ansible-project1

运行1910班的项目

```
[root@localhost ~]# cp -r nsd2019/nsd1910/devweb/ansible_project/myansible/ /tmp
[root@localhost ~]# cd /tmp/myansible/
[root@localhost myansible]# python3 manage.py runserver
# 访问http://127.0.0.1:8000
```

ansible_project

创建项目

- 通过pycharm创建一个名为myansible的项目

```
# myansible/settings.py
ALLOWED_HOSTS = ['*']
LANGUAGE_CODE = 'zh-hans'
TIME_ZONE = 'Asia/Shanghai'
USE_TZ = False
# 运行测试
[root@localhost myansible]# python3 manage.py runserver 0:80
[root@localhost myansible]# ls
db.sqlite3  manage.py  myansible  templates
# 访问http://127.0.0.1/

# 创建名为index和webadmin的应用
[root@localhost myansible]# python3 manage.py startapp index
[root@localhost myansible]# python3 manage.py startapp webadmin
# 集成应用到项目
# myansible/settings.py
INSTALLED_APPS = [
    ... ...
    'index',
    'webadmin',
]
# 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，必须放到列表最下面
    url(r'', include('index.urls')),
]
# 创建index/urls.py和webadmin/urls.py
from django.conf.urls import url

urlpatterns = []
```

制作首页应用

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

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

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

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

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

- 引入boostrap。为了使得两个应用都可以使用相同的boostrap，将boostrap放到项目目录下

```
# 将static目录拷贝到项目目录下
[root@db1 myansible]# cp -r  ../../day0304/mysite/polls/static/ .

# myansible/settings.py
... ...
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "static"),
]

# 将投票应用的basic.html拷贝过来，作为基础模板
[root@db1 myansible]# cp -r  ../../day0304/mysite/templates/basic.html templates/
# 修改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="#" target="_blank">
                <img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>
                主机信息
            </a>
        </div>
        <div class="col-sm-3 text-center">
            <a href="#" target="_blank">
                <img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>
                添加主机
            </a>
        </div>
        <div class="col-sm-3 text-center">
            <a href="#" target="_blank">
                <img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>
                添加模块
            </a>
        </div>
        <div class="col-sm-3 text-center">
            <a href="#" target="_blank">
                <img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>
                执行任务
            </a>
        </div>
    </div>
{% endblock %}
```

制作webadmin应用

配置模型

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

class HostGroup(models.Model):
    groupname = models.CharField(max_length=50)

    def __str__(self):
        return self.groupname

class Host(models.Model):
    hostname = models.CharField(max_length=50)
    ip_addr = models.CharField(max_length=15)
    group = models.ForeignKey(HostGroup)

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

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

    def __str__(self):
        return self.modulename

class Argument(models.Model):
    arg_text = models.CharField(max_length=100)
    module = models.ForeignKey(Module)

    def __str__(self):
        return "%s:[%s]" % (self.module, self.arg_text)

# 生成表
[root@db1 myansible]# python3 manage.py makemigrations
[root@db1 myansible]# python3 manage.py migrate
# 创建管理员用户
[root@db1 myansible]# python3 manage.py createsuperuser
Username (leave blank to use 'root'): admin
Email address: admin@tedu.cn
Password:
Password (again):
Superuser created successfully.

# 将模型注册到管理后台
#  webadmin/admin.py
from django.contrib import admin
from webadmin.models import HostGroup, Host, Module, Argument

for item in [HostGroup, Host, Module, Argument]:
    admin.site.register(item)

# 访问http://x.x.x.x/admin/，添加主机和组
```

配置ansible

- sqlite数据库：文件型数据库， 一个文件就是一个库

```
[root@db1 myansible]# sqlite3 db.sqlite3
sqlite> .help    # 查看帮助
sqlite> .table   # show tables
sqlite> .schema webadmin_hostgroup  # desc webadmin_hostgroup
sqlite> .schema webadmin_host
sqlite> select * from auth_user;
sqlite> .quit
# 配置ansible的工作环境
[root@db1 myansible]# mkdir ansi_cfg
[root@db1 myansible]# cd ansi_cfg/
[root@db1 ansi_cfg]# vim ansible.cfg
[defaults]
inventory = dhosts.py
remote_user = root
[root@db1 ansi_cfg]# touch dhosts.py
[root@db1 ansi_cfg]# chmod +x dhosts.py
# dhosts.py执行的结果至少是以下形式：
{
    'dbservers': {
        'hosts': ['1.1.1.1', '1.1.1.2']
    },
    'webservers': {
        'hosts': ['1.1.1.3', '1.1.1.4', '1.1.1.5']
    },
}
##################################
>>> result = {}
>>> g = 'webservers'
>>> result[g] = {}
>>> result
{'webservers': {}}
>>> result[g]
{}
>>> result[g]['hosts'] = []
>>> result
{'webservers': {'hosts': []}}
>>> result[g]['hosts']
[]
>>> result[g]['hosts'].append('1.1.1.1')
>>> result
{'webservers': {'hosts': ['1.1.1.1']}}
########################################


[root@db1 ansi_cfg]# vim dhosts.py
#!/usr/local/bin/python3
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:////root/nsd2019/nsd1910/devweb/ansible_project/myansible/db.sqlite3',
    encoding='utf8',
)
Session = sessionmaker(bind=engine)

Base = declarative_base()
```

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

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

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

# 测试ansible动态主机清单
[root@db1 ansi_cfg]# ansible all -m ping
```

制作主机信息页

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

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

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

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

# 生成主机信息页
[root@db1 ansi_cfg]# ansible all -m setup --tree /tmp/servers
[root@db1 ansi_cfg]# ansible-cmdb /tmp/servers > ../templates/hosts.html

# 修改templates/index.html中主机信息的超链接
            <a href="{% url 'webadmin_index' %}" target="_blank">
```

制作添加主机页

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

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

def index(request):
    return render(request, 'hosts.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=""表示按提交按钮时，提交给当前 url{% endcomment %}
    <form action="" class="form-inline h4" 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>
    <hr>
    <table class="table table-bordered table-hover table-striped h4">
        <thead class="bg-primary">
            <tr>
                <td>主机组</td>
                <td>主机</td>
            </tr>
        </thead>
        <tbody>
            {% for group in groups %}
                <tr>
                    <td>{{ group.groupname }}</td>
                    <td>
                        {% for host in group.host_set.all %}
                            <div>

                                {{ host.hostname }}:{{ host.ip_addr }}
                            </div>
                        {% endfor %}
                    </td>
                </tr>
            {% endfor %}
        </tbody>
    </table>
{% endblock %}

# 在主页中修改添加主机的超链接
# templates/index.html
            <a href="{% url 'add_hosts' %}" target="_blank">
```