MERN Stack Installation Guide
MongoDB, Express.js, React.js, Node.js

Comprehensive Setup Guide

October 7, 2025

Contents

1 Introduction

The MERN stack is a popular JavaScript stack for building full-stack web applications. It consists of:

- **MongoDB** - Document database

- **Express.js** - Node.js web framework

- **React.js** - Client-side JavaScript framework

- **Node.js** - JavaScript runtime environment

2 Prerequisites

Before starting the installation, ensure you have:

- A Windows/Mac/Linux computer with administrative access

- Stable internet connection

- Visual Studio Code (VS Code) installed

3 Installing Node.js

1. Visit `https://nodejs.org`

2. Download the LTS (Long Term Support) version

3. Run the installer:

```
Terminal
node –version
npm –version
```

4 Installing MongoDB

4.1 Windows Installation

1. Download MongoDB Community Server from `https://www.mongodb.com/try/download/community`

2. Run the installer (.msi file)

3. Select "Complete" installation

4. Install MongoDB Compass (GUI tool)

5. Create data directory:

```
1    md "\data\db" at C:\
2
```

4.2 macOS Installation

Using Homebrew:

```
1 brew tap mongodb/brew
2 brew install mongodb-community
3 brew services start mongodb-community
```

4.3 Linux Installation

Ubuntu/Debian:

```
1 wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-
    key add -
2 echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal
    /mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/
    mongodb-org-6.0.list
3 sudo apt update
4 sudo apt install -y mongodb-org
5 sudo systemctl start mongod
6 sudo systemctl enable mongod
```

5 Project Setup

5.1 Creating Project Structure

```
1 mkdir my-mern-project
2 cd my-mern-project
```

5.2 Backend Setup

```
1 mkdir backend
2 cd backend
3 npm init -y
4 npm install express mongoose dotenv cors nodemon
```

Create `server.js`:

```
1 const express = require('express');
2 const cors = require('cors');
3 const mongoose = require('mongoose');
4 require('dotenv').config();
5
6 const app = express();
7 const port = process.env.PORT || 5000;
8
9 app.use(cors());
10 app.use(express.json());
11
12 mongoose.connect(process.env.MONGODB_URI);
13
14 app.listen(port, () => {
15   console.log(`Server running on port ${port}`);
16 });
```

5.3   Frontend Setup

```
1 npx create-react-app frontend
2 cd frontend
3 npm install axios react-router-dom
```

6   VS Code Extensions

Install the following extensions:

- ES7+ React/Redux/React-Native snippets

- JavaScript (ES6) code snippets

- Prettier - Code formatter

- ESLint

- MongoDB for VS Code

7   Running the Application

7.1   Starting Backend

```
1 cd backend
2 npm run dev
```

7.2   Starting Frontend

```
1 cd frontend
2 npm start
```

8   Troubleshooting

8.1   Common Issues

- **MongoDB Connection Issues**
  - Check if MongoDB service is running
  - Verify database connection string
  - Check network connectivity

- **Port Conflicts**
  - Default ports: 3000 (React), 5000 (Express)
  - Check for processes using these ports

- **npm Install Errors**

```
1     npm cache clean --force
2     rm -rf node_modules
3     npm install
4
```

9   Project Structure

Final project structure:

```
1  my-mern-project/
2        backend/
3              node_modules/
4              models/
5              routes/
6              config/
7              package.json
8              server.js
9        frontend/
10             node_modules/
11             public/
12             src/
13             package.json
14             README.md
```

10   Next Steps

1. Create MongoDB schemas

2. Set up API routes

3. Create React components

4. Implement authentication

5. Add error handling

6. Deploy your application

11   Resources

- MongoDB Documentation: https://docs.mongodb.com/

- Express.js Documentation: https://expressjs.com/

- React Documentation: https://reactjs.org/

- Node.js Documentation: https://nodejs.org/docs/