

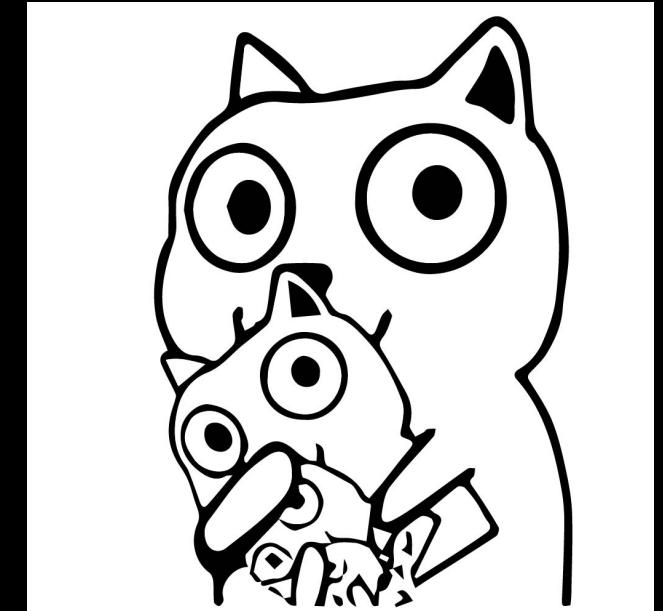
Web APIs in Node.js Core

Past, Present, and Future

Joyee Cheung, Igalia
JSConf EU 2019

About Me

- Joyee Cheung / Qiuyi Zhang (张秋怡)
- Lives in Hangzhou (杭州)
- Compilers team, Igalia
- Node.js TSC
- joyeecheung @ GitHub/Twitter



Web APIs in Node.js core?

Browser APIs as Node.js built-ins

JavaScript in the wild

JS Engines
ECMA TC39

Date
Regexp Error

JavaScript in the wild

Browsers

W3C / WHATWG

JS Engines

ECMA TC39

EventTarget

Date

Regexp Error

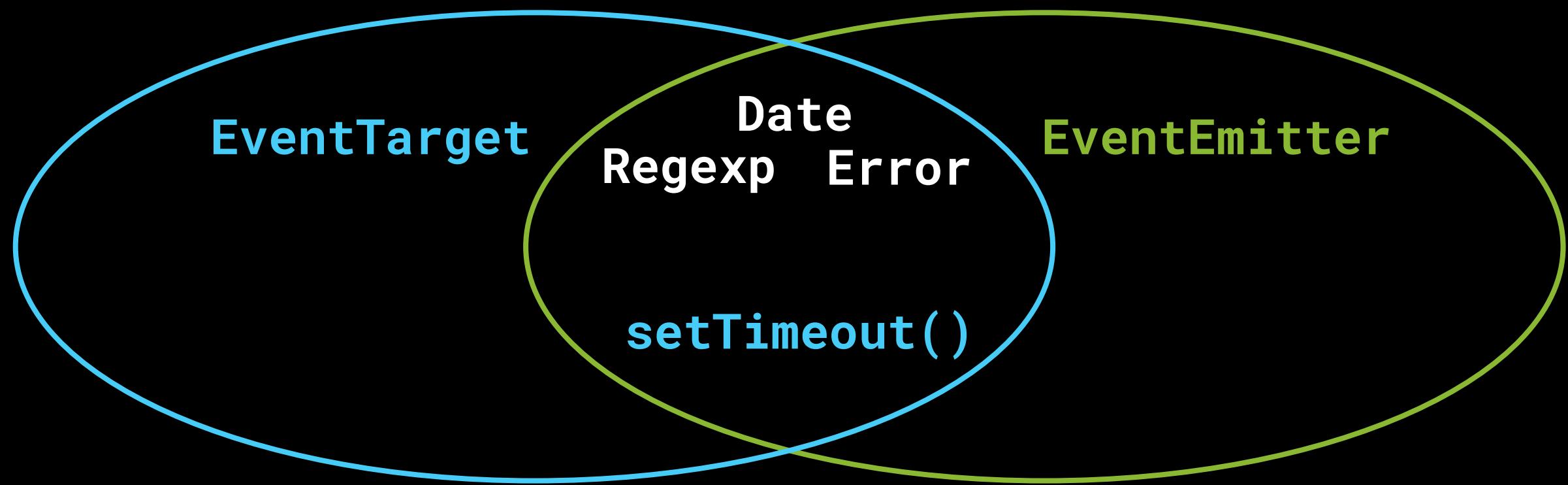
setTimeout()

JavaScript in the wild

Browsers
W3C / WHATWG

JS Engines
ECMA TC39

Node.js Core



JavaScript in the wild

Browsers
W3C / WHATWG

JS Engines
ECMA TC39

Node.js Core

`EventTarget`
`ReadableStream`
`URLSearchParams`
`SubtleCrypto`

`Date`
`Regexp`
`Error`
`setTimeout()`

`EventEmitter`
`Readable`
`querystring`
`crypto`

JavaScript in the wild

Browsers
W3C / WHATWG

JS Engines
ECMA TC39

Node.js Core

`EventTarget`
`ReadableStream`
`SubtleCrypto`

`Date`
`Regexp`
`Error`
`URLSearchParams`
`setTimeout()`

`EventEmitter`
`Readable`
`querystring`
`crypto`

JavaScript APIs on the Web (Browsers)

HTML

Living Standard — Last Updated 23 May 2019



← 8.4 Dynamic markup insertion — [Table of Contents](#) — 8.8 System state and capabilities →

[8.5 Timers](#)

[8.6 Microtask queuing](#)

[8.7 User prompts](#)

[8.7.1 Simple dialogs](#)

[8.7.2 Printing](#)

§ 8.5 Timers

The `setTimeout()` and `setInterval()` methods allow authors to schedule timer-based callbacks.

For web developers (non-normative)

```
handle = self.setTimeout(handler [, timeout [, arguments... ]])
```

Schedules a timeout to run `handler` after `timeout` milliseconds. Any `arguments` are passed straight through to the `handler`.

JavaScript APIs in Node.js

`setTimeout(callback, delay[, ...args])`

#

Added in: v0.0.1

- `callback` `<Function>` The function to call when the timer elapses.
- `delay` `<number>` The number of milliseconds to wait before calling the `callback`.
- `...args` `<any>` Optional arguments to pass when the `callback` is called.
- Returns: `<Timeout>` for use with `clearTimeout()`

Schedules execution of a one-time `callback` after `delay` milliseconds.

The `callback` will likely not be invoked in precisely `delay` milliseconds. Node.js makes no guarantees about the exact timing of when callbacks will fire, nor of their ordering. The callback will be called as close as possible to the time specified.

When `delay` is larger than `2147483647` or less than `1`, the `delay` will be set to `1`. Non-integer delays are truncated to an integer.

If `callback` is not a function, a `TypeError` will be thrown.

This method has a custom variant for promises that is available using `util.promisify()`:

Closer implementations

URL

Living Standard – Last Updated 4 April 2019

Participate:

GitHub whatwg/url (new issue, open issues)

IRC: #whatwg on Freenode

Commits:

[GitHub whatwg/url/commits](https://github.com/whatwg/url/commits)

Snapshot as of this commit

@urlstandard

Tests.

[web-platform-tests url/](#) (ongoing work)

Translations (non-normative):

日本語

Abstract

The URL Shortener API is now available at <https://api.tinyurl.com>

The WHATWG URL API

Class: URL

► History

Browser-compatible `URL` class, implemented by following the W3C specification found in the Standard itself. The `URL` class is also available on the global object.

In accordance with browser conventions, all properties of `URL` objects are on the prototype, rather than as data properties on the object itself. Thus, `Object.prototype` has no `protocol` property, but `URL.prototype.protocol` does. You can't change the value of `URL.prototype.protocol`, but you can add your own `protocol` property to `URL` objects. For example:

```
delete myURL.protocol // returns true
```

Constructor: new URL(input[, base])

- **input** `<string>` The absolute or relative input URL to parse.

And we have tests!

 [web-platform-tests / wpt](#)

 Watch ▾ 313  Star 2,278  Fork 1,680

 Code  Issues 923  Pull requests 504  Projects 3  Wiki  Security  Insights

Test suites for Web platform specs — including WHATWG, W3C, and others <http://irc.w3.org/?channels=testing>

testing html test-automation browser w3c whatwg test-runner web-standards javascript dom web-development
google-chrome blink gecko firefox microsoft-edge safari webkit opera

 35,757 commits  335 branches  8,269 releases  1,068 contributors  View license

Branch: master ▾ [New pull request](#) [Create new file](#) [Upload files](#) [Find File](#) [Clone or download ▾](#)

 **annevk** and **foolip** Editorial: cleanup some area opener/noopener tests ([#17025](#)) ... Latest commit 36f2a92 an hour ago

 **.github** [docs] Build w/GH Actions and deploy w/GH Pages 13 days ago

 **.well-known** Part 1: Stop using |global|, and fix a typo ('param' should have been...) a month ago

 **2dcontext** Enable preload=metadata lazy loading by default on Win, Android, Linux. 6 days ago

WPT conformance in Node.js

```
~/projects/node ➤ ↵ master ➤ ls test/wpt/status/**/*.json
-rw-r--r-- 1 joyee staff 100B Nov 14 2018 test/wpt/status/console.json
-rw-r--r-- 1 joyee staff 1.7K Jun 1 00:42 test/wpt/status/encoding.json
-rw-r--r-- 1 joyee staff 208B Jan 24 17:13 test/wpt/status/html/webappapis/microtask-
-rw-r--r-- 1 joyee staff 3B Feb 17 06:47 test/wpt/status/html/webappapis/timers.json
-rw-r--r-- 1 joyee staff 324B Jan 10 12:54 test/wpt/status/url.json
~/projects/node ➤ ↵ master ➤ cat test/wpt/status/url.json
{
  "toascii.window.js": {
    "requires": ["small-icu"],
    "skip": "TODO: port from .window.js"
  },
  "historical.any.js": {
    "requires": ["small-icu"]
  },
  "urlencoded-parser.any.js": {
    "fail": "missing Request and Response"
  },
}
```

Implemented in Node.js first (experimental)

<https://html.spec.whatwg.org/multipage/webappapis.html#creating-a-json-module-script>

To **create a JSON module script**, given a string *source* and an [environment settings object](#) *settings*:

1. Let *script* be a new [module script](#) that this algorithm will subsequently initialize.
2. Set *script*'s [settings object](#) to *settings*.
3. Set *script*'s [base URL](#) and [fetch options](#) to null.
4. Set *script*'s [parse error](#) and [error to rethrow](#) to null.
5. Let *json* be ? [Call\(%JSONParse%, undefined, « source »\).](#)

If this throws an exception, set *script*'s [parse error](#) to that exception, and return *script*.

6. Set *script*'s [record](#) to the result of [creating a synthetic module record with a default export](#) of *json* with *settings*.
7. Return *script*.

Alternatives in Node.js Core

Web

`URL/URLSearchParams`

`Text{En,De}coder`

`performance`

Node.js

`url/querystring`

`Buffer/StringDecoder`

`Timing methods in process`

Alternatives in Node.js Core

Web

`URL/URLSearchParams`

`Text{En,De}coder`

`performance`

Be careful...

`queueMicrotask()`

Node.js

`url/querystring`

`Buffer/StringDecoder`

`Timing methods in process`

`process.nextTick()`

Alternatives in Node.js Core

Web

`URL/URLSearchParams`

`Text{En,De}coder`

`performance`

Be careful...

`queueMicrotask()`

`Worker`

Node.js

`url/querystring`

`Buffer/StringDecoder`

`Timing methods in process`

`process.nextTick()`

`child_process`

Web APIs in Node.js v12

Stable

`console`

`queueMicrotask()`

`URL/URLSearchParams`

`TextEncoder/TextDecoder`

Timers e.g. `setTimeout()`

Globals

Covered by WPT with some expected failures

Web APIs in Node.js v12

Stable

`console`

`queueMicrotask()`

`URL/URLSearchParams`

`TextEncoder/TextDecoder`

`Timers e.g. setTimeout()`

Experimental

`worker_thread.Worker`

`perf_hooks.performance`

Under builtin modules

Not global

Differs significantly

No WPT coverage yet

Web APIs in Node.js v12

Stable

`console`

`queueMicrotask()`

`URL/URLSearchParams`

`TextEncoder/TextDecoder`

Timers e.g. `setTimeout()`

Experimental

`worker_thread.Worker`

`perf_hooks.performance`

Implemented by V8

WebAssembly

Web APIs haunting the issue tracker...

~~Web APIs haunting the issue tracker...~~

Web APIs under discussion

Web APIs under discussion

- **Stream**
 - <https://github.com/nodejs/whatwg-stream>

Web APIs under discussion

- Stream
 - <https://github.com/nodejs/whatwg-stream>

Stream 1

Push

Web APIs under discussion

- Stream
 - <https://github.com/nodejs/whatwg-stream>

Stream 1

Stream 2

Push

Push/Pull

Web APIs under discussion

- **Stream**
 - <https://github.com/nodejs/whatwg-stream>

Stream 1

Stream 2

Stream 3

Push

Push/Pull

Push+Pull

Web APIs under discussion

- Stream
 - <https://github.com/nodejs/whatwg-stream>

Stream 1

Stream 2

Stream 3

bob (dev)

Push

Push/Pull

Push+Pull

Pull

Web APIs under discussion

- **Stream**
 - <https://github.com/nodejs/whatwg-stream>

Stream 1

Stream 2

Stream 3

bob (dev)

WHAWG?

Push

Push/Pull

Push+Pull

Pull

Web APIs under discussion

- **Stream**
 - <https://github.com/nodejs/whatwg-stream>
- **Fetch**
 - <https://github.com/nodejs/summit/issues/165>
 - New! <https://github.com/nodejs/node/pull/27979>

Web APIs in Node.js Core!

...But why?

Web APIs in Node.js Core!

fetch api

All

Videos

Images

Books

About 73.500.000 results (0,31 seconds)

[Fetch API - Web APIs | MDN](#)

https://developer.mozilla.org/en/docs/Web/API/Fetch_API

Apr 4, 2019 - The **Fetch API** provides an interface for making requests to the network. It will seem familiar to anyone who has used XMLHttpRequest.

Node.js http.request

All

Videos

Images

Maps

About 49.400.000 results (0,28 seconds)

[5 Ways to Make HTTP Requests](#)

<https://www.twilio.com/blog/2017/08/http-requests-nodejs/>

Aug 9, 2017 - How to make **HTTP requests** and **responses** in **Node.js**. When it comes to **Node.js** there are a fair amount of

Web APIs in Node.js Core!



MDN web docs
Technologies ▾ References & Guides ▾ Feedback ▾

Web APIs

When writing code for the Web, there are a great many Web APIs available. Below is a list of all the APIs and interfaces (object types) that you may be able to use while developing your Web app or site.

Web APIs are typically used with JavaScript, although this doesn't always have to be the case.

Specifications

This is a list of all the APIs that are available.

Node.js v12.3.1 Documentation

[Index](#) | [View on single page](#) | [View as JSON](#) | [View another version ▾](#) | [Edit on GitHub](#)

Table of Contents

- [About these Docs](#)
- [Usage & Example](#)
- [API Testing](#)
- [Books](#)
- [Bugs](#)
- [Changelog](#)
- [Child Processes](#)
- [Cluster](#)
- [Command Line Options](#)
- [Console](#)
- [Crypto](#)
- [Debugger](#)
- [Deprecation API](#)

Web APIs in Node.js Core!

The WHATWG URL API

Class: URL

► History

Browser-compatible `URL` class, implemented by following the WHATWG URL Standard. Examples of parsed URLs may be found in the Standard itself. The `URL` class is also available on the `global` object.

In accordance with browser conventions, all properties of the `URL` class are implemented as getters and setters on the class prototype, rather than as data properties on the objects themselves. Deleting legacy `urlObject`s, using the `delete` keyword on any properties of `URL` objects (e.g. `delete myURL.pathname`) (e.g. `delete myURL.pathname`, etc) has no effect but will still return `true`.



Constructor: `new URL(input[, base])`

- `input <string>` The absolute or relative input URL to parse. If `input` is relative, then `base` is required. If `input` is

Web APIs in Node.js Core!

Maintenance and longevity

Node.js Project Governance

- Collaborators
 - Collaborator Activities
- Technical Steering Committee
 - TSC Meetings
- Collaborator Nominations
 - Onboarding
- Consensus Seeking Process

Collaborators

Node.js Core Collaborators maintain the [nodejs/node](#) GitHub repository. The GitHub team for Node.js Core Collaborators is @nodejs/collaborators. Their privileges include but are not limited to:

- Commit access to the [nodejs/node](#) repository

Web APIs in Node.js Core?

Web APIs in Node.js Core?

Consensus Seeking

<https://github.com/nodejs/TSC/blob/master/TSC-Charter.md>

Section 8. Voting

For internal project decisions, Collaborators shall operate under Lazy Consensus. The TSC shall establish appropriate guidelines for implementing Lazy Consensus (e.g. expected notification and review time periods) within the development process.

The TSC follows a [Consensus Seeking](#) decision making model. When an agenda item has appeared to reach a consensus the moderator will ask "Does anyone object?" as a final call for dissent from the consensus.

If an agenda item cannot reach a consensus a TSC member can call for either a closing vote or a vote to table the issue to the next meeting. The call for a vote must be seconded by a majority of the TSC or else the discussion will continue.

For all votes, a simple majority of all TSC members for, or against, the issue wins. A TSC member may choose to participate in any vote through abstention.

Web APIs in Node.js Core?

Small core philosophy

Web APIs in Node.js Core?

Small core philosophy

- Duplicate API for a similar set of functionality
 - While missing features that do not make sense for browsers

Web APIs in Node.js Core?

Small core philosophy

- Duplicate API for a similar set of functionality
 - While missing features that do not make sense for browsers
- In core or as separate “official” modules?
 - Optimizability
 - Dependency on internals

Web APIs in Node.js Core?

More than just a bunch of interfaces

Web APIs in Node.js Core?

More than just a bunch of interfaces

`https://foo.com/script.js`

Origin:

`https://foo.com`

Cookie: TOKEN=XXX

Browsers

Access-Control-Allow-Origin:
`https://foo.com`

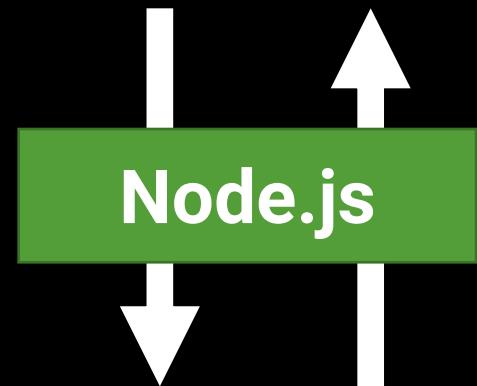
`https://bar.com/api`

Web APIs in Node.js Core?

More than just a bunch of interfaces

```
./node_modules/foo/script.js
```

Origin:
<https://foo.com>
Cookie: TOKEN=XXX



We don't really have an origin ATM..
~~Access-Control-Allow-Origin: https://foo.com~~

```
https://bar.com/api
```

Web APIs in Node.js Core?



`fetch()`

Content Security Policy

CORS

Caching

Service workers

Credentials e.g. Cookies

Web APIs in Node.js Core?

User land migration

- Different vocabulary

`fetch()`

`http.request()`

WHATWG
Streams

`EventTarget`

`require
('stream')`

`EventEmitter`

Web APIs in Node.js Core?

User land migration

- Different vocabulary
- Interoperability with other modules

`fetch()`

`http.request()`

WHATWG
Streams

`EventTarget`

`require
('stream')`

`EventEmitter`

Web APIs in Node.js Core?

Open questions

+

consensus seeking

Typical Workflow



Typical Workflow



Typical Workflow



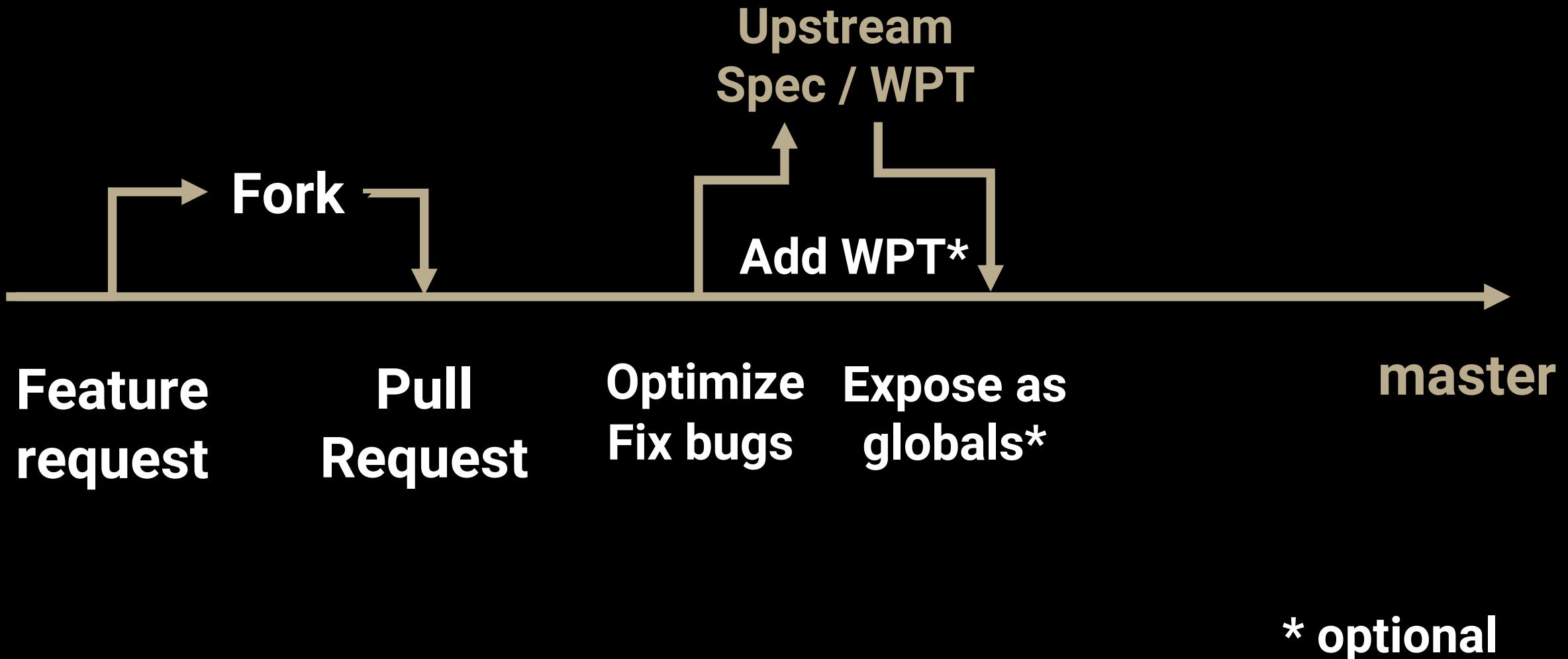
e.g. WebCrypto

<https://github.com/nodejs/node/issues/19826>

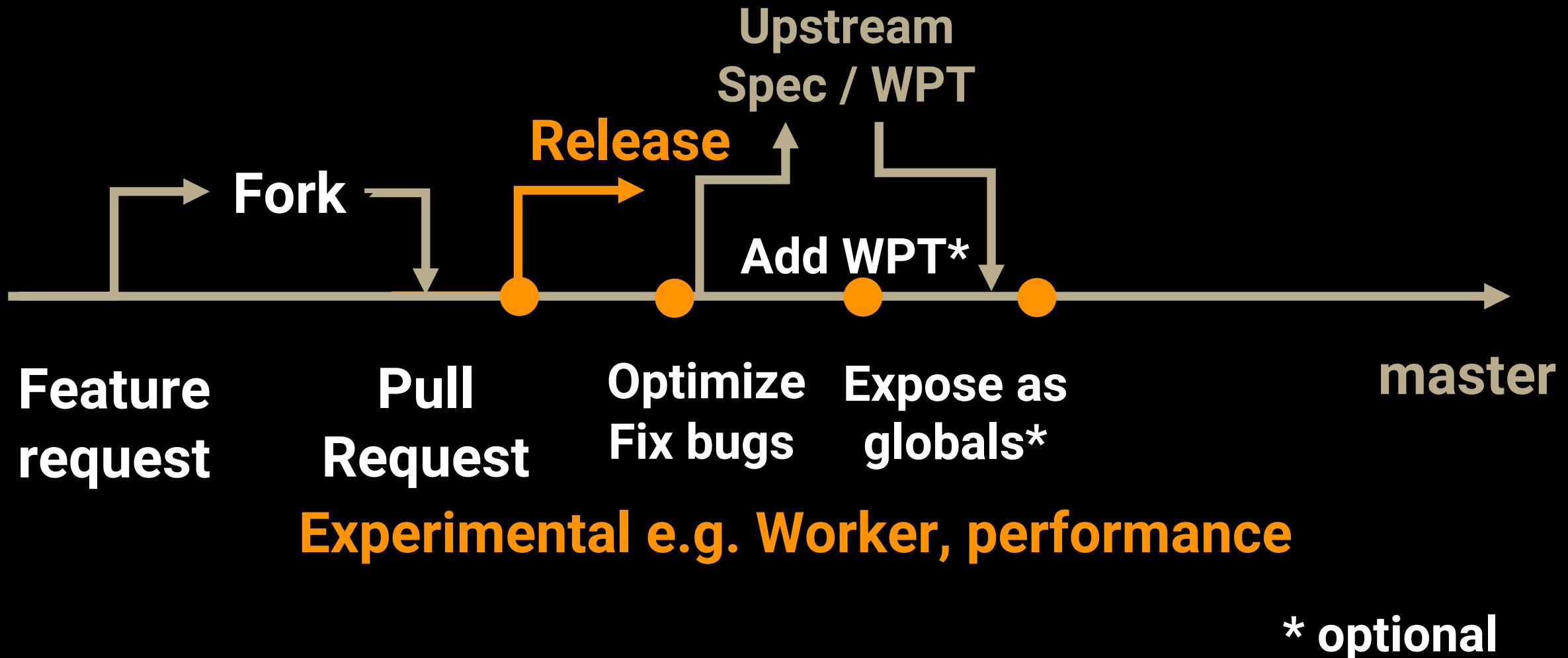
Typical Workflow



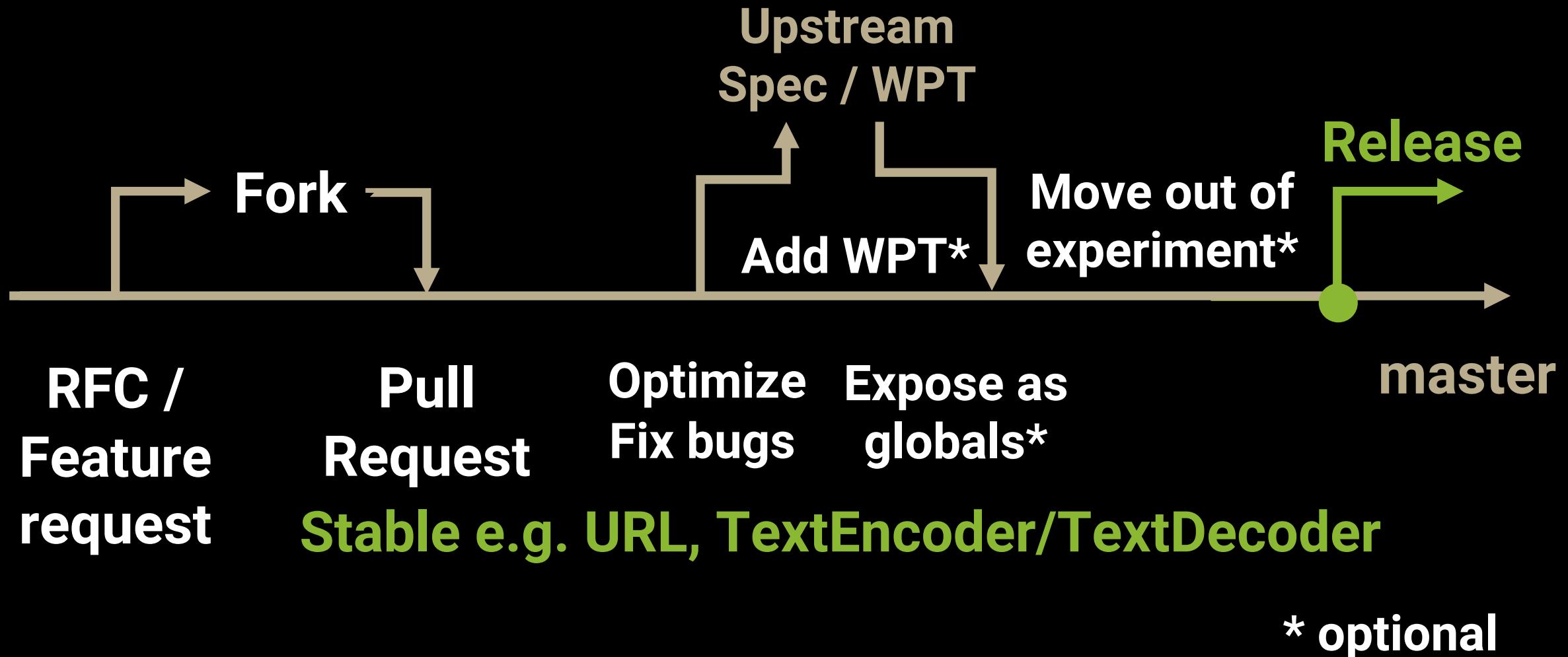
Typical Workflow



Typical Workflow



Typical Workflow



Summary

- **Past**
 - Node.js core APIs diverged from Web APIs for different design goals
 - Node.js has added more and more Web APIs
- **Present**
 - console, encoding, timer, microtask queuing, URL, Workers, performance, WebAssembly
- **Future**
 - Challenges and workflow

Node.js & Open Standards

nodejs / open-standards

Unwatch 41 Star 43 Fork 2

Code Issues 12 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Filters ▾ is:issue is:open Labels 9 Milestones 0 New issue

12 Open ✓ 1 Closed Author ▾ Labels ▾ Projects ▾ Milestones ▾ Assignee ▾ Sort ▾

Issue	Author	Labels	Projects	Milestones	Assignee	Sort
Idea: Open Standards Session at Collaborator Summit in May #18 opened on Apr 2 by jorydotcom	jorydotcom					11
Hunting for ECMAScript proposals that may be interesting to Node #16 opened on Jan 7 by joyeecheung	joyeecheung					3
Import Maps review #13 opened on Dec 4, 2018 by littledan	littledan					7
WHATWG Streams discussion #12 opened on Oct 30, 2018 by mcollina	mcollina					12
Process for shipping open standards #11 opened on Oct 29, 2018 by mcollina	mcollina					13

Thank you!