Build and Deploy Full Stack MongoDB, Express, React, and Node.js Apps

Are you ready to take your full stack web development skills to the next level? In this comprehensive guide, we'll walk you through every step of building and deploying a full stack MongoDB, Express, React, and Node.js (MERN) application.

Beginning MERN Stack: Build and Deploy a Full Stack MongoDB, Express, React, Node.js App by Greg Lim

★★★★☆ 4.4 out of 5

Language	: English
File size	: 11338 KB
Text-to-Speech	: Enabled
Screen Reader	: Supported
Enhanced typesetting	: Enabled
Print length	: 161 pages
Lending	: Enabled

We'll start by introducing the MERN stack and its components. Then, we'll dive into building a simple MERN application from scratch. We'll cover everything from setting up the database to creating the user interface. Once your application is complete, we'll show you how to deploy it to a live server.

By the end of this guide, you'll have the skills and confidence to build and deploy your own full stack MERN applications.

What is the MERN Stack?

The MERN stack is a popular JavaScript framework for building full stack web applications. It consists of the following components:

- **MongoDB**: A NoSQL database that stores data in a flexible, document-oriented format.
- **Express**: A web framework for Node.js that provides a simple and flexible way to create HTTP servers.
- **React**: A JavaScript library for building user interfaces.
- **Node.js**: A JavaScript runtime that allows you to run JavaScript on the server.

The MERN stack is a powerful combination of technologies that can be used to build a wide variety of web applications. It's a good choice for applications that require a flexible database, a fast and efficient web framework, and a modern user interface.

Building a Simple MERN Application

Now that you know what the MERN stack is, let's build a simple application to get started. We'll create a to-do list application that allows users to add, delete, and mark tasks as complete.

To start, you'll need to install the following software:

- Node.js
- MongoDB
- Express

- React

Once you have all of the software installed, you can create a new directory for your project and open it in your code editor. Then, create a new file called `package.json` and add the following code:

```
{ "name": "mern-todo", "version": "1.0.0", "description": "A
simple to-do list application built with the MERN stack.",
"main": "index.js", "scripts": { "start": "node index.js",
"dev": "nodemon index.js" }, "dependencies": { "express":
"^4.17.1", "mongodb": "^3.6.8", "mongoose": "^5.11.6",
"react": "^17.0.2", "react-dom": "^17.0.2" }}
```

This file contains the metadata for your project, including the name, version, description, and dependencies. The `dependencies` object specifies the packages that your project requires. In this case, we're using the `express`, `mongodb`, `mongoose`, `react`, and `react-dom` packages.

Next, create a new file called `index.js` and add the following code:

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

```
const app = express();
```

```
mongoose.connect('mongodb://localhost/mern-todo',
{useNewUrlParser: true, useUnifiedTopology: true });
```

```
app.get('/', (req, res) => { res.send('Hello World!'); });
app.listen(3000, () => { console.log('Server is listening on port 3000'); });
```

This file contains the code for our server. We're using the `express` package to create a web server and the `mongoose` package to connect to the MongoDB database. We also define a route that responds to GET requests to the root URL with the message "Hello World!".

Finally, create a new directory called `client` and add the following code to a file called `index.js`:

```
import React from 'react'; import ReactDOM from 'react-dom';
```

```
const App = () => { return (
```

```
); };
```

```
ReactDOM.render(, document.getElementById('root'));
```

This file contains the code for our client-side application. We're using the `react` and `react-dom` packages to create a simple React application that displays the message "Hello World!".

To run your application, open a terminal window and navigate to your project directory. Then, type the following command:

```
npm start
```

This will start the development server and open your application in a web browser. You should see the message "Hello World!" displayed in the browser.

Deploying a MERN Application

Once you've built your MERN application, you'll need to deploy it to a live server so that other people can access it. There are many different ways to deploy a MERN application, but we'll cover the basics in this section.

First, you'll need to choose a hosting provider. There are many different hosting providers to choose from, so you'll need to research your options and find one that meets your needs. Once you've chosen a hosting provider, you'll need to create an account and set up a hosting plan.

Once you've set up your hosting account, you'll need to deploy your application to the server. The specific steps for deploying a MERN application will vary depending on your hosting provider, but the general process is as follows:

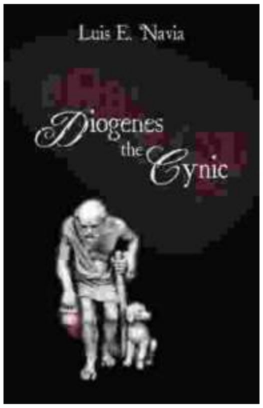
1. Upload your application files to the server.
2. Create a database on the server.
3. Configure your application to use the database.
4. Start your application.

Once your application is deployed, you'll need to test it to make sure that it's working properly. You can do this by visiting the URL of your application in a web browser. If your application is working properly, you should see the message "Hello World!" displayed in the browser.

In this guide, we've shown you how to build and deploy a full stack MERN application. We've covered everything from setting up the database to creating the user interface. Now it's your turn to put your new skills to work and build your own MERN applications.

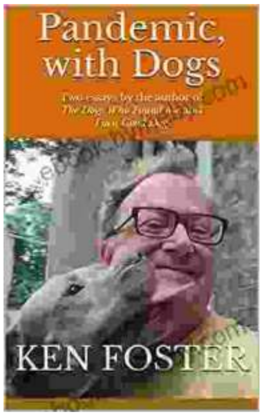
If you're looking for more resources on MERN stack development, here are a few links:

- The MERN Stack Official Website
- The MERN Starter Kit
-



Diogenes the Cynic: The War Against the World

Meet the Philosopher Who Embraced Poverty, Defied Conventions, and Sparked a Revolution In the annals of philosophy, few figures stand...



Pandemic with Dogs: Two Essays

By Susannah Charleson In the midst of the COVID-19 pandemic, as the world grappled with fear, isolation, and uncertainty, a remarkable story unfolded. Dogs, our loyal...