

nsd1906_devweb_day05

运维web化项目

- ansible
- django
- bootstrap

项目规划

- <http://127.0.0.1/> : 项目首页，列出有哪些功能
- <http://127.0.0.1/webadmin/> : 管理首页，列出所有主机信息
- <http://127.0.0.1/webadmin/addhosts/> : 添加/显示主机和主机组
- <http://127.0.0.1/webadmin/addmodules/> : 添加/显示模块和参数
- <http://127.0.0.1/webadmin/tasks/> : 在指定的主机或组上执行管理任务
- 根据规划的url特点创建两个应用：index和webadmin

初始化项目

- 首先，通过pycharm创建名为myansible的项目
- 数据库用默认的sqlite3，它是一个文件型数据库，默认创建在项目的根目录下，名为db.sqlite3。mysql是数据库服务器软件，它可以包含很多个库，mysql还可以通过网络提供服务；sqlite是文件型的，一个文件就是一个库。

创建应用

```
(nsd1906) [root@room8pc16 myansible]# python manage.py startapp index
(nsd1906) [root@room8pc16 myansible]# python manage.py startapp webadmin
```

修改主配置文件

```
# myansible/settings.py
```

```
ALLOWED_HOSTS = ['*']
```

```
INSTALLED_APPS = [
```

```
    ...
```

```
    'index',
```

```
    'webadmin'
```

```
]
```

```
LANGUAGE_CODE = 'zh-hans'
```

```
TIME_ZONE = 'Asia/Shanghai'
```

```
USE_TZ = False
```

```
STATICFILES_DIRS = [ # 指定静态文件目录还有根目录的static
```

```
    os.path.join(BASE_DIR, "static"),
```

```
]
```

将前面课程中的static目录拷贝到项目目录下

```
(nsd1906) [root@room8pc16 myansible]# cp -r /var/ftp/nsd2019/nsd1906/devweb/day02/static/ .
```

```

# 授权，将应用的url交给应用处理
# myansible/urls.py
from django.conf.urls import url, include
from django.contrib import admin

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^webadmin/', include('webadmin.urls')),
    # r''可以匹配任意字符串
    url(r'', include('index.urls')),
]

# webadmin/urls.py
from django.conf.urls import url

urlpatterns = []

# index/urls.py
from django.conf.urls import url

urlpatterns = []

# 生成数据库表
(nsd1906) [root@room8pc16 myansible]# python manage.py makemigrations
(nsd1906) [root@room8pc16 myansible]# python manage.py migrate

# 创建管理员用户
(nsd1906) [root@room8pc16 myansible]# python manage.py createsuperuser

```

编写index应用

```

# 编写url
# index/urls.py
from django.conf.urls import url
from . import views

urlpatterns = [
    url(r'^$', views.index, name='index'),
]

# 编写函数
# index/views.py
from django.shortcuts import render

# Create your views here.
def index(request):
    return render(request, 'index.html')

# 实现模板继承
# 复制投票项目的basic.html到templates目录

```

```
(nsd1906) [root@room8pc16 myansible]# cp ../../day03/mysite/templates/basic.html templates/

# 编写模板
# templates/index.html
{% extends 'basic.html' %}
{% load static %}
{% block title %}Ansible Webadmin{% endblock %}
{% block content %}
<div class="row h4">
    <div class="col-sm-3 text-center">
        <a href="#">
            <br>
            主机信息
        </a>
    </div>
    <div class="col-sm-3 text-center">
        <a href="#">
            <br>
            添加主机
        </a>
    </div>
    <div class="col-sm-3 text-center">
        <a href="#">
            <br>
            添加模块
        </a>
    </div>
    <div class="col-sm-3 text-center">
        <a href="#">
            <br>
            执行任务
        </a>
    </div>
</div>
{% endblock %}
```

编写webadmin应用

```
# 创建模型, webadmin/models.py
from django.db import models

# Create your models here.
class HostGroup(models.Model):
    groupname = models.CharField(max_length=50, unique=True)

    def __str__(self):
        return self.groupname

class Host(models.Model):
    hostname = models.CharField(max_length=200)
    ipaddr = models.CharField(max_length=15)
    group = models.ForeignKey(HostGroup)
```

```

def __str__(self):
    return "%s=>%s:%s" % (self.group, self.hostname, self.ipaddr)

class Module(models.Model):
    modulename = models.CharField(max_length=50, unique=True)

    def __str__(self):
        return self.modulename

class Args(models.Model):
    arg_text = models.CharField(max_length=200)
    module = models.ForeignKey(Module)

    def __str__(self):
        return "%s=>%s" % (self.module, self.arg_text)

# 生成数据库表
(nsd1906) [root@room8pc16 myansible]# python manage.py makemigrations
(nsd1906) [root@room8pc16 myansible]# python manage.py migrate

# 查看数据库
(nsd1906) [root@room8pc16 myansible]# sqlite3 db.sqlite3
sqlite> .help    # 查看帮助
sqlite> .tables  # show tables;
sqlite> .schema webadmin_host  # desc webadmin_host
sqlite> select * from webadmin_host;

# 注册模型到web后台管理界面后，访问http://127.0.0.1/admin
# webadmin/admin.py
from django.contrib import admin
from .models import HostGroup, Host, Module, Args

# Register your models here.
for item in [HostGroup, Host, Module, Args]:
    admin.site.register(item)

```

配置ansible

```

(nsd1906) [root@room8pc16 myansible]# mkdir ansible_cfg
(nsd1906) [root@room8pc16 myansible]# cd ansible_cfg
(nsd1906) [root@room8pc16 ansible_cfg]# vim ansible.cfg
[defaults]
inventory = dhosts.py
remote_user = root
(nsd1906) [root@room8pc16 ansible_cfg]# touch dhosts.py
(nsd1906) [root@room8pc16 ansible_cfg]# chmod +x dhosts.py

# 动态主机清单要求，脚本执行的输出为以下样式：
{

```

```

    'webservers': {
        'hosts': ['192.168.4.5', '192.168.4.6']
    },
    '组2': {
        'hosts': [组2的主机列表]
    }
}

# 解题思路
>>> result = {} # 创建保存结果的字典
>>> if 'webservers' not in result:
...     result['webservers'] = {}
...
>>> result
{'webservers': {}}
# result['webservers']也是字典
>>> result['webservers']
{}
>>> result['webservers']['hosts'] = []
>>> result
{'webservers': {'hosts': []}}
# result['webservers']['hosts']是列表，可以追加数据
>>> result['webservers']['hosts'].append('192.168.4.5')
>>> result['webservers']['hosts'].append('192.168.4.6')
>>> result
{'webservers': {'hosts': ['192.168.4.5', '192.168.4.6']}}

```

```

(nsd1906) [root@room8pc16 ansible_cfg]# vim dhosts.py
#!/root/nsd1906/bin/python

```

```

from sqlalchemy import create_engine, Column, Integer, String, ForeignKey
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker

engine = create_engine(
    'sqlite:///var/ftp/nsd2019/nsd1906/devweb/ansible_project/myansible/db.sqlite3',
    encoding='utf8',
)
Base = declarative_base()
Session = sessionmaker(bind=engine)

class HostGroup(Base):
    __tablename__ = 'webadmin_hostgroup'
    id = Column(Integer, primary_key=True)
    groupname = Column(String(20), unique=True)

class Host(Base):
    __tablename__ = 'webadmin_host'
    id = Column(Integer, primary_key=True)
    hostname = Column(String(200))
    ipaddr = Column(String(15))
    group_id = Column(Integer, ForeignKey('webadmin_hostgroup.id'))

```

```

if __name__ == '__main__':
    session = Session()
    qset = session.query(HostGroup.groupname, Host.ipaddr).join(Host)
    # print(qset.all())
    result = {}
    for g, ip in qset:
        if g not in result:
            result[g] = {}
            result[g]['hosts'] = []
            result[g]['hosts'].append(ip)

    print(result)

```

配置主机信息页

```

# 生成主机信息页
(nsd1906) [root@room8pc16 ansible_cfg]# ansible all -m setup --tree /tmp/webadmin/
(nsd1906) [root@room8pc16 ansible_cfg]# ansible-cmdb /tmp/webadmin/ >
../templates/webadmin.html

# 编写url, webadmin/urls.py
from django.conf.urls import url
from . import views

urlpatterns = [
    url(r'^$', views.index, name='webadmin'),
]

# 编写函数, webadmin/views.py
from django.shortcuts import render

# Create your views here.
def index(request):
    return render(request, 'webadmin.html')

# 修改templates/index.html, 为主机信息加超链接
<a href="{% url 'webadmin' %}" target="_blank">
    <br>
    主机信息
</a>

```

完成添加主机页

```

# webadmin/urls.py
url(r'^addhosts/$', views.add_hosts, name='add_hosts'),

# webadmin/views.py
from django.shortcuts import render
from .models import HostGroup

```

```

# Create your views here.
def index(request):
    return render(request, 'webadmin.html')

def add_hosts(request):
    groups = HostGroup.objects.all()
    return render(request, 'add_hosts.html', {'groups': groups})

# templates/add_hosts.html
{% extends 'basic.html' %}
{% load static %}
{% block title %}添加主机{% endblock %}
{% block content %}
    {% comment %}action为空，表示提交给自己{% endcomment %}
    <form action="" method="post" class="form-inline h4">
        {% csrf_token %}
        <div class="form-group">
            <label>主机组 :</label>
            <input class="form-control" type="text" name="group">
        </div>
        <div class="form-group">
            <label>主机 :</label>
            <input class="form-control" type="text" name="host">
        </div>
        <div class="form-group">
            <label>IP :</label>
            <input class="form-control" type="text" name="ip">
        </div>
        <div class="form-group">
            <input class="btn btn-primary" type="submit" value="提 交">
        </div>
    </form>
    <hr>
    <table class="table table-hover table-striped table-bordered h4">
        <thead class="bg-primary">
            <tr>
                <th>主机组</th>
                <th>主机</th>
            </tr>
        </thead>
        {% for group in groups %}
            <tr>
                <td>{{ group.groupname }}</td>
                <td>
                    <ul class="list-unstyled">
                        {% for host in group.host_set.all %}
                            <li>
                                {{ host.hostname }}: {{ host.ipaddr }}
                            </li>
                        {% endfor %}
                    </ul>
                </td>
            </tr>
        {% endfor %}
    </table>

```

```

        </tr>
        {% endfor %}
    </table>
{% endblock %}

# templates/index.html
<a href="{% url 'add_hosts' %}">
    <br>
    添加主机
</a>

# 完善add_hosts函数
def add_hosts(request):
    # 如果是表单的post方法，则取出相关的参数，创建主机和组
    if request.method == 'POST':
        group = request.POST.get('group').strip()
        host = request.POST.get('host').strip()
        ip = request.POST.get('ip').strip()
        if group: # 如果group字符串非空
            # get_or_create返回的是元组：(组实例, True/False)
            g = HostGroup.objects.get_or_create(groupname=group)[0]
            if host and ip: # 如果host和ip都非空
                g.host_set.get_or_create(hostname=host, ipaddr=ip)

    groups = HostGroup.objects.all()
    return render(request, 'add_hosts.html', {'groups': groups})

```