AJAX

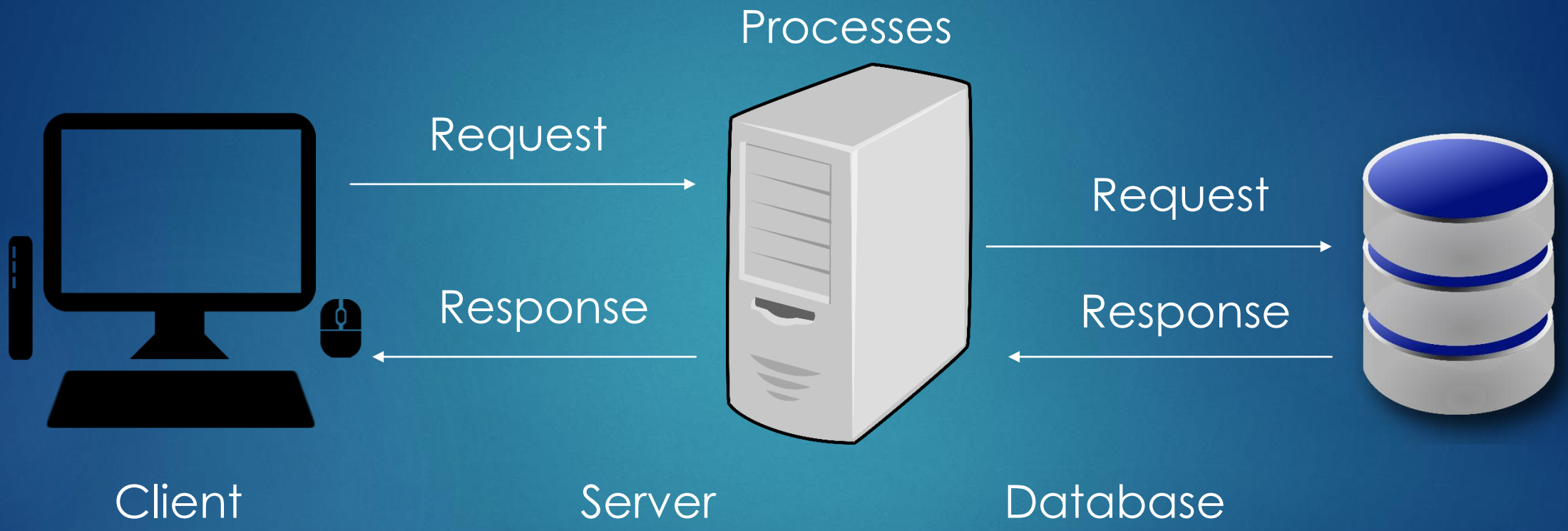
https://github.com/maunashjani/AJAX-Session

@MaunashJani

AJAX

- AJAX Web Application Model,
- How AJAX Works,
- XMLHttpRequest Object – Properties and Methods,
- Handling asynchronous requests using AJAX

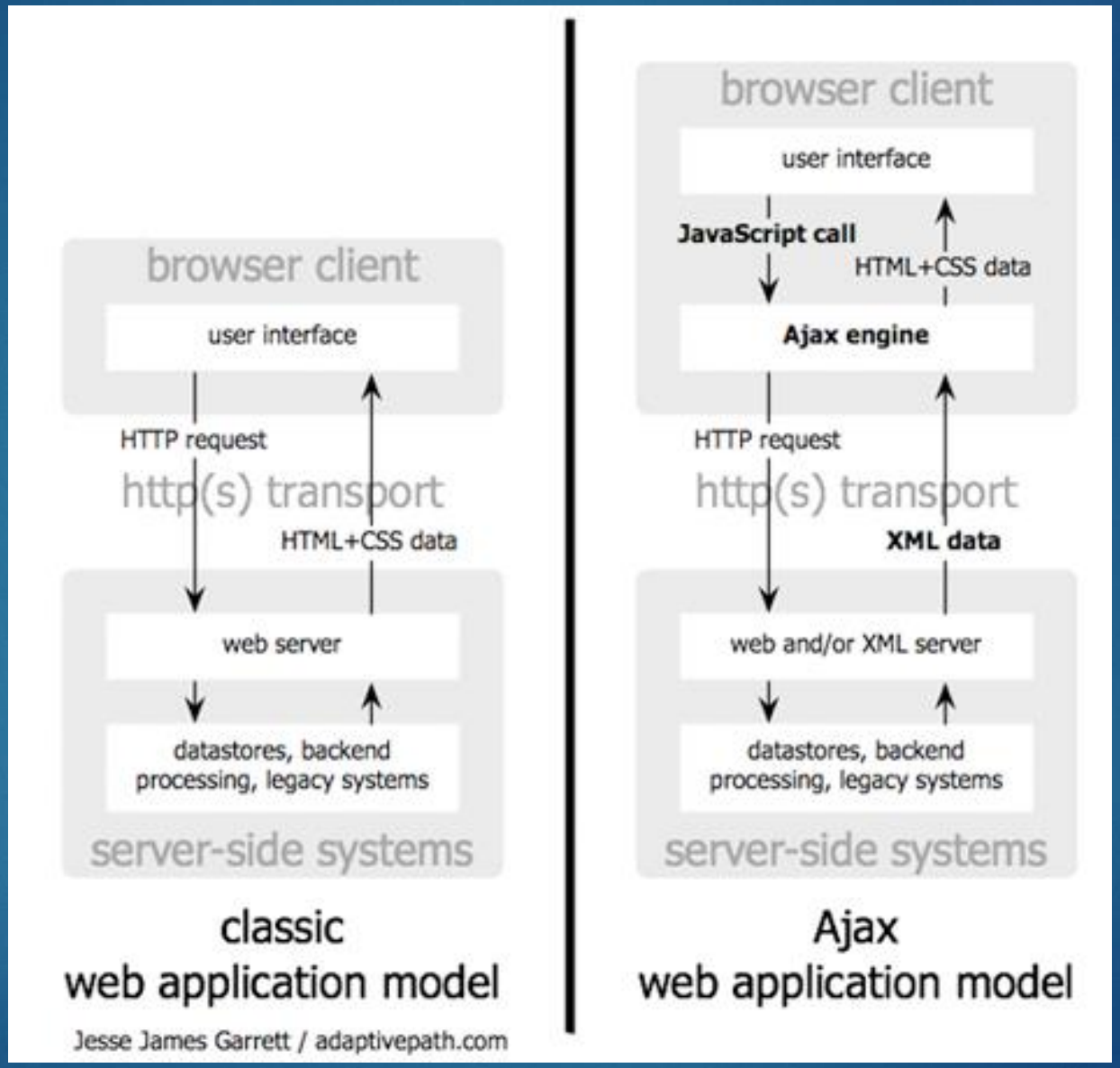
Web – Request / Response



Source: Google Images

Introduction

- AJAX is an acronym for **A**synchronous **J**avaScript **A**nd **X**ML.
- AJAX is not a programming language, but simply a development technique for creating interactive web applications.
- Ajax isn't a technology. It's really several technologies, each flourishing in its own right, coming together in powerful new ways.
- Ajax incorporates:
  - standards-based presentation using HTML and CSS;
  - dynamic display and interaction using the Document Object Model;
  - data interchange and manipulation using XML and XSLT;
  - asynchronous data retrieval using XMLHttpRequest;
  - and JavaScript binding everything together.



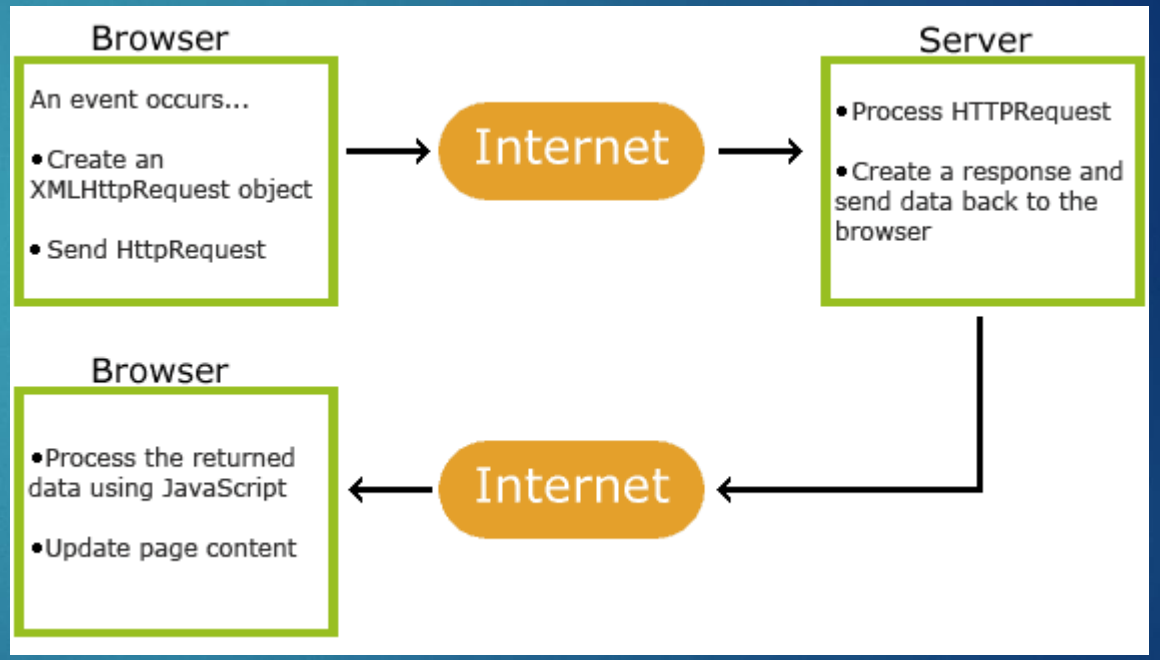
Source: http://adaptivepath.org/ideas/ajax-new-approach-web-applications/

Examples

- Searching in real time with live searches
- Getting the answer with autocomplete
- Chatting with friends
- Dragging and dropping with Ajax
- Gaming with Ajax
- Getting instant login feedback
- Google Maps

How AJAX works?

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



Source: w3schools.com

AJAX - The XMLHttpRequest Object

- The keystone of AJAX is 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.
- Consists of:
  - Methods – send() , abort(), getResponseHeader(), etc.
  - Properties – onreadystatechange, readyState, responseText, status, etc.

Getting Started

Create an XMLHttpRequest Object

- Syntax for creating an XMLHttpRequest object:
- *variable* = new XMLHttpRequest();
- Example:
- var xhttp = new XMLHttpRequest();

- For older versions of IE – 5, 6

```
if (window.XMLHttpRequest)
{
    // code for modern browsers
    xhttp = new XMLHttpRequest();
 }
else
{
    // code for old IE browsers
    xhttp = new ActiveXObject("Microsoft.XMLHTTP");
}
```

Send a Request To a Server

- To send a request to a server, we use the open() and send() methods of the XMLHttpRequest object:
- xhttp.open("GET", “demo.txt", true);
  xhttp.send();

Method	Description
open(method, url, async)	Specifies the type of request method: the type of request: GET or POST url: the server (file) location async: true (asynchronous) or false (synchronous)
send()	Sends the request to the server (used for GET)
send(string)	Sends the request to the server (used for POST)

The onreadystatechange Property

- After making a request, you will receive a response back.
- At this stage, you need to tell the XMLHttp request object which JavaScript function will handle the response,
- by setting the **onreadystatechange** property of the object and naming it after the function to call when the request changes state,
- **Syntax:** xhttp.onreadystatechange = nameOfTheFunction;
- **Example:**

```
xhttp.onreadystatechange = function(){
    // Process the server response here.
};
```

Handling the server response

- First, the function needs to check the request's state.
- If the state has the value of XMLHttpRequest.DONE (corresponding to 4),
- that means that the full server response was received and it's OK for you to continue processing it.
- If the status has the value of 200 it means the response
- Example:

```
if (xhttp.readyState === 4 &&  xhttp.status == 200 ) {
    // Everything is good, the response was received.
} else {
    // Not ready yet.
}
```

Property	Description
onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
status	200: "OK" 403: "Forbidden" 404: "Page not found" For a complete list go to the Http Messages Reference
statusText	Returns the status-text (e.g. "OK" or "Not Found")

Server Response

- Properties

Property	Description
responseText	get the response data as a string
responseXML	get the response data as XML data

- Methods

Method	Description
getResponseHeader()	Returns specific header information from the server resource
getAllResponseHeaders()	Returns all the header information from the server resource

Examples - Handle asynchronous requests using AJAX

1. Simple - Button click
2. Mouseover
3. Request Header Information – All
4. Request Header Information - Specific
5. PHP AJAX Simple
6. PHP AJAX Advanced

THANK YOU

@MaunashJani