

APOLLO SANDBOX http://localhost:4000/graphiql Publish

Documentation

Root > Query

Fields ↓

- empty: String
- categories: [Category!]
- category(...): Category
- me: User
- order(...): Order
- product(...): Product
- products(...): ProductConnection!

Operation

```
query {
  categories {
    id
    name
    slug
  }
}
```

Variables Headers Pre-Operation Script Post-Operation Script JSON

1

+ Add files

Response 200 | 51ms | 70B

```
[{"category": [
  {"id": "6919718e13681d23df4201ea",
   "name": "Electronics",
   "slug": "electronics"
  },
  {"id": "6919718e13681d23df4201eb",
   "name": "Mobiles",
   "slug": "mobiles"
  },
  {"id": "6919718e13681d23df4201ec",
   "name": "Laptops",
   "slug": "laptops"
  },
  {"id": "6919718e13681d23df4201ed",
   "name": "Fashion",
   "slug": "fashion"
  },
  {"id": "6919718e13681d23df4201ef",
   "name": "Beauty",
   "slug": "beauty"
  },
  {"id": "6919718e13681d23df4201f",
   "name": "Home Appliances",
   "slug": "home-appliances"
  },
  {"id": "6919718e13681d23df4201f0",
   "name": "Sports",
   "slug": "sports"
  },
  {"id": "6919718e13681d23df4201f1",
   "name": "Books",
   "slug": "books"
  },
  {"id": "6919718e13681d23df4201f2",
   "name": "Furniture",
   "slug": "furniture"
  },
  {"id": "6919718e13681d23df4201f3",
   "name": "Groceries",
   "slug": "groceries"
  }
]}
```

APOLLO SANDBOX http://localhost:4000/graphiql Publish

Documentation

Root > Query

Fields ↓

- empty: String
- categories: [Category!]
- category(...): Category
- me: User
- order(...): Order
- product(...): Product
- products(...): ProductConnection!

Operation

```
query {
  products(first: 10) {
    edges {
      node {
        id
        name
        price
        category {
          name
        }
      }
    }
  }
}
```

Variables Headers Pre-Operation Script Post-Operation Script JSON

1

+ Add files

Response 200 | 93ms | 12B

```
[{"data": {
  "products": [
    {
      "edges": [
        {
          "node": {
            "id": "6919718e13681d23df4201ff",
            "name": "Organic Rice 5kg",
            "price": "25",
            "category": {
              "name": "Groceries"
            }
          }
        }
      ]
    }
  ]
}, {
  "edges": [
    {
      "node": {
        "id": "6919718e13681d23df4201fd",
        "name": "Electric Kettle 1.5L",
        "price": "75",
        "category": {
          "name": "Home Appliances"
        }
      }
    }
  ],
  "node": {
    "id": "6919718e13681d23df4201fb",
    "name": "Office Table",
    "price": "350",
    "category": {
      "name": "Furniture"
    }
  }
}, {
  "edges": [
    {
      "node": {
        "id": "6919718e13681d23df4201fc",
        "name": "Nivea Skin Care Kit",
        "price": "25",
        "category": {
          "name": "Beauty"
        }
      }
    }
  ],
  "node": {
    "id": "6919718e13681d23df4201fe",
    "name": "Adda Noodle",
    "price": "80",
    "category": {
      "name": "Food"
    }
  }
}]]
```

APOLLO Go to graph index SANDBOX http://localhost:4000/graphiql Publish

Documentation

Root > Query

Fields ↓

- empty: String
- categories: [Category!]
- category(...): Category
- me: User
- order(...): Order
- product(...): Product
- products(...): ProductConnection!

Operation

```
query {
  categories {
    id
    name
  }
}
```

Variables Headers Pre-Operation Script Post-Operation Script JSON

1

+ Add files

Response 200 | 57ms | 548B

```
[{"data": {
  "categories": [
    {
      "id": "6919718e13681d23df4201ea",
      "name": "Electronics"
    },
    {
      "id": "6919718e13681d23df4201eb",
      "name": "Mobiles"
    },
    {
      "id": "6919718e13681d23df4201ec",
      "name": "Laptops"
    },
    {
      "id": "6919718e13681d23df4201ed",
      "name": "Fashion"
    },
    {
      "id": "6919718e13681d23df4201ee",
      "name": "Beauty"
    },
    {
      "id": "6919718e13681d23df4201ef",
      "name": "Home Appliances"
    },
    {
      "id": "6919718e13681d23df4201f0",
      "name": "Sports"
    },
    {
      "id": "6919718e13681d23df4201f1",
      "name": "Books"
    }
  ]
}]]
```

APOLLO

SANDBOX • http://localhost:4000/graphi • Publish

UNNAMED × UNNAMED × UNNAMED × UNNAMED × +

Documentation

Root > Query

Query

Fields ↓ ● v

- _empty: String
- categories: [Category]!
- category(...): Category
- me: User
- order(...): Order
- product(...): Product
- products(...): ProductConnection!

Operation

```
1 query {  
2   category(id: "6919718e13681d23df4201ed") {  
3     id  
4     name  
5     slug  
6   }  
7 }  
8
```

Variables Headers Pre-Operation Script Post-Operation Script JSON

1

+ Add files

200 | 42.0ms | 908

This screenshot shows the Apollo GraphQL playground interface. The left sidebar has sections for Development (Explorer, Schema definition, Schema diff, Schema document, Visualization), Governance (Checks), and a collapsed section. The main area shows a query for a category by ID. The response is a JSON object containing a single category with fields id, name, and slug.

APOLLO

SANDBOX • http://localhost:4000/graphi • Publish

UNNAMED × UNNAMED × UNNAMED × UNNAMED × UNNAMED × +

Documentation

Root > Query

Query

Fields ↓ ● v

- _empty: String
- categories: [Category]!
- category(...): Category
- me: User
- order(...): Order
- product(...): Product
- products(...): ProductConnection!

Operation

```
1 query {  
2   products(first: 5) {  
3     edges {  
4       node {  
5         id  
6         name  
7       }  
8     }  
9   }  
10 }  
11
```

Variables Headers Pre-Operation Script Post-Operation Script JSON

1

+ Add files

200 | 77.0ms | 3868

This screenshot shows the Apollo GraphQL playground interface. The left sidebar has sections for Development (Explorer, Schema definition, Schema diff, Schema document, Visualization), Governance (Checks), and a collapsed section. The main area shows a query for the first 5 products. The response is a JSON object containing a list of products, each with an edge and a node field. The node field contains the product's ID and name.

APOLLO

SANDBOX • http://localhost:4000/graphi • Publish

UNNAMED × UNNAMED × UNNAMED × UNNAMED × UNNAMED × UNNAMED × +

Documentation

Root > Query

Query

Fields ↓ ● v

- _empty: String
- categories: [Category]!
- category(...): Category
- me: User
- order(...): Order
- product(...): Product
- products(...): ProductConnection!

Operation

```
1 query {  
2   product(id: "6919718e13681d23df4201fd") {  
3     id  
4     name  
5     price  
6     description  
7     category {  
8       name  
9     }  
10   }  
11 }  
12
```

Variables Headers Pre-Operation Script Post-Operation Script JSON

1

+ Add files

200 | 39.0ms | 1748

This screenshot shows the Apollo GraphQL playground interface. The left sidebar has sections for Development (Explorer, Schema definition, Schema diff, Schema document, Visualization), Governance (Checks), and a collapsed section. The main area shows a query for a specific product by ID. The response is a JSON object containing the product's details, including its ID, name, price, description, and its category name.

⚠ Please check your email and follow the link to verify your email address. Resend verification email.

The screenshot shows the Apollo GraphQL DevTools interface. On the left, the sidebar has sections for DEVELOPMENT (Explorer, Schema definition, Schema diff, Schema document, Visualization), GOVERNANCE (Checks), and a collapsed section. The main area has tabs for SANDBOX (http://localhost:4000/graphi) and Publish. The SANDBOX tab is active. In the Operation panel, a mutation is defined:

```
1 mutation {  
2   login(email: "user@example.com", password: "customer")  
3 }  
4
```

The Response panel shows a successful response with status 200, 439ms, and 218B. The JSON response is:

```
{  
  "data": {  
    "login": "eyJhbGciOiJIUzI1NiJnR5cC16IkpxVCJ9.  
eyJzdW1tOlZOTES0DRhZWzjMzDUN0EwlmhKCYLLCjybzxL1joQ  
1VTvF9MRV1l1CjYXQ1oE3NjMyODAMDAStAvccCIGtC2Mzg4NDkwMH  
0.OuMIQbuh1RwMCNiHjzTn1m6R2RwBaMs_y2duHZ9xo"  
  }  
}
```

The screenshot shows the Apollo GraphQL DevTools interface. The sidebar is identical to the first screenshot. The main area has tabs for SANDBOX (http://localhost:4000/graphi) and Publish. The SANDBOX tab is active. In the Operation panel, a query is defined:

```
1 query {  
2   me {  
3     id  
4     email  
5     name  
6     role  
7     createdat  
8   }  
9 }  
10
```

The Response panel shows a successful response with status 200, 36.0ms, and 155B. The JSON response is:

```
{  
  "data": {  
    "me": {  
      "id": "691984aea0f3fd547a06cd3f",  
      "email": "user@example.com",  
      "name": "Normal User",  
      "role": "CUSTOMER",  
      "createdAt": "2025-11-16T08:00:46.056Z"  
    }  
  }  
}
```

The screenshot shows the Apollo GraphQL DevTools interface. The sidebar is identical to the previous screenshots. The main area has tabs for SANDBOX (http://localhost:4000/graphi) and Publish. The SANDBOX tab is active. In the Operation panel, a query is defined:

```
1 query {  
2   categories {  
3     id  
4     name  
5     slug  
6   }  
7 }
```

The Response panel shows a successful response with status 200, 270ms, and 730B. The JSON response is:

```
{  
  "data": {  
    "categories": [  
      {  
        "id": "691984aea0f3fd547a06cd42",  
        "name": "Electronics",  
        "slug": "electronics"  
      },  
      {  
        "id": "691984aea0f3fd547a06cd43",  
        "name": "Mobiles",  
        "slug": "mobiles"  
      },  
      {  
        "id": "691984aea0f3fd547a06cd44",  
        "name": "Laptops",  
        "slug": "laptops"  
      },  
      {  
        "id": "691984aea0f3fd547a06cd45",  
        "name": "Fashion",  
        "slug": "fashion"  
      },  
      {  
        "id": "691984aea0f3fd547a06cd46",  
        "name": "Beauty",  
        "slug": "beauty"  
      },  
      {  
        "id": "691984aea0f3fd547a06cd47",  
        "name": "Home Appliances",  
        "slug": "home-appliances"  
      },  
    ]  
  }  
}
```

The screenshot shows the Apollo GraphQL DevTools interface. The left sidebar includes sections for Development, Explorer, Schema (Schema definition, Schema diff, Schema document), Visualization, Governance, and Checks. The main area has tabs for Documentation, Operation, and Response. The Operation tab displays a GraphQL query for fetching products. The Response tab shows the JSON output of the query, which includes product details like name, price, and category, along with their IDs and stock status. The bottom navigation bar includes tabs for Variables, Headers, Pre-Operation Script, Post-Operation Script, and a header input field containing a Bearer token.

```
query { products(first: 5) { totalCount edges { node { id name price inStock } category { name } } } }
```

```
{ "data": { "products": { "totalCount": 10, "edges": [ { "node": { "id": "691984aea0f3fd547a06cd57", "name": "Organic Rice 5kg", "price": "19", "inStock": true, "category": { "name": "Groceries" } } }, { "node": { "id": "691984aea0f3fd547a06cd56", "name": "Electric Kettle 1.5L", "price": "25", "inStock": true, "category": { "name": "Home Appliances" } } }, { "node": { "id": "691984aea0f3fd547a06cd55", "name": "Wooden Office Table", "price": "350", "inStock": true, "category": { "name": "Furniture" } } } ] } }
```

The screenshot shows the GraphQL Playground interface. The left sidebar has tabs for 'DEVELOPMENT' (selected), 'Explorer', 'SCHEMA' (with 'Schema definition', 'Schema diff', 'Schema document', and 'Visualization' options), and 'GOVERNANCE' (with 'Checks' option). The main area has tabs for 'Documentation' (selected), 'Root > Query', and 'Query' (selected). Under 'Query', there is a 'Fields' dropdown with several options: 'categories: [Category!]!', 'category(_): Category', 'product(_): Product', 'products(_): ProductConnection!', 'order(_): Order', 'me: User', and 'meCart: [CartItem!]!'. Below this is a code editor with the following query:

```
query {
  products(first: 5) {
    totalCount
    edges {
      node {
        id
        name
        price
        inStock
      }
    }
  }
}
```

The 'Run' button is highlighted in blue. To the right, the 'Response' tab shows the JSON result of the query:

```
"data": {
  "products": {
    "totalCount": 10,
    "edges": [
      {
        "node": {
          "id": "691984aea0f3fd547a06cd57",
          "name": "Organic Rice 5kg",
          "price": 19,
          "inStock": true
        }
      },
      {
        "node": {
          "id": "691984aea0f3fd547a06cd56",
          "name": "Electric Kettle 1.5L",
          "price": 25,
          "inStock": true
        }
      },
      {
        "node": {
          "id": "691984aea0f3fd547a06cd55",
          "name": "Wooden Office Table",
          "price": 350,
          "inStock": true
        }
      },
      {
        "node": {
          "id": "691984aea0f3fd547a06cd54",
          "name": "Nivea Skin Care Kit",
          "price": 29,
          "inStock": true
        }
      }
    ]
  }
}
```

Below the response are tabs for 'Variables', 'Headers' (selected), 'Pre-Operation Script', and 'Post-Operation Script'. A 'Authorization' field contains 'Bearer eyJhbGciOiJIUzI1NiI'.

The screenshot shows the Apollo GraphQL DevTools interface. The left sidebar includes sections for Development (Explorer, Schema definition, Schema diff, Schema document, Visualization), Governance (Checks), and a collapsed section. The main area has tabs for Sandbox, Documentation, and Operation. The Operation tab is active, displaying a GraphQL query:

```
query {
  products(first: 5, offset: 0) {
    totalCount
    edges {
      node {
        id
        name
        price
        category { name }
      }
    }
  }
}
```

Below the query, there are tabs for Variables, Headers, Pre-Operation Script, and Post-Operation Script. The Headers tab is selected, showing an Authorization header with the value eyJhbGciOiJIUzI1NiIsInR5cC... and a red 'Delete' button. At the bottom, there are buttons for '+ New header' and 'Set shared headers'.

The Response tab is also visible, showing a JSON response with three nodes representing products. Each node contains an id, name, price, and category (Groceries, Electric Kettle 1.5L, Furniture).

APOLLO

SANDBOX • http://localhost:4000/graphc Publish

DEVELOPMENT

Explorer

SCHEMA

Schema definition

Schema diff

Schema document

Visualization

GOVERNANCE

Checks

Documentation

Root > Query

Query

Fields

me: User

categories: [Category]!

category(...): Category

products(...): ProductConnection!

product(...): Product

order(...): Order

myOrders: [Order]!

meCart: [CartItem]!

Operation

```
1 mutation {
2   addToCart(input: { productId: "691a0365215c6351fa6db535", ...
3     quantity: 2 }) {
4     product { id name price }
5     quantity
6   }
7 }
```

Response

200 | 91.0ms | 1238

```
{
  "data": {
    "addToCart": {
      "product": {
        "id": "691a0365215c6351fa6db535",
        "name": "Electric Kettle 1.5L",
        "price": 25
      },
      "quantity": 2
    }
  }
}
```

Variables Headers Pre-Operation Script Post-Operation Script

Authorization eyJhbGciOiJIUzI1NiIsInR5cCI

New header Set shared headers

<< Collapse

This screenshot shows the Apollo GraphQL playground interface. The 'Mutation' tab is selected, displaying a mutation named 'addToCart'. The mutation takes a single argument 'input' which includes a 'productId' and a 'quantity'. The response shows a single item added to the cart with a quantity of 2. The response status is 200, and the time taken is 91.0ms.

APOLLO

SANDBOX • http://localhost:4000/graphc Publish

DEVELOPMENT

Explorer

SCHEMA

Schema definition

Schema diff

Schema document

Visualization

GOVERNANCE

Checks

Documentation

Root > Query

Query

Fields

me: User

categories: [Category]!

category(...): Category

products(...): ProductConnection!

product(...): Product

order(...): Order

myOrders: [Order]!

meCart: [CartItem]!

Operation

```
1 query {
2   meCart {
3     product { id name }
4     quantity
5   }
6 }
```

Response

200 | 32.0ms | 111B

```
{
  "data": {
    "meCart": [
      {
        "product": {
          "id": "691a0365215c6351fa6db535",
          "name": "Electric Kettle 1.5L"
        },
        "quantity": 2
      }
    ]
  }
}
```

Variables Headers Pre-Operation Script Post-Operation Script

Authorization eyJhbGciOiJIUzI1NiIsInR5cCI

New header Set shared headers

<< Collapse

This screenshot shows the Apollo GraphQL playground interface. The 'Query' tab is selected, displaying a query that retrieves the 'meCart' field. The response shows a single item in the cart with a quantity of 2. The response status is 200, and the time taken is 32.0ms.