
testipynb Documentation

Release 0.0.2

Lindsey Heagy

Jan 24, 2019

Contents

1	why?	3
2	installation	5
3	usage	7
4	connections	9
5	API	11
5.1	API	11
Python Module Index		13

Unit-testing for a collection of jupyter notebooks. `testipynb` relies on `nbconvert` to run the notebooks and catches errors so that they are output (with syntax highlighting!) when unit-tests are run.

CHAPTER 1

why?

- If you want to share your notebooks and be confident that they `_should_` work on someone else's machine
- If you are using notebooks to generate figures in a publication and want to ensure they are reproducible (powerful when connected with [cron jobs on travis-ci](#))

```
627 ----- Testing TEM_VerticalConductor_2D_inversion_load.ipynb -----
628
629
630 ... TEM_VerticalConductor_2D_inversion_load FAILED
631
632 ImportError in cell [1]
633 -----
634 from SimPEG import EM, Mesh, Utils
635 import numpy as np
636 from scipy.constants import mu_0
637 from scipy.interpolate import griddata
638 import matplotlib.pyplot as plt
639 from pymatsolver import PardisoSolver
640 from SimPEG import Maps
641 %matplotlib inline
642 -----
643
644
645 ----- >> begin Traceback << -----
646
647
648 ImportErrorTraceback (most recent call last)
649 <ipython-input-1-e892c3c3b7c9> in <module>()
650     4 from scipy.interpolate import griddata
651     5 import matplotlib.pyplot as plt
652 ----> 6 from pymatsolver import PardisoSolver
653     7 from SimPEG import Maps
654     8 get_ipython().magic(u'matplotlib inline')
655
656 ImportError: cannot import name PardisoSolver
657
658
659 ----- >> end Traceback << -----
660
```


CHAPTER 2

installation

```
pip install testipynb
```


CHAPTER 3

usage

```
import testipynb

NBDIR = '../notebooks'

Test = testipynb.TestNotebooks(directory=NBDIR)
Test.assertTrue(Test.run_tests())
```

or in a unit-test file:

```
import testipynb
import unittest

NBDIR = '../notebooks'

Test = testipynb.TestNotebooks(directory=NBDIR, timeout=2100)
TestNotebooks = Test.get_tests()

if __name__ == "__main__":
    unittest.main()
```


CHAPTER 4

connections

testipynb is used in:

- https://github.com/simpeg-research/heagy_2018_AEM

If you use testipynb in one of your repositories and would like it listed, please [edit this file](#)

5.1 API

Module for testing a repository of Jupyter Notebooks

class `testipynb.testipynb.TestNotebooks (**kwargs)`

Class that generates a suite of tests for a directory of notebooks.

```
import testipynb
Test = TestNotebooks(directory="notebooks")
assert True(Test.run_tests())
```

or if you are using pytest, you can create a file called *test_notebooks.py*

```
import testipynb
Test = testipynb.TestNotebooks(directory="notebooks")
TestNotebooks = Test.get_tests()
```

and from a command line, run

```
pytest test_notebooks.py
```

Required Properties:

- **directory** (`String`): directory where the notebooks are stored, a unicode string, Default: `.`
- **ignore** (a list of `String`): list of notebooks to ignore when testing, a list (each item is a unicode string)
- **py2_ignore** (a list of `String`): list of notebook names to ignore if testing on python 2, a list (each item is a unicode string)
- **timeout** (`Integer`): timeout length for the execution of the notebook, an integer in range `[0, inf]`, Default: `600`

directory

directory (`String`): directory where the notebooks are stored, a unicode string, Default: `.`

get_tests (*obj=None*)

Create a unittest.TestCase object to attach the unit tests to.

ignore

ignore (a list of `String`): list of notebooks to ignore when testing, a list (each item is a unicode string)

py2_ignore

py2_ignore (a list of `String`): list of notebook names to ignore if testing on python 2, a list (each item is a unicode string)

run_tests ()

Run the unit-tests. Returns `True` if all tests were successful and code `False` if there was a failure.

```
import nbtest
test = nbtest.TestNotebooks(directory='./notebooks')
passed = test.run_tests()
assert (passed)
```

test_dict

dictionary of the name of the test (keys) and test functions (values) built based upon the directory provided

timeout

timeout (`Integer`): timeout length for the execution of the notebook, an integer in range `[0, inf]`, Default: 600

t

`testipynb.testipynb`, [11](#)

D

directory (testipynb.testipynb.TestNotebooks attribute),
[11](#)

G

get_tests() (testipynb.testipynb.TestNotebooks method),
[11](#)

I

ignore (testipynb.testipynb.TestNotebooks attribute), [12](#)

P

py2_ignore (testipynb.testipynb.TestNotebooks attribute),
[12](#)

R

run_tests() (testipynb.testipynb.TestNotebooks method),
[12](#)

T

test_dict (testipynb.testipynb.TestNotebooks attribute), [12](#)

testipynb.testipynb (module), [11](#)

TestNotebooks (class in testipynb.testipynb), [11](#)

timeout (testipynb.testipynb.TestNotebooks attribute), [12](#)