

Line-by-Line Explanation of MongoDB CRUD with Express.js and Mongoose

This guide provides a comprehensive, line-by-line breakdown of building a REST API using Node.js, Express.js, MongoDB, and Mongoose. Each section describes key concepts, setup, and functionality.

The application performs CRUD operations on user data, demonstrating how to use Mongoose models and Express routes.

Step 1: Setup and Install Dependencies

1. Create a new project directory and initialize it:

```
$ mkdir express-mongoose-crud
```

```
$ cd express-mongoose-crud
```

```
$ npm init -y
```

2. Install required packages:

```
$ npm install express mongoose body-parser
```

- express: A web framework for handling routes.
- mongoose: An ODM library for MongoDB.
- body-parser: Middleware to parse JSON request bodies.

Step 2: Setting Up Express and Mongoose

In index.js:

```
const express = require('express');
```

```
const mongoose = require('mongoose');
```

```
const bodyParser = require('body-parser');
```

Initialize express, connect to MongoDB using mongoose, and set up body-parser middleware.

`mongoose.connect('mongodb://localhost:27017/testDB', {...})` connects to the MongoDB database.

`app.listen(...)` starts the server on port 3000.

Step 3: Define Mongoose Schema and Model (models/User.js)

Create a User schema and model using Mongoose:

```
const userSchema = new mongoose.Schema({ name: String, age: Number, email: String })
```

```
module.exports = mongoose.model('User', userSchema);
```

Schema defines the structure, while model provides the interface to MongoDB.

Step 4: CRUD Routes in index.js

Routes handle CRUD operations for 'User':

- POST /users: Creates a user with data in req.body, saves to MongoDB.
- GET /users: Fetches all users from MongoDB.
- GET /users/:id: Fetches user by ID.
- PUT /users/:id: Updates user by ID using req.body data.

- DELETE /users/:id: Deletes user by ID.

Each route uses Mongoose methods (save, find, findById, findByIdAndUpdate, findByIdAndDelete).

Step 5: Testing the API

Start the server with:

```
$ node index.js
```

Use Postman or cURL to test each endpoint:

- Create user: POST /users with JSON body {name, age, email}.
- Get all users: GET /users
- Get user by ID: GET /users/{id}
- Update user: PUT /users/{id} with JSON body.
- Delete user: DELETE /users/{id}