COS 498

Assignment 2: Express Station

10pts

Due: Sept 24th (Wednesday), 11:59pm

1 Assignment Description

For this assignment, you will create a Node.js server using Express.js that serves static files and provides a custom API endpoint. This assignment will server as the bridge between serving static files and providing information after calculations are done on the server side.

1.1 Goal

Create a nodejs server that serves the same static files as assignment 1 (you're free to make improvements to these static files if you want). In addition to the static files, have at least one custom API endpoint that returns JSON data about the client and the current data/time.

2 Required Components

2.1 Node.js Server Setup

- Initialize a new Node.js project with `npm init`
- Install Express.js as a dependency
- Create a proper `package.json` file with all required metadata
- Set up a main server file (e.g., `server.js`)

2.2 Express Server Implementation

- Create an Express application
- Configure the server to listen on your assigned port (e.g., 30XX)
- Implement proper error handling and logging
- Include appropriate middleware setup

2.3 Static File Serving

- Serve static files from a designated directory (e.g., `public/` or `static/`)
- These static files should be the same as the ones you used in Assignment 1. You can copy them from your assignment 1 folder.
- Ensure proper file organization and directory structure (keep the static files in a directory seperate from the server file).

2.4 API Endpoint

- Create at least one custom API endpoint that returns JSON data
- The endpoint should be accessible via GET request
- Return a JSON object that contains two pieces of information: something about the client, and the current date and time.
- Include proper HTTP status codes and headers

3 Technical Requirements

3.1 Project Structure

Your project should follow a logical directory structure:

```
your-project/
|-- package.json
|-- server.js
|-- public/ (or static/)
|   |-- (Static files from assignment 1)
|-- README.md (optional)
```

3.2 package.json Requirements

- Include proper project metadata (name, version, description, author)
- Specify Express as a dependency
- Include a start script for running the server (so that you can run `npm start` to start the server)

3.3 Server Functionality

- Server should start without errors
- Static files should be accessible via browser
- API endpoint should return valid JSON
- Include basic error handling for common issues

4 API Endpoint Specifications

Your API endpoint should:

- Be accessible at a custom route (e.g., `/api/data`, `/api/users`, etc.)
- Include appropriate Content-Type headers

4.1 Example API Response

Here is an example of what your API response should look like:

```
{
  "status": "success",
  "data": {
    "client_ip": "127.0.0.1",
    "timestamp": "2025-01-15T10:30:00Z"
  }
}
```

5 Testing and Documentation

5.1 Testing Requirements

- Test that your server starts successfully
- Verify static files are served correctly
- Test your API endpoint and verify JSON response
- Document any known issues or limitations

6 Point Distribution

Component	Points
Node.js project setup and package.json	2pts
Express server implementation	3pts
Static file serving	2pts
API endpoint with JSON response	3pts
Server fails to start	-5pts
No comments at the top of the server file	-2pts
Total	**10pts**

7 Submission Guidelines

- On Brightspace, submit which server you are using (umainecos.org or toastcode.net). And where on your account your server is at.
- Ensure all dependencies are listed in `package.json`
- Test your server before submission
- Make sure your API endpoint is accessible and returns valid JSON

8 Getting Started

1. Create a new directory for your project
2. Run `npm init` to initialize your project
3. Install Express: `npm install express`
4. Create your server file and implement the required functionality
5. Test your server with `npm start` (you need to keep it running in the terminal while you are testing it)
6. Document your project in the README (optional)