TxPipe Docs

# Testing tx3

The testing tool is designed to assist developers in writing tests and validating their tx3 files. It provides a structured way to define test cases and verify the expected outcomes of executing tx3 code.

## Writing

Tests for tx3 files are written using the `TOML` format. A TOML test file typically includes the following sections:

### Wallets

Define the initial state of wallets involved in the test. You can use descriptive names like `bob` and `alice` for clarity. Each wallet should specify its initial balance or assets.

```toml
[[wallets]]
name = "bob"
balance = 10000000

[[wallets]]
name = "alice"
balance = 5000000

# Not implemented yet
[[wallets.assets]]
name = "Token"
policy = "0x"
amount = 1
```

- **`name`**

  A human-readable wallet name.

  **example**: `"bob"`

- **`balance`**

  The initial balance that the wallet will contain before the test starts.

  **type**: `number`

  **example**: `10000000`

# Transactions

Specify the test cases of the tx3 template.

```
[[transactions]]
description = "bob sends 2 ada to alice"
template = "transfer"
signer = "bob"
wait_block = true
args = { quantity = 2000000, sender = "@bob", receiver = "@alice" }

[[transactions]]
description = "alice sends back 1 ada to bob"
template = "transfer"
signer = "bob"
args = { quantity = 10000000, sender = "@alice", receiver = "@bob" }
```

- **`description`**

  A human-readable explanation of the transaction.

  **example**: `"bob sends 2 coins to alice"`

- **`template`**

  The name of the predefined transaction template to use from tx3 file.

  **example**: `"transfer"`

- **`signer`**

  The wallet that will sign the transaction. This should match one of the defined wallets.

  **example**: `"bob"`

- **`wait_block`**

  Indicates whether to wait for a block confirmation before proceeding.

  **type**: `boolean`

  **example**: `true`

- **`args`**

  Arguments required by the `template`. These are passed as key-value pairs. Arguments that start with `@` are mapped to wallet address, so `@bob` will be mapped to `addr_...`.

**type**: `object`

**example**: `{ quantity = 2000000, sender = "@alice", receiver = "@bob" }`

# Expect

Define the expect state after the tx3 function has been executed. This section should specify the expected final values for wallets, outputs, or any other relevant state.

```
[[expect]]
type = "Balance"
wallet = "bob"
amount = 9638899

[[expect]]
type = "Balance"
wallet = "alice"
amount = { target = 4638899, threshold = 300000 }
```

- **`type`**

  The type of expect, each type will validate the output different. As far, only `Balance` is implemented.

  **type**: `string`

  **example**: `Balance`

Balance

- **`wallet`**

  The name of the wallet that must be validated.

  **type**: `string`

  **example**: `bob`

- **`amount`**

  The amount validation, it can be absolute or aproximated. Furthermore, `target` and `threshold` are a way to validate aproximated the final balance.

  **type**: `number | object`

  **example**: `9638899 | { target = 4638899, threshold = 300000 }`

# Example

Here is a complete example of a TOML test file:

```toml
file="./main.tx3"

[[wallets]]
name = "bob"
balance = 10000000

[[wallets]]
name = "alice"
balance = 5000000

[[transactions]]
description = "bob sends 2 ada to alice"
template = "transfer"
signer = "bob"
wait_block = true
args = { quantity = 2000000, sender = "@bob", receiver = "@alice" }

[[transactions]]
description = "alice sends 2 ada to bob"
template = "transfer"
signer = "alice"
args = { quantity = 2000000, sender = "@alice", receiver = "@bob" }

[[expect]]
type = "Balance"
wallet = "bob"
amount = 9638899

[[expect]]
type = "Balance"
wallet = "alice"
amount = { target = 4638899, threshold = 300000 }
```

This test file defines two wallets, bob and alice, with initial balance amounts. It specifies the test cases for a transfer template and defines the expected balance amounts for both wallets after the test cases executed.