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	Hardhat/ Foundry	Target function	Expected result	Event check	Hardhat Actual result	Foundry Actual result	conclusion	Improvement
[...previous test]									
25	testCanTransferIsTrue	[both, Foundry, Hardhat]	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 Hardhat [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	A1/1, A1/2, A1/3, A1/7, A8
unpause	A1/4, A1/5, A1/6
deactivateContract	A2

ERC20MintModule (B)

File : **MintModule.sol**

Functions	Test id
mint	B1
MintBatch	B2

ERC20BurnModule (C)

File : **BurnModule.sol**

Functions	Test id
burnFrom	C1,C2,C3,C4
forceBurn	C1/1, C1/2, C1/3b, C4
forceBurnBatch	

ValidationModule (D)

File : **ValidationModule.sol**

Functions	Test id
setRuleEngine	D2/1, D2/1, D2/1
detectTransferRestriction	D1/1, D1/3
messageForTransferRestriction	D1/2, D1/4, D1/5 , D2/4
Transfer (OZ)	D1/6, D1/7
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
BeginDefaultAdmi nTransfer (OZ)r	F5, F6
transferAdminiship Directly	F7, F8

ERC20SnapshotModule(G)

scheduleSnapshot	G1/b, G1C, G4-1-4
scheduleSnapshotNotOptimized	G4 5-10
rescheduleSnapshot	G3
unscheduleLastSnapshot	G5 6-10
unscheduleSnapshotNotOptimized	G5 1-5, G5/11
SnapshotTotalSupply	G1/a1
snapshotBalanceOf	G1/a1

BaseModule (H)

File: BaseModule.sol

Functions	Test id
tokenId	1
terms	2
setTokenId	3, 4
setTerms	5, 6
setInformation	7, 8
setFlag	9, 10

ERC20BaseModule (I)

- CMTAT

Functions	Test id
decimals	I1/3
transferFrom	I3/3, I3/4
Approve	I2/5, I/6
transferBatch	I4

- OpenZeppelin

Functions	Test id
name	I1/1
symbol	I1/2
approve	I2/1, I2/4
transfer	I3/1, I3/2

Proxy (Z)

Functions	Test id
initialize	Z1/1
UpgradeProxy (Hardhat Plugin function)	Z2/1

Test list

Test Z – Proxy

Test Z1

Target File : CMTAT.sol

Test files: Proxy.test.js (Hardhat)

id	Test function	Hardhat/ Foundry	Target function	Expected result	Event check	Actual result	Conclusion	Improvement
1	testCannotBeTakenControlByAttacker	Hardhat	initialize	-The attacker can not take control of the implementation contract. -It can not execute a protected function, an error is generated.	-	As expected	Ok	

Test Z2

Target File : CMTAT.sol

Test files: UpgradeProxy.test.js (Hardhat)

id	Test function	Hardhat/ Foundry	Target function	Expected result	Event check	Actual result	Conclusion	Improvement
1	testKeepStorageForTokens	Hardhat	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 (Hardhat), PauseModule.t.sol (Foundry)

A1 - General

id	Test function	Hardhat/ 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	testCanBePausedByANewPauser	Both	pause	The contract is in pause	Yes	As expected	Ok	
3	testCannotBePausedByNonPauser	both	pause	Revert because the sender has not the right role.	-	As expected	Ok	
4	testCanBeUnpausedByAdmin	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	TestCanBeUnpausedByANewPauser	both	unpause	A contract in pause can get out from this state with a call to the unpause function by an	Yes	As expected	OK	

				address with the right role (PAUSER_ROLE)				
6	testCannotBeUnpausedByNonPauser	both	unpause	Revert because the sender has not the right role.	-	As expected	Ok	
7	testCannotTransferTokenWhenPausedWithTransfer	both	pause	The transfer is reverted because the contract is in pause	-	As expected	Ok	
8	testCannotTransferTokenWhenPausedWithTransferFrom	both	pause	The transfer is reverted because the contract is in pause	-	As expected	Ok	

A2 – DeactivateContract

id	Test function	Hardhat/ Foundry	Target function	Expected result	Event check	Actual result	conclusion	Improvement
1	testCanDeactivatedByAdmin	Hardhat	deactivateContract	The contract is deactivated, the function unpause reverts if an address try to unpause the contract	Yes	As expected	Ok	
2	testCannotBeDeactivatedByNonAdmin	Hardhat	deactivateContract	Revert because the sender has not the right role.	Yes	As expected	Ok	

Test B - ERC20MintModule

Target File : ERC20MintModule.sol

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

B1 - Mint

id	Test function	Hardhat/ Foundry	Target function	Expected result	Event check	Hardhat 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	

B2 - MintBatch

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat Actual result	Foundry Actual result	conclusion	Improvement
1	testCanBeMintedBatchByAdmin	Hardhat	mintBatch	The tokens are minted	Yes	As expected	-	Ok	
2	testCanBeMintedBatchByBurnerRole	Hardhat	mintBatch	The tokens are minted	Yes	As expected	-	Ok	
3b	testCannotBeMintedBatchIfOneBalanceExceeds	Hardhat	mintBatch	Revert because one of the target address has not enough tokens	-	As expected	-	Ok	
4	testCannotBeMintedBatchWithoutBurnerRole	Hardhat	mintBatch	Revert because the sender has not the right role	-	As expected	-	Ok	
5	testCannotMintedBatchIfLengthMismatchMissingAddresses	Hardhat	mintBatch	Revert because the number of account is not equal to the number of tokens by holders	-	As expected	-	Ok	

6	testCannotMintedBatchIfLengthMismatchTooManyAddresses	Hardhat	mintBatch	Revert because the number of account is not equal to the number of tokens by holders	-	As expected	-	Ok	
7	testCannotMintedBatchIfAccountsIsEmpty	Hardhat	mintBatch	Revert because accounts is empty	-	As expected	-	Ok	

Test C - ERC20BurnModule

Target File : **BurnModule.sol**

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

C1 - forceBurn

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat Actual result	Foundry Actual result	conclusion	Improvement
1	testCanBeBurntByAdminWithAllowance	Both	ForceBurn (Hardhat) BurnFrom (Foundry)	The tokens are burn	Yes		As expected	Ok	
2	testCanBeBurntByBurnerRole	Both	ForceBurn (Hardhat) 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	Hardhat	forceBurn	Revert because the target address has not enough tokens	-		As expected	Ok	

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

C2 - forceBurnBatch

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat Actual result	Foundry Actual result	conclusion	Improvement
1	testCanBeBurntBatchByAdmin	Hardhat	forceBurnBatch	The tokens are burn	Yes	As expected	-	Ok	
2	testCanBeBurntBatchByBurnerRole	Hardhat	forceBurnBatch)	The tokens are burn	Yes	As expected	-	Ok	
3b	testCannotBeBurntBatchIfOneBalanceExceeds	Hardhat	forceBurnBatch	Revert because one of the target address has not enough tokens	-	As expected	-	Ok	
4	testCannotBeBurntBatchWithoutBurnerRole	Hardhat	forceBurnBatch	Revert because the sender has not the right role	-	As expected	-	Ok	

5	testCannotBurnBatchIfLengthMismatchMissingAddresses	Hardhat	forceBurnBatch	Revert because the number of account is not equal to the number of tokens by holders	-	As expected	-	Ok	
6	testCannotBurnBatchIfLengthMismatchTooManyAddresses	Hardhat	forceBurnBatch	Revert because the number of account is not equal to the number of tokens by holders	-	As expected	-	Ok	
7	testCannotBurnBatchIfAccountsIsEmpty	Hardhat	forceBurnBatch	Revert because accounts is empty	-	As expected	-	Ok	

Test D – ValidationModule

D1 - ValidationModuleCommon

Target File : **ValidationModule.sol**

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

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat Actual result	Foundry Actual result	conclusion	Improvement
1	testCanDetectTransferRestrictionValidTransfer	both	detectTransferRestriction	The returned code corresponds to that of a valid transfer	-	As expected	As expected	Ok	
2	testCanReturnMessageValidTransfer	both	messageForTransferRestriction	The returned message corresponds to that of a valid transfer	-	As expected	As expected	Ok	
3	testCanDetectTransferRestrictionWithAmountTooHigh	both	detectTransferRestriction	The returned code corresponds to that of a invalid transfer in reason of excessive amount	-	As expected	As expected	Ok	

4	testCanReturnMessageWithAmountTooHigh	both	messageForTransferRestriction	The returned message corresponds to that of a invalid transfer in reason of excessive amount	-	As expected	As expected	Ok	
5	testCanReturnMessageWithUnknownRestrictionCode	Hardhat	messageForTransferRestriction	The returned message corresponds to the message to returned in case of an unknown restriction code	-	As expected	As expected	Ok	
6	testCanTransferAllowedByRule	both	transfer	The transfer is performed	No	As expected	As expected	Ok	
7	testCannotTransferIfNotAllowedByRule	both	transfer	The transfer is not performed, the transaction is reverted.	No	As expected	As expected	Ok	

D2- Set RuleEngine

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat 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	testCanNotBeSetByAdminWithTheSameValue	Hardhat	setRuleEngine	The transaction is reverted	-	As expected	As expected	Ok	
4	testCanReturnMessageWithNoRuleEngine&UnknownRestrictionCode	Hardhat	setRuleEngine	Return the right message	-	As expected	As expected	Ok	

Test E - EnforcementModule

Target File : EnforcementModule.sol

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

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat Actual result	Foundry Actual result	conclusion	Improvement
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 (Hardhat), AuthorizationModule.t.sol (Foundry)

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat Actual result	Foundry Actual result	conclusion	Improvement
1	testAdminCanGrantRole	both	GrantRole (OZ)	The target address obtains the role	Yes	As expected	As expected	Ok	
2	testAdminCanRevokeRole	both	RevokeRole (OZ)	The target address loses the role	Yes	As expected	As expected	Ok	
3	testCannotNonAdminGrantRole	both	GrantRole (OZ)	The target address does not obtain the role	-	As expected	As expected	Ok	
4	testCannotNonAdminRevokeRole	both	RevokeRole (OZ)	The target address keeps its role, the transaction is reverted	-	As expected	As expected	Ok	
5	testCanAdminTransferAdminship	Hardhat	BeginDefaultAdminTransfer (OZ) acceptDefault	We can transfer the admin right to another address	Yes	As expected	As expected	Ok	

			AdminTransfer (OZ)						
6	testCannotNonAdminTransferAdminship	Hardhat	BeginDefaultAdminTransfer (OZ)r	The transaction is reverted, te target address does not obtain the role	-	As expected	As expected	Ok	
7	testCanAdminTransferAdminshipDirectly	Hardhat	transferAdminshipDirectly	We can transfer the admin right to another address, without the address having to approve the change	Yes	As expected	As expected	Ok	
8	testCannotNonAdminTransferAdminshipDirectly	Hardhat	transferAdminshipDirectly	The transaction is reverted, te target address does not obtain the role	-	As expected	As expected	Ok	

Test G – ERC20SnapshotModule

G1 – ERC20SnapshotModuleCommon – Global

G1/a - ZeroPlannedSnapshotTest

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat 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	31

G1/b– OnePlannedSnapshotTest

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat Actual result	Foundry Actual result	conclusion	Improvement
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	The number of tokens (total supply + balance	-	As expected	As expected	Ok	

			+ transfer / _beforeTokenTransfer	of the tokens owner) corresponds to the number of tokens before & after the snapshot					

G1/c – MultiplePlannedSnapshotTest

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat 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	testCanTransferAfterThirdSnapshot	both	ScheduleSnapshot + transfer /	The number of tokens (total supply + balance of the tokens owner) corresponds to the		As expected	As expected	Ok	

			_beforeTokenTransfer	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 – ERC20SnapshotModuleCommon – GetNextSnapshot

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat 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 before & after the snapshot	-	As expected	As expected	Ok	
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 – ERC20SnapshotModuleCommon – Rescheduling

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat Actual result	Foundry Actual result	conclusion	Improvement
1	can reschedule a snapshot with the snapshoter role and emits a SnapshotScheduled event	Hardhat	rescheduleSnapshot	The snapshot is rescheduled	yes	As expected	As expected	Ok	
2	can reschedule a snapshot between a range of snapshot	Hardhat	rescheduleSnapshot	The snapshot is rescheduled	yes	As expected	As expected	Ok	
3	revert if reschedule a snapshot not	Hardhat	rescheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	

	in the range of snapshot								
4	revert if reschedule a snapshot not in the range of snapshot	Hard hat	rescheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	
5	reverts when calling from non-owner	Hard hat	rescheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	
6	reverts when trying to reschedule a snapshot in the past	Hard hat	rescheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	
7	reverts when snapshot is not found	Hard hat	rescheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	
8	reverts when snapshot has been processed	Hard hat	rescheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	

G4 – ERC20SnapshotModuleCommon – Scheduling

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat Actual result	Foundry Actual result	conclusion	Improvement
1	can schedule a snapshot with the snapshoter role	Hard hat	ScheduleSnapshot	The snapshot is scheduled	yes	As expected	As expected	Ok	
2	reverts when calling from non-owner	Hard hat	ScheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	
3	reverts when trying to schedule a snapshot in the past	Hard hat	ScheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	
4	reverts when trying to schedule a snapshot with the	Hard hat	ScheduleSnapshot	The transaction is reverted	-	As expected	As expected	Ok	

	same time twice								
5	can schedule a snapshot in the first place with the snapshoter role	Hard hat	scheduleSnapshotNotOptimized	The snapshot is scheduled	no	As expected	As expected	Ok	
6	can schedule a snaphot in a random place	Hard hat	scheduleSnapshotNotOptimized	The snapshot is scheduled	yes	As expected	As expected	Ok	
7	schedule a snapshot, which will be in the last position	Hard hat	scheduleSnapshotNotOptimized	The snapshot is scheduled	yes	As expected	As expected	Ok	
8	reverts when calling from non-owner	Hard hat	scheduleSnapshotNotOptimized	The transaction is reverted	-	As expected	As expected	Ok	
9	reverts when trying to schedule a	Hard hat	scheduleSnapshotNotOptimized	The transaction is reverted	-	As expected	As expected	Ok	

	snapshot in the past								
10	reverts when trying to schedule a snapshot with the same time twice	Hard hat	scheduleSnapshotNotOptimized	The transaction is reverted	-	As expected	As expected	Ok	

G5 – ERC20SnapshotModuleCommon – unscheduling

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

	after an unschedule	hat	otOptimized	scheduled					
11	reverts when calling from non-owner	Hard hat	unscheduleSnapshotNotOptimized	The transaction is reverted	-	As expected	As expected	Ok	
6	can unschedule a snapshot with the snapshoter role and emits a SnapshotUnschedule event	Hard hat	unscheduleLastSnapshot	The snapshot is unscheduled	yes	As expected	As expected	Ok	
7	reverts when calling from non-owner	Hard hat	unscheduleLastSnapshot	The transaction is reverted	-	As expected	As expected	Ok	
8	reverts if no snapshot is scheduled	Hard hat	unscheduleLastSnapshot	The transaction is reverted	-	As expected	As expected	Ok	

9	reverts when snapshot is not found	Hard hat	unscheduleLastSnapshot	The transaction is reverted					
10	reverts when snapshot has been processed	Hard hat	unscheduleLastSnapshot	The transaction is reverted					

Test H – BaseModule

Target File : BaseModule.sol

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

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat Actual result	Foundry Actual result	conclusion	Improvement
1	testHasTheDefinedTokenId	Hardhat	tokenId	The contract has the defined tokenId	-	As expected	As expected	Ok	
2	testHasTheDefinedTerms	Hardhat	terms	The contract has the defined terms	-	As expected	As expected	Ok	
3	testAdminCanChangeTokenId	Hardhat	setTokenId	The tokenId is set	yes	As expected	As expected	Ok	
4	testCannotNonAdminChangeTokenId	Hardhat	setTokenId	The transaction is reverted	-	As expected	As expected	Ok	

5	testAdminCanUpdateTerms	Hardhat	setTerms	The terms are set	yes	As expected	As expected	Ok	
6	testCannotNonAdminUpdateTerms	Hardhat	setTerms	The transaction is reverted	-	As expected	As expected	Ok	
7	testAdminCanUpdateInformation	Hardhat	setInformation	The information is set	yes	As expected	As expected	Ok	
8	testCannotNonAdminUpdateInformation	Hardhat	setInformation	The transaction is reverted	-	As expected	As expected	Ok	
9	testAdminCanUpdateFlag	Hardhat	setFlag	The flag is set	yes	As expected	As expected	Ok	
10	testAdminCanNotUpdateFlagWithTheSameValue	Hardhat	setFlag	The transaction is reverted	-	As expected	As expected	Ok	

11	testCannotNonAdminUpdateFlag	Hardhat	setFlag	The transaction is reverted	-	As expected	As expected	Ok	

Test I – ERC20BaseModule

Target File : ERC20BaseModule.sol

Test files: ERC20BaseModuleCommon.js (Hardhat)

I1 – Initialization

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

I2 – Allowance

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat Actual result	Foundry Actual result	conclusion	Improvement
1	testApproveAllowance	Hardhat	Approve (OZ)	The spender has the correct allowance	yes	As expected	As expected	Ok	
2	testRedefinedAllowanceWithApprove	Hardhat	approve(OZ)	The spender has the correct allowance	yes	As expected	As expected	Ok	
3	testDefinedAllowanceByTakingInAccountTheCurrentAllowance		approve(CMTAT)	The spender has the correct allowance	yes	As expected	As expected	Ok	
4	testCannotDefinedAllowanceB		approve(CMTAT)	The transaction is reverted	-	As expected	As expected	Ok	

	yTakingIn AccountT heWrong CurrentAll owance								

I3 – Transfer

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat Actual result	Foundry Actual result	conclusion	Improvement
1	testTransferFromOneAccountToAnother	Hardhat	transfer(OZ)	The defined amount of tokens is transferred	yes	As expected	As expected	Ok	
2	testCannotTransferMoreTokensThanOwn	Hardhat	transfer(OZ)	The transaction is reverted	-	As expected	As expected	Ok	
3	testTransferByAnotherAccountWithTheRightAllowance	Hardhat	transferFrom	The defined amount of tokens is transferred	yes	As expected	As expected	Ok	
4	testCannotTransferByAnotherAccountW	Hardhat	transferFrom	The transaction is reverted	.	As expected	As expected	Ok	

	ithInsufficientAllowance								
5	testCannotTransferByAnotherAccountWithInsufficientBalance	Hardhat	transferFrom	The transaction is reverted	-	As expected	As expected	Ok	

I4 – TransferBatch

id	Test function	Hardhat / Foundry	Target function	Expected result	Event check	Hardhat Actual result	Foundry Actual result	conclusion	Improvement
1	testTransferBatch	Hardhat	transferBatch	The defined amount of tokens is transferred	yes	As expected	As expected	Ok	
2	testCannotTransferBatchMoreTokensThanOwn	Hardhat	transferBatch	The transaction is reverted	-	As expected	As expected	Ok	
3	testCannotTransferBatchIfLengthMismatchMissingAddresses	Hardhat	transferBatch	The transaction is reverted	-	As expected	As expected	Ok	
4	testCannotTransferBatchIfLengthMismat	Hardhat	transferBatch	The transaction is reverted	-	As expected	As expected	Ok	

	chTooManyAddresses								
5	testCannotTransferBatchIfTOSIsEmpty	Hardhat	transferBatch	The transaction is reverted	-	As expected	As expected	Ok	

Test L – Deployment

Test to check the value at deployment

Target File : All files

Test files: Proxy.test.js (Hardhat)

id	Test function	Hardhat/ Foundry	Target function	Expected result	Event check	Actual result	Conclusion	Improvement
1	testCannotDeployProxyWithAdminSetToAddressZero	Hardhat	Authorizationmodule - initialize	The transaction is reverted with the right error message	-	As expected	Ok	
2	testCannotDeployStandaloneWithAdminSetToAddressZero	Hardhat	Authorizationmodule - initialize	The transaction is reverted with the right error message	-	As expected	Ok	