

Displaying data retrieved through GraphQL in a Django website

JeremyG

Jun 2021

(Disclaimer : I'm just getting started with Django, and with web dev in general)

I have a backend app that stores different kinds of resources. Some are public and some are private. The application is accessible only to identified users. A GraphQL API allows me to access the resources.

On another server, I'd like to create a website that will be accessible to everyone. I want to use Django to create it.

The website will display a list of resources tagged as "public" in the backend app, with a pagination system and, say, 20 resources by page. The CSS will differ from the backend app and there will be a search section.

From what I understand, I should be able to retrieve the data through the GraphQL API, but I'm a bit confused here. All the documentation and tutos I can find about Django and GraphQL seem to be about setting up a GraphQL API server with Django. All I want to do is to build custom queries and to display them on my different html pages.

How can I do that? Where should I start?

Any hint or direction will be much appreciated, thank you.

✅ Solved by [KenWhitesell](#) in [post #13](#)

Yes. 😊 (Ok, not the most helpful answer, but I couldn't resist.) You will be doing some of the iteration in your template, so it's likely going to be a mix of both. The same way you pass any data from a view into the rendering engine for the template - in the context. That is really a personal...

KenWhitesell 🛡️

Jun 2021

Before you tackle something like this - have you worked your way through either (or both) of the [Official Django Tutorial](#) , or the [Django Girls Tutorial](#) ? If you haven't, that's the place to start. Either (or both) of them are going to provide some very important foundational knowledge on how to pass a request to return a response.

[Skip to main content](#)

If you have, and are progressing beyond that point, then what you're asking can be logically divided into two separate and distinct steps.

- Request from the “public”, get data (see second bullet), return HTML-formatted response.
- Get the data from the GraphQL API.

Logically, you should probably separate these two steps in your mind - it should make it easier. You can work on them in either order, but if I were doing this, I'd work on them in the order listed. First, I'd create the basic page that accepts whatever input the user will provide, call some function, and display the returned data. That function doesn't need to *do* anything yet, just return some data that can be displayed.

Then, I'd work on the function that calls the GraphQL API and retrieves the results. (Probably using the [requests](#) module.) What's important to understand here is that this step has absolutely **nothing** to do with Django. You can write this as a separate Python script and run it independently of *anything* else.

Once you have them both working, then you can replace your test function written at first with the real function to make your query.

last visit

JeremyG

Jun 2021

Hi Ken,

Thanks for your answer!

I did spend some time on the official tutorial already - not as exhaustively as I should though, but having a real project to work on makes the learning process easier to me. I still go back to the tutorial pages a lot and I'll probably catch up later with the rest of it.

Regarding my issue, I get what you're advising me to do with the two different steps. I didn't work on the first step yet (saw your post after 😊) but achieved to get the data from the API with what is for now a simple Python script. I know we're on the Django forum but posting it here for information:

```
import requests
import json
url = 'https://fakeurl.com/graphql/'
query = """{ resources { edges { node { id uri creatorPerson { firstName lastName } } } } }"""
r = requests.post(url, json={'query': query})
print(r.text)
```

I do have a result with the query but it just returns one entity. When I try to launch the same request on the GraphQL client available at the same URL (I guess it's GraphiQL), it returns all the entities. I don't know yet where is this issue coming from. I just asked the API provider if there were any limitations from his side, waiting for an answer now.

[Skip to main content](#)

I will update the thread along the process.

(NB: I changed the GraphQL API URL as it's on a dev server, I can provide it by private message if needed.)

KenWhitesell 

Jun 2021

JeremyG:

I just asked the API provider if there were any limitations from his side, waiting for an answer now.

That seems to me to be the best next step. It's *possible* that their web client appends something to the submitted query to specify "return all" rather than "return first" or "return 1".

JeremyG

Jul 2021

Hi,

KenWhitesell:

That seems to me to be the best next step. It's *possible* that their web client appends something to the submitted query to specify "return all" rather than "return first" or "return 1".

I sorted this part, it was rather obvious eventually: only 1 resource was tagged as public, hence only one appeared through the API. My bad, I was sure there were a lot more but I didn't perfectly check. Now with more resources tagged as public, the API returns what it should.

Now, following [this tutorial](#) , I'm trying to format the result into a data frame. Here is my Python script (URL is obfuscated):

```
import requests
import json
import pandas as pd

url = 'https://mygraphqlserver.com/graphql/'
query = """{ resources { edges { node { id creatorPerson { firstName } } } } }
r = requests.post(url, json={'query': query})
json_data = json.loads(r.text)
resources_dict = json_data['data']['resources']['edges']
resources_output = pd.DataFrame(resources_dict)
print(resources_output)
```

[Skip to main content](#)

```

{
  "data": {
    "resources": {
      "edges": [
        {
          "node": {
            "id": "file/pt0u8h901qni3d",
            "creatorPerson": {
              "firstName": "Jérémy"
            }
          }
        },
        {
          "node": {
            "id": "file/f218c8wn4onncj",
            "creatorPerson": {
              "firstName": "Jérémy"
            }
          }
        },
        {
          "node": {
            "id": "file/i1y7pjk7a6xy2d",
            "creatorPerson": {
              "firstName": "Jérémy"
            }
          }
        }
      ]
    }
  }
}

```

Here is the output of `print(resources_output)` in the terminal:

```

node
0 {'id': 'file/pt0u8h901qni3d', 'creatorPerson':...
1 {'id': 'file/f218c8wn4onncj', 'creatorPerson':...
2 {'id': 'file/i1y7pjk7a6xy2d', 'creatorPerson':.....

```

As you can see, I don't succeed to get a proper data frame as expected (as shown in the tutorial).

Any idea how I can improve my script to get a properly formatted output?

[Skip to main content](#)

I don't have access to that tutorial, so I don't know what output you're expecting to get as output for this particular set of data. But based upon what little I know about pandas and the data you have, this output *looks* right to me. Your variable `resources_dict` is an array of dictionaries, so I wouldn't expect to see anything different.

JeremyG

Jul 2021

I don't have access to that tutorial, so I don't know what output you're expecting to get as output for this particular set of data. But based upon what little I know about pandas and the data you have, this output *looks* right to me. Your variable `resources_dict` is an array of dictionaries, so I wouldn't expect to see anything different.

Thanks for your input, it's encouraging. The tutorial is on Medium, there might be a gateway according to the number of articles already read in the month. My script is quite similar to what they proposed but the output differs (theirs is formatted as a table). No intention to focus on this tutorial, I was just afraid that there could be a "structural issue" with my data as it's nested a lot (resources/edges/node) but your feedback suggests that the data seems right.

What I want to do now is displaying this output in a web page generated with Django. From what I understand, I need to work now on integrating the script into a view and writing a template + some CSS.

Thanks again for your help!

KenWhitesell 

Jul 2021

JeremyG:

The tutorial is on Medium, there might be a gateway according to the number of articles already read in the month.

Yes, that's correct, and is the issue I'm having there.

JeremyG:

What I want to do now is displaying this output in a web page generated with Django. From what I understand, I need to work now on integrating the script into a view and writing a template + some CSS.

[Skip to main content](#) plan.

If you run into any problems you can't resolve, you know where to find us. 😊

JeremyG

Sep 2021

Hi Ken,

Back on this project after a while.

KenWhitesell:

I don't have access to that tutorial, so I don't know what output you're expecting to get as output for this particular set of data. But based upon what little I know about pandas and the data you have, this output *looks* right to me. Your variable `resources_dict` is an array of dictionaries, so I wouldn't expect to see anything different.

I don't exactly understand why the values after the 'creatorPerson' keys aren't displayed and the output returns suspension points instead. I suspect that it's because there's one more nesting level, but how could I retrieve the `firstName` and display it?

In this example, only `firstName` is called but there are more elements that I'd like to retrieve eventually.

Thanks!

KenWhitesell 

Sep 2021

First, don't confuse the output of a `print` function with the actual content of an object. You can create a function in an object to return whatever you want to return when printed.

JeremyG:

I suspect that it's because there's one more nesting level, but how could I retrieve the `firstName` and display it?

Edges is a list of dictionaries. A reference to the first `firstName` entry would look like:

```
json_data['data']['resources']['edges'][0]['node']['creatorPerson']  
['firstName']
```

As you can see, you're just walking through the data structure to retrieve the desired element.

JeremyG

Oct 2021

[Skip to main content](#)

You answer clarified a lot of things on my issue, thanks Ken. My objective is to loop through this list of dictionaries to return data about all the available resources. Here's the state of my Python script:

```
import requests, json

url = 'https://mygraphqlserver.com/graphql/'
query = '''{ resources { edges { node { id creatorPerson { firstName } } } } } }
r = requests.post(url, json={'query': query})
resources = json.loads(r.text)['data']['resources']['edges']
for resource in resources:
    id = resource['node']['id']
    creator_first_name = resource['node']['creatorPerson']['firstName']
    print(f'\Id of the resource: {id}\nFirst name of the creator of the resour
```

It works as expected and returns the items 😊

However, this is a Python script, not written for Django, and I struggle to translate it to views.py and to the template. Where should I loop through the list of dict, in views.py or in my template? If it's in the views file, how can I pass the variables to the template? And if it's in the template, what would be the best way to write it?

It's coming forward, little by little! Thanks a lot for your insights!

KenWhitesell 🛡️

Oct 2021

JeremyG:

Where should I loop through the list of dict, in views.py or in my template?

Yes. 😊

(Ok, *not* the most helpful answer, but I couldn't resist.)

You *will* be doing some of the iteration in your template, so it's likely going to be a mix of both.

JeremyG:

If it's in the views file, how can I pass the variables to the template?

The same way you pass *any* data from a view into the rendering engine for the template - in the context.

[Skip to main content](#)

JeremyG:

And if it's in the template, what would be the best way to write it?

That is really a personal design choice. There are multiple ways of doing it, and it's going to depend upon some external factors - information that I wouldn't have, and you may not even have yet.

One such factor is whether there's a foreseen need to *possibly* (now or in the future), add other data from this structure into your template to be displayed. If there is, it *may* be appropriate to pass all of resources to be used by the template and iterate through it in the template.

e.g.

```
context['resources'] = resources
```

Or, another approach might be to create a list of tuples (or dicts) with *just* the data to be rendered. e.g. (sample only, may not be *exactly* what you would need to use)

```
context['creator_list'] = [  
    (resource['node']['id'], resource['node']['creatorPerson']['firstName'])  
    for resource in resources  
]
```

There are different advantages for each approach. Your decisions should likely be made based upon other architectural factors and requirements within your project.

Suggested	✦ Related	Replies	Views	Activity
<input type="checkbox"/>	Images not showing 2	13	8.6k	25d
<input type="checkbox"/>	Get id_token for send to the URL logout keycloak	0	9	4h
<input type="checkbox"/>	Problem with Deployment - Websocket is already in CLOSING or CLOSED state	3	5	18m
<input type="checkbox"/>	GSoC 2024 Proposal: Significant Enhancement of Django's bulk_create Skip to main content	6	63	4h

	Replies	Views	Activity
<input type="checkbox"/> [Errno 111] Connection refused on smtp.gmail.com	1	17	4h

There is [1 unread](#) and [22 new](#) topics remaining, or browse other topics in