AJAX

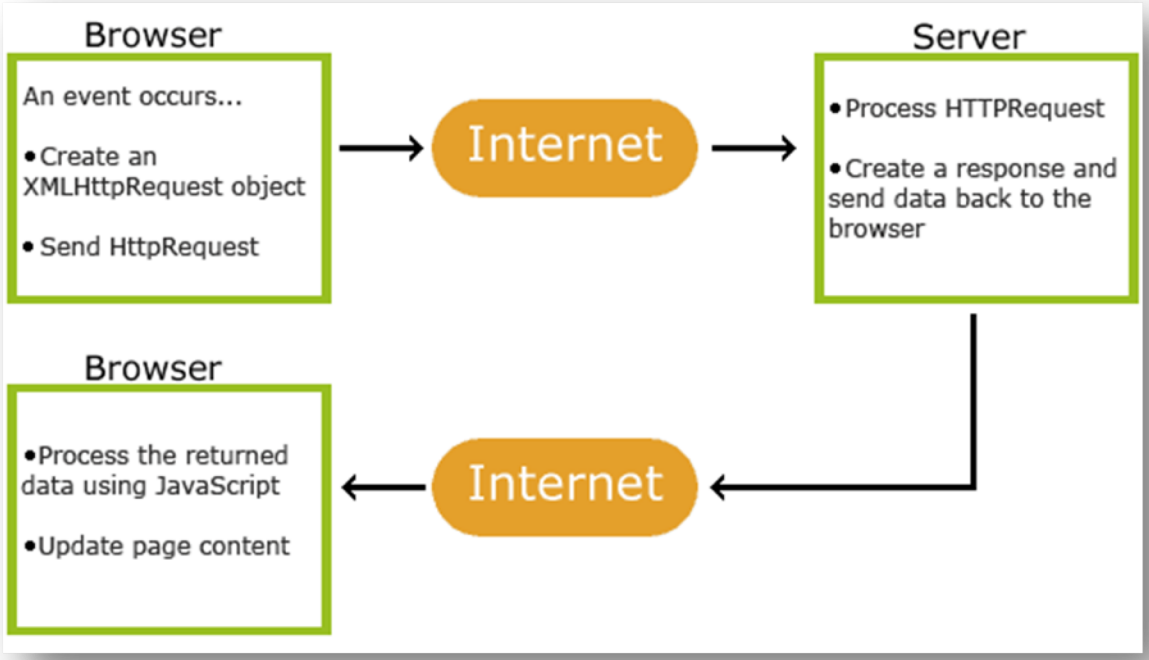
AJAX

- AJAX stands for Asynchronous JavaScript And XML
- AJAX is about updating parts of a web page, without reloading the whole page.
- Ajax or Asynchronous JavaScript and XML enables the programmer to execute a server-side script without refreshing the page.
- 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.
- Examples of applications using AJAX: Google Maps, Gmail, YouTube, and Facebook.

How AJAX Works

- The typical method for using Ajax is the following:
  1. A JavaScript creates an XMLHttpRequest object, initializes it with relevant information as necessary, and sends it to the server.
  2. The server responds by sending the contents of a file or the output of a server side program .
  3. When the response arrives from the server, a JavaScript function is triggered to act on the data supplied by the server.
  4. This JavaScript response function typically refreshes the display using the DOM, avoiding the requirement to reload or refresh the entire page.

How AJAX Works



The Back end

- The part of the Ajax application that resides on the web server is referred to as the “back end”.
- This back end could be simply a file that the server passes back to the client.
- The back end could be a program, written in PHP, Perl, Ruby, Python, C, or some other language that performs an operation and sends results back to the client browser.

مثال:

- في هذا المتال سيتم استخدام الاجاكس عن طريق jquery لإرسال سلسلة حرفية للبحث عن اسم الطالب
- يتم استخدام الحدث keyup وللحصول على الاحرف الموجودة في حقل البحث name وارسالها الى الخادم للبحث في المصفوفة عن اسم الطالب. كلما يقوم المستخدم بإدخال حرف يتم تنفيذ هذا الحدث لوضعها في المتغير str

```
var str = $('#name').val();
```

- تقوم الجي كويري باستخدام الاجاكس لإرسال الاحرف الى ملف getStudents.php والذي يقوم بالبحث في مصفوفة أسماء الطلبة عن الاحرف الأولى من اسم الطالب.

```
url  : 'getStudents.php',
```

- طريقة الارسال المستخدمة هي GET يعني ان البيانات التي يتم ارسالها الى هذا الملف ستكون موجدة في المصفوفة $_GET

```
type : 'GET ‘
```

- البيانات المرسلة حاليا هي موجودة في المتغير str ويتم ارسالها تحت اسم q الى ملف php

```
data : {q:str}
```

- نوع البيانات التي سترجع من ملف php ستكون على هيئة html

```
dataType : 'html’,
```

- البيانات التي سترجع من ملف البي اتش بي ستكون في المتغير data

```
.done(function(data)
```

- يتم وضع ناتج عملية البحث في txtHint الموجود في برنامج html الرئيسي

```
$("#txtHint").innerHTML = data;
```

```
<html>
<head> <script src="js/jquery.min.js"></script> </head>
<body>
    <p><b>Start typing a name in the input field below:</b></p>
    <form>
       First name: <input type="text" id="name">
    </form>
    <p>Suggestions: <span id="txtHint"></span></p>
</body>
<script>
$(document).ready(function() {
  $("#name").keyup(function()
  {
      var str = $('#name').val();
      $.ajax({
        type : 'GET',
        dataType : 'html',
        url  : 'getStudents.php',
        data : {q:str}
      })
      .done(function(data) {
        $("#txtHint").innerHTML = data;
      });
    });
});
</script>
</html>
```

المكتبة

مرجع الكائن

الحدث

استخدام الاجاكس

المتغير الذي سيرجع البيانات

getStudents.php

```
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <?php
        $students = array("Ahmad","Khalid","Samira","Jomaa","salem",
                          "Amin","Tamer","Zyad","Salma","Khaeri","Ali");
        $q = $_GET["q"];
        $hint = "";
        if ($q !== "") {
            $len = strlen($q);
            foreach ($students as $name) {
                if (stristr($q, substr($name, 0, $len))) {
                    $hint .= ", $name";
                }
            }
        }
        echo $hint === "" ? "no suggestion" : $hint;
        ?>
    </body>
</html>
```

Ajax reference

```
$.ajax({
        type: "GET",    // POST
        url: "XYZ",
        data: {
           "data":data,
        },
        dataType: "json",  //HTML

      success: function( datas, textStatus, jqXHR) {
        //if received a response from the server في حالة وجود استجابة من الخادم
        },
      error: function(jqXHR, textStatus, errorThrown){
       //If there was no response from the server في حالة عدم وجود استجابة من الخادم
        },
      beforeSend: function(jqXHR, settings){
         //capture the request before it was sent to server قبل ارسال الطلب الى الخادم
        },
    complete: function(jqXHR, textStatus){
        //this is called after the response or error functions are finished بتم استدعاءه بعد الانتهاء
        //so that we can take some action
        }

    });
```

Ajax Database Example (viewst.php)

```
<html>
<head>
  <title>Ajax Database Example</title>
  <script src="js/jquery.min.js"></script>
</head>
<body>
        <h2>Student Search</h2>
        <form method="post" action="">
          <div class="form-group">
            <label for="stname">Student Name</label>
            <input type="text" name="stname" id="stname">
          </div>
        </form>
        <table>
           <thead>
             <tr> <th>ID</th> <th>NAME</th><th>PASSWORD</th></tr>
           </thead>
           <tbody id="tablebody">

           </tbody>
        </table>
</body>
```

viewst.php ...

```
<script>
$(document).ready(function() {
    $("#stname").keyup(function()
    {
      var str = $('#stname').val();
      $.ajax({
        type : 'POST',
        dataType : 'html',
        url  : 'getstudents.php',
        data : {q:str}
      })
      .done(function(result) {
          $("#tablebody").html("result");
      });
    });
});
</script>
</html>
```

dbconfig.php

```
<?php
    $DB_host = "localhost";
    $DB_user = "net2user";
    $DB_pass = "ADGK1234!";
    $DB_name = "student3";

    try
    {
    $DB_con = new PDO('mysql:host='.$DB_host.';dbname='.$DB_name,$DB_user,
              $DB_pass, array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
              PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"));
    }
 catch(PDOException $e)
    {
        echo "problem conneting to db ".$e->getMessage();
    }
?>
```

Ajax Database Example (getstudents.php)

```
<?php
     include "dbconfig.php”; // connection to database
     $name = $_POST['q'];
     $query =
     "SELECT id,name,password from student where name LIKE '$name%'";
     try{
          $stmt = $DB_con->prepare($query);
          $stmt->execute();
         $tablerows = ‘’;
          while($row=$stmt->fetch(PDO::FETCH_ASSOC))
           {
                $tablerows .= “<tr><td>”.$row['id’].”</td>
                <td>”.$row['name’].”</td>
                <td>”.$row['password’].”</td>
                </tr>”;
           }
       }
      catch (Exception $ex)
       {
         echo $ex->getMessage();
       }
    echo $tablerows;
?>
```

JSON

- **J**ava**S**cript **O**bject **N**otation
- JSON is a syntax for storing and exchanging data.
- JSON is an easier-to-use alternative to XML.
- JSON is a lightweight data-interchange format
- JSON is language independent. JSON uses JavaScript syntax, but the JSON format is text only, just like XML. Text can be read and used as a data format by any programming language.
- JSON is "self-describing" and easy to understand

JSON Example

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

Convert JSON to JavaScript objects

- JavaScript program can use standard JavaScript functions to convert JSON data into native JavaScript objects.

Example

```
<!DOCTYPE html>
<html>
<body>
  <h2>JSON Object Creation in JavaScript</h2>
  <p id="demo"></p>
  <script>
    var text = '{"name":"Ali","street":"uni","phone":"091234567"}';
    var obj = JSON.parse(text);
    document.getElementById("demo").innerHTML =
    obj.name + "<br>" +
    obj.street + "<br>" +
    obj.phone;
  </script>
</body>
</html>
```

Thanks!