

노트북: 첫 번째 노트북

만든 날짜: 2020-10-21 오전 8:08

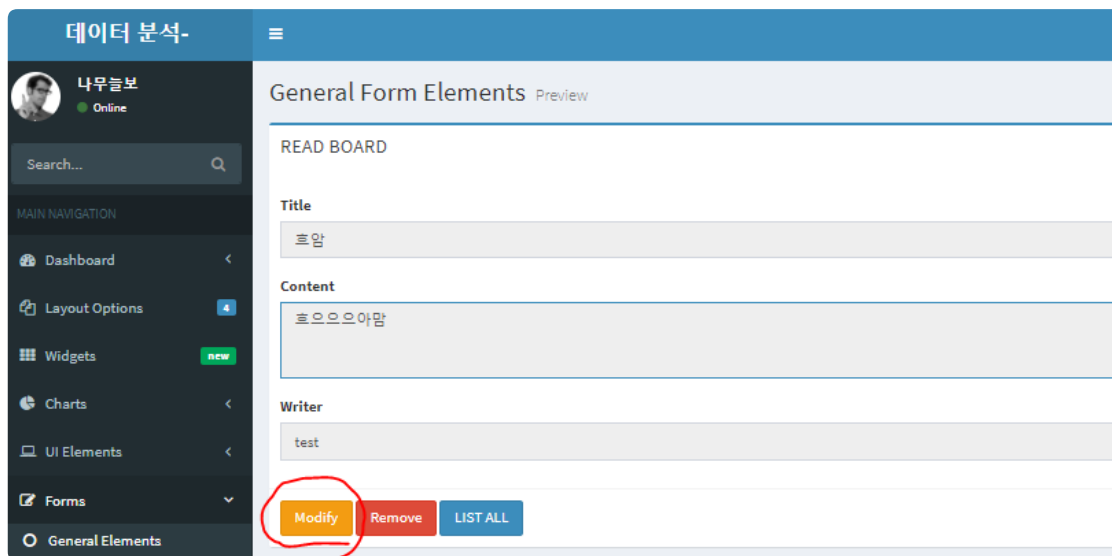
URL: <https://continuous-development.tistory.com/103?category=805760>

Django

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

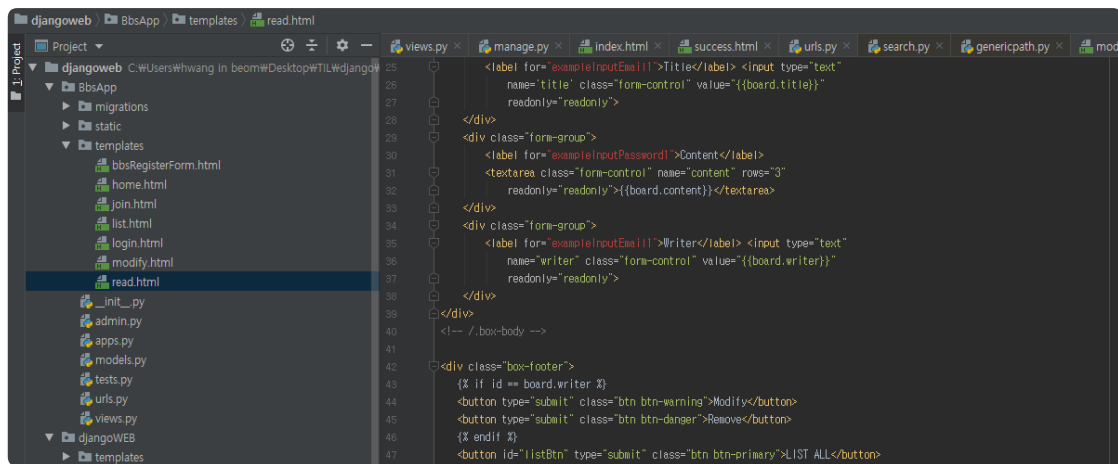
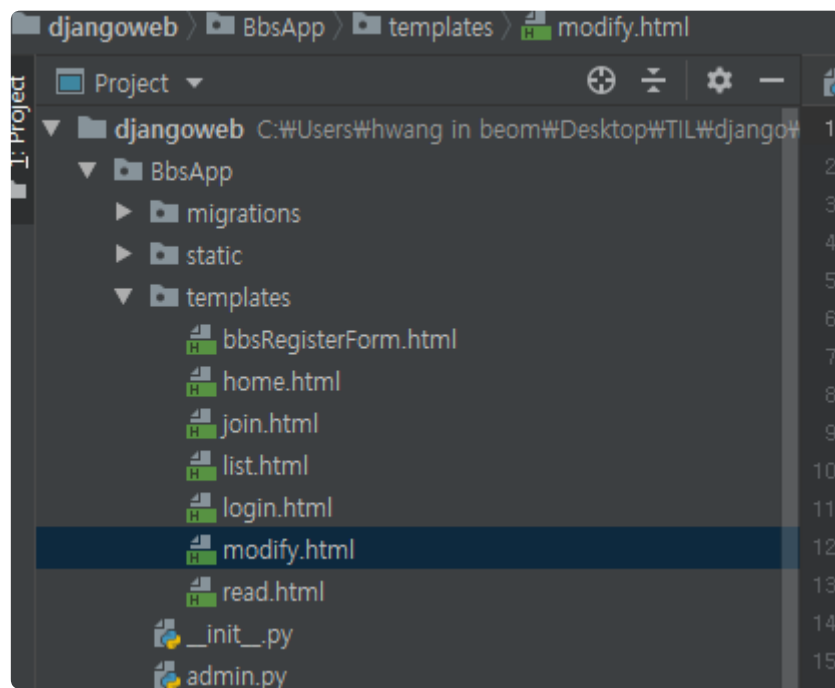
2020. 10. 1. 20:57 수정 삭제 공개

게시글 수정

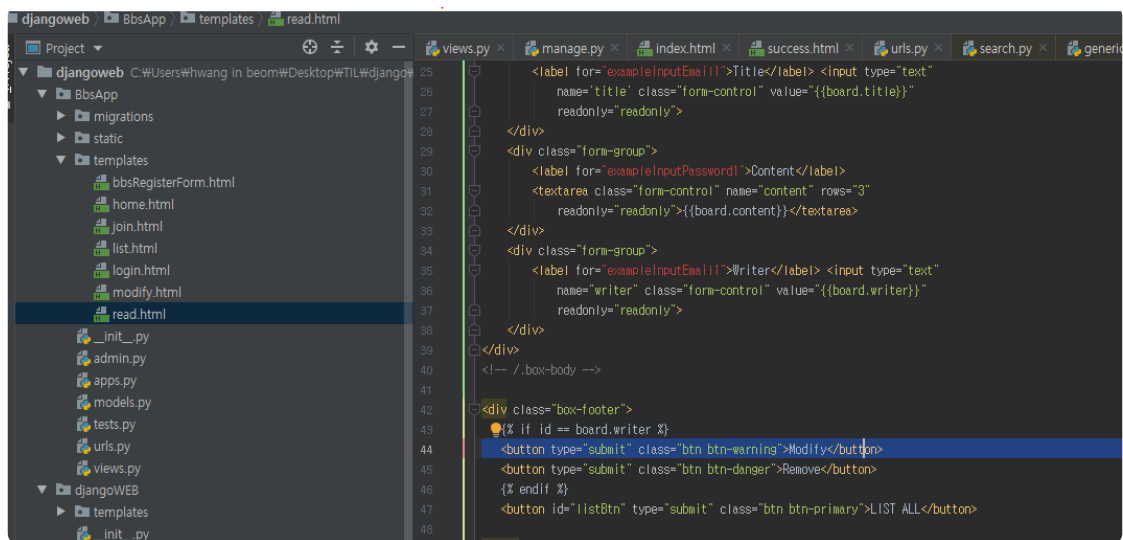


이 부분을 진행하려 한다.

일단 해당 템플릿이 있는지 확인하고



해당 modify가 있는 템플릿은 read.html이다



이 modify 부분을 찾고 해당 버튼을 클릭했을 alert가 뜨게끔 연결한다.

```

<div class="box-footer">
    {% if id == board.writer %}
    <button type="submit" class="btn btn-warning">Modify</button>
    <button type="submit" class="btn btn-danger">Remove</button>
    {% endif %}
    <button id="listBtn" type="submit" class="btn btn-primary">LIST ALL</button>
</div>

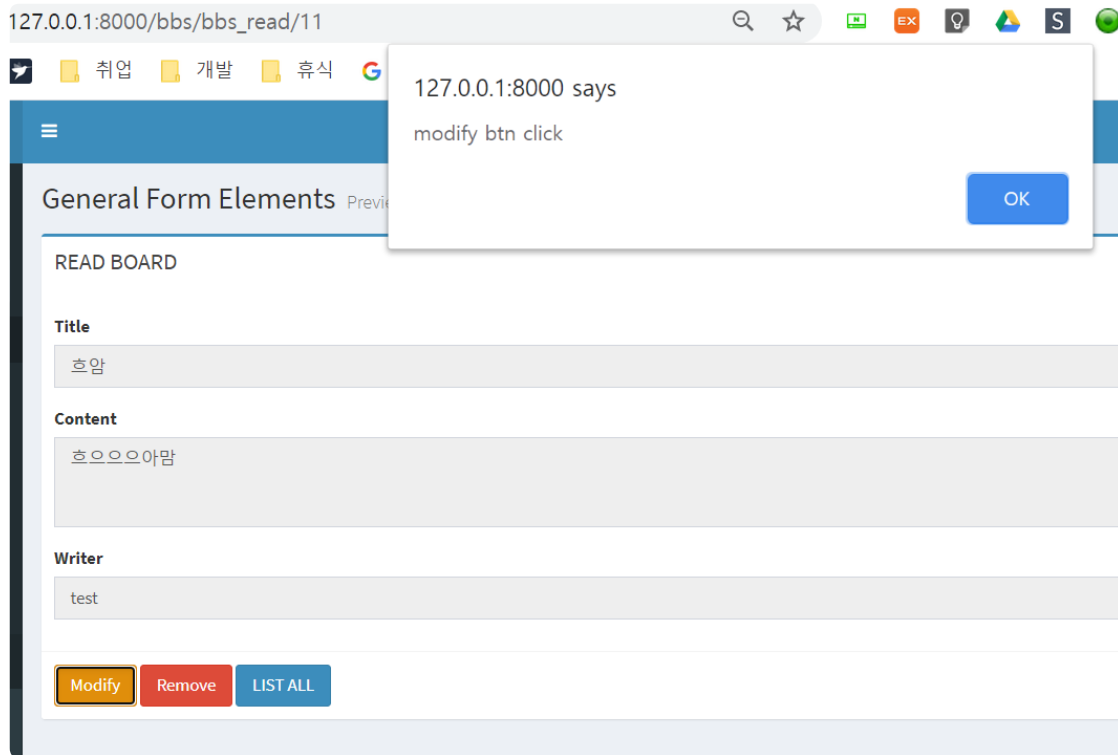
<script>

$(document).ready(function(){
    $('#listBtn').click(function(){
        location.href = '../bbs_list';
        //window.alert('click')
    })
    $('.btn-danger').click(function(){
        $('#removeFrm').attr('action', '../bbs_remove/');
        $('#removeFrm').submit();
    })
    $('.btn-warning').click(function(){
        alert('modify btn click')
        <!--$('#removeFrm').attr('action', '../bbs_remove/');-->
        <!--$('#removeFrm').submit();-->
    })
});

```

여기서는 btn-waring이라는 아이디를 가지고 있는 부분을 스크립트로 구현한다.

해당 부분의 class명을 통해 접근해야 되기 때문에 #으로 접근하고 해당 버튼을 클릭했을 때 alert 가 뜨게 한다.



Modify를 눌렀을 때 해당 alert이 정상적으로 뜨는 걸 볼 수있다.

이제 Modify를 하는 데 있어서 세부적인 로직을 짜 보자.

이 이벤트를 발생하는 데 있어서 과정은

- 1.url로 보내준다.
- 2.view로 보내준다.
- 3.view에서 해당 로직을 만들어준다.
4. 여기서는 modify라는 페이지로 보내준다.

이 modify 페이지로 보내는 데 있어서 해당 글에 대한 정보를 들고 가서 뿌려줘야 한다.

urls -> views -> template

```

39 </div>
40 </div>
41 <!-- /.box-body -->
42
43 <div class="box-footer">
44     {% if id == board.writer %}
45     <button type="submit" class="btn btn-warning">Modify</button>
46     <button type="submit" class="btn btn-danger">Remove</button>
47     {% endif %}
48     <button id="listBtn" type="submit" class="btn btn-primary">LIST ALL</button>
49 </div>
50
51
52
53
54
55 <script>
56 $(document).ready(function(){
57     $('#listBtn').click(function(){
58         location.href = '../bbs_list';
59         //window.alert('click')
60     })
61     $('.btn-danger').click(function(){
62         $('#removeFrm').attr('action', '../bbs_remove/');
63         $('#removeFrm').submit();
64     })
65     $('.btn-warning').click(function(){
66         $('#removeFrm').attr('action', '../bbs_modifyForm/');
67         $('#removeFrm').submit();
68     })
69 })

```

templates에서 modifyForm이라는 url로 보낸다.

```

1 from django.conf.urls import url
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    path('bbs_read/<int:id>', views.bbsRead, name='bbs_read'), # 이렇게 get 방식에서 값을 받아온 것을 이런 형식으로 받아온다.
15    path('bbs_remove/', views.bbsRemove, name='bbs_remove'),
16    path('bbs_modifyForm/', views.bbsModify, name='bbs_modifyForm'),
17
18 ]

```

그다음 url에서 받는 부분을 구현하고 view로 넘긴다.

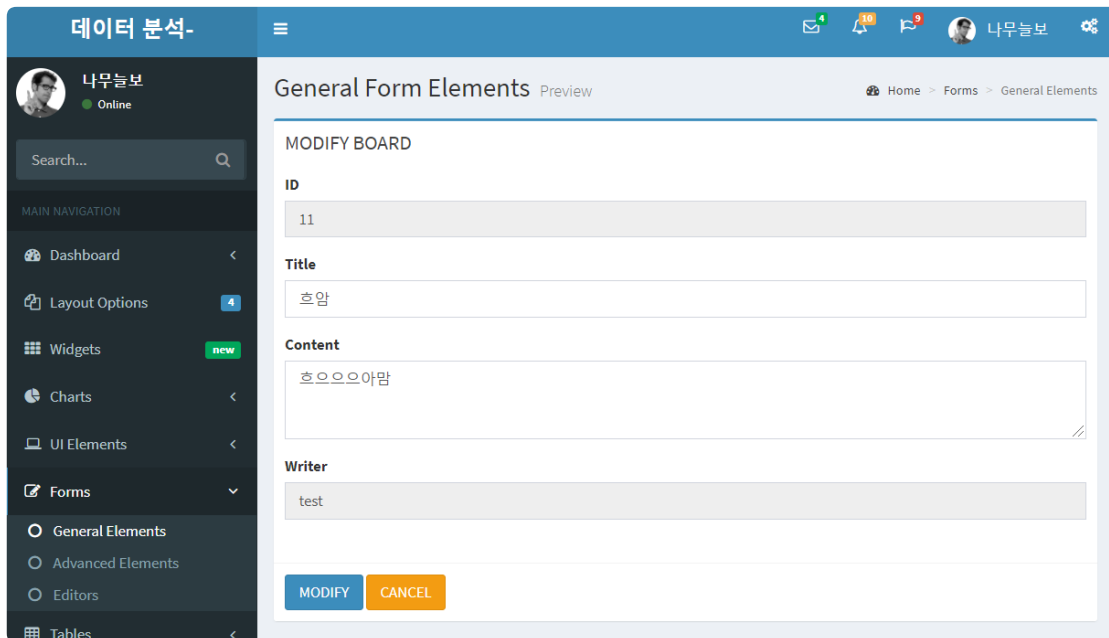
```

98 id = request.session['user_id'] # 이러한 방법도 가능
99 new = request.session['user_new']
100 board = board
101
102 print('para = ', board.writer)
103 print(id)
104 return render(request, 'read.html', context)
105
106
107 def bbsRemove(request):
108     id = request.POST['id']
109     Bbs.objects.get(id=id).delete()
110
111     return redirect('bbs_list') # render로 할 경우에 데이터를 가져올 수가 없다. 그래서 redirect로 한다.
112
113 def bbsModify(request):
114     id = request.POST['id']
115     board = Bbs.objects.get(id=id) # 객체를 가져온다.
116     context = {
117         'id': request.session['user_id'], # 이러한 방법도 가능
118         'new': request.session['user_new'],
119         'board': board
120     }
121
122     return render(request, 'modify.html', context)

```

Bbs.objects.get을 통해 해당 user_id에 해당하는 객체를 가져온다.
이렇게 view 단을 구현한다. 여기서 보낸 id 값을 받고 이 아이디를 통해
모델에 접근해 게시판에 대한 객체를 받아온다.

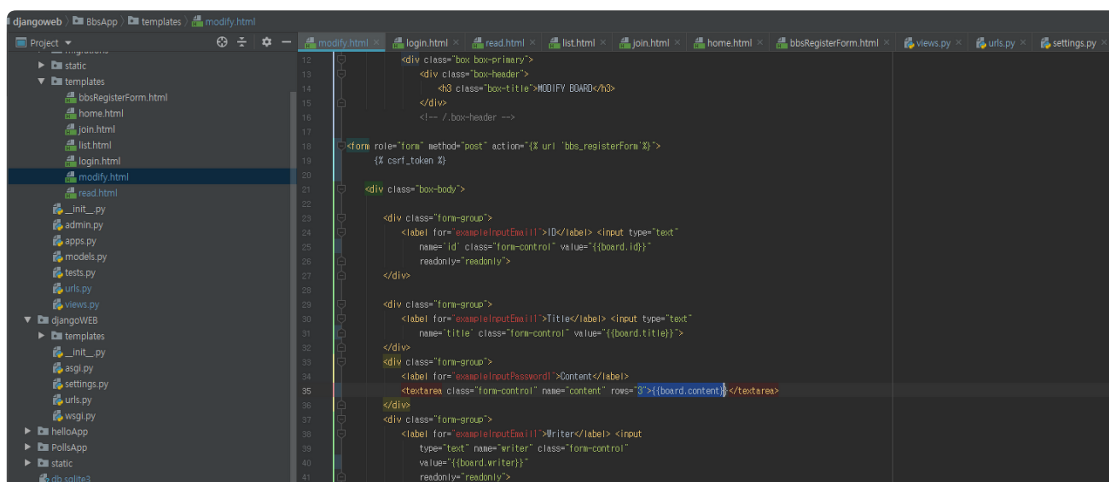
이 받아온 객체를 context에 넣어 modify의 템플릿으로 보낸다.



The screenshot shows a web application interface. On the left is a sidebar with a user profile '나무늘보' (Online) and a navigation menu including Dashboard, Layout Options, Widgets, Charts, UI Elements, Forms, General Elements, Advanced Elements, Editors, and Tables. The main content area is titled 'General Form Elements Preview' and shows a 'MODIFY BOARD' form. The form contains the following fields: ID (11), Title (함암), Content (함으으으아맘), and Writer (test). At the bottom of the form are two buttons: 'MODIFY' and 'CANCEL'.

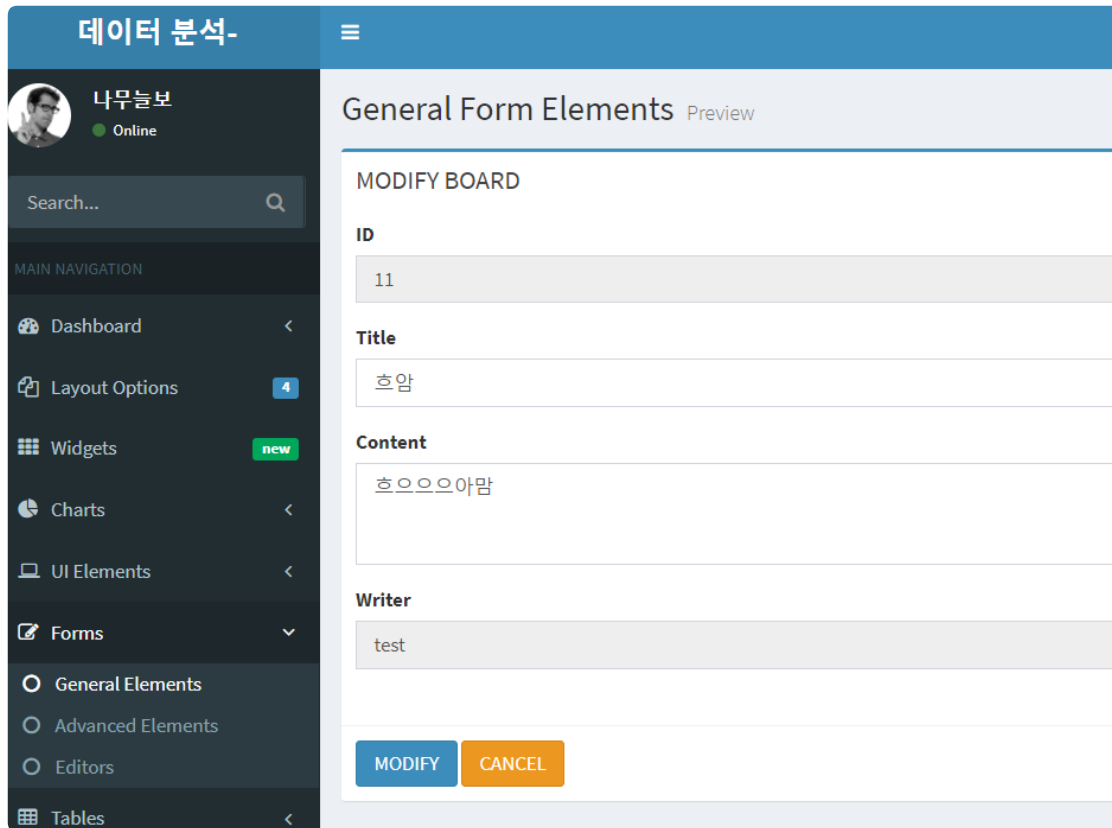
그러면 해당 페이지로 이동된다.

값이 뜨는 것은 보내준 context를



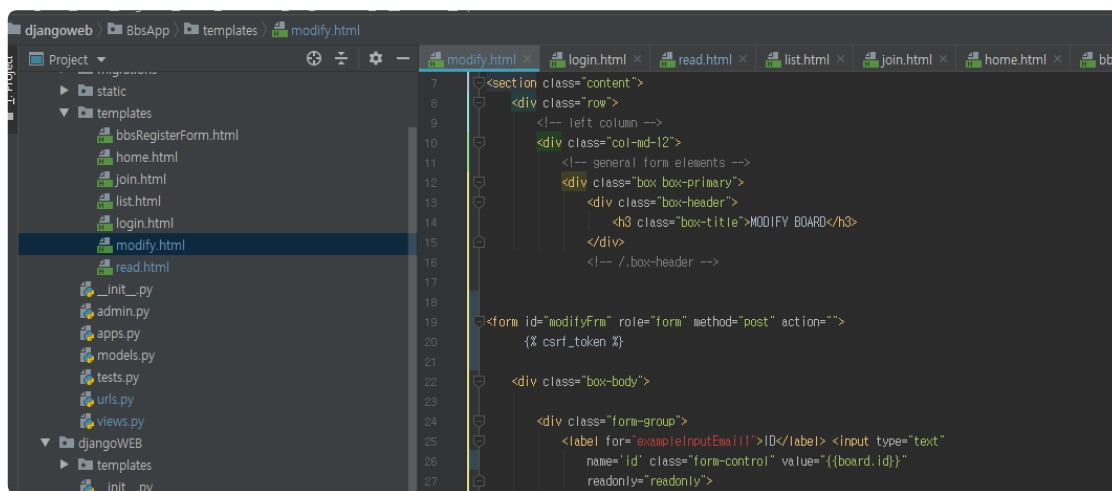
```
13 <div class="box box-primary">
14   <div class="box-header">
15     <h3 class="box-title">MODIFY BOARD</h3>
16   </div>
17   <div class="box-body">
18     <form role="form" method="post" action="{% url 'bbs_registerForm %}'>
19       {% csrf_token %}
20
21       <div class="form-group">
22         <label for="exampleInputId">ID</label> <input type="text"
23           name="id" class="form-control" value="{{ board.id }}"
24           readonly="">
25       </div>
26
27       <div class="form-group">
28         <label for="exampleInputTitle">Title</label> <input type="text"
29           name="title" class="form-control" value="{{ board.title }}">
30       </div>
31
32       <div class="form-group">
33         <label for="exampleInputContent">Content</label>
34         <textarea class="form-control" name="content" rows="5">{{ board.content }}</textarea>
35       </div>
36
37       <div class="form-group">
38         <label for="exampleInputWriter">Writer</label> <input
39           type="text" name="writer" class="form-control"
40           value="{{ board.writer }}"
41           readonly="">
```

이렇게 값을 넣어준다.



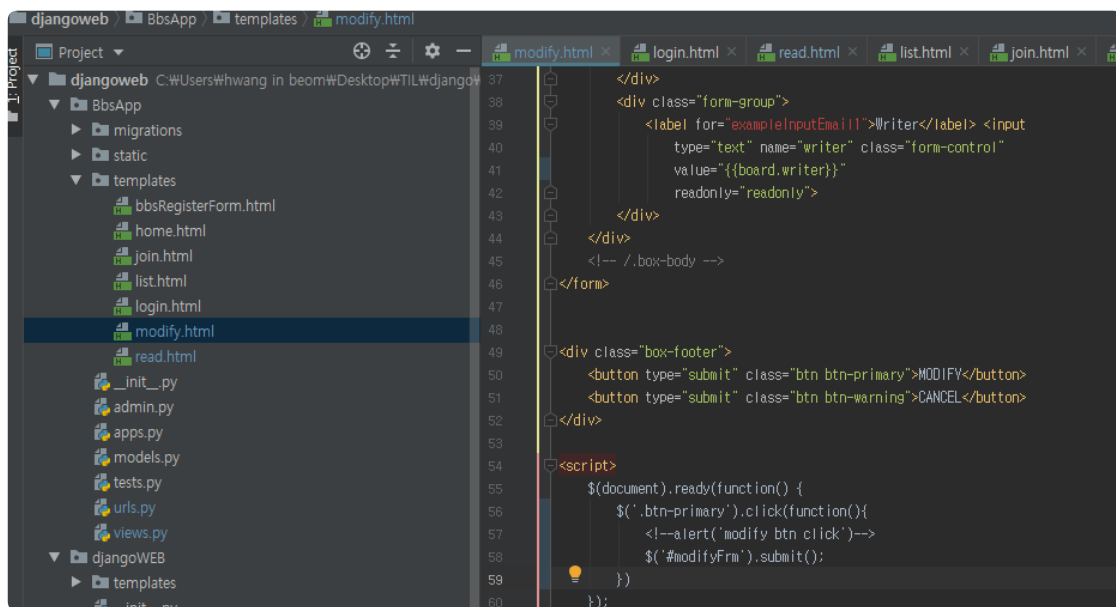
The screenshot shows a web application interface with a sidebar on the left and a main content area on the right. The sidebar has a header '데이터 분석-' and a user profile '나무늘보' with a status 'Online'. Below the profile is a search bar and a 'MAIN NAVIGATION' menu with items: Dashboard, Layout Options (4), Widgets (new), Charts, UI Elements, Forms, General Elements, Advanced Elements, Editors, and Tables. The main content area is titled 'General Form Elements Preview' and contains a 'MODIFY BOARD' form. The form has four fields: 'ID' with the value '11', 'Title' with the value '흐암', 'Content' with the value '흐으으아맘', and 'Writer' with the value 'test'. At the bottom of the form are two buttons: 'MODIFY' (blue) and 'CANCEL' (orange).

이제 아래 modify 버튼과 cancel 버튼을 구현하자.

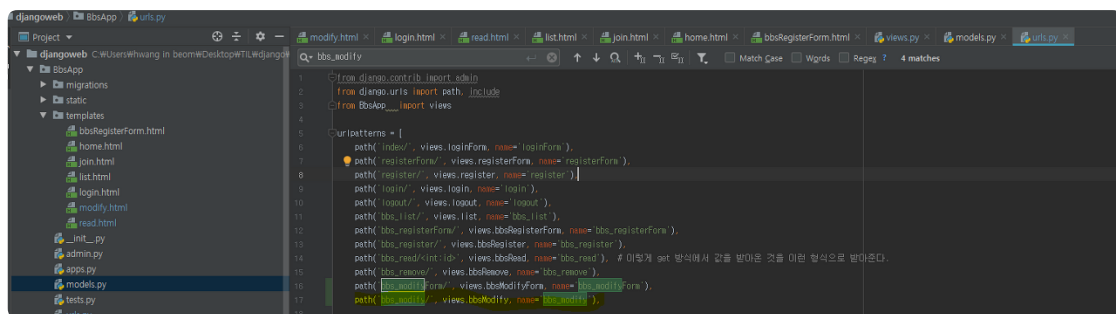


```
7 <section class="content">
8   <div class="row">
9     <!-- left column -->
10    <div class="col-md-12">
11      <!-- general form elements -->
12      <div class="box box-primary">
13        <div class="box-header">
14          <h3 class="box-title">MODIFY BOARD</h3>
15        </div>
16      <!-- /.box-header -->
17    </div>
18  </div>
19  <form id="modifyFrm" role="form" method="post" action="">
20    { csrf_token }
21  </form>
22  <div class="box-body">
23    <div class="form-group">
24      <label for="exampleInputEmail1">ID</label> <input type="text"
25      name="id" class="form-control" value="{{board.id}}"
26      readonly="readonly">
27    </div>
```

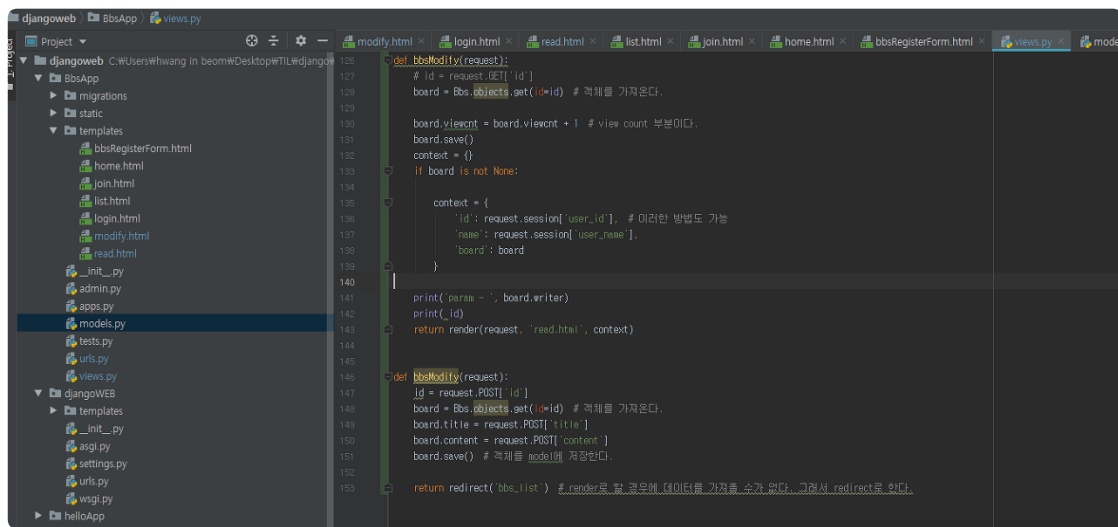
이 부분에 form에 대한 id 값을 정의해주고



이렇게 해당 class 값의 버튼이 클릭되었을 때 form의 id 값을 submit 한다는 스크립트를 만든다.



url 부분을 구현하고



해당 view 부분을 구현한다.

여기서 우리가 받아온 id 값을 request로 받고 그 id 값을 통해 objects를 가져온다.

그다음 그 객체에 우리가 가져온 값을 넣고 save로 저장한다. 그 후 redirect를 통해 bbs_list로 이동하게끔 만든다.

redirect는 render와 다르게 해당 url로 다시 보낸다. 이 말은 해당 urls.py에 있는 url로 이동을 시켜 해당 url의 있는 view를 타게끔 만든다. render는 해당 템플릿을 지정을 통해 템플릿으로 바로 보내는 차이가 있다.

'Django' 카테고리의 다른 글

[Django] #16 - Django csv 파일 사용하기 만들기

[Django] #15 - Django 게시판 만들기(검색기능 구현)- 6

[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 게시글 수정

django 수정



나무늘보스

혼자 끄적끄적하는 블로그 입니다.