Complete Guide: Deploying NestJS API to Render

Overview

This document provides a step-by-step guide to deploying a NestJS backend API to Render using GitHub and MongoDB Atlas.

Tech Stack

- Backend: NestJS
- Database: MongoDB Atlas
- Hosting: Render
- Code Repository: GitHub

Step 1: Prepare NestJS Project

1. Install dependencies:

```
npm install
```

2. Ensure **package.json** contains the following scripts:

```
"build": "nest build",
"start": "nest start",
"start:dev": "nest start --watch",
"start:prod": "node dist/main.js"
```

3. Configure **main.ts** to use a dynamic port:

```
const port = process.env.PORT || 3000;
```

Step 2: Configure Environment Variables

1. Install configuration package:

```
npm install @nestjs/config
```

2. Configure **app.module.ts**:

```
ConfigModule.forRoot({ isGlobal: true });
```

3. Use MongoDB connection string with Mongoose:

```
MongooseModule.forRoot(process.env.MONGODB_URI)
```

Step 3: Push Code to GitHub

```
git init
git add .
git commit -m "Initial commit"
git remote add origin https://github.com/yourusername/yourrepo.git
git push -u origin main
```

Step 4: Create Web Service in Render

1. Log in to Render
2. Click **New** → **Web Service**
3. Connect your GitHub repository
4. Select the NestJS project

Step 5: Configure Render Settings

- Environment: Node
- Build Command: `npm install && npm run build`
- Start Command: `npm run start:prod`
- Instance Type: Free or Paid

Step 6: Add Environment Variables in Render

Key: **MONGODB_URI**
Value: MongoDB Atlas connection string

Step 7: Configure MongoDB Atlas

1. Go to **Network Access**
2. Add IP Address
3. Allow access from anywhere: `0.0.0.0/0`

Step 8: Deployment Result

The API is successfully deployed and accessible at:
https://green-co-api.onrender.com

Deployment Flow:
Client → Render (NestJS API) → MongoDB Atlas