# Fetch API - Send POST Requests

Just like XMLHttpRequest, Fetch API also provides a JavaScript interface to manage requests and responses to and from the web server asynchronously. It provides a fetch() method to fetch resources or send the requests to the server asynchronously without reloading the web page. With the help of the fetch() method, we can perform various requests like POST, GET, PUT, and DELETE. So in this article, we will learn how to send POST requests with the help of Fetch API.

## Send POST Request

Fetch API also support POST request. The POST request is an HTTP request which is used to send data or form to the given resource or the web server. In Fetch API, we can use POST requests by specifying the additional parameters like method, body headers, etc.

## Syntax

```
fetch(URL, {
   method: "POST",
   body: {//JSON Data},
   headers:{"content-type": "application/json; charset=UTF-8"}
})
.then(info =>{
   // Code
})
.catch(error =>{
   // catch error
});
```

Here the fetch() function contains the following parameters −

- **URL** − It represents the resource which we want to fetch.
- **method** − It is an optional parameter. It is used to represent the request like, GET, POST, DELETE, and PUT.
- **body** − It is also an optional parameter. You can use this parameter when you want to add a body to your request.
- **header** − It is also an optional parameter. It is used to specify the header.

## Example

In the following program, we will send a JSON document to the given URL. So for that, we define a fetch() function along with a URL, a POST request, a body(that is JSON document) and a header. So when the fetch() function executes it sends the given body to the specified URL and converts response data into JSON format using the response.json() function. After that, we will display the response.

Open Compiler

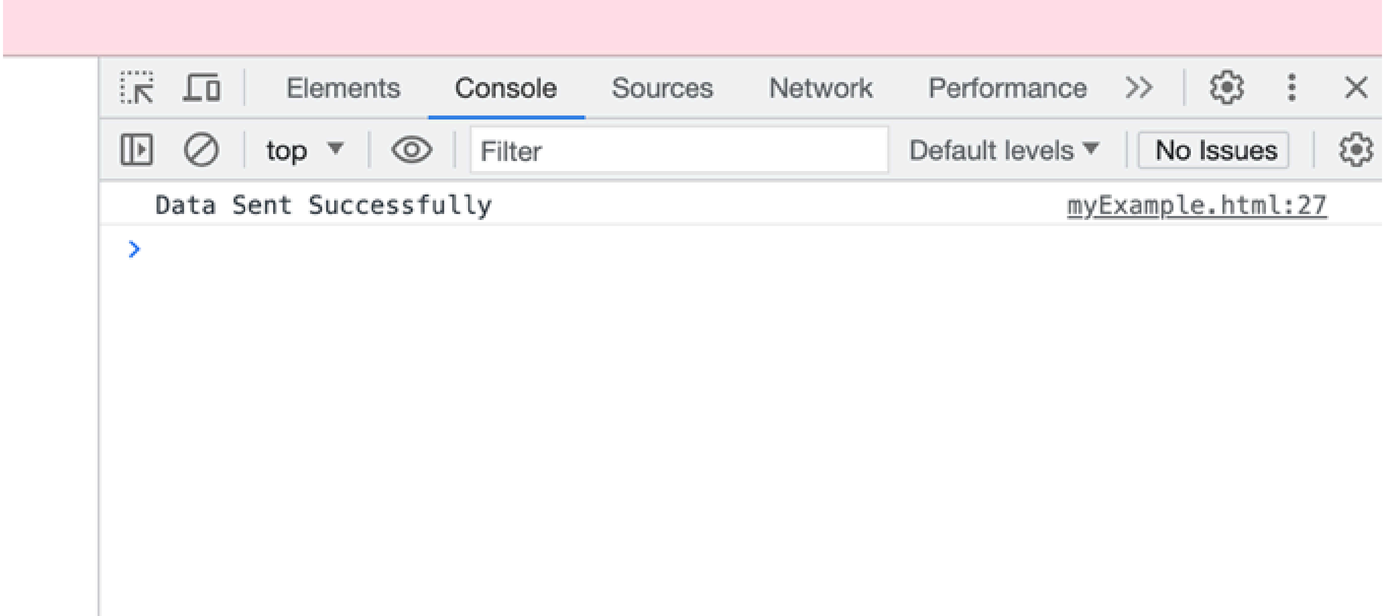
```
<!DOCTYPE html>
<html>
<body>
<script>
   // Retrieving data from the URL using POST request
   fetch("https://jsonplaceholder.typicode.com/todos", {
      // Adding POST request
      method: "POST",

      // Adding body which we want to send
      body: JSON.stringify({
      id: 32,
      title: "Hello! How are you?",
      }),
      // Adding headers
      headers:{"Content-type": "application/json; charset=UTF-8"}
   })
   // Converting received information into JSON
   .then(response => response.json())
   .then(myData => {
      // Display the retrieve Data
      console.log("Data Sent Successfully");

      // Display output
      document.getElementById("manager").innerHTML = myData;
   });
</script>
<h2>Display Data</h2>
```

```
<div>
   <!-- Displaying retrevie data-->
   <table id = "manager"></table>
</div>
</body>
</html>
```

## Output



```
Data Sent Successfully                                    myExample.html:27
```

## Conclusion

So this is how we can send the POST request using Fetch API. Using this request we can easily send data to the specified URL or server. Also using the fetch() function you can modify your request according to your requirements. Now in the next article, we will learn how to send a PUT request.