Pair-Programming Assignment 2 (PPA 2)

Introducing APIs, JSON, and Version Control

In this assignment, you will extend the Node.js server created in Pair-Programming Assignment 1. Instead of serving only HTML pages, your server will now expose an API endpoint that returns structured data in JSON format. You will also begin using GitHub to manage and share your code. This marks the transition from a static website to a collaborative client–server application.

Learning Objectives

- Understand what an API is and why APIs are used
- Understand JSON as a data exchange format
- Create your first REST-style endpoint
- Use fetch with async and await in the browser
- Use GitHub to track changes and collaborate with a partner

Conceptual Background: APIs and JSON

An API, or Application Programming Interface, is a contract that defines how one piece of software communicates with another. In web applications, APIs are commonly exposed over HTTP.

JSON, or JavaScript Object Notation, is a lightweight data format used to exchange structured data between clients and servers. JSON is easy for both humans and programs to read and write.

How This Fits into the Final System

The appointment scheduling system you are building will eventually support creating, updating, and managing time slots and appointments. In this assignment, you begin by exposing read-only appointment slot data through an API endpoint.

Later assignments will add data creation, validation, persistence to disk, and analytics, resulting in a fully functional system by the end of the semester.

Assignment Tasks

- Extend your existing Node.js server from PPA 1
- Create an in-memory array of appointment time slots on the server
- Add a new endpoint GET /api/slots that returns the slots as JSON
- Update the provider page to fetch slot data from the API
- Render the returned data dynamically using JavaScript in the browser
- Initialize a Git repository and push your work to GitHub

Example Data Model

```
// In-memory data model for appointment slots
// This data lives only in memory for now
// Persistence will be added in a later assignment

const slots = [
  {
    id        : 1,
    startTime : "2026-03-01T09:00",
    endTime   : "2026-03-01T09:30",
    status    : "available"
  },

  {
    id        : 2,
    startTime : "2026-03-01T10:00",
    endTime   : "2026-03-01T10:30",
    status    : "available"
  }
];
```

Example API Endpoint

```
// GET /api/slots
// Returns all appointment slots as JSON

if (req.url === "/api/slots" && req.method === "GET") {

  res.writeHead(200, { "Content-Type": "application/json" });
  res.end(JSON.stringify(slots));

  return;
}
```

Example Frontend Fetch Code

```
// Fetch appointment slots from the server
// This code runs in the browser

async function loadSlots() {

  const response = await fetch("/api/slots");
  const data     = await response.json();

  console.log(data);

}
```

Using GitHub for This Assignment

GitHub is a platform for storing and sharing code. It allows you to track changes over time and collaborate safely with your partner. For this course, GitHub acts as the source of truth for your project.

Basic GitHub Workflow

- Create a new repository on GitHub
- Clone the repository to your local machine
- Make changes to your code
- Commit your changes with a clear message
- Push your commits back to GitHub

Typical Commands

```
// Clone the repository (run once)
git clone <repository-url>


// Check current status
git status


// Stage changes
git add .


// Commit changes with a message
git commit -m "Add API endpoint for slots"


// Push changes to GitHub
git push
```

Both partners should commit regularly. Do not wait until the end of the assignment to push your work. Small, frequent commits make collaboration and debugging easier.

Deliverables

- A working GET /api/slots endpoint
- JSON data returned from the server
- Frontend JavaScript that fetches and displays the data
- A GitHub repository with at least two commits from each partner

Reading Alignment

Flanagan, JavaScript: The Definitive Guide, Chapters 3–4.