Coffee Store

Espresso-Emporium

Start Project - Server Side

⏯ Step - 1

---

1. Open **cmd** (win + R)
2. mkdir Coffee-store-server
3. cd Coffee-store-server
4. npm init -y
5. npm i express cors mongodb dotenv
6. code .

⏯ Step - 2

---

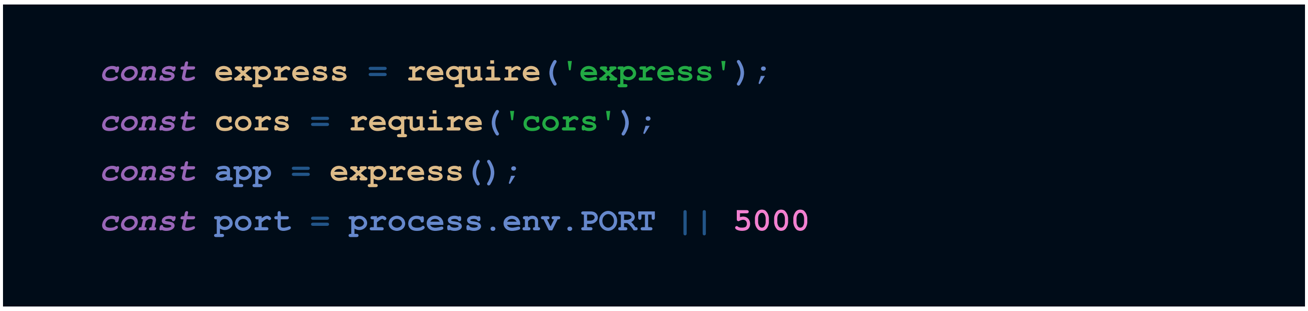
1. Create index.js
2. Open package.json
3. Write “start” : **“node index.js”** in **“Script”**

⏯ Step - 3

---

1. Open index.js

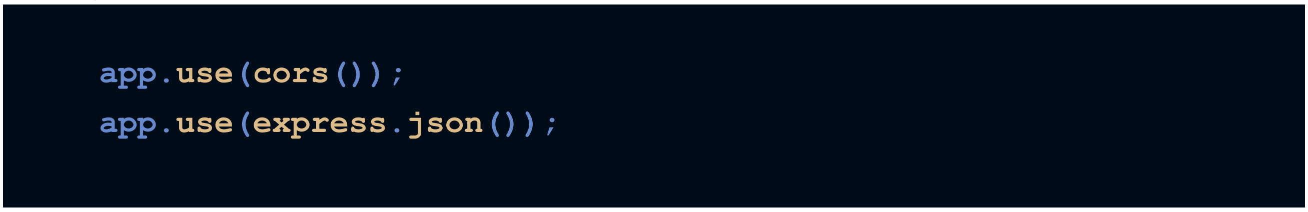
index.js



```
const express = require('express');
const cors = require('cors');
const app = express();
const port = process.env.PORT || 5000
```

➢ **MiddleWare**

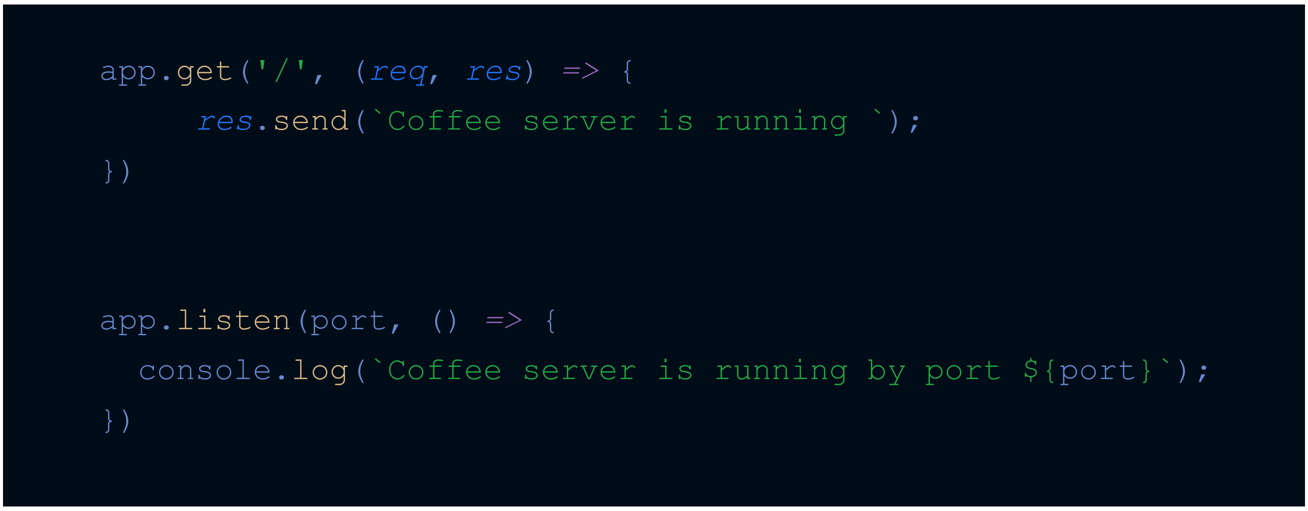
index.js



```
app.use(cors());
app.use(express.json());
```

⏯ Step - 4

index.js



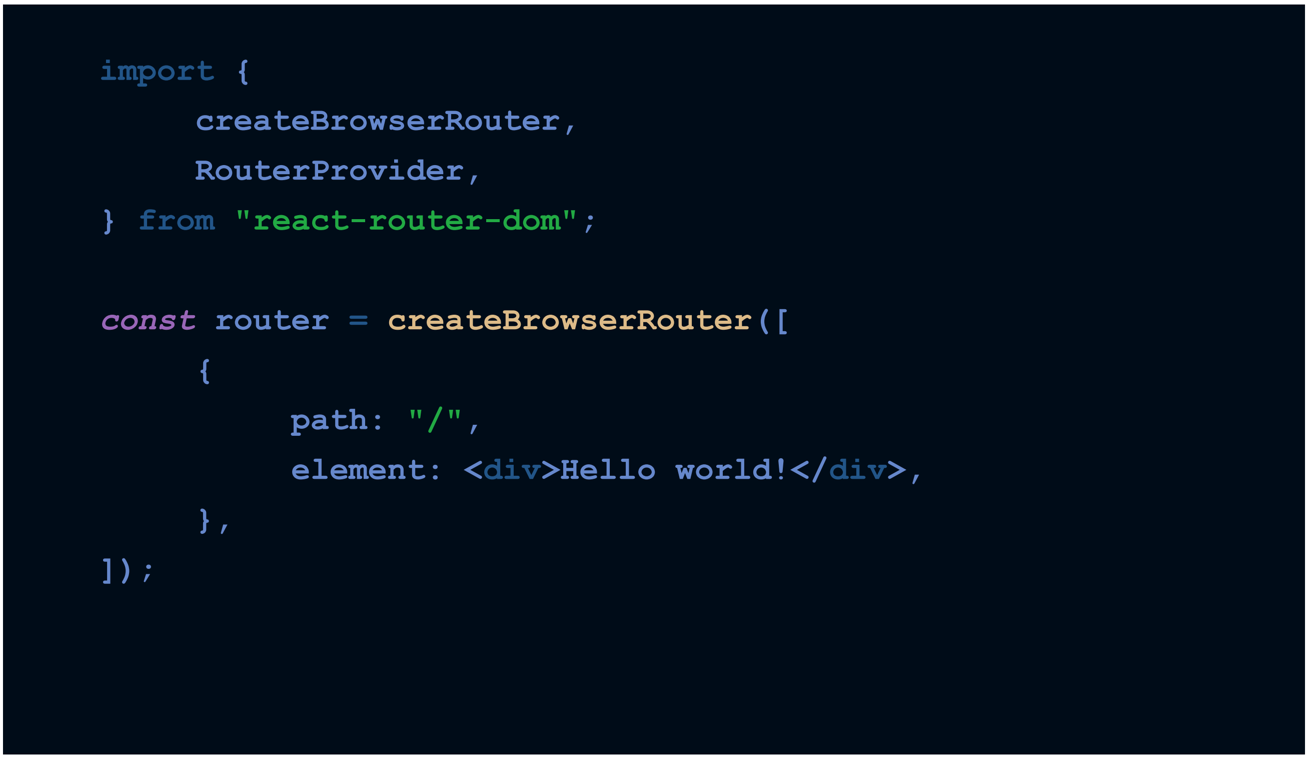
```
app.get('/', (req, res) => {
     res.send(`Coffee server is running `);
})


app.listen(port, () => {
  console.log(`Coffee server is running by port ${port}`);
})
```

Start Project - Client Side

⏯ Step - 1

1. Open cmd (win + R)
2. npm create vite@latest coffee-store-client - -   - -template react
3. cd coffee-store-client
4. npm install react-router-dom localforage match-sorter sort-by

src/Main.jsx



```
import {
     createBrowserRouter,
     RouterProvider,
} from "react-router-dom";

const router = createBrowserRouter([
     {
          path: "/",
          element: <div>Hello world!</div>,
     },
]);
```

```
createRoot(document.getElementById('root')).render(
     <StrictMode>
          <RouterProvider router={router} />
     </StrictMode>,
)
```

⏯ Step - 2 (*Tailwind Installation*)

1. npm install -D tailwindcss postcss autoprefixer
2. npx tailwindcss init -p
3. Configure template path in tailwind-config.js

```
content: [
     "./index.html",
     "./src/**/*.{js,ts,jsx,tsx}",
],
```

5. Open index.css

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

⏯ Step - 3 (*DaisyUi Installation*)

1. Npm i -D daisyui@latest
2. Insert ⤵ in Tailwind.config.js

tailwind.config.js

```
plugins: [
     require('daisyui'),
],
```

Create Database - MongoDB

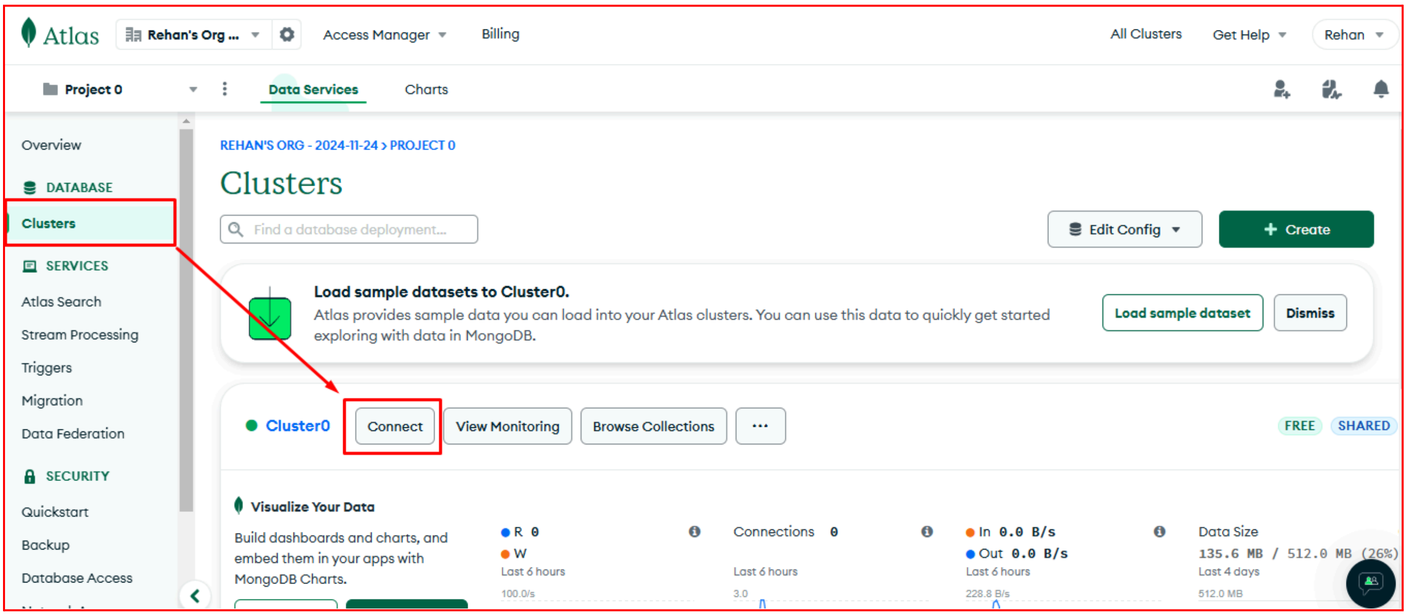
⏯ Step - 1 (*Setup Database*)

1. Goto mongoDB Atlas Database website and login your account
2. Click Database Access from side menu
3. Click Add New Database User
4. Authentication Method (Password)
5. Password Authentication

   Enter user-name
   Blank enter password field
   Click autogenerate secure Password

6. Copy user name and Password
7. Scroll down 🔻 and Click Add Built in Role
8. Select Atlas Admin
9. Click Add User Button

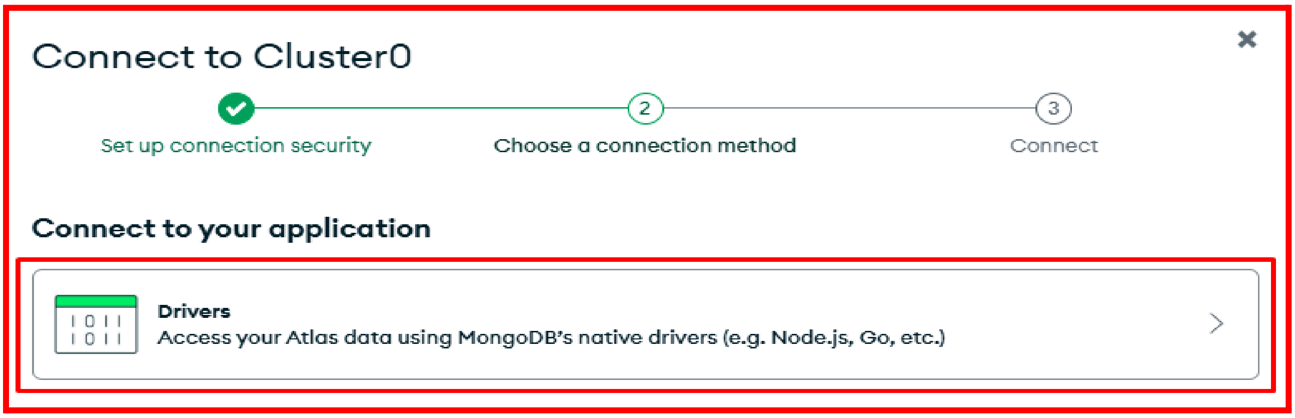
10. Click Clusters From Side Menu bar
11. Click Connect



Load sample datasets to Cluster0.
Atlas provides sample data you can load into your Atlas clusters. You can use this data to quickly get started exploring with data in MongoDB.

Visualize Your Data
Build dashboards and charts, and embed them in your apps with MongoDB Charts.

12. Select Drivers

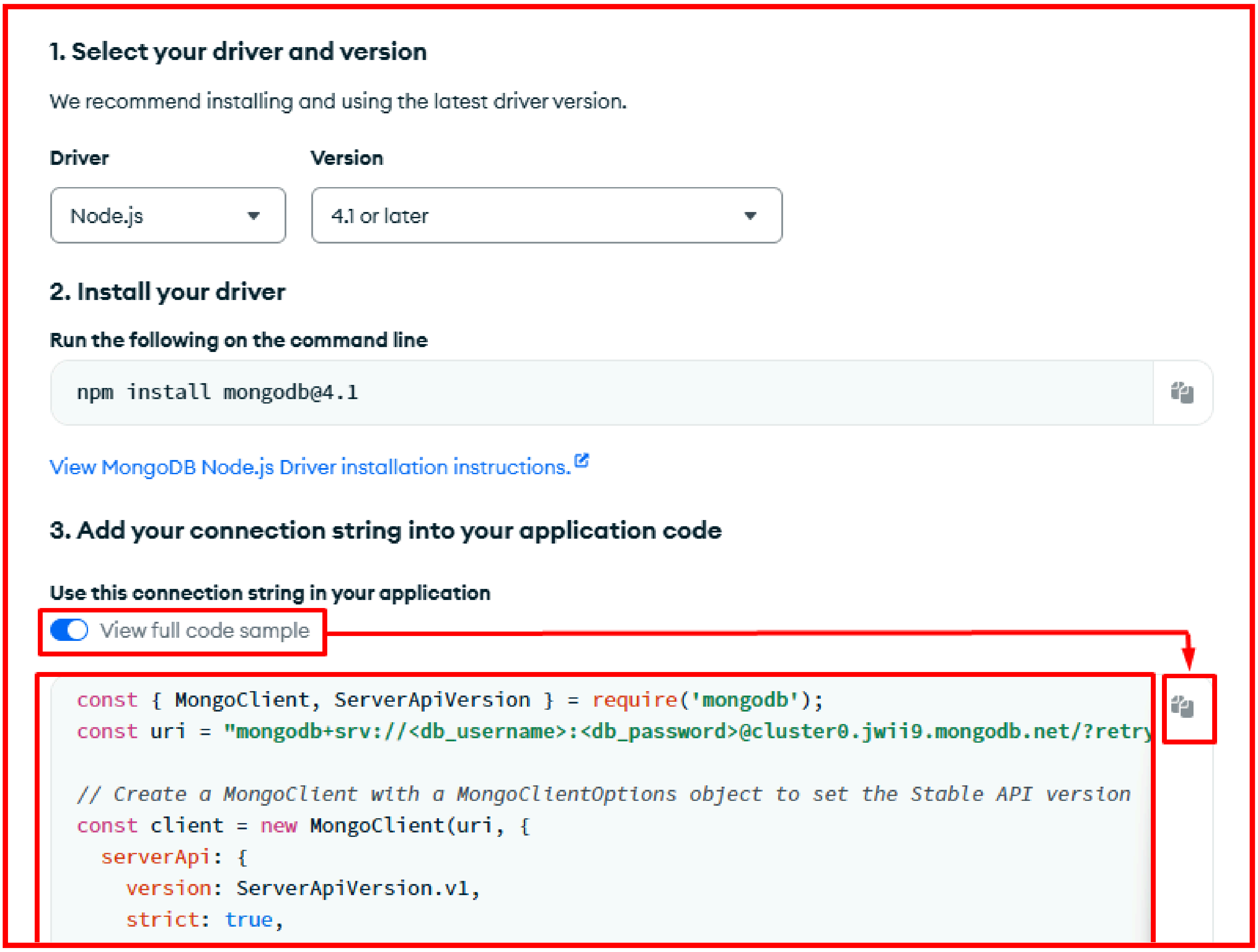


Connect to Cluster0

Connect to your application

Drivers
Access your Atlas data using MongoDB's native drivers (e.g. Node.js, Go, etc.)

13. Click on Code Switch Button and Copy code and Click Done Button



1. Select your driver and version

We recommend installing and using the latest driver version.

Driver: Node.js

Version: 4.1 or later

2. Install your driver

Run the following on the command line

```
npm install mongodb@4.1
```

View MongoDB Node.js Driver installation instructions.

3. Add your connection string into your application code

Use this connection string in your application

● View full code sample

```
const { MongoClient, ServerApiVersion } = require('mongodb');
const uri = "mongodb+srv://<db_username>:<db_password>@cluster0.jwii9.mongodb.net/?retry

// Create a MongoClient with a MongoClientOptions object to set the Stable API version
const client = new MongoClient(uri, {
  serverApi: {
    version: ServerApiVersion.v1,
    strict: true,
```

14. Paste code in Server side index.js file

⏯ Step - 2 (*Hide Database user & password from Server side by Dotenv Config*)

index.js

```
    require('dotenv').config()
```

⏯ Step - 3 (*Create .env file in server side project root folder*)

.env

```
    // Added Database username and password
    DB_USER=coffeeStore
    DB_PASSWORD=KmoM7S81oFKC5VTz
```

⏯ Step - 4 (*Insert user and password in uri variable with Template String*)

index.js

```
const uri =
`mongodb+srv://${process.env.DB_USER}:${process.env.DB_PASSWORD}@cluster
0.jwii9.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0`;
```

⏯ Step - 5 (*Hide file from git init for secure Database*)

1. Create **.gitignore** file in server side project folder

.gitignore

```
    node_modules
    .env
```