EM-DAT GRAPHQL API (GNU R version)

Code

We illustrate how to use the EM-DAT GraphQL API in R.

We will use the package `ghql` (https://docs.ropensci.org/ghql/index.html) as an example but as shown below, the same result is possible with a simple HTTP GET request.

Hide

```
# Setup your API key
api_key = '...'

# Setup your GraphQL query string
query_str = '
    query floods {
      api_version
      public_emdat(
        cursor: {limit: 1, offset: 1}
        filters: {
          from: 1990,
          to: 2019,
          classif: ["nat-hyd-flo"],
          include_hist: true
          }
      ) {
        total_available
        info {
          timestamp
          filters
          cursor
        }
        data {
          disno
          classif_key
          type
          subtype
          river_basin
          total_dam
          total_dam_adj
          total_deaths
          total_affected
          entry_date
          last_update
        }
      }
    }
'
```

Using the `gqhl` package

Hide

```
library("ghql")
library("dplyr", quietly = TRUE)

# Initializing the client
client <- GraphqlClient$new(
  url =  "https://api.emdat.be/v1",
  headers =  list(Authorization = api_key)
)

# Parsing the query string
q <- Query$new()
q$query('floods',query_str)

# Executing prepared queries on the client
json_resp <- client$exec(q$queries$floods)
jsonlite::prettify(json_resp)
```

```
{
    "data": {
        "api_version": "v1",
        "public_emdat": {
            "total_available": 4112,
            "info": {
                "timestamp": "2023-11-15T16:29:01Z",
                "filters": {
                    "classif": [
                        "nat-hyd-flo-*"
                    ],
                    "from": 1990,
                    "include_hist": true,
                    "to": 2019
                },
                "cursor": {
                    "limit": 1,
                    "offset": 1
                }
            },
            "data": [
                {
                    "disno": "1990-0002-TUN",
                    "classif_key": "nat-hyd-flo-riv",
                    "type": "Flood",
                    "subtype": "Riverine flood",
                    "river_basin": null,
                    "total_dam": 242800,
                    "total_dam_adj": 543835,
                    "total_deaths": 37,
                    "total_affected": 152000,
                    "entry_date": "2006-07-19",
                    "last_update": "2023-09-25"
                }
            ]
        }
    }
}
```


Hide

```
# The conversion from the json response to a list is straightforward
resp <- json_resp |> jsonlite::fromJSON()

# The dataset is translated to a tibble
tbl <- tibble(resp$data$public_emdat$data)
tbl
```

disno	classif_key	type	subtype	river_basin	total_dam	total_dam_a
<chr>	<chr>	<chr>	<chr>	<lgl>	<int>	<in
1990-0002-TUN	nat-hyd-flo-riv	Flood	Riverine flood	*NA*	242800	5438

1 row | 1-7 of 11 columns

Using the `httr2` package

It is also possible to pass a query through a HTTP GET request, passing the GraphQL query in the request body.

Hide

```
library("httr2")

# Prepare the request with the query as a raw string in the body (see above)
req <- request("https://api.emdat.be/v1") |>
  req_method("GET") |>
  req_headers(Authorization = api_key) |>
  req_body_json(list(query = query_str))

# Execute the request
resp <- req |> req_perform()
resp |>
  resp_body_string() |>
  jsonlite::prettify()
```

```
{
    "data": {
        "api_version": "v1",
        "public_emdat": {
            "total_available": 4112,
            "info": {
                "timestamp": "2023-11-15T16:29:01Z",
                "filters": {
                    "classif": [
                        "nat-hyd-flo-*"
                    ],
                    "from": 1990,
                    "include_hist": true,
                    "to": 2019
                },
                "cursor": {
                    "limit": 1,
                    "offset": 1
                }
            },
            "data": [
                {
                    "disno": "1990-0002-TUN",
                    "classif_key": "nat-hyd-flo-riv",
                    "type": "Flood",
                    "subtype": "Riverine flood",
                    "river_basin": null,
                    "total_dam": 242800,
                    "total_dam_adj": 543835,
                    "total_deaths": 37,
                    "total_affected": 152000,
                    "entry_date": "2006-07-19",
                    "last_update": "2023-09-25"
                }
            ]
        }
    }
}
```