

# DjangoでListViewを用いて検索画面を実装する方法

---

DjangoでListViewを用いて検索画面を実装する方法をご紹介します。

## 目次 [非表示]

- 1 条件
- 2 検索画面の実装
  - 2.1 構成
  - 2.2 models.py
  - 2.3 urls.py
  - 2.4 forms.py
  - 2.5 views.py
  - 2.6 base.html
  - 2.7 index.html
  - 2.8 pagination.html
- 3 実行結果
  - 3.1 DBレコード
  - 3.2 検索の例
    - 3.2.1 初期画面
    - 3.2.2 検索フォーム入力なしで検索
    - 3.2.3 タイトルのみ指定
    - 3.2.4 タイトルと内容の指定
- 4 サンプルソース
- 5 参考
  - 5.1 stackoverflow

---

## 条件

---

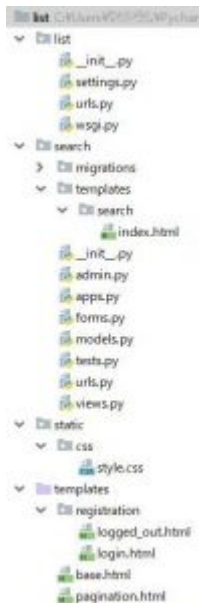
- Django 2.1.7
  - Python 3.7.0
- 

## 検索画面の実装

---

# 構成

ソースの構成は以下の通りです。



以下、ポイントとなるソースをピックアップします。

ソースの詳細はGitHubをご参照ください。

<https://github.com/kzmrt/list>

## models.py

以下のようなモデルを想定します。

```
from django.db import models
from django.urls import reverse
from django.contrib.auth.models import AbstractUser
class CustomUser(AbstractUser):
    def __str__(self):
        return self.username + ":" + self.email
class Post(models.Model):
    """投稿モデル"""
    class Meta:
        db_table = 'post'
    title = models.CharField(verbose_name='タイトル', max_length=255)
    text = models.CharField(verbose_name='内容', max_length=255, default='', blank=True)
    author = models.ForeignKey(
        'search.CustomUser',
        on_delete=models.CASCADE,
    )
    created_at = models.DateTimeField(verbose_name='登録日時', auto_now_add=True)
    updated_at = models.DateTimeField(verbose_name='更新日時', auto_now=True)
```

```
def __str__(self):
    return self.title + ',' + self.text
@staticmethod
def get_absolute_url(self):
    return reverse('search:index')
```

## urls.py

今回はトップ画面（検索画面）のみ作成します。

```
from django.urls import path
from . import views
app_name = 'search'
urlpatterns = [
    # トップ画面
    path('', views.IndexView.as_view(), name='index'),
]
```

## forms.py

カスタムユーザモデルに加えて、検索用のフォームも定義します。

```
from django.contrib.auth.forms import UserCreationForm, UserChangeForm
from .models import CustomUser
from django import forms
class CustomUserCreationForm(UserCreationForm):
    class Meta(UserCreationForm):
        model = CustomUser
        fields = ('username', 'email')
class CustomUserChangeForm(UserChangeForm):
    class Meta:
        model = CustomUser
        fields = ('username', 'email')
class SearchForm(forms.Form):
    title = forms.CharField(
        initial='',
        label='タイトル',
        required = False, # 必須ではない
    )
    text = forms.CharField(
        initial='',
        label='内容',
        required=False, # 必須ではない
    )
```

## views.py

post受け取り時の処理、フォームの配置、クエリ発行について処理を記述します。

処理のポイントは以下の通りです。

- def post()でセッションに検索フォームの値を渡す。
- def get\_context\_data()でセッションから検索フォームの値を取得して、検索フォームの初期値としてセットする。
- def get\_queryset()でセッションから取得した検索フォームの値に応じてクエリ発行を行う。
- def post()で検索時にページネーションに関連したエラーを防ぐ処理を記述。

```
import logging
from django.contrib.auth.mixins import LoginRequiredMixin
from django.views import generic
from .models import Post
from .forms import SearchForm
from django.db.models import Q
logger = logging.getLogger('development')
class IndexView(LoginRequiredMixin, generic.ListView):
    paginate_by = 5
    template_name = 'search/index.html'
    model = Post
    def post(self, request, *args, **kwargs):
        form_value = [
            self.request.POST.get('title', None),
            self.request.POST.get('text', None),
        ]
        request.session['form_value'] = form_value
        # 検索時にページネーションに関連したエラーを防ぐ
        self.request.GET = self.request.GET.copy()
        self.request.GET.clear()
        return self.get(request, *args, **kwargs)
    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        # sessionに値がある場合、その値をセットする。(ページングしてもform値が変わらないように)
        title = ''
        text = ''
        if 'form_value' in self.request.session:
            form_value = self.request.session['form_value']
            title = form_value[0]
            text = form_value[1]
        default_data = {'title': title, # タイトル
                       'text': text, # 内容
                       }
        test_form = SearchForm(initial=default_data) # 検索フォーム
        context['test_form'] = test_form
        return context
    def get_queryset(self):
        # sessionに値がある場合、その値でクエリ発行する。
        if 'form_value' in self.request.session:
            form_value = self.request.session['form_value']
            title = form_value[0]
            text = form_value[1]
            # 検索条件
            condition_title = Q()
            condition_text = Q()
```

```

    if len(title) != 0 and title[0]:
        condition_title = Q(title__icontains=title)
    if len(text) != 0 and text[0]:
        condition_text = Q(text__contains=text)
    return Post.objects.select_related().filter(condition_title & condition_text)
else:
    # 何も返さない
    return Post.objects.none()

```

## base.html

bootstrap4を用いてデザインの設定を行います。  
(別途、style.cssを追加してデザイン調整を行っています。)

```

{% load staticfiles %}
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- bootstrap4 -->
    {% load bootstrap4 %}
    {% bootstrap_css %}
    {% bootstrap_javascript jquery='full' %}
    <!-- bootstrap4 -->
    <link href="https://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet">
    <link href="{% static 'css/style.css' %}" rel="stylesheet">
    <script type="text/javascript">
      $(function() {
        var topBtn = $('#page-top');
        topBtn.hide();
        //スクロールが500に達したらボタン表示
        $(window).scroll(function () {
          if ($(this).scrollTop() > 500) {
            topBtn.fadeIn();
          } else {
            topBtn.fadeOut();
          }
        });
        //スクロールしてトップ
        topBtn.click(function () {
          $('body,html').animate({
            scrollTop: 0
          }, 500);
          return false;
        });
      });
    </script>
    <title>{% block title %}Search{% endblock %}</title>
  </head>
  <body>
    <!-- ナビゲーションバーの設定 -->
    <nav class="navbar navbar-expand-md navbar-dark bg-dark sticky-top">
      <div class="container">
        <a class="navbar-brand" href="{% url 'search:index' %}">Search</a>

```

```

        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#n
controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarResponsive">
        <ul class="navbar-nav ml-auto">
            {% if user.is_authenticated %}
                <li class="nav-item">
                    <a class="nav-link" href="{% url 'search:index' %}">TOP</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="{% url 'logout' %}">ログアウト</a>
                </li>
            {% else %}
                {% url 'login' as login %}
                {% ifnotequal request.path login %}
                    <li class="nav-item">
                        <a class="nav-link" href="{% url 'login' %}" class="login">ログイン<
                    </li>
                {% endifnotequal %}
            {% endif %}
        </ul>
    </div>
</div>
</nav>
<main>
    {% block content %}
    {% endblock %}
</main>
<p id="page-top"><a href="#">PAGE TOP</a></p>
<footer class="py-4 bg-dark">
    <div class="container text-center">
        <p class="text-light"><small>Copyright &copy;2019 Search, All Rights Reserved.<
    </div>
</footer>
</body>
</html>

```

## index.html

検索フォームおよび、検索結果の表示について記述します。

```

{% extends 'base.html' %}
{% block content %}
<div class="col-lg-6 offset-lg-3">
    <h1>検索条件</h1>
    <form method="POST">
        {% csrf_token %}
        {% for field in test_form %}
            <div class="form-group form-inline">
                <label class="col-md-offset-2 col-md-3 control-label">{{ field.label }}:</label>
                <div class="col-md-8">
                    {{ field }}
                </div>
            </div>
        </div>
    </form>
{% endfor %}

```

```

        <input class="btn btn-success offset-md-8 col-md-3" type="submit" id="button" name="button"
    </form>
    <h1>検索結果</h1>
    <section class="post-list">
        {% if object_list|length == 0 %}
            <p>検索結果が存在しません。</p>
        {% else %}
            <table class="table table-hover table-bordered">
                <tr>
                    <th>タイトル</th>
                    <th>内容</th>
                </tr>
                <tbody>
                    {% for post in object_list %}
                        <tr>
                            <td width="35%">{{ post.title }}</td>
                            <td width="65%">{{ post.text }}</td>
                        </tr>
                    {% endfor %}
                </tbody>
            </table>
        {% endif %}
    </section>
    <div class="col-6 offset-3 justify-content-center">
        {% if is_paginated %}
            {% include 'pagination.html' %}
        {% endif %}
    </div>
</div>
{% endblock %}

```

## pagination.html

ページネーションはindex.htmlで読み込みます。

```

<div class="container">
    <div class="row">
        <div class="col-md-6 offset-md-3 py-2">
            <nav aria-label="ページ送り">
                <div class="text-center">
                    <ul class="pagination justify-content-center">
                        {% if page_obj.has_previous %}
                            <li class="page-item"><a class="page-link" href="?page={{ page_obj.prev
                        </li>

                        {% else %}
                            <li class="page-item disabled"><span class="page-link">&laquo;</span></li>
                        {% endif %}
                        {% for i in page_obj.paginator.page_range %}
                            {% if page_obj.number == i %}
                                <li class="page-item active" aria-current="page">
                                    <a class="page-link" href="#">{{ i }} <span class="sr-only">(現
                                </li>
                            {% else %}
                                <li class="page-item"><a class="page-link" href="?page={{ i }}">{{
                            {% endif %}
                        {% endfor %}
                    </ul>
                </div>
            </nav>
        </div>
    </div>
</div>

```

```

        {% if page_obj.has_next %}
            <li class="page-item"><a class="page-link" href="?page={{ page_obj.next
</li>

        {% else %}
            <li class="page-item disabled"><span class="page-link">&raquo;</span></
        {% endif %}
    </ul>
</div>
</nav>
</div>
</div>
</div>

```

## 実行結果

## DBレコード

POSTテーブルには、以下のように12レコード存在するものとします。

	id	title	text	created_at	updated_at	author_id
Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	test1	11111	2019-02-27 ...	2019-02-27 ...	1
2	2	test2	22222	2019-02-27 ...	2019-02-27 ...	1
3	3	test3	33333	2019-02-27 ...	2019-02-27 ...	1
4	4	test4	44444	2019-02-27 ...	2019-02-27 ...	1
5	5	test5	55555	2019-02-27 ...	2019-02-27 ...	1
6	6	test6	66666	2019-02-27 ...	2019-02-27 ...	1
7	7	test7	77777	2019-02-27 ...	2019-02-27 ...	1
8	8	test8	88888	2019-02-27 ...	2019-02-27 ...	1
9	9	test9	99999	2019-02-27 ...	2019-02-27 ...	1
10	10	test10	1010101010	2019-02-27 ...	2019-02-27 ...	1
11	11	test11	1111111111	2019-02-27 ...	2019-02-27 ...	1
12	12	test12	1212121212	2019-02-27 ...	2019-02-27 ...	1

## 検索の例

## 初期画面



Search

検索 | 設定 | ヘルプ

検索条件

キーワード:

内容:

検索

検索結果

検索結果が作成されました。

Copyright ©2020 Search, All Rights Reserved.

## 検索フォーム入力なしで検索

Search

検索 | 設定 | ヘルプ

検索条件

キーワード:

内容:

検索

検索結果

タイトル	内容
test01	01111
test02	02222
test03	03333
test04	04444
test05	05555

1

2

3

4

5

Copyright ©2020 Search, All Rights Reserved.

## タイトルのみ指定

Search

検索 | 設定 | ヘルプ

検索条件

キーワード:

test01

内容:

検索

検索結果

タイトル	内容
test01	01111
test02	022220000
test03	033331111
test04	044441111

Copyright ©2020 Search, All Rights Reserved.

## タイトルと内容の指定

Search

検索 | 設定 | ヘルプ

検索条件

検索条件

検索条件

検索条件

検索

検索結果

タイトル	内容
test02	test021102

Copyright ©2020 Search. All Rights Reserved.

---

# サンプルソース

---

GitHubに当該記事のサンプルソースを公開しています。

<https://github.com/kzmrt/list>