

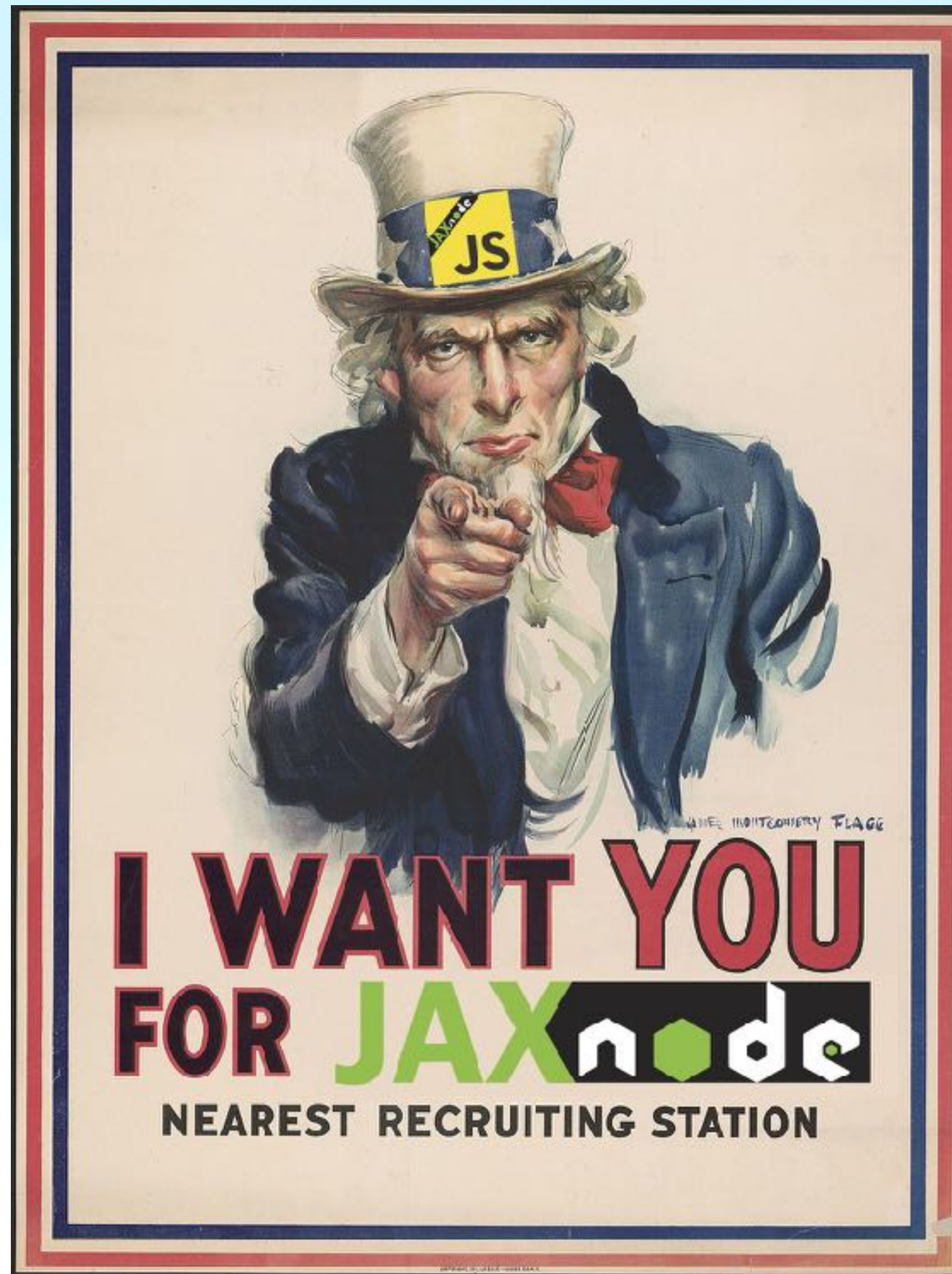
Deno 2.0

What is new in JavaScript runtimes



David Fekke: JaxNode October 2024

Looking for Speakers



JavaScript runtimes

JS News

- Next year will mark the 30th anniversary of JavaScript
- Node 22 going LTS
- Deno just went 2.0
- Bun has become a popular alternative runtime
- Amazon introduced LLRT for Lambda

DENO 2



What is Deno

- Newish (6yo) JavaScript runtime built on V8
- Started by Ryan Dahl, the inventor of Node.js
- Zero config runtime
- Native TypeScript support
- Written in Rust with V8 bindings
- Adds support for WebAPI entities like `window` and `fetch`.
- Built with Security features

Deno 2.0

What is new in Deno?

- Deno 2.0 released earlier this month
- Backwards compatible with Node
- Support for ``node_modules`` and ``package.json``
- Commercial support and LTS
- Standard Library
- Workspaces support
- Package Management



Deno Install

How to install Deno

- `$ choco install deno`
- `$ brew install deno`
- `$ cargo install deno`
- `$ curl -fsSL https://deno.land/install.sh | sh`
- `$ irm https://deno.land/install.ps1 | iex`

Deno security

- Network and file system access turned off by default
- Use flags to allow access
- `—allow-net` or `—allow-net="jaxnode.com"`
- `—allow-read` and `—allow-write`
- `—allow-env` for environment variables
- `Deno run —allow-all`

Deno Commands

CLI

- `Deno run script_name.js`
- `Deno fmt` // Format your code
- `Deno compile` // Create an executable
- `Deno lint` // Linter
- `Deno install` // Install external modules
- `Deno run --watch main.ts` // run and watch for changes

Deno Features

- TypeScript zero config support
- Standard Library
- Full support for WebAPIs
- Deno.json
- JSR
- Workspaces and Monorepos // New Deno 2

TypeScript Support

TypeScript is a first class language in Deno

- Type checking
- `$ deno check //` This command will run a type checker against your code
- `$ deno check --doc //` Will check JSDoc code snippets
- You can also check JS
- You can also check against DOM objects
- Support for `*.d.ts` files

Module support

- Deno supports NPM now, NodeAPIs, along with JSR and standard library
- `import * as emoji from "npm:node-emoji";`
- CommonJS and cjs support
- Support for both `deno.json` and `package.json` files

./util.ts

```
export function sayHello(thing: string) {  
  console.log(`Hello, ${thing}!`);  
}
```



```
export interface Foo {}  
export class Bar {}  
export const baz = "baz";
```

./main.ts

```
import { sayHello } from "./util.ts";  
sayHello("World");
```

```
import * as util from "./util.ts";  
util.sayHello("World");
```

```
import { camelCase } from "jsr:@luca/cases@1";  
console.log(camelCase("hello world")); // helloWorld
```

```
import OpenAI from "https://deno.land/x/openai@v4.53.0/mod.ts";  
const client = new OpenAI();
```


Handling Imports

How to reference modules in Deno

- ``import { doFunc } from "jsr:@namespace/module";``
- You can import from NPM, Node, JSR or deno
- Use “npm:” for NPM modules
- Use “node:” for Node APIs
- Use “deno:” for Deno APIs
- Use “jsr:” for JSR modules

Node modules folder support

node_modules

- Deno does not use a local `node_modules` folder
- Some Node projects like Next.js and Svelte require node_modules
- Create using `\$ deno run --node-modules-dir=auto main.ts`
- Use the “nodeModulesDir”: “auto” or “manual” in the deno.json file

Standard Library

Built in modules: <https://jsr.io/@std>

- Deno includes a standard library, so many modules like lodash and not needed
- Some examples include @std/assert, @std/cache, @std/bytes, @std/collections, @std/fmt, @std/front-matter, @std/html, @std/path, @std/net
- Check and see if functions you need are not already in the standard library
- Varying levels of stability
- Also available from JSR



VS



JSR vs NPM

- JSR is the JavaScript Registry for open-source packages
- NPM has most modules, but legacy issues + security concerns
- JSR has TypeScript and ESM support
- Modules using CommonJS are not accepted
- JSR automatically generates documentation
- Reference correct extensions such as `./greet.ts` instead of `./greet`
- You can import NPM packages to your JSR packages
- Has full support for @namespaces

Deno APIs

Built in support for File and Networking

- `Deno.serve()` // Http server
- `Deno.open()` //. Opens and reads a file
- `Deno.connect()` // Connects to TCP service
- `Import.meta` object for url properties
- ``console.log(Is ${import.meta.url} the main module?', import.meta.main);``
- FFI for calling functions from native code

Web APIs

- These APIs work on the browser and in Deno
- Location with `—location` frag
- Fetch API
- Workers
- Web Storage

Node API

Deno 2 now has support for Node APIs

- node:os, node:path, node:querystring, node:stream, node:url, node:buffer, node:console, node:crypto and node:assert
- This provides most of the Node.js compatibility
- Node compatibility is 1.0

JSX

Support for JSX and TSX components

- Deno transpiler has default settings for transpiling JSX code
- React 17 added new JSX transforms. You can add with following options:
- Override with Preact settings

```
{  
  "compilerOptions": {  
    "jsx": "react-jsx",  
    "jsxImportSource": "react"  
  },  
  "imports": {  
    "react": "npm:react"  
  }  
}
```

Deno support for Major frameworks

Deno supports the following frameworks

- Express.js
- Svelte
- Next.js
- Remix
- Astro

Deno support

- Deno is open source project and commercial company
- Community support through Discord and StackOverflow
- Enterprise commercial support

Deno Hosting

How do I run this in Prod?

- Documentation details on hosting through major cloud providers such as AWS, Digital Ocean, GC and Cloudflare
- They have a docker container for generic hosting
- Deno also has their own hosting
- Simply deploy from the terminal using the `deployctl` command
- Good docs on CI integration

Deno hosting pricing

- Free, Pro and Enterprise plans
- Pro is \$20 a month
- Enterprise contact for pricing
- Free and Pro limited to 50 ms per request
- Free and Pro both allow custom domains

Jupyter Notebooks

Jupyter Notebooks uses Deno for JavaScript

- Run using `deno jupyter --unstable`
- Assumes that you already have Jupyter set up in your PATH
- You can run in VSCode

Fresh

Web Framework for Deno and modern JavaScript

- Comes with Tailwind and Plugin API
- Has Island architecture using Preact
- Stream HTML from the server
- Support for partials

Node.js vs Deno

Can I use Deno in Prod?

- Deno has better support for Web APIs and web standards
- Deno has support for multiple module repositories
- Both use V8 for the JavaScript engine
- Deno offers support for the major frameworks
- Buyer beware, not everything works, including third party services like Clerk
- You will have to test existing Node apps to see how they work
- Great for Deno specific apps

Demos

Questions?