CS 4220

- Current Trends in Web Design & Development -

Cydney Auman

AGENDA

01 Node.js

02 Node.js Module System

03 NPM

04 Code Demo

05 Lab Time!

What is Node.js?

Node.js is a platform. It is open-source, cross-platform, and allows for executing JavaScript code outside of a browser. The creators of Node.js took JavaScript and allowed it to run on your machine.

The main idea is that Node.js uses an event-driven, non-blocking I/O model to remain lightweight and efficient. I/O refers primarily to interaction with the system's disk and network.

In Node.js, there is no such call to start the Event Loop. Node.js simply enters the event loop after encountering asynchronous code. Node.js exits the event loop when there are no more callbacks to perform.

What Node.js Achieves

Node.js is *not* the solution for every development situation. It is a platform that fills a particular need and understanding Node.js strengths is essential. You would never use Node.js for CPU-intensive operations. Using it for heavy computations will nullify all of its advantages.

One of the more difficult problems in writing programs that communicate over a network is managing input and output — meaning the reading and writing of data to and from the network, the hard drive, databases and etc.

Node.js was initially conceived for the purpose of **making this asynchronous I/O easy** and **convenient**. It is meant to build fast and scalable network applications. This is because it is capable of handling a huge number of simultaneous connections. An added benefit is that it has become a way to unify the stack by having both JavaScript server side and client side.

Where Node.js Excels

Any place where we can perform asynchronous I/O - reading and writing to network connections, reading/writing to the filesystem, and reading/writing to the database.

All of these are common tasks in web applications and execute very fast in Node.js

- Server Side API on top of a Database
- Command Line Utilities/Apps
- Web Applications (Full JavaScript Stack)
- Real Time Applications (Chat Apps, Monitoring Dashboard, etc)

Node.js and Module System

Node.js attempts to remain as lightweight as possible. So, Node.js puts little functionality in the global scope.

If you want to access other built-in functionality, you have to use the **module system** for it. Node.js utilizes the CommonJS module system, based on the **require** function.

This `require()` system is built into Node.sj and is used to load anything from built-in modules to downloaded packages to JavaScript files that are part of your own program.

Node.js Core API

Information on the entire Node.js API and it's core module system can be found on the Node.js site.
Node.js API (https://nodejs.org/dist/latest-v12.x/docs/api/)

Path

- Utilities for working with file and directory paths.

```
const path = require('path')
```

File System

- Handles File I/O

```
const fs = require('fs')
```

HTTP/HTTPS

- Interfaces designed to support features of the http or https protocol.

```
const http = require('http')
const https = require('https')
```

Using Node.js Core API

The core Node.js API is extensive. It is difficult for any one person to memorize the every module and every method. This is why Node.js provides extensive documentation on each module provided.

Path

- Utilities for working with file and directory paths.
  `https://nodejs.org/dist/latest-v12.x/docs/api/path.html`

File System

- Handles File I/O
  `https://nodejs.org/dist/latest-v12.x/docs/api/fs.html`

HTTP/HTTPS

- Interfaces designed to support features of the http or https protocol.
  `https://nodejs.org/dist/latest-v12.x/docs/api/http.html`

Node Package Manager (NPM)

NPM is two things: an online service where one can download (and upload) packages and a program (bundled with Node.js) that helps you install and manage them.

The online service portion of **NPM** is contains repository of JavaScript modules, many of which are specifically written for Node.js.

When you installed Node on your computer, it came bundled with the **npm** command. This allows you to interact with this repository.

About NPM - https://docs.npmjs.com/about-npm/

```
console.log('Week 06');
console.log('Code Examples');
```

Lab, Homework and Prep

Lab Time

- Work on Lab Week 6
- Work on Homework Week 5

Preparation for Next Week

- Read Eloquent Javascript Chapters 20
- Watch YouTube Get Started with NPM (Slide 8)