

Tuesday: Testing Django Apps

Testing Django Apps

As we discussed earlier in Flask. Good code is code that contains tests. Django allows us to write tests for our applications very easily. Each app contains a test module where unit tests should be defined.

news/tests.py

```
from django.test import TestCase
from .models import Editor, Article, tags
```

Django provides a test module `django.test` where we import the `TestCase` class. We then import our models from the models module.

Testing Instance

We can now write our first test. We want to test our Editor module.

```
class EditorTestClass(TestCase):

    # Set up method
    def setUp(self):
        self.james= Editor(first_name = 'James', last_name = 'Muriuki', email = 'james@moringaschool.com')
```

We first create `EditorTestClass` which is the test class for the Editor module. This class inherits from the `TestCase` class. We then create a `setUp` method that allows us to create an instance of the Editor class before every test.

```
# Testing instance
def test_instance(self):
    self.assertTrue(isinstance(self.james, Editor))
```

We create a `test_instance` test to confirm that the object is being instantiated correctly.

Running Tests

Let us see how we can run our tests.

```
(virtual)$ python3.6 manage.py test news
System check identified no issues (0 silenced).
...
-----
Ran 1 tests in 0.043s

OK
Destroying test database for alias 'default'...
```

Django creates a test database for us runs our tests and then destroys the database once the tests are done running.

Testing the Save method

Now we can see how to write tests for sthe save functionality.

```
# Testing Save Method
def test_save_method(self):
    self.james.save_editor()
    editors = Editor.objects.all()
    self.assertTrue(len(editors) > 0)
```

We can run this to confirm the test fails. This is because we haven't defined the `save_editor()` method in our model.

```
(virtual)$ python manage.py test news
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.E
=====
ERROR: test_save_method (news.tests.EditorTestClass)
-----
Traceback (most recent call last):
  File "/home/james/Desktop/exercisefiles/moringaTribune/tribune/news/tests.py", line 20, in test_
save_method
    self.james.save_editor()
AttributeError: 'Editor' object has no attribute 'save_editor'

-----
Ran 2 tests in 0.015s

FAILED (errors=1)
Destroying test database for alias 'default'...
```

Let us define this method.

```
class Editor(models.Model):
    first_name = models.CharField(max_length =30)
    last_name = models.CharField(max_length =30)
    email = models.EmailField()

    def __str__(self):
        return self.first_name

    def save_editor(self):
        self.save()
```

We can now run the tests again to confirm it has passed.

Task: Now that you see how we create and run tests in Django Create the remaining tests. Remember to create tests for the Article and tags models:

1. Test for deleting a model object.
2. Test for displaying all model objects saved.
3. Test for updating single object properties.