


GraphQL Read all Books


GraphiQL  Prettify Merge Copy History

```
1 query {
2   books {
3     id
4     name
5     author {
6       id
7       name
8     }
9   }
10 }
11
```

```
{
  "data": {
    "books": [
      {
        "id": 1,
        "name": "Harry Potter and the Chamber of Secrets",
        "author": {
          "id": 1,
          "name": "J. K. Rowling"
        }
      },
      {
        "id": 2,
        "name": "Harry Potter and the Prisoner of Azkaban",
        "author": {
          "id": 1,
          "name": "J. K. Rowling"
        }
      },
      {
        "id": 3,
        "name": "Harry Potter and the Goblet of Fire",
        "author": {
          "id": 1,
          "name": "J. K. Rowling"
        }
      },
      {
        "id": 4,
        "name": "The Merchant of Venice",
        "author": {
          "id": 2,
          "name": "Shakespeare"
        }
      },
      {
        "id": 5,
        "name": "Antony and Cleopatra",
        "author": {
          "id": 2,
          "name": "Shakespeare"
        }
      },
      {
        "id": 6,
        "name": "Geetanjali",
        "author": {
          "id": 3,
          "name": "Rabindra Nath Tagore"
        }
      }
    ]
  }
}
```

QUERY VARIABLES


GraphQL Read Books by id

GraphiQL  Prettify Merge Copy History

```
1 query {
2   book(id: 6) {
3     id
4     name
5     author {
6       id
7       name
8     }
9   }
10 }
11
```

```
{
  "data": {
    "book": {
      "id": 6,
      "name": "Geetanjali",
      "author": {
        "id": 3,
        "name": "Rabindra Nath Tagore"
      }
    }
  }
}
```


GraphQL create books

GraphiQL  Prettify Merge Copy History

```
1 mutation {
2   addBook(name: "New added Book", authorId: 3) {
3     id
4     name
5     author {
6       id
7       name
8     }
9   }
10 }
11
```

```
{
  "data": {
    "addBook": {
      "id": 7,
      "name": "New added Book",
      "author": {
        "id": 3,
        "name": "Rabindra Nath Tagore"
      }
    }
  }
}
```


GraphQL update books

GraphiQL  Prettify Merge Copy History

```
1 mutation {
2   updateBookName(id: 7, newName: "Updated new added book") {
3     id
4     name
5   }
6 }
7
```

```
{
  "data": {
    "updateBookName": {
      "id": 7,
      "name": "Updated new added book"
    }
  }
}
```

GraphQL delete books

GraphiQL  Prettify Merge Copy History

```
1 mutation {
2   deleteBook(id: 7)
3 }
4
```

```
{
  "data": {
    "deleteBook": "Book with Id 7 deleted successfully"
  }
}
```

GraphQL Read all Authors

GraphiQL

Prettify

Merge

Copy

History

1 query {
2 authors {
3 id
4 name
5 books {
6 id
7 name
8 }
9 }
10 }
11 }

book

books

authors

author

__typename

__schema

__type

Book A Single Book

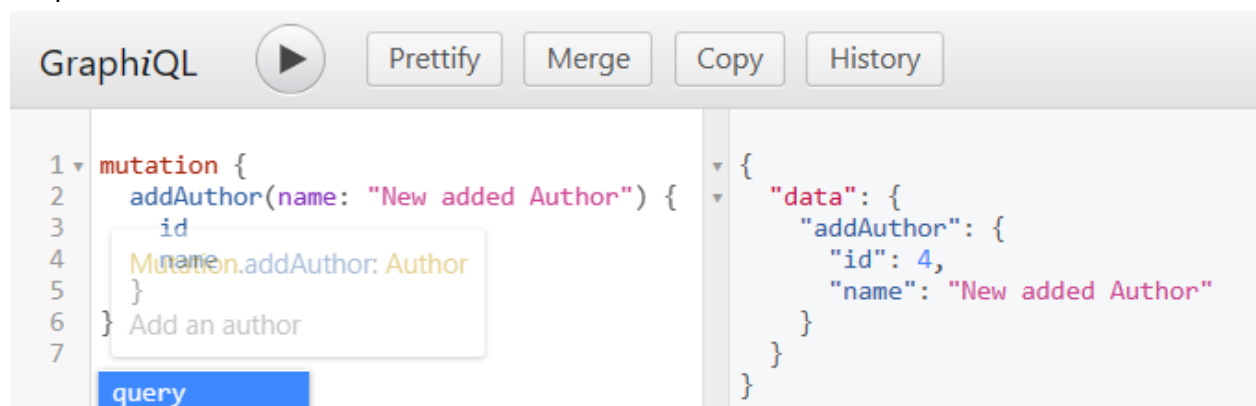
```
{  
  "data": {  
    "authors": [  
      {  
        "id": 1,  
        "name": "J. K. Rowling",  
        "books": [  
          {  
            "id": 1,  
            "name": "Harry Potter and the Chamber of Secrets"  
          },  
          {  
            "id": 2,  
            "name": "Harry Potter and the Prisoner of Azkaban"  
          },  
          {  
            "id": 3,  
            "name": "Harry Potter and the Goblet of Fire"  
          }  
        ]  
      },  
      {  
        "id": 2,  
        "name": "Shakespeare",  
        "books": [  
          {  
            "id": 4,  
            "name": "The Merchant of Venice"  
          },  
          {  
            "id": 5,  
            "name": "Antony and Cleopatra"  
          }  
        ]  
      },  
      {  
        "id": 3,  
        "name": "Rabindra Nath Tagore",  
        "books": [  
          {  
            "id": 6,  
            "name": "Geetanjali"  
          }  
        ]  
      }  
    ]  
  }  
}
```

GraphQL Read Authors by id



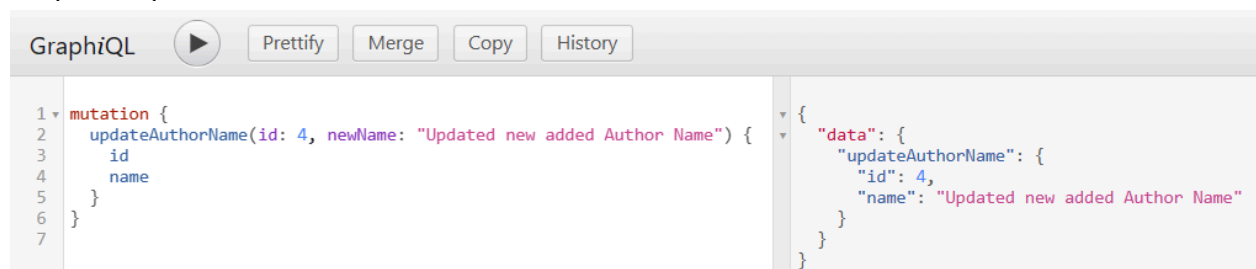
The screenshot shows the GraphQL IDE interface. On the left, a query is entered: `query { author(id: 2) { name } }`. On the right, the JSON response is displayed: `{ "data": { "author": { "name": "Shakespeare" } } }`. The interface includes a play button and buttons for 'Prettify', 'Merge', 'Copy', and 'History'.

GraphQL create Authors



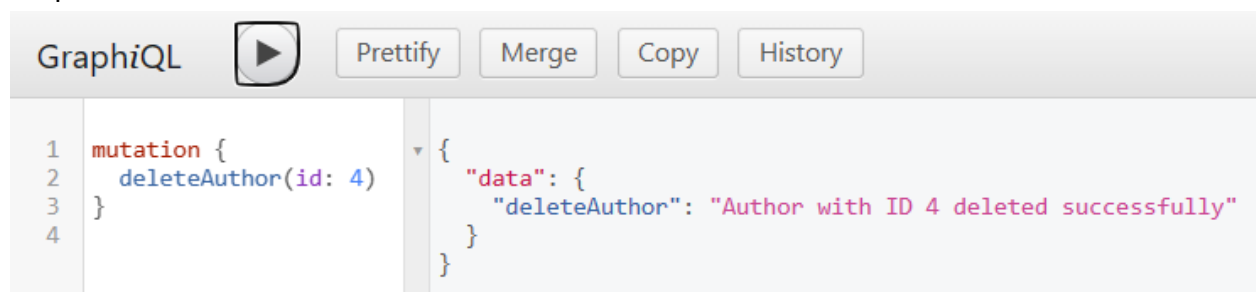
The screenshot shows the GraphQL IDE interface. On the left, a mutation is entered: `mutation { addAuthor(name: "New added Author") { id name } }`. A tooltip for the `name` field shows `name: Author`. On the right, the JSON response is displayed: `{ "data": { "addAuthor": { "id": 4, "name": "New added Author" } } }`. The interface includes a play button and buttons for 'Prettify', 'Merge', 'Copy', and 'History'.

GraphQL update Authors



The screenshot shows the GraphQL IDE interface. On the left, a mutation is entered: `mutation { updateAuthorName(id: 4, newName: "Updated new added Author Name") { id name } }`. On the right, the JSON response is displayed: `{ "data": { "updateAuthorName": { "id": 4, "name": "Updated new added Author Name" } } }`. The interface includes a play button and buttons for 'Prettify', 'Merge', 'Copy', and 'History'.

GraphQL delete Authors



The screenshot shows the GraphQL IDE interface. On the left, a mutation is entered: `mutation { deleteAuthor(id: 4) }`. On the right, the JSON response is displayed: `{ "data": { "deleteAuthor": "Author with ID 4 deleted successfully" } }`. The interface includes a play button and buttons for 'Prettify', 'Merge', 'Copy', and 'History'.

Rest read all books

GET

/books Returns all the books

⌵

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:4000/books' \
-H 'accept: application/json'
```

Request URL

```
http://localhost:4000/books
```

Server response

Code

Details

200

Response body

```
{
  "name": "Harry Potter and the Chamber of Secrets",
  "authorId": 1
},
{
  "id": 2,
  "name": "Harry Potter and the Prisoner of Azkaban",
  "authorId": 1
},
{
  "id": 3,
  "name": "Harry Potter and the Goblet of Fire",
  "authorId": 1
},
{
  "id": 4,
  "name": "The Merchant of Venice",
  "authorId": 2
},
{
  "id": 5,
  "name": "Antony and Cleopatra",
  "authorId": 2
},
{
  "id": 6,
  "name": "Geetanjali",
  "authorId": 3
}
]
```

Download

Rest read books by id

GET

/books/{id} Returns a single book by ID

⌵

Parameters

Cancel

Name	Description
id <small>* required</small>	ID of the book to get
integer (path)	<input type="text" value="2"/>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:4000/books/2' \
-H 'accept: application/json'
```

Request URL

```
http://localhost:4000/books/2
```

Server response

Code

Details

200

Response body

```
{
  "id": 2,
  "name": "Harry Potter and the Prisoner of Azkaban",
  "authorId": 1
}
```

Download

Rest create books

POST

/books

Adds a new Book

Parameters

Cancel

Reset

No parameters

Request body

required

application/json

```
{
  "id": 7,
  "title": "new",
  "author": "3"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:4000/books' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 7,
    "title": "new",
    "author": "3"
  }'
```

Request URL

http://localhost:4000/books

Server response

Code

Details

200

Response body

```
{
  "id": 7,
  "title": "new",
  "author": "3"
}
```

Download

Rest update books

PUT

/books/{id} Update a book

Cancel

Reset

Parameters

Name	Description
id * required integer (path)	ID of the book to update

7

Request body * required

application/json

```
{
  "title": "updated new book",
  "author": "3"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'PUT' \
  'http://localhost:4000/books/7' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "title": "updated new book",
    "author": "3"
  }'
```

Request URL

http://localhost:4000/books/7

Server response

Code	Details
200	<div>Response body</div> <pre>{ "id": 7, "title": "updated new book", "author": "3" }</pre> <div>Download</div>

Rest delete books

DELETE

/books/{id} Delete a book

Parameters

Cancel

Name	Description
id * required integer (path)	ID of the book to delete

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
'http://localhost:4000/books/7' \
-H 'accept: */*'
```

Request URL

```
http://localhost:4000/books/7
```

Server response

Code	Details
200	<div>Response body<div>Book deleted successfully</div><div>CopyDownload</div></div>