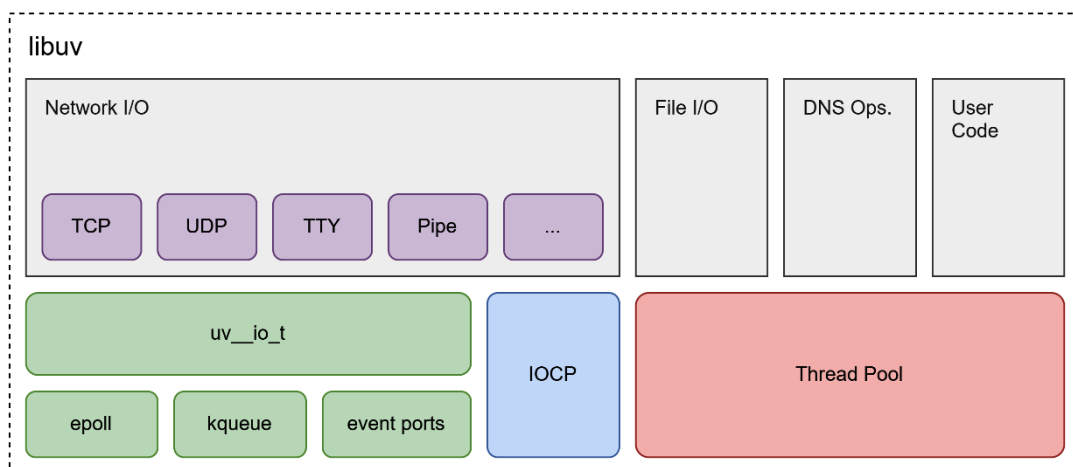
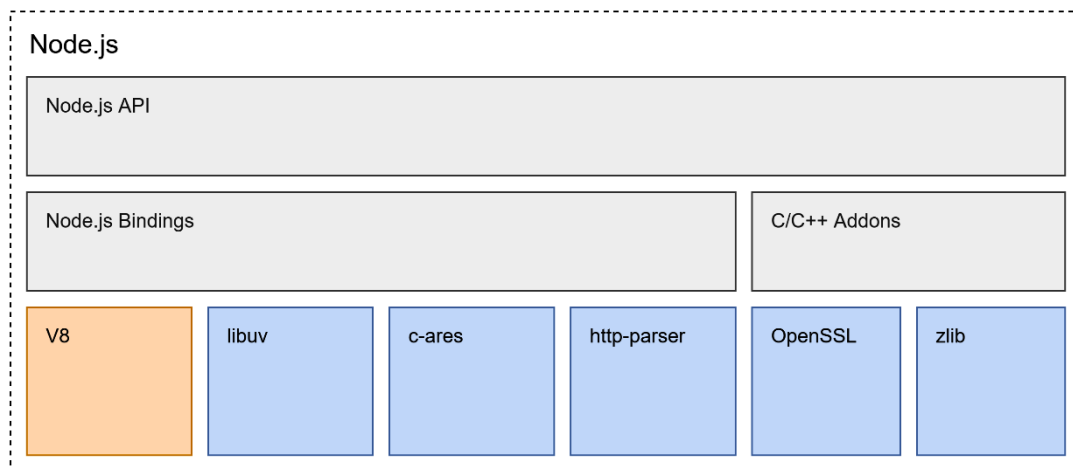
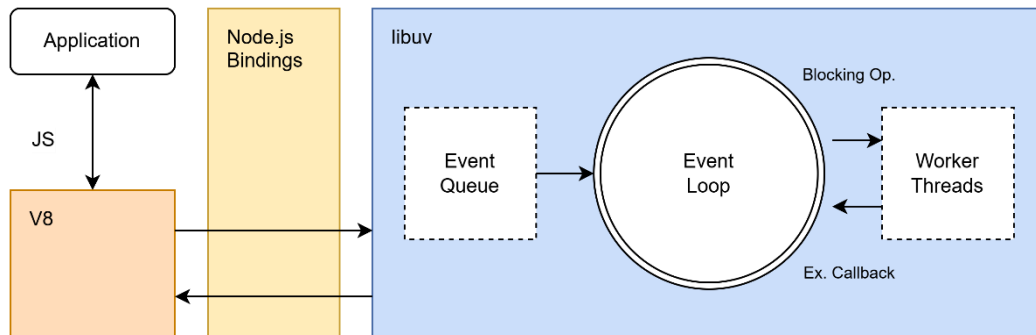
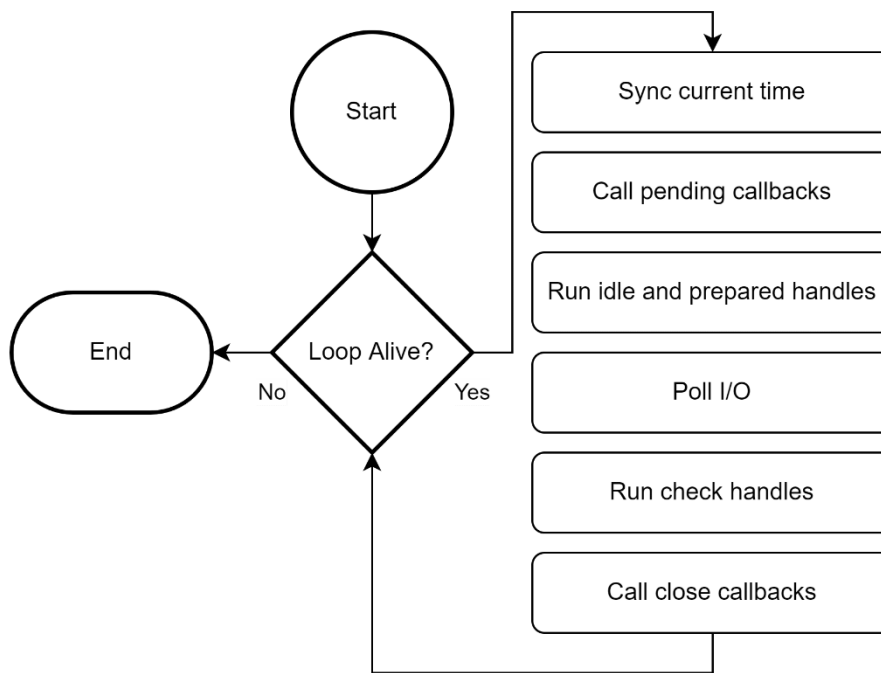


Chapter 01: Learning about the Internals of Node.js





← → ↻ Chrome | chrome://inspect/#devices

DevTools

Devices

Pages

Extensions

Apps

Shared workers

Service workers

Other

Discover USB devices

Discover network targets

Open dedicated DevTools for Node

Remote Target #LOCALHOST

Target (v14.18.1) trace

node[3420] file:///inspect

Port forwarding...

Configure...

DevTools

ProfilerConsoleSourcesMemoryPiral

FilesystemSnippets[eval]internal/timers.js x

+ Add folder to workspace

Sync changes in DevTools with the local filesystem

[Learn more about Workspaces](#)

```
483
484
485 function processTimers(now) {
486   debug('process timer lists %d', now);
487   nextExpiry = Infinity;
488
489   let list;
490   let ranAtLeastOneList = false;
491   while (list = timerListQueue.peek()) {
492     if (list.expiry > now) {
493       nextExpiry = list.expiry;
494       return refCount > 0 ? nextExpiry : -nextExpiry;
495     }
496     if (ranAtLeastOneList)
497       runNextTicks();
498     else
499       ranAtLeastOneList = true;
500     listOnTimeout(list, now);
501   }
502   return 0;
503 }
504
505 function listOnTimeout(list, now) {
506   const msec = list.msec;
507
508   debug('timeout callback %d', msec);
509
510   let ranAtLeastOneTimer = false;
511   let timer;
512   while (timer = L.peek(list)) {
513     const diff = now - timer._idleStart;
514
515     // Check if this loop iteration is too early for the next timer.
516     // This happens if there are more timers scheduled for later in the li
517     if (diff < msec) {
```

Debugger paused

Watch

Breakpoints

No breakpoints

Scope

Local

this: process

list: undefined

now: 540334

ranAtLeastOneList: undefined

Closure (getTimerCallbacks)

Closure

Globalglobal

Call Stack

processTimers

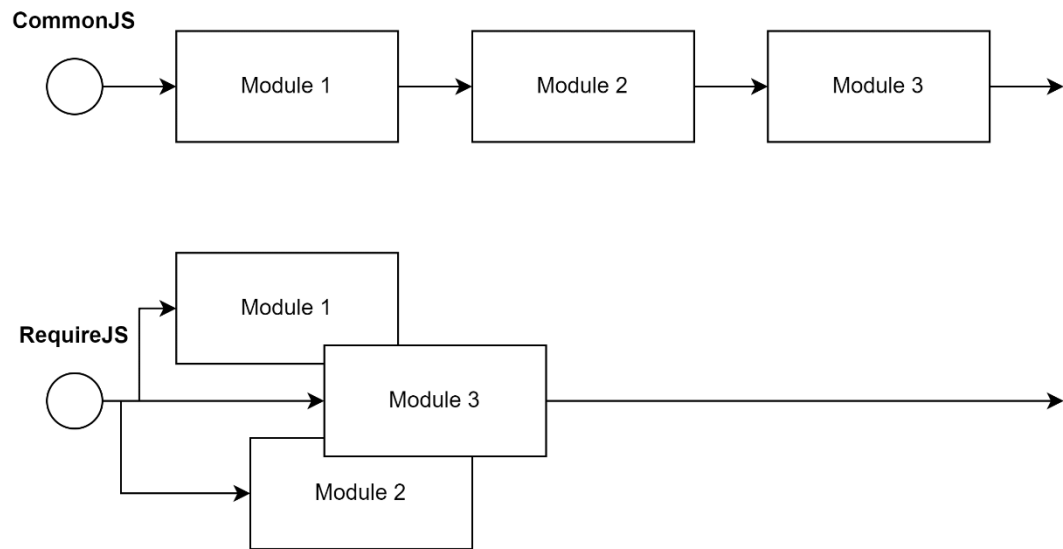
internal/timers.js:486

{}

Line 484, Column 1

Coverage: n/a

Chapter 02: Dividing Code into Modules and Packages

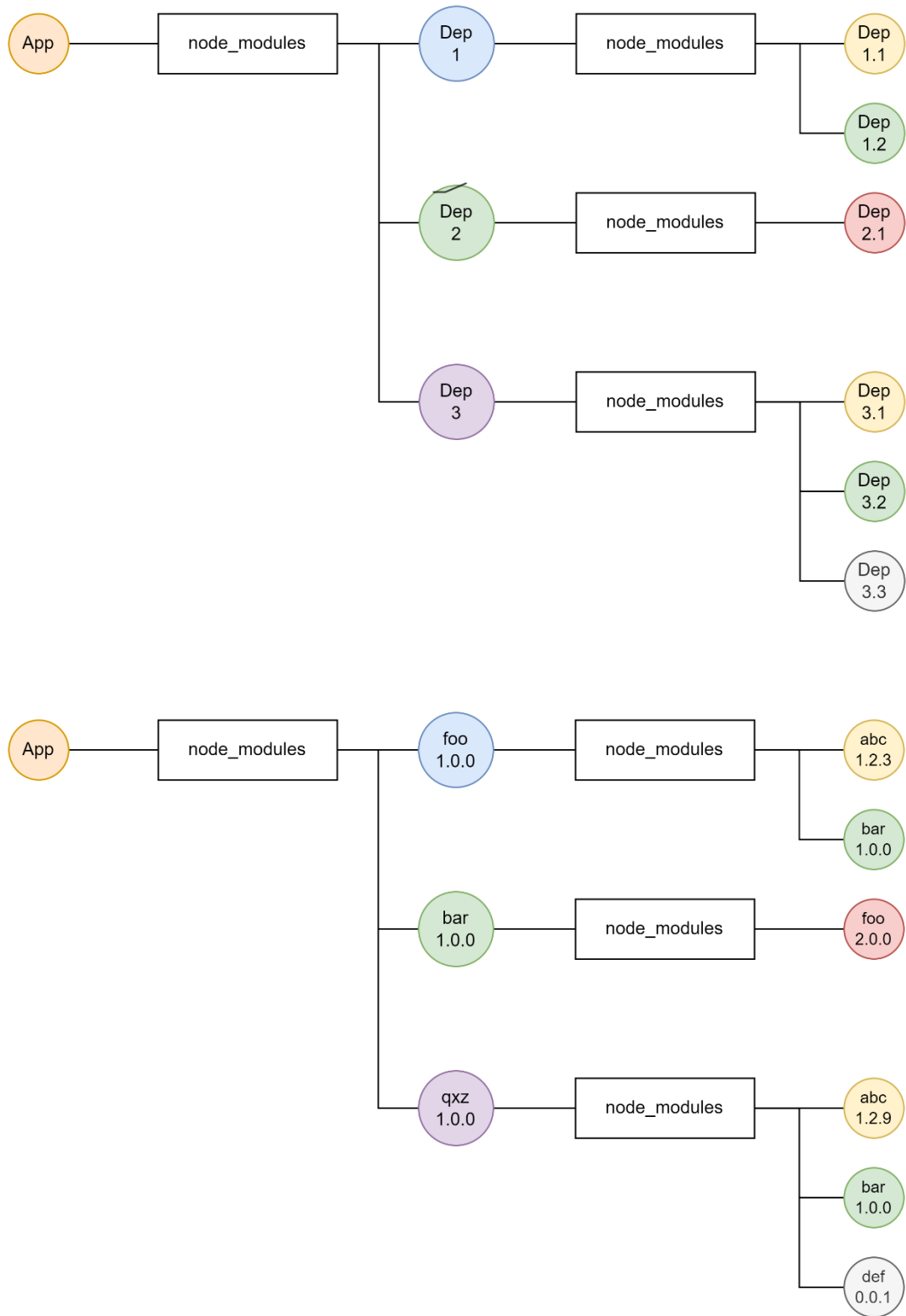


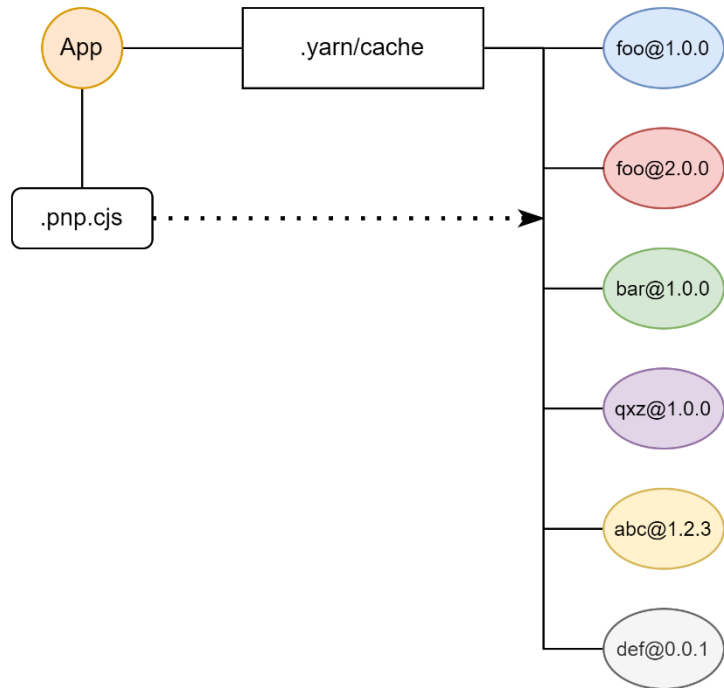
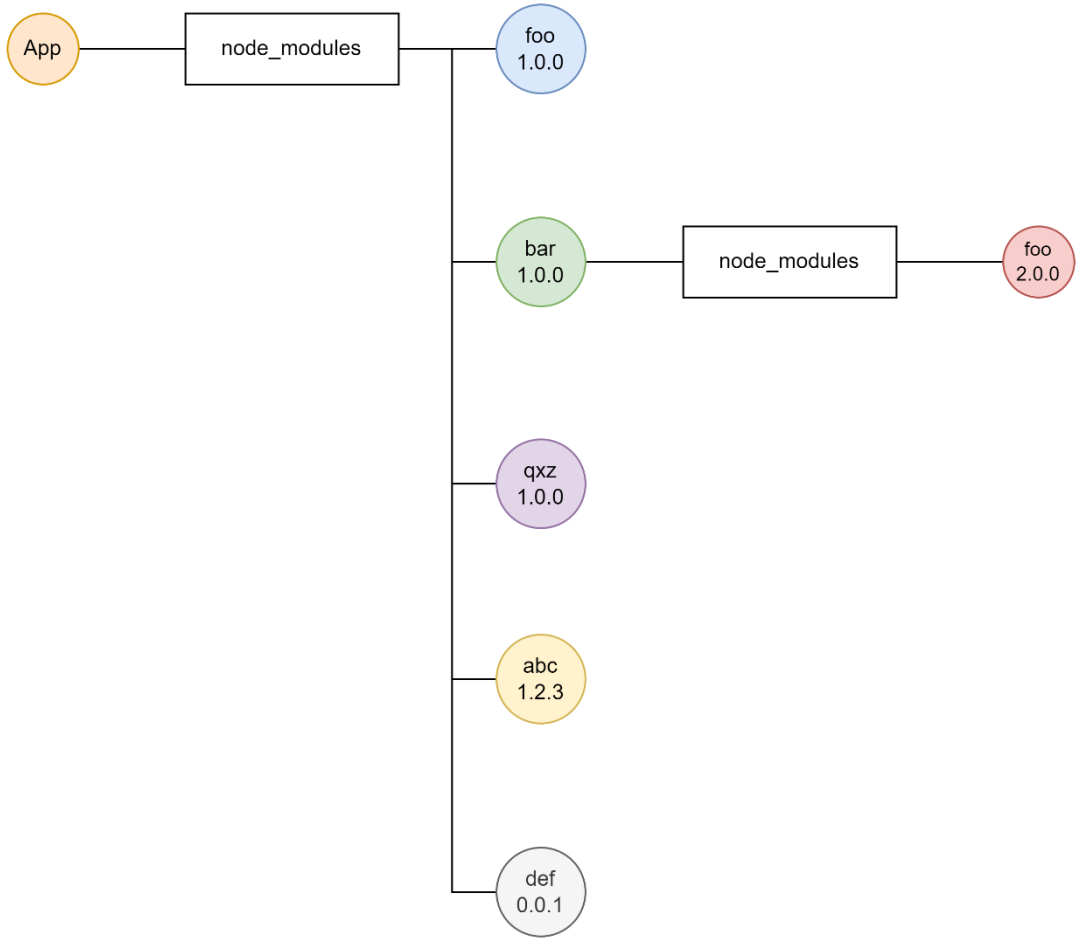
[r.js: Optimizer and Node/Rhino/Nashorn/xpcshell adapter](#)

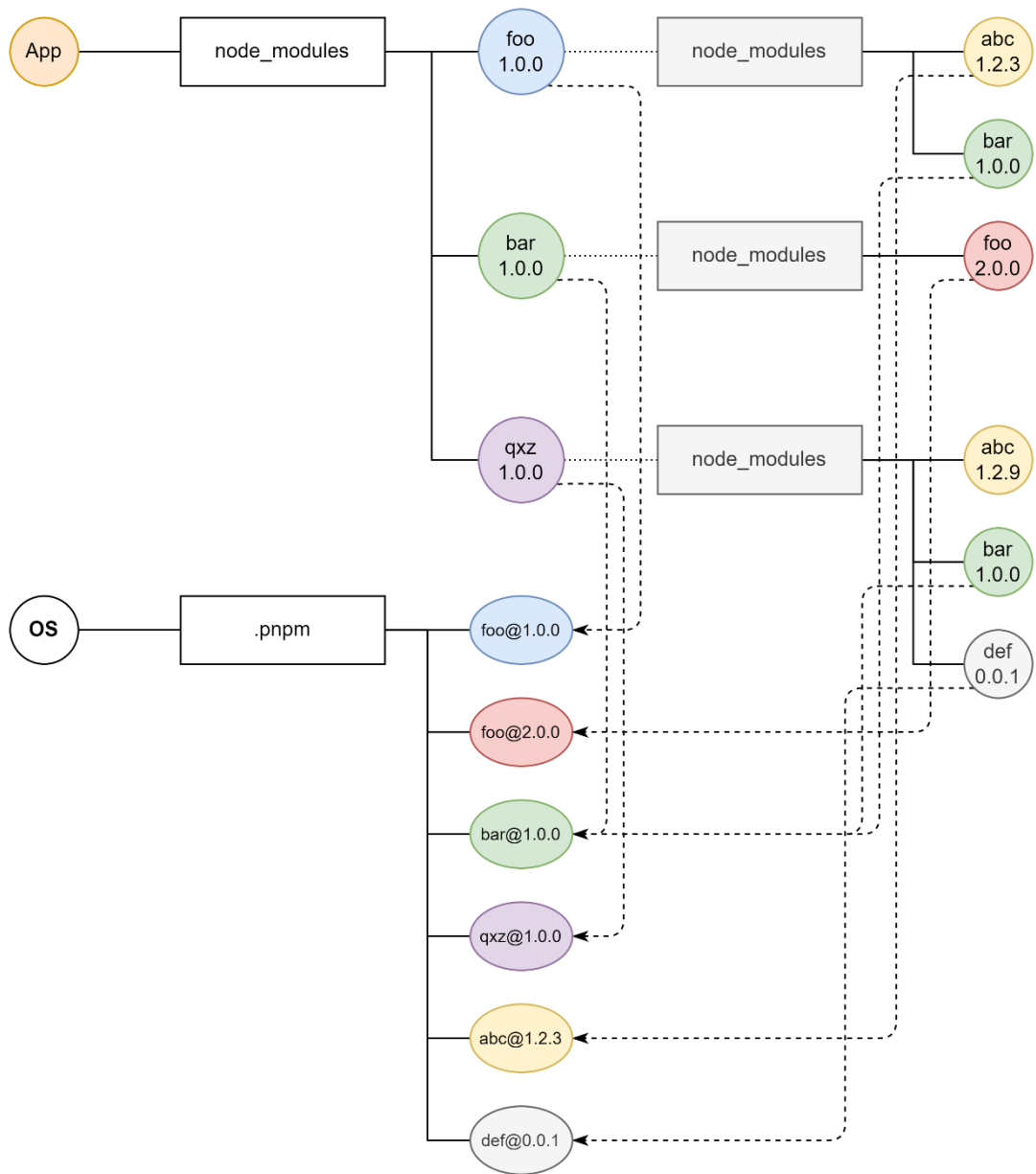
[Download](#)

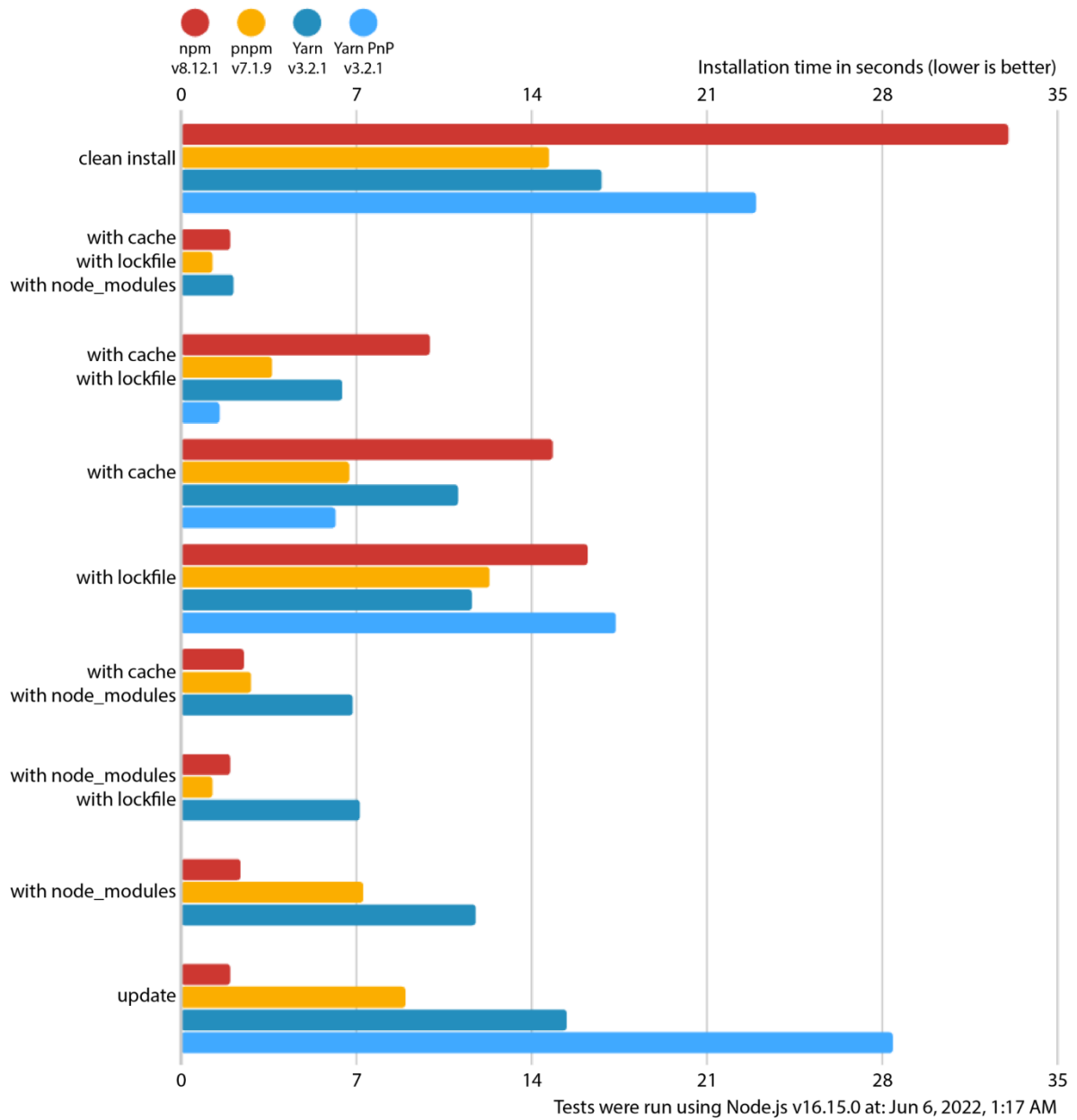
The r.js file allows you to run the optimizer as well as run modules in Node, Rhino, Nashorn or xpcshell.

Chapter 03: Choosing a Package Manager

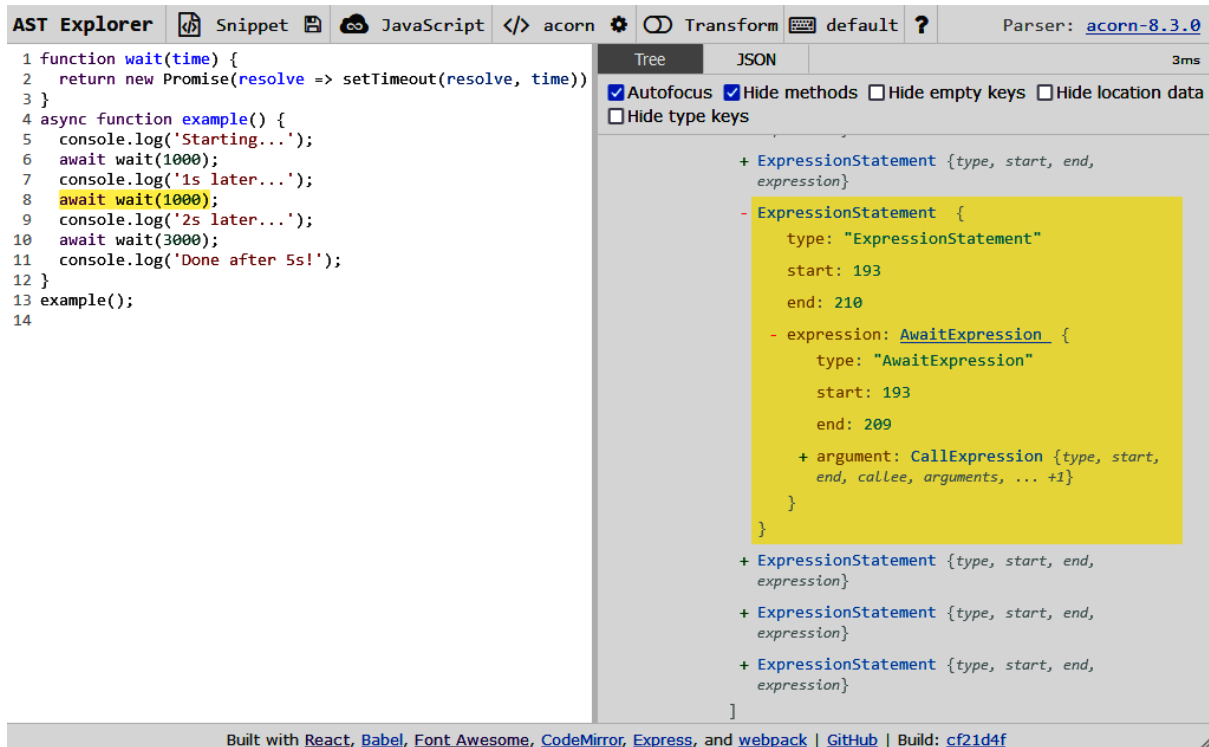
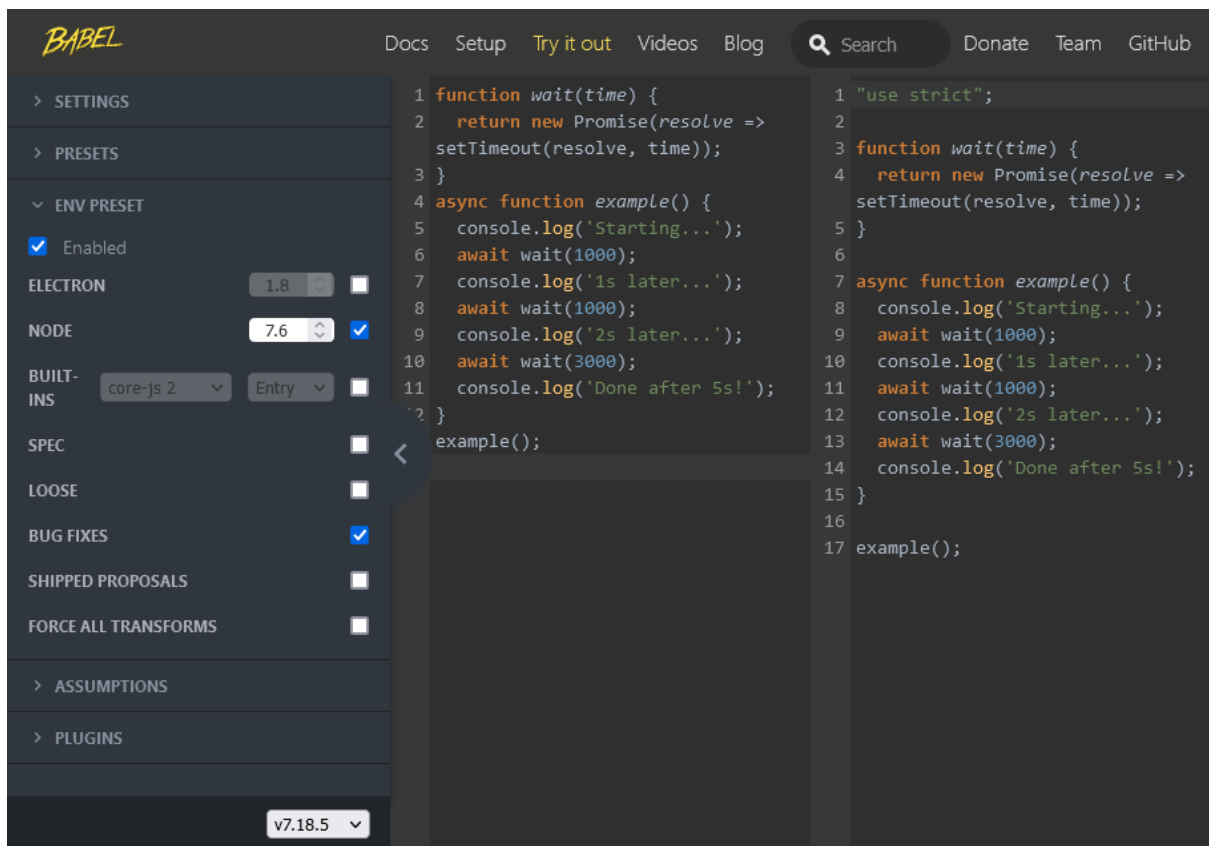
















Chapter 04: Using Different Flavors of JavaScript



Chapter 05: Enhancing Code Quality with Linters and Formatters

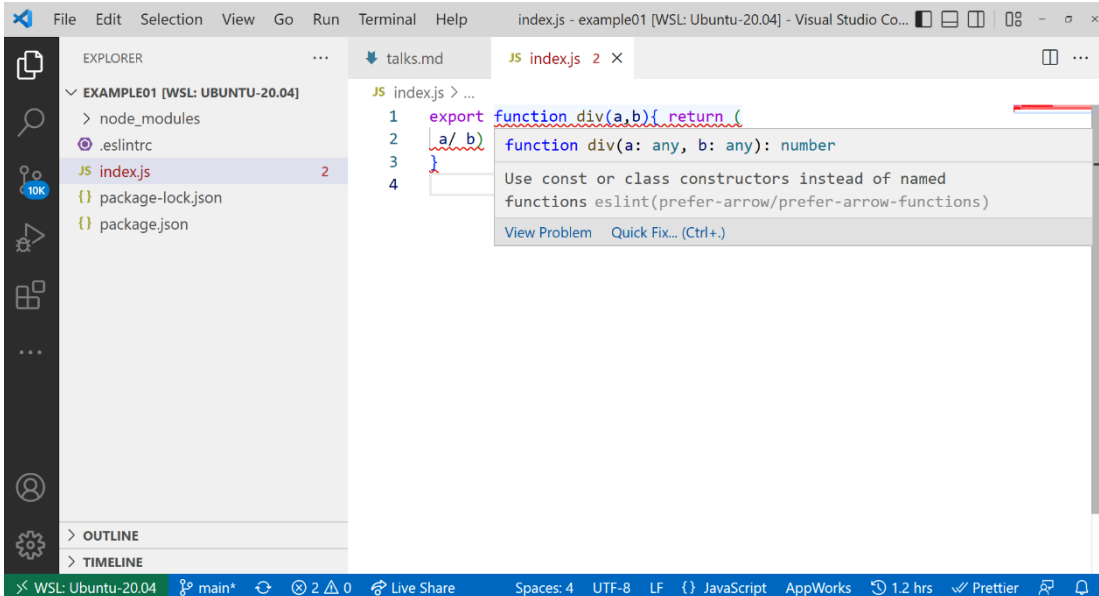


ESLint

Microsoft  |  21,016,085 installs |      (196) | Free

Integrates ESLint JavaScript into VS Code.

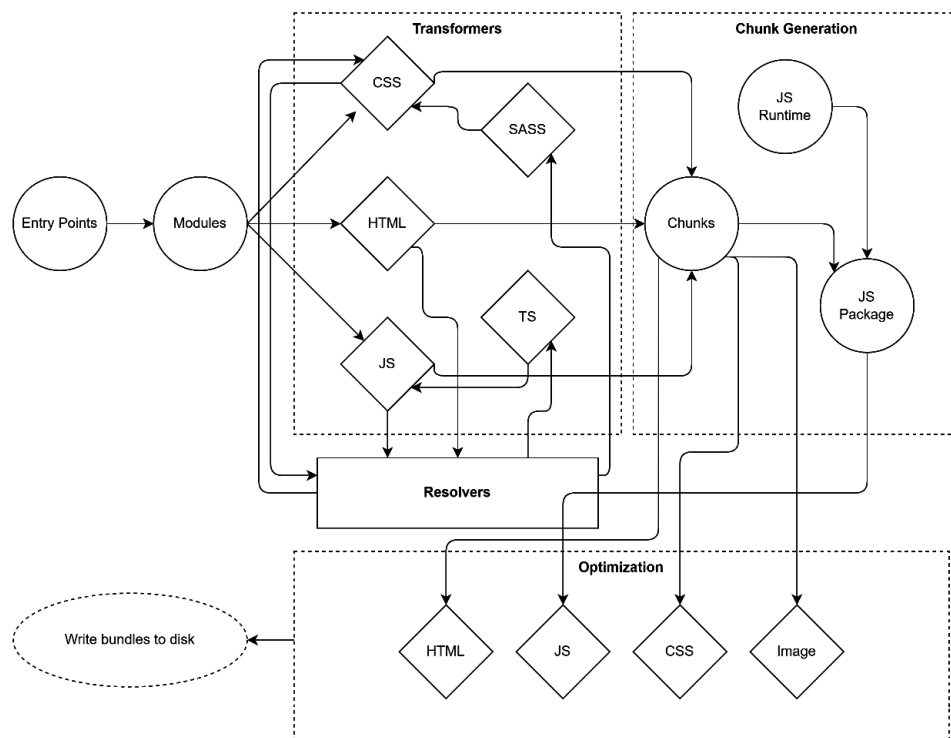
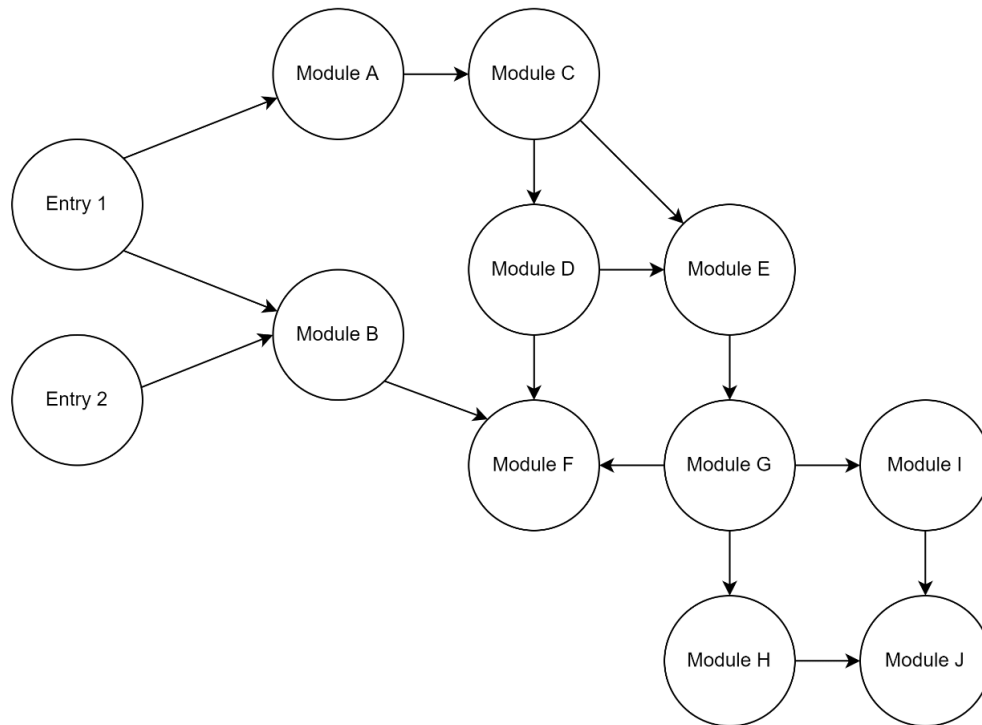
[Install](#) [Trouble Installing?](#)

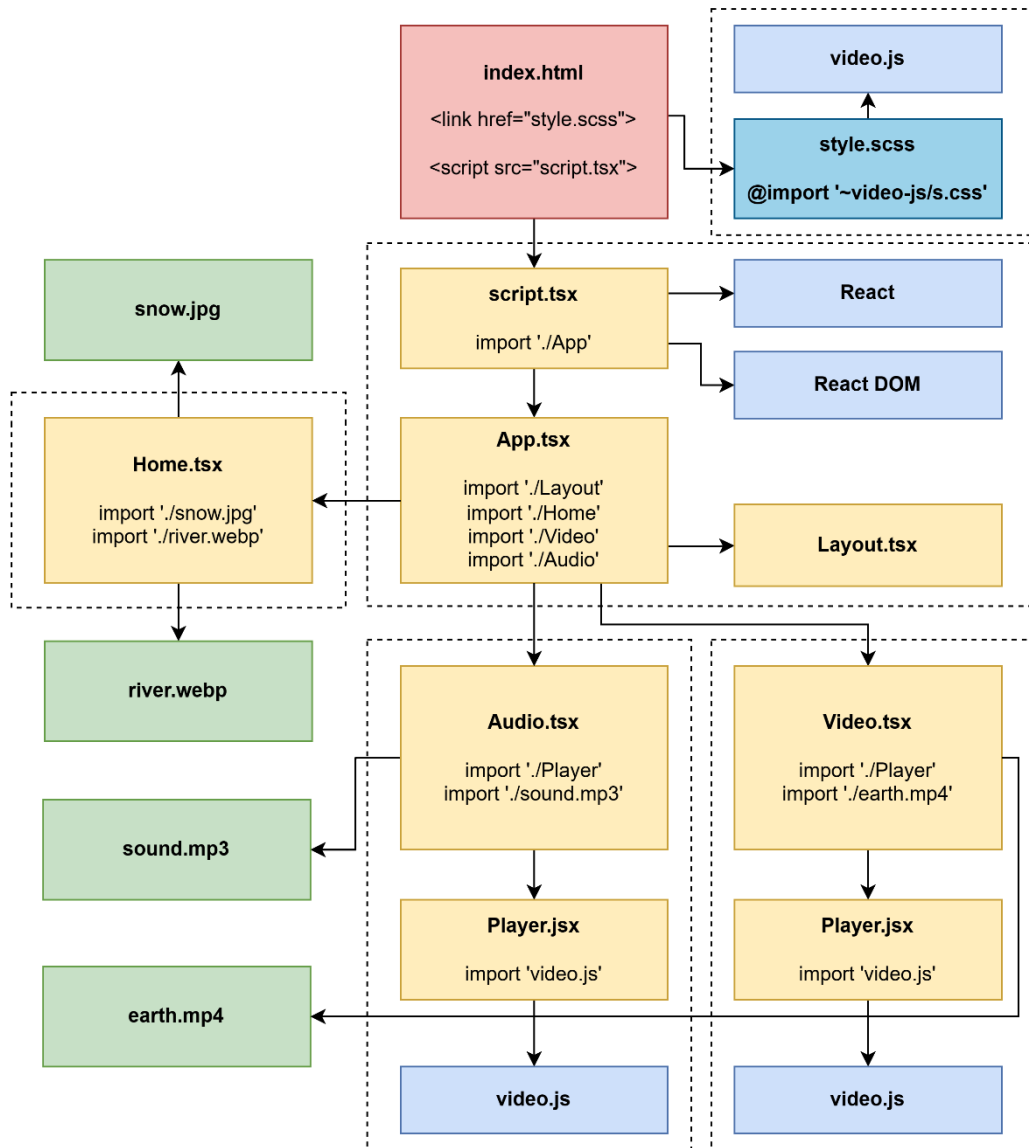


```
1 export function div(a,b){ return (  
2   a/ b)  
3 }  
4
```

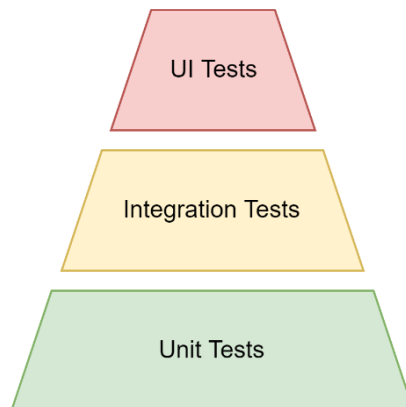
Use const or class constructors instead of named functions
eslint(prefer-arrow/prefer-arrow-functions)
View Problem Quick Fix... (Ctrl+.)

Chapter 06: Building Web Apps with Bundlers








Chapter 07: Improving Reliability with Testing Tools




Cypress

 example05 (main)


 v10.6.0

 Docs ▾

 Log In

Welcome to Cypress!


[Review the differences between each testing type →](#)



E2E Testing

Build and test the entire experience of your application from end-to-end to ensure each flow matches your expectations.

○ Not Configured



Component Testing

Build and test your components from your design system in isolation in order to ensure each state matches your expectations.

○ Not Configured

example05

Not Secure | https://microfrontends.art/_/#/specs/runner?file=cypress/e2e/mf.cy.js

Chrome is being controlled by automated test software.

Specs

mf.cy.js 142ms

empty spec

✓ passes

TEST BODY

```
1 visit https://microfrontends.art
2 title
3 -assert expected The Art of Micro Frontends -
  Florian Rappl to include Micro Frontends
4 get h1
5 -assert expected <h1.sc-breuTD.UQgRF> to have
  text The Art of Micro Frontends
```

<https://microfrontends.art/>

Chrome 104 1000x660 (36%)

Chapter 08: Publishing npm Packages

No Puns Monday

ProductsPricingDocumentation

npm

Search packages

Search

florianrappl

ProfilePackagesAccountBilling InfoAccess TokensAdd OrganizationSign Out

Popular libraries

lodashreacttslibchalkaxioscommanderreact-domexpress

Discover packages

Front-end

Back-endCLI

Documentation

CSSTesting

By

Package2,10

Downloads43,6

Downloads183,

lib-test-florian-rappl

1.0.0 • Public • Published 2 minutes ago

Readme

ExploreBETA

0 Dependencies

0 Dependents

1 Versions

This package does not have a README. Add a README to your package so that users know how to get started.

Keywords

none

Install

> npm i lib-test-florian-rappl

Version

License

1.0.0

MIT

Unpacked Size

Total Files

2.3 kB

3

Last publish

2 minutes ago



Search Packages



LOGIN

No Package Published Yet.

To publish your first package just:

1. Login

`npm adduser --registry http://localhost:4873/`



2. Publish

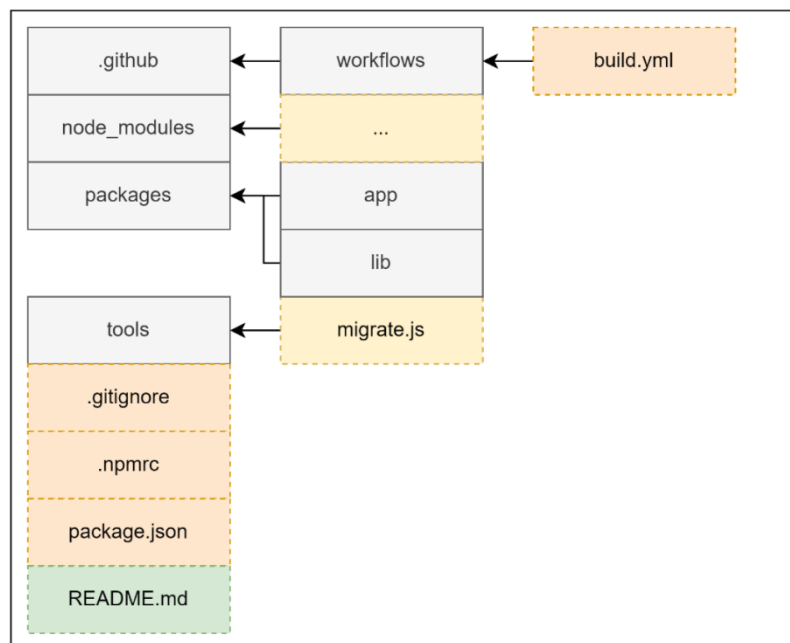
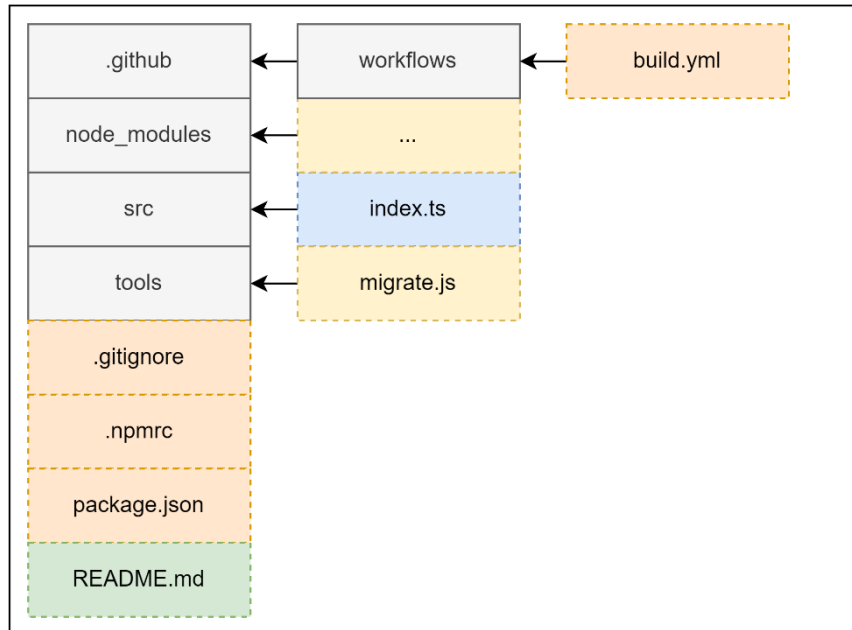
`npm publish --registry http://localhost:4873/`

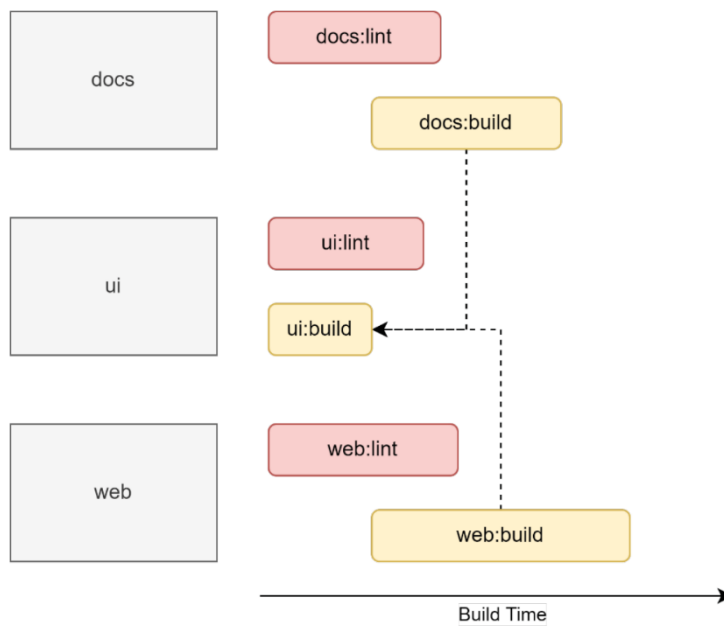
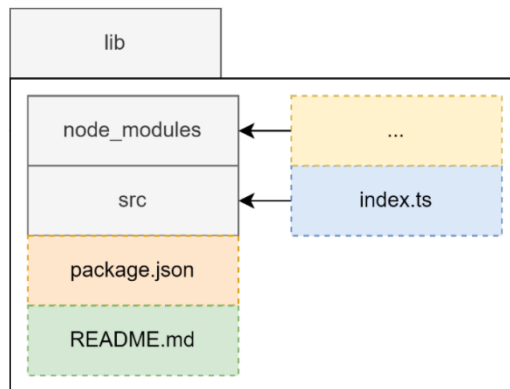
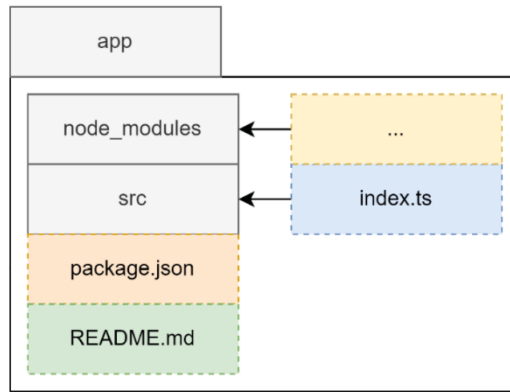


3. Refresh this page

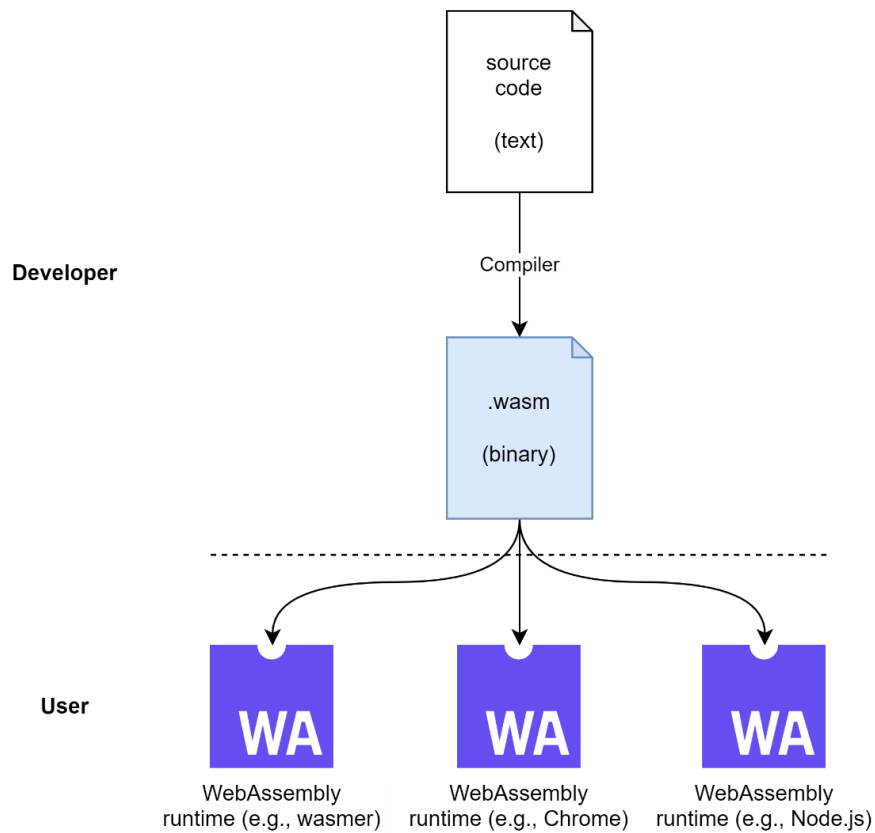
[LEARN MORE](#)

Chapter 09: Structuring Code in Monorepos





Chapter 10: Integrating Native Code with WebAssembly



☐ exceptions ☒ mutable globals ☐ saturating float to int ☐ sign extension

☐ simd ☐ threads ☒ multi value ☐ tail call ☐ bulk memory ☐ reference types

WATexample: simpleDownloadBUILD LOGBASE64

```
1 (module
2   (export "sum" (func $module/sum))
3   (func $module/sum (param $0 i32) (param $1 i32) (result i32)
4     local.get $0
5     local.get $1
6     i32.add
7   )
8 )
9
```

```
0000000: 0061 736d      ; WASM_BINARY_
0000004: 0100 0000      ; WASM_BINARY_
; section "Type" (1)
0000008: 01            ; section code
0000009: 00            ; section size
000000a: 01            ; num types
; func type 0
000000b: 60            ; func
000000c: 02            ; num params
000000d: 7f            ; i32
```

JSJS LOG

```
1 const wasmInstance =
2   new WebAssembly.Instance(wasmModule, {});
3 const { sum } = wasmInstance.exports;
4 for (let i = 0; i < 10; i++) {
5   console.log(sum(i, i));
6 }
7
```

```
0
2
4
6
8
10
12
14
16
18
```

Chapter 11: Using Alternative Runtimes

