

# AJAX Implementation in Web

CS-493: Enterprise Application Development



Supervisor  
Mr. Atif Hussain

Project Developer

Amna Zafar

2021-CS-27

University of Engineering and Technology  
Lahore, Pakistan

# Contents

<b>1</b>	<b>Introduction to AJAX</b>	<b>3</b>
1.1	What is AJAX? . . . . .	3
1.2	How AJAX works? . . . . .	3
1.3	Role of XMLHttpRequest in AJAX . . . . .	3
1.4	Abstract of Project . . . . .	4
<b>2</b>	<b>Challenges for AJAX</b>	<b>4</b>
2.1	Limitations . . . . .	4
2.2	Solution to overcome challenges . . . . .	4
<b>3</b>	<b>Future of AJAX</b>	<b>5</b>
<b>4</b>	<b>For Practical Implementation</b>	<b>5</b>
4.1	Create Object . . . . .	5
4.2	Open Object . . . . .	5
4.3	On Response . . . . .	6
4.4	Send Request . . . . .	6

# 1 Introduction to AJAX

## 1.1 What is AJAX?

AJAX stands for Asynchronous JavaScript And XML. AJAX is not a programming language. AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page. AJAX just uses a combination of:

- A browser built-in XMLHttpRequest object (to request data from a web server)
- JavaScript and HTML DOM (to display or use the data)

By combining these technologies, web pages appear more responsive since small packets of data are exchanged with the server and web pages are not reloaded each time that a user makes an input change. Ajax defines a method of initiating client to server communication without page reloads. It provides a way to enable partial page updates. From a web page user perspective, it means improved interaction with a web application, which gives the user more control of their environment, similar to that of a desktop application. In a traditional web application, HTTP requests, that are initiated by the user's interaction with the web interface, are made to a web server. The web server processes the request and returns an HTML page to the client. During HTTP transport, the user is unable to interact with the web application.

## 1.2 How AJAX works?

- An event occurs in a web page (the page is loaded, a button is clicked)
- An XMLHttpRequest object is created by JavaScript
- The XMLHttpRequest object sends a request to a web server
- The server processes the request
- The server sends a response back to the web page
- The response is read by JavaScript
- Proper action (like page update) is performed by JavaScript

## 1.3 Role of XMLHttpRequest in AJAX

XMLHttpRequest is an object that sends an HTTP request to the server and interacts with the server to open a URL and retrieve data without loading the complete page so only some part of the web page changed. Here, it is a global constructor function XMLHttpRequest built into the browser exposed in javascript that is available without any package added and is not only to XML documents, nowadays JSON is mostly used to exchange data between browser and server. Uses of XMLHttpRequest object:

- It is used to make AJAX calls to exchange data from a remote web server.
- With the help of this object, users send requests to the server asynchronously and the server sends the data which we requested for Ajax.
- It is used in different protocols to make requests like HTTP, HTTPS, FTP, and FRPS.

- It is used to retrieve any type of data like XML, JSON, etc.

### Syntax

```
let objectname = new XMLHttpRequest();
```

## 1.4 Abstract of Project

For implementation of AJAX I can take products data from api. When project runs a drop down is shown two categories is shown electronics and jewelery. You can select your according to choice and data is fetched from api and shown on webpage.

## 2 Challenges for AJAX

### 2.1 Limitations

- **Browser support**

Not all browsers can support JavaScript and XMLHttpRequest objects. These objects can be handled differently by browsers with JavaScript or XMLHttpRequest support. Ajax implementations in different browsers must be taken into consideration.

- **Security**

Not all concerns can be addressed. When developing Ajax applications, security and privacy issues must be considered.

- **Accessibility**

Not all browsers support JavaScript or XMLHttpRequest objects, so you must ensure that your web application is accessible to all users.

- **Bookmark and navigation**

Search engine Ajax applications cannot be searched; Ajax elements and advanced features can be used within searchable applications.

### 2.2 Solution to overcome challenges

Here are some solutions and best practices to overcome those challenges:

- **Cross-Browser Compatibility**

Use a JavaScript library or framework like jQuery or Axios to handle AJAX requests. These libraries abstract away many cross-browser compatibility issues.

- **Handling Asynchronous Nature**

Utilize Promises or async/await syntax to handle asynchronous code more effectively. Make use of callbacks or promises for handling the results of AJAX requests to maintain code readability and manage flow.

- **SEO and Bookmarking**

Use techniques like Progressive Enhancement or Isomorphic JavaScript to ensure that essential content is accessible to search engines and users with disabled JavaScript. Implement proper URL management for AJAX-driven content to enable bookmarking and back/forward navigation.

- **Performance Optimization**

Minimize the amount of data transferred by sending only necessary information in AJAX requests. Implement caching mechanisms for repeated AJAX requests to reduce server load and improve response times. Consider lazy loading for content that is not immediately visible on the page.

### 3 Future of AJAX

Let's reflect on how AJAX fits into these practices:

- **Enhanced User Experience**

AJAX allows web pages to update content dynamically without requiring a full page reload. This capability greatly enhances the user experience by providing a more seamless and responsive interaction, a key requirement for SPAs.

- **Data Fetching in SPAs**

SPAs load a single HTML page and dynamically update content as the user interacts with the application. AJAX is fundamental for fetching data from a server without reloading the entire page, allowing SPAs to efficiently update content and provide a smoother navigation experience.

- **React, Angular, and Vue.js Integration**

AJAX is seamlessly integrated into popular JavaScript frameworks like React, Angular, and Vue.js. These frameworks provide abstractions and utilities to simplify the implementation of AJAX requests, making it easier for developers to manage asynchronous data fetching and updates within their applications.

- **State Management**

SPAs heavily rely on efficient state management to handle user interactions and update the UI accordingly. AJAX is used to retrieve and update the application's state by fetching data from APIs and ensuring that the UI remains in sync with the underlying data.

### 4 For Practical Implementation

#### 4.1 Create Object

First create xhr object.

Syntax:

```
datatype variableName = new XMLHttpRequest();
```

#### 4.2 Open Object

Now open the created object to fetch data from any source. xhr is object name that I created.

Syntax:

```
xhr.open(  
  "GET",  
  category === "Electronics"  
    ? "https://fakestoreapi.com/products/category/electronics"  
    : "https://fakestoreapi.com/products/category/jewelery",  
  true  
);
```

## 4.3 On Response

When response is ready load data and put on web page.

Syntax:

```
xhr.onload = function () {  
  if (xhr.status >= 200 && xhr.status < 300) {  
    // handle the fetched data  
    document.getElementById("data-container").innerHTML =  
      "<p>" + xhr.responseText + "</p>";  
  } else {  
    // Error - handle the error  
    console.log("Error fetching data. Status:", xhr.status);  
  }  
};
```

## 4.4 Send Request

For sending request use the following syntax.

Syntax:

```
xhr.send();
```