

## Django 3.2智能分页get\_elided\_page\_range方法初尝!

原创 大江狗 Python Web与Django开发 4月21日

收录于话题

#Django实战 38 #Django 36

Django 3.2版本本月正式发布了, 其中一个重要变化就是新增了智能分页`get_elided_page_range`方法。Django 3.2之前的Paginator类将连续输出所有页码, 非常不智能。本文将演示如何使用`get_elided_page_range`方法实现智能分页。



### get\_elided\_page\_range方法解读

该方法接收3个参数, 第一个参数`number`为当前页码数, `on_each_side`为当前页码左右两边的页数, `on_ends`为首尾页码范围。

```
Paginator.get_elided_page_range(number, on_each_side=3, on_ends=2)
```

如果当前页码为第10页, 使用该方法将输出如下页码范围(`page_range`)。10左右两边各有3页, 首尾各有2页, 其余页码码用...代替。将`page_range`这个变量传递到前端模板进行遍历即可实现智能分页。

```
[1, 2, '...', 7, 8, 9, 10, 11, 12, 13, '...', 49, 50]
```

### Django 3.2之前版本非智能分页

我们有个House模型。为了模拟总页数非常多的情形，我们将每页记录数量设为1条。视图函数如下所示：

```

1 # Filter houses
2 def house_filter(request):
3     qs = House.objects.all()
4     paginator = Paginator(qs, 1) # 每页1条记录
5     page = request.GET.get('page', 1) # 获取当前page页码，默认为1
6     try:
7         page_obj = paginator.page(page) # 分页
8     except PageNotAnInteger:
9         page_obj = paginator.page(1)
10    except EmptyPage:
11        page_obj = paginator.page(paginator.num_pages)
12    is_paginated = True if paginator.num_pages > 1 else False
13
14    context = {'page_obj': page_obj, 'paginator': paginator,
15              'is_paginated': is_paginated,}
16    return render(request, 'house/house_index.html', context)
17

```

假如前端模板样式用bootstrap 4，对应分页的前端模板如下所示：

```

1 # house/house_index.html
2 {% block content %}
3     <div class="table-responsive">
4         <table class="table table-striped table-hover">
5             <thead>
6                 <tr>
7                     <th scope="col">描述</th>
8                     <th scope="col">小区</th>
9                     <th scope="col">城市</th>
10                    <th scope="col">房型</th>
11                    <th scope="col">朝向</th>
12                    <th scope="col">面积</th>
13                    <th scope="col">价格(万元)</th>
14                </tr>
15            </thead>
16            <tbody>
17                {% if page_obj %}
18                    {% for item in page_obj %}
19                        <tr>
20                            <td>{{ item.description }}</td>
21                            <td>{{ item.community }}</td>
22                            <td>{{ item.community.get_city_display }}</td>
23                            <td>{{ item.get_bedroom_display }}</td>
24                            <td>{{ item.get_direction_display }}</td>
25                            <td>{{ item.area }}</td>
26                            <td>{{ item.price }}</td>
27                        </tr>
28                    {% endfor %}
29                {% endif %}
30            </tbody>
31        </table>
32
33        {% if is_paginated %}
34            <ul class="pagination">
35                {% if page_obj.has_previous %}

```

```
36         <li class="page-item"><a class="page-link" href="?page={{ page_obj.previous_page_number }}">Previous<
37     {% else %}
38         <li class="page-item disabled"><span class="page-link">Previous</span></li>
39     {% endif %}
40
41     # 注意下面这行
42     {% for i in page_obj.paginator.page_range %}
43         {% if page_obj.number == i %}
44             <li class="page-item active"><span class="page-link"> {{ i }} <span class="sr-only">(current)</span><
45         {% else %}
46             <li class="page-item"><a class="page-link" href="?page={{ i }}">{{ i }}</a></li>
47         {% endif %}
48     {% endfor %}
49
50     {% if page_obj.has_next %}
51         <li class="page-item"><a class="page-link" href="?page={{ page_obj.next_page_number }}">Next</a></li>
52     {% else %}
53         <li class="page-item disabled"><span class="page-link">Next</span></li>
54     {% endif %}
55 </ul>
56 {% endif %}
57
58 </div>
59 </div>
60 </div>
61 {% endblock %}
```

分页效果如下所示，所有页码都展示了，非常的长，显然不是我们想要的。



本例所使用House模型及完整样式代码见Django 3.0实战: 仿链家二手房信息查询网(附GitHub源码)。如果你还不知道如何在Django函数视图和基于类的视图中使用分页，请阅读下篇文章(点击阅读原文即可跳转)。

o <https://pythondjango.cn/django/basics/14-pagination/>

现在我们就要对前面的视图和模板代码做出修改，实现智能分页。

Django 3.2智能分页

首先我们修改视图函数，使用get\_elided\_page\_range方法获取智能分页范围(page\_range)，并将这个变量传递给前端模板。

```
1 # Filter houses
2 def house_filter(request):
3     qs = House.objects.all()
4     paginator = Paginator(qs, 1) # 每页1条记录
5     page = request.GET.get('page', 1) # 获取当前page页码，默认为1
6     try:
7         page_obj = paginator.page(page) # 分页
8     except PageNotAnInteger:
9         page_obj = paginator.page(1)
10    except EmptyPage:
11        page_obj = paginator.page(paginator.num_pages)
```

```

12 is_paginated = True if paginator.num_pages > 1 else False
13 # 注意下面一行, 获取智能页码范围, 并传递给模板
14 page_range = paginator.get_elided_page_range(page, on_each_side=3, on_ends=2)
15 context = {'page_obj': page_obj, 'paginator': paginator,
16            'is_paginated': is_paginated, 'page_range': page_range }
17 return render(request, 'house/house_index.html', context)

```

模板中我们只需要修改一行代码, 如下所示:

```

1 {% block content %}
2     <div class="table-responsive">
3         <table class="table table-striped table-hover">
4             <thead>
5                 <tr>
6                     <th scope="col">描述</th>
7                     <th scope="col">小区</th>
8                     <th scope="col">城市</th>
9                     <th scope="col">房型</th>
10                    <th scope="col">朝向</th>
11                    <th scope="col">面积</th>
12                    <th scope="col">价格(万元)</th>
13                </tr>
14            </thead>
15            <tbody>
16                {% if page_obj %}
17                    {% for item in page_obj %}
18                        <tr>
19                            <td>{{ item.description }}</td>
20                            <td>{{ item.community }}</td>
21                            <td>{{ item.community.get_city_display }}</td>
22                            <td>{{ item.get_bedroom_display }}</td>
23                            <td>{{ item.get_direction_display }}</td>
24                            <td>{{ item.area }}</td>
25                            <td>{{ item.price }}</td>
26                        </tr>
27                    {% endfor %}
28                {% endif %}
29            </tbody>
30        </table>
31
32        {% if is_paginated %}
33            <ul class="pagination">
34                {% if page_obj.has_previous %}
35                    <li class="page-item"><a class="page-link" href="?page={{ page_obj.previous_page_number }}">Previous<
36                {% else %}
37                    <li class="page-item disabled"><span class="page-link">Previous</span></li>
38                {% endif %}
39
40                # 注意下面一行代码
41                {% for i in page_range %}
42                    {% if page_obj.number == i %}
43                        <li class="page-item active"><span class="page-link"> {{ i }} <span class="sr-only">(current)</span><
44                    {% else %}
45                        <li class="page-item"><a class="page-link" href="?page={{ i }}">{{ i }}</a></li>
46                    {% endif %}
47                {% endfor %}
48

```

2021/9/3Django 3.2智能分页get\_elided\_page\_range方法初尝!

```
49         {% if page_obj.has_next %}
50             <li class="page-item"><a class="page-link" href="?page={{ page_obj.next_page_number }}">Next</a></li>
51         {% else %}
52             <li class="page-item disabled"><span class="page-link">Next</span></li>
53         {% endif %}
54     </ul>
55     {% endif %}
56 </div>
57 </div>
58 </div>
59 {% endblock %}
```

如果你还没注意模板中代码的变化，请再观察一下：

```
1 # Django 3.2之前
2 {% for i in page_obj.paginator.page_range %}
3 # Django 3.2之后
4 {% for i in page_range %}
```

新的展示效果是不是帅多了？



本例仅展示了如何在函数视图中使用智能分页。如果你希望在基于类的视图中也使用智能分页，你可以重写get\_context\_data方法向模板传递智能页码范围page\_range即可。

大江狗  
2021.4

相关阅读

- Django 3.2正式发布! 主要变化抢先看!
- Django 3.0实战: 仿链家二手房信息查询网(附GitHub源码)
- Django代码分享: 可以重用的Bootstrap 4分页模板

收录于话题 #Django实战38个

上一篇

Django实战: 使用django-filter和django-tables2打造美观的管理后台(附图)

下一篇

Django实战: 使用通用类视图开发任务管理CRUD小应用(附GitHub源码)

阅读原文

喜欢此内容的人还喜欢

Django数据导入导出神器django-import-export使用教程  
Python Web与Django开发

乡村振兴最难在哪，这份调查搞清楚了  
瞭望智库