

12. Node

Monday, May 22, 2023

5:59 PM

a runtime on machines

// Check version #

node -v

```
$ node -v  
v18.15.0
```

Johnny-Five

Raspberry Pi

terminal:

node

global

process

```
> process  
process {  
  version: 'v18.15.0',  
  versions: {  
    node: '18.15.0',  
    v8: '10.2.154.26-node.25',  
    uv: '1.44.2',  
    zlib: '1.2.13',  
    brotli: '1.0.9',  
    ares: '1.18.1',  
    modules: '108',  
    nghttp2: '1.51.0',  
    napi: '8',  
    llhttp: '6.0.10',  
    uvwasi: '0.0.15',  
    acorn: '8.8.2',  
    simdutf: '3.1.0',
```



```
undici: '5.20.0',
openssl: '3.0.8+quic',
cldr: '42.0',
icu: '72.1',
tz: '2022g',
unicode: '15.0',
ngtcp2: '0.8.1',
nghttp3: '0.7.0'
},
```

```
arch: 'x64',
platform: 'win32',
release: {
  name: 'node',
  lts: 'Hydrogen',
  sourceUrl: 'https://nodejs.org/download/release/v18.15.0/node-v18.15.0.tar.gz',
  headersUrl: 'https://nodejs.org/download/release/v18.15.0/node-v18.15.0-headers.tar.gz',
  libUrl: 'https://nodejs.org/download/release/v18.15.0/win-x64/node.lib'
},
_rawDebug: [Function: _rawDebug],
moduleLoadList: [
  'Internal Binding builtins',
  'Internal Binding errors',
  'Internal Binding util',
  'NativeModule internal/errors',
  'Internal Binding config',
  'Internal Binding timers',
  'Internal Binding async_wrap',
  'Internal Binding constants',
  'Internal Binding types',
  'NativeModule internal/util',
  'NativeModule internal/util/types',
  'NativeModule internal/validators',
  'NativeModule internal/promise_hooks',
```

// Go Back to terminal:
process.exit();

=====

fetch() in Node:

<https://blog.logrocket.com/fetch-api-node-js/>

It was difficult to perform async requests across websites.
until 1998, IE5's XMLHttpRequest that was designed to fetch XML

data via HTTP.

In 2015, Fetch API was launched as a modern successor to XMLHttpRequest.

Fetch uses promises, allowing simpler & cleaner API, avoiding callback hell.

How to use Fetch

The Fetch API is provided as a high-level function, and in its most basic version, it takes a URL and produces a promise that resolves to the response:

```
fetch("http://example.com/api/endpoint")
  .then((response) => {
    // Do something with response
  })
  .catch(function (err) {
    console.log("Unable to fetch -", err);
  });
```

