

[Django] #12 - Django 게시판 만들기(게시판 등록 하기) - 3 — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2020-10-20 오전 8:24

URL: <https://continuous-development.tistory.com/101?category=805760>

Django

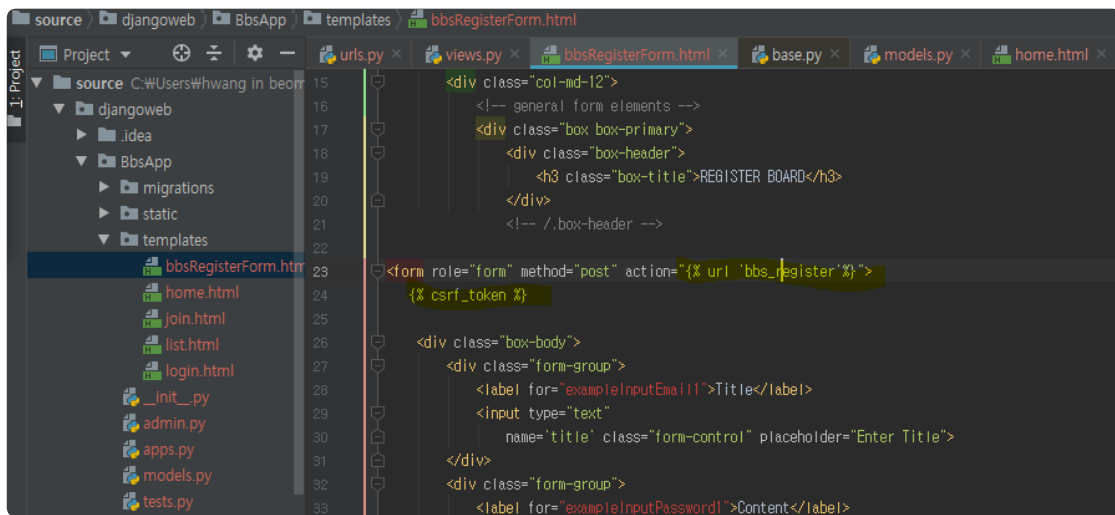
[Django] #12 - Django 게시판 만들기(게시판 등록 하기) - 3

2020. 10. 1. 14:40 수정 삭제 공개

게시판 등록

The screenshot shows a web browser with multiple tabs. The active tab is titled '새로운 글쓰기' (New Post). The address bar shows the URL '127.0.0.1:8000/bbs/bbs_registerForm/'. The browser's left sidebar is open, showing a navigation menu with 'Forms' selected. The main content area displays a registration form titled 'REGISTER BOARD'. The form has three input fields: 'Title' (with placeholder 'Enter Title'), 'Content' (with placeholder 'Enter ...'), and 'Writer' (with placeholder 'OOO'). Below the fields is a 'Submit' button. The browser's top bar shows the user '나무늘보' is online.

해당 부분에서 Submit을 눌렀을 때 게시판에 글이 등록되는 기능을 구현해보자. 일단 첫 번째로

A screenshot of an IDE showing the Django project structure. The 'templates' directory is expanded, showing 'bbsRegisterForm.html' as the active file. The code in the file is a Django template for a registration form. It includes a form with a title input field and a content input field. The form's action is set to '{% url 'bbs_register' %}', and it includes a CSRF token. The form is styled with Bootstrap classes like 'box-primary', 'box-header', and 'box-body'.

```
15 <div class="col-md-12">
16 <!-- general form elements -->
17 <div class="box box-primary">
18 <div class="box-header">
19 <h3 class="box-title">REGISTER BOARD</h3>
20 </div>
21 <!-- /.box-header -->
22
23 <form role="form" method="post" action="{% url 'bbs_register' %}">
24 <div class="box-body">
25 <div class="form-group">
26 <input type="text"
27 <label form="exampleInputEmail1">Title</label>
28 <input type="text"
29 <label form="exampleInputPassword1">Content</label>
30 </div>
31 </div>
32 </div>
33 </div>
```

해당 템플릿에서 Submit의 위치를 찾는다. 그 버튼을 찾은 후 action 부분이 실행되면 해당 url로 이동하게끔 구현한다.

{% %} 태그는 Django에서만 쓸수 있는 태그이다. 이 형식으로 url을 타게끔 한다.

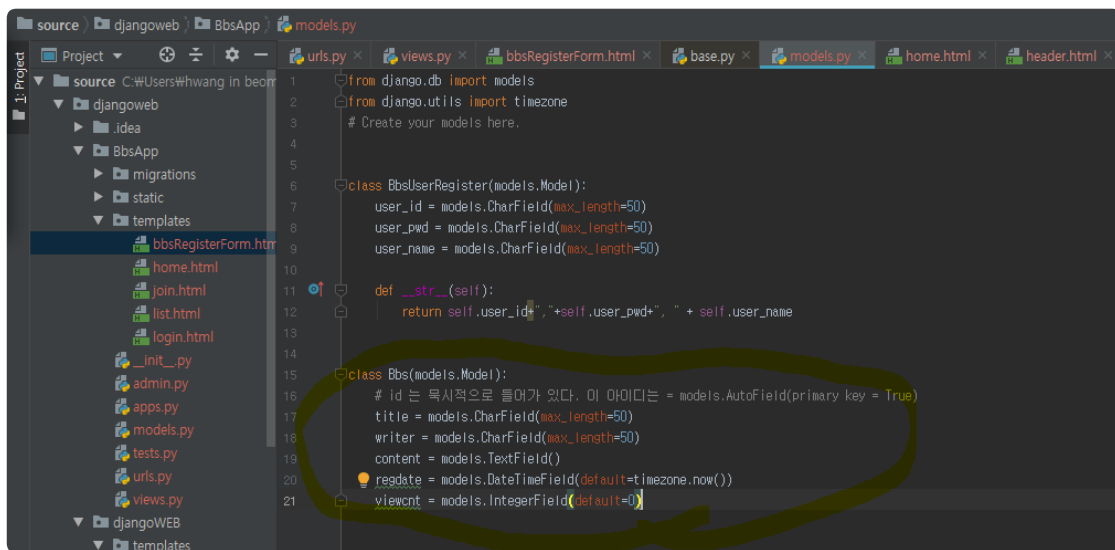
만약 post 방식으로 값을 보낸다면 csrf_token을 넣어줘야 한다. csrf_token은 Django에서 보안상의 이유로 사용하는 토큰이다.

이 url 패스에 따른 url을 설정해주자.

A screenshot of an IDE showing the 'urls.py' file. The file contains Django URL patterns for the project. It imports 'path' and 'include' from 'django.urls' and 'views' from 'BbsApp'. The 'urlpatterns' list includes paths for 'index', 'registerForm', 'register', 'login', 'logout', 'bbs_list', 'bbs_registerForm', and 'bbs_register'. The 'bbs_register' path is highlighted in yellow.

```
1 from django.contrib import admin
2 from django.urls import path, include
3 from BbsApp import views
4
5 urlpatterns = [
6     path('index/', views.loginForm, name='loginForm'),
7     path('registerForm/', views.registerForm, name='registerForm'),
8     path('register/', views.register, name='register'),
9     path('login/', views.login, name='login'),
10    path('logout/', views.logout, name='logout'),
11    path('bbs_list/', views.list, name='bbs_list'),
12    path('bbs_registerForm/', views.bbsRegisterForm, name='bbs_registerForm'),
13    path('bbs_register/', views.bbsRegister, name='bbs_register'),
14]
```

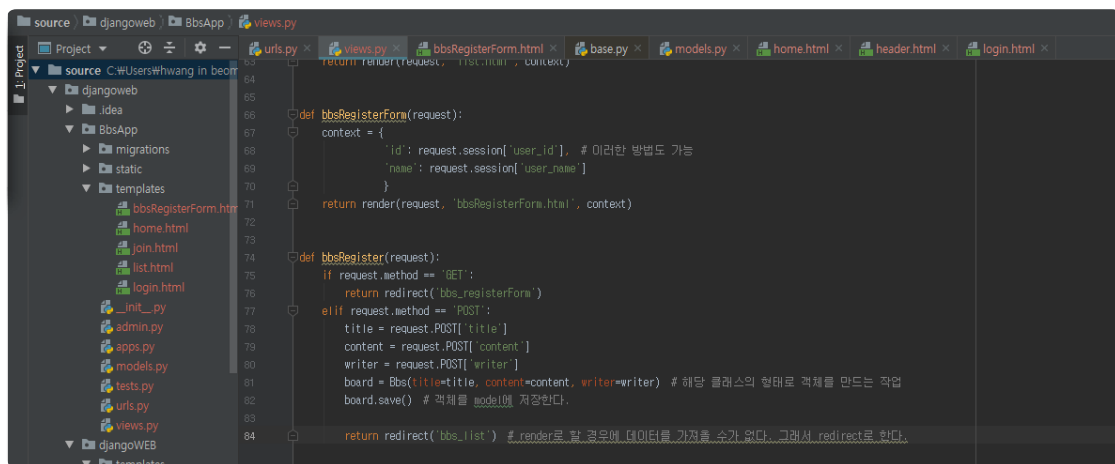
또한 이 url에 대한 view 부분을 구성하자.



```
1 from django.db import models
2 from django.utils import timezone
3 # Create your models here.
4
5
6 class BbsUserRegister(models.Model):
7     user_id = models.CharField(max_length=50)
8     user_pwd = models.CharField(max_length=50)
9     user_name = models.CharField(max_length=50)
10
11     def __str__(self):
12         return self.user_id + "," + self.user_pwd + " " + self.user_name
13
14
15 class Bbs(models.Model):
16     # id 는 묵시적으로 들어가 있다. 이 아이디는 = models.AutoField(primary key = True)
17     title = models.CharField(max_length=50)
18     writer = models.CharField(max_length=50)
19     content = models.TextField()
20     regdate = models.DateTimeField(default=timezone.now())
21     viewcnt = models.IntegerField(default=0)
```

view를 구성하기전에 게시판은 하나의 글이 생기는 것이다. 이것은 하나의 객체 = 테이블의 행이 생기는 것으로 models에서 하나의 클래스(테이블)를 생성해주어야 한다. 우리가 원하는 형태 클래스(테이블)를 만들어 준다. 여기서는 Bbs라는 클래스이다. models 부분을 보면 해당 Bbs에 title, writer, content를 사용하게 된다.

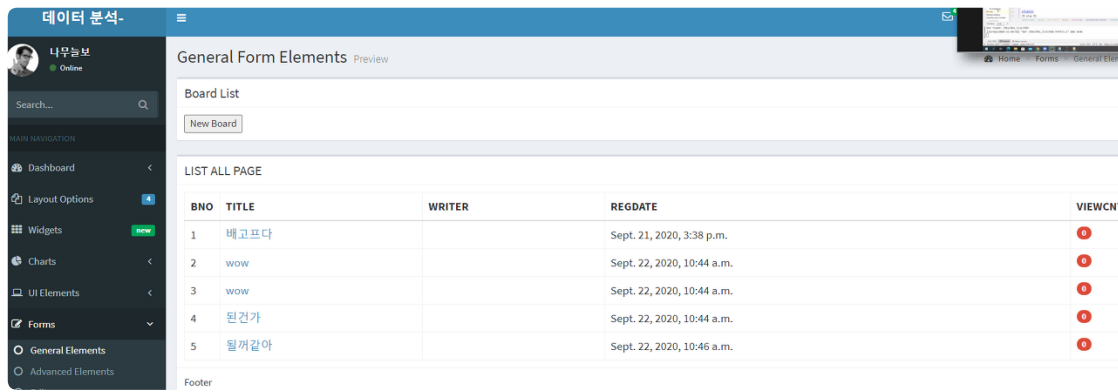
우리는 이 모델을 통해 게시글이 추가되는 작업을 진행해야 한다. 이 작업은 view에서 이뤄지게 된다.



```
64 return render(request, 'list.html', context)
65
66 def bbsRegisterForm(request):
67     context = {
68         'id': request.session['user_id'], # 이러한 방법도 가능
69         'name': request.session['user_name']
70     }
71     return render(request, 'bbsRegisterForm.html', context)
72
73 def bbsRegister(request):
74     if request.method == 'GET':
75         return redirect('bbs_registerForm')
76     elif request.method == 'POST':
77         title = request.POST['title']
78         content = request.POST['content']
79         writer = request.POST['writer']
80         board = Bbs(title=title, content=content, writer=writer) # 해당 클래스의 형태로 객체를 만드는 작업
81         board.save() # 객체를 model에 저장한다.
82
83     return redirect('bbs_list') # render로 할 경우에 데이터를 가져올 수가 없다. 그래서 redirect로 한다.
```

해당 모델을 가져와 객체를 만들고 그 안에 값을 넣어준다. Bbs를 호출하고 그 안에 title과 context, writer를 넣어준다. 이전에 해당 모델을 import 하는 작업 또한 진행해줘야 한다. 그다음 board.save를 통해 해당 객체를 저장하는 작업을 한다.

하고 redirect로 `bbs_list`로 보이게 한다.



BNO	TITLE	WRITER	REGDATE	VIEWCNT
1	배고프다		Sept. 21, 2020, 3:38 p.m.	0
2	wow		Sept. 22, 2020, 10:44 a.m.	0
3	wow		Sept. 22, 2020, 10:44 a.m.	0
4	원건가		Sept. 22, 2020, 10:44 a.m.	0
5	탈꺼갈아		Sept. 22, 2020, 10:46 a.m.	0

'Django' 카테고리의 다른 글

[Django] #14 - Django 게시판 만들기(게시글 수정) - 5

[Django] #13 - Django 게시판 만들기(게시글 보기) - 4

[Django] #12 - Django 게시판 만들기(게시판 등록 하기) - 3

[Django] #11 - Django 게시판 만들기(등록 버튼 구현) - 2

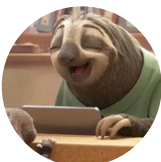
[Django] #10 - Django 게시판 만들기(기본적인 데이터 넣기) - 1

[Django] #9 - Django 로그아웃 버튼 구현

Django 등록

Django 등록 구현

Django 등록버튼 구현



나무늘보스

혼자 끄적끄적하는 블로그 입니다.