

MUTATIONS

In this GraphQL tutorial I'll explain what mutations are, and how we can use them to add, change or delete our data.




```
const graphql = require('graphql');
const Book = require('../models/book');
const Author = require('../models/Author');
const _ = require('lodash');
const {...} = graphql;
```

```
const Mutation = new GraphQLObjectType({
  name: 'Mutation',
  fields: {
    addAuthor: {
      type: AuthorType,
      args: {
        name: { type: GraphQLString },
        age: { type: GraphQLInt }
      },
      resolve(parent, args){
        let author = new Author({
          name: args.name,
          age: args.age
        });
        return author.save();
      }
    }
  }
});
```

```
module.exports = new GraphQLSchema({
  query: RootQuery,
  mutation: Mutation
});
```

Insert
code

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



The screenshot shows the GraphQL Playground interface. At the top, the title "GraphiQL" is followed by a play button icon (a circle with a right-pointing triangle), which is highlighted with a red square and a red arrow. To the right of the play button are four buttons: "Prettify", "Merge", "Copy", and "History". Below the toolbar, the left pane contains a query written in a monospaced font with syntax highlighting. The query is:

```
1 mutation{
2   addAuthor(name: "Mary", age: 18) {
3     id
4   }
5 }
```

 A red rectangle highlights the entire query. The right pane displays the JSON response, also with syntax highlighting:

```
{
  "data": {
    "addAuthor": {
      "id": "614d7b11e2756f8148c28d6e"
    }
  }
}
```

 At the bottom of the interface, there is a section labeled "QUERY VARIABLES" which is currently empty.

GraphiQL

Play button icon (highlighted with a red square and arrow)

Prettify Merge Copy History

```
1 mutation{
2   addAuthor(name: "Mary", age: 18) {
3     id
4   }
5 }
```

```
{
  "data": {
    "addAuthor": {
      "id": "614d7b11e2756f8148c28d6e"
    }
  }
}
```

QUERY VARIABLES

Go to MongoDB Compass

MongoDB Compass - Connect

Connect View Help

New Connection

★ Favorites

🕒 Recents

- AN HOUR AGO
cluster0.9h62k.mongodb.net
- AUG 23, 2021 4:47 PM
localhost:27017
- AUG 23, 2021 4:32 PM
cluster0.zpxqd.mongodb.net
- AUG 18, 2021 3:53 PM
localhost:27017
- AUG 18, 2021 3:28 PM
localhost:27017
- AUG 18, 2021 8:07 AM
localhost:27017
- AUG 15, 2021 4:08 PM
localhost:27017
- AUG 15, 2021 4:07 PM
localhost:27017
- AUG 14, 2021 3:26 PM
cluster0.zpxqd.mongodb.net
- AUG 9, 2021 2:40 PM
cluster0.zpxqd.mongodb.net

New Connection

☆ FAVORITE

Fill in connection fields individually

Paste your connection string (SRV or Standard ⓘ)

Insert connection string from app.js

mongodb+srv://marizza:demo8888@cluster0.9h62k.mongodb.net/gql

Connect

New to Compass and don't have a cluster?

If you don't already have a cluster, you can create one for free using [MongoDB Atlas](#).

CREATE FREE CLUSTER

How do I find my connection string in Atlas?

If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.

[See example](#)

How do I format my connection string?

[See example](#)

Go to MongoDB Compass

The screenshot shows the MongoDB Compass web interface. The left sidebar displays the database structure: 4 DBS, 11 COLLECTIONS, and a list of hosts. The 'gql' database is expanded, showing the 'authors' collection. The main panel shows the 'gql.authors' collection with 1 document. The document is highlighted with a red box and contains the following JSON:

```
{
  "_id": ObjectId("614d7b11e2756f8148c28d6e"),
  "name": "Mary",
  "age": 18,
  "__v": 0
}
```

The interface also includes a filter bar with the filter `{ field: 'value' }`, a search bar, and a table view of the documents. The status bar at the bottom indicates 'Displaying documents 1 - 1 of 1'.

```
const graphql = require('graphql');
const Book = require('../models/book');
const Author = require('../models/Author');
const _ = require('lodash');
const {...} = graphql;

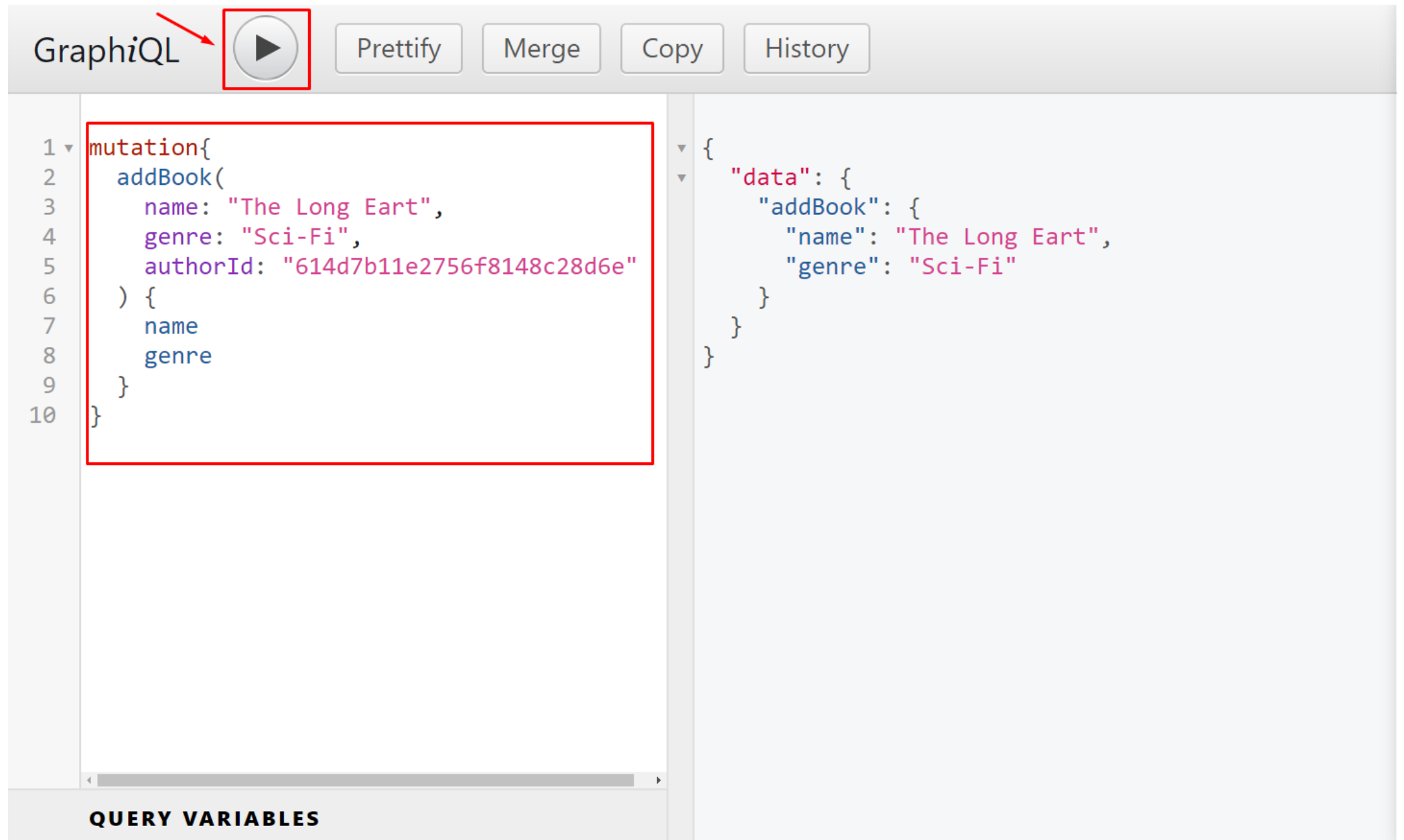
const Mutation = new GraphQLObjectType({
  name: 'Mutation',
  fields: {
    addAuthor: {...},
    addBook: {
      type: BookType,
      args: {
        name: { type: GraphQLString },
        genre: { type: GraphQLString },
        authorId: { type: GraphQLID }
      },
      resolve(parent, args){
        let book = new Book({
          name: args.name,
          genre: args.genre,
          authorId: args.authorId
        });
        return book.save();
      }
    }
  }
});

module.exports = new GraphQLSchema({
  query: RootQuery,
  mutation: Mutation
});
```

Insert
code



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



The screenshot shows the GraphQL Playground interface. At the top, there's a toolbar with buttons: "Prettify", "Merge", "Copy", and "History". A red arrow points to a play button icon (a circle with a right-pointing triangle) which is also enclosed in a red box. Below the toolbar, the interface is split into two main panels. The left panel contains a query editor with line numbers 1 through 10. A red box highlights the query text:

```
1 mutation{
2   addBook(
3     name: "The Long Eart",
4     genre: "Sci-Fi",
5     authorId: "614d7b11e2756f8148c28d6e"
6   ) {
7     name
8     genre
9   }
10 }
```

 The right panel shows the JSON response:

```
{
  "data": {
    "addBook": {
      "name": "The Long Eart",
      "genre": "Sci-Fi"
    }
  }
}
```

 At the bottom of the interface, there's a section labeled "QUERY VARIABLES" which is currently empty.

Go to MongoDB Compass

MongoDB Compass - cluster0.9h62k.mongodb.net/gql.books

Connect View Collection Help

Local

4 DBS 11 COLLECTIONS

☆ FAVORITE

HOSTS

cluster0-shard-00-02.9h6...
cluster0-shard-00-00.9h6...
cluster0-shard-00-01.9h62...

CLUSTER

Replica Set (atlas-14acvk-...
3 Nodes

EDITION

MongoDB 4.4.9 Enterprise

Filter your data

- > admin
- > config
- > gql
 - authors
 - books
- > local

gql.books Documents

gql.books

DOCUMENTS 1 TOTAL SIZE 112B AVG. SIZE 112B INDEXES 1 TOTAL SIZE 20.0KB AVG. SIZE 20.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA

VIEW

Displaying documents 1 - 1 of 1

```
{
  "_id": ObjectId("614dd84b9e01c3cf0009db01"),
  "name": "The Long Eart",
  "genre": "Sci-Fi",
  "authorId": "614d7b11e2756f8148c28d6e",
  "__v": 0
}
```