



**ANASTASIA LABS**

**Proof of Achievement – Milestone 2**

Testing Suite for Lucid-Evolution

**Project Number** 1100024

**Project Manager** Jonathan Rodriguez

## Contents

<b>Testing Suite .....</b>	<b>1</b>
Packages .....	1
GIF Testrun .....	1
Test cases overview .....	2

**Project Name** Lucid Evolution: Redefining Off-Chain Transactions in Cardano

**URL** [Catalyst Proposal](#)

## Testing Suite

Our testing suite, integrated into the Lucid Evolution library through GitHub Actions, automatically runs on `push` to the main branch and during pullrequest events. It includes tests in order to ensure each function performs as expected.

Automated tests are triggered to validate code functionality.

## Packages

### Lucid

- [coinselection.test.ts](#)
- [onchain.test.ts](#)
- [read.test.ts](#)
- [tx.test.ts](#)
- [txHash.test.ts](#)
- [wallet.test.ts](#)

### Utils

- [apply-param.test.ts](#)
- [cbor.test.ts](#)
- [native.test.ts](#)
- [utxo.test.ts](#)

### Provider

- [koios.test.ts](#)
- [kupmios.test.ts](#)

## GIF Testrun

[Test Result](#)

This GIF, **an automated test** running in terminal, showcases that our test packages are working as intended by running succesful test cases

## Test cases overview

To summarize our testing process, you can find the descriptions for our test cases for each test cluster. Our testing suite verifies that the added functionalities to lucid-evolution work as intended and we are always on the lookout for additional findings to improve our library with.

### Lucid

#### Coin Selection

This test case ensures the functionality of the `coinSelection`. By checking for various scenarios, each test case focuses on specific aspects of the selection algorithm, ensuring that the function works correctly under different conditions and efficiently selects the appropriate UTxOs based on the input criteria

#### Onchain Tests

This comprehensive test suite ensures that various functionalities related to transactions, staking, minting, burning, and parameterized contracts work as expected in the lucid-evolution library

#### Read Tests

In order to ensure that the library correctly integrates with a provider API to perform operations like wallet selection, UTxO retrieval

#### Tx Test

This scripts is designed to validate the minting and burning functions of tokens by verifying transaction creation signing and submission.

#### Tx Hash Test

It ensures the correctness of the transaction signing and hash computation-. It uses a predefined transaction and signs it with a selected wallet, than computes the hash in order to compare the computed hash with the signed transaction hash

#### Wallet Test

To validate wallet management. Things like switching providers, generating seed phrases and correctly selecting a wallet

## Utility

### Apply Parameters

Its designed to validate if the functionality to apply parameters to a script using `applyParamsToScript`. It tests if the function correctly embeds paramteres into a given script

### CBOR Test

Ensures the functionality of the `applyDoubleCborEncoding`, to check CBOR strings encoding consistency and accuracy

### Native Test

`parseCMLNative` needs to accurately parse native scripts, by the verifying the interpretation of a complex script we confirm the functions reliability in handling this scenario

### UTxO Test

We check for the accurate deserialization by `coreToUtxo` of CBOR ecnoded UTxO data into JS objects.

## Provider

### Koios

The Koios API will be called for various functionalities like `koios.getProtocolParameters()`, `koios.getUtxos(address)` etc. to ensure reliability

### Kupmios

We verify the kupmios service to interact with the blockchain. Tests for retrieval of parameters, UTxOs, delegations, datums and more are included to see the resilience