testipynb Documentation

Release 0.0.3

Lindsey Heagy

Aug 12, 2020

Contents

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)

```
---------------------- Testing TEM_VerticalConductor_2D_inversion_load.ipynb ----------------------


 ... TEM_VerticalConductor_2D_inversion_load FAILED

ImportError in cell [1]
-----------
from SimPEG import EM, Mesh, Utils
import numpy as np
from scipy.constants import mu_0
from scipy.interpolate import griddata
import matplotlib.pyplot as plt
from pymatsolver import PardisoSolver
from SimPEG import Maps
%matplotlib inline
-----------


---------------- >> begin Traceback << ----------------


ImportErrorTraceback (most recent call last)
<ipython-input-1-e892c3c3b7c9> in <module>()
      4 from scipy.interpolate import griddata
      5 import matplotlib.pyplot as plt
----> 6 from pymatsolver import PardisoSolver
      7 from SimPEG import Maps
      8 get_ipython().magic(u'matplotlib inline')

ImportError: cannot import name PardisoSolver


---------------- >> end Traceback << ----------------
```

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

CHAPTER 5

API

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")
assertTrue(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

Python Module Index

t

Index