

# Apollo Client with GraphQL Variables, Nested Objects

The lesson will show you how to implement GraphQL variables and nested objects with Apollo Client.

## WE'LL COVER THE FOLLOWING ^

- Exercise
- Reading Task

You learned about GraphQL pagination and other GraphQL features in previous chapters when you built the React with GraphQL application without Apollo. This lesson will introduce a couple of these features, like GraphQL variables. The `login` argument for the `organization` field in the previous query can be substituted with such a variable.

First, we have to introduce the variable in your GraphQL query:

### Environment Variables ^

Key:

Value:

REACT\_APP\_GITHUB...

Not Specified...

GITHUB\_PERSONAL...

Not Specified...

```
const GET_ORGANIZATION = gql`
  query($organization: String!) {
    organization(login: $organization) {
      name
      url
    }
  }
`;
```



index.js

And secondly, we'll define it in a variables object in our query object:

### Environment Variables ^

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

```
client
  .query({
    query: GET_ORGANIZATION,
    variables: {
      organization: 'the-road-to-learn-react',
    },
  })
  .then(console.log);
```

index.js

That's how you pass variables to the query using an instance of the Apollo Client in your application. Next, we'll add the nested `repositories` list field to our organization. There, we can request all GitHub repositories in an organization. We will also rename the query variable as well and change it when we use the Apollo Client.

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

index.js

```
.ed_required_keys

.ed_keys
```

```
import 'dotenv/config';
import 'cross-fetch/polyfill';
import ApolloClient, { gql } from 'apollo-boost';

const client = new ApolloClient({
  uri: 'https://api.github.com/graphql',
  request: operation => {
    operation.setContext({
      headers: {
        authorization: 'Bearer ==GITHUB_PERSONAL_ACCESS_TOKEN==',
      },
    });
  },
});

const GET_REPOSITORIES_OF_ORGANIZATION = gql`
```

```
query($organization: String!) {  
  organization(login: $organization) {  
    name  
  
    url  
    repositories(first: 5) {  
      edges {  
        node {  
          name  
          url  
        }  
      }  
    }  
  }  
}  
`;  
  
client  
  .query({  
    query: GET_REPOSITORIES_OF_ORGANIZATION,  
    variables: {  
      organization: 'the-road-to-learn-react',  
    },  
  })  
  .then(console.log);
```



index.js

## Exercise #

1. Confirm your [source code for the last lesson](#)

## Reading Task #

Read more about [pagination in GraphQL](#).