node|deno|bun & dcp-client

Wrestling with JS Runtime compatability

Node (2009)

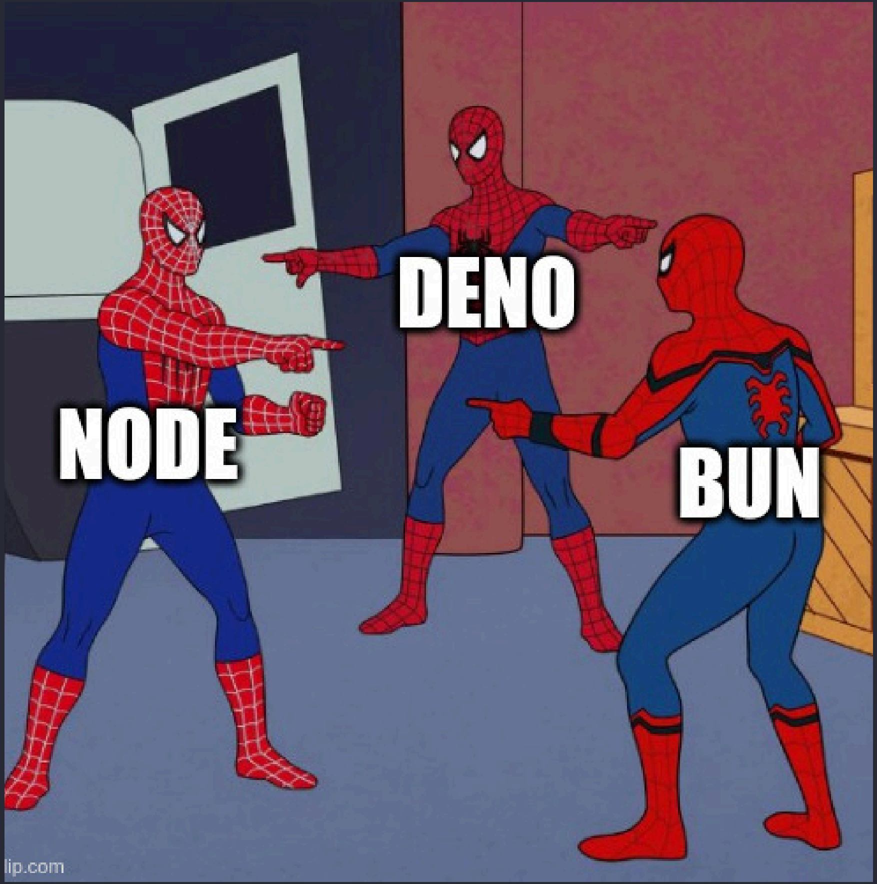
- The OG
- `npm` , pain
- V8

Deno (2018)

- An anagram of Node
- Better security, different handling of pkgs
- V8
- *Rust*

Bun (2021)

- The newest kid on the block
- Focused on perf, wants to replace Node
- JavaScriptCore
- *Zig*



Node & dcp-client

Initializing a project w/ `dcp-client` :

```
$ npm init --yes
$ npm add dcp-client
$ touch index.js
$ ls
index.js  node_modules  package-lock.json  package.json
```

Importing `dcp-client` :

```
const dcpClient = require('dcp-client');

dcpClient.init().then((dcp) => {
  // ...
});
```

Running the app:

```
node index.js
```

Node & dcp-client

Issues?

None!

Deno & dcp-client

Initializing a project:

```
$ deno init
$ ls
deno.jsonc  main.ts  main_bench.ts  main_test.ts
```

Importing `dcp-client` :

```
import dcpClient from 'npm:dcp-client';

const dcp = await dcpClient.init();

// ...
```

Running the app:

```
# Likely missing other necessary allow options.
deno run [--watch] --allow-env --allow-read main.ts
```

Deno & dcp-client

Issues?

- Slightly different implementation of `` `node:module` `` API. i.e., missing `` `require('node:module').Module` `` .
  - The fix: use `` `require('node:modules')` `` as a constructor instead of `` `require('node:module').Module` ``

Deno & dcp-client

Issues?

- No implementation of `node:vm` module. e.g., `require('node:vm').createContext()` .

  ```
  $ deno run --allow-env --allow-read main.ts
  error: Uncaught Error: Not implemented: createContext
  ...
  ```

  where

  > *The node:vm module enables compiling and running code within V8 Virtual Machine contexts. A common use case is to run the code in a different V8 Context. This means invoked code has a different global object than the invoking code.*

  - Can potentially come after the `ShadowRealm` proposal (Stage 3) [1]. i.e.,

  > *ShadowRealms are a distinct global environment, with its own global object containing its own intrinsics and built-ins (standard objects that are not bound to global variables, like the initial value of Object.prototype).*

---

1. https://github.com/denoland/deno/issues/18315#issuecomment-1477316394 ↩

Bun & dcp-client

Initializing a project:

```
$ bun init
$ bun add dcp-client
$ ls
bun.lockb  index.ts  node_modules  package.json  README.md  tsconfig.json
```

Importing `dcp-client` :

```
const dcpClient = require('dcp-client');

const dcp = await dcpClient.init();

// ...
```

Running the app:

```
bun run [--watch] index.ts
```

Bun & dcp-client

Issues?

- Slightly different implementation of `node:module` API. i.e., missing `require('node:module').Module` .
  - The fix: use `require('node:modules')` as a constructor instead of `require('node:module').Module`
- Different expected/known error messages for platform detection. e.g., "Module not found" errors, "Illegel return statement".
  - The fix: add checks for Bun specific error messages.

Bun & dcp-client

Issues?

- Bug in `` `require.cache` `` implementation. e.g.,

```
  console.log(require('dcp/dcp-url'));
  try
  {
    console.log(require('dcp/dcp-url'));
        ^
error: Cannot require module "dcp/dcp-url"
```

Minimal repro:

```js
require.cache.foo = {
  exports: {
    bar: 'foobar',
  },
};

console.log(require('foo'));

try {
  console.log(require('foo'));
} catch (error) {
  console.error(error);
}

console.log(require('foo'));
```

Outputs:

```
$ bun run --watch test.js
{
  bar: "foobar"
}
 5 | };
 6 |
 7 | console.log(require('foo'));
 8 |
 9 | try {
10 |    console.log(require('foo'));
            ^
error: Cannot require module "foo"
      at /home/bryan/src/github.com/bryan-hoang/dcp-bun-app/t
      at /home/bryan/src/github.com/bryan-hoang/dcp-bun-app/t
      at globalThis (/home/bryan/src/github.com/bryan-hoang/d

{
  bar: "foobar"
}
```

What has been done so far & Learnings

- I filed MRs in the `` `dcp` `` & `` `dcp-client` `` to try and improve on the issues I encountered so far.
- Deno & Bun developer experience is a breath of fresh air (compared to Node & NPM at least).
- https://github.com/bryan-hoang/dcp-deno-app
- https://github.com/bryan-hoang/dcp-bun-app

Next steps

- Deno
  - Wait for `node:vm` to be implemented. Or, try out `ShadowRealm` API once it’s available.
- Bun
  - File a bug report with the minimal repro for the module not found error code & `require.cache` implementation.

The end, for now.

Thank you for listening!

Questions?