The Structure of a Query

To be able to write better queries, it is important to understand their structure. In this lesson that is exactly what we will be doing.

WE'LL COVER THE FOLLOWING

The Query Statement

Now, let's take a step back to examine the structure of the GraphQL query. After we were introduced to variables, we encountered the `query` statement in our query structure for the first time. Before using the `query` statement, we used the **shorthand version of a query** in which we omitted the statement. However, as we are using variables, the `query` statement must be included. Try the following query without variables but with the `query` statement to verify that the longer version of a query works.

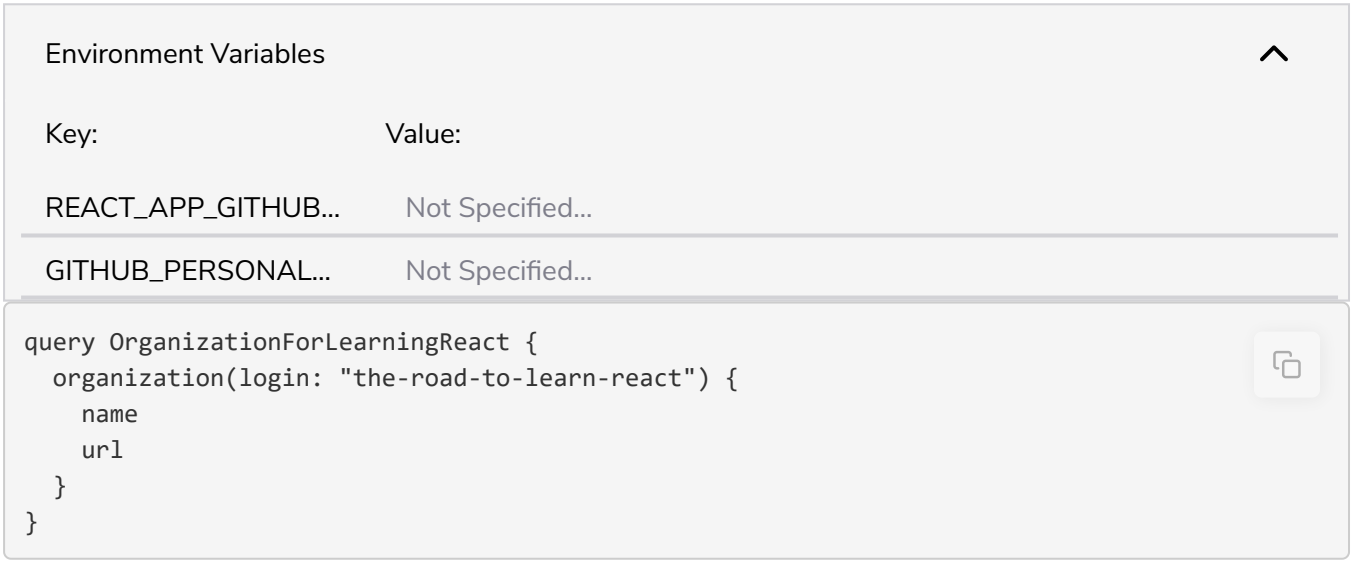
Environment Variables

Key:	Value:
REACT_APP_GITHUB...	Not Specified...
GITHUB_PERSONAL...	Not Specified...

```
query {
  organization(login: "the-road-to-learn-react") {
    name
    url
  }
}
```

While it's not the shorthand version of the query, it still returns the same data as before which is the desired outcome.

The query statement is also called an **operation type** in GraphQL lingua; for instance, it can also be a `mutation` statement. In addition to the operation type, we can even define an **operation name**.



Compare this to anonymous and named functions in our code. A **named query** provides a certain level of clarity about what we want to achieve with the query in a declarative way. Furthermore, it helps us with debugging multiple queries hence, it should be used when you want to implement an application. Your final query, without showing the variables panel again, could look like the following:

Accessing a Nested Object

So far, you've only accessed one object: an organization with a couple of its fields. The GraphQL schema implements a whole graph. So, let's see how to

access a **nested object** from within the graph with a query. It's not that different from what we have been doing before:

Environment Variables

Key:	Value:
REACT_APP_GITHUB...	Not Specified...
GITHUB_PERSONAL...	Not Specified...

```
query OrganizationForLearningReact(
  $organization: String!,
  $repository: String!
) {
  organization(login: $organization) {
    name
    url
    repository(name: $repository) {
      name
    }
  }
}
```

Provide a second variable to request a specific repository of the organization:

Environment Variables

Key:	Value:
REACT_APP_GITHUB...	Not Specified...
GITHUB_PERSONAL...	Not Specified...

```
{
  "organization": "the-road-to-learn-react",
  "repository": "the-road-to-learn-react-chinese"
}
```

Fields in GraphQL can be nested objects. In our example above we have queried two associated objects from the graph. The requests are made on a graph that can have a deeply nested structure. While exploring the "Docs" sidebar in GraphiQL, you might have seen that we can jump from object to object in the graph.

Directive

A **directive** can be used to query data from our GraphQL API in a more powerful way, and they can be applied to fields and objects.

Below, we use two types of directives:

Below, we use two types of directives:

- an **include directive**, which includes the field when the Boolean type is set to true
- a **skip directive**, which instead, excludes the field when the Boolean type is set to true

With these directives, we can apply conditional structures to the shape of our query. The following query showcases the include directive however, we can substitute it with the skip directive to achieve the opposite effect:

Environment Variables

Key:	Value:
REACT_APP_GITHUB...	Not Specified...
GITHUB_PERSONAL...	Not Specified...

```
query OrganizationForLearningReact(
  $organization: String!,
  $repository: String!,
  $withFork: Boolean!
) {
  organization(login: $organization) {
    name
    url
    repository(name: $repository) {
      name
      forkCount @include(if: $withFork)
    }
  }
}
```

Now we can decide whether to include the information for the `forkCount` field based on provided variables.

Environment Variables

Key:	Value:
REACT_APP_GITHUB...	Not Specified...
GITHUB_PERSONAL...	Not Specified...

```
{
  "organization": "the-road-to-learn-react",
  "repository": "the-road-to-learn-react-chinese",
  "withFork": true
}
```

The query in GraphQL gives you all you need to read data from a GraphQL API.

The last few lessons may have felt like a whirlwind of information. To get more comfortable with queries, you can read about them in GraphQL's documentation. Furthermore, try exploring GitHub's query schema by using the "Docs" sidebar in GraphiQL.

In the next few lessons, we will explore mutations.