

FormData Methods for Inspection and Modification



By Sam Thorogood

Evangelises Chrome and the mobile web in the Developer Relations team at Google.

`FormData` is the XHR user's best friend, and it's getting an upgrade in Chrome 50. We're adding methods allowing you to inspect your `FormData` objects or modify them after-the-fact. You can now use `get()`, `delete()`, and iteration helpers like `entries`, `keys`, and more. ([Check out the full list.](#))

If you're not already using `FormData`, it's a simple, well-supported API that allows you to programmatically build a virtual form and send it to a far away place using `window.fetch()` or `XMLHttpRequest.send(formData)`.

For some examples, read on!

Parse real forms like a pro

`FormData` can be constructed from a real HTML form, taking a snapshot of all its current values. However, the object used to be entirely opaque. All you could do was send it on, unchanged, to a server. Now, you can take it, modify it, bop it, observe it, shrink it, change it, and finally, upload it:

```
function sendRequest(theFormElement) {  
  var formData = new FormData(theFormElement);  
  formData.delete("secret_user_data"); // don't include this one!  
  if (formData.has("include_favorite_color")) {  
    formData.set("color", userPrefs.getColor());  
  }  
  // log all values like <input name="widget">  
  console.info("User selected widgets", formData.getAll("widget"));  
  
  window.fetch(url, {method: 'POST', body: formData});  
}
```



You can also send `FormData` via the older `XMLHttpRequest`:

```
var x = new XMLHttpRequest();
x.open('POST', url);
x.send(formData);
```



Don't throw away your FormData

If you're building your own `FormData` from scratch, you might have found it frustrating that you couldn't reuse it - you've spent a lot of time on those fields! As both the `window.fetch()` and `XMLHttpRequest.send()` methods takes a snapshot of the `FormData`, you can now safely reuse and modify your work! Check this example out:

```
// append allows multiple values for the same key
var formData = new FormData();
formData.append("article", "id-123");
formData.append("article", "id-42");

// send like request
formData.set("action", "like");
window.fetch(url, {method: 'POST', body: formData});

// send reshare request
formData.set("action", "reshare"); // overrides previous "action"
window.fetch(url, {method: 'POST', body: formData});
```



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.