



CMPUT 410 Lab 6-7 – Django

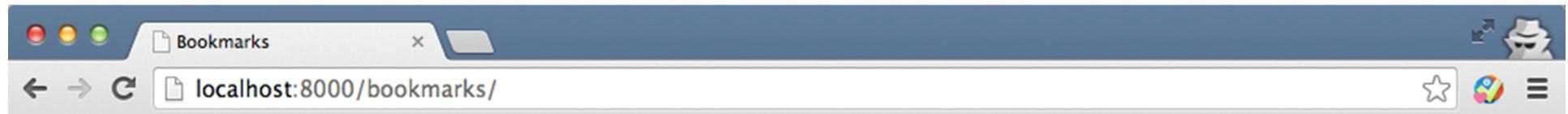
Ali Sajedi



Feb 25, 2015

Overview

- ▶ **Exercises**
 - ▶ a simple bookmarking app
 - ▶ files given to you:
 - the UI-Prototype (HTML/CSS/JavaScript)
 - populate_db.py
 - ▶ your task:
 - integrate them with django into a functional website
- ▶ **Tutorial**
 - ▶ django basics
 - ▶ building the app
- ▶ **References**



Bookmarks

My Links

Add a link

Tags

Official Python Tutorial

python tutorial

How to Think like a Computer Scientist

python

Learn Python in 10 Minutes

python tutorial

Official Django Tutorial

tutorial django

Django Rocks

django

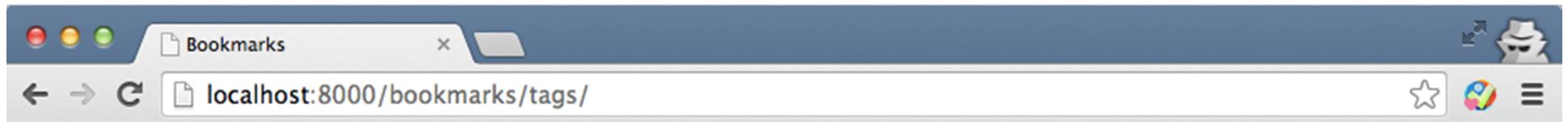
How to Tango with Django

tutorial django

Bottle

bottlepy bottle

Flask



Bookmarks

My Links

Add a link

Tags

python

tutorial

django

bottlepy

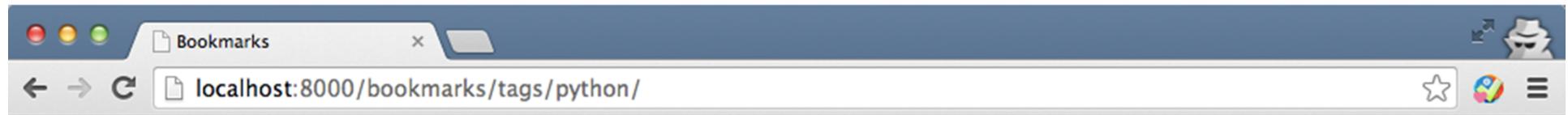
bottle

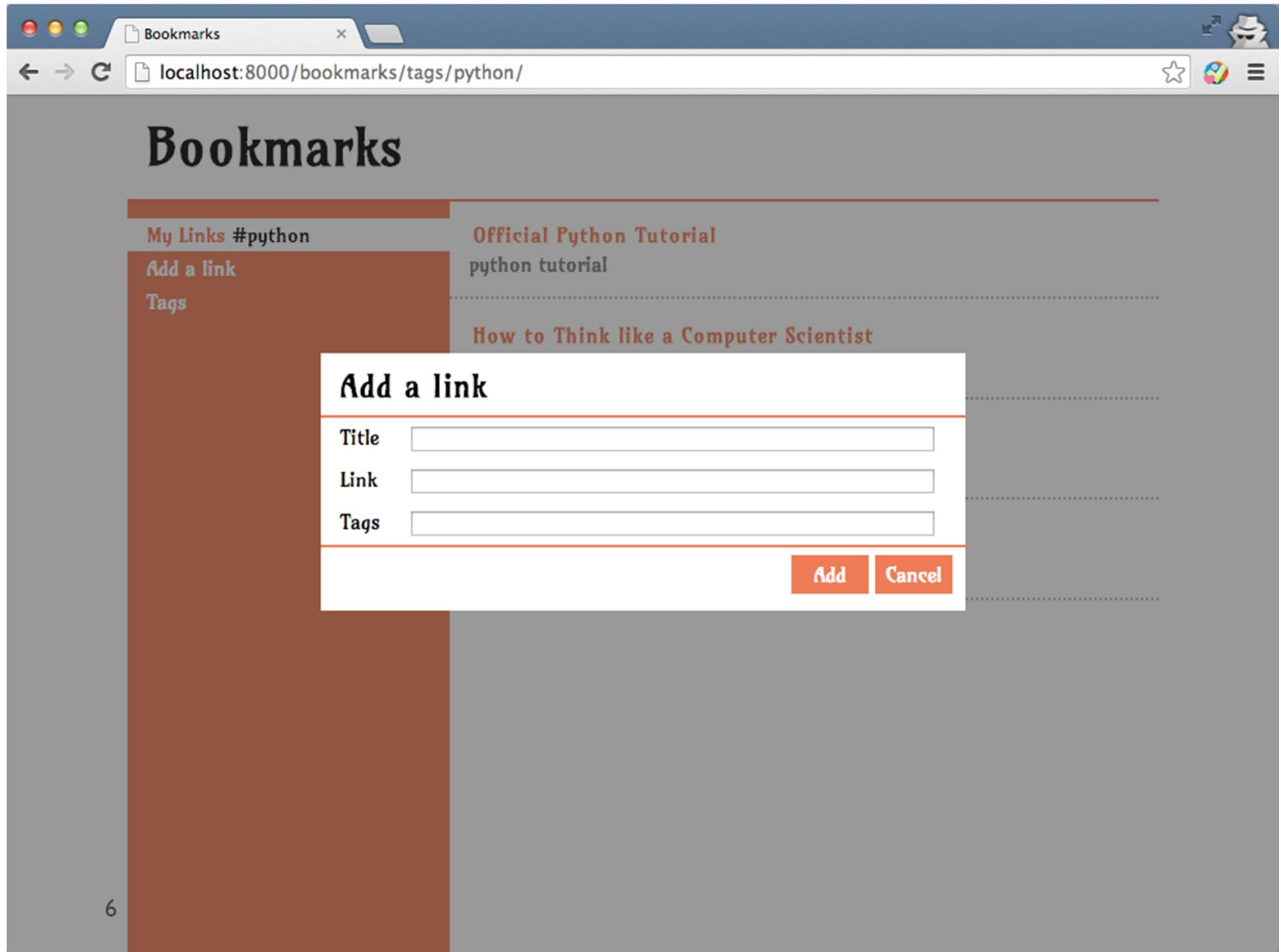
flask

bookmark

1

2





Exercises

- ▶ “tags”
 - ▶ show a list of all the tags without duplicate
 - ▶ tags should be linked to /bookmarks/tags/<tag_name>/
- ▶ “add a link”
 - ▶ send a POST request to /bookmarks/add_link
 - ▶ store the link data in the database
 - ▶ redirect to the homepage
- ▶ Lab6 task: finishing either one
- ▶ Lab7 task: finishing both (no errors)

Note: Participation in Lab7 is mandatory.

Django Basics

▶ Getting Started

▶ Install Django

- ▶ # activate virtualenv
- ▶ \$ pip install django

▶ Create a new project

- ▶ \$ django-admin.py startproject bookmarks
- ▶ \$ cd bookmarks

▶ Create a new application

- ▶ \$./manage.py startapp main

▶ Run server

- ▶ \$./manage.py runserver
- ▶ Ctrl+C to terminate the server

Bookmark Application

- ▶ Base Directory ('bookmarks/')
 - ▶ *manage.py*
 - ▶ provides administrative commands
 - ▶ *populate_db.py*
 - ▶ initializes the database with data
 - ▶ static/
 - ▶ css/javascript files
 - ▶ templates/
 - ▶ HTML templates
- ▶ Project Directory ('bookmarks/bookmarks/')
 - ▶ *settings.py*
 - ▶ global settings and resources:
 - ▶ *urls.py*
 - ▶ routes to applications
- ▶ Application Directory ('bookmarks/main/')
 - ▶ *views.py*
 - ▶ handles requests
 - ▶ *urls.py*
 - ▶ defines URL routings
 - ▶ *models.py*
 - ▶ object-relational mapping (ORM)
 - ▶ *admin.py*
 - ▶ configures the admin panel

Bookmark Application

- ▶ **Base Directory** ('bookmarks/')
 - ▶ *manage.py*
 - ▶ provides administrative commands
 - ▶ ***populate_db.py***
 - ▶ initializes the database with data
 - ▶ **static/**
 - ▶ css/javascript files
 - ▶ **templates/**
 - ▶ HTML templates - ▶ **Project Directory** ('bookmarks/bookmarks/')
 - ▶ *settings.py*
 - ▶ global settings and resources:
 - ▶ *urls.py*
 - ▶ routes to applications - ▶ **Application Directory** ('bookmarks/main/')
 - ▶ *views.py*
 - ▶ handles requests
 - ▶ *urls.py*
 - ▶ defines URL routings
 - ▶ *models.py*
 - ▶ object-relational mapping (ORM)
 - ▶ *admin.py*
 - ▶ configures the admin panel
- ```
$ cp/lab6-7-handouts/populate_db.py .
$ cp -r/lab6-7-handouts/ui-prototype/static .
$ mkdir -p templates/main/
$ cp/lab6-7-handouts/ui-prototype/index.html
 templates/main/
```

# Bookmark Application

---

- ▶ Base Directory ('bookmarks/')
  - ▶ *manage.py*
    - ▶ provides administrative commands
  - ▶ *populate\_db.py*
    - ▶ initializes the database with data
  - ▶ static/
    - ▶ css/javascript files
  - ▶ templates/
    - ▶ HTML templates
- ▶ Project Directory ('bookmarks/bookmarks/')
  - ▶ **settings.py**
    - ▶ global settings and resources:
  - ▶ *urls.py*
    - ▶ routes to applications
- ▶ Application Directory ('bookmarks/main/')
  - ▶ *views.py*
    - ▶ handles requests
  - ▶ *urls.py*
    - ▶ defines URL routings
  - ▶ *models.py*
    - ▶ object-relational mapping (ORM)
  - ▶ *admin.py*
    - ▶ configures the admin panel

In *settings.py*:

add app name 'main' to 'INSTALLED\_APPS'

add path definitions for 'templates' and 'static' at the end

## bookmarks/bookmarks/settings.py

---

```
INSTALLED_APPS = (
 ...
 'main',
)
...
Indicates the local filesystem path for Django to get static files
STATIC_PATH = os.path.abspath(os.path.join(BASE_DIR, 'static'))
STATICFILES_DIRS = (
 STATIC_PATH,
)
Templates
TEMPLATE_PATH = os.path.abspath(os.path.join(BASE_DIR, 'templates'))
TEMPLATE_DIRS = (
 TEMPLATE_PATH,
)
```

# Bookmark Application

---

- ▶ Base Directory ('bookmarks/')
  - ▶ *manage.py*
    - ▶ provides administrative commands
  - ▶ *populate\_db.py*
    - ▶ initializes the database with data
  - ▶ static/
    - ▶ css/javascript files
  - ▶ templates/
    - ▶ HTML templates
- ▶ Project Directory ('bookmarks/bookmarks/')
  - ▶ *settings.py*
    - ▶ global settings and resources:
  - ▶ *urls.py*
    - ▶ routes to applications
- ▶ Application Directory ('bookmarks/main/')
  - ▶ *views.py*
    - ▶ handles requests
  - ▶ *urls.py*
    - ▶ defines URL routings
  - ▶ **models.py**
    - ▶ object-relational mapping (ORM)
  - ▶ *admin.py*
    - ▶ configures the admin panel

Define classes in bookmarks/main/models.py

`class Link: ...`

`class Tag: ...`

## bookmarks/main/models.py

---

```
from django.db import models

class Tag(models.Model):
 name = models.CharField(max_length=128,unique=True)
 def __unicode__(self):
 return self.name

class Link(models.Model):
 title = models.CharField(max_length=128,unique=True)
 url = models.URLField()
 # Many to many foreign key
 tags = models.ManyToManyField(Tag)

 def __unicode__(self):
 return self.title
```

# Bookmark Application

---

- ▶ Base Directory ('bookmarks/')
  - ▶ *manage.py*
    - ▶ provides administrative commands
  - ▶ *populate\_db.py*
    - ▶ initializes the database with data
  - ▶ static/
    - ▶ css/javascript files
  - ▶ templates/
    - ▶ HTML templates
- ▶ Project Directory ('bookmarks/bookmarks/')
  - ▶ *settings.py*
    - ▶ global settings and resources:
  - ▶ *urls.py*
    - ▶ routes to applications
- ▶ Application Directory ('bookmarks/main/')
  - ▶ *views.py*
    - ▶ handles requests
  - ▶ *urls.py*
    - ▶ defines URL routings
  - ▶ *models.py*
    - ▶ object-relational mapping (ORM)
  - ▶ **admin.py**
    - ▶ configures the admin panel

Register models in bookmarks/main/admin.py

```
from django.contrib import admin
from main.models import Link, Tag
admin.site.register(Link)
admin.site.register(Tag)
```

## Synchronize the database

---

```
navigate to the base directory “bookmarks/”
```

```
$./manage.py syncdb
```

```
answer “yes” to setup admin username/password
```

```
$./manage.py runserver
```

```
visit http://localhost:8000/admin/
```

```
login with username/password
```

# Populate Testing Data

---

```
navigate to the base directory “bookmarks/”
```

```
$ python populate_db.py
```

```
if there is any error, double check this line (should match
the project name)
```

```
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'bookmarks.settings')
```

```
visit http://localhost:8000/admin/
```

# Bookmark Application

---

- ▶ Base Directory ('bookmarks/')
  - ▶ *manage.py*
    - ▶ provides administrative commands
  - ▶ *populate\_db.py*
    - ▶ initializes the database with data
  - ▶ static/
    - ▶ css/javascript files
  - ▶ templates/
    - ▶ HTML templates
- ▶ Project Directory ('bookmarks/bookmarks/')
  - ▶ *settings.py*
    - ▶ global settings and resources:
  - ▶ *urls.py*
    - ▶ routes to applications
- ▶ Application Directory ('bookmarks/main/')
  - ▶ *views.py*
    - ▶ handles requests
  - ▶ *urls.py*
    - ▶ defines URL routings
  - ▶ *models.py*
    - ▶ object-relational mapping (ORM)
  - ▶ *admin.py*
    - ▶ configures the admin panel

In **bookmarks/bookmarks/urls.py**

add url pattern:

```
url(r'^bookmarks/', include('main.urls')),
```

## bookmarks/bookmarks/urls.py

```
from django.conf.urls import patterns, include, url

from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns("",
 # Examples:
 # url(r'^$', 'bookmarks_2.views.home', name='home'),
 # url(r'^blog/', include('blog.urls')),
 url(r'^admin/', include(admin.site.urls)),
 url(r'^bookmarks/', include('main.urls'))
)
```

localhost:8000/bookmarks/

# Bookmark Application

---

- ▶ Base Directory ('bookmarks/')
  - ▶ *manage.py*
    - ▶ provides administrative commands
  - ▶ *populate\_db.py*
    - ▶ initializes the database with data
  - ▶ static/
    - ▶ css/javascript files
  - ▶ templates/
    - ▶ HTML templates
- ▶ Project Directory ('bookmarks/bookmarks/')
  - ▶ *settings.py*
    - ▶ global settings and resources:
  - ▶ *urls.py*
    - ▶ routes to applications
- ▶ Application Directory ('bookmarks/main/')
  - ▶ *views.py*
    - ▶ handles requests
  - ▶ *urls.py*
    - ▶ defines URL routings
  - ▶ *models.py*
    - ▶ object-relational mapping (ORM)
  - ▶ *admin.py*
    - ▶ configures the admin panel

Create `bookmarks/main/urls.py`

\$ `cd main`

\$ `touch urls.py`

## bookmarks/main/urls.py

---

```
from django.conf.urls import patterns, url
from main import views
```

```
urlpatterns = patterns(",
 url(r'^$', views.index, name='index'),
 url(r'^tags/$', views.tags, name='tags'),
 url(r'^tags/(?P<tag_name>\w+)/$', views.tag, name='tag'),
 url(r'^add_link/$', views.add_link, name='add_link'),
)
```

## bookmarks/main/urls.py

```
from django.conf.urls import patterns, url
from main import views
```

localhost:8000/bookmarks/

```
urlpatterns = patterns(",
 url(r'^$', views.index, name='index'),
 url(r'^tags/$', views.tags, name='tags'),
 url(r'^tags/(?P<tag_name>\w+)/$', views.tag, name='tag'),
 url(r'^add_link/$', views.add_link, name='add_link'),
)
```

## bookmarks/main/urls.py

```
from django.conf.urls import patterns, url
from main import views
```

localhost:8000/bookmarks/tags/

```
urlpatterns = patterns(",
 url(r'^$', views.index, name='index'),
 url(r'^tags/$', views.tags, name='tags'),
 url(r'^tags/(?P<tag_name>\w+)/$', views.tag, name='tag'),
 url(r'^add_link/$', views.add_link, name='add_link'),
)
```

## bookmarks/main/urls.py

```
from django.conf.urls import patterns, url
from main import views
```

localhost:8000/bookmarks/tags/<tag\_name>/

```
urlpatterns = patterns(",
 url(r'^$', views.index, name='index'),
 url(r'^tags/$', views.tags, name='tags'),
 url(r'^tags/(?P<tag_name>\w+)/$', views.tag, name='tag'),
 url(r'^add_link/$', views.add_link, name='add_link'),
)
```

# Bookmark Application

---

- ▶ Base Directory ('bookmarks/')
  - ▶ *manage.py*
    - ▶ provides administrative commands
  - ▶ *populate\_db.py*
    - ▶ initializes the database with data
  - ▶ static/
    - ▶ css/javascript files
  - ▶ templates/
    - ▶ HTML templates
- ▶ Project Directory ('bookmarks/bookmarks/')
  - ▶ *settings.py*
    - ▶ global settings and resources:
  - ▶ *urls.py*
    - ▶ routes to applications
- ▶ Application Directory ('bookmarks/main/')
  - ▶ **views.py**
    - ▶ handles requests
  - ▶ *urls.py*
    - ▶ defines URL routings
  - ▶ *models.py*
    - ▶ object-relational mapping (ORM)
  - ▶ *admin.py*
    - ▶ configures the admin panel

Add functions to bookmarks/main/views.py

```
def index(request): ... def tags(request): ...
```

```
def tag(request, tag_name): ...
```

```
def add_link(request): ...
```

```
1 from django.shortcuts import render
2 from django.shortcuts import render_to_response
3 from django.template import RequestContext
4 from django.shortcuts import redirect
5
6 from main.models import Link
7 from main.models import Tag
8
9 def index(request):
10 # Request the context of the request.
11 # The context contains information such as the client's machine details, for example.
12 context = RequestContext(request)
13
14 # Get all links
15 links = Link.objects.all()
16
17 return render_to_response('main/index.html', {'links': links}, context)
18
19 def tags(request):
20 context = RequestContext(request)
21
22 # Get all tags
23 tags = Tag.objects.all()
24
25 return render_to_response('main/tags.html', {'tags': tags}, context)
26
27 def tag(request, tag_name):
28 context = RequestContext(request)
29 the_tag = Tag.objects.get(name=tag_name)
30 links = the_tag.link_set.all()
31 return render_to_response('main/index.html', {'links': links, 'tag_name': '#' + tag_name}, context)
32
33 def add_link(request):
34 context = RequestContext(request)
35 if request.method == 'POST':
36 url = request.POST.get("url", "")
37 tags = request.POST.get("tags", "")
38 title = request.POST.get("title", "")
39 # TODO: Your code here!
40 return redirect(index)
```

# Templates & Static Files

---

## ▶ templates/main/index.html (1)

```
1 <!DOCTYPE html>
2 {% load static %}
3 <html>
4 <head>
5 <title>Bookmarks</title>
6 <link href='http://fonts.googleapis.com/css?family=Amarante' rel='stylesheet' type='text/css'>
7 <link rel="stylesheet" href='http://fonts.googleapis.com/css?family=Frijole' type="text/css" />
8 <link rel="stylesheet" href="{% static 'css/style.css' %}" media="all" />
9 <script src="{% static 'js/jquery-1.11.0.min.js' %}"></script>
10 <script src="{% static 'js/script.js' %}"></script>
11 </head>
```

# templates/main/index.html (2)

```
12 <body>
13 <div id="wrapper">
14 <header>
15 <h1>Bookmarks</h1>
16 </header>
17 <div id="main">
18 <div id="sidebar">
19
20 <li class="active" id="home">
21 My Links
22 {% if tag_name %}
23 {{ tag_name }}
24 {% endif %}
25
26 Add a link
27 Tags
28
29 </div><!--
30 ---><div id="content">
31 {% for link in links %}
32 <div class="link">
33 <h4>{{ link.title }}</h4>
34 <p class="tags">
35 {% for tag in link.tags.all %}
36 {{ tag.name }}
37 {% endfor %}
38 </p>
39 </div>
40 {% endfor %}
41 </div>
42 </div>
43 </div>
```

# templates/main/index.html (3)

```
44 <div class="popBackground"> </div>
45 <div class="popWindow">
46 <form id="addForm" method="POST" action="/bookmarks/add_link/">
47 {% csrf_token %}
48 <div class="winHead">
49 <h2>Add a link</h2>
50 </div>
51 <div class="winBody">
52 <p><label for="title">Title</label>
53 <input type="text" class="text" id="title" name="title"/></p>
54 <p><label for="link">Link</label>
55 <input type="text" class="text" id="link" name="url"/></p>
56 <p><label for="tags">Tags</label>
57 <input type="text" class="text" id="tags" name="tags"/></p>
58 </div>
59 <div class="winFoot">
60 Add
61 Cancel
62 </div>
63 </form>
64 </div>
65 </body>
66</html>
```

# Happy Coding!

---

# References

---

- ▶ Tango with Django:

<http://www.tangowithdjango.com/book/index.html>