

nsd1906_devweb_day06

制作添加模块页

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

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

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

# templates/add_modules.html
{% extends 'basic.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>
        <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>模块</th>
                <th>参数</th>
            </tr>
        </thead>
        {% for module in modules %}
            <tr>
                <td>{{ module.modulename }}</td>
                <td>
                    <ul class="list-unstyled">
                        {% for arg in module.args_set.all %}
                            <li>
```

```

                {{ arg.arg_text }}
            </li>
        {% endfor %}
    </ul>
</td>
</tr>
{% endfor %}
</table>
{% endblock %}

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

# 完成添加模块的功能
# webadmin/views.py
def add_modules(request):
    if request.method == 'POST':
        module = request.POST.get('module').strip()
        arg = request.POST.get('param').strip()
        if module: # 如果模块名非空
            m = Module.objects.get_or_create(modulename=module)[0]
            if arg: # 如果参数也是非空的
                m.args_set.get_or_create(arg_text=arg)

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

```

实现执行任务页面

```

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

# webadmin/views.py
from .models import HostGroup, Module, Host
def tasks(request):
    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 'basic.html' %}
{% load static %}
{% block title %}执行任务{% endblock %}
{% block content %}
    <ul class="nav nav-tabs h4">

```



```

                <div class="radio">
                    <label>
                        <input type="radio" name="args" value="{{
arg.arg_text }}">
                            {{ arg.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 %}

```

```

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

```

```

# 实现调用ansible的adhoc模式，执行任务
# 将ansible的adhoc.py拷贝过来，作为一个模块导入
(nsd1906) [root@room8pc16 myansible]# cp ../../../../devops/day03/adhoc2.py webadmin/

```

```

# 修改函数 webadmin/views.py
from .adhoc2 import adhoc

```

```

def tasks(request):
    if request.method == 'POST':
        group = request.POST.get('group')
        ip = request.POST.get('ip')
        module = request.POST.get('module')
        args = request.POST.get('args')
        dest = None # dest是ansible执行任务的目标
        # 如果group和ip都是非空的，那么在组上执行任务
        if group:
            dest = group
        elif ip:
            dest = ip

        if dest: # 如果dest非空则扫行任务
            if module and args: # 如果module和args也是非空的
                adhoc('ansible_cfg/dhosts.py', dest, module, args)

    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)
```

实现删除参数功能

思路：

- 删除参数，需要调用函数
- 可以通过url来调用函数

```
# 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, Args

def del_arg(request, arg_id):
    arg = Args.objects.get(id=arg_id)
    arg.delete()

    return redirect('add_modules')

# templates/add_modules.html
<table class="table table-hover table-striped table-bordered h4">
  <thead class="bg-primary">
    <tr>
      <th>模块</th>
      <th>参数</th>
    </tr>
  </thead>
  {% for module in modules %}
    <tr>
      <td>{{ module.modulename }}</td>
      <td>
        <ul class="list-unstyled">
          {% for arg in module.args_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>
```