

노트북: 첫 번째 노트북

만든 날짜: 2020-10-21 오전 8:09

URL: <https://continuous-development.tistory.com/104?category=805760>

---

Django

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

2020. 10. 3. 18:37 수정 삭제 공개

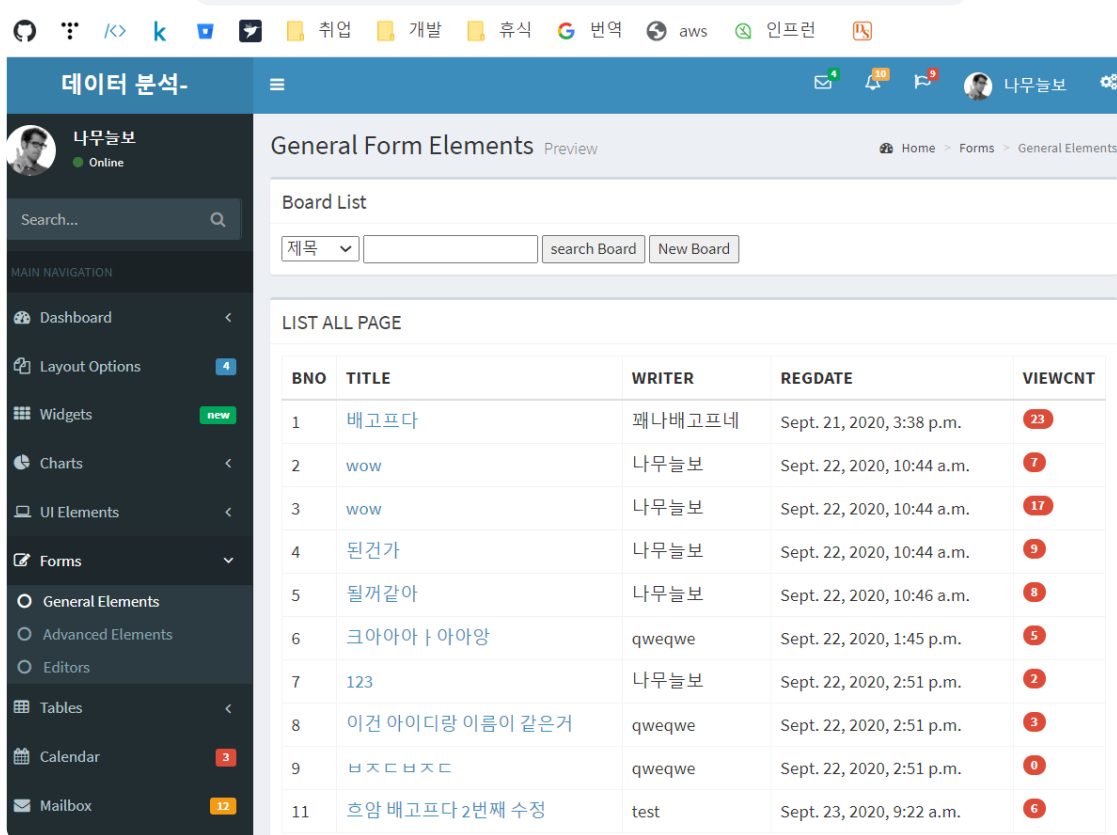
search를 구현하기 전에 Ajax의 개념에 대해 알고 갈 필요가 있다.

### Ajax란?

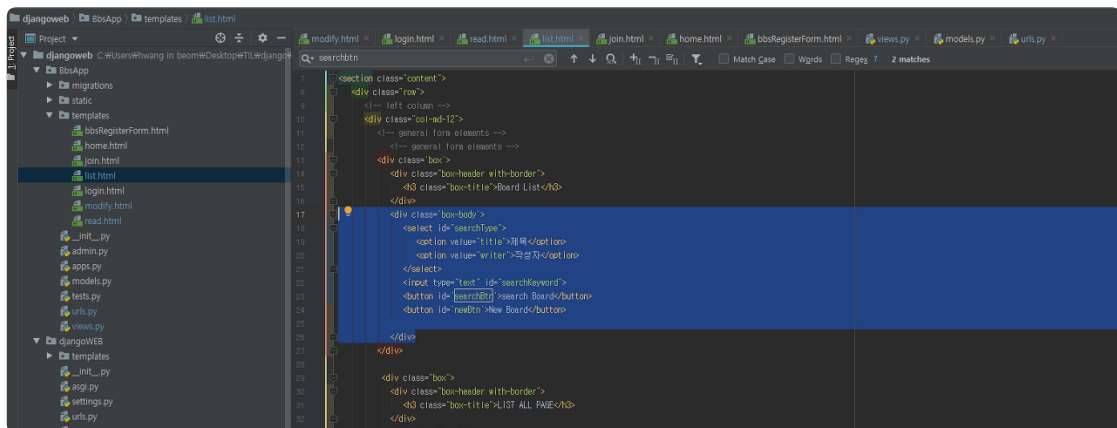
Ajax는 JavaScript의 라이브러리중 하나이며 Asynchronous Javascript And Xml(비동기식 자바스크립트와 xml)의 약자입니다. 브라우저가 가지고있는 XMLHttpRequest 객체를 이용해서 전체 페이지를 새로 고치지 않고도 페이지의 일부만을 위한 데이터를 로드하는 기법이며 Ajax를 한마디로 정의하자면 JavaScript를 사용한 비동기 통신, 클라이언트와 서버 간에 XML 데이터를 주고받는 기술이라고 할 수 있습니다.

즉 사용자의 요청에 따라 결과물을 페이지 전체를 로드하는게 아닌 부분만을 로드해 더 빠르게 응답하는 기능을 가진다. 그래서 비동기 자바스크립트라고 한다.

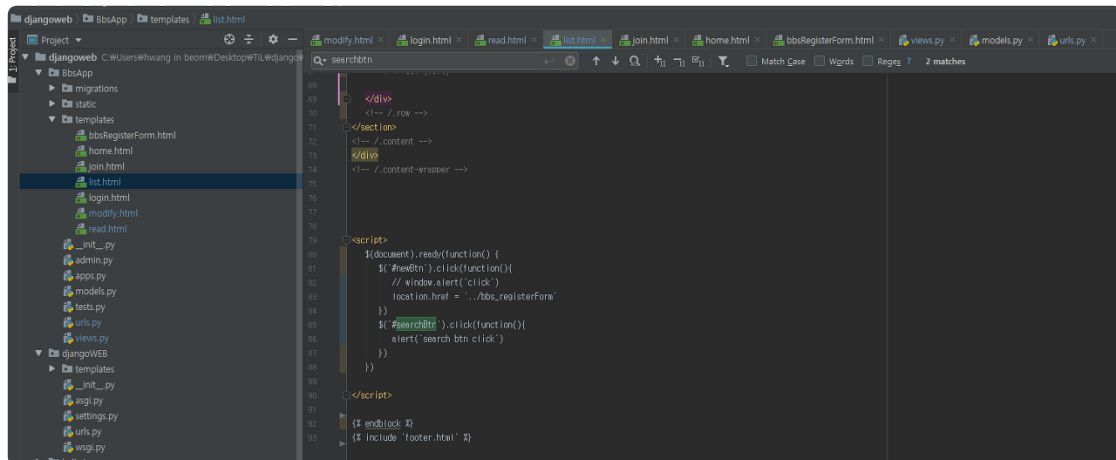
# search 구현



이런식의 검색바를 만들어 검색에 따라서 AJAX를 따라 밑에 값만 바뀌게 구현하겠다. 검색에 따라 페이지를 로드하기보다는 이것이 더 효율적이기 때문이다.

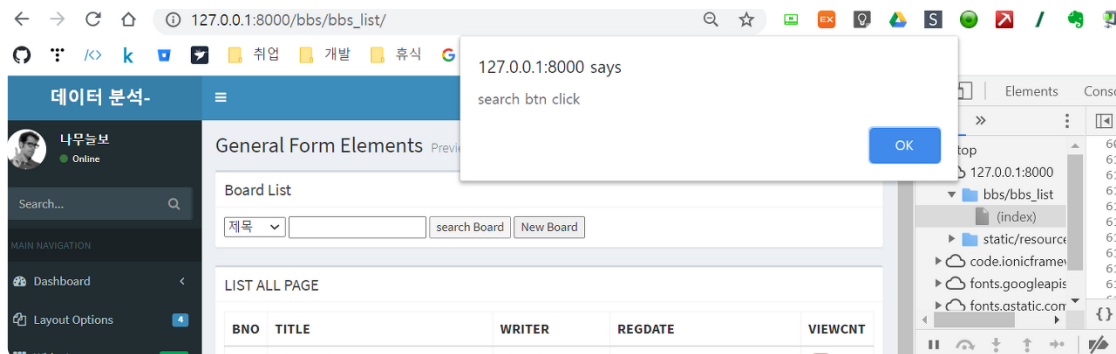


이런 식으로 template을 작성하고 이 id에 따라 밑에 스크립트 단에



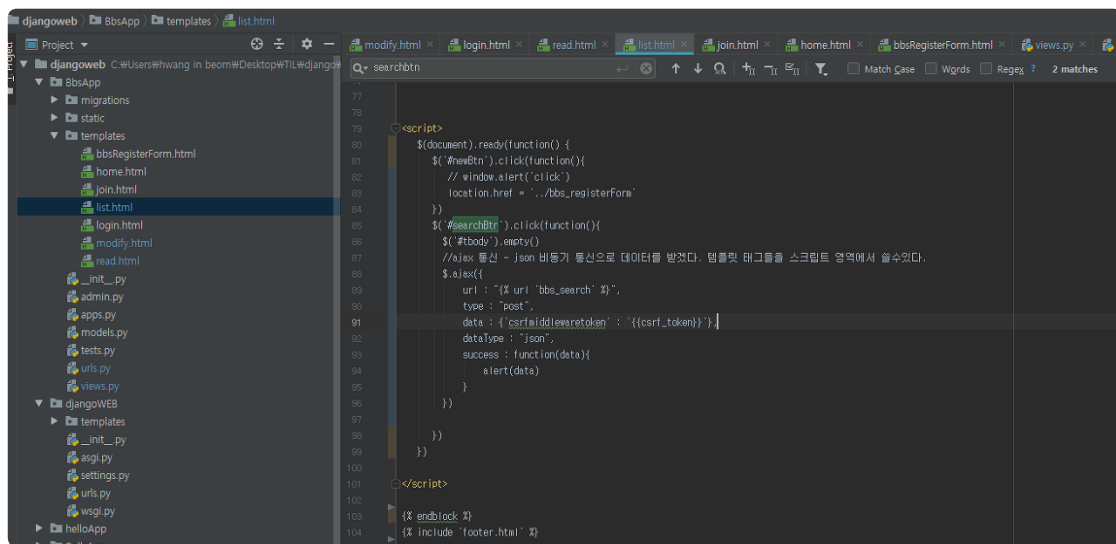
```
68 </div>
69
70 <!-- /.row -->
71 </section>
72 <!-- /.content -->
73 </div>
74 <!-- /.content-wrapper -->
75
76
77
78
79 <script>
80 $(document).ready(function() {
81   $('#searchBtn').click(function(){
82     // window.alert('click')
83     location.href = '../bbs_registerForm'
84   })
85   $('#searchBtn').click(function(){
86     alert('search btn click')
87   })
88 })
89 </script>
90
91 { { endblock 'content' }}
92 { { include 'footer.html' }}
```

해당 부분에 대한 로직을 만들어주자. 일단 첫 번째로 alert으로 해당 부분의 스크립트가 정상적으로 작동하는지 확인해보자.



일단 버튼을 눌렀을 때 작동이 된다.

1 차적으로 ajax 로 받아주는 작업을 진행해야 한다.



list.html에서 script 부분에서 \$. ajax라는 것을 만들고 해당 로직을 만들어준다.

버튼을 눌렀을 때 tbody를 초기화 하고 key / value형식으로

url : 이동하는 곳

type : get/post

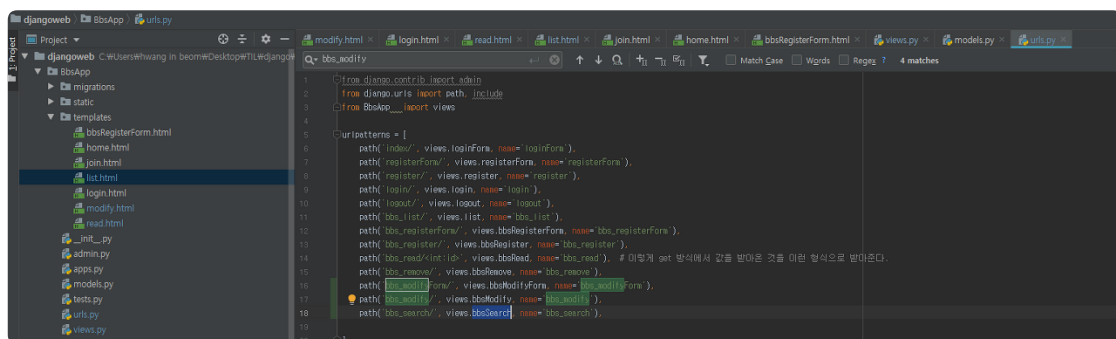
data : 넣는데이터 값(보안토큰을 넣어준다.)

dataType : 데이터 타입 여기서는 json 형태로 제공한다.

success : ajax가 실행될 때 구현되는 로직

이렇게 구성되어 있다.

이제 url 부분을 잡아준다.



이제 보내주는 url 부분에 대해서 urls.py을 구현하고

```
135 if board is not None:
136     content = {
137         'id': request.session['user_id'], # 이러한 방법도 가능
138         'name': request.session['user_name'],
139         'board': board
140     }
141     print('para = ', board.writer)
142     print(_id)
143     return render(request, 'read.html', context)
144
145 def bbsModify(request):
146     id = request.POST['id']
147     board = Bbs.objects.get(id=id) # 객체를 가져온다.
148     board.title = request.POST['title']
149     board.content = request.POST['content']
150     board.save() # 객체를 model에 저장한다.
151     return redirect('/bbs/list') # render로 할 경우에 데이터를 가져올 수가 없다, 그래서 redirect로 한다.
152
153 def bbsSearch(request):
154     print('ajax json bbsSearch')
155     dict = {'test': 'json_search'}
156     return HttpResponse(json.dumps(dict), content_type='application/json')
```

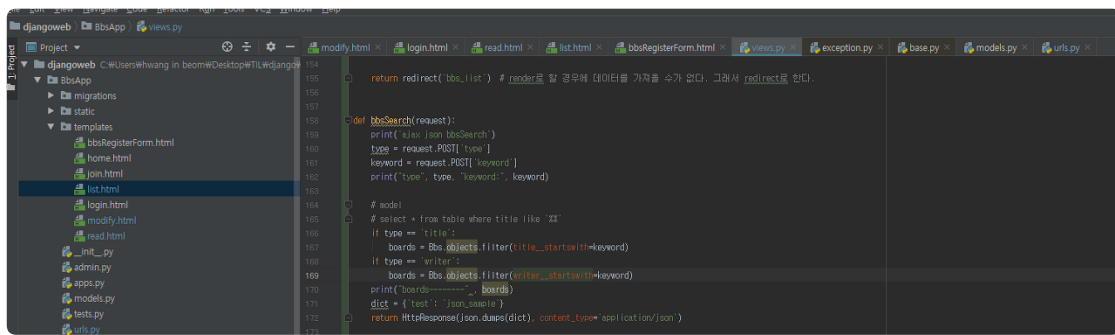
view를 구현한다 여기서는 1차적으로 print 값이 나오는 지를 확인한다.

```
77
78
79 <script>
80 $(document).ready(function() {
81     $('#newBtn').click(function(){
82         // window.alert('click')
83         location.href = '../bbs_registerForm'
84     })
85     $('#searchBtn').click(function(){
86         $('#tbody').empty()
87         //ajax 통신 - json 비동기 통신으로 데이터를 받겠다. 템플릿 태그들을 스크립트 영역에서 쓸수있다.
88         $.ajax({
89             url: "{% url 'bbs_search' %}",
90             type: "post",
91             data: {'csrfmiddlewaretoken': '{{csrf_token}}'},
92             dataType: "json",
93             success: function(data){
94                 alert(data.test)
95             }
96         })
97     })
98 })
99
100
```

여기서 success 부분은 정상적으로 됐을 때 작동하는 부분에 대해 로직을 다시 짠다.

이걸 가지고 데이터 값을 넣으려 하려면

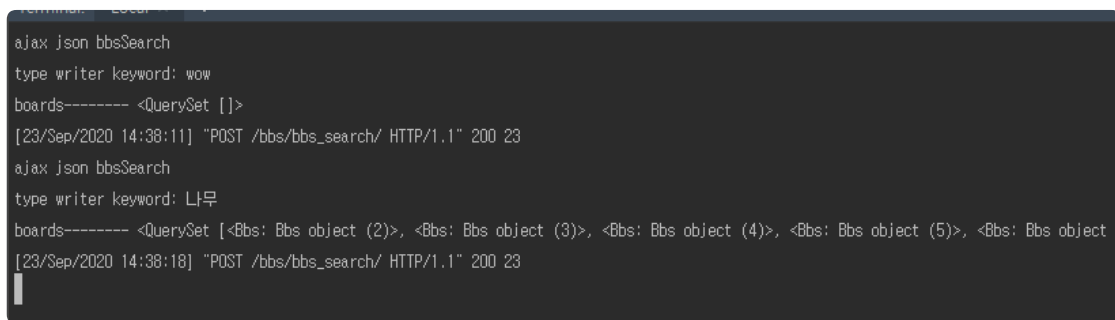




```
154 return HttpResponseRedirect(reverse('bbs:board_list')) # random 값 얻어서 데이터를 가져올 수가 없다. 그래서 HttpResponseRedirect 한다.
155
156
157
158 def bbsSearch(request):
159     print("ajax json bbsSearch")
160     type = request.POST['type']
161     keyword = request.POST['keyword']
162     print("type", type, "keyword", keyword)
163
164     # model
165     # select * from table where title like 'X'
166     if type == 'title':
167         boards = Bbs.objects.filter(title__startswith=keyword)
168     if type == 'writer':
169         boards = Bbs.objects.filter(writer__startswith=keyword)
170     print("boards-----", boards)
171     dict = {'test': 'json, sample'}
172     return JsonResponse(json.dumps(dict), content_type='application/json')
```

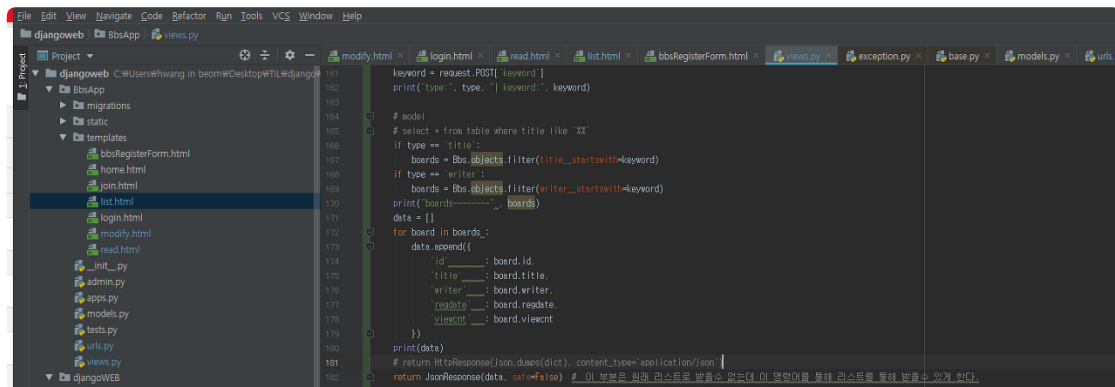
이제 검색을 받는 것에 대해 값을 찾는 것을 구현한다.

Bbs.objects.filter를 쓴 뒤 (칼럼명\_\_startswith = )를 통해 값을 가져온다.  
여기서 \_\_ 이거는 아래 두 개다



```
ajax json bbsSearch
type writer keyword: wow
boards----- <QuerySet []>
[23/Sep/2020 14:38:11] "POST /bbs/bbs_search/ HTTP/1.1" 200 23
ajax json bbsSearch
type writer keyword: 나무
boards----- <QuerySet [ <Bbs: Bbs object (2)>, <Bbs: Bbs object (3)>, <Bbs: Bbs object (4)>, <Bbs: Bbs object (5)>, <Bbs: Bbs object (6)> ]>
[23/Sep/2020 14:38:18] "POST /bbs/bbs_search/ HTTP/1.1" 200 23
```

위와 같은 결과가 나온다.



```
161 keyword = request.POST['keyword']
162 print("type", type, "keyword", keyword)
163
164 # model
165 # select * from table where title like 'X'
166 if type == 'title':
167     boards = Bbs.objects.filter(title__startswith=keyword)
168 if type == 'writer':
169     boards = Bbs.objects.filter(writer__startswith=keyword)
170 print("boards-----", boards)
171 data = []
172 for board in boards:
173     data.append({
174         'id': board.id,
175         'title': board.title,
176         'writer': board.writer,
177         'regdate': board.regdate,
178         'viewcnt': board.viewcnt
179     })
180 print(data)
181 # return JsonResponse(json.dumps(dict), content_type='application/json')
182 return JsonResponse(data, safe=False) # 이 부분은 원래 리스트로 보낼 수 있는데 이 몇몇이름 통해 리스트를 통해 보낼 수 있게 한다.
```

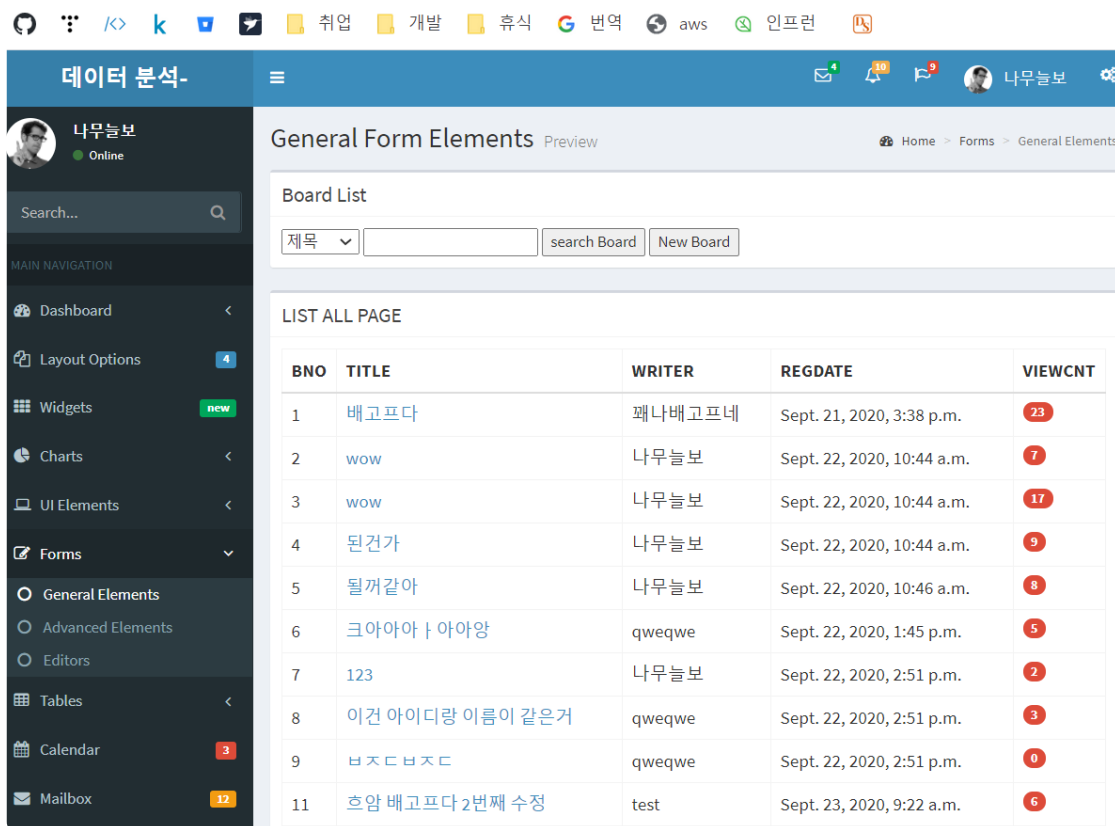
멀쩡히 나오는 걸 확인했으니 이제 객체를 하나씩 빼서 data라는 리스트에 넣어주고 이것을 JsonResponse로 넣어서 보내준다. 이렇게 보내주게 되면 해당 ajax로 이동하게 된다.

```

83 location.href = '../bbs_registerForm'
84 })
85 $( '#searchBtn' ).click(function(){
86     <!--alert( $( '#searchKeyword' ).val() )-->
87     <!--alert( $( '#searchType' ).val() )-->
88     $( '#body' ).empty()
89     //ajax 통신 - json 비동기 통신으로 데이터를 받겠다. 성공한 태그들을 스크립트 영역에서 쓸수있다. 스크립트에서는 오브젝트 형태로 받아야 쓸 수 있다.
90     $.ajax({
91         url : '{% url "bbs_search" %}',
92         type : 'post',
93         data : { 'csrfmiddlewaretoken' : '{{ csrf_token }}',
94             type : $( '#searchType' ).val(),
95             keyword : $( '#searchKeyword' ).val()
96         },
97         dataType : 'json',
98         success : function(data){
99             var txt = ""
100             $.each(data, function(id, obj) {
101                 txt += "<tr><td>"+obj.id+"</td>";
102                 txt += "<td>"+obj.title+"</td>";
103                 txt += "<td>"+obj.writer+"</td>";
104                 txt += "<td>"+obj.regdate+"</td>";
105                 txt += "<td>"+obj.viewcnt+"</td></tr>";
106             });
107             $( '#body' ).append(txt);
108         })
109     })
110 })
111 })
112 })
113 })

```

마지막으로 ajax 부분에서 success 부분을 구현한다. 위와 같이 구현하면



검색어 입력에 따라 원하는 결과가 나오게 된다.

참조 : [coding-factory.tistory.com/143](https://coding-factory.tistory.com/143)



## '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

Djagno 검색

Djagno 검색기능

django search



나무늘보스

혼자 끄적끄적하는 블로그 입니다.