The Move Book

Move is a language for writing smart contracts - programs that are stored and run on the blockchain. A single program is organized into a package. A package is published on the blockchain and is identified by an address . A published package can be interacted with by sending transactions calling its functions. It can also act as a dependency for other packages.

To create a new package, use the sui move new command. To learn more about the command, run sui move new --help .

Package consists of modules - separate scopes that contain functions, types, and other items.

Locally, a package is a directory with a Move.toml file and a sources directory. The Move.toml file - called the "package manifest" - contains metadata about the package, and the sources directory contains the source code for the modules. Package usually looks like this:

The tests directory is optional and contains tests for the package. Code placed into the tests directory is not published on-chain and is only available in tests. The examples directory can be used for code examples, and is also not published on-chain.

During development, package doesn't have an address and it needs to be set to 0x0 . Once a package is published, it gets a single unique address on the blockchain containing its modules' bytecode. A published package becomes immutable and can be interacted with by sending transactions.

Package Structure

Locally, a package is a directory with a Move.toml file and a sources directory. The Move.toml file - called the "package manifest" - contains metadata about the package, and the sources directory contains the source code for the modules. Package usually looks like this:

```
bash sources/ my_module.move another_module.move ... tests/ ... examples/ using_my_module.move
Move.toml
```

The tests directory is optional and contains tests for the package. Code placed into the tests directory is not published on-chain and is only available in tests. The examples directory can be used for code examples, and is also not published on-chain.

During development, package doesn't have an address and it needs to be set to 0x0 . Once a package is published, it gets a single unique address on the blockchain containing its modules' bytecode. A published package becomes immutable and can be interacted with by sending transactions.

```
bash 0x... my_module: <bytecode> another_module: <bytecode>
```

Published Package

During development, package doesn't have an address and it needs to be set to 0x0 . Once a package is published, it gets a single unique address on the blockchain containing its modules' bytecode. A published package becomes immutable and can be interacted with by sending transactions.

```
bash 0x... my_module: <bytecode> another_module: <bytecode>
```

Links