

Submitted by: Afraz Butt, 2021-CS-12
Submitted to: Mr. Atif Hussain
Site Link: <https://buttafraz22.github.io/ajax-demo/>

AJAX

AJAX (Asynchronous JavaScript and XML – though sometimes just Asynchronous JavaScript) is not a singular technology but rather a group of web development techniques that work together to make web applications more dynamic and responsive.

What it does?

1. Allows content of parts of the webpage to be updated without the entire page being reloaded. This provides for a smoother experience for users.
2. Lets users interact with the server in real-time while other things are going on in the background, without having to wait for an entire web page refresh.

How does it work?

1. **Initiation:** A user interacts with the web page like clicking a button or submitting a form.
2. **JavaScript takes charge:** JavaScript code on the page detects the user action and initiates an XMLHttpRequest object.
3. **Reaching out:** This object acts like a messenger, sending a request to the server for specific data.
4. **Server does its thing:** The server processes the request and prepares the requested data.
5. **Data travels back:** The server sends back data put on this web page by this person who ordered something from this website after all was done according to his/her liking or disliking
6. **JavaScript does its magic again:** When JavaScript receives data, it acts magically by changing elements in order to reflect new information products or states that have been sent over through AJAX protocols.

Role of XMLHttpRequest And How AJAX has evolved with JSON

Understanding XMLHttpRequest (XHR) is crucial to grasping the role of AJAX. Let's dive deeper:

XMLHttpRequest (XHR):

1. This object forms the core of communication in AJAX.
2. It acts as a messenger, allowing JavaScript to make asynchronous requests to the server.
3. It can send various types of data (including XML, JSON, plain text) and receive responses.
4. Developers need to handle different states of the request (e.g., waiting, loading, completed) using callbacks or promises.

Evolution with JSON:

While AJAX initially used XML for data exchange, JSON (JavaScript Object Notation) rose in popularity due to its:

- Simplicity: Easier for humans to read and write compared to XML's verbose structure.
- Lightweight: Smaller data size leads to faster transmission and processing.
- Native integration with JavaScript: Objects created from JSON data directly match JavaScript object structures, simplifying manipulation.

This shift marked a significant evolution in AJAX:

- Reduced complexity: No need for complex XML parsing libraries.
- Improved performance: Smaller data transfers and easier processing.
- Enhanced developer experience: More intuitive data handling.

While XHR remains the foundation for making requests, it's often used in conjunction with libraries that simplify working with JSON and asynchronous operations.

Future of AJAX

AJAX's core principles live on, but its future might lie beyond XMLHttpRequest. Expect:

- Websocket
- GraphQL

AJAX's legacy paves the way for seamless, responsive web experiences, even as technology continues to evolve.