In JavaScript, you can make an HTTP request using the built-in `fetch()` function or the `XMLHttpRequest` object. Here's an example of how to use `fetch()` to make a GET request:

javascript
```
fetch('https://example.com/data')
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error => console.error(error));
```

This code makes a GET request to `https://example.com/data`, and then logs the response data to the console as a JSON object. If there's an error, it will log the error to the console.

Here's an example of how to use `XMLHttpRequest` to make a POST request:

javascript
```
const xhr = new XMLHttpRequest();
xhr.open('POST', 'https://example.com/submit');
xhr.setRequestHeader('Content-Type', 'application/json');
xhr.onload = function() {
  if (xhr.status === 200) {
    console.log(xhr.response);
  } else {
    console.error(xhr.statusText);
  }
};
xhr.onerror = function() {
  console.error('Request failed');
};
xhr.send(JSON.stringify({ name: 'John', age: 30 }));
```

This code creates a new `XMLHttpRequest` object, sets the request method to POST and the request URL to `https://example.com/submit`. It also sets the `Content-Type` header to `application/json`. The `onload` function handles the response, and the `onerror` function handles any errors. Finally, the code sends the request with the `send()` method and includes some data in the request body as a JSON string.