GraphiQL
Docs
QUERY VARIABLES

```
# Welcome to GraphiQL
#
# GraphiQL is an in-browser IDE for writing, validating, and
# testing GraphQL queries.
#
# Type queries into this side of the screen, and you will
# see intelligent typeaheads aware of the current GraphQL type schema and
# live syntax and validation errors highlighted within the text.
#
# To bring up the auto-complete at any point, just press Ctrl-Space.
#
# Press the run button above, or Cmd-Enter to execute the query, and the result
# will appear in the pane to the right.

```

```
{
  "errors": [
    {
      "message": "Syntax Error GraphQL request (15:1) Unexpected EOF\n\n14: \n15: \n    ^\
      "locations": [
        {
          "line": 15,
          "column": 1
        }
      ]
    }
  ]
}
```

GraphiQL
Docs
QUERY VARIABLES

```
# Welcome to GraphiQL
#
# GraphiQL is an in-browser IDE for writing, validating, and
# testing GraphQL queries.
#
# Type queries into this side of the screen, and you will
# see intelligent typeaheads aware of the current GraphQL type schema and
# live syntax and validation errors highlighted within the text.
#
# To bring up the auto-complete at any point, just press Ctrl-Space.
#
# Press the run button above, or Cmd-Enter to execute the query, and the result
# will appear in the pane to the right.

```

```
{
  "errors": [
    {
      "message": "Syntax Error GraphQL request (15:1) Unexpected EOF\n\n14: \n15: \n    ^\
      "locations": [
        {
          "line": 15,
          "column": 1
        }
      ]
    }
  ]
}
```

GraphiQL
▶
‹ Docs
QUERY VARIABLES

```
1  # Welcome to GraphiQL
2  #
3  # GraphiQL is an in-browser IDE for writing, validating, and
4  # testing GraphQL queries.
5  #
6  # Type queries into this side of the screen, and you will
7  # see intelligent typeaheads aware of the current GraphQL type schema and
8  # live syntax and validation errors highlighted within the text.
9  #
10 # To bring up the auto-complete at any point, just press Ctrl-Space.
11 #
12 # Press the run button above, or Cmd-Enter to execute the query, and the result
13 # will appear in the pane to the right.
14
```

```
{
  "errors": [
    {
      "message": "Syntax Error GraphQL request (15:1) Unexpected EOF\n\n14: \n15: \n
      "locations": [
        {
          "line": 15,
          "column": 1
        }
      ]
    }
  ]
}
```

WHY THIS IS SO AWESOME

Transparency. When I look at the method call getAllUsers I have a *reasonable* idea of what the response will probably be, but I don't know *for sure*. Will it include comments by that user? Is any data excluded?

With GraphQL I know *exactly* what data is being fetched, because each piece is declared in the call.