

ansible_django_project2

制作webadmin应用

url规划

```
http://127.0.0.1/webadmin : 显示主机信息  
http://127.0.0.1/webadmin/add_hosts : 添加主机  
http://127.0.0.1/webadmin/add_modules : 添加模块  
http://127.0.0.1/webadmin/tasks : 执行ansible任务
```

编写模型

```

# 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=100)
    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):
    modlurname = models.CharField(max_length=50)

    def __str__(self):
        return self.modlurname

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)

(nsd1908) [root@localhost myansible]# python manage.py makemigrations
(nsd1908) [root@localhost myansible]# python manage.py migrate

# 将模型注册到后台管理界面
# webadmin/admin.py
from django.contrib import admin
from .models import HostGroup, Host, Module, Argument

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

# 登陆到http://127.0.0.1/admin/ , 添加两个组和一些主机

```

配置ansible

```
(nsd1908) [root@localhost myansible]# mkdir ansi_cfg
(nsd1908) [root@localhost myansible]# cd ansi_cfg
(nsd1908) [root@localhost ansi_cfg]# vim ansible.cfg
[defaults]
inventory = ./dhosts.py
remote_user = root
# 实现到远程主机的免密登陆
(nsd1908) [root@localhost ansi_cfg]# ssh-keygen
(nsd1908) [root@localhost ansi_cfg]# ssh-copy-id 127.0.0.1

(nsd1908) [root@localhost ansi_cfg]# touch dhosts.py
(nsd1908) [root@localhost ansi_cfg]# chmod 755 dhosts.py
```

创建动态主机清单文件

动态主机清单文件就是一个脚本，它必须输出以下格式：

```
{
    'dbservers': {'hosts': ['主机1', '主机2']},
    'webservers': {'hosts': ['主机3', '主机4']},
}

# vim dhosts.py
#!/root/nsd1908/bin/python
from sqlalchemy import create_engine, Column, Integer, String, ForeignKey, Date
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker

engine = create_engine(
    'sqlite:///var/ftp/nsd2019/nsd1908/devweb/ansible_pro/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), unique=True)

class Host(Base):
    __tablename__ = 'webadmin_host'
    id = Column(Integer, primary_key=True)
    hostname = Column(String(100))
    ipaddr = Column(String(15))
    group_id = Column(ForeignKey('webadmin_hostgroup.id'))

if __name__ == '__main__':
    session = Session()
    qset = session.query(Host.ipaddr, HostGroup.groupname).join(HostGroup)
    # print(qset.all())
    result = {}
    for ip, group in qset:
        if group not in result:
            result[group] = {}
            result[group]['hosts'] = []
        result[group]['hosts'].append(ip)
    print(result)
(nsd1908) [root@localhost ansi_cfg]# ./dhosts.py
{'dbservers': {'hosts': ['127.0.0.1']}, 'webservers': {'hosts': ['127.0.0.1',
'127.0.0.1']}}
(nsd1908) [root@localhost ansi_cfg]# ansible all -m ping
```

制作webadmin首页

```
# 通过ansible-cmdb生成信息
(nsd1908) [root@localhost ansi_cfg]# ansible all -m setup --tree /tmp/nsd1908out/
(nsd1908) [root@localhost ansi_cfg]# ansible-cmdb /tmp/nsd1908out/ >
../templates/hosts.html

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

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

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

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

# templates/index.html
<a href="{% url 'mainpage' %}" target="_blank">
    <br>
    主机信息
</a>
```

制作添加主机页

```

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

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

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

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

def add_hosts(request):
    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非空
            g = HostGroup.objects.get_or_create(groupname=group)[0]
            if host and ip: # 如果host和ip非空
                g.host_set.get_or_create(hostname=host, ipaddr=ip)

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

# templates/add_hosts.html
{% extends 'base.html' %}
{% load static %}
{% block title %}添加主机{% endblock %}
{% block content %}
# action=""表示提交表单时, 提交到当前 url, 即http://127.0.0.1/webadmin/add_hosts/
<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>
    <input class="btn btn-primary" type="submit" value="提 交">
</form>
<hr>
<table class="table table-bordered table-striped table-hover h4">
    <thead>

    <tr class="bg-primary">

```

```

        <td>主机组</td>
        <td>主机</td>
    </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>
{% endblock %}

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

```

制作添加模块页

```

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

# webadmin/views.py
from .models import HostGroup, Module

def add_modules(request):
    if request.method == 'POST':
        module = request.POST.get('module').strip()
        arg = request.POST.get('param').strip()
        if module:
            m = Module.objects.get_or_create(modlurname=module)[0]
            m.argument_set.get_or_create(arg_text=arg)

    modules = Module.objects.all()
    return render(request, 'add_modules.html', {'modules': modules})

# templates/add_modules.html
{% extends 'base.html' %}
{% load static %}
{% block title %}添加模块{% endblock %}
{% block content %}
    <form action="" method="post" class="form-inline h4">
        {% csrf_token %}
        <div class="form-group">
            <label>模块:</label>
            <input class="form-control" type="text" name="module">
        </div>
        <div class="form-group">
            <label>参数:</label>
            <input class="form-control" type="text" name="param">
        </div>
        <input class="btn btn-primary" type="submit" value="提交">
    </form>
    <hr>
    <table class="table table-bordered table-striped table-hover h4">
        <thead>
            <tr class="bg-primary">
                <td>模块</td>
                <td>参数</td>
            </tr>
        </thead>
        {% for module in modules %}
            <tr>
                <td>{{ module.modlurname }}</td>
                <td>
                    <ul class="list-unstyled">
                        {% for arg in module.argument_set.all %}
                            <li>{{ arg.arg_text }}</li>
                        {% endfor %}
                    </ul>
                </td>
            </tr>
        {% endfor %}
    </table>

```



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

# templates/index.html
<a href="{% url 'add_modules' %}" target="_blank">
    <br>
    添加模块
</a>
```

制作执行任务页

```

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

# webadmin/views.py
from .models import HostGroup, Module, Host
from .adhoc2 import adhoc # 把devops课程中的adhoc2当作模块导入

def tasks(request):
    if request.method == 'POST':
        ip = request.POST.get('ip')
        group = request.POST.get('hostgroup')
        module = request.POST.get('mod')
        arg = request.POST.get('param')
        # 决定在主机还是组上执行任务，优先选组
        if group:
            dest = group
        elif ip:
            dest = ip
        else:
            dest = None

        if dest: # 如果dest不是None
            adhoc(['ansi_cfg/dhosts.py'], dest, module, arg)

    hosts = Host.objects.all()
    groups = HostGroup.objects.all()
    modules = Module.objects.all()
    context = {'hosts': hosts, 'groups': groups, 'modules': modules}
    return render(request, 'tasks.html', context)

# templates/tasks.html
{% extends 'base.html' %}
{% load static %}
{% block title %}执行任务{% endblock %}
{% block content %}
<div class="h4">
    <ul class="nav nav-tabs">
        <li class="active"><a href="#server" data-toggle="tab">主机</a></li>
        <li><a href="#servergroup" data-toggle="tab">主机组</a></li>
    </ul>
    <form action="" method="post">
        {% csrf_token %}
        <div class="tab-content">
            <div class="tab-pane active fade in form-group" id="server">
                <select class="form-control" name="ip">
                    <option value="">无</option>
                    {% for host in hosts %}
                        <option value="{{ host.ipaddr }}">{{ host.hostname }}
</option>
                    {% endfor %}
                </select>
            </div>
            <div class="tab-pane fade form-group" id="servergroup">
                <select class="form-control" name="hostgroup">

```

```

        <option value="">无</option>
        {% for group in groups %}
            <option value="{{ group.groupname }}">{{ group.groupname }}
</option>

        {% endfor %}
    </select>
</div>
</div>
<table class="table table-bordered table-striped table-hover h4">
    <thead>
        <tr class="bg-primary">
            <td>模块</td>
            <td>参数</td>
        </tr>
    </thead>
    {% for module in modules %}
        <tr>
            <td>
                <label>
                    <input type="radio" name="mod" value="{{
module.modlurname }}">
                    {{ module.modlurname }}
                </label>
            </td>
            <td>
                <ul class="list-unstyled">
                    {% for arg in module.argument_set.all %}
                        <li>
                            <label>
                                <input type="radio" name="param" value="{{
arg.arg_text }}">
                                {{ arg.arg_text }}
                            </label>
                        </li>
                    {% endfor %}
                </ul>
            </td>
        </tr>
    {% endfor %}
</table>
<div class="form-group text-center">
    <input class="btn btn-primary" type="submit" value="执 行">
</div>
</form>
</div>
{% endblock %}

# templates/index.html
<a href="{% url 'tasks' %}" target="_blank">
    <br>
    执行任务
</a>

```

为参数增加删除功能

删除参数，需要执行一个函数。函数通过访问url来调用。

```
# webadmin/urls.py
... ..
url(r'^del_arg/(\d+)/$', views.del_arg, name='del_arg'),
... ..

# webadmin/views.py
from django.shortcuts import render, redirect
from .models import HostGroup, Module, Host, Argument

def del_arg(request, arg_id):
    arg = Argument.objects.get(id=arg_id)
    arg.delete()
    return redirect('add_modules')

# templates/add_modules.html
{% for module in modules %}
    <tr>
        <td>{{ module.modlurname }}</td>
        <td>
            <ul class="list-unstyled">
                {% for arg in module.argument_set.all %}
                    <li>
                        {{ arg.arg_text }}
                        <a href="{% url 'del_arg' arg.id %}">del</a>
                    </li>
                {% endfor %}
            </ul>
        </td>
    </tr>
{% endfor %}
```