RICH INTERNET APPLICATION (RIA)

Module - 04

Rich Internet Application(RIA)

- **(RIA) are Web-based applications that have some characteristics of graphical desktop applications.**
- **Built with powerful development tools, RIAs can run faster and be more engaging.**
- **They can offer users a better visual experience and more interactivity than traditional browser applications that use only HTML and HTTP.**

RIA Characteristics

- **Direct interaction:**
- **Partial-page updating:**
- **Better feedback:**
- **Consistency of look and feel:**
- **Offline use**:
- **Performance impact:**

Rich Internet applications: The tools

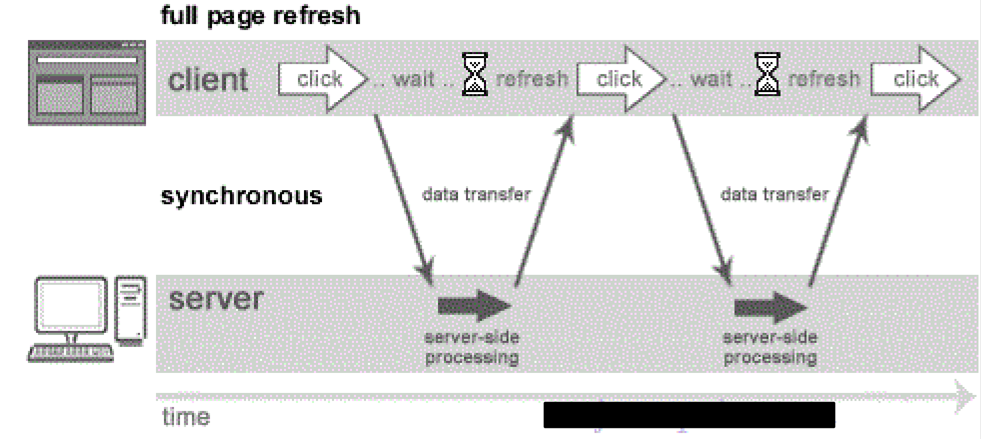
- AJAX
- Adobe Flash, Flex and Adobe Integrated Runtime (AIR)
- Microsoft Silverlight
- Curl (an object-oriented language with embedded HTML markup)
- Google Gears
- OpenLaszlo and Webtop
- Oracle WebCenter

Introduction to AJAX

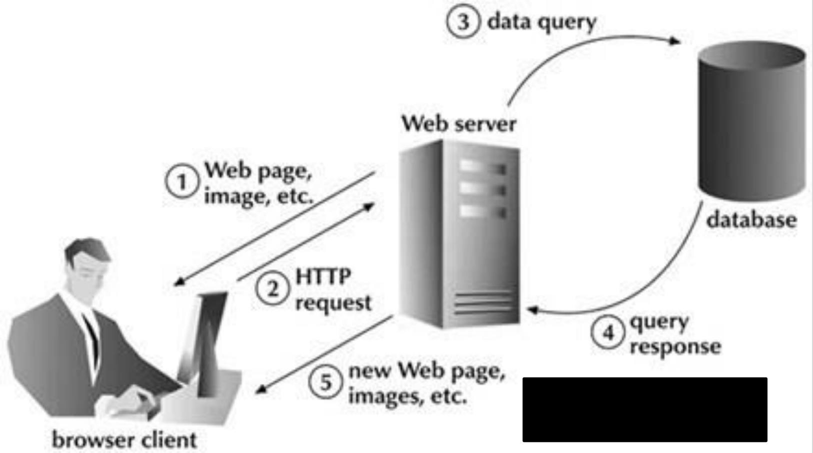
- AJAX is an acronym for **Asynchronous JavaScript and XML**.
- It is a group of inter-related technologies like JavaScript, DOM, XML, HTML/XHTML, CSS, XMLHttpRequest etc.
- AJAX allows you to send and receive data asynchronously without reloading the web page. So it is fast.
- AJAX allows you to send only important information to the server not the entire page. So only valuable data from the client side is routed to the server side. It makes your application interactive and faster.
- Where it is used?
- There are too many web applications running on the web that are using ajax technology like **gmail**, **facebook**,**twitter**, **google map**, **youtube** etc.

Understanding Synchronous vs Asynchronous

- **Synchronous (Classic Web-Application Model)**
- A synchronous request blocks the client until operation completes i.e. browser is unresponsive. In such case, javascript engine of the browser is blocked.



[redacted]

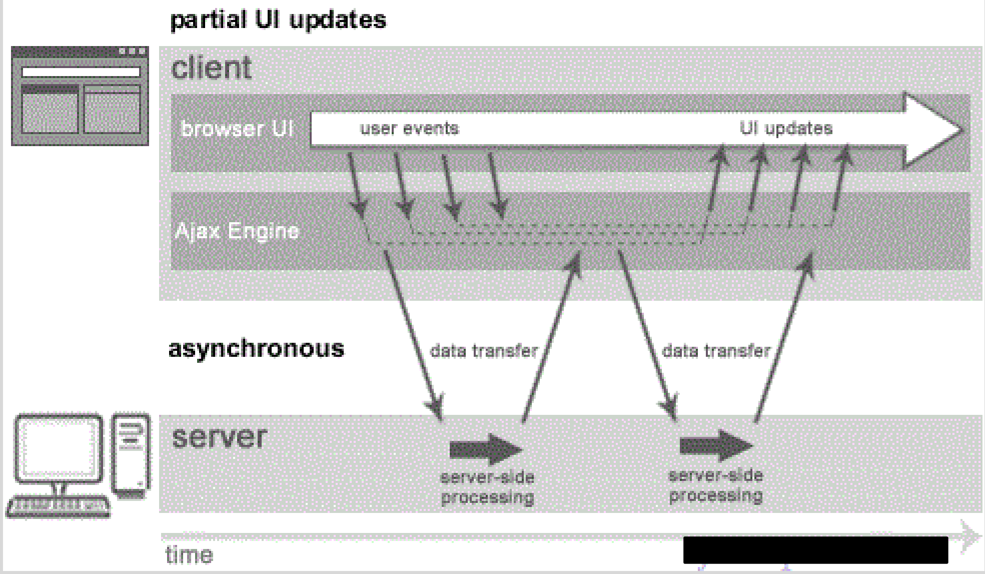


[redacted]

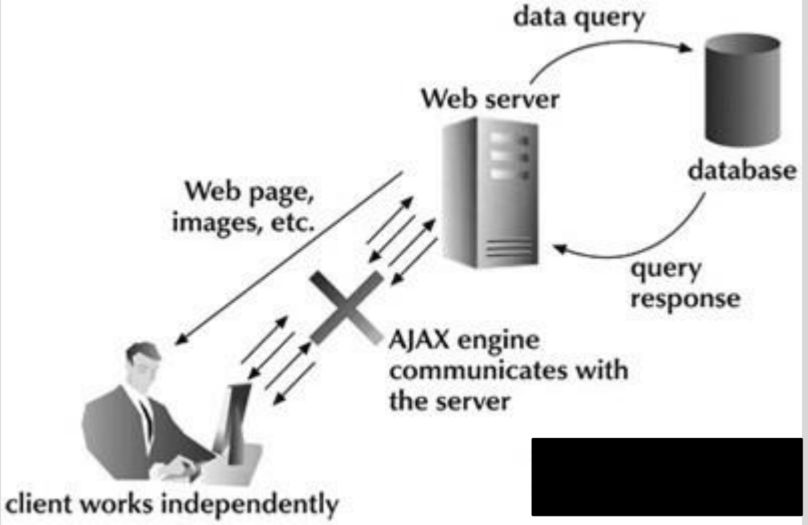
- full page is refreshed at request time and user is blocked until request completes.

Asynchronous (AJAX Web-Application Model)

- An asynchronous request doesn’t block the client i.e. browser is responsive. At that time, user can perform another operations also. In such case, javascript engine of the browser is not blocked.
- full page is not refreshed at request time and user gets response from the ajax engine.



[redacted]


data query
Web server
database
Web page, images, etc.
query response
AJAX engine communicates with the server
client works independently

[redacted]

AJAX Technologies

- HTML/XHTML and CSS
- DOM
- XML or JSON
- XMLHttpRequest
- JavaScript

XMLHttpRequest

- An object of XMLHttpRequest is used for asynchronous communication between client and server.
- It performs following operations in background:

1. Sends data from the client in the background
2. Receives the data from the server
3. Updates the webpage without reloading it.

The $.ajax() method is used for the creation of XMLHTTPRequest object.

Properties of XMLHttpRequest object

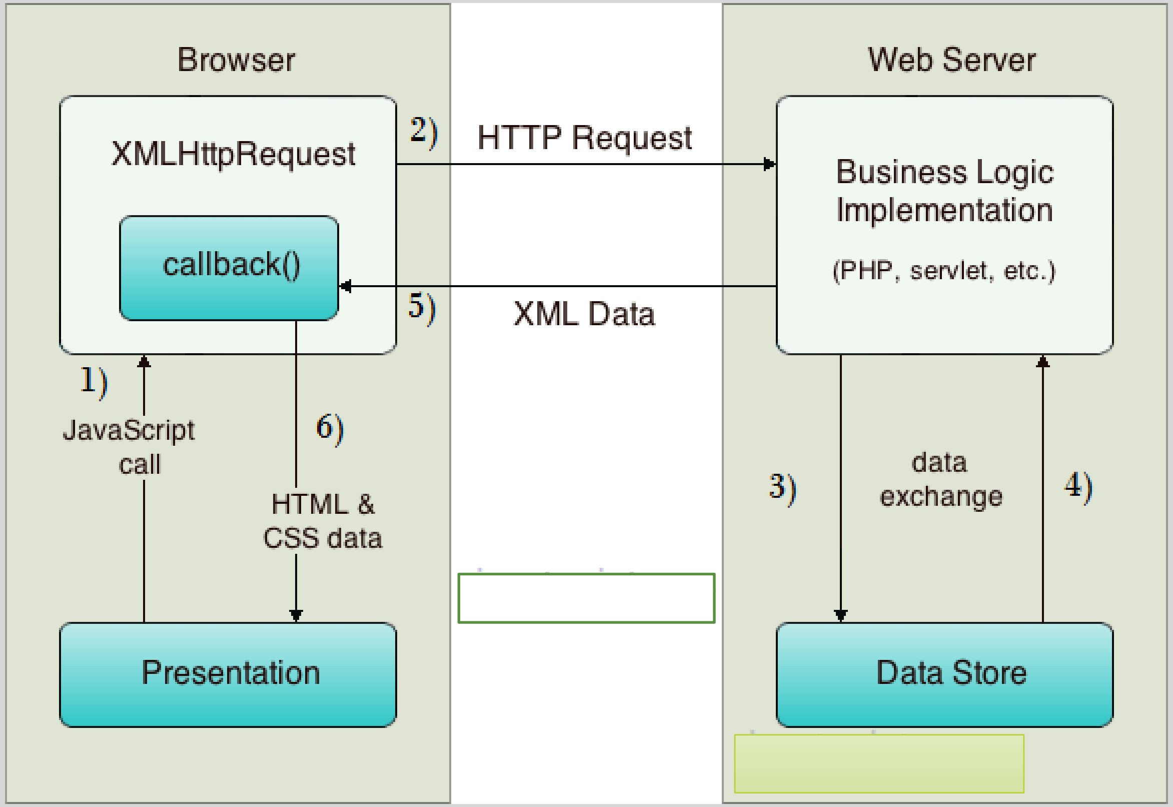
Property	Description
onReadyStateChange	It is called whenever readystate attribute changes. It must not be used with synchronous requests.
readyState	represents the state of the request. It ranges from 0 to 4. **0** UNOPENED open() is not called. **1** OPENED open is called but send() is not called. **2** HEADERS_RECEIVED send() is called, and headers and status are available. **3** LOADING Downloading data; responseText holds the data. **4** DONE The operation is completed fully.
reponseText	returns response as text.
responseXML	returns response as XML

Methods of XMLHttpRequest object

Method	Description
void open(method, URL)	opens the request specifying get or post method and url.
void open(method, URL, async)	same as above but specifies asynchronous or not.
void open(method, URL, async, username, password)	same as above but specifies username and password.
void send()	sends get request.
void send(string)	send post request.
setRequestHeader(header,value)	it adds request headers.

How AJAX works?

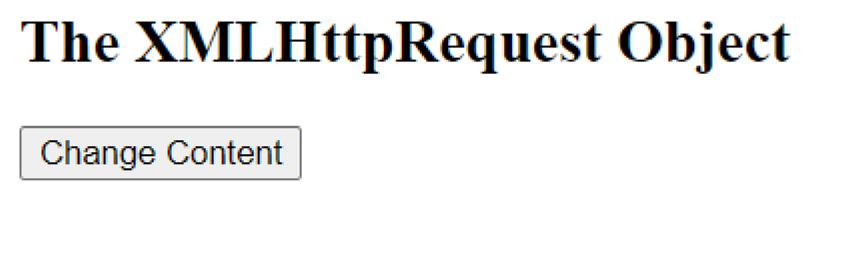
- AJAX communicates with the server using XMLHttpRequest object.



AJAX EXample

```
<!DOCTYPE html>

<html>

<body>

<div id="demo">

<h2>The XMLHttpRequest Object</h2>

<button type="button" onclick="loadDoc()">Change Content</button>

</div>
```

```
<script>
function loadDoc() {
  const xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt");
  xhttp.send();
}
</script>

</body>
</html>
```



The XMLHttpRequest Object

- **Ajax_info.txt**
- **AJAX**
- AJAX is not a programming language.
- AJAX is a technique for accessing web servers from a web page.
- AJAX stands for Asynchronous JavaScript And XML.

jQuery and AJAX

- jQuery provides several methods for AJAX functionality.
- With the jQuery AJAX methods, you can request text, HTML, XML, or JSON from a remote server using both HTTP Get and HTTP Post - And you can load the external data directly into the selected HTML elements of your web page.
- Writing regular AJAX code can be a bit tricky, because different browsers have different syntax for AJAX implementation.
- This means that you will have to write extra code to test for different browsers. However, the jQuery team has taken care of this for us, so that we can write AJAX functionality with only one single line of code.

jQuery - AJAX load() Method

- **jQuery load() Method**

The jQuery `load()` method is a simple, but powerful AJAX method.
The `load()` method loads data from a server and puts the returned data into the selected element.

- **Syntax:**
- `$(`*`selector`*`).load(`*`URL,data,callback`*`);`
- The required URL parameter specifies the URL you wish to load.
- The optional data parameter specifies a set of querystring key/value pairs to send along with the request.
- The optional callback parameter is the name of a function to be executed after the load() method is completed.

Example

-
  ```
  <h2>jQuery and AJAX is Awsome!!!</h2>
  <p id="p1">This is some text in a paragraph.</p>
  ```

-
  ```
  $("#div1").load("demo_test.txt");
  ```
- It is also possible to add a jQuery selector to the URL parameter.

-
  ```
  $("#div1").load("demo_test.txt #p1");
  ```

The optional callback parameter specifies a callback function to run when the `load()` method is completed. The callback function can have different parameters:

- `responseTxt` - contains the resulting content if the call succeeds
- `statusTxt` - contains the status of the call
- `xhr` - contains the XMLHttpRequest object

jQuery - AJAX get() and post() Methods

- The jQuery get() and post() methods are used to request data from the server with an HTTP GET or POST request.
- jQuery $.get() Method
- method requests data from the server with an HTTP GET request.
- **Syntax:**
- `$.get(`*`URL,callback`*`);`
- The required URL parameter specifies the URL you wish to request.
- The optional callback parameter is the name of a function to be executed if the request succeeds.

jQuery $.post() Method

- method requests data from the server using an HTTP POST request.
- **Syntax:**
- `$.post(URL,data,callback);`
- The required URL parameter specifies the URL you wish to request.
- The optional data parameter specifies some data to send along with the request.
- The optional callback parameter is the name of a function to be executed if the request succeeds.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $.get("demo_test.asp", function(data, status){
      alert("Data: " + data + "\nStatus: " + status);
    });
  });   });
</script>
</head>
<body>
<button>Send an HTTP GET request to a page and get the result back</button>
</body>
</html>
```