

1. Create User

The screenshot shows the GraphQL playground interface with the URL `http://localhost:4000`. The sidebar on the left has sections for Development, Schema, and Governance. The main area shows an 'Operation Collections' section with a 'User CRUD Task' entry containing five operations: CreateUser, GetUsers, GetUser, UpdateUser, and DeleteUser. The 'CreateUser' operation is selected, showing its GraphQL code:

```
1 mutation {
2   createUser(firstName: "newFirstName", lastName: "newLastName", age: 20) {
3     id
4     firstName
5   }
6 }
```

The 'Response' tab shows the JSON response from the server:

```
{
  "data": {
    "createUser": {
      "id": "698990e67459729376a9c27e",
      "firstName": "newFirstName"
    }
  }
}
```

2. Get Users

The screenshot shows the GraphQL playground interface with the URL `http://localhost:4000`. The sidebar on the left has sections for Development, Schema, and Governance. The main area shows an 'Operation Collections' section with a 'User CRUD Task' entry containing five operations: CreateUser, GetUsers, GetUser, UpdateUser, and DeleteUser. The 'GetUsers' operation is selected, showing its GraphQL code:

```
1 query {
2   getUsers {
3     id
4     firstName
5     lastName
6     age
7   }
8 }
```

The 'Response' tab shows the JSON response from the server:

```
{
  "data": {
    "getUsers": [
      {
        "id": "6989822d0ed845baa849cff5",
        "firstName": "Sarmita",
        "lastName": "V D",
        "age": 22
      },
      {
        "id": "698982690ed845baa849cffb",
        "firstName": "ABC",
        "lastName": "X Y",
        "age": 25
      },
      {
        "id": "698990e67459729376a9c27e",
        "firstName": "newFirstName",
        "lastName": "newLastName",
        "age": 20
      }
    ]
  }
}
```

3. Get User

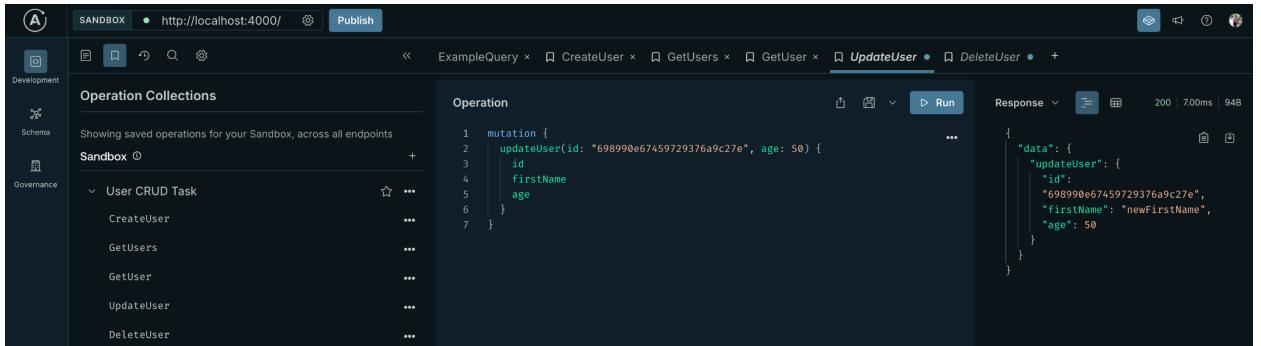
The screenshot shows the GraphQL playground interface with the URL `http://localhost:4000`. The sidebar on the left has sections for Development, Schema, and Governance. The main area shows an 'Operation Collections' section with a 'User CRUD Task' entry containing five operations: CreateUser, GetUsers, GetUser, UpdateUser, and DeleteUser. The ' GetUser' operation is selected, showing its GraphQL code:

```
1 query {
2   getUser(id: "698990e67459729376a9c27e") {
3     id
4     firstName
5     age
6   }
7 }
```

The 'Response' tab shows the JSON response from the server:

```
{
  "data": {
    "getUser": {
      "id": "698990e67459729376a9c27e",
      "firstName": "newFirstName",
      "age": 20
    }
  }
}
```

4. Update User



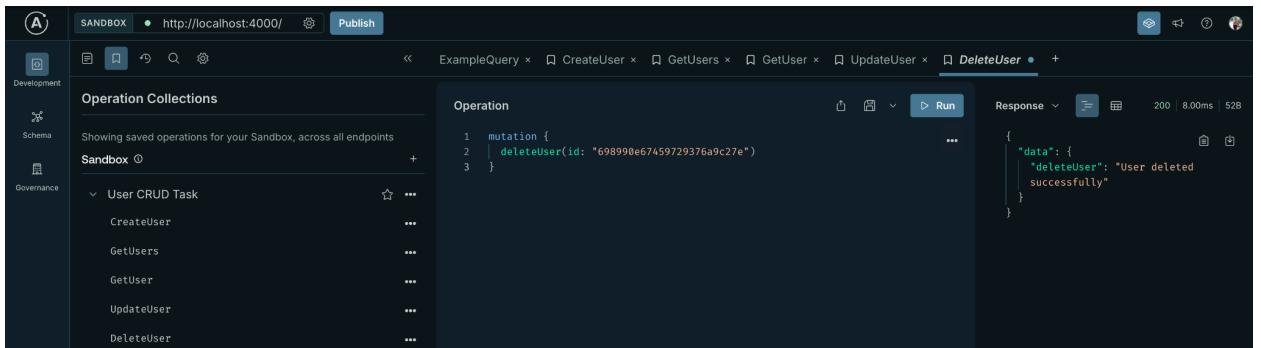
The screenshot shows the GraphQL playground interface with the URL <http://localhost:4000/>. The sidebar on the left has sections for Development, Schema, and Governance. The main area shows an "Operation Collections" section with a "Sandbox" entry expanded, revealing "User CRUD Task" with sub-options: CreateUser, GetUsers, GetUser, UpdateUser, and DeleteUser. The central "Operation" panel displays a GraphQL mutation for updating a user:

```
1 mutation {
2   updateUser(id: "698990e67459729376a9c27e", age: 50) {
3     id
4     firstName
5     age
6   }
7 }
```

The "Response" panel on the right shows the JSON response from the server:

```
{
  "data": {
    "updateUser": {
      "id": "698990e67459729376a9c27e",
      "firstName": "newFirstName",
      "age": 50
    }
  }
}
```

5. Delete User



The screenshot shows the GraphQL playground interface with the URL <http://localhost:4000/>. The sidebar on the left has sections for Development, Schema, and Governance. The main area shows an "Operation Collections" section with a "Sandbox" entry expanded, revealing "User CRUD Task" with sub-options: CreateUser, GetUsers, GetUser, UpdateUser, and DeleteUser. The central "Operation" panel displays a GraphQL mutation for deleting a user:

```
1 mutation {
2   deleteUser(id: "698990e67459729376a9c27e")
3 }
```

The "Response" panel on the right shows the JSON response from the server:

```
{
  "data": {
    "deleteUser": "User deleted successfully"
  }
}
```