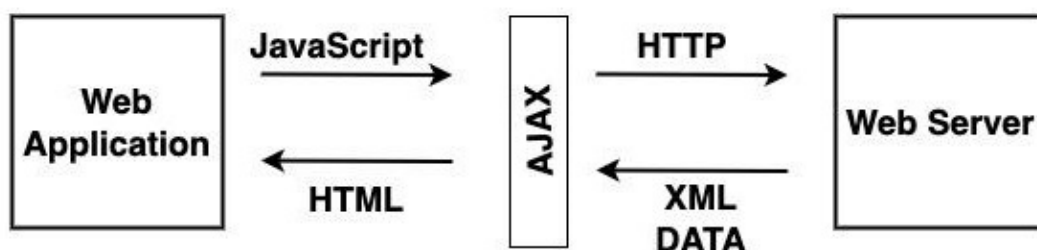


PHP - AJAX Introduction

PHP powered web applications often make use of AJAX, together they are useful to create dynamic and interactive web applications. AJAX stands for **Asynchronous Javascript and XML**. It allows webpages to be updated asynchronously without reloading the entire page.

In AJAX applications, the exchange of data between a web browser and the server-side PHP script is asynchronous. PHP is a server-side scripting language that can be used to generate dynamic content and process data.

AJAX creates an additional layer known as AJAX engine in between the web application and web server due to which we can make background server calls using JavaScript and retrieve the required data, can update the requested portion of a web page without causing full reload of the page. It reduces the page refresh timing and provides a fast and responsive experience to the user.



What is Required to Run AJAX?

The technologies that are used by AJAX are already implemented in all the Modern browsers. So the client does not require any extra module to run the AJAX application. The technologies used by AJAX are –

- **Javascript** – It is an important part of AJAX. It allows you to create client-side functionality. Or we can say that it is used to create AJAX applications.
- **XML** – It is used to exchange data between web server and client.
- **The XMLHttpRequest** – It is used to perform asynchronous data exchange between a web browser and a web server.
- **HTML and CSS** – It is used to provide markup and style to the webpage text.
- **DOM** – It is used to interact with and alter the webpage layout and content dynamically.

To use AJAX with PHP, you will need to use the XMLHttpRequest object in JavaScript to send requests to the PHP server. The PHP server will then process the request and return a response, typically in the form of JSON or XML. The JavaScript code can then parse the response and update the web page accordingly.

The XMLHttpRequest object in JavaScript is a browser-based API that allows developers to make HTTP requests to a server without reloading the page. This is the foundation of AJAX programming, which allows for dynamic and interactive web applications.

The XMLHttpRequest object can be used to –

- Retrieve data from a server, such as JSON, XML, or HTML.
- Send data to a server, such as form data or file uploads.
- Update a web page without reloading it.
- Create chat applications and other interactive features.

To use the XMLHttpRequest object, you first need to create a new instance of it. Then, you can use the `open()` method to specify the HTTP method and request URL. Next, you can set any request headers, if needed. Finally, you can send the request using the `send()` method.

Example

Here is a simple JavaScript code of how to use the XMLHttpRequest object to retrieve data from a server –

```
// Create a new XMLHttpRequest object
var xhr = new XMLHttpRequest();

// Set the HTTP method and request URL
xhr.open("GET", "test.php");

// Send the request
xhr.send();

// Listen for the onload event to be fired
xhr.onload = function() {

    // Check the status code to ensure the request was successful
    if (xhr.status === 200) {

        // Get the response data.
        var users = JSON.parse(xhr.responseText);
```

```
        // Do something with the user data.  
    } else {  
        // Handle the error  
    }  
};
```

The PHP script on the server retrieves the data from AJAX request and sends back the response.

```
// Get the request data.  
$name = $_GET["name"];  
  
// Create a response object.  
$response = new stdClass();  
$response->message = "Hello, $name!";  
  
// Send the response back to the client.  
header("Content-Type: application/json");  
echo json_encode($response);
```