Deno

A Javascript Runtime

Deno Background

Ryan Dahl’s regrets with Node

- No Promises (at the start)
- Node JS and Web JS had a different experience
- Huge Bundles (node_module sizes)

Thus Deno was Created

Deno Runtime

Built In Rust (Using the V8 Engine)

JavaScript, TypeScript, and WebAssembly runtime with secure defaults

- Also created by Ryan Dahl (Creator of Node)
- Provides web platform functionality and adopts web platform standards
- Supports TypeScript out of the box
- Has a set of reviewed (audited) standard modules that are guaranteed to work with Deno (deno.land/std)
- Can bundle scripts into a single JavaScript file or executable

Install & Create New App

- curl -fsSL https://deno.land/install.sh | sh – Bash
- brew install deno – Homebrew

- deno init

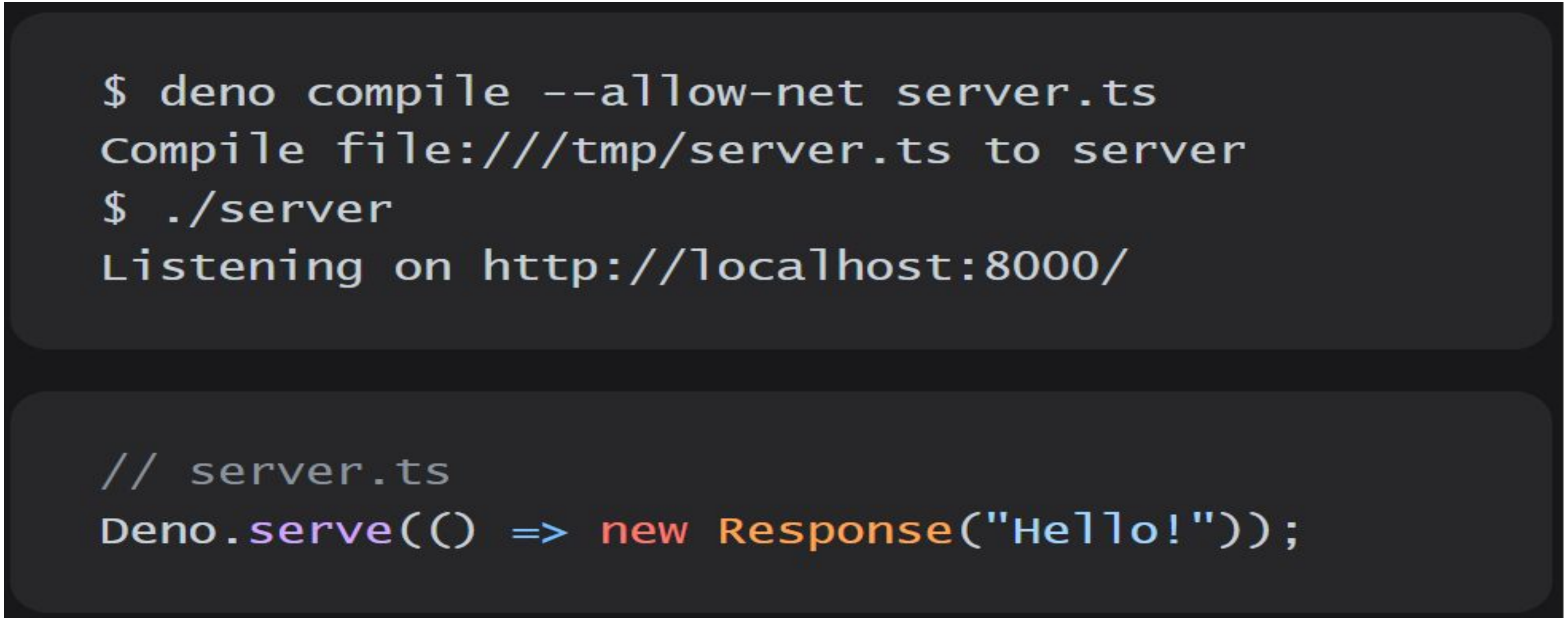
Deno Tooling

Provided built-in development tooling

- Linter - `deno lint`
- Test Runner - `deno test`
- Formatter - `deno fmt`
- Language server for your editor

Simple Server in Deno

deno run --allow-net server.js



```
$ deno compile --allow-net server.ts
Compile file:///tmp/server.ts to server
$ ./server
Listening on http://localhost:8000/
```

```
// server.ts
Deno.serve(() => new Response("Hello!"));
```

Gotchas

- With Deno being secure by default, there is a tendency to run deno with a command like,
  - `deno run --unstable -A somefile.ts`
  - `deno run --unstable -A somefile.js`
- Deno-Node compatibility mode was removed in version 1.25
  - NPM compatibility can still be used for frontend apps

Node vs. Deno Cheat Sheet

Node	Deno
node file.js	deno run file.js
npm i -g	deno install
npm run	deno task
package.json	deno.json
eslint	deno lint
prettier	deno fmt
jest/mocha/chai/etc	deno test

Questions

Connect with Me

- linkedin.com/in/kctolli
- developer@ktolliver.org
- ktolliver.org