

HTTP methods in Fetch()

Adding HTTP methods in fetch()

In JavaScript, when using the `fetch()` function to make HTTP requests, you can specify the HTTP method (GET, POST, PUT, DELETE, etc.) directly within the `fetch()` call by including it as a property within the options object. The options object is the second argument passed to the `fetch()` function and is used to configure various aspects of the request.

Here's a breakdown of how it works:

1. Creating the fetch Request:

You start by calling the `fetch()` function with a URL as its first argument. This URL represents the endpoint you want to make the HTTP request to. The second argument is an options object, where you can specify various settings for the request, including the HTTP method.

2. Specifying the HTTP Method:

Within the options object, you can specify the HTTP method you want to use by setting the `method` property to the desired method. For example:

```
// PUT request  
  
fetch('https://api.example.com/data', {  
  method: PUT  
})
```

In this example, the method property is set to PUT, indicating that you want to perform a PUT request.

Similarly, you can specify other HTTP methods like POST, PUT, or DELETE by changing the value of the method property accordingly.

By specifying the HTTP method within the options object when calling fetch(), you can make concise and readable requests directly within the fetch() call, eliminating the need for separate method-specific functions or libraries for each HTTP method. This approach is both convenient and efficient when working with HTTP requests in JavaScript.