CMTAT.sol

Test files: BurnModuleCommon.js (Truffle), BurnModule.t.sol (Foundry)

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	testCanBeBurntByAdminWithAllowance	Both	ForceBurn (Truffle) BurnFrom (Foundry)	The tokens are burn	Yes		As expected	Ok	
2	testCanBeBurntByBurnerRole	Both	ForceBurn (Truffle) BurnFrom (Foundry)	The tokens are burn	Yes		As expected	Ok	
3a	testCannotBeBurntWithoutAllowance	Foundry	burnFrom	Revert because the sender has not sufficient allowance on the tokens	-		As expected	Ok	
3b	testCannotBeBurntIfBalanceExceeds	Truffle	forceBurn	Revert because the target address has not enough tokens	-		As expected	Ok	

4	testCannotBeBurntWithoutBurnerRole	Both	ForceBurn (Truffle) BurnFrom (Foundry)	Revert because the sender has not the right role	-		As expected	Ok	

Test D - ValidationModule

Target File : **ValidationModule.sol**

Test files: ValidationModuleCommon.js (Truffle), ValidationModule.t.sol (Foundry)

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	testCanBeSetByAdmin	both	setRuleEngine	The RuleEngine is set	Yes	As expected	As expected	Ok	
2	testCannotBeSetByNonAdmin	both	setRuleEngine	The transaction is reverted	-	As expected	As expected	Ok	
3	testCanDetectTransferRestrictionValidTransfer	both	detectTransferRestriction	The returned code corresponds to that of a valid transfer	-	As expected	As expected	Ok	
4	testCanReturnMessageValid	both	messageForTransferRestriction	The returned `message` corresponds to that of a valid	-	As expected	As expected	Ok	

	Transfer			transfer					
5	testCanDetectTransferRestrictionWithAmountTooHigh	both	detectTransferRestriction	The returned code corresponds to that of a invalid transfer in reason of excessive amount	-	As expected	As expected	Ok	
6	testCanReturnMessageWithAmountTooHigh	both	messageForTransferRestriction	The returned `message` corresponds to that of a invalid transfer in reason of excessive amount	-		As expected	Ok	
7	testCanTransferAllowedByRule	both	transfer	The transfer is performed	No		As expected	Ok	
8	testCannotTransferIfNotAllowedByRule	both	transfer	The transfer is not performed, the transaction is reverted.	No		As expected	Ok	

Set RuleEngine

Test E - EnforcementModule

Target File : EnforcementModule.sol

Test files: EnforcementModuleCommon.js (Truffle), EnforcementModule.t.sol (Foundry)

id	Test function	Truffle	Target function	Expected result	Event	Truffle	Foundry	concl	Improveme

		/ Foundry			check	Actual result	Actual result	usion	nt
1	testAdminCanFreezeAddress	both	freeze	The target address is frozen	Yes	As expected	As expected	Ok	
2	testEnforcerRoleCanFreezeAddress	both	freeze	The target address is frozen	Yes	As expected	As expected	Ok	
3	testAdminCanUnfreezeAddress	both	unfreeze	The target address is no longer frozen	Yes	As expected	As expected	Ok	
4	testEnforcerRoleCanUnfreezeAddress	both	unfreeze	The target address is no longer frozen, the transaction is reverted	Yes	As expected	As expected	Ok	
5	testCannotNonEnforcerFreezeAddress	both	freeze	The address is not frozen, the transaction is reverted	-	As expected	As expected	Ok	

6	testCannotNonEnforcerUnfreezeAddress	both	unfreeze	The address is still frozen, the transaction is reverted	-	As expected	As expected	Ok	

Test F – AuthorizationModule

Target File : AuthorizationModule.sol, AccessControlUpgradeable.sol (OpenZeppelin)

Test files: AuthorizationModuleCommon.js (Truffle), AuthorizationModule.t.sol (Foundry)

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	testAdminCanGrantRole	both	grantRole	The target address obtains the role	Yes	As expected	As expected	Ok	
2	testAdminCanRevokeRole	both	revokeRole	The target address loses the role	Yes	As expected	As expected	Ok	
3	testCannotNonAdminGrantRo	both	grantRole	The target address does not obtain the role	-	As expected	As expected	Ok	

	le								
4	testCannotNonAdminRevokeRole	both	revokeRole	The target address keeps its role, the transaction is reverted	-	As expected	As expected	Ok	

Test G – SnapshotModule

G1 – SnapshotModuleCommon – Global

scheduleSnapshot	G1/b, G1C, G4-1-4
scheduleSnapshotNotOptimized	G4 5-10
rescheduleSnapshot	G3
unscheduleLastSnapshot	G5 6-10

unscheduleSnapshotNotOptimized	G5 1-5
SnapshotTotalSupply	G1/a1
snapshotBalanceOf	G1/a1

G1/a - ZeroPlannedSnapshotTest

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	testCanGetBalanceAddress&TotalSupply	both	SnapshotTotalSupply snapshotBalanceOf	The number of tokens corresponds to the number of tokens minted	-	As expected	As expected	Ok	19

G1/b– OnePlannedSnapshotTest

id	Test function	Truffle /	Target function	Expected result	Event	Truffle	Foundry	conclusion	Improvement

		Foundry			check	Actual result	Actual result		
1	testCanMintTokens	both	scheduleSnapshot + mint / _beforeTokenTransfer	The number of tokens (total supply + balance of the tokens owner) corresponds to the number of tokens minted before & after the snapshot	-	As expected	As expected	Ok	
2	testCanBurnTokens	both	scheduleSnapshot + BurnFrom / forceBurn _beforeTokenTransfer	The number of tokens (total supply + balance of the tokens owner) corresponds to the number of tokens before & after the snapshot	-	As expected	As expected	Ok	
3	testCanTransferTokens	both	scheduleSnapshot + transfer / _beforeTokenTransfer	The number of tokens (total supply + balance of the tokens owner) corresponds to the number of tokens before & after the snapshot	-	As expected	As expected	Ok	

G1/c – MultiplePlannedSnapshotTest

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	testCanTransferTokensAfterFirstSnapshot	both	scheduleSnapshot + transfer / _beforeTokenTransfer	The number of tokens (total supply + balance of the tokens owner) corresponds to the number of tokens before & after the snapshot	-	As expected	As expected	Ok	
2	testCanTransferAfterSecondSnapshots	both	scheduleSnapshot + transfer / _beforeTokenTransfer	The number of tokens (total supply + balance of the tokens owner) corresponds to the number of tokens burned before & after the snapshot	-	As expected	As expected	Ok	
3	testCanTransferAfterThird	both	ScheduleSnapshot	The number of tokens (total supply + balance		As expected	As expected	Ok	

	Snapshot		+ transfer / _beforeTokenTransfer	of the tokens owner) corresponds to the number of tokens burned before & after the snapshot					
4	testCanTransferTokensMultipleTimes	both	ScheduleSnapshot + transfer / _beforeTokenTransfer	The number of tokens (total supply + balance of the tokens owner) corresponds to the number of tokens burned before & after the snapshot		As expected	As expected		

G2 – SnapshotModuleCommon – GetNextSnapshot

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	return empty array if all snapshots are in the past	both	transfer _beforeTokenTransfer	The number of tokens (total supply + balance of the tokens owner) corresponds to the number of tokens	-	As expected	As expected	Ok	

				before & after the snapshot					
2	return only future snapshots if some snapshots are in the past	both	transfer _beforeTokenTransfer	The number of tokens (total supply + balance of the tokens owner) corresponds to the number of tokens burned before & after the snapshot	-	As expected	As expected	Ok	

G3 – SnapshotModuleCommon – Rescheduling

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	can reschedule a snapshot with the snapshoter role and emits a	Truffle	rescheduleSnapshot	The snapshot is rescheduled	yes	As expected	As expected	Ok	

	SnapshotSchedule event								
2	can reschedule a snapshot between a range of snapshot	Truffle	rescheduleSnapshot	The snapshot is rescheduled	yes	As expected	As expected	Ok	
3	revert if reschedule a snapshot not in the range of snapshot	Truffle	rescheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	
4	revert if reschedule a snapshot not in the range of snapshot	Truffle	rescheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	
5	reverts when calling from non-owner	Truffle	rescheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	
6	reverts when trying to reschedule a		rescheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	

	snapshot in the past								
7	reverts when snapshot is not found	Truffle	rescheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	
8	reverts when snapshot has been processed	Truffle	rescheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	

G4 – SnapshotModuleCommon – Scheduling

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	can schedule a snapshot with the snapshoter role	Truffle	ScheduleSnapshot	The snapshot is scheduled	yes	As expected	As expected	Ok	

2	reverts when calling from non-owner	Truffle	ScheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	
3	reverts when trying to schedule a snapshot in the past	Truffle	ScheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	
4	reverts when trying to schedule a snapshot with the same time twice	Truffle	ScheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	
5	can schedule a snapshot in the first place with the snapshoter role	Truffle	scheduleSnapshotNotOptimized	The snapshot is scheduled	no	As expected	As expected	Ok	
6	can schedule a snaphot in	Truffle	scheduleSnapshotNotOptimized	The snapshot is scheduled	yes	As expected	As expected	Ok	

	a random place								
7	schedule a snapshot, which will be in the last position	Truffle	scheduleSnapshotNotOptimized	The snapshot is scheduled	yes	As expected	As expected	Ok	
8	reverts when calling from non-owner	Truffle	scheduleSnapshotNotOptimized	The transaction is reverted	-	As expected	As expected	Ok	
9	reverts when trying to schedule a snapshot in the past	Truffle	scheduleSnapshotNotOptimized	The transaction is reverted	-	As expected	As expected	Ok	
10	reverts when trying to schedule a snapshot with the same time twice	Truffle	scheduleSnapshotNotOptimized	The transaction is reverted	-	As expected	As expected	Ok	

G5 – SnapshotModuleCommon – unscheduling

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	can remove a snapshot as admin	Truffle	unscheduleSnapshotNotOptimized	The snapshot is unscheduled	no	As expected	As expected	Ok	
2	can remove a random snapshot with the snapshoter role	Truffle	unscheduleSnapshotNotOptimized	The transaction is reverted	-	As expected	As expected	Ok	
3	Revert if no snapshot	Truffle	unscheduleSnapshotNotOptimized	The transaction is reverted	-	As expected	As expected	Ok	
4	can unschedule a snaphot in a	Truffle	unscheduleSnapshotNotOptimized	The transaction is reverted	-	As expected	As expected	Ok	

	random place								
5	can schedule a snaphot after an unschedule	Truffle	unscheduleSnapshotNotOptimized	The snapshot is scheduled	no	As expected	As expected	Ok	
6	can unschedule a snapshot with the snapshoter role and emits a SnapshotUnschedule event	Truffle	unscheduleLastSnapshot	The snapshot is unscheduled	yes	As expected	As expected	Ok	
7	reverts when calling from non-owner	Truffle	unscheduleLastSnapshot	The transaction is reverted	-	As expected	As expected	Ok	
8	reverts if no snapshot is scheduled	Truffle	unscheduleLastSnapshot	The transaction is reverted	-	As expected	As expected	Ok	
9	reverts when snapshot is	Truffl	unscheduleLastSna	The transaction is					

	not found	e	pshot	reverted					
10	reverts when snapshot has been processed	Truffle	unscheduleLastSnapshot	The transaction is reverted					

Test H – BaseModule

Target File : BaseModule.sol

Test files: BaseModuleCommon.js (Truffle), BaseModule.t.sol (Foundry)

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	testHasTheDefinedTokenId	Truffle	tokenId	The contract has the defined tokenId	-	As expected	As expected	Ok	

2	testHasTheDefinedTerms	Truffle	terms	The contract has the defined terms	-	As expected	As expected	Ok	
3	testAdminCanChangeTokenId	Truffle	setTokenId	The tokenId is set	yes	As expected	As expected	Ok	
4	testCannotNonAdminChangeTokenId	Truffle	setTokenId	The transaction is reverted	-	As expected	As expected	Ok	
5	testAdminCanUpdateTerms	Truffle	setTerms	The terms are set	yes	As expected	As expected	Ok	
6	testCannotNonAdminUpdateTerms	Truffle	setTerms	The transaction is reverted	-	As expected	As expected	Ok	
7	testAdminCanUpdateInformation	Truffle	setInformation	The information is set	yes	As expected	As expected	Ok	
8	testCannotNonAdmi	Truffle	setInformation	The transaction is reverted	-	As expected	As expected	Ok	

	nUpdateInformation								
9	testAdminCanUpdateFlag	Truffle	setFlag	The flag is set	yes	As expected	As expected	Ok	
10	testAdminCanNotUpdateFlagWithTheSameValue	Truffle	setFlag	The transaction is reverted	-	As expected	As expected	Ok	
11	testCannotNonAdminUpdateFlag	Truffle	setFlag	The transaction is reverted	-	As expected	As expected	Ok	
12	testAdminCanKillContract	Truffle	kill	The contract is destroyed	-	As expected	As expected	Ok	
13	testCannotNonAdminKillContract	Truffle	kill	The transaction is reverted	-	As expected	As expected	Ok	

Test I – ERC20BaseModule

Target File : ERC20BaseModule.sol

Test files: ERC20BaseModuleCommon.js (Truffle)

ERC20BaseModule (I)

- CMTAT

Functions	Test id
decimals	I1/3
transferFrom	I3/3, I3/4
Approve	I2/5, I/6

- OpenZeppelin

Functions	Test id
name	I1/1
symbol	I1/2
approve	I2/1, I2/4

increaseAllowance	I2/2
decreaseAllowance	I2/3
transfer	I3/1, I3/2

I1 – Initialization

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	testHasTheDefinedName	Truffle	name (OZ)	The contract has the defined name	-	As expected	As expected	Ok	
2	testHasTheDefinedSymbol	Truffle	symbol (OZ)	The contract has the defined symbol	-	As expected	As expected	Ok	
3	testDecimalsEqual0	Truffle	decimals	The contract has the right decimal number (zero)	yes	As expected	As expected	Ok	

I2 – Allowance

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	testApproveAllowance	Truffle	Approve (OZ)	The spender has the correct allowance	yes	As expected	As expected	Ok	
2	testIncreaseAllowance	Truffle	IncreaseAllowance(OZ)	The spender has the correct allowance	yes	As expected	As expected	Ok	
3	testDecreaseAllowance	Truffle	DecreaseAllowance (OZ)	The spender has the correct allowance	yes	As expected	As expected	Ok	
4	testRedefinedAllowanceWithApprove	Truffle	approve(OZ)	The spender has the correct allowance	yes	As expected	As expected	Ok	
5	testDefinedAllowanc		approve(CMTAT)	The spender has the correct allowance	yes	As expected	As expected	Ok	

	eByTakingInAccountTheCurrentAllowance								
6	testCannotDefinedAllowanceByTakingInAccountTheWrongCurrentAllowance		approve(CMTAT)	The transaction is reverted	-	As expected	As expected	Ok	

I3 – Transfer

id	Test function	Truffle / Foundry	Target function	Expected result	Event check	Truffle Actual result	Foundry Actual result	conclusion	Improvement
1	testTransferFromOneAccountToAnother	Truffle	transfer(OZ)	The defined amount of tokens is transferred	yes	As expected	As expected	Ok	

2	testCannotTransferMoreTokensThanOwn	Truffle	transfer(OZ)	The transaction is reverted	-	As expected	As expected	Ok	
3	testTransferByAnotherAccountWithTheRightAllowance	Truffle	transferFrom	The defined amount of tokens is transferred	yes	As expected	As expected	Ok	
4	testCannotTransferByAnotherAccountWithInsufficientAllowance	Truffle	transferFrom	The transaction is reverted	.	As expected	As expected	Ok	
5	testCannotTransferByAnotherAccountWithInsufficientBalance	Truffle	transferFrom	The transaction is reverted	-	As expected	As expected	Ok	
