

**PUSH  
BUTTON  
RECEIVE  
BEACON**

# Beacon API

JavaScript-based Web API in modern browsers for sending small amounts of data to the web server *without waiting for a response*.

We don't even get a response if we want one.

[www.w3.org/TR/beacon](http://www.w3.org/TR/beacon)

# WHY WOULD WE WANT THAT?

A number of use cases fall into the *send and forget* model.

## **Analytics**

**Log page views, time on page, interaction patterns, what was shown?**

## **Diagnostic logging**

**Trap errors in production and report back**

## **Multivariate tests**

**Track which variants were used and how they performed**

*and so on...*

# WE CAN DO THIS ALREADY

I know what you're thinking. We can do this with the Fetch API or XHR.

But can we?

Yes.

*however...*

# NON-BLOCKING

Unlike everything else, Beacon is non-blocking.

It doesn't return a result, so the browser can background the request and let code execution continue.

# Turns out this is **super important**

Many logging behaviours are fired when the user leaves the page on the `unload` or `beforeunload` events.

Slow running code (like duh, HTTP requests) in these events kills performance.

Beacon **queues the request** and lets the browser continue, executing it in the background when it can.

# BASIC USE

```
let result = navigator.sendBeacon(url, data);
```



# Posting data

The data parameter can be in one of several formats, all of which are taken directly from the Fetch API.

- Blob
- BufferSource
- FormData
- URLSearchParams

# Posting data

```
let url = '/api/my-endpoint';  
  
let data = new FormData();  
data.append('hello', 'world');  
  
let result = navigator.sendBeacon(url, data);  
  
if (result) console.log('Successfully queued!');
```

# Browser support

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			66		10.3				4
		60	67		11.2				6.2
11	17	61	68	11.1	11.4	all	67	11.8	7.2
	18	62	69	12	12				
		63	70	TP					
			71						

# Browser support

It's easy to test for:

```
if (navigator.sendBeacon) {  
    // Beacon code  
} else {  
    // No Beacon. Maybe fall back to XHR?  
}
```

# Logging page view time

```
let startTime = performance.now();

let logVisit = function() {

    if (!navigator.sendBeacon) return true;

    let url = '/api/log-visit';

    let data = new FormData();
    data.append('start', startTime);
    data.append('end', performance.now());
    data.append('url', document.URL);

    navigator.sendBeacon(url, data);
};

window.addEventListener('beforeunload', logVisit);
```

# Considerations when logging

## GDPR

Maybe don't log stuff you don't need to log. Don't identify any sessions without considering privacy and your permitted use.

## Do Not Track

If you can, honour the user's DNT header.

DNT: 0

DNT: 1

# Thanks!

I'm @drewm

Slides on Notist: [noti.st/drewm](https://noti.st/drewm)