



Sending Form Data

This lesson teaches how to send data from a form to the server.

If the web server expects direct form data, you can use the JavaScript `FormData` object to encapsulate the information to be sent. Here's an example form for choosing the strongest animal of all and associated JavaScript code which handles the form submission.

Output
JavaScript
HTML
<div><h3>Which one is the strongest?</h3><p><input checked="" type="radio"/> The elephant <input type="radio"/> The rhinoceros <input type="radio"/> The hippopotamus</p><p>Your name: <input type="text"/></p></div>



Console

Clear

The event listener starts by disabling the default form submission behavior, which is to send a synchronous HTTP `POST` request to a server. Instead, a `FormData` object is created with the form itself (the `e.target` expression) as a parameter. All form fields are automatically added as key/value pairs in this object.

Once the form fields are encapsulated in the `FormData` object, the `fetch()` method seen previously is used to send an asynchronous request to the `https://thejsway.com/en/animals` URL. The second parameter of

the `fetch()` call sets the HTTP method as `POST` and adds the form data into the body of the request.

Lastly, the page's `result` element is updated when the server responds to the asynchronous request.

Which one is the strongest?

- ☐ The elephant
- ☐ The rhinoceros
- ☒ The hippopotamus

Your name:

Hello Baptiste, you voted: HIP

The `FormData` object can also be used independently of any form, to send custom key/value pairs to a server. Here is a very basic example form containing only a button.

When the user clicks on the button, custom data is added to a `FormData` object and sent to the server through an asynchronous `POST` request.

Output

JavaScript

HTML

```
<html>
<head>
</head>
<body>
  <button id="buyButton">Buy a new t-shirt</button>
  <p id="result"></p>
</body>
</html>
```



Command received! Size: L, color: blue