

nsd1904_devweb_ansible_project2

添加模块功能

1. url

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

2. 视图函数

```
# webadmin/views.py
def add_modules(request):
    if request.method == 'POST':
        mod_name = request.POST.get('mod').strip()
        arg = request.POST.get('param').strip()
        if mod_name:
            module = Module.objects.get_or_create(modulename=mod_name)[0]
            if arg:
                module.argument_set.get_or_create(arg_text=arg)

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

3. 模板

```
{% extends 'base.html' %}
{% load static %}
{% block title %}添加模块{% 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>模块 : </label>
                    <input class="form-control" type="text" name="mod">
                </div>
                <div class="form-group">
                    <label>参数 : </label>
                    <input class="form-control" type="text" name="param">
                </div>
                <div class="form-group">
                    <input class="btn btn-primary" type="submit" value="提 交">
                </div>
            </form>
        </div>
    </div>
```

```

</div>
<hr>
<table class="table table-bordered table-hover table-striped h4">
  <thead>
    <tr class="bg-primary text-center">
      <td>模块</td>
      <td>参数</td>
    </tr>
  </thead>
  {% for module in modules %}
    <tr>
      <td>{{ module.modulename }}</td>
      <td>
        <ul class="list-unstyled">
          {% for arg in module.argument_set.all %}
            <li>
              {{ arg.arg_text }}
            </li>
          {% endfor %}
        </ul>
      </td>
    </tr>
  {% endfor %}
</table>
{% endblock %}

```

4. 修改templates/index.html中“添加模块”的超链接

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

5. 测试

编写“执行任务”页面

1. url

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

2. 视图函数

```
# webadmin/views.py
def tasks(request):
    groups = HostGroup.objects.all()
    hosts = Host.objects.all()
    modules = Module.objects.all()
    context = {'hosts': hosts, 'groups': groups, 'modules': modules}
    return render(request, 'tasks.html', context)
```

3. 模板

```
# templates/tasks.html
{% extends 'base.html' %}
{% load static %}
{% block title %}执行任务{% endblock %}
{% block content %}
    {{ groups }}
    {{ hosts }}
    {{ modules }}
{% endblock %}
```

4. 修改templates/index.html中“执行任务”的超链接

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

5. 完善模板

```
# templates/tasks.html
{% extends 'base.html' %}
{% load static %}
{% block title %}执行任务{% endblock %}
{% block content %}
    <ul class="nav nav-tabs h4">
        <li class="active"><a href="#server" data-toggle="tab">主机</a></li>
        <li><a href="#server-group" data-toggle="tab">主机组</a></li>
    </ul>
    <form action="" method="post">
        {% csrf_token %}
        <div class="tab-content" style="margin-top: 10px">
            <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="server-group">
                <select class="form-control" name="group">
                    <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-hover table-striped h4">
            <thead>
                <tr class="bg-primary text-center">
                    <td>模块</td>
                    <td>参数</td>
                </tr>
```

```

</thead>
{% for module in modules %}
    <tr>
        <td>
            <div class="radio">
                <label>
                    <input type="radio" name="mod" value="{{ module.modulename
}}">
                    {{ module.modulename }}
                </label>
            </div>
        </td>
        <td>
            <ul class="list-unstyled">
                {% for param in module.argument_set.all %}
                    <li>
                        <div class="radio">
                            <label>
                                <input type="radio" name="args" value="{{
param.arg_text }}">
                                {{ param.arg_text }}
                            </label>
                        </div>
                    </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>
{% endblock %}

```

6. 完善tasks函数，使之可以执行任务

```

# 将ansible课程部分的adhoc2.py拷贝到webadmin目录
# cp ../../../../devops/day03/adhoc2.py webadmin/
# webadmin/tasks.py
from .adhoc2 import adhoc

def tasks(request):
    if request.method == 'POST':
        ip = request.POST.get('ip')
        group = request.POST.get('group')
        module = request.POST.get('mod')
        args = request.POST.get('args')
        if ip: # 如果ip非空，执行目标设定为主机
            dest = ip
        elif group: # 如果ip为空，group不为空，执行目标设定为组
            dest = group

```

```

else:
    dest = None

if dest: # 如果执行目标非空,才执行任务
    adhoc(sources=['ansi_cfg/dhosts.py'], hosts=dest, module=module, args=args)

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

```

添加删除参数功能

1. 在“添加模块”的页面中，在参数后面加上del超链接
2. 点击超链接，就会访问一个url
3. url与函数相关联
4. 调用函数，作用是删除参数
5. 函数最后重定向到当前URL

```

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

# templates/add_modules.html, 在参数后面加上del的超链接
{% extends 'base.html' %}
{% load static %}
{% block title %}添加模块{% endblock %}
{% block content %}


<form action="" class="form-inline" method="post">
    {% csrf_token %}
    <div class="form-group">
        <label>模块 : </label>
        <input class="form-control" type="text" name="mod">
    </div>
    <div class="form-group">
        <label>参数 : </label>
        <input class="form-control" type="text" name="param">
    </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>模块</td>
            <td>参数</td>


```

```

        </tr>
    </thead>
    {% for module in modules %}
        <tr>
            <td>{{ module.modulename }}</td>
            <td>
                <ul class="list-unstyled">
                    {% for arg in module.argument_set.all %}
                        <li>
                            <div class="col-sm-9">
                                {{ arg.arg_text }}
                            </div>
                            <div class="col-sm-3">
                                <a href="{% url 'del_arg' arg.id %}">
                                    del
                                </a>
                            </div>
                        </li>
                    {% endfor %}
                </ul>
            </td>
        </tr>
    {% endfor %}
</table>
{% endblock %}

```

```

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

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

```