API Session – Chapter 3

- Own APIs –

# Create Own APIs Prerequisites

First of all, we need few software’s to be installed to create our own APIs.

1. NodeJS
2. NPM - Node Package Manager (Comes along with NodeJS)
3. Json Server (Can be installed by NPM)

# Installing Node.js

**Node.js** is a JavaScript runtime built on Chrome's V8 JavaScript engine. It's used for building server-side applications.

## Downloading Node.js

1. **Visit the official Node.js website:** <https://nodejs.org/en/download/>
2. Choose the appropriate installer for your operating system (Windows, macOS, or Linux).

* **Recommended:** Download the LTS (Long Term Support) version for stability.

## Installing Node.js

The installation process is straightforward and usually involves clicking through the installer's prompts.

### Windows

* Double-click the downloaded .msi file.
* Follow the on-screen instructions.

**Note:** On Tools for Native Modules screen, Make sure the checkbox is **NOT** selected.

# Setting the Node.js Path in Windows

**Note:** Typically, the Node.js installer automatically adds the necessary paths to the system environment variables. However, if you encounter issues or want to manually verify the setup, follow these steps:

## 1. Locate the Node.js Installation Directory

* Open File Explorer and navigate to the directory where you installed Node.js. This is usually something like:
  + C:\Program Files\nodejs

## 2. Identify the node.exe and npm.exe Paths

* Inside the Node.js installation directory, you'll find two important executable files:
  + node.exe: This is the Node.js runtime.
  + npm.exe: This is the Node Package Manager (npm).
  + The complete paths to these files would look something like:
    - C:\Program Files\nodejs\node.exe
    - C:\Program Files\nodejs\npm.exe

## 3. Access Environment Variables

* Right-click on the "This PC" icon and select "Properties".
* Click on "Advanced system settings".
* In the System Properties window, click on the "Environment Variables" button.

## 4. Edit the System Path Variable

* Under "System variables", find the "Path" variable and click "Edit".
* Click "New" and paste the path to the Node.js installation directory (e.g., C:\Program Files\nodejs).
* Click "OK" to save the changes.

## 5. Verify the Installation

* Open a new command prompt window.
* Type **node -v** | **node --version** and press Enter. This should display the installed Node.js version.
* Type **npm -v** | **npm --version** and press Enter. This should display the installed npm version.

If you see the versions, Node.js is correctly installed and configured.

**Important Note:** If you're using a 64-bit operating system, you might have installed Node.js in the C:\Program Files (x86) directory. Adjust the paths accordingly.

# Installing JSON Server using npm

**JSON Server** is a great tool for quickly creating mock REST APIs. Here's how to install it using npm:

## Prerequisites

* **Node.js and npm:** Ensure you have Node.js and npm installed on your system. You can verify this by opening your terminal or command prompt and running node -v and npm -v.

## Steps

1. Open your terminal or command prompt.
2. **Navigate to your project directory:** Use the cd command to change to the directory where you want to use JSON Server.
3. **Run the installation command:** Type the following command and press Enter:

**npm install -g json-server**

This installs JSON Server globally on your system, making it accessible from any project.

# Creating a Sample Student JSON File and Running JSON Server

## Creating the Student JSON File

Let's create a JSON file named student.json with some sample student data:

**JSON**

{

"students": [

{

"id": 1,

"name": "John Doe",

"rollNumber": 1234,

"email": "johndoe@example.com"

},

{

"id": 2,

"name": "Jane Smith",

"rollNumber": 5678,

"email": "janesmith@example.com"

}

]

}

## Running JSON Server

1. Open your terminal or command prompt.
2. **Navigate to the directory** where you saved the student.json file.
3. Run the following command**:** **npx json-server student.json| json-server student.json**

This command will start a JSON server using the data from student.json. By default, it runs on port 3000.

## Access your mock API

* http://localhost:3000/students to get a list of all students
* http://localhost:3000/students/1 to get the details of student with ID 1

### Note

* The id property is crucial for JSON Server to identify individual records.
* You can add more students, properties, or even other data structures to the student.json file as needed.
* JSON Server provides basic CRUD operations (Create, Read, Update, Delete) out of the box.