**# 🚀 Laravel GraphQL API with Lighthouse**

This project is a basic **\*\*GraphQL API\*\*** built using **\*\*Laravel\*\*** and the **\*\*Lighthouse\*\*** GraphQL server package. It includes CRUD operations for a `User` model and uses the GraphQL Playground (via GraphiQL) for testing queries and mutations.

---

**## 📦 Setup Instructions**

**### 1. Clone the Repository**

```bash

git clone https://github.com/<your-username>/laravel-graphql.git

cd laravel-graphql

**2. Install Dependencies**

composer install

cp .env.example .env

php artisan key:generate

**⚙️ Commands Used to Set Up**

# Create new Laravel project

laravel new laravel-graphql

# Install Lighthouse GraphQL

composer require nuwave/lighthouse

# Publish the default GraphQL schema

php artisan vendor:publish --tag=lighthouse-schema

# Publish the Lighthouse config file

php artisan vendor:publish --tag=lighthouse-config

# Optional: Install GraphiQL Playground for testing

composer require mll-lab/laravel-graphiql

**🗄️ Run Migrations**

php artisan migrate:fresh

**📡 GraphQL Schema**

User Type:

type User {

  id: ID!

  name: String!

  email: String!

  created\_at: String!

  updated\_at: String

}

**Mutations:**

type Mutation {

  createUser(name: String!, email: String!, password: String!): User

  updateUser(id: ID, email: String, password: String): User

  deleteUser(id: ID): User

}

**🧪 Testing the API**

**Start the Laravel server:**

php artisan serve

**Then open the GraphiQL UI (if installed):**

📍 http://127.0.0.1:8000/graphiql

**📌 GraphQL Examples (with Comments)**

These examples demonstrate how to interact with the GraphQL API using queries and mutations.

**📖 Read: Get Paginated List of Users**

query MyQuery {

  users(first: 2, page: 5) {

    data {

      id

      email

      name

      created\_at

      updated\_at

    }

    paginatorInfo {

      firstItem     # Index of the first item on the current page

      total         # Total number of users in the database

      hasMorePages  # Indicates if there are more pages

    }

  }

}

**🔍 Read: Search User by ID**

query {

  userById(id: 2) {

    id

    name

    email

  }

}

**➕ Create: Add New User**

mutation {

  createUser(

    input: {

      name: "test name",

      email: "name@gmail.com",

      password: "password"

    }

  ) {

    id

    name

    email

  }

}

**✏️ Update: Modify Existing User by ID**

mutation {

  updateUser(

    id: 12,

    input: {

      name: "demo-updated",

      password: "password"

    }

  ) {

    id

    name

    email

  }

}

**❌ Delete: Remove User by ID**

mutation {

  deleteUser(id: 12) {

    id

    name

    email

  }

}

**🧠 Credits**

**Built with ❤️ using:**

**Laravel**

**Lighthouse**

**GraphiQL**

**```**