Creating Persistent Data Storage with Node, Express, and MongoDB

Daniel Stern
CODE WHISPERER

http://github.com/danielstern

Creating Persistent Data Storage with Node, Express, and MongoDB - Roadmap

Install and start MongoDB

Initialize DB with default application state

Use Express to create several routes, configure routes to modify DB contents

Authentication will be added later

Installing MongoDB

What Is MongoDB?

Database for storing persistent data

Non-relational (collections, not tables, fluid data structure)

Convenient JSON-based communication works with Node

Alternative to relational databases such as MySQL

Demo

Install MongoDB

Start MongoDB through the command line interface

For deployment, a separate, production MongoDB database will be used

Initializing the Database

Demo

Install tool for verifying contents of local databases (Robo 3T or other)

Create Node script which populates our DB with the default application state

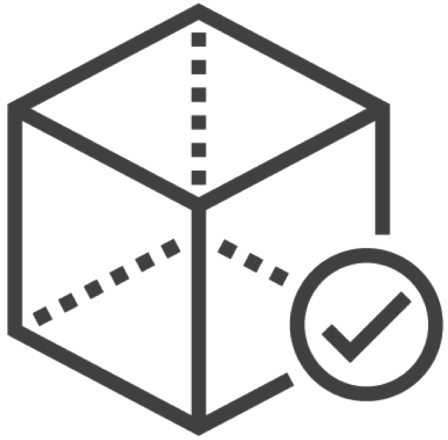
Execute script

Verify database contents

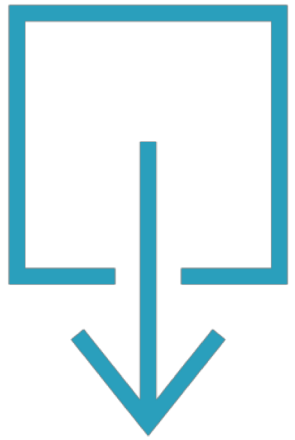
Creating a Server

Why Do We Need a Server? (Recap)

Confidential logic must be hidden from end user



Provides consistent way of working with database

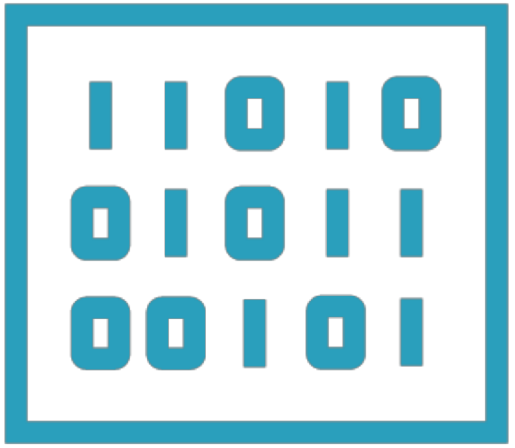


Useful for serving HTML and JS files of finished application

What Is Express?

NPM library which wraps existing *http* toolset

Process of creating a new server and "listening" to specified port



Functional way of describing responses to HTTP requests from application

Demo

Create Express server and listen for HTTP requests

Add routes for getting and updating task information

Create NPM script to initialize server

Persistent Data Storage Summary

MongoDB is a database tool used to make data exist indefinitely

Express routes can be used to allow certain types of access to DB while limiting other kinds of access

Express routes can respond to HTTP requests from our front-end and respond with data from server

Coming up in the Next Module...

- Integrate front end with back end (user changes will persist)
- Update sagas to make HTTP requests where required (using Axios)
- Update application to get state from server instead of from hard-coded data