Fetch API - Send GET Requests

Fetch API provides an 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 refreshing the web page. Using the fetch() method we can perform various requests like POST, GET, PUT, and DELETE. In this article, we will learn how to send GET requests using Fetch API.

Send GET Request

The GET request is an HTTP request used to retrieve data from the given resource or the web server. In Fetch API, we can use GET requests either by specifying the method type in the fetch() function or without specifying any method type in the fetch() function.

Syntax

```
fetch(URL, {method: "GET"})
.then(info =>{
   // Code
})
.catch(error =>{
   // catch error
});
```

Here in the fetch() function, we specify the GET request in the method type.

Or

```
fetch(URL)
.then(info =>{
   // Code
})
.catch(error =>{
   // catch error
});
```

Here, in the fetch() function, we do not specify any method type because by default fetch() function uses a GET request.

Example

In the following program, we will retrieve id and titles from the given URL and display them in the table. So for that, we define a fetch() function with a URL from where we retrieve data and a GET request. This function will retrieve data from the given URL and then convert the data into JSON format using the response.json() function. After that, we will display the retrieved data that is id and title in the table.

Open Compiler

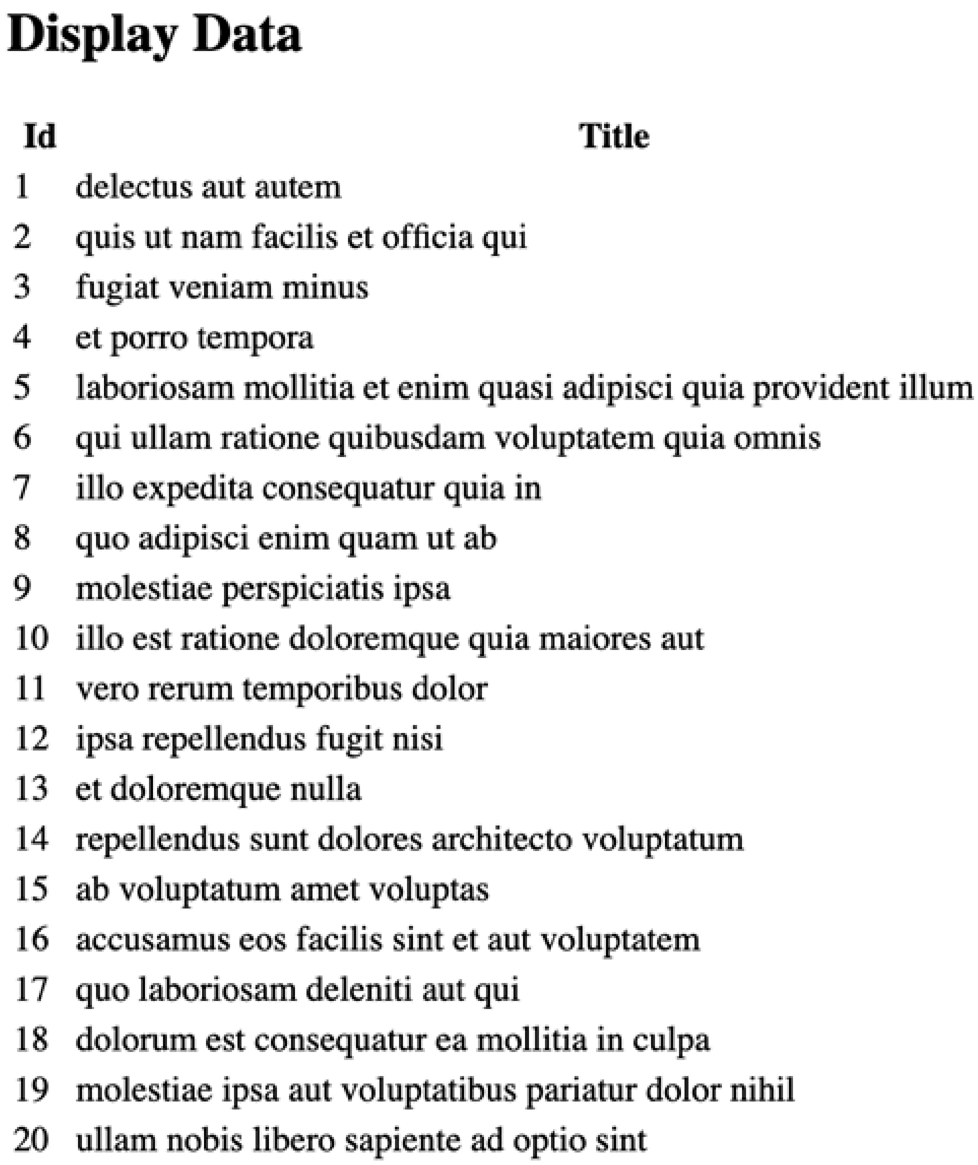
```
<!DOCTYPE html>
<html>
<body>
<script>
   // GET request using fetch()function
   fetch("https://jsonplaceholder.typicode.com/todos", {
      // Method Type
      method: "GET"})

   // Converting received data to JSON
   .then(response => response.json())
   .then(myData => {
      // Create a variable to store data
      let item = `<tr><th>Id</th><th>Title</th></tr>`;

      // Iterate through each entry and add them to the table
      myData.forEach(users => {
         item += `<tr>
         <td>${users.id} </td>
         <td>${users.title}</td>
         </tr>`;
      });
      // Display output
      document.getElementById("manager").innerHTML = item;
   });
</script>
<h2>Display Data</h2>
<div>
```

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

Output



Display Data

Id	Title
1	delectus aut autem
2	quis ut nam facilis et officia qui
3	fugiat veniam minus
4	et porro tempora
5	laboriosam mollitia et enim quasi adipisci quia provident illum
6	qui ullam ratione quibusdam voluptatem quia omnis
7	illo expedita consequatur quia in
8	quo adipisci enim quam ut ab
9	molestiae perspiciatis ipsa
10	illo est ratione doloremque quia maiores aut
11	vero rerum temporibus dolor
12	ipsa repellendus fugit nisi
13	et doloremque nulla
14	repellendus sunt dolores architecto voluptatum
15	ab voluptatum amet voluptas
16	accusamus eos facilis sint et aut voluptatem
17	quo laboriosam deleniti aut qui
18	dolorum est consequatur ea mollitia in culpa
19	molestiae ipsa aut voluptatibus pariatur dolor nihil
20	ullam nobis libero sapiente ad optio sint

Conclusion

So this is how we can send the GET request using Fetch API so that we can request a specific resource or document from the given URL. Using the fetch() function we can also customise the GET request according to our requirements. Now in the next article, we will learn how to send a POST request.