

What Is HTMX?

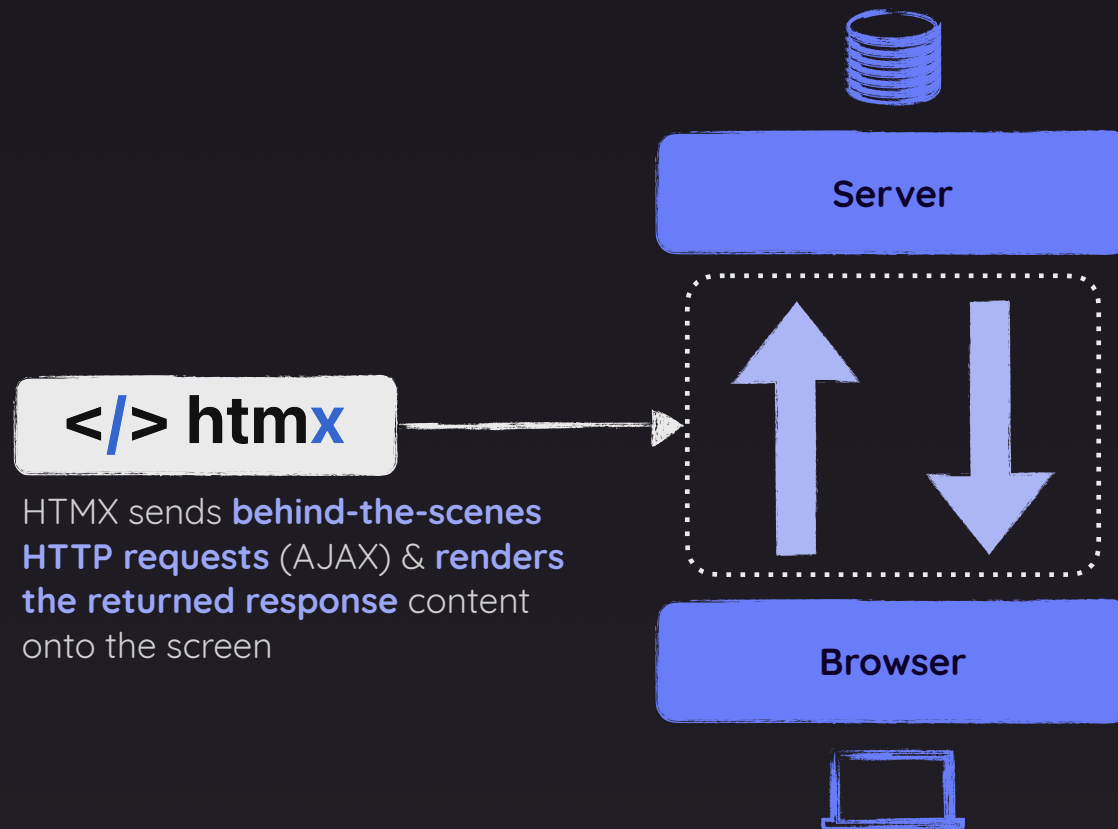
HTMX extends HTML

It adds new attributes you can
set on HTML elements

HTMX is a JavaScript library
that helps you avoid JavaScript

HTMX Deals With HTTP Requests

HTMX sends AJAX requests & renders response content onto the screen



Frontend & Backend

HTMX is used on the frontend & runs in the browser

Option 1

HTMX in a decoupled frontend

Use HTMX in a decoupled frontend app - just as you would use React or Angular etc.

But unlike React etc., HTMX focuses on updating the UI because of AJAX requests

Option 2

HTMX in an integrated fullstack app

HTMX is pretty useless without a (tightly connected) backend

HTMX expects HTML code as response content

HTMX Works With Any Tech Stack!

ANY LANGUAGE

You can use HTMX with Node, PHP, Python, Java or any other language

ANY FRAMEWORK

And you can of course use any framework of your choice (or none at all)

ANY HTML RENDERING PROCESS

You can also render the HTML code in any way you like - hardcoded as a string, with help of a template engine - it doesn't matter!

What if you want to add client-side interactivity that's not related to **AJAX** requests?

What if you want to add client-side interactivity
that's not related to AJAX requests?

You can do that!

You can add your own (vanilla) JS code to the
website and you can also use any JS libraries or
frameworks of your choice!

HTTP Methods

HTTP supports different methods

HTML only supports these two methods

GET

The default request if you enter a URL or click a link

Typically used to fetch data (e.g., an HTML document or JSON data)

POST

Often set as a “method” on <form> elements

Typically used to submit data from the client to the server (to potentially save it there)

Since HTMX sends AJAX requests via JS under the hood, it supports these methods

DELETE

Typically used to delete data on the server

PATCH

Typically used to update data on the server

PUT

Typically used to replace data on the server

Can be used via JavaScript (can be set on AJAX requests)

Important

It's still up to the server to perform the actual work. You could, theoretically, create & store data upon a DELETE request.