

Device Orientation Changes Are Coming to Chrome 50



By Jeff Posnick

Web DevRel @ Google

Developers working on virtual or augmented reality web apps are undoubtedly familiar with the `DeviceOrientationEvent`. For the uninitiated, “[This End Up: Using Device Orientation](#) [↗](#)” provides a great overview of how a `deviceorientation` event listener can respond to a device twisting and turning.

In earlier versions of Chrome, the `alpha`, `beta`, and `gamma` values included in the `DeviceOrientationEvent` were provided as absolute degrees with respect to the Earth’s coordinate frame. Providing absolute degrees requires using a device’s magnetometer sensor to detect the Earth’s magnetic field, and that in turn is susceptible to nearby magnetic field fluctuations that could throw off the readings. In practice, this could lead to a web app registering a bunch of `DeviceOrientationEvents` due to a nearby magnet, despite the device itself not actually moving. For a virtual reality application that only cares about tracking changes in orientation, this magnetic noise is bad news.

What’s changing?

Starting with Chrome 50, the degrees included in the `DeviceOrientationEvent` are by default no longer absolute with respect to the Earth’s coordinate frame. This means that `DeviceOrientationEvents` should only be triggered when there’s actual movement, as detected by some combination of a device’s accelerometer and gyroscope. The magnetometer, and false readings due to magnetic field fluctuations, are out of the picture.

But I still need absolute degrees!

If you’re writing JavaScript that needs to use absolute degrees, perhaps as part of an augmented reality web application that needs to map directly onto the physical world, you’re not out of luck. The previous behavior, dependent on a device’s magnetometer, is available via a new `deviceorientationabsolute` event. From a developer’s perspective, it’s analogous

to the existing `DeviceOrientationEvent`, with the guarantee that the `absolute` property will be set to `true`.

Detecting what's supported

Developers who would prefer absolute degrees can use feature detection to determine whether they're on a browser that supports the new `DeviceOrientationAbsoluteEvent` event:

```
if ('ondeviceorientationabsolute' in window) {  
    // We can listen for the new deviceorientationabsolute event.  
} else if ('ondeviceorientation' in window) {  
    // We can still listen for deviceorientation events.  
    // The `absolute` property of the event tells us whether  
    // or not the degrees are absolute.  
}
```



Cross-browser compatibility


The values reported in the `DeviceOrientationEvent` have never been consistent.

Safari and Firefox on iOS uses relative values for the degrees, which matches the implementation change introduced in Chrome 50. The change should lead to more consistency with web applications that were written with iOS in mind.

Firefox (on platforms other than iOS), Edge, and Chrome versions prior to 50 use absolute degree values for the `DeviceOrientationEvent` when run on devices with the appropriate sensors.

As of this writing, Chrome 50 is the first browser to support the new `DeviceOrientationAbsoluteEvent`.

Advanced orientation tracking with the `DeviceMotionEvent`

[Boris Smus](#)  has a [fantastically detailed article](#) detailing with some of the downsides of using the `DeviceOrientationEvent`, and how to implement a bespoke sensor fusion using `DeviceMotionEvents`. They provide low-level access to the accelerometer and gyroscope, and can lead to a more accurate virtual reality experience for your users.

Additional resources

- [The DeviceOrientation Event specification](#)
- [A deviceorientation diagnostics page](#)

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.