UNIT – 2 (part-2)
AJAX & JSON

BCA SEM – 2

WEB PROGRAMMING

Code : CS-09

Presented By : Dhruvita Savaliya

Topics :

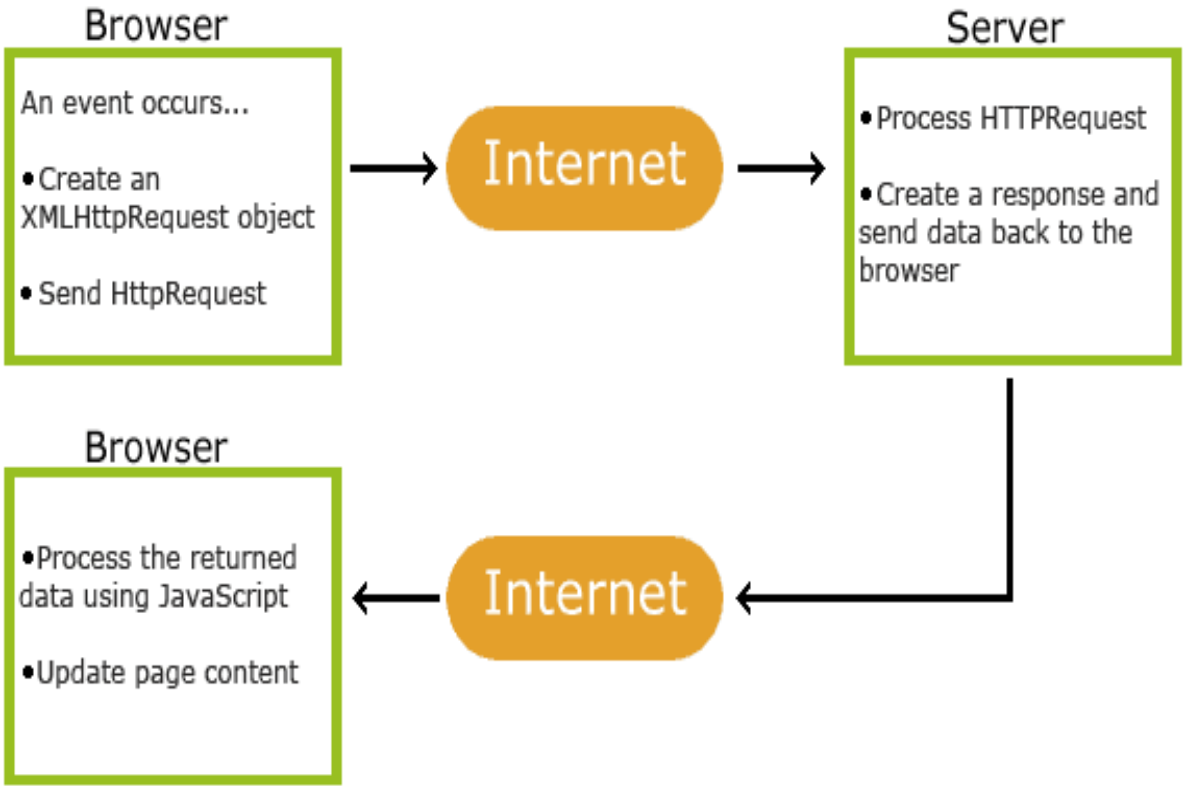
- What is AJAX
- PHP with AJAX
- MySql with AJAX
- What is JQuery AJAX
- JQuery AJAX with PHP

- Introduction to JSON
- Installation & Configuration
- Resource Types
- JsonSerializable
- JSON Functions :
  - json_decode
  - json_encode

What is AJAX :

- AJAX stands for **A**synchronous **Ja**vaScript and **X**ML.
- It is used to communicate with the server without refreshing the web page and thus increasing the user experience and better performance.
- AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS and Java Script.
- 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.
- Examples of applications using AJAX: Google Maps, Gmail, Youtube, and Facebook tabs.
- Conventional web application transmit information to and from the sever using synchronous requests. This means you fill out a form, hit submit, and get directed to a new page with new information from the server.
- With AJAX when submit is pressed, JavaScript will make a request to the server, interpret the results and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.

- **How AJAX Works :**



- AJAX is based on internet standards, and uses a combination of:
- XMLHttpRequest object (to exchange data asynchronously with a server)
- JavaScript/DOM (to display/interact with the information)
- CSS (to style the data)
- XML (often used as the format for transferring data)
- AJAX applications are browser- and platform-independent!

Why use AJAX?

- It allows developing rich interactive web applications just like desktop applications.
- Validation can be performed done as the user fills in a form without submitting it. This can be achieved using auto completion. The words that the user types in are submitted to the server for processing. The server responds with keywords that match what the user entered.
- It can be used to populate a dropdown box depending on the value of another dropdown box
- Data can be retrieved from the server and only a certain part of a page updated without loading the whole page. This is very useful for web page parts that load things like
  - Tweets
  - Comments
  - Users visiting the site etc.

- **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.

- **Example :**
- **File Name : Ajax_xml.html**

```
<!DOCTYPE html>
<html>
<head>
<script>
function showUser(str) {
 if (str=="") {
  document.getElementById("txtHint").innerHTML="";
  return;
 }
 var xmlhttp=new XMLHttpRequest();
 xmlhttp.onreadystatechange=function() {
  if (this.readyState==4 && this.status==200) {
   document.getElementById("txtHint").innerHTML=this.resp
   onseText;
  }
 }
xmlhttp.open("GET","getuser_xml.php?q="+str,true);
 xmlhttp.send();
}
</script>
</head>
<body>
<form>
<select name="users" onchange="showUser(this.value)">
<option value="">Select a person:</option>
<option value="1">Peter Griffin</option>
<option value="2">Lois Griffin</option>
<option value="3">Joseph Swanson</option>
<option value="4">Glenn Quagmire</option>
</select>
</form>
<br>
<div id="txtHint"><b>Person info will be listed
    here.</b></div>

</body>
</html>
```

- **File Name :getuser_xml.php**

```
<?php
    include("conn.php");
    $q = $_GET['q'];
    $data=mysqli_query($con,"select *from
    d where id=$q");
    if(mysqli_num_rows($data)>0)
    {
            echo "<table border=1>";
            echo "<tr>
                    <th>id</th>
                    <th>name</th>
                    <th>age</th>
            </tr>";

            while($d=mysqli_fetch_array($da
    ta))
            {
                    echo "<tr>";
                    echo
"<td>".$d['id']."</td>";
                    echo
"<td>".$d['name']."</td>";
                    echo
"<td>".$d['age']."</td>";
                    echo "</tr>";
            }
            echo "</table>";

            mysqli_close($con);
}
else
{
            echo "data not found";
}
?>
```

JSON :

- Introduction to JSON
- Resource Types (with array , object)
- JsonSerializable
- JSON Functions : json_decode, json_encode

- **What is JSON :**
- JSON stands for **J**ava**S**cript **O**bject **N**otation.
- JSON is a lightweight data-interchange format.
- JSON is used to store and exchange the data .
- The official media type for the JSON is application/json and to save those file **.json** extension.
- JSON is plain text written in JavaScript object notation.
- JSON is easy to read and write than XML(Extensible Markup Language).
- It is a format for structuring data. This format is used by different web applications to communicate with each other.
- JSON supports array, object, string, number and values.
- JSON does not support any comments and namespaces.
- **NOTE :** The JSON syntax is derived from JavaScript object notation, but the JSON format is text only.

- **Advantage of JSON :**
- Human readable format
- Language Independent
- Support all popular programming languages
- Easy to organized and access
- It is light-weight
- **Disadvantage of JSON :**
- You can not transfer video , audio , images or any other binary information using JSON.
- **Most common Use of JSON :**
- JSON is used when you want to share data between two API's (Application Programming Interface) , both request and response.
- JavaScript can easily parse JSON string that's why we use the JSON

- **Recourse Type of JSON / JSON data Format / Syntax :**
  1. **recourse type as array**
  2. **recourse type as Object**

- Data is represented in name/value pairs.
- Curly braces hold objects and each name is followed by ':'(colon), the name/value pairs are separated by , (comma).
- Square brackets hold arrays and values are separated by ,(comma).
- you must give key in double quotes it is not optional .
- **Syntax :**

  "firstName": ”some_text"
- **Example of Resource type as Array :**

```
{
    “students”:
    [
        {“name”:”Ram”,”age”:23},
        {“name”:”Shyam”,”age”:25}
    ]
}
```

- **JavaScript Object / Literals VS JSON :**
- **JavaScript :**

  var student={name:"hello",age:23}; //in key name you can write it in single quotes (' ') and also write it in double quotes (" ") and it optional.

  alert(student.name);

- **JSON as resource type <u>Object</u> :**

  var student={"name":"hello","age":23}

  alert(student.name);

- **Data type in JSON :**

Sr.No.	Type & Description
1	**Number** double- precision floating-point format in JavaScript
2	**String** double-quoted Unicode with backslash escaping
3	**Boolean** true or false
4	**Array** an ordered sequence of values
5	**Value** it can be a string, a number, true or false, null etc
6	**Object** an unordered collection of key:value pairs
7	**Whitespace** can be used between any pair of tokens
8	**null** Empty

- **Example of all DataTypes :**

```
{
   "name":"Rasmus Ledorf",
   "age":55,
   "married":true,
   "kids":,
   "hobbies":["devlopment","music"],
   "vehicle":{
                  {"type":"car","name":"innova"},
                  {"type":"bike","name":"honda"}
              }
};
```

JSON Serialization :

- JsonSerializable::jsonSerialize — Specify data which should be serialized to JSON. It is an interface of JSON serializable.
- **Syntax :**

  abstract public mixed JsonSerializable::jsonSerialize(void)
- The function has no parameter and return data which can be serialized by json_encode(), which is a value of any type other than a resource.
- **Parameters :**
- This function has no parameters.
- **Return Values :**
- Returns data which can be serialized by json_encode(), which is a value of any type other than a resource.

- **Example :**

```
<?php
class demo implements JsonSerializable {
    public function __construct(array $arr) {
            $this->array = $arr;
    }
    public function jsonSerialize() {
            return $this->array;
    }
}
// Declare an array
$arr = ['a' => 'fy', 'b' => 'sy','c'=>'ty'];
echo("Elements after converting to JSON convertible form<br>");
echo json_encode(new demo($arr));
?>
```

- **Output :**

```
Elements after converting to JSON convertible form
    {"a":"fy","b":"sy","c":"ty"}
```

JSON Functions

json_encode	Returns the JSON representation of a value.
json_decode	Decodes a JSON string.

- **Encoding JSON in PHP (json_encode) :**
- PHP json_encode() function is used for encoding JSON in PHP. This function returns the JSON representation of a value on success or FALSE on failure.
- **Syntax :**

string json_encode ($value [, $options = 0])

- **Parameters :**

value – The value being encoded. This function only works with UTF-8 encoded data.

options – This optional value is a bitmask consisting of JSON_HEX_QUOT, JSON_HEX_TAG, JSON_HEX_AMP, JSON_HEX_APOS, JSON_NUMERIC_CHECK, JSON_PRETTY_PRINT, JSON_UNESCAPED_SLASHES, JSON_FORCE_OBJECT.

- **Example :**

```
<?php
    $arr = array('a' => 1, 'b' => 2, 'c' => 3, 'd' => 4, 'e' => 5);
     echo json_encode($arr);
?>
```

- **Output :**

```
{"a":1,"b":2,"c":3,"d":4,"e":5}
```

- **Decoding JSON in PHP (json_decode) :**
- PHP json_decode() function is used for decoding JSON in PHP. This function returns the value decoded from json to appropriate PHP type.
- **Syntax :**

mixed json_decode ($json [,$assoc = false [, $depth = 512 [, $options = 0]]])

- **Paramaters :**

json_string – It is an encoded string which must be UTF-8 encoded data.

assoc – It is a boolean type parameter, when set to TRUE, returned objects will be converted into associative arrays.

depth – It is an integer type parameter which specifies recursion depth

options – It is an integer type bitmask of JSON decode, JSON_BIGINT_AS_STRING is supported.

- **Example :**

```
<?php
   $json =
   '{"a":1,"b":2,"c":3,"d":4,"e":5}';
    var_dump(json_decode($json));
?>
```

- **Output :**

```
object(stdClass)#1 (5) { ["a"] =>
   int(1) ["b"] => int(2) ["c"] =>
   int(3) ["d"] => int(4) ["e"] =>
   int(5) }
```

Questions for Exam :

- What is AJAX ?
- What is JSON ?
- What is JSON Serialization ?
- Explain Json_encode()
- Explain Json_decode()