graphql-dsl

Release 0.1.3

Maxim Avanov

May 31, 2021

CONTENTS

1 Quick intro **3**

1.1 Simple queries 4

2 Documentation Indices and tables **7**

- *Quick intro*
  - *Simple queries*
- *Documentation Indices and tables*

CHAPTER

ONE

QUICK INTRO

Let's take a manually written GraphQL query:

```
query {
    hero {
        name
    }
    droid(id: "2000") {
        name
    }
}
```

With `graphql-dsl` you can construct a similar query with the following Python snippet:

```
from typing import NamedTuple
from graphql_dsl import *

class Hero(NamedTuple):
    name: str

class Droid(NamedTuple):
    name: str

class HeroAndDroid(NamedTuple):
    hero: Hero
    droid: Droid

class Input(NamedTuple):
    droid_id: ID

q = GQL( QUERY | HeroAndDroid
       | WITH  | Input
       | PASS  | Input.droid_id * TO * HeroAndDroid.droid * AS * 'id'
       )

print(q.query)
```

and the output will be:

```
query HeroAndDroid($droidId:ID!){hero{name}droid(id:$droidId){name}}
```

The query builder supports both `NamedTuple` and `@dataclass` types, yet the latter has a slightly different field reference syntax (because dataclasses don't define class-level field getters):

```
from dataclasses import dataclass
from graphql_dsl import *

@dataclass
class Hero:
    name: str

@dataclass
class Droid:
    name: str

@dataclass
class HeroAndDroid:
    hero: Hero
    droid: Droid

@dataclass
class Input:
    droid_id: ID

q = GQL( QUERY | HeroAndDroid
       | WITH  | Input
       | PASS  | (Input, 'droid_id') * TO * (HeroAndDroid, 'droid') * AS * 'id'
       )
```

1.1 Simple queries

Let's use Countries API and prepare the simplest query for it.

We want to fetch a list of all country codes

```
from typing import Sequence, NamedTuple

class Country(NamedTuple):
    code: str

class Query(NamedTuple):
    countries: Sequence[Country]
```

We can start composing our query with:

```
from graphql_dsl import QUERY

countries_query = QUERY | Query
```

If we don't need to provide input parameters to the query, we can immediately compile it:

```
from graphql_dsl import GQL

compiled_query = GQL(countries_query)
```

Now we are able to call the service and receive the typed result from it:

```
import requests
```

(continues on next page)

(continued from previous page)

```
response = requests.post(
    url="https://countries.trevorblades.com/",
    json={
        "operationName": compiled_query.name,
        "query": compiled_query.query,
    }
)

data = compiled_query.get_result(response.json())
assert isinstance(data, Query)

# will print AD, AE, AF, AG, AI, AL, AM, AO, ...
print(', '.join(country.code for country in data.countries))
```

CHAPTER

TWO

DOCUMENTATION INDICES AND TABLES

- genindex
- modindex
- search