

Django Blog Project Part 1: Models

This is the first part of a weeklong project in which we will use django to build a blogging website. In this lab, we will set up the models and data for a blog. In the intro lab, we did the models, views, and templates at once, but in the next several labs, we will be focusing on each part individually. In larger projects, this is how things will often work – one person will do models, one will do views, and another will do the templates. A designer doesn't need to worry about the database, and the database designer doesn't need to worry about the display style.

If you get confused, look at the resources available to you:

1. Lecture slides
 2. The intro lab
 3. Other students
 4. Django documentation
 - a. Models: <https://docs.djangoproject.com/en/1.3/topics/db/models/>
 - b. Field reference: <https://docs.djangoproject.com/en/1.3/ref/models/fields/>
 5. Google
 6. Instructors
-
1. If you are reading this, you must have **forked** and cloned the repository, right?
Good job.

Did you remember to add the `instructors` remote?
Check the readme for instructions.

2. Make sure that your working directory in terminal is the folder that you cloned.
3. Create a new django project project called “myblog” or whatever you like.
NOTE THE DOT AT THE END OF THE COMMAND BELOW. IMPORTANT

```
django-admin.py startproject myblog .
```

4. Edit your settings.py file and fill in these variables:

```
import dj_database_url
DATABASES = {
    'default' : dj_database_url(default="sqlite:/blog_dev.db")
}

TIME_ZONE = 'Africa/Accra'
```

5. Create a new application:

```
django-admin.py startapp blog
cd blog
```

6. Spend some time thinking and talking with people around you about what fields and models we should have for a blog. Suggestions are on the next page.

7. Ok so here's what we think is a good design, but feel free to do it differently.

a. Post:

- i. title (60 characters)
- ii. body (large text)
- iii. created (date created)
- iv. updated (date updated)

b. Comment:

- i. body (large text)
- ii. author (60 characters)
- iii. created (date created)
- iv. updated (date updated)
- v. post (foreign key linking Comment to Post)

8. Add 'blog' to the `INSTALLED_APPS` tuple in `settings.py`

9. Activate the admin application, and register the `Post` and `Comment` models with it.

Try to do this yourself, but if you get stuck, there are step-by-step instructions to doing it at the end of this file.

10. **Sync** the database

Again, try to do this yourself, but if you get stuck, step-by-step instructions follow

11. and **run** the server

Again, try to do this yourself, but if you get stuck, step-by-step instructions follow

12. Go to admin and add posts to confirm that everything is working

<http://localhost:8000/admin>

13. If everything is working, **add**, **commit**, and **push** your work to your github repository.

Then open the file `heroku-instructions.pdf` (in the same folder as this file) and follow the instructions there to launch your blog.

(if the file is not there, type `git pull instructors master` to update your repository with any changes that the instructors have pushed)

Step by step instructions for 9, 10, 11

9: Note: '...' simply stands for "any other code here"

a. Open `settings.py` and change the installed app setting:

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

b. Open `urls.py` and uncomment the admin lines:

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

c. At the top of your `models.py`, add:

```
from django.contrib import admin
```

d. At the end of your `models.py`, add:

```
admin.site.register(Post)  
admin.site.register(Blog)
```

10: `python manage.py syncdb`

11: `python manage.py runserver`