DENO, INTRODUCTION AND PRACTICAL DEMOS

MARKO MANOJLOVIĆ
SOFTWARE ENGINEER AT QUANTOX

QUANTOX
SOFTWARE DEVELOPMENT & IT CONSULTING

Deno

- Runtime for JavaScript/TypeScript
- Created by Ryan Dahl, original creator of Node.js
- How to pronounce (Deno vs Dino)
- Based on Rust and V8 JS engine
- Focused on productivity
- Development ecosystem
- https://github.com/markoprogrammer/deno-examples

QUANTOX
SOFTWARE DEVELOPMENT & IT CONSULTING

Deno Value Propositions

- Secure by default (need to be added explicit permissions)
- TypeScript out of the box (zero configuration and setup)
- Ships only a single executable file
- Has built-in utilities (code formatter, linter, inspector, docs generator and etc.)
- Standard modules that are guaranteed to work with Deno (https://deno.land/std)

QUANTOX
SOFTWARE DEVELOPMENT & IT CONSULTING

Features

- Decentralized Packages
- Standard Library
- Modern JS
- ES Modules
- Built in Testing
- Browser Compatible API
- Web assembly

Installation

- Mac/Linux

```
curl -fsSL https://deno.land/x/install/install.sh | sh
```

- Mac

```
brew install deno
```

- Windows

a) `iwr https://deno.land/x/install/install.ps1 -useb | iex`

b) `choco install deno`

QUANTOX
SOFTWARE DEVELOPMENT & IT CONSULTING

deno run

- Basic example

  ```
  deno run https://raw.githubusercontent.com/markoprogrammer/deno-examples/master/deno-example-1.js
  ```

- Basic server

  ```
  deno run --allow-net https://raw.githubusercontent.com/markoprogrammer/deno-examples/master/deno-example-2.js
  ```

-

QUANTOX
SOFTWARE DEVELOPMENT & IT CONSULTING

Deno upgrade

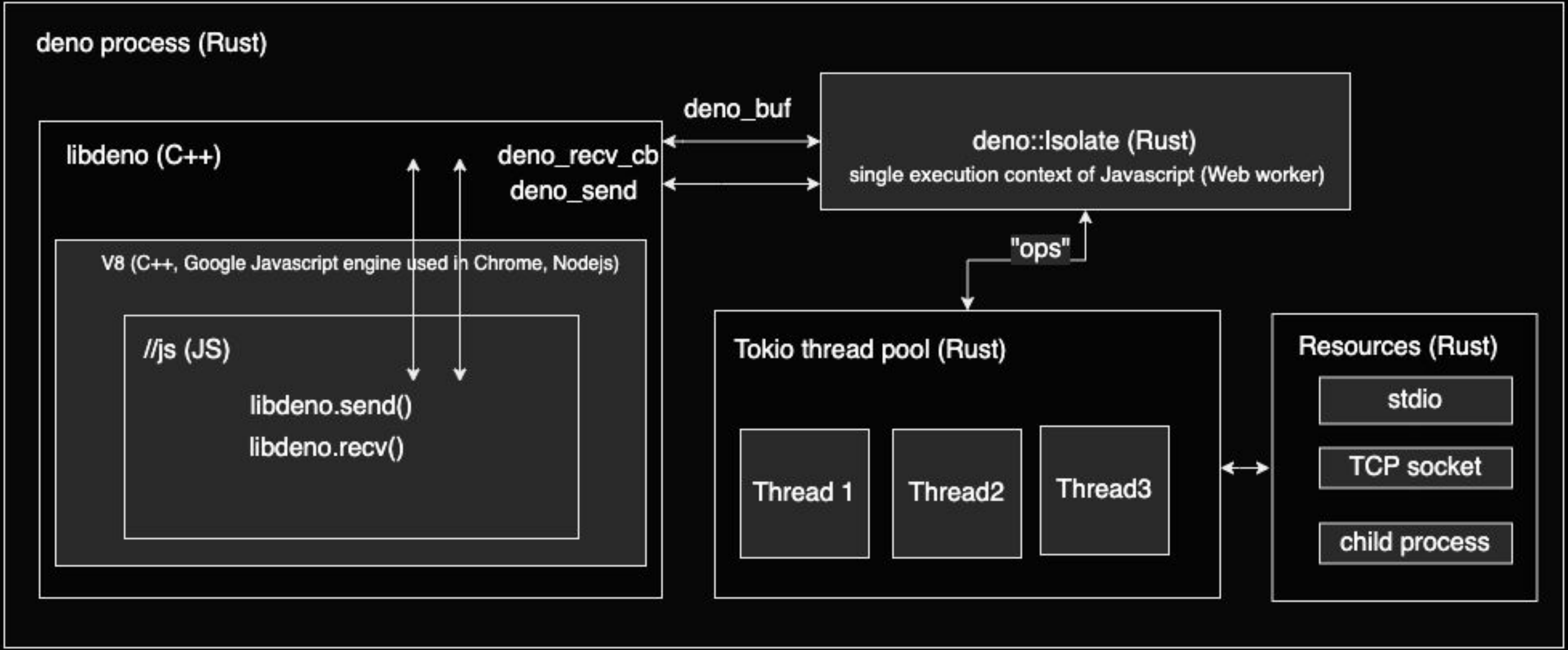
- Latest revision

```
deno upgrade
```

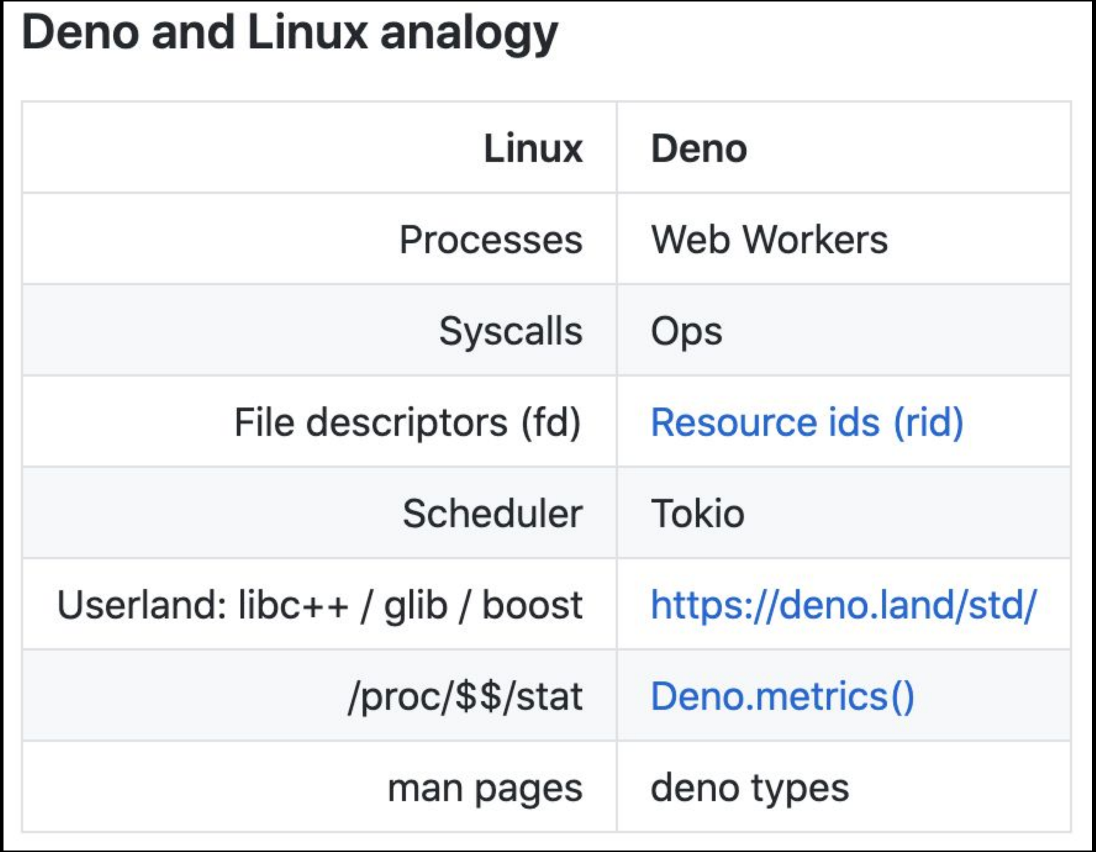
- Upgrade/Downgrade to specific revision

```
deno upgrade --version 1.3.1
```

Architecture of Deno runtime (Tokio event-driven non blocking I/O and V8)



Architecture of Deno runtime



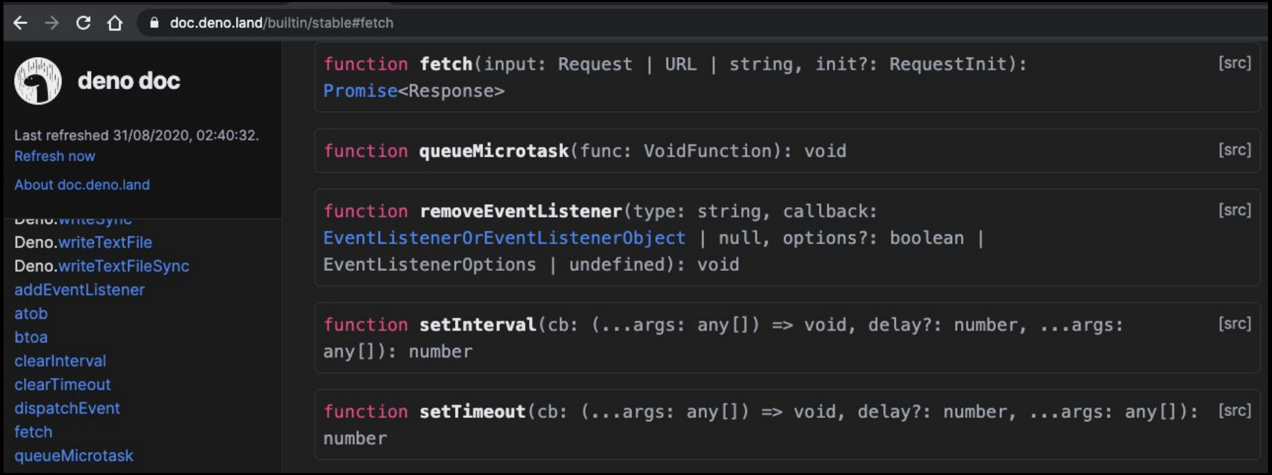
Deno and Linux analogy

Linux	Deno
Processes	Web Workers
Syscalls	Ops
File descriptors (fd)	Resource ids (rid)
Scheduler	Tokio
Userland: libc++ / glib / boost	https://deno.land/std/
/proc/$$/stat	Deno.metrics()
man pages	deno types

QUANTOX
SOFTWARE DEVELOPMENT & IT CONSULTING

Deno runtime documentation

- https://doc.deno.land/builtin/stable
- All runtime APIs
- Documentation of all runtime functions
- How to extend Deno using Rust plugins
- How to embed Deno



doc.deno.land/builtin/stable#fetch

deno doc

Last refreshed 31/08/2020, 02:40:32.
Refresh now

About doc.deno.land

Deno.writeSync
Deno.writeTextFile
Deno.writeTextFileSync
addEventListener
atob
btoa
clearInterval
clearTimeout
dispatchEvent
fetch
queueMicrotask

```
function fetch(input: Request | URL | string, init?: RequestInit):
Promise<Response>
```
[src]

```
function queueMicrotask(func: VoidFunction): void
```
[src]

```
function removeEventListener(type: string, callback:
EventListenerOrEventListenerObject | null, options?: boolean |
EventListenerOptions | undefined): void
```
[src]

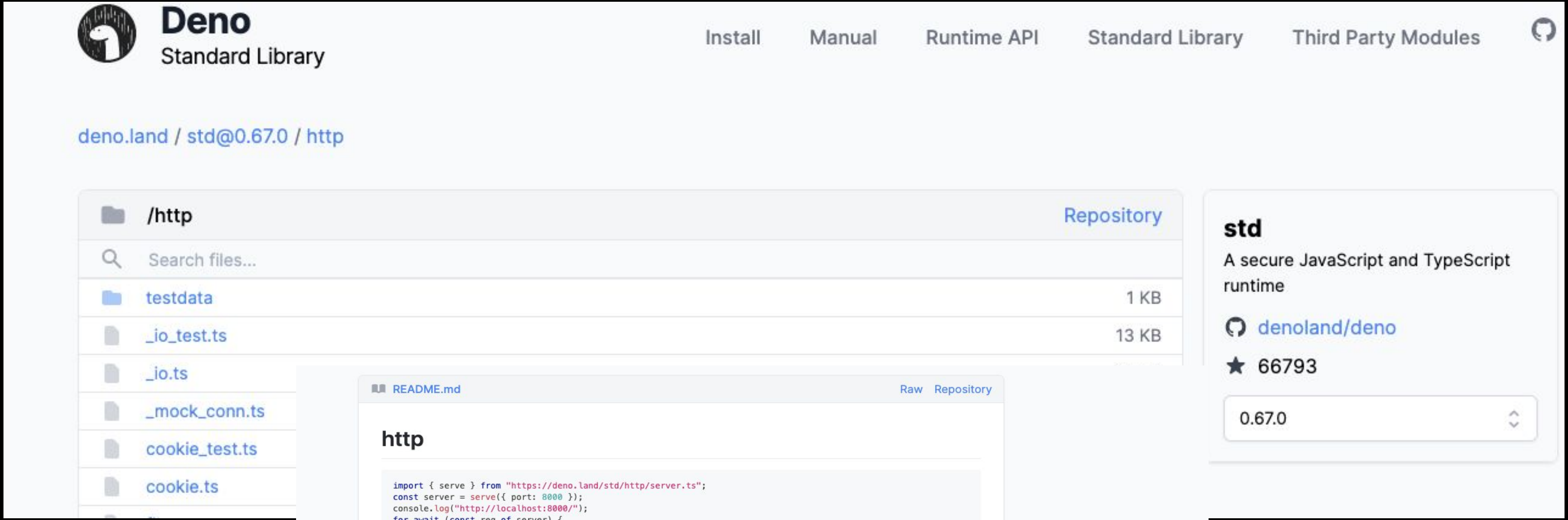
```
function setInterval(cb: (...args: any[]) => void, delay?: number, ...args:
any[]): number
```
[src]

```
function setTimeout(cb: (...args: any[]) => void, delay?: number, ...args: any[]):
number
```
[src]

QUANTOX
SOFTWARE DEVELOPMENT & IT CONSULTING

Standard Modules

- https://deno.land/std@0.67.0
- Audited standard modules reviewed by Deno core team
- Guaranteed to work with a specific Deno version
- Independent from deno runtime code under /std
- Import with URL standard module that you need, not all



Deno
Standard Library

Install · Manual · Runtime API · Standard Library · Third Party Modules

deno.land / std@0.67.0 / http

/http — Repository

testdata	1 KB
_io_test.ts	13 KB
_io.ts	
_mock_conn.ts	
cookie_test.ts	
cookie.ts	

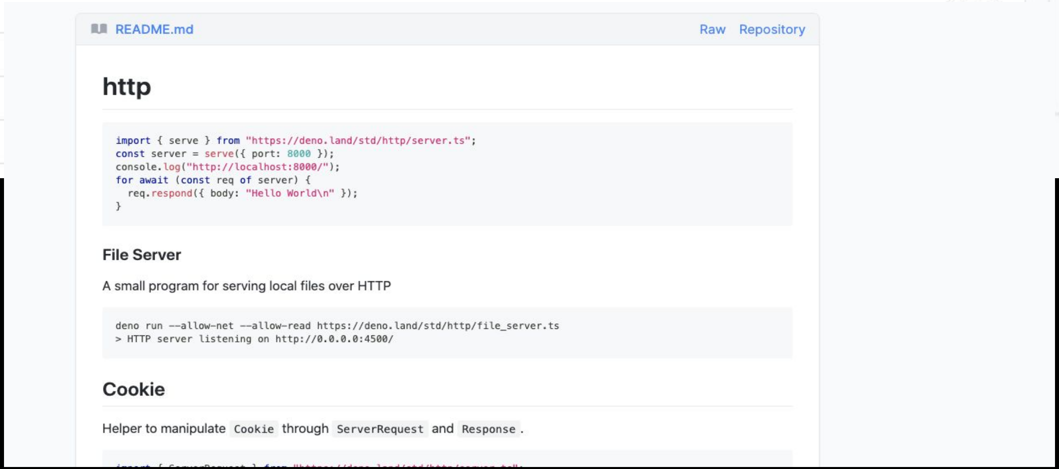
std

A secure JavaScript and TypeScript runtime

denoland/deno

★ 66793

0.67.0



README.md — Raw · Repository

http

```
import { serve } from "https://deno.land/std/http/server.ts";
const server = serve({ port: 8000 });
console.log("http://localhost:8000/");
for await (const req of server) {
  req.respond({ body: "Hello World\n" });
}
```

File Server

A small program for serving local files over HTTP

```
deno run --allow-net --allow-read https://deno.land/std/http/file_server.ts
> HTTP server listening on http://0.0.0.0:4500/
```

Cookie

Helper to manipulate `Cookie` through `ServerRequest` and `Response`.

Third party modules

- https://deno.land/x - Third party module hosting provided:
- Import module from any location on the web

    Like Github, private server/CDN

- Build in doc generator for viewing documentation of third party module

    https://doc.deno.land/

    https://doc.deno.land/https/raw.githubusercontent.com/markoprogrammer/deno-examples/master/deno-example-3.ts

Deno doc and Deno info

- deno doc

```
deno doc https://raw.githubusercontent.com/markoprogrammer/deno-examples/master/deno-example-3.ts
```

- deno info

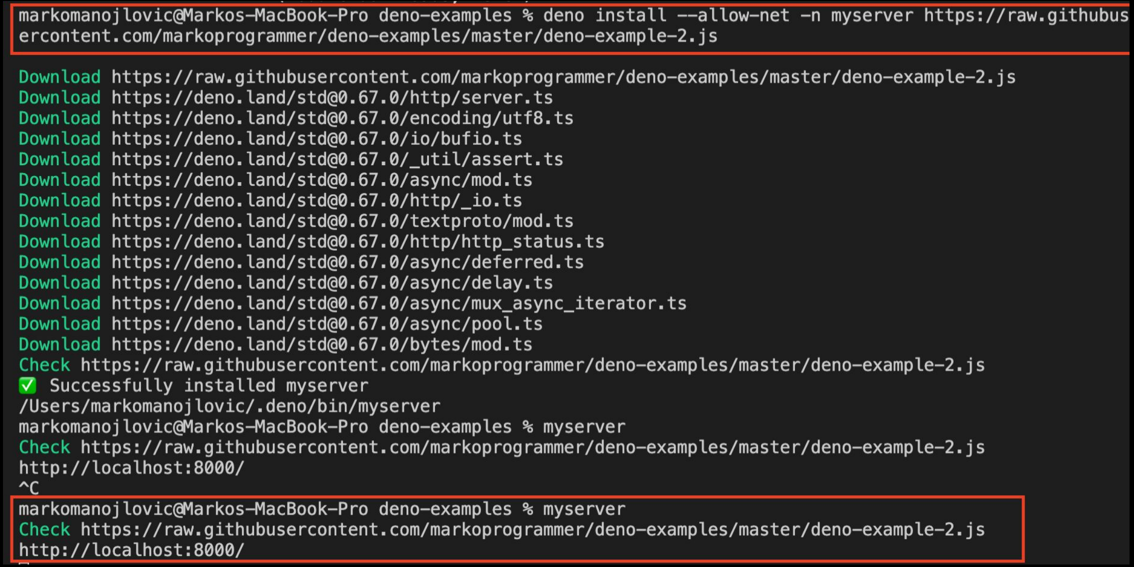
```
deno info https://raw.githubusercontent.com/markoprogrammer/deno-examples/master/deno-example-2.js
```

QUANTOX
SOFTWARE DEVELOPMENT & IT CONSULTING

deno install - Install deno module as script in terminal

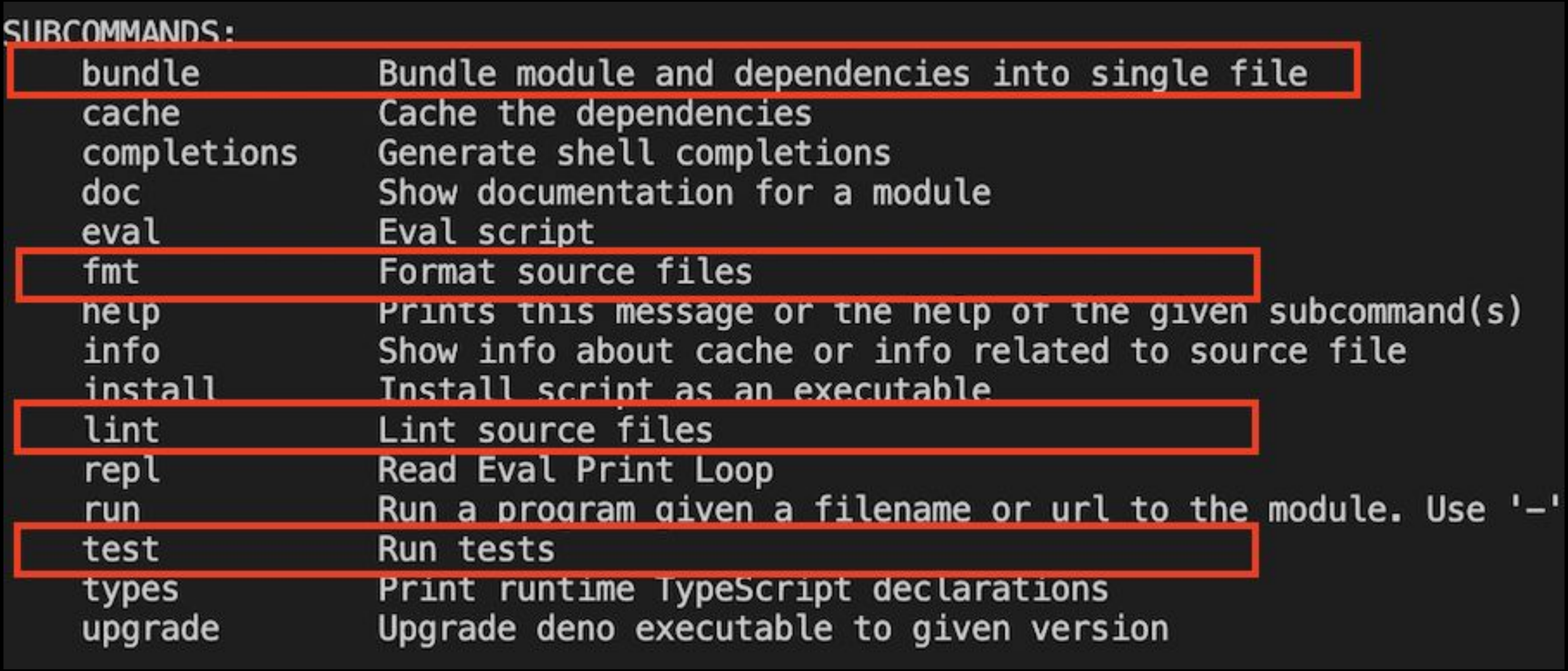
- deno install --allow-net -n myserver https://raw.githubusercontent.com/markoprogrammer/deno-examples/master/deno-example-2.js
- Then installed as available command in terminal run with:

myserver



```
markomanojlovic@Markos-MacBook-Pro deno-examples % deno install --allow-net -n myserver https://raw.githubus
ercontent.com/markoprogrammer/deno-examples/master/deno-example-2.js

Download https://raw.githubusercontent.com/markoprogrammer/deno-examples/master/deno-example-2.js
Download https://deno.land/std@0.67.0/http/server.ts
Download https://deno.land/std@0.67.0/encoding/utf8.ts
Download https://deno.land/std@0.67.0/io/bufio.ts
Download https://deno.land/std@0.67.0/_util/assert.ts
Download https://deno.land/std@0.67.0/async/mod.ts
Download https://deno.land/std@0.67.0/http/_io.ts
Download https://deno.land/std@0.67.0/textproto/mod.ts
Download https://deno.land/std@0.67.0/http/http_status.ts
Download https://deno.land/std@0.67.0/async/deferred.ts
Download https://deno.land/std@0.67.0/async/delay.ts
Download https://deno.land/std@0.67.0/async/mux_async_iterator.ts
Download https://deno.land/std@0.67.0/async/pool.ts
Download https://deno.land/std@0.67.0/bytes/mod.ts
Check https://raw.githubusercontent.com/markoprogrammer/deno-examples/master/deno-example-2.js
✅ Successfully installed myserver
/Users/markomanojlovic/.deno/bin/myserver
markomanojlovic@Markos-MacBook-Pro deno-examples % myserver
Check https://raw.githubusercontent.com/markoprogrammer/deno-examples/master/deno-example-2.js
http://localhost:8000/
^C
markomanojlovic@Markos-MacBook-Pro deno-examples % myserver
Check https://raw.githubusercontent.com/markoprogrammer/deno-examples/master/deno-example-2.js
http://localhost:8000/
```

deno help



```
SUBCOMMANDS:
    bundle         Bundle module and dependencies into single file
    cache          Cache the dependencies
    completions    Generate shell completions
    doc            Show documentation for a module
    eval           Eval script
    fmt            Format source files
    help           Prints this message or the help of the given subcommand(s)
    info           Show info about cache or info related to source file
    install        Install script as an executable
    lint           Lint source files
    repl           Read Eval Print Loop
    run            Run a program given a filename or url to the module. Use '-'
    test           Run tests
    types          Print runtime TypeScript declarations
    upgrade        Upgrade deno executable to given version
```

QUANTOX
SOFTWARE DEVELOPMENT & IT CONSULTING

Deno project and dependency

- Decentralized packages
- No npm/package.json packages
- Packages imported from URL
- Modules are cached to hard drive after first load
- Out of the box available set of development tools

    formater, linter, debugger, doc generators, testing tools, dependency info and cache, switching between versions and etc (future ecosystem)

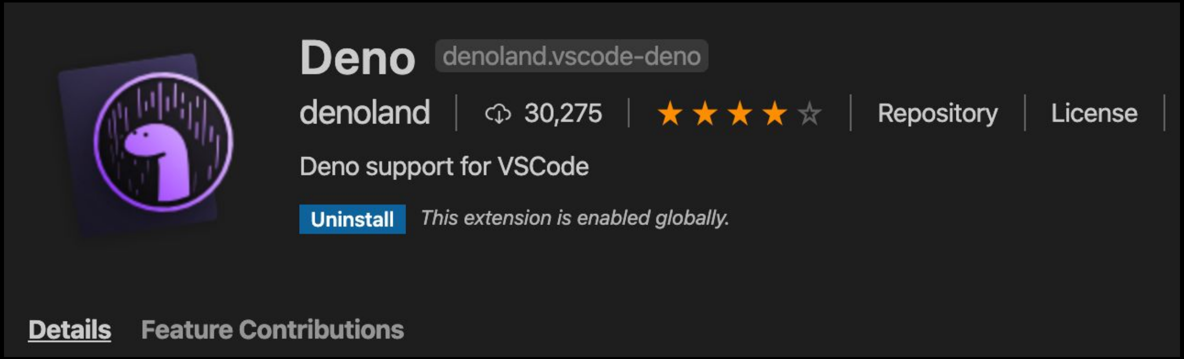
    Supported vscode Deno extension

- Security Allow flags - https://deno.land/manual/getting_started/permissions

QUANTOX
SOFTWARE DEVELOPMENT & IT CONSULTING

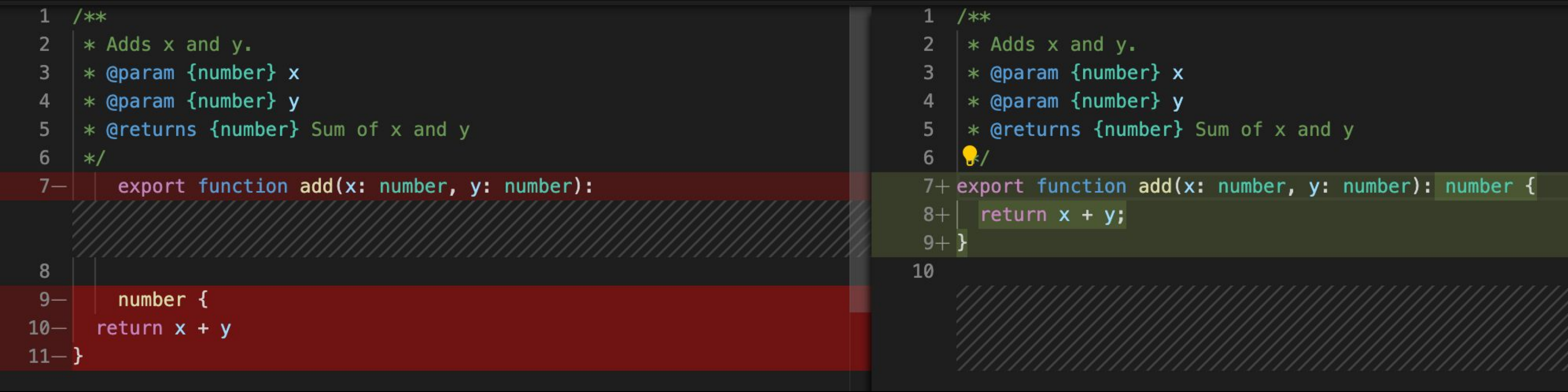
Extensions

- .js or .ts (possible also .tsx or .jsx)
- Deno has a built-in JSX support powered by TypeScript.
- By default, JSX files (.jsx, .tsx) will be transformed by React.createElement(). So you must import React on the head of your jsx file.
- Deno official vscode extension


Deno denoland.vscode-deno
denoland | 30,275 | ★★★★☆ | Repository | License
Deno support for VSCode
Uninstall *This extension is enabled globally.*

Deno fmt and Deno lint

- deno fmt deno-example-3.ts



```
/**
 * Adds x and y.
 * @param {number} x
 * @param {number} y
 * @returns {number} Sum of x and y
 */
-   export function add(x: number, y: number):

-   number {
-  return x + y
-}
```

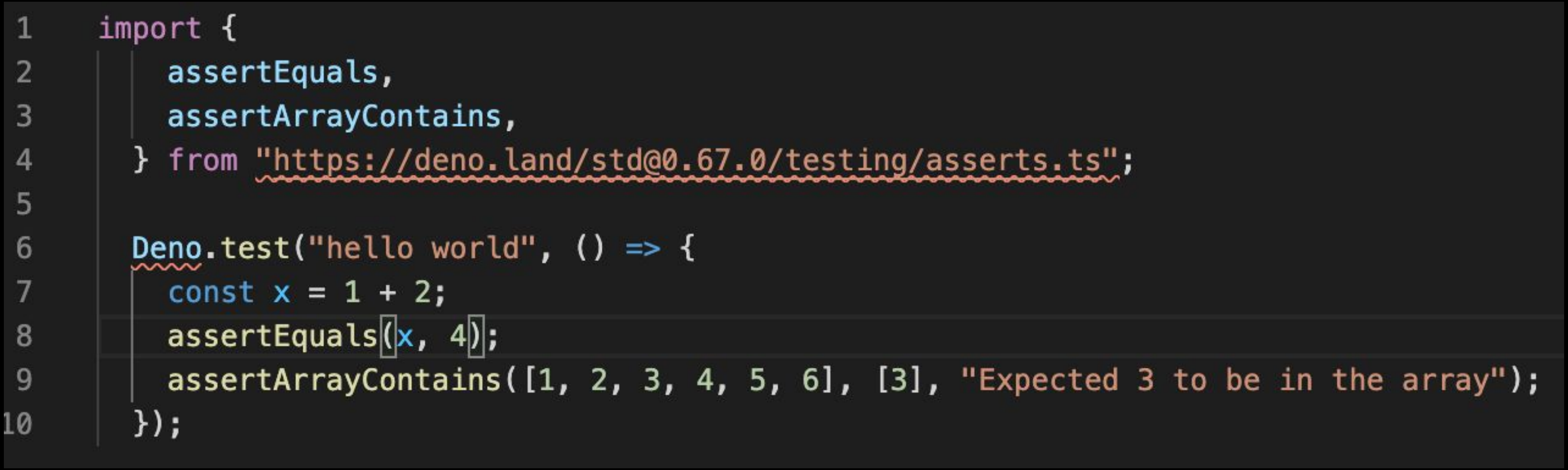
```
/**
 * Adds x and y.
 * @param {number} x
 * @param {number} y
 * @returns {number} Sum of x and y
 */
+export function add(x: number, y: number): number {
+  return x + y;
+}

```

- deno lint --unstable - Currently unstable and not ready yet
- Equivalent to ESlint and Prettier out of the box supported tools

QUANTOX
SOFTWARE DEVELOPMENT & IT CONSULTING

Deno test runner

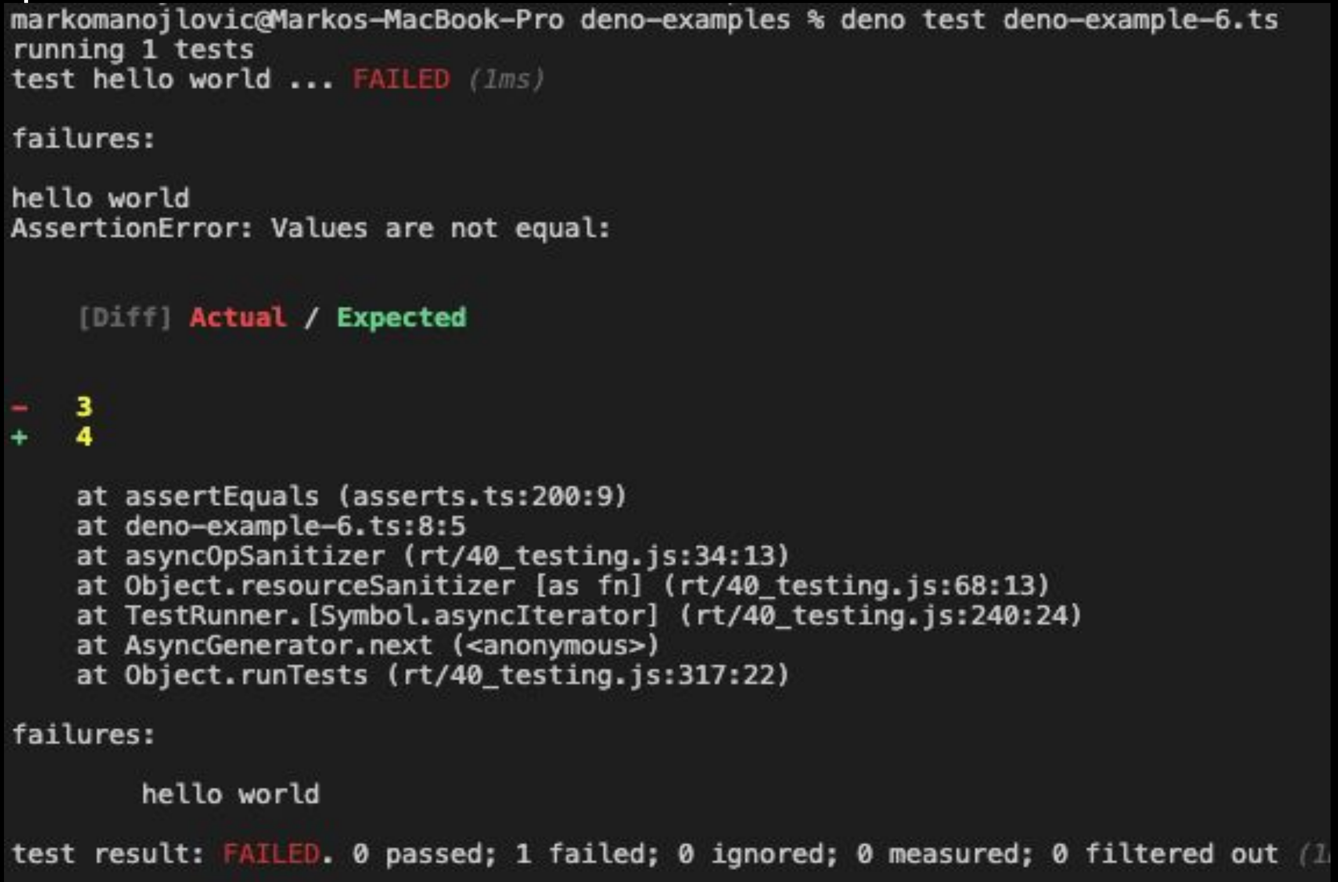
- Built-in test runner for testing Javascript and Typscript code
- Supported Assertions as part of standard module



```
import {
    assertEquals,
    assertArrayContains,
  } from "https://deno.land/std@0.67.0/testing/asserts.ts";

  Deno.test("hello world", () => {
    const x = 1 + 2;
    assertEquals(x, 4);
    assertArrayContains([1, 2, 3, 4, 5, 6], [3], "Expected 3 to be in the array");
  });
```

QUANTOX
SOFTWARE DEVELOPMENT & IT CONSULTING

- deno test deno-example-6.ts



```
markomanojlovic@Markos-MacBook-Pro deno-examples % deno test deno-example-6.ts
running 1 tests
test hello world ... FAILED (1ms)

failures:

hello world
AssertionError: Values are not equal:


    [Diff] Actual / Expected


-   3
+   4

    at assertEquals (asserts.ts:200:9)
    at deno-example-6.ts:8:5
    at asyncOpSanitizer (rt/40_testing.js:34:13)
    at Object.resourceSanitizer [as fn] (rt/40_testing.js:68:13)
    at TestRunner.[Symbol.asyncIterator] (rt/40_testing.js:240:24)
    at AsyncGenerator.next (<anonymous>)
    at Object.runTests (rt/40_testing.js:317:22)

failures:

        hello world

test result: FAILED. 0 passed; 1 failed; 0 ignored; 0 measured; 0 filtered out (1
```

Deno debugger

- Could be configured debugger
- With Chrome

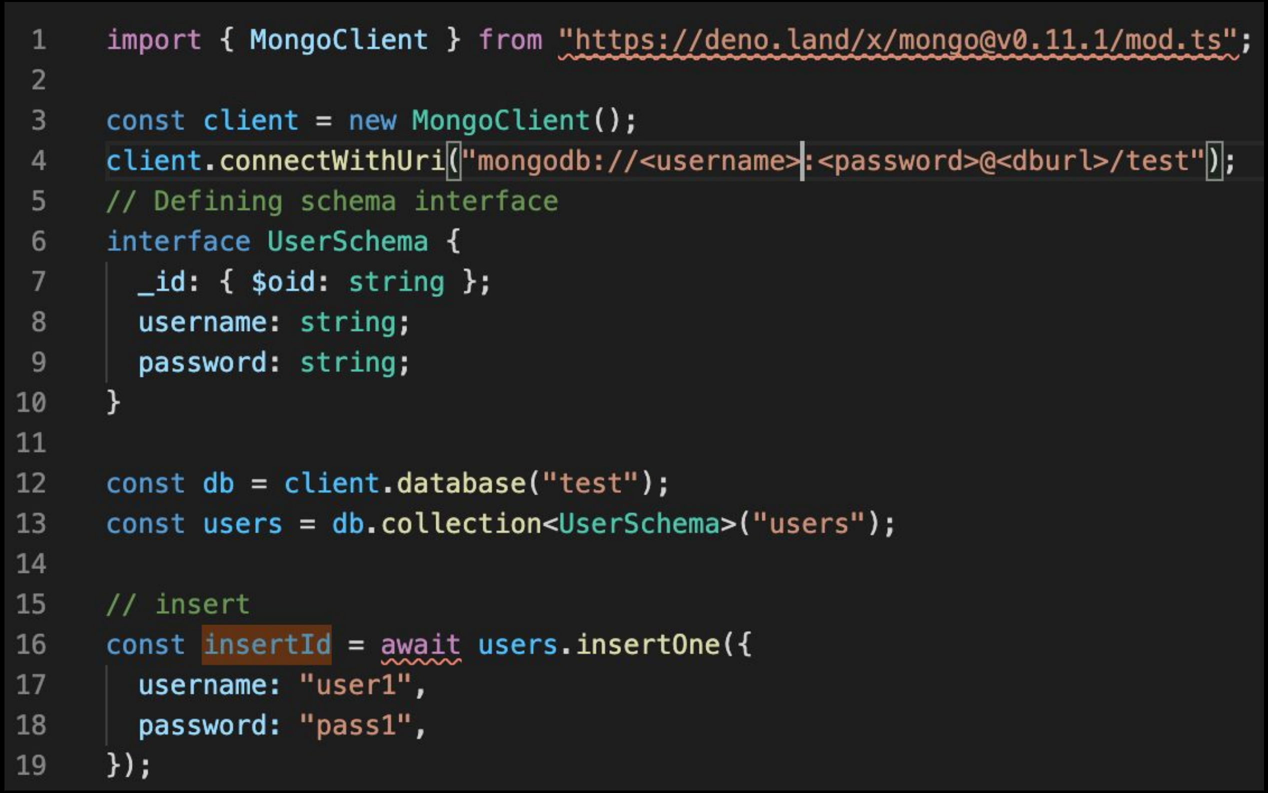
  Devtools

  VSCode

  or JetBrains IDEs

QUANTOX
SOFTWARE DEVELOPMENT & IT CONSULTING

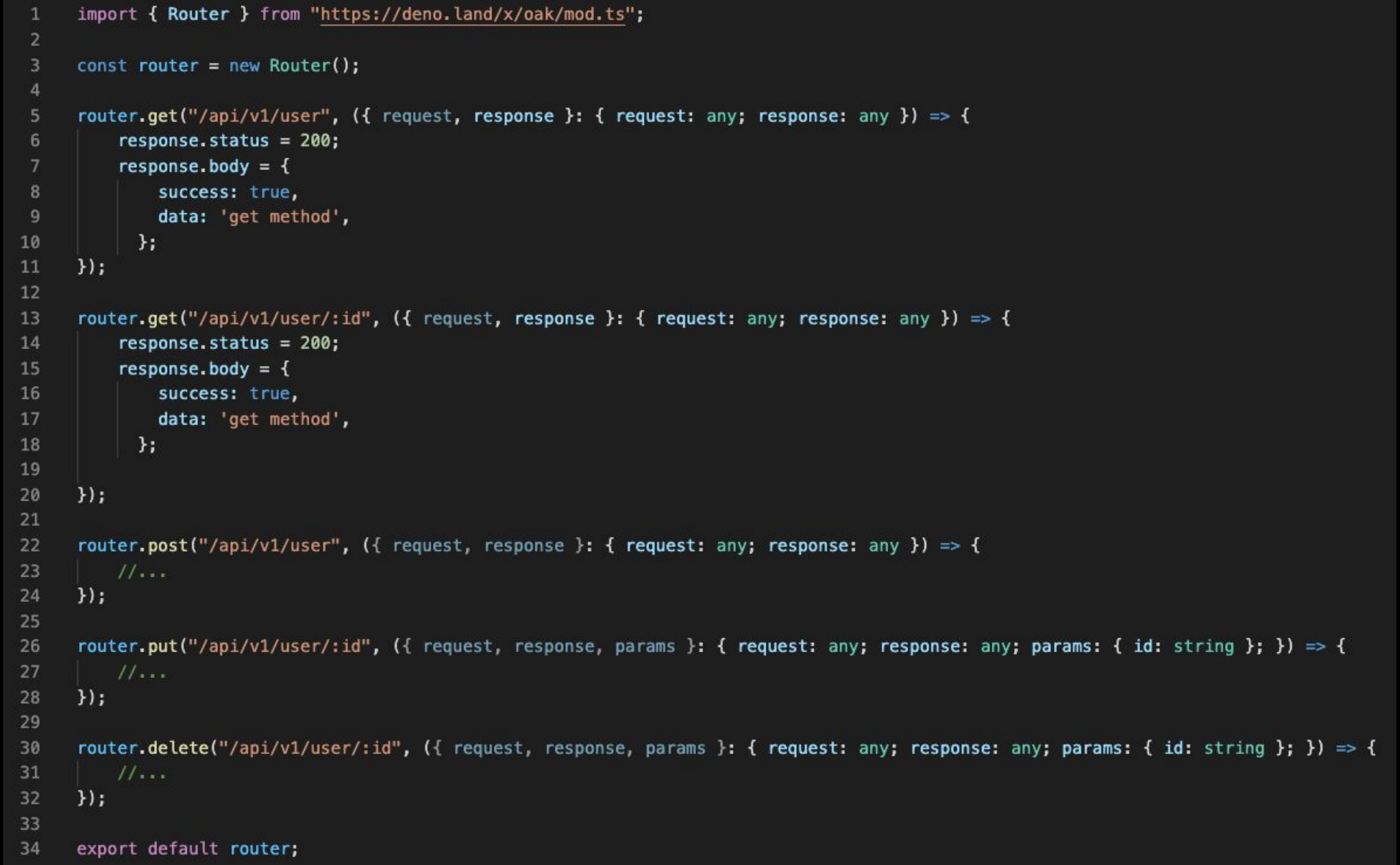
Deno mongodb connection (deno_mongo driver, based on Rust)


```
import { MongoClient } from "https://deno.land/x/mongo@v0.11.1/mod.ts";

const client = new MongoClient();
client.connectWithUri("mongodb://<username>:<password>@<dburl>/test");
// Defining schema interface
interface UserSchema {
  _id: { $oid: string };
  username: string;
  password: string;
}

const db = client.database("test");
const users = db.collection<UserSchema>("users");

// insert
const insertId = await users.insertOne({
  username: "user1",
  password: "pass1",
});
```

QUANTOX
SOFTWARE DEVELOPMENT & IT CONSULTING

Deno build rest APIs

- Oak middleware - https://github.com/oakserver/oak
- Currently most popular one
- Similar to NodeJs Express middleware
- Support static hosting
- Support Router



```typescript
import { Router } from "https://deno.land/x/oak/mod.ts";

const router = new Router();

router.get("/api/v1/user", ({ request, response }: { request: any; response: any }) => {
  response.status = 200;
  response.body = {
    success: true,
    data: 'get method',
   };
});

router.get("/api/v1/user/:id", ({ request, response }: { request: any; response: any }) => {
  response.status = 200;
  response.body = {
    success: true,
    data: 'get method',
   };

});

router.post("/api/v1/user", ({ request, response }: { request: any; response: any }) => {
  //...
});

router.put("/api/v1/user/:id", ({ request, response, params }: { request: any; response: any; params: { id: string }; }) => {
  //...
});

router.delete("/api/v1/user/:id", ({ request, response, params }: { request: any; response: any; params: { id: string }; }) => {
  //...
});

export default router;
```

Deno Demo

- Simple Rest APIs server with crud
- Integration with MongoDB
- React client application
- React SSR demo (ReactDOMServer.renderToString and with ReactDOM.hydrate)

QUANTOX
SOFTWARE DEVELOPMENT & IT CONSULTING

Conclusion

- Rayan Dahl smart engineer
- All bad things from Node are solved here
- Focused on uniform ecosystem, powered by open source community, and all will grow in some standard manner
- There will not be question what lib to use for testing http and etc...

Advantages and Disadvantages of Deno

- New technology
- NodeJs exist 10 years and has big community and compatible open source modules and libraries
- Deno will grow but need time, if becomes competitive with NodeJs early adapters will benefit with their skills
- Currently not compatible with existing popular services like Firebase and etc
- Because community is small and not supported all known concepts and libraries
- Is opportunity to write some open source module in this early phase, from which you will benefit in your personal career

Personal thoughts of Deno

- First time when I heard about Deno
- This year big hype
- Then I was little bit disappointed, competitor of node but not support all in this early phase
- Then after I get deeper...
- I saw how much this is good planned and designed as development ecosystem for Javascript projects, standardized tools and resources driven and improved by power of open source community, example by Apple Ecosystem
- Almost zero project configuration

QUANTOX
SOFTWARE DEVELOPMENT & IT CONSULTING

- https://github.com/markoprogrammer/deno-examples

marko.manojlovic@quantox.com

Thank you for your attention!

- Questions?