mdn web docs

PerformanceObserver

> **Note:** This feature is available in Web Workers

The `PerformanceObserver` interface is used to observe performance measurement events and be notified of new performance entries as they are recorded in the browser's *performance timeline*.

Constructor

`PerformanceObserver()`

Creates and returns a new `PerformanceObserver` object.

Static properties

`PerformanceObserver.supportedEntryTypes` Read only

Returns an array of the `entryType` values supported by the user agent.

Instance methods

`PerformanceObserver.observe()`

Specifies the set of entry types to observe. The performance observer's callback function will be invoked when performance entry is recorded for one of the specified `entryTypes`.

`PerformanceObserver.disconnect()`

Stops the performance observer callback from receiving performance entries.

`PerformanceObserver.takeRecords()`

Returns the current list of performance entries stored in the performance observer, emptying it out.

Examples

Creating a PerformanceObserver

The following example creates a `PerformanceObserver` watching for "mark" (`PerformanceMark`) and "measure" (`PerformanceMeasure`) events. The `perfObserver` callback provides a `list` (`PerformanceObserverEntryList`) which allows you to get observed performance entries.

```
JS
function perfObserver(list, observer) {
  list.getEntries().forEach((entry) => {
    if (entry.entryType === "mark") {
      console.log(`${entry.name}'s startTime: ${entry.startTime}`);
    }
    if (entry.entryType === "measure") {
      console.log(`${entry.name}'s duration: ${entry.duration}`);
    }
  });
}
const observer = new PerformanceObserver(perfObserver);
observer.observe({ entryTypes: ["measure", "mark"] });
```

Specifications

Specification
Performance Timeline # dom-performanceobserver

Browser compatibility

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	Deno
`PerformanceObserver`	Chrome 52	Edge 79	Firefox 57	Opera 39	Safari 11	Chrome 52 Android	Firefox 57 for Android	Opera 41 Android	Safari 11 on iOS	Samsung 6.0 Internet	WebView 52 Android	Deno ?
`PerformanceObserver()` constructor	Chrome 52	Edge 79	Firefox 57	Opera 39	Safari 11	Chrome 52 Android	Firefox 57 for Android	Opera 41 Android	Safari 11 on iOS	Samsung 6.0 Internet	WebView 52 Android	Deno ?
`droppedEntriesCount()` parameter	Chrome 95	Edge 95	Firefox No	Opera 81	Safari No	Chrome 95 Android	Firefox No for Android	Opera 67 Android	Safari No on iOS	Samsung 17.0 Internet	WebView 95 Android	Deno ?
`disconnect`	Chrome 52	Edge 79	Firefox 57	Opera 39	Safari 11	Chrome 52 Android	Firefox 57 for Android	Opera 41 Android	Safari 11 on iOS	Samsung 6.0 Internet	WebView 52 Android	Deno ?
`observe`	Chrome 52	Edge 79	Firefox 57	Opera 39	Safari 11	Chrome 52 Android	Firefox 57 for Android	Opera 41 Android	Safari 11 on iOS	Samsung 6.0 Internet	WebView 52 Android	Deno ?

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	Deno
`supportedEntryTypes()` static method	Chrome 73	Edge 79	Firefox 68	Opera 60	Safari 13	Chrome 73 Android	Firefox 68 for Android	Opera 52 Android	Safari 13 on iOS	Samsung 11.0 Internet	WebView 73 Android	Deno ?
`takeRecords`	Chrome 65	Edge 79	Firefox 60	Opera 52	Safari 15	Chrome 65 Android	Firefox 60 for Android	Opera 47 Android	Safari 15 on iOS	Samsung 9.0 Internet	WebView 65 Android	Deno ?
Available in workers	Chrome 62	Edge 79	Firefox 57	Opera 49	Safari 11	Chrome 62 Android	Firefox 57 for Android	Opera 46 Android	Safari 11 on iOS	Samsung 8.0 Internet	WebView 62 Android	Deno ?

Tip: you can click/tap on a cell for more information.

Full support Partial support No support Experimental. Expect behavior to change in the future.

See also

- `MutationObserver`
- `ResizeObserver`
- `IntersectionObserver`