

# Fetch API Vs XMLHttpRequest

An **XMLHttpRequest** object is used to communicate with the server asynchronously, which means we can exchange data to or from the server in the background without refreshing the whole page. XMLHttpRequest is the most commonly used technique that's why it is used by most of the mainstream browsers like Google Chrome, Safari, Mozilla Firefox, or Opera. It also supports plain text, JSON data, and many more data formats. It is very easy to use and provides various methods and properties to perform operations. We can create an XMLHttpRequest object using XMLHttpRequest() constructor.

## Syntax

```
variableName = new XMLHttpRequest()
```

Where using a new keyword along with the XMLHttpRequest() constructor we are able to create a new XMLHttpRequest object. This object must be created before calling the open() function to initialize it before calling send() function to send the request to the web server.

**Fetch API** provides an interface that is used to fetch/retrieve resources from the server. It is a modern alternative to XMLHttpRequest. It supports the generic definition of Request, and Response due to which we can access them in the future if required for Cache API, Service work, handling or modifying requests and responses, etc. It is very easy, simple and consistent. Or we can say that it provides a modern and flexible approach for creating HTTP requests and handling responses as compared to XMLHttpRequest. It is based on the Promise API which provides clear syntax and better error handling.

## Syntax

```
fetch(res)
```

Where fetch() takes one mandatory parameter which is res. The res parameter defines the resource that you want to fetch, it can be either a string or any other object or can be a request object. Apart from mandatory parameters, it can also take one optional parameter that can be either method, headers, body, mode cache, same-origin, etc.

Explore our [latest online courses](#) and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

## Fetch API VS XMLHttpRequest

Following are the difference between the Fetch API and XMLHttpRequest –

Difference	Fetch API	XMLHttpRequest
<b>Syntax</b>	As we know Fetch API is a promise-based API so it provides more clear syntax and better error-handling methods.	XHR is based on a callback approach and its syntax is not as good as Fetch API.
<b>Cross-Origin Resource Sharing(CROS)</b>	Fetch API handle CROS request wisely without any additional configuration.	XMLHttpRequest requires a special configuration to handle or make cross-origin requests.
<b>Request and Response Header</b>	Fetch API provides more flexible ways to work with request and response headers.	XMLHttpRequest provides a limited number of methods to work with request and response headers.
<b>Streaming and Parsing</b>	Fetch API provides good support for streaming and parsing large responses, so it improves performance and reduces memory usage.	XMLHttpRequest requires a custom program to get this functionality.
<b>Browser Compatibilities</b>	Fetch API is new so it may not be supported by older versions of the browsers.	XMLHttpRequest has been used for many years so it is supported by almost all browsers.
<b>Cookies and Credential Control</b>	Fetch API provides good control over cookies and credentials due to which we can easily do authentication and authorisation as compared to XMLHttpRequest.	XMLHttpRequest provide less support to cookies and credentials.
<b>Timeouts</b>	Fetch API does not support timeouts so the request will continue till the browser selects the request.	XMLHttpRequest supports timeouts.

## Conclusion

So these are the major difference between Fetch API and XMLHttpRequest. Fetch API is more powerful and easier to understand as compared to XMLHttpRequest. It is also supported by all modern browsers but XMLHttpRequest is only supported by old browsers. Now in the next article, we will learn about fetch() function.