High Resolution Timestamps for Events

By Jeff Posnick
Web DevRel @ Google

The `timeStamp` property of the `Event` interface indicates the time at which a given event took place.

In versions of Chrome prior to 49, this `timeStamp` value was represented as a `DOMTimeStamp`, which was a whole number of milliseconds since the system epoch, much like the value returned by `Date.now()`.

Starting with Chrome 49, `timeStamp` is a `DOMHighResTimeStamp` value. This value is still a number of milliseconds, but with microsecond resolution, meaning the value will include a decimal component. Additionally, instead of the value being relative to the epoch, the value is relative to the `PerformanceTiming.navigationStart`, i.e. the time at which the user navigated to the page.

The benefits of additional time stamp accuracy can be seen in these examples:

- Calculating mouse velocity
- Measuring scroll “jank”

Cross-browser and legacy considerations

If you have existing code that compares `Event.timeStamp` values from two events, you should not have to adjust your code on account of the shift to `DOMHighResTimeStamp`. Moreover, on browsers that support `DOMHighResTimeStamp`, your existing code will benefit from the increased microsecond accuracy, as well as the fact that the `DOMHighResTimeStamp` is guaranteed to increase monotonically, regardless of whether the system clock changes in the middle of your web page’s execution.

If, instead of comparing two `Event.timeStamp` values, your code needs to determine how long ago an event took place, the new `DOMHighResTimeStamp` value can be compared directly to `performance.now()`. And if you need to transform `Event.timeStamp` to an absolute number of milliseconds since the system epoch, you can get that value by adding a `DOMHighResTimeStamp` to `performance.timing.navigationStart`.

In both of those cases, `DOMTimeStamp` and `DOMHighResTimeStamp` behave differently, but you can simplify your cross-browser code by using this conversion function, courtesy of Majid Valipour. It takes an `Event` object as a parameter and returns a `DOMHighResTimeStamp`-like value, ready to be compared to `performance.now()` or added to `performance.timing.navigationStart`.

Except as otherwise noted, the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code samples are licensed under the Apache 2.0 License. For details, see our Site Policies. Java is a registered trademark of Oracle and/or its affiliates.

Last updated July 2, 2018.