



↪ Apollo Server

By the end of this lesson, developers will be able to:

- Create a GraphQL API server with Apollo Server Express
- Create a MongoDB database connection with Connect Mongo
- Create environment variables for your localhost server

Introduction

In the previous section, we wrote a strongly typed schema for our GraphQL API server. In this section we will complete the implementation with Apollo Server and Express. Apollo Server Express allows us to build a GraphQL query interface on top of an existing Express server.

Let's begin with an overview of these libraries and their dependencies:

- **Apollo Server**: An open-source GraphQL server that is compatible with any kind of GraphQL client. We won't be using Express for our server in this project. Instead, we will use the power of Apollo Server to expose our GraphQL API.
- **Express**: Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.
- **dotenv**: We will use dotenv to load environment variables from our .env file.
- **nodemon**: A tool that helps develop Node-based applications by automatically restarting the node application when changes in the directory are detected. We don't want to be closing and starting the server every time there's a change in our code. Nodemon inspects changes every time in our app and automatically restarts the server.

↪ Installation

Heads Up! This lesson's code edits will only affect your *API* project.

First, change directory into your root level **api** folder:

```
cd api
```

Let's begin with the installation of Apollo Server Express:

```
npm install apollo-server-express cors ts-node
npm install --save-dev nodemon
```