Assignment 13: Node.js and MongoDB

Sejal Dua, Tufts University

03/26/2020

URL: https://sejaldua.com/stock-ticker-query/
Heroku URL: https://ancient-caverns-12492.herokuapp.com/

Contents

1 Specification

Task: Your assignment is to use Node.js, MongoDB, and Heroku together to create a small web app.

PART 1: Create a MongoDB database and a collection called *companies* to store company names and their stock ticker.

PART 2: Create a web app to read a comma delimited file of companies and their stock ticker code– and insert entries into the MongoDB companies collection.

PART 3: Create an HTML file with a form that asks the user to enter a stock ticker OR a company name (you may want to do this as two separate input boxes– or one input box with a radio button to select the type of input). The form action is directed to a Node.js app that displays the record associated with the company name or stock ticker.

2 All Relevant Code Files

2.1 insert_record.js: inserts records into database

```
var fs = require(’fs’);
const fastcsv = require("fast-csv");
const MongoClient = require(’mongodb’).MongoClient;
const url = "mongodb+srv://sejaldua:(PASSWORD)@cluster0-eeltb.mongodb.net/test?retryWrites=true&w=majority";

function main()
{
    let stream = fs.createReadStream("companies.csv");
    let csvData = [];
    let csvStream = fastcsv
    .parse()
    .on("data", function(data) {
        csvData.push({
        company: data[0],
        ticker: data[1]
        });
    })
    .on("end", function() {
        // remove the first line: header
        csvData.shift();

        console.log(csvData);

        MongoClient.connect(url, {useNewUrlParser: true, useUnifiedTopology: true, native_parser:true}, function(err, db) {
            if (db) {
                if(err) { return console.log(err); }

                var dbo = db.db("stocks");
                var collection = dbo.collection(’companies’);

                console.log("here");
                collection.insertMany(csvData, (err, res) => {
                    if (err) throw err;
                    console.log(‘Inserted: \${res.insertedCount} rows‘);
                    db.close();
                });
            }
            else {
                console.log("oops");
            }
            console.log("Success!");
        });
    });
    stream.pipe(csvStream);
}

main();
```

2.2 server.js: server-side parsing and fetching

```
const MongoClient = require('mongodb').MongoClient;
const url1 = "mongodb+srv://sejaldua:(PASSWORD)@cluster0-eeltb.mongodb.net/test?retryWrites=true&w=majority";
var port = process.env.PORT || 8000;

var http = require('http');
var url = require('url');

http.createServer(function (req, res) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    var obj = url.parse(req.url, true).query;
    var checkBox = (obj.toggle) ? true : false;
    var query = obj.query;

    const client = new MongoClient(url1, {useNewUrlParser: true, useUnifiedTopology: true, native_parser: true});

    connect(client, checkBox, query, res);

}).listen(port);

async function connect(client, checkBox, query, res) {
    client.connect(err => {
        const collection = client.db("stocks").collection("companies");
        console.log("Success!");
        console.log(checkBox, query);

        collection.find().toArray(function (err, items) {
            if (err) {
                console.log("Error: " + err);
                return;
            }
            else {
                var found = false;
                for (i = 0; i < items.length; i++) {
                    if (checkBox) {
                        if (items[i].company == query) {
                            res.write("<html><body style='text-align: center; background-color: black; color: rgb(55, 238, 18); margin-top: 30px; font-size: 14px;'>" + query + "'s ticker symbol is: " + items[i].ticker + "</body></html>");
                            found = true;
                            break;
                        }
                    }
                    else {
                        if (items[i].ticker == query) {
                            res.write("<html><body style='text-align: center; background-color: black; color: rgb(55, 238, 18); margin-top: 30px; font-size: 14px;'>" + query + " is the ticker symbol for " + items[i].company + "</body></html>");
                            found = true;
                            break;
                        }
                    }
                }

                if (!found) {
                    res.write("<html><body style='text-align: center; background-color: black; color: rgb(55, 238, 18); margin-top: 30px; font-size: 14px;'>Oops! The company or ticker you have queried is not in our stock database.</body></html>");
                }
            }
        });
    });
}
```

2.3 package.json: app specification for Heroku

```json
{
  "name": "ancient-caverns-12492",
  "version": "1.0.0",
  "description": "COMP20 HW: Databases (Node + Mongo + Heroku)",
  "main": "server.js",
  "scripts": {
    "start": "node server.js"
  },
  "repository": {
    "type": "git",
    "url": ""
  },
  "keywords": [
    "node",
    "mongodb",
    "stocks",
    "companies"
  ],
  "author": "Sejal Dua",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.19.0",
    "express": "^4.17.1",
    "mongodb": "^3.5.6"
  },
"homepage": "http://ancient-caverns-12492.herokuapp.com/index.html"
}
```

2.4 index.html: homepage with stock query HTML form

```html

<html>
    <head>
    <title>Assignment Week 14: Stock Ticker Query</title>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="style.css" rel="stylesheet">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    </head>

    <body>
    <div id="box">
        <h2>Please enter a company name OR a company ticker.
        </h2>
        <form name="form1" action="https://ancient-caverns-12492.herokuapp.com/">
            <table>
                <tr>
                    <th id="one">
                        <label class="switch" for="toggle">
                            <input type="checkbox" name="toggle" id="toggle" onclick="myFunction()">
                            <span class="slider"></span>
                        </label>
                    </th>
                    <th id="two">
                        <div id="text" >Ticker Symbol</div>
                    </th>
                    <th id="three">
                        <input type="text" name="query" id="field" placeholder="AAPL">
                    </th>
                    <th id="four">
                        <input type="submit">
                    </th>
                </tr>
            </table>

        <br><br>




        </form>

        <script>
            function myFunction() {
                var checkBox = document.getElementById("toggle");
                var text = document.getElementById("text");
                if (checkBox.checked == true) {
                    text.innerHTML = "Company Name";
                    document.getElementById("field").placeholder="Apple Inc.";
                }
                else {
                    text.innerHTML = "Ticker Symbol";
                    document.getElementById("field").placeholder="AAPL";
                }
            }
        </script>
    </div>
    </body>
</html>
```

3 Node.js vs. PHP

I prefer Node.js over PHP because I like how modular it is and how nicely it works with MongoDB. I will admit that it was pretty frustrating trying to figure out how to deploy the app onto Heroku, but I think the process of writing server-side code using node is much less of a hassle than PHP. I have also noticed that there seems to be more documentation and online support regarding Node.js applications than vanilla PHP server-side programming. I like this because if I ever need to make an application with server-side code, I can probably find a similar node demo and reverse-engineer it to what I need.