Name:	
Email:	

Objective

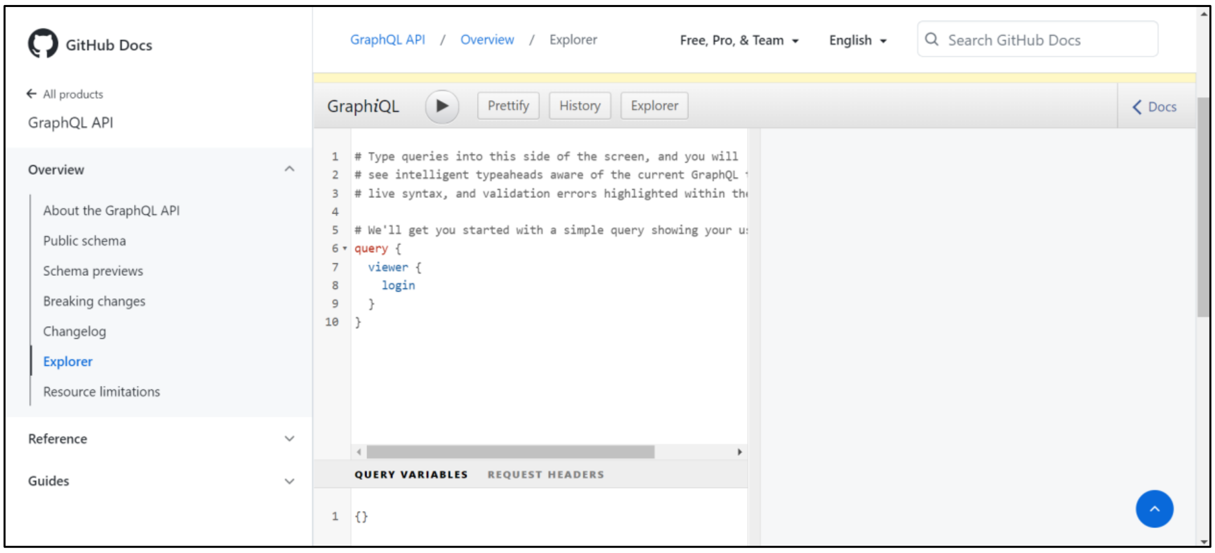
The objective of this workshop is use Github's GraphQL Explorer (playground) to query and mutate your repositories.

Setup

Go to https://docs.github.com/en/graphql/overview/explorer and allow GraphQL to access your repository.

Click on 'Sign in with GitHub' button (top left); this will lead you GitHub's login page. After successfully login in, you will be asked to approve Explorer's access to your account either on the browser or via Github's mobile application.

Once you have approved the access, you will be returned to the Explorer's page



```
# Type queries into this side of the screen, and you will
# see intelligent typeaheads aware of the current GraphQL t
# live syntax, and validation errors highlighted within the

# We'll get you started with a simple query showing your u
query {
  viewer {
    login
  }
}
```

```
{}
```

Query and mutation schema can be found under Reference (bottom left) or with the following URL https://docs.github.com/en/graphql/reference.

Workshop

Write GraphQL operations for the following

1. Get the login name, name and URL of your repository

2. List your first 50 repositories in reverse chronological order; include the following details in each of your repositories: repository name, description, URL and number of forks

3. Repeat the above query but for a different user (eg. Angular, Google, Facebook). The result should include the repository owner's login name, name and URL as well as all the repositories' details as listed in Q2 above.

   Order the repositories by the most recent push.

4. Create a reusable query to search GitHub's repositories by topic. Your query should parameterize the topic's name as well as the number of repositories to return in your query.

   Your query should order by repositories by the most recent push.

5. Create a reusable query to search either repository or discussion. Your query should accept the search string (eg. Angular Universal), the type (either repository or discussion) and the number of results to return.

   If the result is a repository display the repository's name, description and URL. If the result is discussion, display the author's login, the title of the discussion and the body.

   Hint: look at Union Type https://www.apollographql.com/docs/apollo-server/schema/unions-interfaces/#union-type

6. Use mutation to create a new repository. After creating the repository perform an initial push.

   Create a branch protection rule on the `main` (or `master`) branch; configure the rule as follows

   - Do not allow the branch to be deleted
   - Requires a pull request before merging. The number of approvals should be 2

   Note: You cannot create a branch protection if your repository has no commits.

7. Write the `curl` command to perform Question 2. You should try the request with a REST client eg. Postman

Submission

Copy this Word document to your repository and commit it.

```
git add .
git commit -m ‘worksheet03’
git push origin master
```