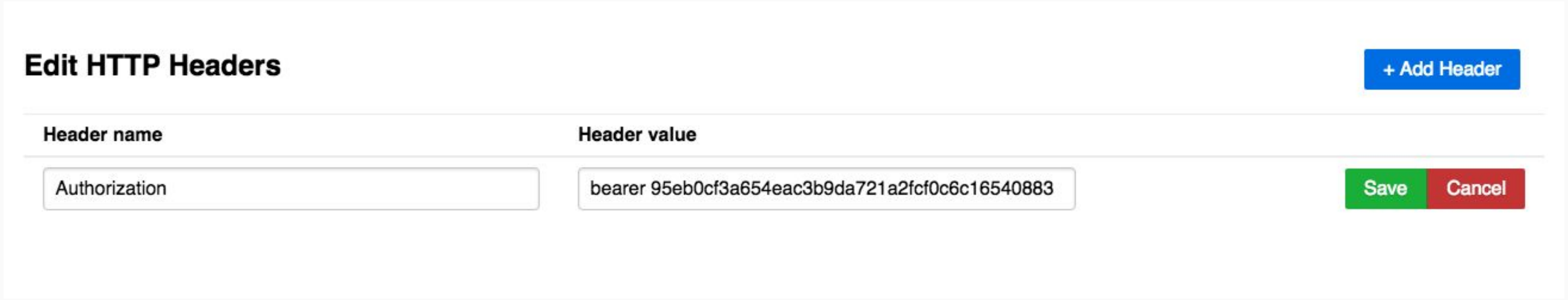
GitHub Lite With GraphQL #2

Issarapong

Growth Session X (#10) - December 21-22 2017

Learning how to request data with a GraphQL API

⇒ `Bearer` in HTTP Header as a `APIKEY`

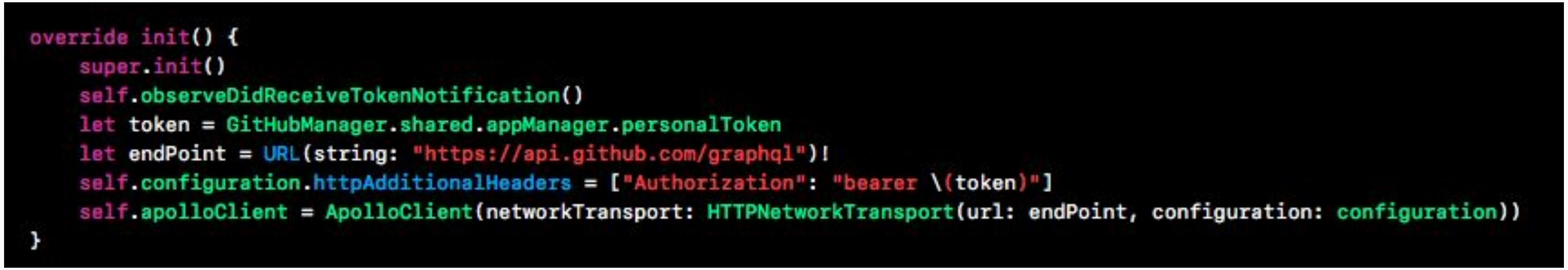


Edit HTTP Headers

Header name	Header value
Authorization	bearer 95eb0cf3a654eac3b9da721a2fcf0c6c16540883

Learning how to request GraphQL API

⇒ `bearer` in HTTP Header as a `APIKEY`

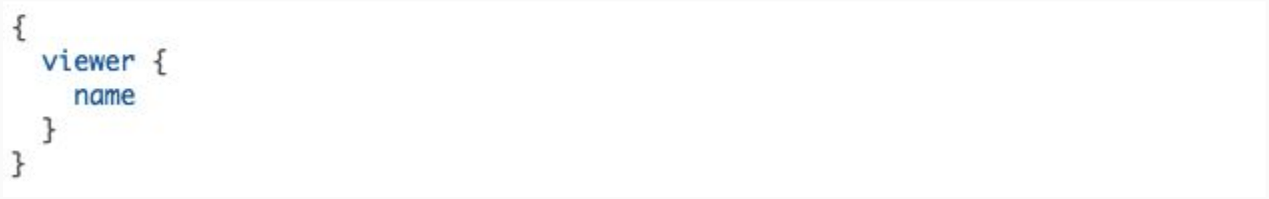


```swift
override init() {
    super.init()
    self.observeDidReceiveTokenNotification()
    let token = GitHubManager.shared.appManager.personalToken
    let endPoint = URL(string: "https://api.github.com/graphql")!
    self.configuration.httpAdditionalHeaders = ["Authorization": "bearer \(token)"]
    self.apolloClient = ApolloClient(networkTransport: HTTPNetworkTransport(url: endPoint, configuration: configuration))
}
```

GraphQL

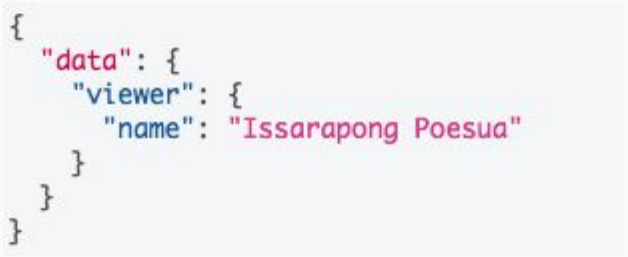
Friendly GraphQL GUI for GraphQL API

- Constructing `GraqhQLRequest` with GraphQL Query Syntax



```
{
  viewer {
    name
  }
}
```

- Reading `JSONResponse`

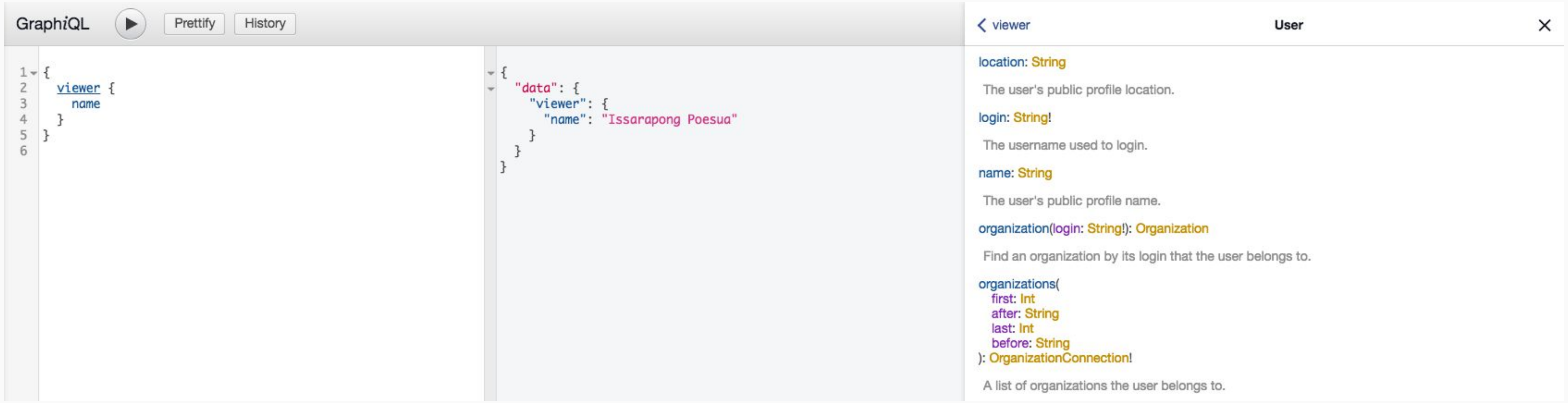


```
{
  "data": {
    "viewer": {
      "name": "Issarapong Poesua"
    }
  }
}
```

GraphQL

Friendly GraphQL GUI for GraphQL API

- Reading API document



```
{
  viewer {
    name
  }
}
```

```
{
  "data": {
    "viewer": {
      "name": "Issarapong Poesua"
    }
  }
}
```

location: String

The user's public profile location.

login: String!

The username used to login.

name: String

The user's public profile name.

organization(login: String!): Organization

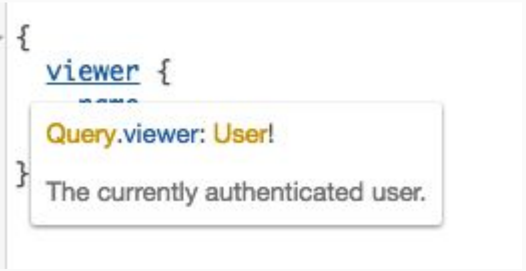
Find an organization by its login that the user belongs to.

```
organizations(
  first: Int
  after: String
  last: Int
  before: String
): OrganizationConnection!
```

A list of organizations the user belongs to.

GraphQL

Models & Query Func will be appear on the RHS of the GraphiQL.app

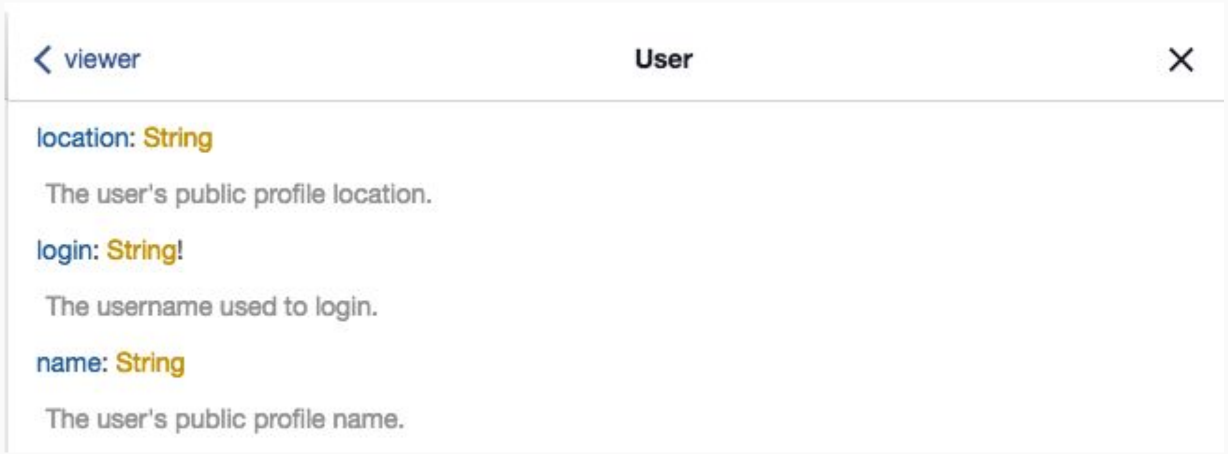


```
{
  viewer {
```

Query.viewer: User!

The currently authenticated user.

```
}
```



location: String

The user's public profile location.

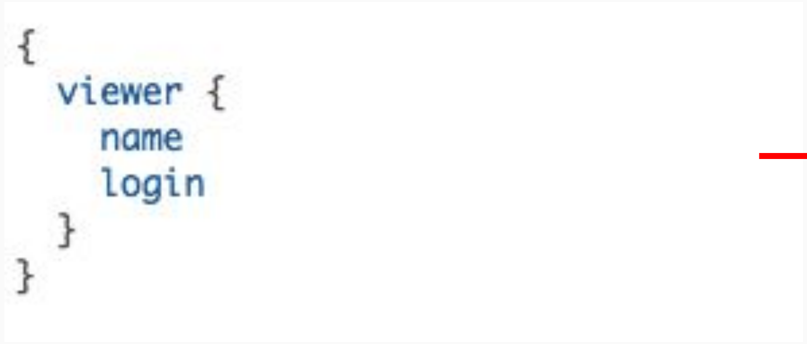
login: String!

The username used to login.

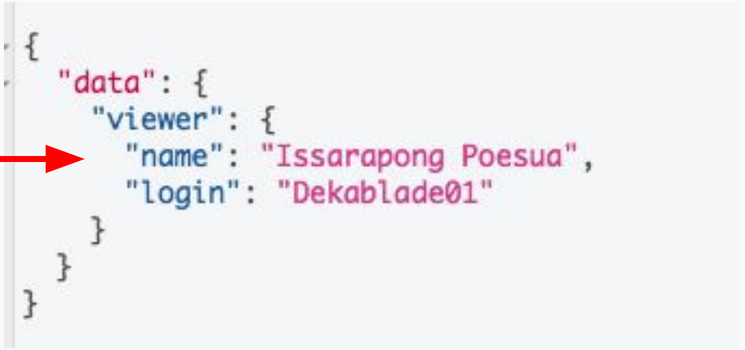
name: String

The user's public profile name.

GraphQL Query

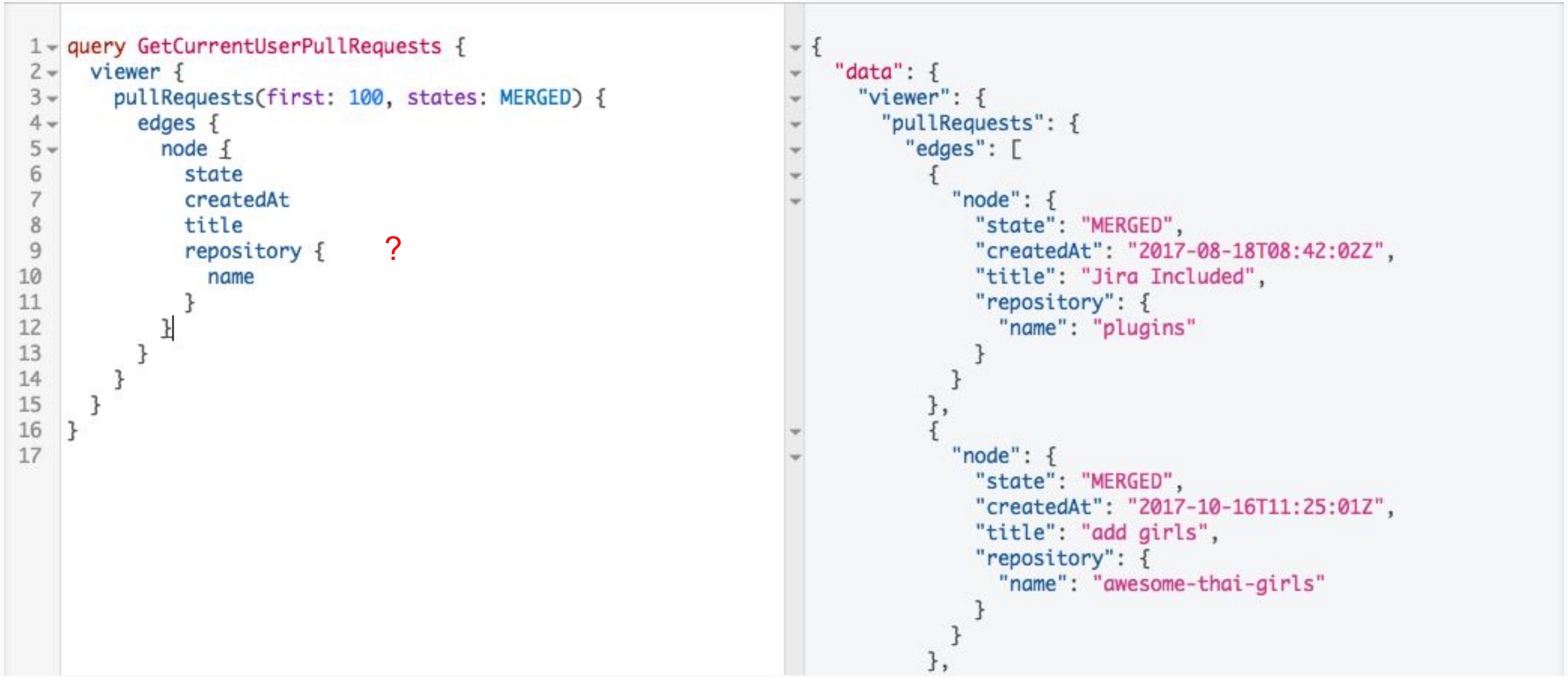


```
{
  viewer {
    name
    login
  }
}
```



```
{
  "data": {
    "viewer": {
      "name": "Issarapong Poesua",
      "login": "Dekablade01"
    }
  }
}
```

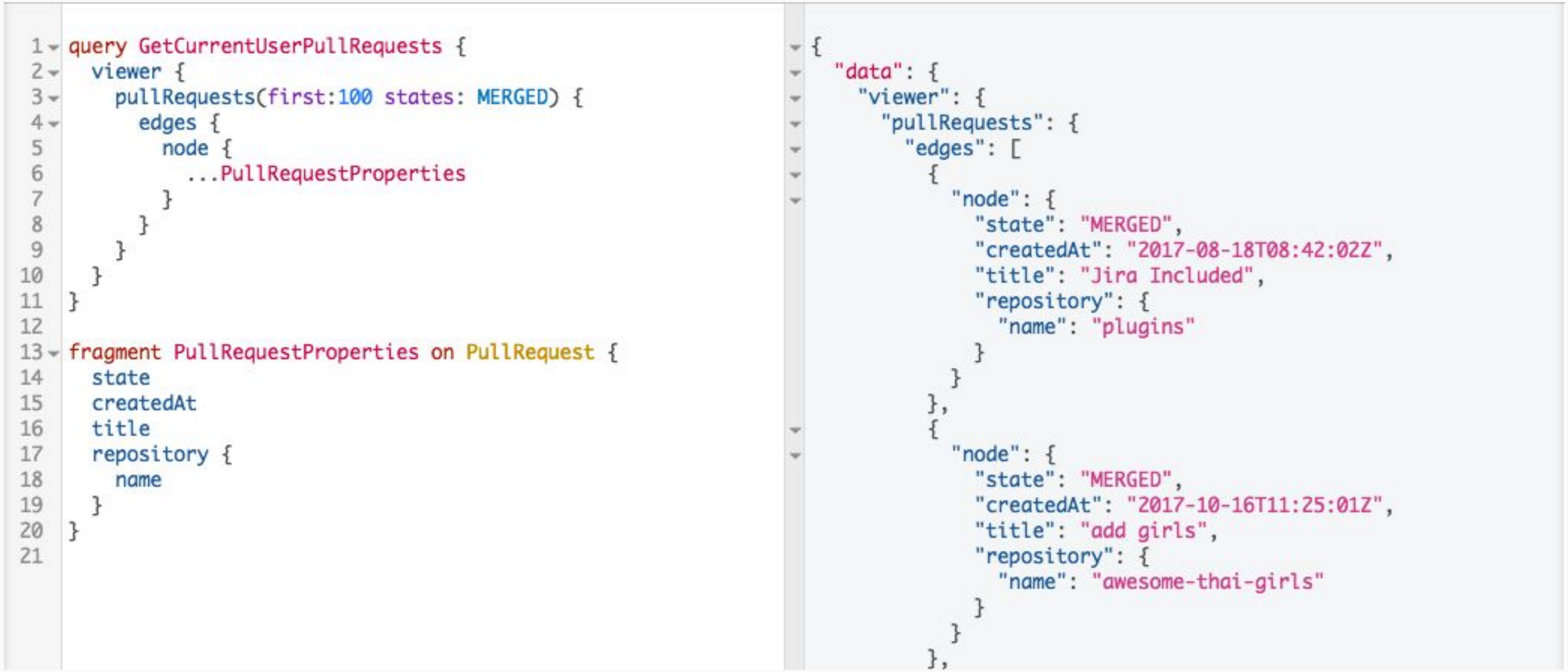
Wrapping Query



```
query GetCurrentUserPullRequests {
  viewer {
    pullRequests(first: 100, states: MERGED) {
      edges {
        node {
          state
          createdAt
          title
          repository {
            name
          }
        }
      }
    }
  }
}
```

```
{
  "data": {
    "viewer": {
      "pullRequests": {
        "edges": [
          {
            "node": {
              "state": "MERGED",
              "createdAt": "2017-08-18T08:42:02Z",
              "title": "Jira Included",
              "repository": {
                "name": "plugins"
              }
            }
          },
          {
            "node": {
              "state": "MERGED",
              "createdAt": "2017-10-16T11:25:01Z",
              "title": "add girls",
              "repository": {
                "name": "awesome-thai-girls"
              }
            }
          },
```

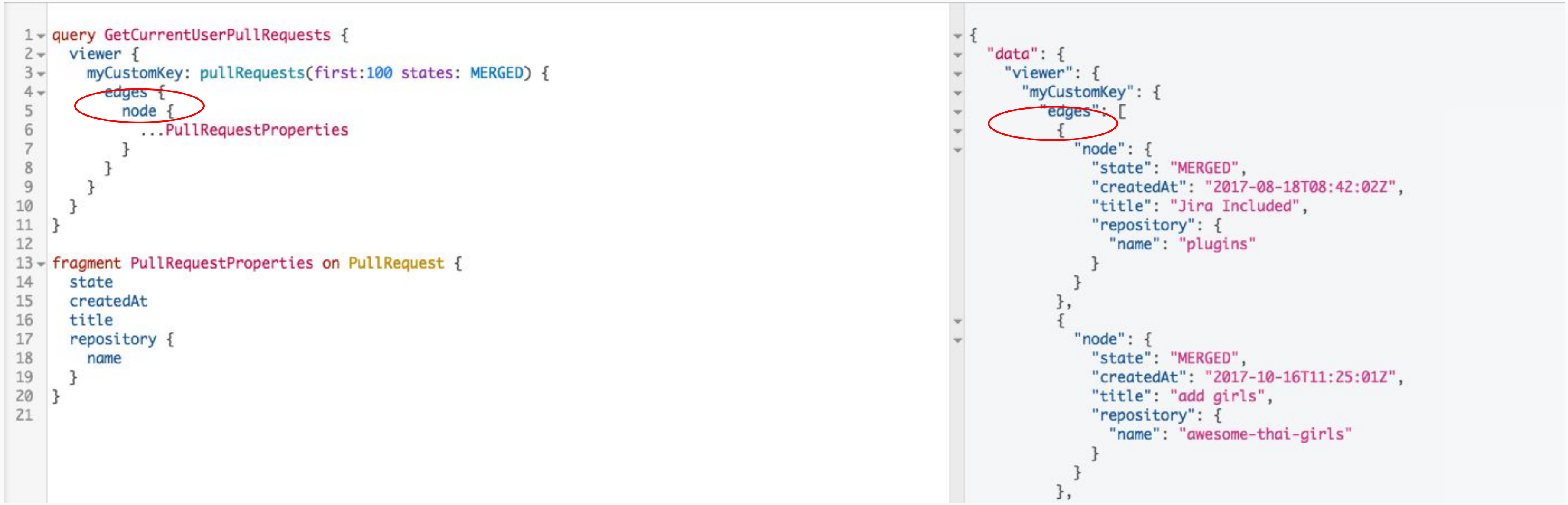
Wrapping Query with Fragment



```
query GetCurrentUserPullRequests {
  viewer {
    pullRequests(first:100 states: MERGED) {
      edges {
        node {
          ...PullRequestProperties
        }
      }
    }
  }
}

fragment PullRequestProperties on PullRequest {
  state
  createdAt
  title
  repository {
    name
  }
}
```

```
{
  "data": {
    "viewer": {
      "pullRequests": {
        "edges": [
          {
            "node": {
              "state": "MERGED",
              "createdAt": "2017-08-18T08:42:02Z",
              "title": "Jira Included",
              "repository": {
                "name": "plugins"
              }
            }
          },
          {
            "node": {
              "state": "MERGED",
              "createdAt": "2017-10-16T11:25:01Z",
              "title": "add girls",
              "repository": {
                "name": "awesome-thai-girls"
              }
            }
          },
```

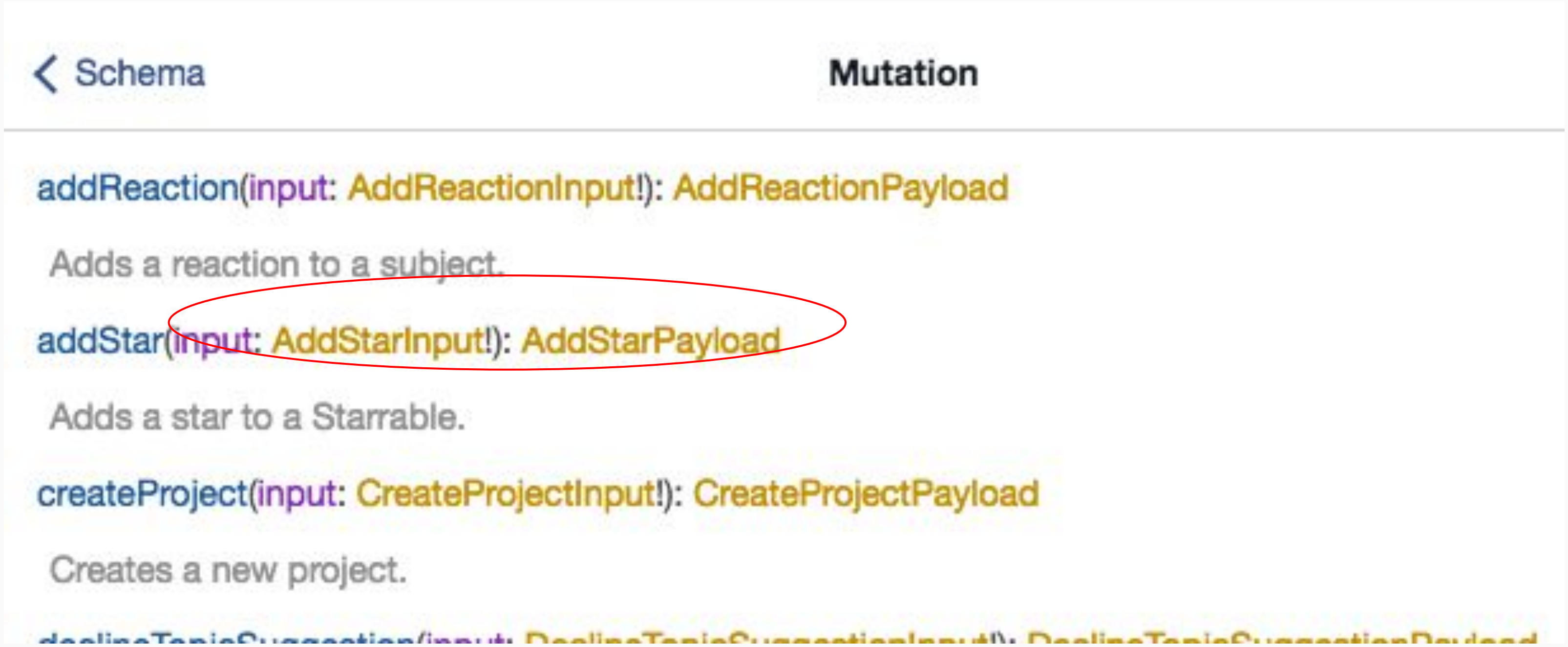
Customizing JSON Key



```
query GetCurrentUserPullRequests {
  viewer {
    myCustomKey: pullRequests(first:100 states: MERGED) {
      edges {
        node {
          ...PullRequestProperties
        }
      }
    }
  }
}

fragment PullRequestProperties on PullRequest {
  state
  createdAt
  title
  repository {
    name
  }
}
```

```
{
  "data": {
    "viewer": {
      "myCustomKey": {
        "edges": [
          {
            "node": {
              "state": "MERGED",
              "createdAt": "2017-08-18T08:42:02Z",
              "title": "Jira Included",
              "repository": {
                "name": "plugins"
              }
            }
          },
          {
            "node": {
              "state": "MERGED",
              "createdAt": "2017-10-16T11:25:01Z",
              "title": "add girls",
              "repository": {
                "name": "awesome-thai-girls"
              }
            }
          },
```

Mutations



< Schema

Mutation

addReaction(input: AddReactionInput!): AddReactionPayload

Adds a reaction to a subject.

addStar(input: AddStarInput!): AddStarPayload

Adds a star to a Starrable.

createProject(input: CreateProjectInput!): CreateProjectPayload

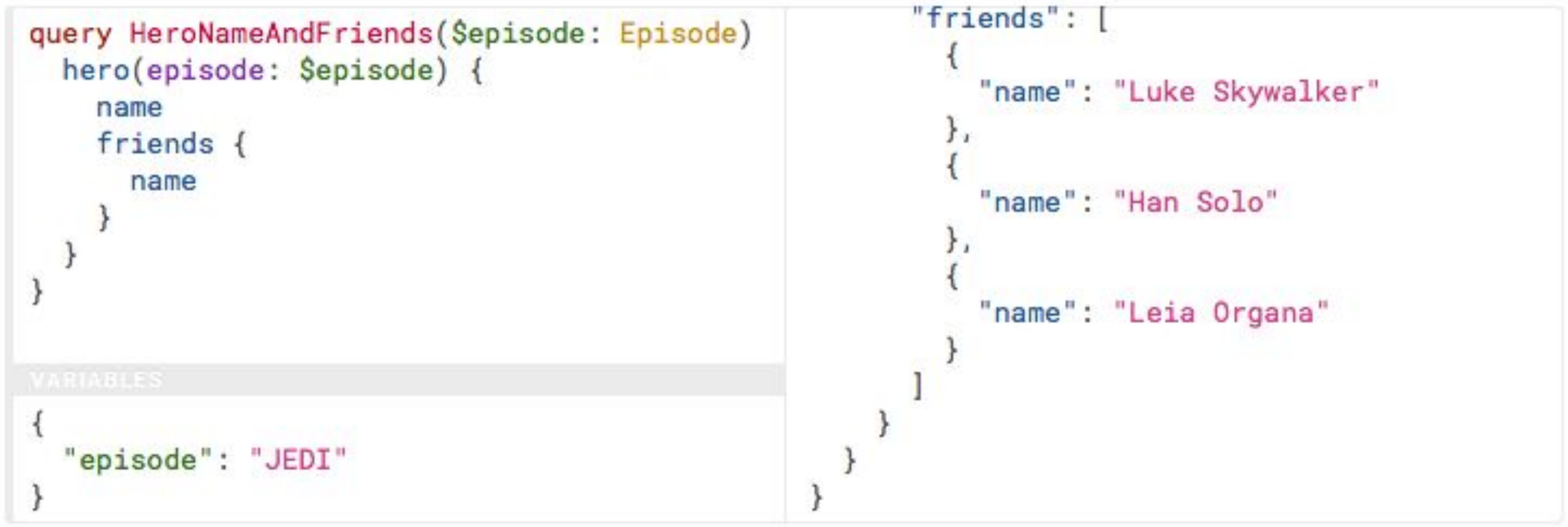
Creates a new project.

declineTopicSuggestion(input: DeclineTopicSuggestionInput!): DeclineTopicSuggestionPayload

Mutations

Mutations

Mutations



```
query HeroNameAndFriends($episode: Episode)
  hero(episode: $episode) {
    name
    friends {
      name
    }
  }
}
```

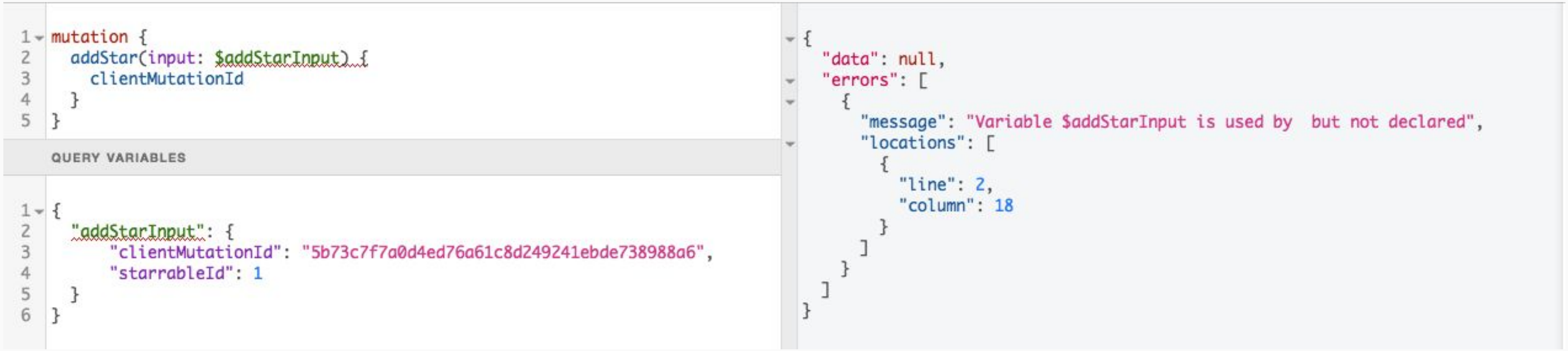
VARIABLES

```
{
  "episode": "JEDI"
}
```

```
        "friends": [
          {
            "name": "Luke Skywalker"
          },
          {
            "name": "Han Solo"
          },
          {
            "name": "Leia Organa"
          }
        ]
      }
    }
}
```

http://graphql.org/learn/queries/

Mutations



```
mutation {
  addStar(input: $addStarInput) {
    clientMutationId
  }
}
```

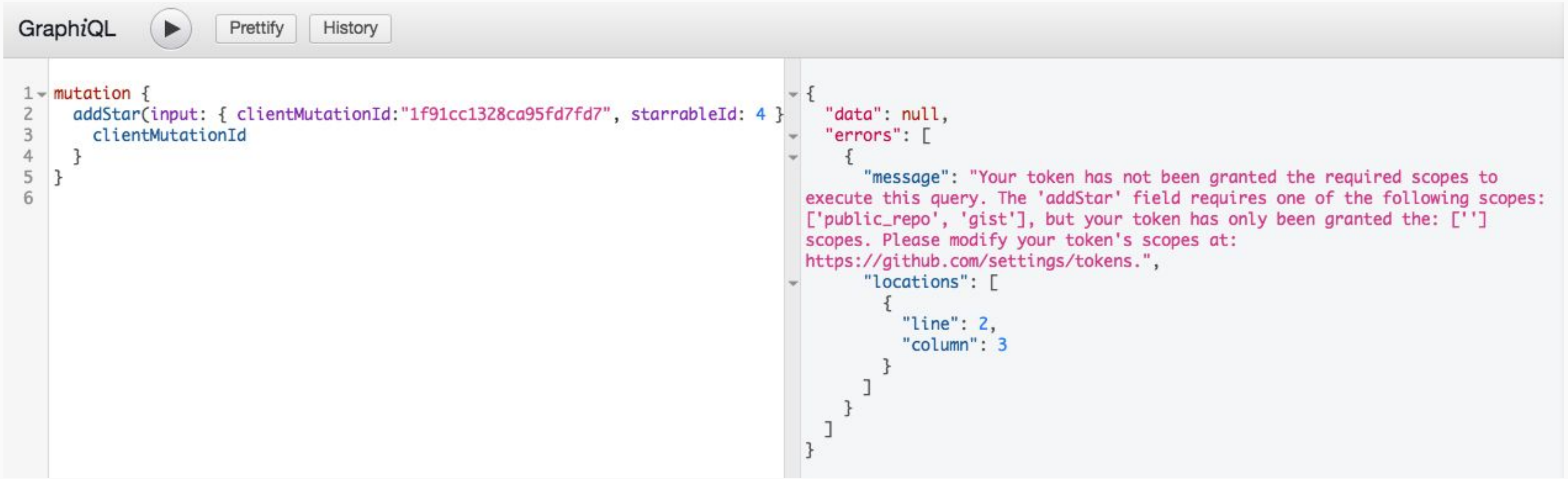
QUERY VARIABLES

```
{
  "addStarInput": {
      "clientMutationId": "5b73c7f7a0d4ed76a61c8d249241ebde738988a6",
      "starrableId": 1
  }
}
```

```
{
  "data": null,
  "errors": [
    {
      "message": "Variable $addStarInput is used by  but not declared",
      "locations": [
        {
          "line": 2,
          "column": 18
        }
      ]
    }
  ]
}
```

Mutations

Mutations

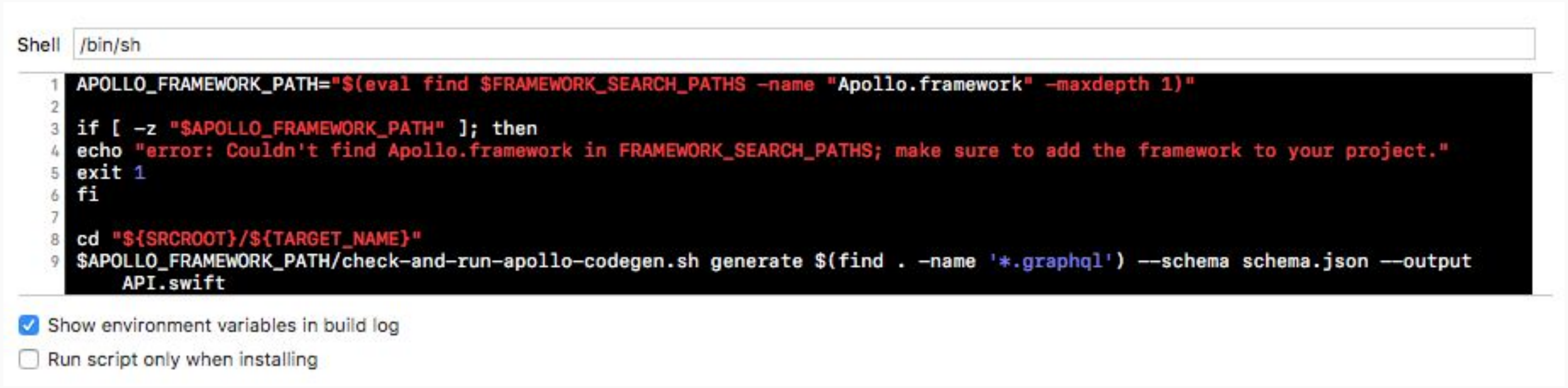


```
mutation {
  addStar(input: { clientMutationId:"1f91cc1328ca95fd7fd7", starrableId: 4 }
    clientMutationId
  }
}
```

```
{
  "data": null,
  "errors": [
    {
      "message": "Your token has not been granted the required scopes to
execute this query. The 'addStar' field requires one of the following scopes:
['public_repo', 'gist'], but your token has only been granted the: ['']
scopes. Please modify your token's scopes at:
https://github.com/settings/tokens.",
      "locations": [
        {
          "line": 2,
          "column": 3
        }
      ]
    }
  ]
}
```

Mutations

Apollo code-gen ⇒ Build script



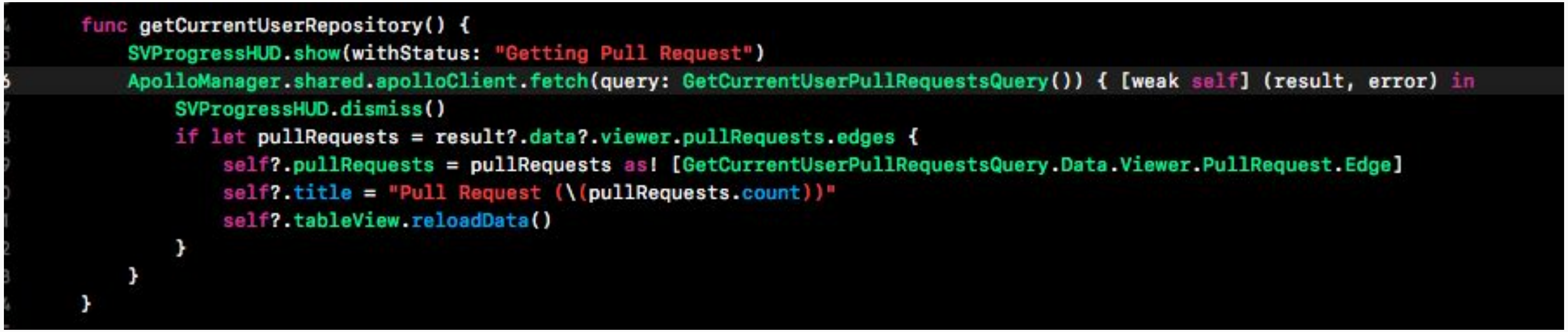
Shell /bin/sh

```
APOLLO_FRAMEWORK_PATH="$(eval find $FRAMEWORK_SEARCH_PATHS -name "Apollo.framework" -maxdepth 1)"

if [ -z "$APOLLO_FRAMEWORK_PATH" ]; then
echo "error: Couldn't find Apollo.framework in FRAMEWORK_SEARCH_PATHS; make sure to add the framework to your project."
exit 1
fi

cd "${SRCROOT}/${TARGET_NAME}"
$APOLLO_FRAMEWORK_PATH/check-and-run-apollo-codegen.sh generate $(find . -name '*.graphql') --schema schema.json --output
    API.swift
```

[✓] Show environment variables in build log
[ ] Run script only when installing

GitHub Lite

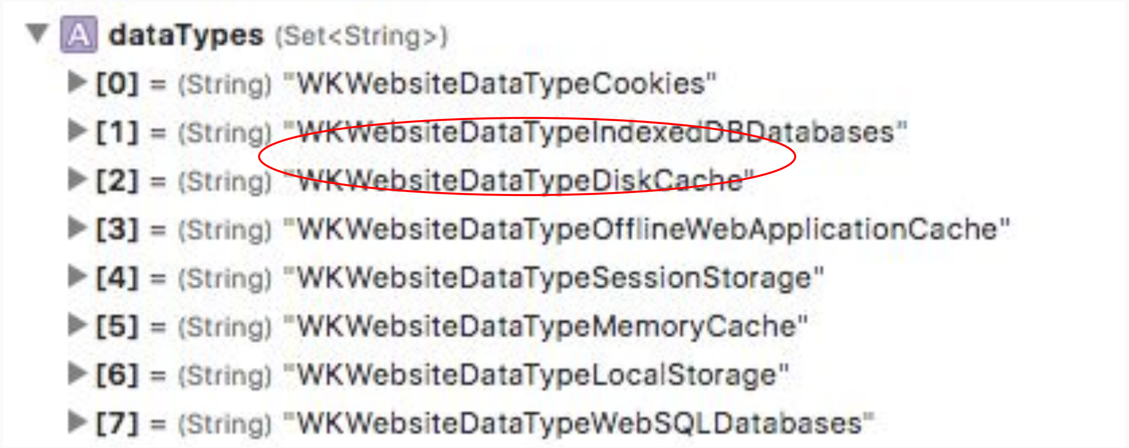
⇒ Generated Swift code



```swift
    func getCurrentUserRepository() {
        SVProgressHUD.show(withStatus: "Getting Pull Request")
        ApolloManager.shared.apolloClient.fetch(query: GetCurrentUserPullRequestsQuery()) { [weak self] (result, error) in
            SVProgressHUD.dismiss()
            if let pullRequests = result?.data?.viewer.pullRequests.edges {
                self?.pullRequests = pullRequests as! [GetCurrentUserPullRequestsQuery.Data.Viewer.PullRequest.Edge]
                self?.title = "Pull Request (\(pullRequests.count))"
                self?.tableView.reloadData()
            }
        }
    }
```

GitHub Lite

- GitHub OAuth app
- WKWebView
  ⇒ Clearing WKWebViewStorage



```
▼ dataTypes (Set<String>)
  ▶ [0] = (String) "WKWebsiteDataTypeCookies"
  ▶ [1] = (String) "WKWebsiteDataTypeIndexedDBDatabases"
  ▶ [2] = (String) "WKWebsiteDataTypeDiskCache"
  ▶ [3] = (String) "WKWebsiteDataTypeOfflineWebApplicationCache"
  ▶ [4] = (String) "WKWebsiteDataTypeSessionStorage"
  ▶ [5] = (String) "WKWebsiteDataTypeMemoryCache"
  ▶ [6] = (String) "WKWebsiteDataTypeLocalStorage"
  ▶ [7] = (String) "WKWebsiteDataTypeWebSQLDatabases"
```

Thanks!

Contact Nimbl3

hello@nimbl3.com

399 Sukhumvit Road, Interchange 21
Klongtoey nua, Wattana
Bangkok 10110

20th Floor, Central Tower
28 Queen's Road
Central, Hong Kong

nimbl3.com