JavaScript Object Notation (JSON)
JSON stands for JavaScript Object Notation. JSON has become a widely accepted and popular format for data due to it's platform neutral nature, lightweight format, and it's ability to convert directly to native JavaScript Objects. JSON is being used everywhere from Web APIs, to noSQL databases, to server side language libraries and client side frameworks.For example, the sender may be a Java application running on Unix and the receiver could be a Visual Basic program running on Windows. In this instance one would require a standardized format to send the data because the applications could not communicate directly with each other. Even if they could, using a protocol such as TCP/IP or FTP, the data structures would have be to be defined somewhere. That's where the data-interchange format comes in. When XML came onto the scene in the late nineties, it was hailed as a godsend by many business managers because it was intuitive to look at and based on the well-known HTML language. Developers, on the other hand, were less impressed. It turns out that parsing XML was not nearly as simple as one might think!
JSON is text, written with JavaScript object notation.

```
{"employees":[
{ "firstName":"John", "lastName":"Doe" },
{ "firstName":"Anna", "lastName":"Smith" },
{ "firstName":"Peter", "lastName":"Jones" }
]}
```

Why JSON is Better Than XML

XML is much more difficult to parse than JSON.
JSON is parsed into a ready-to-use JavaScript object.

For AJAX applications, JSON is faster and easier than XML:

Using XML

- Fetch an XML document
- Use the XML DOM to loop through the document
- Extract values and store in variables

Using JSON

- Fetch a JSON string
- JSON.Parse the JSON string

JSON (in Javascript) is a string! People often assume all Javascript objects are JSON and that JSON is a Javascript object. This is incorrect.

In Javascript var x = {x:y} is **not JSON**, this is a **Javascript object**. The two are not the same thing. The JSON equivalent (represented in the Javascript language) would be var x = '{"x":"y"}'. x is an object of type **string** not an object in it's own right. To turn this into a fully fledged Javascript object you must first parse it, var x = JSON.parse('{"x":"y"}');, x is

now an object but this is not JSON anymore.

JSON Syntax Rules

JSON syntax is derived from JavaScript object notation syntax:

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

Object Syntax

JSON objects are surrounded by curly braces {}.JSON objects are written in key/value pairs.

Keys must be strings, and values must be a valid JSON data type (string, number, object, array, boolean or null).Keys and values are separated by a colon.Each key/value pair is separated by a comma.

You can access the object values by using dot (.) notation:

```
Methon-I        myObj = { "name":"John", "age":30, "car":null };  ///obj creation
                        x = myObj.name;      //accesssing

Method—II               var JSONObj = new Object();
```

JSON Array

JSON array represents ordered list of values. JSON array can store multiple values. It can store string, number, boolean or object in JSON array.In JSON array, values must be separated by comma.

The **[** (square bracket) represents JSON array.

Eg ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]

JSON Array of Objects

Let's see a simple JSON array example having 4 objects.

```
{"employees":[

  {"name":"Ram", "email":"ram@gmail.com", "age":23},

 {"name":"Shyam", "email":"shyam23@gmail.com", "age":28},
]}
```

JSON Multidimensional Array

We can store array inside JSON array, it is known as array of arrays or multidimensional array.

- [
- ["a", "b", "c"],
- ["m", "n", "o"],
- ["x", "y", "z"]]
  JSON FILE extension .json
- **As a simple data format with no document-based configurations, merely parsing a JSON document is not open to security misconfiguration. However, given that JSON is designed to be a subset of JavaScript, it is tempting to parse a JSON document by simply passing it to a JavaScript engine (e.g., the *eval* method). Certain implementations of JSON exchanges in JavaScript work this way and can open up an application to vulnerabilities, e.g., through JSONP.

Cross-Site Request Forgery (CSRF)

JSON Injection

The attacker can inject partial JSON structs to the application and manipulate the JSON output which may lead from denial of service to unauthorized access to system resources

- **XMLHttpRequest Object**

The XMLHttpRequest object can be used to request data from a web server.

The XMLHttpRequest object is **a developers dream**, because you can:

- Update a web page without reloading the page
- Request data from a server - after the page has loaded
- Receive data from a server - after the page has loaded
- Send data to a server - in the background

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.

In modern web-development XMLHttpRequest is used for three reasons:

- Historical reasons: we need to support existing scripts with XMLHttpRequest.
- We need to support old browsers, and don't want polyfills (e.g. to keep scripts tiny).
- We need something that fetch can't do yet, e.g. to track upload progress.

Despite its name, XMLHttpRequest can be used to retrieve any type of data, not just XML.

The **onreadystatechange** property specifies a function to be executed every time the status of the XMLHttpRequest object changes:

```
xhttp.onreadystatechange = function()
```

The **responseText** property returns the server response as a text string.
The text string can be used to update a web page:

```
document.getElementById("demo").innerHTML = xhttp.responseText;
```

JSON & Client Side Frameworks ------jQuery and JSON,,,AngularJS

server-side web frameworks.-------Django (Python),Flask (Python),Express (Node.js/JavaScript)

Ruby on Rails (Ruby),Laravel (PHP),,Mojolicious (Perl),,,Spring Boot (Java),,,ASP.NET

When receiving data from a web server, the data is always a string.

Parse the data with JSON.parse(), and the data becomes a JavaScript object.

Use the JavaScript function JSON.parse() to convert text into a JavaScript object:

```
var obj = JSON.parse('{ "name":"John", "age":30, "city":"New York"}');
```

***

- **JSON From the Server**

You can request JSON from the server by using an AJAX request

As long as the response from the server is written in JSON format, you can parse the string into a JavaScript object.

- **Example**

Use the XMLHttpRequest to get data from the server:

```
var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
  if (this.readyState == 4 && this.status == 200) {
    var myObj = JSON.parse(this.responseText);
    document.getElementById("demo").innerHTML = myObj.name;
  }
};
xmlhttp.open("GET", "json_demo.txt", true);
```

xmlhttp.send();

What is Ajax?
AJAX = Asynchronous JavaScript and XML.
AJAX is a technique for creating fast and dynamic web pages.

AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

Classic web pages, (which do not use AJAX) must reload the entire page if the content should change.

Ajax is an acronym for Asynchronous Javascript and XML. It is used to communicate with the server without refreshing the web page and thus increasing the user experience and better performance.

Prerequisites:
There are no such pre-requisites required to understand the latter portion of the article. Only the basic knowledge of HTML, CSS, and Javascript are good to go.

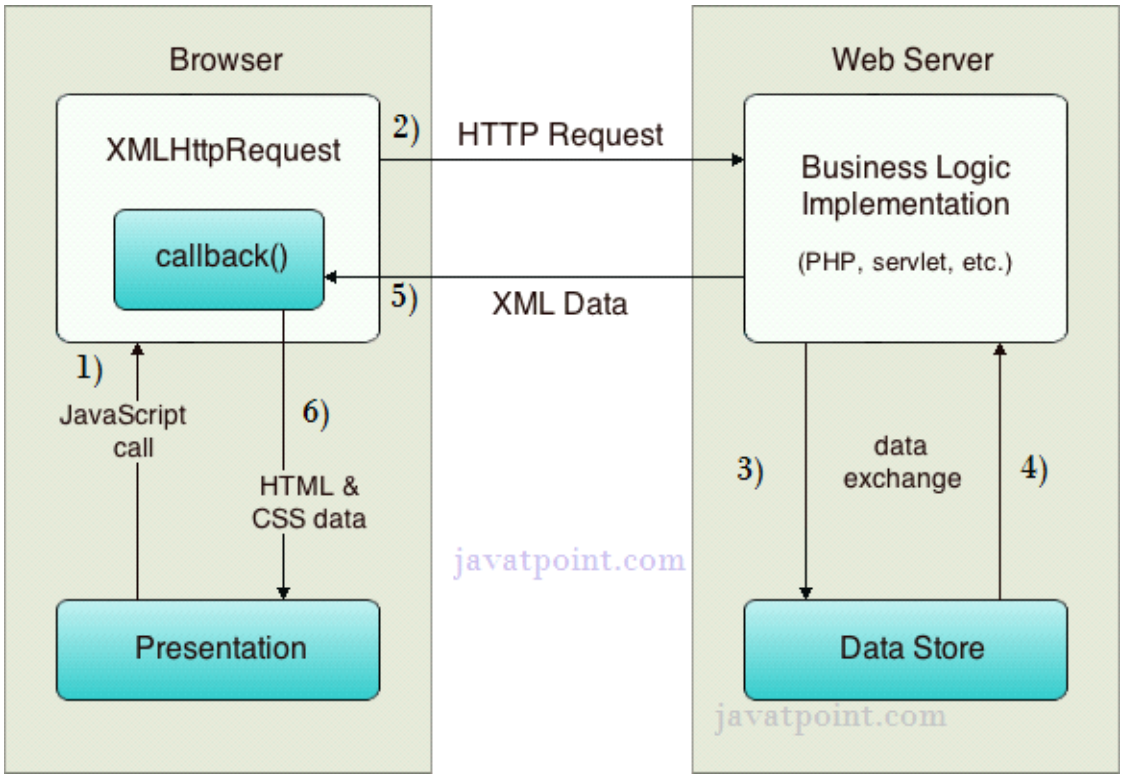
How does it work?
First, let us understand what does asynchronous actually mean. There are two types of requests synchronous as well as asynchronous. Synchronous requests are the one which follows sequentially i.e if one process is going on and in the same time another process wants to be executed, it will not be allowed that means the only one process at a time will be executed. This is not good because in this type most of the time CPU remains idle such as during I/O operation in the process which are the order of magnitude slower than the CPU processing the instructions. Thus to make the full utilization of the CPU and other resources use asynchronous calls. For more information visit this link. Why the word javascript is present here. Actually, the requests are made through the use of javascript functions. Now the term XML which is used to create **XMLHttpRequest object**.
Thus the summary of the above explanation is that Ajax allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. Now discuss the important part and its implementation. For implementing Ajax, only be aware of XMLHttpRequest object. Now, what actually it is. It is an object used to exchange data with the server behind the scenes. Try to remember the paradigm of OOP which says that object communicates through calling methods (or in general sense message passing). The same case applied here as well. Usually, create this object and use it to call the methods which result in effective communication. All modern browsers support the XMLHttpRequest object.

The open() method prepares the request before sending it to the server. The send method

is used to send the request to the server.
Sending the parameter through getting or POST request.

- **How AJAX works?**

AJAX communicates with the server using XMLHttpRequest object. Let's try to understand the flow of ajax or how ajax works by the image displayed below.



As you can see in the above example, XMLHttpRequest object plays a important role.

- User sends a request from the UI and a javascript call goes to XMLHttpRequest object.
- HTTP Request is sent to the server by XMLHttpRequest object.
- Server interacts with the database using JSP, PHP, Servlet, ASP.net etc.
- Data is retrieved.
- Server sends XML data or JSON data to the XMLHttpRequest callback function.
- HTML and CSS data is displayed on the browser.

Advantages:

- Speed is enhanced as there is no need to reload the page again.
- AJAX make asynchronous calls to a web server, this means client browsers avoid waiting for all the data to arrive before starting of rendering.
- Form validation can be done successfully through it.
- Bandwidth utilization – It saves memory when the data is fetched from the same

page.

- More interactive.

Disadvantages:

- Ajax is dependent on Javascript. If there is some Javascript problem with the browser or in the OS, Ajax will not support.
- Ajax can be problematic in Search engines as it uses Javascript for most of its parts.
- Source code written in AJAX is easily human readable. There will be some security issues in Ajax.
- Debugging is difficult.
- Problem with browser back button when using AJAX enabled pages. Ajax frameworks---jQueHYPERLINK "https://en.wikipedia.org/wiki/JQuery"ry,

```
req.open("GET", "abc.php?x=25", true);
req.send();
```

Ajax frameworks---jQuery,ASP.NET AJAX, AngularJS ,AJAX.OOP, Bindows,

- **JPSpan**

JPSpan is an Ajax framework built with the intention of integrating client-side JavaScript with serverside code written in PHP. PHP has been around since the mid nineties and has grown from simple beginnings to a full-fledged object-oriented language that can run on both Windows and UNIX/Linux platforms. The main advantages of using PHP over other platforms, such as Java or .NET, are that it is smaller, much simpler to install, and more lightweight, needing only a fraction of the memory of the Java runtime or the .NET CLR.

- **How It Works**

JPSpan works by analyzing a PHP class using reflection. It then emits a number of JavaScript methods that accept the same arguments as the methods of the class on the server. When one of these methods is called, it uses a cross-browser library to instantiate the appropriate XmlHttpRequest for the platform it is running on and posts the data to the server. The response is then read and passed to a function on the client for further use within the page.

JPSpan is an open source Ajax framework for PHP (available for download at www.sourceforge.net/projects/jpspan) that creates JavaScript wrappers for PHP objects and methods. It accomplishes this task by using reflection to inspect the composition of an object. Reflection is a common feature of object-oriented languages, allowing developers to determine information about the makeup of an object at runtime. By using reflection, JPSpan is able to create appropriate JavaScript wrappers for each method of the object. These JavaScript functions handle the cross-browser creation of XHR objects as well as the

instantiation of objects on the server and data type conversion of arguments and results.

The **DWR (Direct Web Remoting) project** is an open source solution under the Apache license for the developer who wants to use AJAX and XMLHttpRequest in an easy way. It has a set of JavaScript functions that remove the complexity from calling methods in a Java object running on the application server from the HTML page. It handles parameters of different types and helps keep the HTML code readable.

Dynamic HTML, or **DHTML**, is a collection of technologies used together to create interactive and animated websitHYPERLINK "https://en.wikipedia.org/wiki/Website"es[1] by using a combination of a static markup language (such as HTML), a client-side scripting language (such as JavaScript), a presentation definition language (such as CSS), and the Document Object Model (DOM) the loading function loadJSON() is used asynchronously to upload JSON data.

jQuery - AJAX load() Method-----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*);
jQuery $.get() Method
The $.get() method requests data from the server with an HTTP GET request.

Syntax: $.get(*URL,callback*);

***jQuery $.post() Method
The $.post() method requests data from the server using an HTTP POST request.
Syntax: $.post(URL,data,callback);

UI scripts provide a way to package client-side JavaScript into a reusable form, similar to how script includes store server-side JavaScript. Administrators can create UI scripts and run them from client scripts and other client-side script objects and from HTML code.
UI scripts are not supported for mobile.

***__Bootstrap__ is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design

templates for typography, forms, buttons, navigation and other interface components. Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

Why **Bootstrap?**

- Bundled Javascript plugins

  Extensive list of components

  Base styling for most HTML elements

Bootstrap's grid system allows up to 12 columns across the page.Bootstrap's grid system uses a series of containers, rows, and columns to layout and align content. It's built with flexbox and is fully responsive. Below is an example and an in-depth look at how the grid comes together.

The Bootstrap grid system has four classes:

- xs (for phones - screens less than 768px wide)
- sm (for tablets - screens equal to or greater than 768px wide)
- md (for small laptops - screens equal to or greater than 992px wide)
- lg (for laptops and desktops - screens equal to or greater than 1200px wide)

Grid System Rules

Some Bootstrap grid system rules:

- Rows must be placed within a .container (fixed-width) or .container-fluid (full-width) for proper alignment and padding
- Use rows to create horizontal groups of columns
- Content should be placed within columns, and only columns may be immediate children of rows
- Predefined classes like .row and .col-sm-4 are available for quickly making grid layouts
- Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and last column via negative margin on .rows
- Grid columns are created by specifying the number of 12 available columns you wish to span. For example, three equal columns would use three .col-sm-4
- Column widths are in percentage, so they are always fluid and sized relative to their parent element

- Three Equal Columns

.col-sm-4 .col-sm-4 .col-sm-4

* Three Unequal Columns
.col-sm-3. Col-sm-6. Col-sm-3

Offsetting ColumnsMove columns to the right using .col-md-offset-* classes. These classes increase the left margin of a column by * columns:
Change the order of the grid columns with .col-md-push-* and .col-md-pull-* classes:

Jumbotron

Lightweight, flexible component for showcasing hero unit style content.A lightweight, flexible component that can optionally extend the entire viewport to showcase key marketing messages on your site.A **jumbotron** is displayed as a grey box with rounded corners. It also enlarges the font sizes of the text inside it. Tip: Inside a **jumbotron** you can put nearly any valid HTML, including other **Bootstrap** elements/classes.

- **Wells**

The .well class adds a rounded border around an element with a gray background color and some padding:

- **Example** ** Basic Well **

Badges scale to match the size of the immediate parent element by using relative font sizing and em units.

<h2>Example heading New</h2>