

MIT AITI

Django Lab 1: Writing your first application



In this lab you will write your first Django application!

Everyone must complete this lab. Use Ubuntu. All file editing will be done in gedit (Text Editor). In this lab, each step shows you the exact commands to run after the \$ sign. Make sure you type everything exactly as shown. Be especially careful with Python files – indentation matters.

To set up for this lab:

Open VirtualBox and open the AITI VM. Maximize it so it takes up the whole computer screen – you won't need any programs for this lab other than VirtualBox.

Open Terminal. Type:

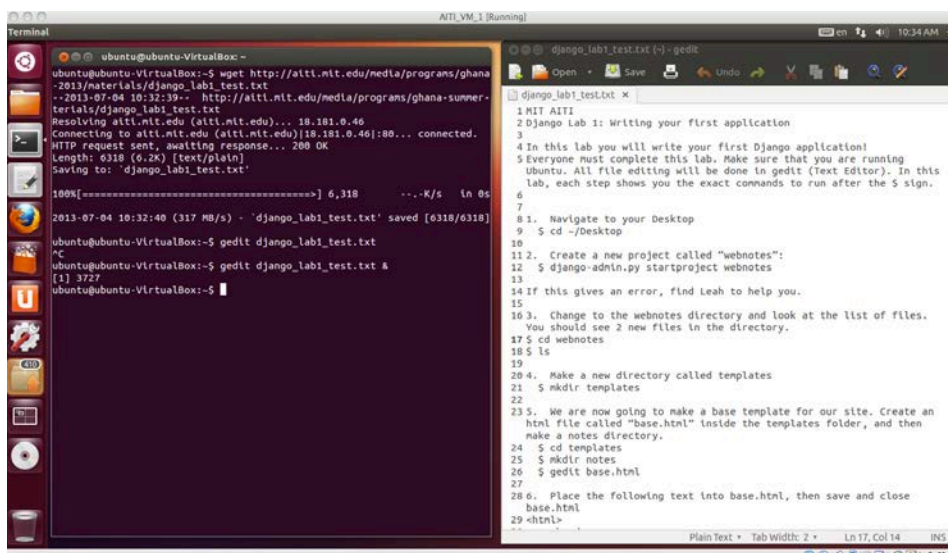
```
$ wget http://aiti.mit.edu/media/programs/ghana-summer-2013/materials/django_lab1.txt
```

This will download this lab as a text file so you can copy and paste the code for the files. If the internet's not working, don't worry -- you can just type in the code from this handout.

Open the lab with:

```
$ gedit django_lab1.txt &
```

I recommend dragging and resizing your terminal and gedit windows so each one takes up half the screen. This makes it easy to use both at once. Your screen should look something like this:



Now you're ready to start the lab!

Steps to do the lab:

1. Navigate to your Desktop

```
$ cd ~/Desktop
```

2. Create a new project called "webnotes":

```
$ django-admin.py startproject webnotes
```

If this gives an error, find Leah to help you get set up.

3. Change to the `webnotes` directory and look at the list of files. You should see 2 new files in the directory.

```
$ cd webnotes
```

```
$ ls
```

4. Make a new directory called `templates`

```
$ mkdir templates
```

5. We are now going to make a base template for our site. Create an html file called "base.html" inside the `templates` folder, and then make a `notes` directory.

```
$ cd templates
```

```
$ mkdir notes
```

```
$ gedit base.html
```

6. Place the following text into `base.html`, then save and close `base.html`

```
<html>
  <head>
    <meta http-equiv="Content-type" content="text/html;
charset=utf-8" />
    <title>Webnotes</title>
  </head>
  <body>
    <div id="content">
      {% block content %}
      {% endblock %}
    </div>
  </body>
</html>
```

7. Change into the `notes` directory and add two files: `detail.html` and `list.html`. These files will extend the `base.html` file that you have created, by overriding the `content` section.

```
$ cd notes
```

```
$ gedit detail.html
```

```
{% extends 'base.html' %}
{% block content %}
  <h2>{{ note.title }}</h2>
  <p>{{ note.content }}</p>
{% endblock %}
```

```
$ gedit list.html
```

```
{% extends 'base.html' %}
```

```

{% block content %}
    {% if note_list %}
    <ul>
        {% for note in note_list %}
            <li><a href="{% url 'notes_detail'
note.id %}">{{ note.title }}</a></li>
        {% endfor %}
    </ul>
    {% else %}
    You don't have webnotes
    {% endif %}
{% endblock %}

```

8. Edit the `settings.py` file so that django knows where to look for your templates:

```

$ cd ~/Desktop/webnotes/webnotes
$ gedit settings.py

```

a. At the top of the file, add these two lines:

```

import os # put this at the top of the file
PROJECT_ROOT =
os.path.realpath(os.path.dirname(os.path.dirname(__file__)))

```

b. find the line that starts with "TEMPLATE_DIRS". You can search within a file using ctrl-F. Replace the lines with the following:

```

TEMPLATE_DIRS = (
    os.path.join(PROJECT_ROOT, 'templates')
)

```

9. Keep `settings.py` open and edit the database configuration, by specifying `sqlite3`.

Find the following two variables in `settings.py` and edit them to have these values:

```

'ENGINE': 'django.db.backends.sqlite3',
'NAME': 'webnotes.db',

```

10. Now start (create) a new application inside this webnotes project:

```

$ cd ~/Desktop/webnotes
$ django-admin.py startapp notes

```

11. The application will be placed in a new folder called "notes". Navigate into it and look at the list of files.

```

$ ls
$ cd notes
$ ls

```

12. Open `models.py` and add a notes model *****

```

$ gedit models.py

```

```

from django.db import models
class Notes(models.Model):
    title = models.CharField(max_length=255)
    content = models.TextField()
    def __unicode__(self): #note: 2 underscores on each side
        return self.title

```

13. Open `views.py` and add a list and detail view. These views define what data will be displayed and how it will be displayed.

```
$ gedit views.py
```

```
from django.template import Context, loader
from django.http import HttpResponseRedirect
from models import Notes

def notes_list(request):
    note_list = Notes.objects.all()
    t = loader.get_template('notes/list.html')
    c = Context({
        'note_list': note_list,
    })
    return HttpResponseRedirect(t.render(c))

def notes_detail(request, id):
    note = Notes.objects.get(pk=id)
    t = loader.get_template('notes/detail.html')
    c = Context({
        'note': note,
    })
    return HttpResponseRedirect(t.render(c))
```

14. Create a new file called “`urls.py`” and define some URLs that will access your views:

```
$ gedit urls.py
```

```
from django.conf.urls.defaults import *
import views

urlpatterns = patterns('',
    url(r'^list/$', views.notes_list, name='notes_list'),
    url(r'^detail/(?P<id>\d+)/$', views.notes_detail,
        name='notes_detail'),
)
```

15. Create a new file called `admin.py` and register the notes application with the built-in admin application

```
$ gedit admin.py
```

```
from notes.models import Notes
from django.contrib import admin

admin.site.register(Notes)
```

16. Navigate back to `webnotes/webnotes`, open the `settings.py` file, and go to the `INSTALLED_APPS` section. Add ‘notes’ and uncomment the

```
django.contrib.admin app.
```

```
$ cd ~/Desktop/webnotes/webnotes
```

```
$ gedit settings.py
```

```
INSTALLED_APPS = (  
    ...  
    'notes',  
    'django.contrib.admin',  
)
```

17. Open the `urls.py` for your project (not the one for the notes app). Tell your project `urls.py` that it should include the URLs from your notes app. Uncomment the references to the admin urls.

```
$ gedit urls.py
```

```
...  
from django.contrib import admin  
admin.autodiscover()  
...  
urlpatterns = patterns('',  
    url(r'^notes/', include('notes.urls')),  
    ...  
    url(r'^admin/', include(admin.site.urls)),  
    ...
```

18. Open a new tab to run the server from, by right-clicking and selecting “Open tab”. Go to the top-level directory containing `manage.py`. Tell django to build a database based on your models.

```
$ cd ..  
$ python manage.py syncdb
```

When asked to make a superuser, say “yes” and enter a username, email, and password. DO NOT FORGET THIS.

19. Run the development server

```
$ python manage.py runserver
```

Create superuser

20. visit <http://localhost:8000/admin>, and add a note

21. visit <http://localhost:8000/notes/list> and see if your note is there

22. Once all of this is done, it's time to make a change. Add an author to the notes application and output it in the detail page. Here's how to do it:

- a. Open `models.py`

```
$ gedit ~/Desktop/webnotes/notes/models.py
```

- b. Add an author field.

Hint: it's very similar to the `title` field.

- c. Edit the details template

```
$ gedit ~/Desktop/webnotes/templates/notes/detail.html
```

- d. Output the author after the title in an `<h3>` tag

- e. Go to the terminal where you ran `python manage.py runserver`, press `ctrl+c` to stop it

- f. Delete all previous data so that you have a clean database. rm is the command to delete a file.
\$ rm webnotes.db
- g. Synchronize the database
\$ python manage.py syncdb
- h. Start the server again
\$ python manage.py runserver
- i. If your change worked, you should see an author field when adding notes in the admin interface

Congrats, you just made your first Django webapp!