MIT AITI
Mobile, Python, Software Development

Lab 04: Views

This is a continuation of Lab 3. You will now be adding views to your Blog app.

If you get stuck, take a look at these resources:

1. Lecture slides
2. Previous labs
3. Other group members
4. Django documentation
    a. Making Queries: https://docs.djangoproject.com/en/1.4/topics/db/queries/
    b. QuerySets: https://docs.djangoproject.com/en/dev/ref/models/querysets/
    c. Views: https://docs.djangoproject.com/en/1.4/topics/http/views/
    d. Urls: https://docs.djangoproject.com/en/1.4/topics/http/urls/
5. Google
6. Instructors

Steps:

1. Open up your `blog` project
   ```
   $ cd ~/Desktop/myblog
   ```

2. Add the following line to `myblog/urls.py`
   ```
   url(r'^blog/', include('blog.urls')),
   ```

3. Open up your `blog` app
   ```
   $ cd blog
   ```

4. Create a `urls.py` file and add this text (check for correct indentations – it's formatted slightly wrong in this document!):
   ```
   from django.conf.urls import patterns, include, url

   urlpatterns = patterns('',
       url(r'^$', 'blog.views.home'),
       url(r'^list/(?P<limit>\d+)?$', 'blog.views.blog_list'),

   url(r'^(detail|info)/(?P<id>\d+)/((?P<showComments>.*)/)?$'
   , 'blog.views.blog_detail'),
   )
   ```

5. Now add another `url` field that searches for a url like search/something, where something is zero or more word characters. Capture this part of the url and pass it to `blog.views.blog_search`. For example, yourdomain.com/blog/search/gorilla passes gorilla to the view method as the first parameter.

6. Now edit the `views.py` file and put in the following:

```
from django.template import Context, loader
from django.http import HttpResponse
from models import Blog, Comment

def blog_list(request, limit=100):
  blog_list = Blog.objects.all()
  print type(blog_list)
  print blog_list
  return HttpResponse('going to give a list')

def blog_detail(request, id, showComments=False):
    pass
def blog_search(request, term):
    pass

def home(request):
  print 'blog home under construction'
  return HttpResponse('hello world')
```

7. You need to implement two views, the `blog_detail` and `blog_search`.
   a) For `blog_detail`, get a single blog and print it (this will show up in the `runserver` terminal when you load the page). Then, if `showComments` is `True`, get the comments associated with the current blog (the current blog is the blog with the id), and print them. To retrieve a single blog, use `Blog.objects.get(pk=id)`. To retrieve the comments of blog `b`, use `b.comment_set.all()`.
   b) For `blog_search`, get all the blogs that contain a search string (case insensitive), and print them.

10. Add some blog posts with comments associated with them

11. Run your server (`python manage.py runserver`), and load the following pages, making sure they work:
   a) blog/list
   b) blog/detail/1
   c) blog/search/<your term here>
   d) blog/