
twig Documentation

Release 0.1.0-alpha.3

Jason Carver

Oct 25, 2018

Contents

1	Contents	3
1.1	Overview	3
1.1.1	Start your own Vyper project	3
1.2	Release Notes	4
1.2.1	v0.1.0-alpha.1	4
2	Indices and tables	5

A tool for Ethereum smart contract development.

1.1 Overview

1.1.1 Start your own Vyper project

Twig lets you write tests using `web3.py` and `pytest-ethereum` against your `Vyper` smart contracts.

Twig setup

- clone twig repo
- `python -m venv venv` (note: requires `>=python3.6`)
- `. venv/bin/activate`
- `pip install -e .`

Project setup

Note: `contracts/name_registry.vy` and `tests/test_name_registry.py` have been included as examples on how to get started, but are safe to delete.

- create your vyper contract file in `contracts/` (i.e. `root/contracts/example.vy`)
- write your vyper contract test file in `tests/` (i.e. `root/tests/test_example.py`)
- create your contract instance to test against using `pytest-ethereum`'s **'`deployer`'** [`<https://pytest-ethereum.readthedocs.io/en/latest/overview.html#deployer>`](https://pytest-ethereum.readthedocs.io/en/latest/overview.html#deployer). In the following example, the `my_contract` fixture returns a `web3.contract` instance for testing against.

```
@pytest.fixture
def my_contract_package(deployer):
    # returns an ethpm.Package instance loaded with a "my_contract" deployment on the_
    ↪ethpm.Package.w3 instance
    return deployer.deploy("my_contract")

@pytest.fixture
def my_contract(my_contract_package):
    # returns an my_contract web3.contract instance to test against
    return my_contract_package.deployments.get_instance("my_contract")
```

1.2 Release Notes

1.2.1 v0.1.0-alpha.1

- Launched repository, claimed names for pip, RTD, github, etc

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`