Dajngo Views

----

1. Create a list API with limit offset pagination.

2. Add search, filters(minimum two) and sort (asc&desc).
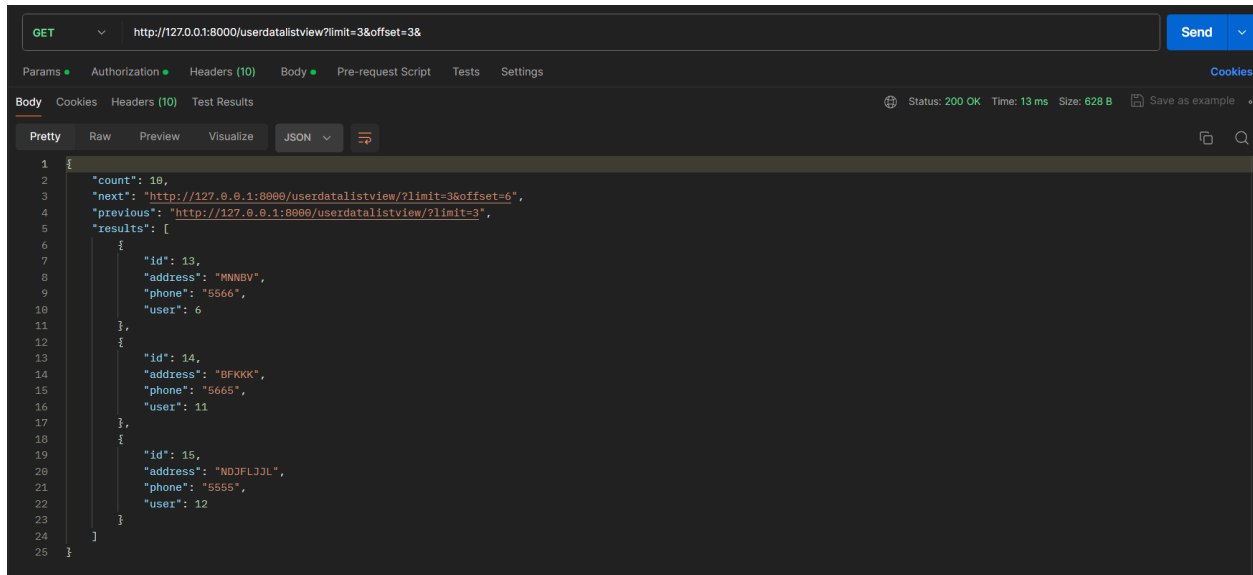
View.py

```python
class UserDataListview(generics.ListAPIView):
    serializer_class = UserDataSerializer

    def get_queryset(self):
        q = self.request.GET.get('q')
        queryset =UserData.objects.all()
        if q :
            queryset =UserData.objects.filter(address__icontains=q)

        p = self.request.GET.get('p')
        if p:
            queryset = queryset.filter(phone=p)

        ordering = self.request.GET.get('ordering')
        if ordering:
            if ordering == 'asc':
                queryset = queryset.order_by('id')
            elif ordering == 'desc':
                queryset = queryset.order_by('-id')
        return queryset
```
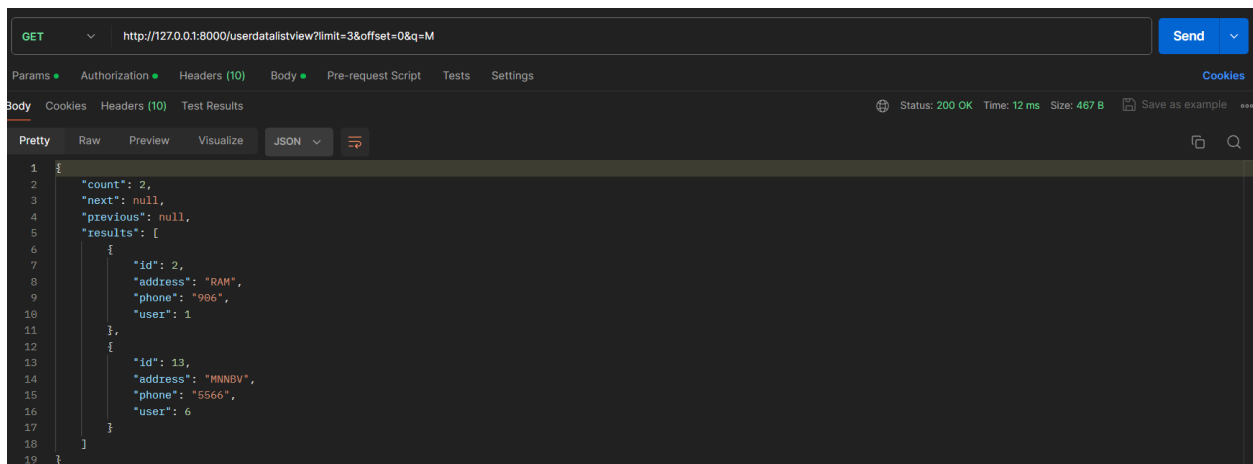
Pagenations

GET | http://127.0.0.1:8000/userdatalistview?limit=3&offset=3& | Send

Params • Authorization • Headers (10) Body • Pre-request Script Tests Settings Cookies

Body Cookies Headers (10) Test Results · Status: 200 OK Time: 13 ms Size: 628 B Save as example

Pretty Raw Preview Visualize JSON

```
1  {
2      "count": 10,
3      "next": "http://127.0.0.1:8000/userdatalistview/?limit=3&offset=6",
4      "previous": "http://127.0.0.1:8000/userdatalistview/?limit=3",
5      "results": [
6          {
7              "id": 13,
8              "address": "MNNBV",
9              "phone": "5566",
10             "user": 6
11         },
12         {
13             "id": 14,
14             "address": "BFKKK",
15             "phone": "5665",
16             "user": 11
17         },
18         {
19             "id": 15,
20             "address": "NDJFLJJL",
21             "phone": "5555",
22             "user": 12
23         }
24     ]
25 }
```

Search

GET | http://127.0.0.1:8000/userdatalistview?limit=3&offset=0&q=M | Send

Params • Authorization • Headers (10) Body • Pre-request Script Tests Settings Cookies

Body Cookies Headers (10) Test Results · Status: 200 OK Time: 12 ms Size: 467 B Save as example

Pretty Raw Preview Visualize JSON

```
1  {
2      "count": 2,
3      "next": null,
4      "previous": null,
5      "results": [
6          {
7              "id": 2,
8              "address": "RAM",
9              "phone": "906",
10             "user": 1
11         },
12         {
13             "id": 13,
14             "address": "MNNBV",
15             "phone": "5566",
16             "user": 6
17         }
18     ]
19 }
```

Filter

GET | http://127.0.0.1:8000/userdatalistview?limit=3&offset=0&q=MN&P=5566 | Send

Params ● | Authorization ● | Headers (10) | Body ● | Pre-request Script | Tests | Settings | Cookies

none | form-data | x-www-form-urlencoded | ● raw | binary | GraphQL | JSON | Beautify

```
1  {
2      "username": "newadmin",
3      "password": "Albin@123"
4  }
```

Body | Cookies | Headers (10) | Test Results | Status: 200 OK | Time: 13 ms | Size: 419 B | Save as example

Pretty | Raw | Preview | Visualize | JSON

```
1  {
2      "count": 1,
3      "next": null,
4      "previous": null,
5      "results": [
6          {
7              "id": 13,
8              "address": "MNNBV",
9              "phone": "5566",
10             "user": 6
11         }
12     ]
13 }
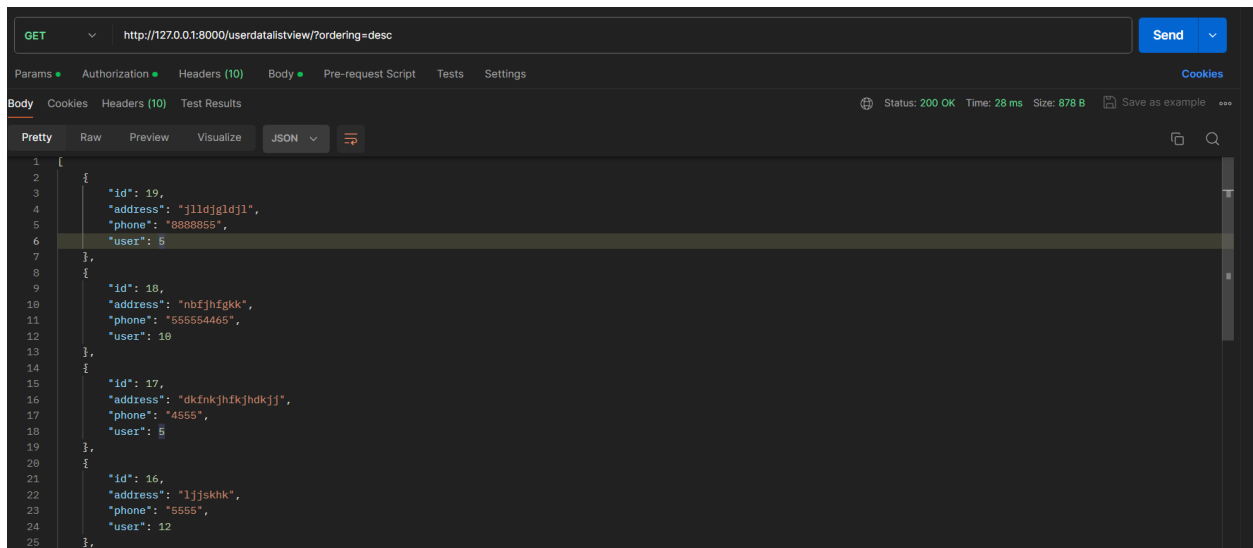```

## Sort Ascending

GET | http://127.0.0.1:8000/userdatalistview/?ordering=asc | Send

Params ● | Authorization ● | Headers (10) | Body ● | Pre-request Script | Tests | Settings | Cookies

Body | Cookies | Headers (10) | Test Results | Status: 200 OK | Time: 10 ms | Size: 878 B | Save as example

Pretty | Raw | Preview | Visualize | JSON

```
1  [
2      {
3          "id": 1,
4          "address": "Abcdf",
5          "phone": "987456123",
6          "user": 1
7      },
8      {
9          "id": 2,
10         "address": "RAM",
11         "phone": "906",
12         "user": 1
13     },
14     {
15         "id": 12,
16         "address": "ABCD",
17         "phone": "1234567890",
18         "user": 1
19     },
20     {
21         "id": 13,
22         "address": "MNNBV",
23         "phone": "5566",
24         "user": 6
25     },
```

Sort Descending

https://github.com/Albin-Kuriachan/python_beniex/tree/main/17.04.2024