Node.js & Deno

Tamas Piros

Agenda

- **Week 1: Node.js & Deno**
- Week 2: SQL & NoSQL
- Week 3: Creating and deploying REST APIs (part 1)
- Week 4: Creating and deploying REST APIs (part 2)

Agenda

- Week 1: Node.js & Deno
  - History of Node.js & Deno
  - Introduction to the Node.js ecosystem
  - Built-in Node.js modules
  - npm
  - package.json
  - Working with the CLI
  - Deno overview

History of Node.js

- Created in 2009 by Ryan Dahl
- Widespread adoption around 2011 (LinkedIn, Uber)
- 2013 - Blogging platform released on Node.js (Ghost)
- 2014 - introduction of io.js
- 2015 - Merge of Node.js and io.js + release of Node.js v4
- 2017 - 3 billion npm packages downloaded per week
- 2021 - Node.js v16 (module support, http/2, experimental Web Streams API etc)

Installation

- https://nodejs.org/en/download/
- LTS vs Current
- nvm (node version manager)
    - Multiple installations allowed - great for development

Node.js

- Asynchronous event-driven JavaScript runtime (runs on V8)
- Non-blocking, concurrent operations
- Written in C, C++
- HTTP is a first-class citizen
    - Node.js can be (and is) used as web libraries and frameworks
- No threads, however child processes can be spawned

JavaScript vs JavaScript

- Node.js has full JavaScript support
- Browser vs Node.js
  - Some Web (Browser) APIs are not available in Node.js
  - document and window are not available in Node.js
  - Environment control

CommonJS vs Modules

- Node.js historically used CommonJS (`require()`)
- Today, Node.js has support for ES Modules (`import/export`)
  - Multiple ways to enable modules: .mjs, package.json “module” enable
  - Built-in modules also support ESM

Getting started - “hello world”

- Write JavaScript
- Save as “app.js” (or as any .js file, or .mjs for a module)
- Run via the CLI by executing `node app.js`
  - Node.js also has a REPL (node -e)

Built-in Modules

- Node.js comes built-in with a number of functional modules
- http(s), fs, path, os, querystring, readline, stream, net, events, util, path
    - https://nodejs.org/api/

npm

- Vast ecosystem of packages
- npm cli tool is built-in to Node.js
- Install (locally or globally), update, audit packages
- Publish package (privately or publicly)
- npm enterprise
- npm vs yarn
  - Installation of packages: **npm**: serial, **yarn**: parallel

`package.json`

- Controls what packages are available for installation
  - Production / dev packages
  - Versioning: https://tpiros.dev/blog/what-is-semantic-versioning/
- Scripts (commands to run) - build, test, dev
- `package-lock.json`

nodemon

- Auto start on change in monitored files
- Useful for development
- Settings can be enabled via ‘nodemon.json’

Deno

- Ryan Dahl (2018) (node -> deno)

- JavaScript / TypeScript runtime (based on V8) written in Rust

- Similar to Node.js in that it has an event-driven architecture, and non-blocking features

- No package manager

- Uses modern syntax and URL based imports

Bun (bun.sh)

- Bun is a fast all-in-one JavaScript runtime
    - Created in 2022 (currently in beta)
    - https://blog.logrocket.com/bun-javascript-runtime-taking-node-js-deno/