Create Web API Server Labs

Perform these labs on your own computer using VS Code or any editor and command prompt to ensure you understand the lessons presented in the corresponding videos and lectures.

Lab 1: Create Node Web Server

Open a command prompt on your computer. Create a new folder on one of your drives called Samples or use the folder where you normally place your projects.

```
mkdir \Samples
cd \Samples
```

Create project folder underneath this folder and navigate to that folder

```
mkdir WebAPI
cd WebAPI
```

Open VS Code in this new folder.

```
code .
```

Open a new terminal by selecting Terminal | New Terminal… from the menu. Type in the following command

```
npm init
```

You can press the Enter key after each prompt if you want.

Install Express and Nodemon

Add express and nodemon into this project.

```
npm install express nodemon -save
```

Install CORS

Add CORS into this project.

```
npm install cors
```

Lab 2: Add a JSON File to Supply Data

Create a new folder named **assets**.

Create a new file in the **assets** folder named **people.json**. Put the following data into this new file.

```
[
  {
    "personId": 1,
    "firstName": "Paul",
    "lastName": "Shaefer",
    "emailAddress": "Pauls@netinc.com",
    "startDate": "1991-03-01"
  },
  {
    "personId": 2,
    "firstName": "Michael",
    "lastName": "Kawoski",
    "emailAddress": "Michaelk@netinc.com",
    "startDate": "1999-08-22"
  },
  {
    "personId": 3,
    "firstName": "Sara",
    "lastName": "Winchell",
    "emailAddress": "Saraw@netinc.com",
    "startDate": "2001-02-16"
  },
  {
    "personId": 4,
    "firstName": "John",
    "lastName": "Kroon",
    "emailAddress": "Johnk@netinc.com",
    "startDate": "2002-03-29",
    "salary": 90000
  },
  {
    "personId": 5,
    "firstName": "Tim",
    "lastName": "Nicker",
    "emailAddress": "Timn@netinc.com",
    "startDate": "2008-01-01"
  },
  {
    "personId": 6,
    "firstName": "Russ",
    "lastName": "Martlog",
    "emailAddress": "Russm@netinc.com",
    "startDate": "2009-02-15"
  },
  {
    "personId": 7,
    "firstName": "James",
```

```
    "lastName": "Birdy",
    "emailAddress": "Jamesb@netinc.com",
    "startDate": "2015-05-02"
  },
  {
    "personId": 8,
    "firstName": "Trey",
    "lastName": "Chen",
    "emailAddress": "Treyc@netinc.com",
    "startDate": "2005-06-30"
  },
  {
    "personId": 9,
    "firstName": "Jim",
    "lastName": "Jones",
    "emailAddress": "Jimj@netinc.com",
    "startDate": "2003-10-01"
  },
  {
    "personId": 10,
    "firstName": "John",
    "lastName": "Pittsburgh",
    "emailAddress": "Johnp@netinc.com",
    "startDate": "2007-04-16"
  },
  {
    "personId": 11,
    "firstName": "Jeanne",
    "lastName": "Russell",
    "emailAddress": "Jeanner@netinc.com",
    "startDate": "2004-09-11"
  },
  {
    "personId": 12,
    "firstName": "David",
    "lastName": "Lafeet",
    "emailAddress": "Davidl@netinc.com",
    "startDate": "2011-11-11"
  },
  {
    "personId": 13,
    "firstName": "Khanh",
    "lastName": "Voon",
    "emailAddress": "Khanhv@netinc.com",
    "startDate": "2006-07-05"
  },
  {
```

```
    "personId": 14,
    "firstName": "Jim",
    "lastName": "Russell",
    "emailAddress": "Jimr@netinc.com",
    "startDate": "2012-08-17"
  },
  {
    "personId": 15,
    "firstName": "David",
    "lastName": "Tarkas",
    "emailAddress": "Davidt@netinc.com",
    "startDate": "1999-03-16"
  },
  {
    "personId": 16,
    "firstName": "Craig",
    "lastName": "Showman",
    "emailAddress": "Craigs@netinc.com",
    "startDate": "2001-10-15"
  },
  {
    "personId": 17,
    "firstName": "Brooks",
    "lastName": "Anderson",
    "emailAddress": "Brooksa@netinc.com",
    "startDate": "2003-07-15"
  },
  {
    "personId": 18,
    "firstName": "Mark",
    "lastName": "Parks",
    "emailAddress": "Markp@netinc.com",
    "startDate": "2013-01-15"
  },
  {
    "personId": 19,
    "firstName": "John",
    "lastName": "Smith",
    "emailAddress": "JohnSmith@netinc.com",
    "startDate": "2002-04-01"
  }
]
```

Lab 3: Create a Repository Object to Read JSON File

Create a new folder named **repos**.

Create a new file in the **repos** folder named **peopleRepo.js**. Put the following code into this new file.

```
let fs = require('fs');

const FILE_NAME = './assets/people.json';

let peopleRepo = {
  get: function (resolve, reject) {
    fs.readFile(FILE_NAME, function (err, data) {
      if (err) {
        reject(err);
      }
      else {
        resolve(JSON.parse(data));
      }
    });
  },
  getById: function (id, resolve, reject) {
    fs.readFile(FILE_NAME, function (err, data) {
      if (err) {
        reject(err);
      }
      else {
        let resp = JSON.parse(data).find(row =>
row.personId == id);
        resolve(resp);
      }
    });
  },
  insert: function (newData, resolve, reject) {
    fs.readFile(FILE_NAME, function (err, data) {
      if (err) {
        reject(err);
      }
      else {
        let resp = JSON.parse(data);
        // Get array of personId values
        let ids = resp.map(row => { return row.personId;
});
        // Get max id and add one
        let maxId = Math.max(...ids) + 1;
        // Assign new personId
        newData.personId = maxId;
        // Add new person to array
        resp.push(newData);
        fs.writeFile(FILE_NAME, JSON.stringify(resp,
null, 2), function (err) {
          if (err) {
            reject(err);
```

```
          }
          else {
            resolve(newData);
          }
        });
      }
    });
  },
  update: function (changedData, id, resolve, reject) {
    fs.readFile(FILE_NAME, function (err, data) {
      if (err) {
        reject(err);
      }
      else {
        let list = JSON.parse(data);
        let current = list.find(row => row.personId ==
id);
        if (current) {
          Object.assign(current, changedData);
          fs.writeFile(FILE_NAME, JSON.stringify(list,
null, 2), function (err) {
            if (err) {
              reject(err);
            }
            else {
              resolve(changedData);
            }
          });
        }
      }
    });
  },
  delete: function (id, resolve, reject) {
    fs.readFile(FILE_NAME, function (err, data) {
      if (err) {
        reject(err);
      }
      else {
        let list = JSON.parse(data);
        let index = list.findIndex(row => row.personId
== id);
        if (index != -1) {
          list.splice(index, 1);
          fs.writeFile(FILE_NAME, JSON.stringify(list,
null, 2), function (err) {
            if (err) {
              reject(err);
```

```
            }
            else {
              // Deleted the data
              resolve(true);
            }
          });
        }
        else {
          // Did not find the id to delete
          resolve(false);
        }
      }
    });
  }
};

module.exports = peopleRepo;
```

Lab 4: Create the Web API Endpoints

Create a new file named **index.js** in the root of this project.

Add the following code into this new file.

```
// Bring in the express server and create application
let express = require('express');
let app = express();
let peopleRepo = require('./repos/peopleRepo');
let cors = require('cors');

// Use the express Router object
let router = express.Router();

// Configure middleware to support JSON data parsing in
request object
app.use(express.json());

// Enable CORS for all requests
// NOTE: All traffic coming from your local machine are
allowed.
// References:
https://expressjs.com/en/resources/middleware/cors.html
let corsOptions = {
  "origin": "http://localhost:3000",
  "methods": "GET,HEAD,PUT,PATCH,POST,DELETE",
  "optionsSuccessStatus": 204
};
//app.use(cors(corsOptions));
app.use(cors());

// Return a list of all data
router.get('/people', function (req, res, next) {
  peopleRepo.get(function (data) {
    res.status(200).json({
      "status": 200,
      "statusText": "OK",
      "message": "All people retrieved.",
      "data": data
    });
  }, function (err) {
    next(err);
  });
});

// Return a single json object
router.get('/people/:id', function (req, res, next) {
  peopleRepo.getById(req.params.id, function (data) {
    if (data) {
      res.status(200).json({
        "status": 200,
        "statusText": "OK",
```

```
        "message": "Single person retrieved.",
        "data": data
      });
    }
    else {
      res.status(404).json({
        "status": 404,
        "statusText": "Not Found",
        "message": "The person '" + req.params.id + "'
could not be found.",
        "error": {
          "code": "NOT_FOUND",
          "message": "The person '" + req.params.id + "'
could not be found."
        }
      });
    }
  }, function (err) {
    next(err);
  });
});

router.post('/people', function (req, res, next) {
  peopleRepo.insert(req.body, function (data) {
    res.status(201).json({
      "status": 201,
      "statusText": "Created",
      "message": "New Person Added.",
      "data": data
    });
  }, function (err) {
    next(err);
  });
});

router.put('/people/:id', function (req, res, next) {
  peopleRepo.getById(req.params.id, function (data) {
    if (data) {
      // Attempt to update the data
      peopleRepo.update(req.body, req.params.id,
function (data) {
        res.status(200).json({
          "status": 200,
          "statusText": "OK",
          "message": "Person '" + req.params.id + "'
updated.",
          "data": data
```

```
        });
      });
    }
    else {
      res.status(404).json({
        "status": 404,
        "statusText": "Not Found",
        "message": "The person '" + req.params.id + "'
could not be found.",
        "error": {
          "code": "NOT_FOUND",
          "message": "The person '" + req.params.id + "'
could not be found."
        }
      });
    }
  }, function (err) {
    next(err);
  });
});

router.delete('/people/:id', function (req, res, next) {
  peopleRepo.getById(req.params.id, function (data) {
    if (data) {
      // Attempt to delete the data
      peopleRepo.delete(req.params.id, function (data) {
        res.status(204).send();
      })
    }
    else {
      res.status(404).json({
        "status": 404,
        "statusText": "Not Found",
        "message": "The person '" + req.params.id + "'
could not be found.",
        "error": {
          "code": "NOT_FOUND",
          "message": "The person '" + req.params.id + "'
could not be found."
        }
      });
    }
  }, function(err) {
    next(err);
  });
});
```

```
// Configure router so all routes are prefixed with
/api/v1
app.use('/api/', router);

// Create server to listen on port 5000
let server = app.listen(5000, function () {
  console.log('Node server is running on
http://localhost:5000.');
});
```

Update the Package File

Open the **package.json** file

Add a "start" property to the "scripts" property.

```
"scripts": {
  "start": "nodemon index.js",
  "test": "echo \"Error: no test specified\" && exit 1"
},
```

Try it Out

In the terminal window, type in:

```
npm start
```

Open **Postman**

Type in the following.

```
http://localhost:5000/api/people
```

You should see the list of people returned as JSON.