



Standard **Standard** Ecma-XXX

1st Edition / 31 May 2025

Minimum Common Web Platform API

Standard

Ecma International
Rue du Rhone 114 CH-1204 Geneva
Tel: +41 22 849 6000
Fax: +41 22 849 6001
Web: <https://www.ecma-international.org>



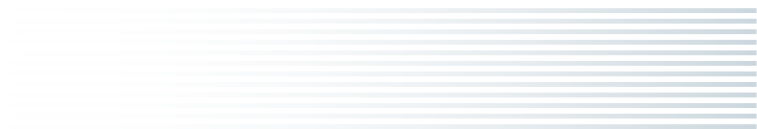
is the registered trademark of Ecma International



COPYRIGHT PROTECTED DOCUMENT

| Table of Contents | page |
|---|-------------|
| 1 Terminology | 4 |
| 2 Common API Index | 4 |
| 3 The Global Scope | 6 |
| 4 Requirements for navigator.userAgent | 6 |
| 5 Extensions | 7 |
| References | 7 |
| Normative References | 7 |
| Index | 8 |
| Terms defined by this specification | 8 |
| Terms defined by reference | 8 |





Introduction

The Minimum Common Web Platform API is a curated subset of standardized Web Platform APIs intended to define a minimum set of capabilities common to Browser and Non-Browser JavaScript-based runtime environments.

Contributing to this Specification

This version:

<https://min-common-api.proposal.wintercg.org/>

Issue Tracking:

[GitHub](#)

Editor:

[James M Snell](#) (Cloudflare)

1. Terminology

The Web Platform is the combination of technology standards defined by organizations such as the W3C, the WHATWG, and others as implemented by Web Browsers.

A *Web-interoperable Runtime* is any ECMAScript-based application runtime environment that implements the subset of Web Platform APIs outlined in this specification. While this term is intentionally broad to also encompass Web Browsers, the primary focus here is on outlining expectations for non-browser runtimes.

2. Common API Index

All [Web-interoperable Runtimes](#) conforming to this specification SHALL implement each of the following Web Platform APIs in accordance with their normative requirements except where modified here. Where any conforming runtime environment chooses (either by necessity or otherwise) to diverge from a normative requirement of the specification, clear explanations of such divergence MUST be made clearly and readily available in the documentation.

Interfaces:

- [AbortController](#)
- [AbortSignal](#)
- [Blob](#)
- [ByteLengthQueuingStrategy](#)
- [CompressionStream](#)
- [CountQueuingStrategy](#)
- [Crypto](#)
- [CryptoKey](#)
- [DecompressionStream](#)
- [DOMException](#)
- [Event](#)
- [EventTarget](#)
- [File](#)

- `FormData`

The `FormData` constructor optionally takes `HTMLFormElement` and `HTMLElement` as parameters. TODO: Figure out what implementations without DOM support should do here. Node.js and Deno throw if the first parameter is not `undefined` but ignore the second parameter. Cloudflare Workers ignores all parameters.

- `Headers`

- `ReadableByteStreamController`

- `ReadableStream`

- `ReadableStreamBYOBReader`

- `ReadableStreamBYOBRequest`

- `ReadableStreamDefaultController`

- `ReadableStreamDefaultReader`

- `Request`

- `Response`

- `SubtleCrypto`

- `TextDecoder`

- `TextDecoderStream`

- `TextEncoder`

- `TextEncoderStream`

- `TransformStream`

- `TransformStreamDefaultController`

- `URL`

- `URLPattern`

- `URLSearchParams`

- `WebAssembly.Global`

- `WebAssembly.Instance`

- `WebAssembly.Memory`

- `WebAssembly.Module`

- `WebAssembly.Table`

- `WritableStream`

- `WritableStreamDefaultController`

Global methods / properties:

- `globalThis`
- `globalThis.atob()`
- `globalThis.btoa()`
- `globalThis.console`
- `globalThis.crypto`
- `globalThis.fetch()`
- `globalThis.navigator.userAgent`
- `globalThis.performance.now()`
- `globalThis.performance.timeOrigin`
- `globalThis.queueMicrotask()`
- `globalThis.setTimeout()` / `globalThis.clearTimeout()`
- `globalThis.setInterval()` / `globalThis.clearInterval()`
- `globalThis.structuredClone()`
- `globalThis.WebAssembly.compile()`
- `globalThis.WebAssembly.compileStreaming()`
- `globalThis.WebAssembly.instantiate()`
- `globalThis.WebAssembly.instantiateStreaming()`
- `globalThis.WebAssembly.validate()`

3. The Global Scope

The exact type of the global scope (`globalThis`) can vary across runtimes. Most Web Platform APIs are defined in terms that assume Web Browser environments that specifically expose types like `Window`, `Worker`, `WorkerGlobalScope`, and so forth. To simplify conformance, all Interfaces, methods, and properties defined by this specification MUST be exposed on the runtime's relevant global scope (e.g., `globalThis.crypto`, `globalThis.ReadableStream`, etc).

With many runtimes, adding a new global-scoped property can introduce breaking changes when the new global conflicts with existing application code. Many Web Platform APIs define global properties using the `readonly` attribute. To avoid introducing breaking changes, runtimes conforming to this specification MAY choose to ignore the `readonly` attribute for properties being added to the global scope.

4. Requirements for `navigator.userAgent`

The `globalThis.navigator.userAgent` property is provided such that application code can reliably identify the runtime within which it is running. The value of the property is a string conforming to the `User-Agent` construction in RFC 7231:


```
User-Agent      = product *( RWS ( product / comment ) )
product        = token [ "/" product-version ]
product-version = token
```

While runtimes that implement `globalThis.navigator.userAgent` MUST provide a value that is conformant with the structure defined by RFC 7231, the value SHOULD be treated as a single, complete, opaque, unstructured value. It is RECOMMENDED that the value be limited to a single **product** token excluding the optional **product-version**. For instance, `navigator.userAgent = 'MyRuntime'`. The value SHOULD NOT include any **comment** components.

5. Extensions

Runtime-specific extensions to any Web Platform API MAY be implemented by conforming runtimes. Such extensions MUST be defined so that their use neither contradicts nor causes the non-conformance of normative functionality of any Web Platform API.

Application use of such extensions must be carefully considered, as doing so reduces interoperability and portability of code across runtimes.

References

Normative References

[CONSOLE]

Dominic Farolino; Robert Kowalski; Terin Stock. *Console Standard*. Living Standard. URL: <https://console.spec.whatwg.org/>

[DOM]

Anne van Kesteren. *DOM Standard*. Living Standard. URL: <https://dom.spec.whatwg.org/>

[ENCODING]

Anne van Kesteren. *Encoding Standard*. Living Standard. URL: <https://encoding.spec.whatwg.org/>

[FETCH]

Anne van Kesteren. *Fetch Standard*. Living Standard. URL: <https://fetch.spec.whatwg.org/>

[FileAPI]

Marijn Kruisselbrink. *File API*. URL: <https://w3c.github.io/FileAPI/>

[HR-TIME-3]

Yoav Weiss. *High Resolution Time*. URL: <https://w3c.github.io/hr-time/>

[HTML]

Anne van Kesteren; et al. *HTML Standard*. Living Standard. URL: <https://html.spec.whatwg.org/multipage/>

[STREAMS]

Adam Rice; et al. *Streams Standard*. Living Standard. URL: <https://streams.spec.whatwg.org/>

[URL]

Anne van Kesteren. *URL Standard*. Living Standard. URL: <https://url.spec.whatwg.org/>

[URLPATTERN]

Ben Kelly; Jeremy Roman; 宍戸俊哉 (Shunya Shishido). *URL Pattern Standard*. Living Standard. URL: <https://urlpattern.spec.whatwg.org/>

[WASM-JS-API-2]

. Ms2ger. *WebAssembly JavaScript Interface*. URL: <https://webassembly.github.io/spec/js-api/>

[WASM-WEB-API-2]

. Ms2ger. *WebAssembly Web API*. URL: <https://webassembly.github.io/spec/web-api/>

[WEBCRYPTO-2]

Daniel Huigens. *Web Cryptography Level 2*. URL: <https://w3c.github.io/webcrypto/>

[WEBIDL]

Edgar Chen; Timothy Gu. *Web IDL Standard*. Living Standard. URL: <https://webidl.spec.whatwg.org/>

[XHR]

Anne van Kesteren. *XMLHttpRequest Standard*. Living Standard. URL: <https://xhr.spec.whatwg.org/>

Index

Terms defined by this specification

- [Web-interoperable Runtime](#), in § 1

Terms defined by reference

- [\[CONSOLE\]](#) defines the following terms:
 - console
- [\[DOM\]](#) defines the following terms:
 - AbortController
 - AbortSignal
 - Event
 - EventTarget
- [\[ENCODING\]](#) defines the following terms:
 - TextDecoder
 - TextDecoderStream
 - TextEncoder
 - TextEncoderStream
- [\[FETCH\]](#) defines the following terms:
 - Headers
 - Request
 - Response
 - fetch(input)
- [\[FileAPI\]](#) defines the following terms:
 - Blob
 - File
- [\[HR-TIME-3\]](#) defines the following terms:
 - now()
 - performance
 - timeOrigin
- [\[HTML\]](#) defines the following terms:
 - HTMLElement
 - HTMLFormElement
 - Window
 - Worker
 - WorkerGlobalScope
 - atob(data)
 - btoa(data)
 - clearInterval(id)
 - clearTimeout(id)
 - navigator
 - queueMicrotask(callback)
 - setInterval(handler, timeout, ...arguments)
 - setTimeout(handler, timeout, ...arguments)

- structuredClone(value, options)
- userAgent
- [STREAMS] defines the following terms:
 - ByteLengthQueuingStrategy
 - CountQueuingStrategy
 - ReadableByteStreamController
 - ReadableStream
 - ReadableStreamBYOBReader
 - ReadableStreamBYOBRequest
 - ReadableStreamDefaultController
 - ReadableStreamDefaultReader
 - TransformStream
 - TransformStreamDefaultController
 - WritableStream
 - WritableStreamDefaultController
- [URL] defines the following terms:
 - URL
 - URLSearchParams
- [URLPATTERN] defines the following terms:
 - URLPattern
- [WASM-JS-API-2] defines the following terms:
 - Global
 - Instance
 - Memory
 - Module
 - Table
 - WebAssembly
 - compile(bytes)
 - instantiate(bytes)
 - validate(bytes)
- [WASM-WEB-API-2] defines the following terms:
 - compileStreaming(source)
 - instantiateStreaming(source)
- [WEBCRYPTO-2] defines the following terms:
 - Crypto
 - CryptoKey
 - SubtleCrypto
 - crypto
- [WEBIDL] defines the following terms:
 - DOMException
- [XHR] defines the following terms:
 - FormData

Copyright & Software License

Ecma International

Rue du Rhone 114

CH-1204 Geneva

Tel: +41 22 849 6000

Fax: +41 22 849 6001

Web: <https://ecma-international.org/>

Copyright Notice

ALTERNATIVE COPYRIGHT NOTICE AND COPYRIGHT LICENSE

© 2025 Ecma International

By obtaining and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission under Ecma's copyright to copy, modify, prepare derivative works of, and distribute this work, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the work or portions thereof, including modifications:

(i) The full text of this COPYRIGHT NOTICE AND COPYRIGHT LICENSE in a location viewable to users of the redistributed or derivative work.

(ii) Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the Ecma alternative copyright notice should be included.

(iii) Notice of any changes or modifications, through a copyright statement on the document such as "This document includes material copied from or derived from !DOCUMENT!.
Copyright © Ecma International."

Disclaimers

THIS WORK IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE DOCUMENT WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the work without specific, written prior permission. Title to copyright in this work will at all times remain with copyright holders.

Software License

All Software contained in this document ("Software") is protected by copyright and is being made available under the "BSD License", included below. This Software may be subject to third party rights (rights from parties other than Ecma International), including patent rights, and no licenses under such third party rights are granted under this license even if the third party concerned is a member of Ecma International. SEE THE ECMA CODE OF CONDUCT IN PATENT MATTERS AVAILABLE AT <https://ecma-international.org/memento/codeofconduct.htm> FOR INFORMATION REGARDING THE LICENSING OF PATENT CLAIMS THAT ARE REQUIRED TO IMPLEMENT ECMA INTERNATIONAL STANDARDS.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor Ecma International may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE ECMA INTERNATIONAL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ECMA INTERNATIONAL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.