

AUTHOR TYPE

In this GraphQL tutorial we'll make our second data type - the Author type. We'll also set up our root query for an Author as well.



```
...
const {
  GraphQLObjectType,
  GraphQLString,
  GraphQLSchema,
  GraphQLID,
  GraphQLInt
} = graphql;
// dummy data
var books = [...];
var authors = [
  { name: 'Patrick Rothfuss', age: 44, id: '1' },
  { name: 'Brandon Sanderson', age: 42, id: '2' },
  { name: 'Terry Pratchett', age: 66, id: '3' }
];
const BookType = new GraphQLObjectType({...});

const AuthorType = new GraphQLObjectType({
  name: 'Author',
  fields: ( ) => ({
    id: { type: GraphQLID },
    name: { type: GraphQLString },
    age: { type: GraphQLInt }
  })
});
const RootQuery = new GraphQLObjectType({
  name: 'RootQueryType',
  fields: {
    book: {...},
    author: {
      type: AuthorType,
      args: { id: { type: GraphQLID } },
      resolve(parent, args){
        return _.find(authors, { id: args.id });
      }
    }
  }
});
```

Insert
code



Go to <http://localhost:4000/graphql>

The screenshot shows the GraphQL Playground interface in a web browser. The address bar shows `localhost:4000/graphql`. The interface has a top bar with the title "GraphiQL" and buttons for "Prettify", "Merge", "Copy", and "History". The main editor area contains a GraphQL schema definition for `RootQueryType`. The schema includes two fields: `book(id: ID!): Book` and `author(id: ID!): Author`. The right sidebar shows the "Schema" tab selected, displaying the schema definition. The "QUERY VARIABLES" section is visible at the bottom.

```
1 # Welcome to GraphiQL
2 #
3 # GraphiQL is an in-browser tool for writing
4 # testing GraphQL queries.
5 #
6 # Type queries into this side of the screen
7 # and you'll see the AST and query plan
8 # validation errors highlighted within the
9 #
10 # GraphQL queries typically start with a
11 # with a # are ignored.
12 #
13 # An example GraphQL query might look like
14 #
15 # {
16 #   field(arg: "value") {
17 #     subField
18 #   }
19 # }
20 #
21 # Keyboard shortcuts:
22 #
23 # Prettify Query: Shift-Ctrl-P (or press
```

Schema **RootQueryType**

Search RootQueryType...

No Description


FIELDS

`book(id: ID!): Book`
`author(id: ID!): Author`

QUERY VARIABLES

Go to <http://localhost:4000/graphql>

GraphiQL



Prettify

Merge

Copy

History

```
1 {
2   author(id: 2){
3     name
4     age
5     id
6   }
7 }
```

```
{
  "data": {
    "author": {
      "name": "Brandon Sanderson",
      "age": 42,
      "id": "2"
    }
  }
}
```

QUERY VARIABLES

< Schema

RootQueryType

×

🔍 Search RootQueryType...

No Description

FIELDS

book(id: ID): Book

author(id: ID): Author