AJAX

Asynchronous JavaScript + XML

https://www.thruskills.com

+91 831 737 5392

What you will learn

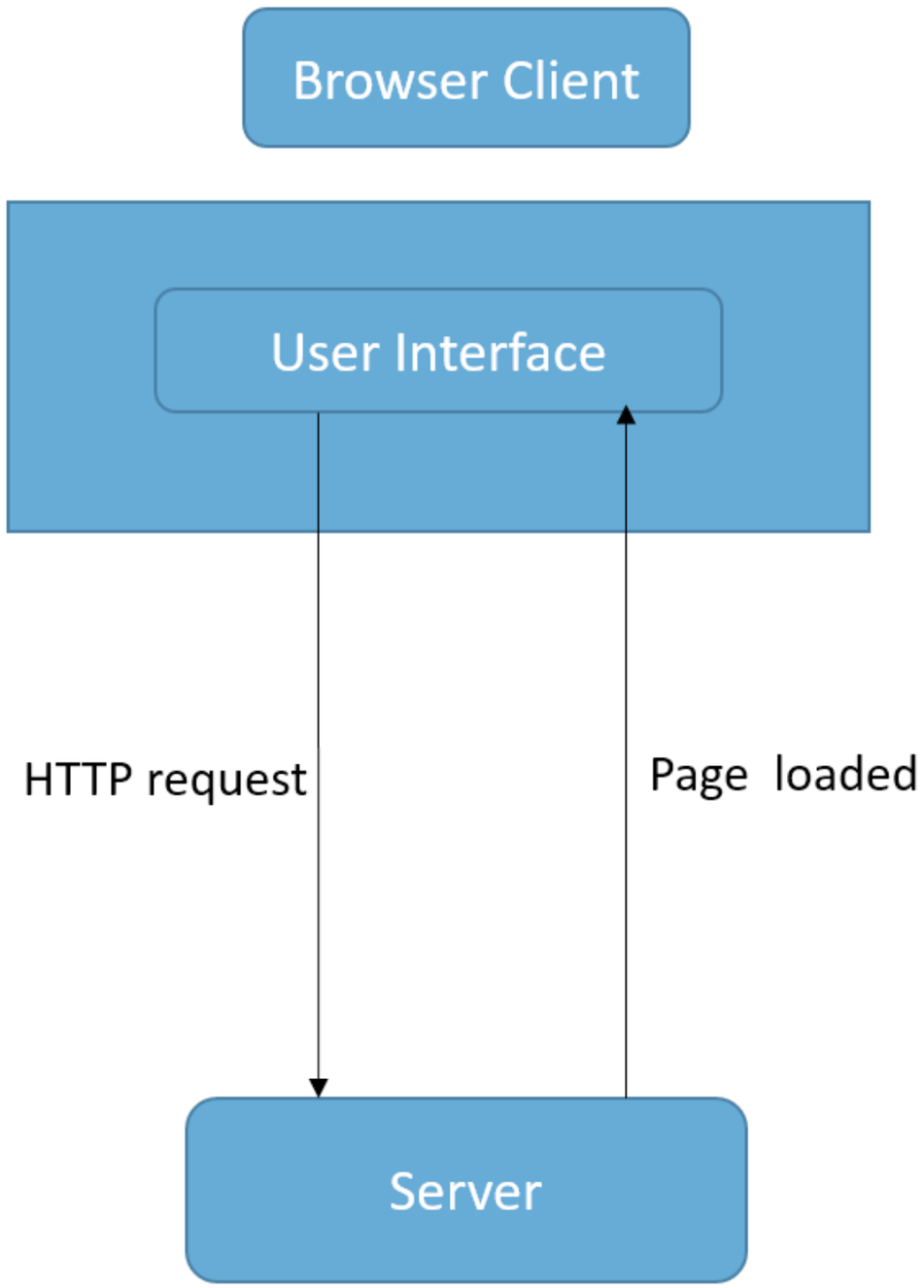
- Introduction to Ajax
- Ajax Calls Using - XMLHttpRequest
- Ajax Calls Using - Fetch API
- Ajax Calls Using - jQuery API

What are Ajax?

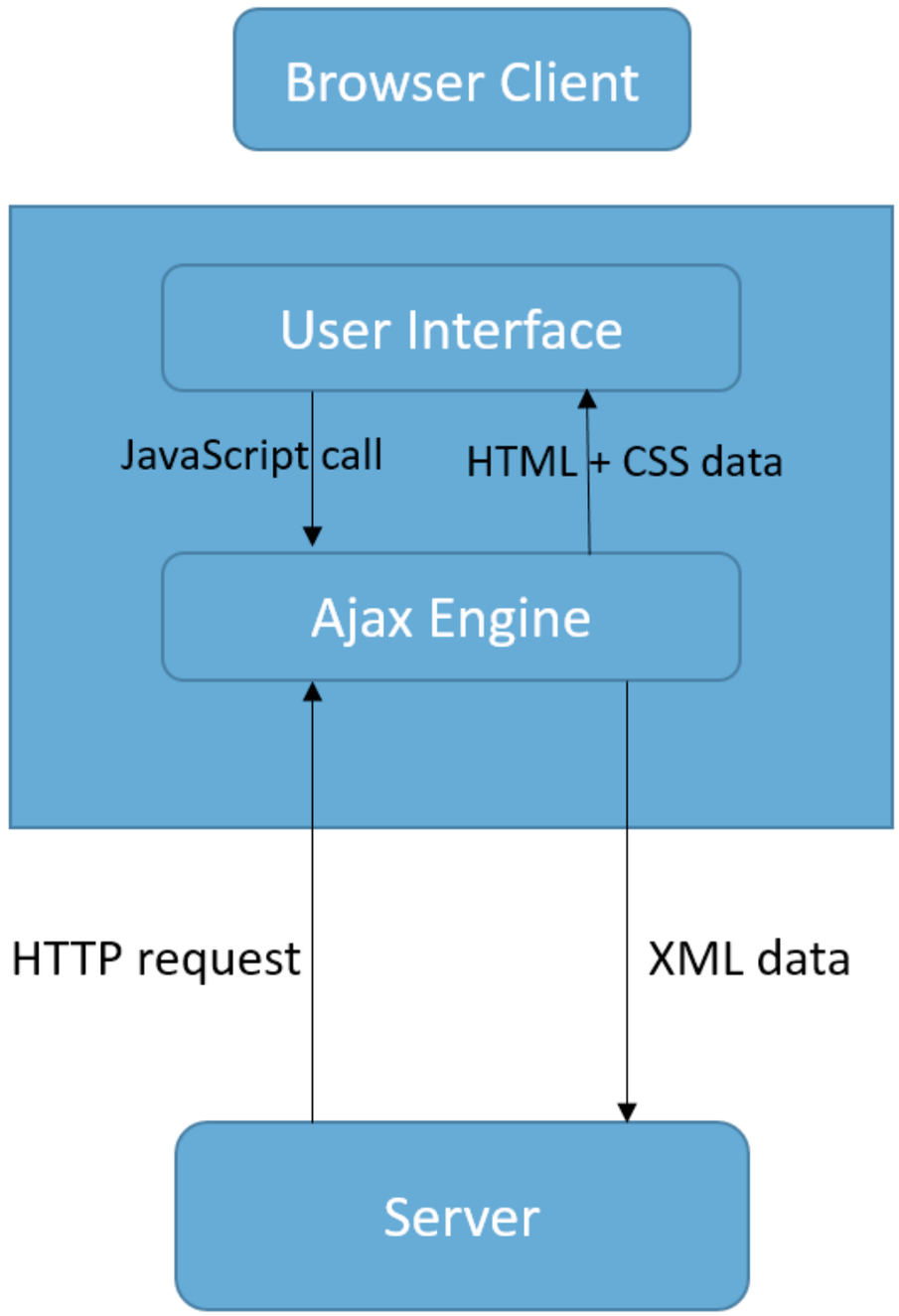
Asynchronous JavaScript + XML

While Ajax is not a technology in itself, is a term coined in 2005 by Jesse James Garrett, that describes a "new" approach to using a number of existing technologies together, including HTML, Cascading Style Sheets, JavaScript, The Document Object Model, XML, XSLT, and most importantly the XMLHttpRequest object.

What are Ajax?



Classic web application



Ajax web application

What are Ajax?

AJAX's most appealing characteristic is its "**asynchronous**" nature, which means it can communicate with the server, exchange data, and update the page without having to refresh the page.

The two major features of AJAX allow you to do the following:

- Make requests to the server without reloading the page
- Receive and work with data from the server

XMLHttpRequest (XHR)

Use **`XMLHttpRequest`** (XHR) objects to interact with servers. You can retrieve data from a URL without having to do a full page refresh. This enables a Web page to update just part of a page without disrupting what the user is doing.

`XMLHttpRequest` is used heavily in Ajax programming.

Despite its name, XMLHttpRequest can be used to retrieve any type of data, not just XML, and it supports protocols other than HTTP (including file and ftp).

The XMLHttpRequest Object

All modern browsers support the XMLHttpRequest object.

The **XMLHttpRequest** object can be used to exchange data with a server behind the scenes.

This means that it is possible to update parts of a web page, without reloading the whole page.

Create an XMLHttpRequest Object

All modern browsers (Chrome, Firefox, IE7+, Edge, Safari Opera) have a built-in XMLHttpRequest object.

Syntax for creating an XMLHttpRequest object:

```
variable = new XMLHttpRequest();
```

Example:

```
var xhr = new XMLHttpRequest();
```

XMLHttpRequest Object Methods

Method	Description
new XMLHttpRequest()	Creates a new XMLHttpRequest object
abort()	Cancels the current request
getAllResponseHeaders()	Returns header information
getResponseHeader()	Returns specific header information
open(method,url,async,user,psw)	Specifies the request
send()	Sends the request to the server
send(string)	Sends the request to the server
setRequestHeader()	Adds a label/value pair to the header to be sent

XMLHttpRequest Object Properties

Property	Description
onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest.
responseText	Returns the response data as a string
responseXML	Returns the response data as XML data
status	Returns the status-number of a request
statusText	Returns the status-text (e.g. "OK" or "Not Found")

XMLHttpRequest Object Properties…

Property	Description
responseURL	Returns the serialized URL of the response
timeout	Is an unsigned long representing the number of milliseconds a request can take before automatically being terminated.
upload	Is an XMLHttpRequestUpload, representing the upload process.
withCredentials	Is a Boolean that indicates whether or not cross-site Access-Control requests should be made using credentials such as cookies or authorization headers.

XMLHttpRequest: readyState

0: request not initialized

1: server connection established

2: request received

3: processing request

4: request finished and response is ready

XMLHttpRequest: status

200: “OK"

403: “Forbidden"

404: "Not Found"

XMLHttpRequest: status

200: “OK"

403: “Forbidden"

404: "Not Found"

XMLHttpRequest: Demos

The Fetch API

The **Fetch** API provides an interface for fetching resources (including across the network). It will seem familiar to anyone who has used XMLHttpRequest, but it provides a more powerful and flexible feature set. This article explains some of the basic concepts of the Fetch API.

At the heart of Fetch are the Interface abstractions of HTTP Requests, Responses, Headers, and Body payloads, along with a global fetch method for initiating asynchronous resource requests.

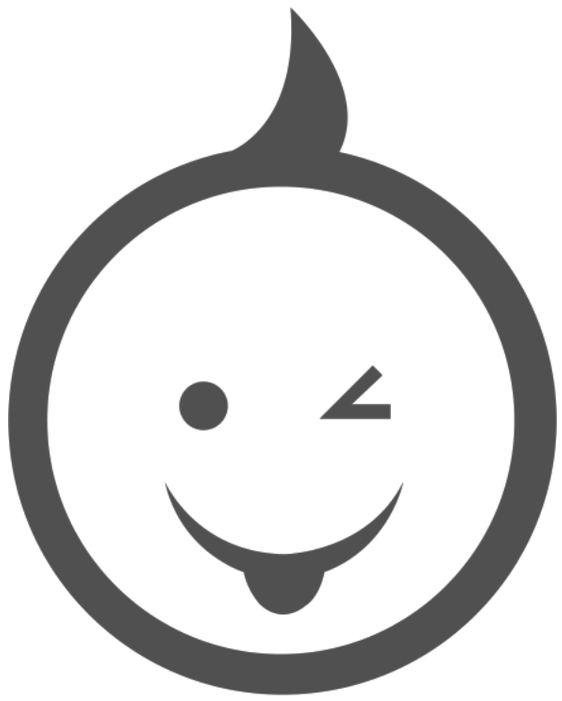
The Fetch API...

Provides a JavaScript interface for accessing and manipulating parts of the HTTP pipeline, such as requests and responses. It also provides a global fetch() method that provides an easy, logical way to fetch resources asynchronously across the network.

Fetch provides a better alternative that can be easily used by other technologies such as Service Workers.

Fetch also provides a single logical place to define other HTTP-related concepts such as CORS and extensions to HTTP.

Any Questions



Thank You

https://www.thruskills.com

+91 831 737 5392

thruskills