

## 2장. 템플릿 파일 및 폼



# 목 차

1

템플릿 파일 - 질문 목록 페이지

2

템플릿 파일 - 질문/답변 상세 페이지

3

폼(form) - 질문등록 및 답변 등록

# 질문 목록 페이지 만들기

- index 페이지 – views.py

```
def index(request):  
    # 질문/답변 게시판  
    #question_list = Question.objects.all() #모든 데이터 조회하기  
    # 작성일을 기준으로 내림차순 정렬  
    question_list = Question.objects.order_by('-create_date')  
    return render(request, 'board/question_list.html',  
                  {'question_list':question_list})
```

-create\_date: 작성한 날짜의 역순으로 정렬

{**'question\_list'**:question\_list} : 딕셔너리 자료형으로 작성

# 템플릿으로 질문 목록 페이지 만들기

- question\_list.html

- 경로 : C:/webproject/mysite/templates/board/question\_list.html

```
<h1>질문/답변 게시판</h1>
{% if question_list %}
    {% for question in question_list %}
        <ul>
            <li>
                <a href="{% url 'board:detail' question.id %}">
                    {{ question.subject }}
                </a>
            </li>
        </ul>
    {% endfor %}
{% else %}
    <p>질문이 없습니다.</p>
{% endif %}
```



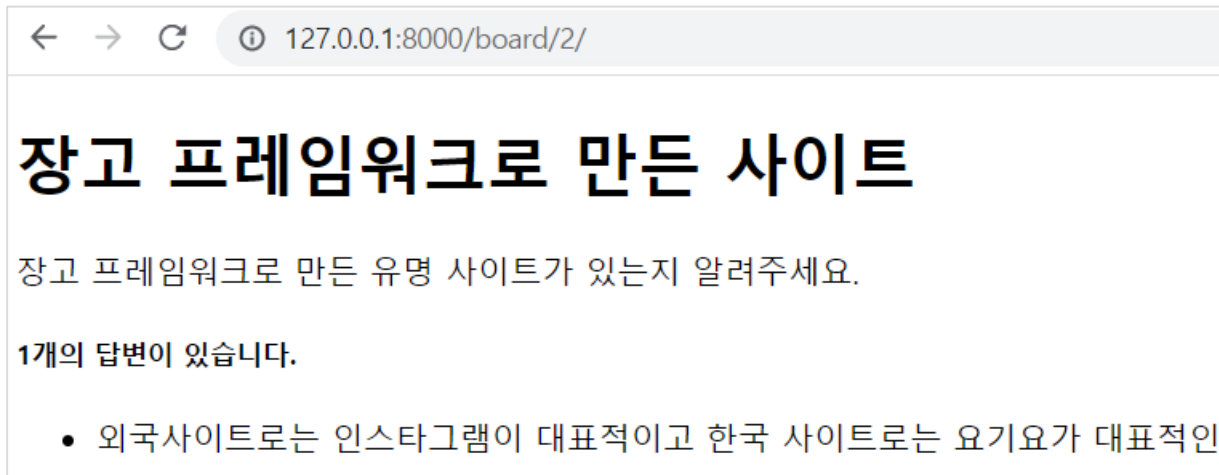
## 템플릿으로 질문 목록 페이지 만들기

- 템플릿(template) 태그

템플릿 태그	설 명
<code>{% if question_list %}</code>	question_list가 있다면(조건문)
<code>{% for question in question_list %}</code>	question_list를 반복하면 순차적으로 question에 대입(반복문)
<code>{{ question.id }}</code>	question 객체의 id 출력(출력문)
<code>{{ forloop.counter0 }}</code> <code>{{ forloop.counter }}</code>	for 반복문의 인덱스로 0부터 시작 0, 1 , 2, 3 ...

# 질문/답변 상세 페이지 만들기

## ◆ 질문/답변 상세 페이지



# 질문/답변 상세 페이지 만들기

## ◆ 질문/답변 상세 페이지

1. detail 페이지 URL 매핑하기 – board/urls.py

```
from django.urls import path
from pybo import views

app_name = 'board'
urlpatterns = [
    path("", views.index),
    path('<int:question_id>/', views.detail, name='detail'),
]
```

# 질문/답변 상세 페이지 만들기

## ◆ 질문/답변 상세 페이지

2. detail 함수 구현 – board/views.py

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



# 질문/답변 상세 페이지 만들기

## ◆ 질문/답변 상세 페이지

### 3. board/detail.html 작성

```
<h1>{{ question.subject }}</h1>
<div>
    {{ question.content }}
</div>
<h5>{{ question.answer_set.count }}개의 답변이 있습니다.</h5>
<div>
    <ul>
        {% for answer in question.answer_set.all %}
        <li>{{ answer.content }}</li>
        {% endfor %}
    </ul>
</div>
```

# 질문 등록 폼 만들기

## ● 질문 등록 폼 만들기

### 1. question\_list.html 에 질문 등록 메뉴 만들기

#### 질문/답변 게시판

- [pybo란 무엇인가?](#)
- [장고 프레임워크로 만든 사이트](#)
- [오미크론](#)
- [겨울철 난방온도](#)

[질문 등록](#)

```
{% else %}
    <p>질문이 없습니다.</p>
{% endif %}
<a href="{% url 'board:question_create' %}">질문 등록</a>
```

# 질문 등록 폼 만들기

## 2. QuestonForm 만들기 – board/forms.py

board 디렉터리 하위에 **forms.py** 생성하기

```
from django import forms
from board.models import Question, Answer

class QuestionForm(forms.ModelForm):
    class Meta:
        model = Question
        fields = ['subject', 'content']
        labels = {
            'subject': '제목',
            'content': '내용'
        }
```

forms.ModelForm 폼 상속

- 모델 폼 객체를 저장하면  
연결된 모델의 데이터를  
저장할 수 있다.

- 내부 클래스로 반드시  
Meta Class를 가져야 한다.

# 질문 등록 폼 만들기

## 3. 질문 등록 URL 매핑하기 – board/urls.py

```
urlpatterns = [  
    ...  
    path('/question/create/', views.question_create, name='question_create'),  
]
```

## 4. question\_create 함수 정의 – pybo/views.py

```
def question_create(request):  
    # 질문 등록  
    form = QuestionForm()  
    context = {'form': form}  
    return render(request, 'pybo/question_form.html', context)
```

# 질문 등록 폼 만들기

## 5. question\_form.html 만들기

← → ↻ ⓘ 127.0.0.1:8000/board/question/create/

### 질문 등록

제목:

내용: 

깃허브가 무엇인가요?

```
<h2>질문 등록</h2>
<form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">저장하기</button>
</form>
```

# 질문 등록 폼 만들기

## 6. question\_create 함수 POST 코드 작성

```
def question_create(request):  
    #질문 등록  
    if request.method == "POST":  
        form = QuestionForm(request.POST)  
        if form.is_valid():  
            question = form.save(commit=False) # 가져장  
            question.create_date = timezone.now()  
            question.save() #실제 저장  
            return redirect('board:index') #index 경로로 이동  
        else: # request.method == "GET"  
            form = QuestionForm()  
            context = {'form': form} # context 변수에 딕셔너리 자료 저장  
            return render(request, 'board/question_form.html', context)
```

# GET 방식과 POST 방식 비교

## ➤ GET 방식과 POST 방식 비교

### 질문/답변 게시판

- [pybo란 무엇인가?](#)
- [장고 프레임워크로 만든 사이트](#)
- [오미كرون](#)
- [겨울철 난방온도](#)

[질문 등록](#)

**GET**

← → ↻ ⓘ 127.0.0.1:8000/board/question/create/

### 질문 등록

제목:

내용:

저장하기

**POST**

GET 방식 요청 : 질문 등록하기 -> question/create 페이지 보여줌

POST 방식 요청: 폼에 내용 입력 -> 저장하기

# 답변 등록 폼 만들기

## ● 답변 등록 폼 만들기

### 장고 프레임워크로 만든 사이트

장고 프레임워크로 만든 유명 사이트가 있는지 알려주세요.

1개의 답변이 있습니다.

- 외국사이트로는 인스타그램이 대표적이고 한국 사이트로는 요기요

답변 등록



# 답변 등록 폼 만들기

## 1. detail.html에 답변 등록 폼 만들기

```
<h5>{{ question.answer_set.count }}개의 답변이 있습니다.</h5>
<div>
  <ul>
    {% for answer in question.answer_set.all %}
    <li>{{ answer.content }}</li>
    {% endfor %}
  </ul>
</div>
<form action="{% url 'board:answer_create' question.id %}"
  method="post">
  {% csrf_token %}
  <div>
    <textarea name="content" id="content"
      rows="10" cols="40"></textarea>
  </div>
  <input type="submit" value="답변 등록">
</form>
```

# 답변 등록 폼 만들기

## 2. AnswerForm 클래스 정의 – board/forms.py(새로 생성)

```
class AnswerForm(forms.ModelForm):  
    class Meta:  
        model = Answer  
        fields = ['content']  
        labels = {  
            'content': '답변내용'  
        }
```

# 답변 등록 페이지 만들기

## 3. 답변 등록 URL 매핑하기 – board/urls.py

```
urlpatterns = [  
    ...  
    path('/answer/create/<int:question_id>/', views.answer_create,  
          name='answer_create'),  
]
```

# 답변 등록 페이지 만들기

## 4. answer\_create 함수 재정의 – board/views.py

```
def answer_create(request, question_id):  
    #답변 등록  
    question = Question.objects.get(id=question_id)  
    if request.method == "POST":  
        form = AnswerForm(request.POST) # 폼의 입력된 데이터 전달 받음  
        if form.is_valid():  
            answer = form.save(commit=False) # 가져장  
            answer.question = question # 외래키 저장  
            answer.create_date = timezone.now()  
            answer.save() #실제 저장  
            return redirect('board:detail', question_id=question.id)  
            # 해당 id 경로로 이동  
        else:  
            form = AnswerForm()  
    context = {'question':question, 'form':form}  
    return render(request, 'board/detail.html', context)
```