Node.js Architecture Overview

Lecture 03: Node.js – Architectures & Core Features, and Modules

Course: Advanced Web Technologies (CSC337)
Duration: 2 Hours
Lecture Type: Theory

Lecture Outline

1. **Introduction to Node.js**
    - What is Node.js?
    - Why use Node.js?
    - Real-world applications of Node.js
2. **Node.js Architecture**
    - Single-threaded, non-blocking I/O model
    - Event-driven architecture
    - The V8 Engine
3. **Core Features of Node.js**
    - Asynchronous programming
    - Built-in modules
    - Package management with NPM
4. **Node.js Modules**
    - Types of modules (Core, Local, Third-party)
    - Importing and exporting modules
    - Creating custom modules
5. **Student Activity: Quick Quiz**
6. **Summary and Q/A Session**

---

1. Introduction to Node.js

What is Node.js?

Node.js is an open-source, cross-platform JavaScript runtime that executes JavaScript code outside a web browser. It is primarily used for building server-side applications.

Why Use Node.js?

- **Fast Execution:** Uses the V8 JavaScript engine.
- **Non-Blocking I/O:** Handles multiple requests simultaneously.
- **Single Programming Language:** Both frontend and backend can be written in JavaScript.
- **Large Ecosystem:** Supported by a huge community and NPM (Node Package Manager).

Real-World Applications

- **Web Applications:** Netflix, LinkedIn, PayPal use Node.js.
- **IoT Devices:** Raspberry Pi applications.
- **Real-Time Applications:** Chat applications, stock market tracking.

---

2. Node.js Architecture

Single-Threaded, Non-Blocking I/O Model

Unlike traditional multi-threaded architectures, Node.js operates on a single thread. However, it uses an event loop and asynchronous callbacks to handle multiple requests without waiting.

Event-Driven Architecture

Node.js follows an event-driven model where:

- An event occurs (e.g., a user request).
- A callback function is executed to handle the request.
- The event loop continuously listens for new events.

The V8 Engine

- Developed by Google.
- Converts JavaScript code directly into machine code, making execution fast.

---

3. Core Features of Node.js

Asynchronous Programming

- Traditional languages like PHP or Python execute code line by line.
- Node.js uses callbacks or Promises to handle multiple operations simultaneously.

Example of Asynchronous Code:

```
javascript


console.log("Start");
setTimeout(() => {
    console.log("Inside setTimeout");
}, 2000);
console.log("End");
```

Output:

```pgsql
Start
End
Inside setTimeout (after 2 seconds)
```

Built-in Modules

Node.js has several built-in modules like:

- `fs` (File System) – Read/write files.
- `http` – Create web servers.
- `path` – Work with file paths.

Package Management with NPM

- World's largest repository for JavaScript libraries.
- Install third-party modules easily using:

```nginx
npm install package_name
```

---

4. Node.js Modules

Modules help organize and reuse code in Node.js.

Types of Modules

1. **Core Modules:** Built into Node.js (e.g., `fs`, `http`).
2. **Local Modules:** Custom modules created by developers.
3. **Third-Party Modules:** Installed via NPM (e.g., `express`, `mongoose`).

Importing and Exporting Modules

Example of creating a custom module:

math.js (Custom Module)

```javascript
function add(a, b) {
    return a + b;
}
module.exports = add;
```

app.js (Importing the Module)

```javascript
const add = require('./math');
console.log(add(5, 3)); // Output: 8
```

---

5. Student Activity: Quick Quiz

Q1: What is Node.js primarily used for?

a) Frontend Development
b) Backend Development
c) Mobile App Development
d) Database Management

Q2: Which module is used to create a web server in Node.js?

a) fs
b) http
c) path
d) os

Q3: What is the function of the event loop in Node.js?

a) Blocks the execution of code
b) Handles asynchronous operations
c) Stops Node.js from executing
d) Converts JavaScript to Python

(Answers will be discussed in class.)

---

6. Summary & Q/A Session

- **Node.js is a fast, non-blocking JavaScript runtime used for backend development.**
- **It operates on a single-threaded, event-driven model, making it efficient for real-time applications.**
- **Core features include asynchronous programming, built-in modules, and an extensive package ecosystem.**
- **Modules in Node.js help organize code, and they can be core, local, or third-party.**

Any Questions?

Let’s discuss any doubts or real-world use cases you can think of! 🚀