

nsd1812_devweb_day04

api

在python shell中运行指令

```
(djenv) [root@room8pc16 mysite]# python manage.py shell
```

导入模型

```
>>> from polls.models import Question, Choice
```

创建问题

- 实例化

```
>>> q1 = Question(question_text='你期待哪家公司给你发Offer?', pub_date='2019-05-27 9:00:00')
>>> q1.save()
```

- 使用objects管理器

django为每一个ORM类都创建了名为objects的管理器，可以通过这个管理器的方法实现对表的管理。

```
>>> q2 = Question.objects.create(question_text='从达内结业 后，你希望去哪个城市工作？ ',
pub_date='2019-06-01 12:00:00')
```

创建选项

- 实例化

```
>>> c1 = Choice(choice_text='腾讯', question=q1)
>>> c1.save()
```

- 使用管理器

```
>>> c2 = Choice.objects.create(choice_text='华为', question=q1)
```

- 通过问题实例的管理器创建选项

问题和选项有主外键约束，这是一对多的关系，即一个问题可以有多个选项。每个问题的实例都有一个xxx_set管理器。问题的选项是Choice，那么管理器就是choice_set，如果选项的类名是XuanXiang，那么管理器的名称是xuanxiang_set。

```
>>> c3 = q1.choice_set.create(choice_text='阿里巴巴')
```

删除选项

```
>>> c3.delete()
```

修改问题

```
>>> q2.question_text = '从达内结业后，你打算去哪个城市工作？'  
>>> q2.save()
```

查询

```
# 查询所有的问题  
>>> Question.objects.all()  
>>> for q in Question.objects.all():  
...     print(q.pub_date)  
...  
2019-05-25 17:22:00  
2019-04-25 12:00:00  
2019-05-27 09:24:00  
2019-05-27 09:00:00  
2019-06-01 12:00:00  
>>> Question.objects.order_by('pub_date') # 根据发布时间升序排列  
>>> for q in Question.objects.order_by('pub_date'):  
...     print(q.pub_date)  
...  
2019-04-25 12:00:00  
2019-05-25 17:22:00  
2019-05-27 09:00:00  
2019-05-27 09:24:00  
2019-06-01 12:00:00  
>>> Question.objects.order_by('-pub_date') # 根据发布时间降序排列  
  
# 获取某一个问题的实例  
# get必须得到一个实例，否则报错。  
>>> Question.objects.get(id=1) # 返回1号问题  
>>> Question.objects.get(id=10) # 如果不存在，则报错  
>>> Question.objects.get(id__gt=1) # id>1的问题，不只一项，报错。  
  
# 获取多个实例  
# filter可以得到0到多个实例的集合  
>>> Question.objects.filter(id=1) # 具有一项的查询集  
>>> Question.objects.filter(id=10) # 查询集为空  
>>> Question.objects.filter(id__gt=1) # 查询集有多项
```

查询条件

查询条件采用的形式是“属性__操作符=值”，id=1实际上是id__exact=1的简写。

```
>>> Question.objects.filter(id__exact=1) # 等于1
>>> Question.objects.filter(id__gt=1) # 大于1
>>> Question.objects.filter(id__gte=1) # 大于等于1
>>> Question.objects.filter(id__lt=1) # 小于1
>>> Question.objects.filter(id__lte=1) # 小于等于1

# 其他属性与数字属性类似，也是使用__
>>> Question.objects.filter(question_text__startswith='从达内')
>>> Question.objects.filter(pub_date__month=5)
```

完善首页

修改函数，取出所有的问题

```
from polls.models import Question

def index(request):
    questions = Question.objects.order_by('-pub_date')
    return render(request, 'index.html', {'questions': questions})
```

修改模板，显示所有问题

```
<!-- index.html -->
<body>
<div class="container">
    <h1>投票首页</h1>
    <div>
        <ol>
            {% for question in questions %}
                <li>
                    <a href="">{{ question.question_text }}</a>
                    {{ question.pub_date }}
                </li>
            {% endfor %}
        </ol>
    </div>
</div>
</body>
```

说明：{{var}}表示变量，{% %}是模板语法标签，在{}以外的是html语法。

为超链接加上具体的url

```
<li>
    <a href="http://127.0.0.1/polls/{{ question.id }}" target="_blank">
        {{ question.question_text }}
    </a>
    {{ question.pub_date }}
</li>
```

修改超链接

```
<a href="{% url 'detail' question_id=question.id %}" target="_blank">
```

引入bootstrap

```
# 将网页制作用到的static目录拷贝到polls目录下
[root@room8pc16 mysite]# cp -r ../../day02/static/ polls/
# 修改index.html首页,使用bootstrap
# index.html
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>投票首页</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">
</head>
<body>
<div class="container">
  <div id="linux-carousel" class="carousel slide">
    <ol class="carousel-indicators">
      <li class="active" data-target="#linux-carousel" data-slide-to="0"></li>
      <li data-target="#linux-carousel" data-slide-to="1"></li>
      <li data-target="#linux-carousel" data-slide-to="2"></li>
    </ol>
    <div class="carousel-inner">
      <div class="item active">
        <a href="http://www.sogou.com" target="_blank">
          
        </a>
      </div>
      <div class="item">
        
      </div>
      <div class="item">
        
      </div>
    </div>
    <a href="#linux-carousel" data-slide="prev" class="carousel-control left">
      <span class="glyphicon glyphicon-chevron-left"></span>
    </a>
    <a href="#linux-carousel" data-slide="next" class="carousel-control right">
      <span class="glyphicon glyphicon-chevron-right"></span>
    </a>
  </div>
  <h1 class="text-center text-warning">投票首页</h1>
  <div class="h4">
    <ol>
      {% for question in questions %}
        <li>
```

```

        <a href="{% url 'detail' question_id=question.id %}" target="_blank">
            {{ question.question_text }}
        </a>
        {{ question.pub_date }}
    </li>
{% endfor %}
</ol>
</div>
<div class="footer text-center">
    <a href="">达内云计算 NSD1812</a>
</div>
</div>

<script src="{% static 'js/jquery.min.js' %}"></script>
<script src="{% static 'js/bootstrap.min.js' %}"></script>
<script type="text/javascript">
    $('#linux-carousel').carousel({
        interval : 3000
    });
</script>
</body>
</html>

```

使用模板继承

- 把共性的内容，写到基础模板中
- 在基础模板添加block占位符
- 创建html页面，继承模板，将个性内容写到block中

```

# 拷贝index.html为base.html
# cp index.html base.html
# 将base.html中的个性内容用block替代
# base.html
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>{% block title %}{% endblock %}</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">
</head>
<body>
<div class="container">
    <div id="linux-carousel" class="carousel slide">
        <ol class="carousel-indicators">
            <li class="active" data-target="#linux-carousel" data-slide-to="0"></li>
            <li data-target="#linux-carousel" data-slide-to="1"></li>
            <li data-target="#linux-carousel" data-slide-to="2"></li>
        </ol>
        <div class="carousel-inner">
            <div class="item active">

```

```

        <a href="http://www.sogou.com" target="_blank">
            
        </a>
    </div>
    <div class="item">
        
    </div>
    <div class="item">
        
    </div>
</div>
<a href="#linux-carousel" data-slide="prev" class="carousel-control left">
    <span class="glyphicon glyphicon-chevron-left"></span>
</a>
<a href="#linux-carousel" data-slide="next" class="carousel-control right">
    <span class="glyphicon glyphicon-chevron-right"></span>
</a>
</div>
<div>
    {% block content %}{% endblock %}
</div>
<div class="footer text-center">
    <a href="">达内云计算 NSD1812</a>
</div>
</div>

<script src="{% static 'js/jquery.min.js' %}"></script>
<script src="{% static 'js/bootstrap.min.js' %}"></script>
<script type="text/javascript">
    $('#linux-carousel').carousel({
        interval : 3000
    });
</script>
</body>
</html>

```

将index.html中的共性删除，将个性内容写到block中

index.html

```

{% extends 'base.html' %}
{% load static %}
{% block title %}投票首页{% endblock %}
{% block content %}
    <h1 class="text-center text-warning">投票首页</h1>
    <div class="h4">
        <ol>
            {% for question in questions %}
                <li>
                    <a href="{% url 'detail' question_id=question.id %}" target="_blank">
                        {{ question.question_text }}
                    </a>
                    {{ question.pub_date }}
                </li>
            {% endfor %}
        </div>
    </div>

```

```
</ol>
</div>
{% endblock %}
```

实现投票详情页

修改函数

```
def detail(request, question_id):
    question = Question.objects.get(id=question_id)
    return render(request, 'detail.html', {'question': question})
```

修改模板文件

```
# detail.html
{% extends 'base.html' %}
{% load static %}
{% block title %}投票详情{% endblock %}
{% block content %}
    <h1 class="text-center text-warning">{{ question.id }}号问题投票详情</h1>
    <h3>{{ question.question_text }}</h3>
    <form action="" method="post">
    {% csrf_token %}
        {% for choice in question.choice_set.all %}
            <div class="radio">
                <label>
                    <input type="radio" name="choice_id" value="{{ choice.id }}">
                    {{ choice.choice_text }}
                </label>
            </div>
        {% endfor %}
        <div class="form-group">
            <input class="btn btn-primary" type="submit" value="投票">
        </div>
    </form>
{% endblock %}
```

实现投票功能

- 在投票详情页选择某一项后，投票
- 投票需要修改数据库。数据库通过调用函数进行修改
- 访问某一URL，触发函数调用

为投票函数定义URL

```
# polls/urls.py
url(r'^(?P<question_id>\d+)/vote/$', views.vote, name='vote'),
```

定义投票函数

```
from django.shortcuts import render, redirect

def vote(request, question_id):
    # 取出问题
    question = Question.objects.get(id=question_id)
    if request.method == 'POST':
        # 取出用户的选择
        choice_id = request.POST.get('choice_id')
        # 取出选项实例
        choice = question.choice_set.get(id=choice_id)
        choice.votes += 1
        choice.save()

    # redirect相当于打开一个新窗口，访问网址。
    # 如果仍然采用render，将会把request的数据继续向result传递
    return redirect('result', question_id=question_id)
```

修改表单的action行为

```
action="{% url 'vote' question_id=question.id %}"
```

实现投票结果

修改函数

```
def result(request, question_id):
    question = Question.objects.get(id=question_id)
    return render(request, 'result.html', {'question': question})
```

修改结果模板

```
<!--result.html -->
{% extends 'base.html' %}
{% load static %}
{% block title %}投票结果{% endblock %}
{% block content %}
    <h1 class="text-center text-warning">{{ question.id }}号问题投票结果</h1>
    <h3>{{ question.question_text }}</h3>
    <table class="table table-bordered table-striped table-hover h4">
        <tr class="info text-center">
            <td>选项</td>
            <td>票数</td>
        </tr>
        {% for choice in question.choice_set.all %}
            <tr>
                <td>{{ choice.choice_text }}</td>
                <td>{{ choice.votes }}</td>
```



```
</tr>  
{% endfor %}
```

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