DeedzCoin Smart Contract Error Scenarios

This document outlines various error scenarios that can occur while interacting with the DeedzCoin smart contract. Each scenario is accompanied by a description of the error and an example usage case.

1. NotSupplier

Description: This error occurs when a function is executed by an entity that is not recognized as a supplier.
Example Usage: If a function is restricted to be called only by the supplier (as per the onlySupplier modifier), and a non-supplier entity attempts to call it, this error will be triggered.

2. Invalid_SupplierZeroAddress

Description: This error occurs when the transferSupplierRole function is executed with a zero-address provided as the new supplier address.
Example Usage: If an attempt is made to transfer the supplier role to the zero address, this error will be triggered.

3. Invalid_TransferTimeInThePast

Description: This error occurs when the transferWithLockActualTime function is executed with a provided time that is lower than the current block timestamp.
Example Usage: If someone tries to lock tokens until a time in the past, this error will be triggered.

4. Invalid_TransferToZeroAddress

Description: This error occurs when the transferWithLockActualTime or transferWithLock function is executed, and the provided address is the zero address.
Example Usage: If someone tries to transfer and lock tokens to the zero address, this error will be triggered.

5. Invalid_TokensAlreadyLocked

Description: This error occurs when the transferWithLockActualTime or transferWithLock function is executed with a provided reason that already exists for a different lock amount.
Example Usage: If someone tries to lock tokens with a reason that has already been used for a different amount of tokens, this error will be triggered.

6. Invalid_TransferAmountZero

Description: This error occurs when the transferWithLockActualTime or transferWithLock function is executed with a lock amount of zero.
Example Usage: If someone tries to lock zero tokens, this error will be triggered.

7. Invalid_TokensNotLocked

Description: This error occurs when the extendLock or increaseLockAmount function is executed with a provided reason for locked tokens that does not exist or has already been claimed.
Example Usage: If someone tries to extend or increase the lock of tokens for a reason that does not exist or has already been claimed, this error will be triggered.

By being aware of these error scenarios and their causes, users and developers can effectively interact with the DeedzCoin smart contract, reducing the likelihood of encountering unexpected issues.