

```
1  /*
2    Author: Naman Arora
3    Date: 29/11/2020
4  */
5  const express = require('express');
6  const app = express();
7  const port = 5001;
8  const path = require("path");
9  let publicPath = path.resolve(__dirname, "public");
10 app.use(express.static(publicPath));
11 const fetch = require("node-fetch");
12
13 app.listen(port, () => console.log(`App listening on port ${port}!`));
14
15 app.get('/create', createDatabase);
16 app.get('/query/:year/:name', queryDatabase);
17 app.get('/destroy', destroyDatabase);
18
19 // Send html as default
20 app.get('/', (req, res) => {
21   res.sendFile('index.html', {root: __dirname})
22 })
23
24 // _____Create Database
25
26 /*
27 This will make a table in a DynamoDB database on the cloud.
28 the it will fetch the raw data from the S3 object.
29 And finally upload it to the newly created database.
30 */
31 function createDatabase(req, res) {
32   // Set up AWS and DynamoDB
33   var AWS = require("aws-sdk");
34   AWS.config.update({region: "us-east-1"});
35   var dynamodb = new AWS.DynamoDB();
36
37   // Specify movie parameters for the table
38   var params = {
39     TableName: "Movies",
40     KeySchema: [
41       { AttributeName: "year", KeyType: "HASH"}, // Partition key
42       { AttributeName: "title", KeyType: "RANGE" } // Sort key
43     ],
44     AttributeDefinitions: [
45       { AttributeName: "year", AttributeType: "N" },
46       { AttributeName: "title", AttributeType: "S" }
47     ],
48     ProvisionedThroughput: {
49       ReadCapacityUnits: 5,
50       WriteCapacityUnits: 5
51     }
52   };
53
54   // Create the table
55   dynamodb.createTable(params, function(err, data) {
```

```

55     if (err) {
56         console.error("Error in creating table. Error JSON:",
JSON.stringify(err, null, 2));
57     } else {
58         console.log("Table creation successful. Table description JSON:",
JSON.stringify(data, null, 2));
59     }
60 });
61
62 // Fetch movie data from the S3 object
63 // Have a 5 second delay
64 let p = new Promise((res) => {
65     setTimeout(() => {
66         // Fetch S3 data
67         const s3 = new AWS.S3();
68         let objectData = null;
69         var getParams = {
70             // provided in the assignment description
71             Bucket: 'csu44000assign2useast20',
72             Key: 'moviedata.json'
73         }
74
75         s3.getObject(getParams, function(err, data) {
76             if (err) {
77                 return err;
78             }
79
80             objectData = data.Body.toString('utf-8');
81             var jsonData = JSON.parse(objectData);
82             var docClient = new AWS.DynamoDB.DocumentClient();
83
84             // Now add data to database
85             console.log("Please wait.....Ipporting movies to DynamoDB");
86             jsonData.forEach(function(movie) {
87                 var params = {
88                     TableName: "Movies",
89                     Item: {
90                         "year": movie.year,
91                         "title": movie.title,
92                         "info": movie.info
93                     }
94                 };
95
96                 docClient.put(params, function(err, data) {
97                     if (err) {
98                         console.error("Unable to add movie", movie.title,
". Error JSON:", JSON.stringify(err, null, 2));
99                     } else {
100                         console.log("Item added successfully: ",
movie.title);
101                     }
102                 });
103             });
104         });
105     });

```

```

105     }, 5000),
106     });
107
108     console.log("All records sucessfully received !!!");
109 }
110
111 // _____Query
112 Database_____
113 /*
114 Here we find all the movies in a given year.
115 that begin with the entered text string.
116 There are two input boxes to allow a movie name and a year to be entered.
117 Also display them on the webpage.
118 */
119 function queryDatabase(req, res) {
120     // get the year & movie
121     let userYear = req.params.year;
122     let userMovie = req.params.name;
123
124     console.log("-QUERYING-");
125     console.log("-YEAR: " + userYear + "-");
126     console.log("-MOVIE: " + userMovie + "-");
127
128     var AWS = require("aws-sdk");
129     AWS.config.update({region: "us-east-1"});
130     var docClient = new AWS.DynamoDB.DocumentClient();
131
132     // set the query parameters
133     var params = {
134         TableName: "Movies",
135         ProjectionExpression: "#yr, title",
136         KeyConditionExpression: "#yr = :yyyy and begins_with ( title,
137         :letter1 )",
138         ExpressionAttributeNames: {
139             "#yr": "year"
140         },
141         ExpressionAttributeValues: {
142             ":yyyy": parseInt(userYear),
143             ":letter1": userMovie
144         }
145     };
146
147     // make object for return
148     var jsonStr = '{"list":[]}';
149     var jsonObj = JSON.parse(jsonStr);
150
151     // Do the query
152     docClient.query(params, function(err, data) {
153         if (err) {
154             console.log("Query unsuccessful. Error:", JSON.stringify(err,
155             null, 2));
156         } else {
157             console.log("Query succeeded.");
158             data.Items.forEach(function(item) {
159                 // Push the movie to the JSON array

```

```
157     jsonObj['list'].push({"movieName":JSON.stringify(item.title)});
158         console.log(" -", item.year + ": " + item.title);
159     });
160 }
161 // Lastly return the list of movies
162 res.json(jsonObj);
163 });
164 }
165
166 // _____Destroy
Database_____
167 /*
168 Delete the database table.
169 */
170 function destroyDatabase(req, res) {
171     var AWS = require("aws-sdk");
172     AWS.config.update({region: "us-east-1"});
173     var dynamodb = new AWS.DynamoDB();
174
175     var params = {
176         TableName : "Movies"
177     };
178
179     dynamodb.deleteTable(params, function(err, data) {
180         if (err) {
181             console.error("Unable to delete table. Error JSON:",
JSON.stringify(err, null, 2));
182         } else {
183             console.log("Deleted the table. Table description JSON:",
JSON.stringify(data, null, 2));
184         }
185     });
186 }
```