# 7.API.v4

October 27, 2019

## 1 API v4

Este notebook apenas apresenta a API v4.

Anteriormente, o minicurso abordou a API v3 do GitHub, que utiliza REST. Agora, o minicurso abordará a API v4, que usa GraphQL (https://developer.github.com/v4/).

Antes de qualquer coisa, vamos iniciar o servidor de proxy, caso ele esteja fechado:

```
python proxy.py https://api.github.com/
```

Além do servidor de proxy, precisamos carregar o token e preparar a função de autenticação.

[1]:
```
from ipywidgets import FileUpload, interact
@interact(files=FileUpload())
def set_token(files={}):
    global token
    if files:
        for key, values in files.items():
            token = values['content'].decode("utf-8").strip()
            print("Token Loaded!")
```

```
interactive(children=(FileUpload(value={}, description='Upload'), Output()), _dom_classes=('wi
```

[2]:
```
import requests

def token_auth(request):
    request.headers["User-Agent"] = "Minicurso" # Necessário
    request.headers["Authorization"] = "token {}".format(token)
    return request
```

Agora podemos tentar conectar na API v4 e verificar se a autenticação funcionou. Note que usamos POST e URL original é https://api.github.com/graphql.

[3]:
```
SITE = "http://localhost:5000/" # ou https://api.github.com

query = """
{
  rateLimit {
```

```
    limit
    cost
    remaining
    resetAt
  }
}
"""

response = requests.post(SITE + "graphql", json={'query': query},␣
 ↪auth=token_auth)
response.status_code
```

[3]: 200

[4]:
```
import pdffallback
data = response.json()
pdffallback.show(data, convert=True)
```

```
{'data': {'rateLimit': {'cost': 1,
                        'limit': 5000,
                        'remaining': 4991,
                        'resetAt': '2019-10-25T06:44:31Z'}}}
```

A consulta com a API v4 é um pouco mais verbosa, porém existe uma única URL de acesso e o resultado vem exatamente o que consultamos.

A seguir temos um exemplo de uma consulta quase completa em relação ao que fizemos na APIv3.

[5]:
```
query = """
query {
  repository(owner:"gems-uff", name:"sapos") {
      stargazers {
          totalCount
      }
      forks {
          totalCount
      }
      watchers {
          totalCount
      }
      primaryLanguage {
          name
      }
      open_issues: issues(states:OPEN, first:100) {
          totalCount
          edges {
              node {
                  number
```

```
                closedAt
                createdAt
                labels(first:100) {
                  edges {
                    node {
                      name
                    }
                  }
                  pageInfo {
                    startCursor
                    hasNextPage
                    endCursor
                  }
                }
            }
        }
        pageInfo {
          startCursor
          hasNextPage
          endCursor
        }
    }
    closed_issues: issues(states:CLOSED, first:100) {
        totalCount
        edges {
            node {
                number
                closedAt
                createdAt
                labels(first:100) {
                  edges {
                    node {
                      name
                    }
                  }
                  pageInfo {
                    startCursor
                    hasNextPage
                    endCursor
                  }
                }
            }
        }
        pageInfo {
          startCursor
          hasNextPage
          endCursor
```

```
            }
        }
        mentionableUsers(first:100) {
            edges {
                node {
                    login
                }
            }
            pageInfo {
              startCursor
              hasNextPage
              endCursor
            }
        }
    }

}
"""

response = requests.post(SITE + "graphql", json={'query': query},␣
 ↪auth=token_auth)
print(response.status_code)
data = response.json()
pdffallback.show(data, convert=True)
```

```
200
{'data': {'repository': {'closed_issues': {'edges': [{'node': {'closedAt':
'2013-07-19T14:21:51Z',
                                                                'createdAt':
'2013-06-29T15:23:48Z',
                                                                'labels':
{'edges': [{'node': {'name': 'bug'}}],
'pageInfo': {'endCursor': 'Y3Vyc29yOnYyOpKjYnVnzgAUplA=',
       'hasNextPage': False,
…
```

Uma única consulta é capaz de retornar boa parte das informações que precisamos.

Mas ATENÇÃO! Paginação ainda é necessária e é feita com os argumentos `first:100` e `after:{endCursor}`.

[6]:
```
data["data"]["repository"]["closed_issues"]["pageInfo"]
```

[6]:
```
{'startCursor': 'Y3Vyc29yOnYyOpHOAPbUBA==',
 'hasNextPage': True,
 'endCursor': 'Y3Vyc29yOnYyOpHOAa3MGw=='}
```

[7]:
```
query_base = """
query {
  repository(owner:"gems-uff", name:"sapos") {
      closed_issues: issues(states:CLOSED, first:100, after:"%s") {
          totalCount
          edges {
              node {
                  number
                  closedAt
                  createdAt
                  labels(first:100) {
                    edges {
                      node {
                        name
                      }
                    }
                    pageInfo {
                      startCursor
                      hasNextPage
                      endCursor
                    }
                  }
              }
          }
          pageInfo {
            startCursor
            hasNextPage
            endCursor
          }
      }
  }
}
"""
query = query_base %␣
 ↪(data["data"]["repository"]["closed_issues"]["pageInfo"]['endCursor'], )

response = requests.post(SITE + "graphql", json={'query': query},␣
 ↪auth=token_auth)
print(response.status_code)
data2 = response.json()
pdffallback.show(data2, convert=True)
```

```
200
{'data': {'repository': {'closed_issues': {'edges': [{'node': {'closedAt':
'2014-02-24T18:37:47Z',
                                                               'createdAt':
'2014-02-24T15:02:00Z',
```

```
                                                                    'labels':
{'edges': [{'node': {'name': '3.3.1'}}],
'pageInfo': {'endCursor': 'Y3Vyc29yOnYyOpKlMy4zLjHOBRR8KQ==',
       'hasNextPage': False,
…
```

[8]:
```
data2["data"]["repository"]["closed_issues"]["pageInfo"]
```

[8]:
```
{'startCursor': 'Y3Vyc29yOnYyOpHOAa3MmA==',
 'hasNextPage': True,
 'endCursor': 'Y3Vyc29yOnYyOpHOEd1YdA=='}
```

Mais uma página.

[9]:
```
query = query_base %␣
 ↪(data2["data"]["repository"]["closed_issues"]["pageInfo"]['endCursor'], )

response = requests.post(SITE + "graphql", json={'query': query},␣
 ↪auth=token_auth)
print(response.status_code)
data3 = response.json()
pdffallback.show(data3, convert=True)
```

```
200
{'data': {'repository': {'closed_issues': {'edges': [{'node': {'closedAt':
'2018-03-02T20:39:40Z',
                                                                    'createdAt':
'2018-02-28T13:59:51Z',
                                                                    'labels':
{'edges': [{'node': {'name': '4.4.13'}},
    {'node': {'name': '4.4.14'}},
    {'node': {'name': 'bug'}}],
…
```

[10]:
```
data3["data"]["repository"]["closed_issues"]["pageInfo"]
```

[10]:
```
{'startCursor': 'Y3Vyc29yOnYyOpHOEfFpzw==',
 'hasNextPage': False,
 'endCursor': 'Y3Vyc29yOnYyOpHOHWjK4g=='}
```

Foi a última.

### 1.0.1 Schema

O schema da API v4 pode ser encontrado na documentação: https://developer.github.com/v4/object/repository/

Além disso, é possível fazer consultas para obter o schema.

[11]:
```
SITE = "http://localhost:5000/" # ou https://api.github.com

query = """
query {
  __type(name: "Repository") {
    name
    kind
    description
    fields {
      name
      description
    }
  }
}
"""

response = requests.post(SITE + "graphql", json={'query': query},␣
↪auth=token_auth)
print(response.status_code)
data = response.json()
pdffallback.show(data, convert=True, count=10)
```

```
200
{'data': {'__type': {'description': 'A repository contains the content for a '
                                    'project.',
                     'fields': [{'description': 'A list of users that can be '
                                                'assigned to issues in this '
                                                'repository.',
                                 'name': 'assignableUsers'},
                                {'description': 'A list of branch protection '
                                                'rules for this repository.',
                                 'name': 'branchProtectionRules'},
                                {'description': 'Returns the code of conduct '
```

…

Continua: 8.Git.pdf