nsd1904_devweb_ansible_project1

项目展示

运行nsd1903班ansible_pro项目

```
$ source ~/nsd1904/bin/activate
$ cd nsd2019/nsd1903/devweb/ansible_pro/myansible
$ python manage.py runserver
访问127.0.0.1:8000
```

功能说明

- http://127.0.0.1:8000/：首页，列出所有功能项
- http://127.0.0.1:8000/webadmin/：展示托管主机详细信息
- http://127.0.0.1:8000/webadmin/addhosts/：实现添加主机/组功能
- http://127.0.0.1:8000/webadmin/addmodules/：实现添加模块/参数功能
- http://127.0.0.1:8000/webadmin/tasks/：在指定的主机或组上执行管理任务

编写项目

项目初始化

1. 通过pycharm创建名为myansible的项目
2. 创建名为index和webadmin的应用

```
(nsd1904) [root@room8pc16 myansible]# python manage.py startapp index
(nsd1904) [root@room8pc16 myansible]# python manage.py startapp webadmin
```

3. 修改配置文件

```
# myansible/settings.py
ALLOWED_HOSTS = ['*']
INSTALLED_APPS = [
    ... ...
    'index',
    'webadmin',
]
LANGUAGE_CODE = 'zh-hans'
TIME_ZONE = 'Asia/Shanghai'
USE_TZ = False
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "static"),
]
```

4. 授权

```
# 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''，可以匹配任何url，务必放到最后
    url(r'', include('index.urls')),
]
# index/urls.py,  # webadmin/urls.py
from django.conf.urls import url

urlpatterns = [
]
```

配置**index**应用

1. 配置URL

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

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

2. 编写index函数

```
# index/views.py
from django.shortcuts import render

def index(request):
    return render(request, 'index.html')
```

3. 创建模板文件

```
# templates/index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Ansible Webadmin</title>
</head>
<body>
Ansible Webadmin
</body>
</html>
```

4. 引入bootstrap：拷贝前一个项目的static目录到项目的根目录下

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Ansible Webadmin</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">
</head>
<body>
Ansible Webadmin
</body>
</html>
```

5. 实现模板继承

```
# 拷贝前一个项目的base.html到templates目录
# 修改templates/index.html
{% extends 'base.html' %}
{% load static %}
{% block title %}Ansible Webadmin{% endblock %}
{% block content %}
<div class="row text-center h4" style="margin-bottom: 50px">
    <div class="col-sm-3">
        <a href="#">
            <img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>
            主机信息
        </a>
    </div>
    <div class="col-sm-3">
        <a href="#">
            <img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>
            添加主机
        </a>
    </div>
    <div class="col-sm-3">
        <a href="#">
            <img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>
            添加模块
        </a>
    </div>
    <div class="col-sm-3">
        <a href="#">
            <img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>
            执行任务
        </a>
    </div>
</div>
{% endblock %}
```

编写webadmin应用

1. 编写模型Model

```
# webadmin/models.py
from django.db import models

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=50)
    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 Argument(models.Model):
    arg_text = models.CharField(max_length=200)
    module = models.ForeignKey(Module)

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

2. 生成表

```
(nsd1904) [root@room8pc16 myansible]# python manage.py makemigrations
(nsd1904) [root@room8pc16 myansible]# python manage.py migrate
```

3. 查看表结构

```
# sqlite数据库是文件型数据库，一个文件就是一个库
(nsd1904) [root@room8pc16 myansible]# sqlite3 db.sqlite3
sqlite> .help    # 查看帮助
sqlite> .tables   # 相当于show tables
sqlite> .schema webadmin_host   # 相当于desc webadmin_host
sqlite> SELECT * from auth_user;   # sql语句
```

4. 创建管理员

```
(nsd1904) [root@room8pc16 myansible]# python manage.py createsuperuser
```

5. 将模型注册到后台

```
# webadmin/admin.py
from django.contrib import admin
from .models import HostGroup, Host, Module, Argument

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

6. 登陆后台，添加主机、组。http://127.0.0.1/admin/

配置ansible环境

1. 创建配置文件

```
(nsd1904) [root@room8pc16 myansible]# mkdir ansi_cfg
(nsd1904) [root@room8pc16 myansible]# vim ansi_cfg/ansible.cfg
[defaults]
inventory = dhosts.py
remote_user = root
```

2. 实现动态主机清单

动态主机清单生成的结果样式要求如下：

```
{
    "组1": {
        "hosts": ["主机1", "主机2"]
    },
    "组2": {
        "hosts": ["主机3", "主机4"],
    },
}
```

```
# touch dhosts.py
# chmod +x dhosts.py
# vim dhosts.py
#!/root/nsd1904/bin/python
from sqlalchemy import create_engine, Column, Integer, String, ForeignKey
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker
import json

engin = create_engine(
    'sqlite:////var/ftp/nsd2019/nsd1904/devweb/ansible_pro/myansible/db.sqlite3',
    encoding='utf8',
)
Base = declarative_base()
Session = sessionmaker(bind=engin)

class HostGroup(Base):
    __tablename__ = 'webadmin_hostgroup'
    id = Column(Integer, primary_key=True)
    groupname = Column(String(50), unique=True)

class Host(Base):
    __tablename__ = 'webadmin_host'
    id = Column(Integer, primary_key=True)
    hostname = Column(String(50))
    ipaddr = Column(String(15))
    group_id = Column(ForeignKey('webadmin_hostgroup.id'))

if __name__ == '__main__':
    session = Session()
    result = {}
    qset = session.query(HostGroup.groupname, Host.ipaddr).join(Host)
    # print(qset.all())
    for group, ip in qset:
        if group not in result:
            result[group] = {}  # {'dbservers': {}}
            result[group]['hosts'] = []  # {'dbservers': {'hosts': []}}
        result[group]['hosts'].append(ip)
    print(json.dumps(result))   # 输出要求是json格式，而不是字典
```

3. 测试

```
# ansible all --list-hosts
```

4. 远程主机需要配置免密登陆

5. 收集远程主机信息

```
[root@room8pc16 ansi_cfg]# ansible all -m setup --tree /tmp/out
```

6. 生成网页文件

```
(nsd1904) [root@room8pc16 ansi_cfg]# ansible-cmdb /tmp/out/ > ../templates/polls_index.html
```

实现**webadmin**首页，即主机信息页

1. url

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

urlpatterns = [
    url(r'', views.index, name='polls_index'),
]
```

2. 视图函数

```
# webadmin/views.py
from django.shortcuts import render

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

3. 修改templates/index.html中“主机信息”的链接

```
<div class="col-sm-3">
    <a href="{% url 'polls_index' %}" target="_blank">
        <img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>
        主机信息
    </a>
</div>
```

4. 在首页上点击“主机信息”进行测试

实现添加主机页面

1. url

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

2. 视图函数

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

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

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

3. 模板文件

```
# templates/add_hosts.html
{% extends 'base.html' %}
{% load static %}
{% block title %}添加主机{% endblock %}
{% block content %}
{{ groups }}
{% endblock %}
```

4. 修改templates/index.html中“添加主机”的超链接

```
        <a href="{% url 'add_hosts' %}" target="_blank">
```

5. 测试

6. 修改模板文件

```
{% extends 'base.html' %}
{% load static %}
{% block title %}添加主机{% endblock %}
{% block content %}
    <div class="row h4">
        <div class="col-sm-12">
            <form action="" class="form-inline" method="post">
                {% 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地址：</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>
        </div>
    </div>
    <hr>
    <table class="table table-bordered table-hover table-striped h4">
        <thead>
            <tr class="bg-primary text-center">
                <td>主机组</td>
                <td>主机</td>
            </tr>
        </thead>
        {% for group in groups %}
            <tr>
                <td>{{ group }}</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>
{% endblock %}
```

7. 修改视图函数，完成添加主机功能

```
# webadmin/views.py
def add_hosts(request):
    if request.method == 'POST':
        groupname = request.POST.get('group').strip()
        hostname = request.POST.get('host').strip()
        ip = request.POST.get('ip').strip()
        if groupname:  # 如果组字符串非空
            # get_or_create是取出或创建组，返回值是: (组实例，状态)
            group = HostGroup.objects.get_or_create(groupname=groupname)[0]
            if hostname and ip:  # 如果主机名和ip非空
                group.host_set.get_or_create(hostname=hostname, ipaddr=ip)

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