Oum El Bouaghi University
Faculty of Exact Sciences, Natural and Life Science
Department of Mathematics and Computer Science

Advanced Web programming

Introduction to AJAX (Asynchronous JavaScript And XML)

Dr. Saighi Asma

Concerned students

Faculty	Department	Level	Speciality
ESNL	MI	B3	ISSE

- Is not a technology.
- Is not a software.
- Is not a plug-in.

It involves the combined use of:

- HTML
- CSS
- DOM / JavaScript
- XMLHttpRequest (JavaScript)

AJAX

- **A**synchronous **J**avaScript **A**nd **X**ML:
  - **A**synchronous: Regarding the loading of the web page and web page sections.
  - **J**avaScript: script langage in the client side (browser).
  - XML: Tag-based langage enabling, among other functions, the structuring of data.

- Through JavaScript, it allows the **retrieval** of XML data (as well as JSON) available ona web server.

AJAX

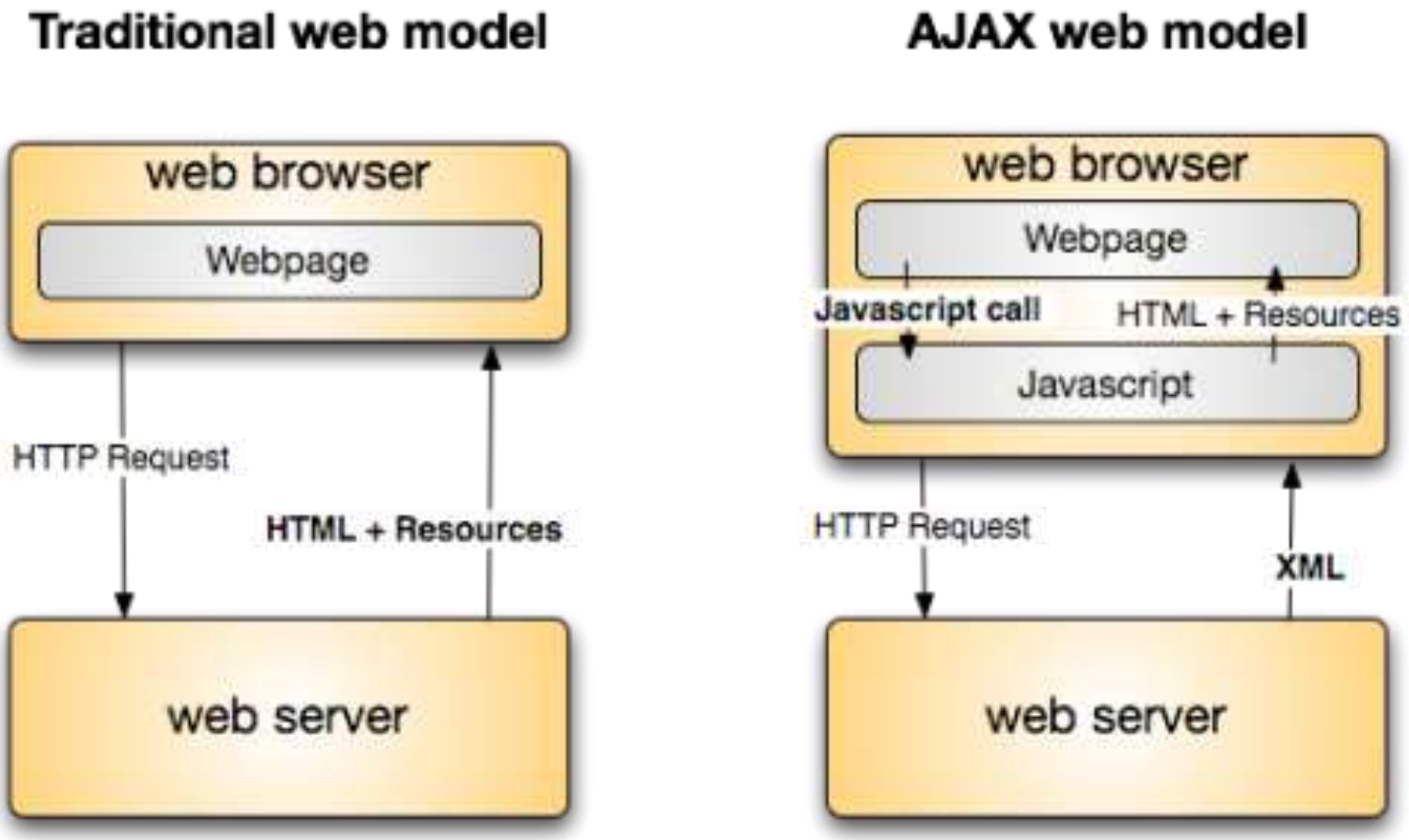
- Allows partial update of the web page displayed by the browser without having to reload the entire page.

- The principle involves processing on the client machine using JavaScript based on the data received from the server.

Rich Client

+ **Balanced distribution of processing between client and server.**
+ **Saved bandwidth.**
+ **Reduced server resource consumption.**

Classic and AJAX web application model



Thin client vs Rich client

Component	Role	
	Thin client	**Rich client**
Server	– General control of the application. – Provides static ressources: Document model, Script library, style sheets. – Dynamic data processing – Dynamic composition of the interface.	– General control of the application. – Provides static ressources: – Document model, Script library, style sheets. – Dynamic data processing.
Client	– User event management. – Dynamic interface composition.	– Delegated control based on the type of view. – User event management. – Dynamic interface composition. – Processing of received data.
Dialogue	http, (X)HTML	http, XML, JSON

Technologies used in AJAX

AJAX = DHTML (DOM + CSS + Javascript) + XmlhttpRequest

- **Javascript**
  - Object-oriented and loosely typed scripting langage.
  - Javascript Functions invoked when an event occurs on the page.
  - "Glue" for the entire functioning of AJAX.

- **DOM (Document Object Model)**
  - API for accessing srtuctured documents.
  - Represents the structure of XML and HTML documents.

- **CSS (Cascading Style Sheets)**
  - Allows a clearseparation of the content and form presentation.
  - Can be modified by Javascript code.

- **XmlhttpRequest**
  - Javascript objet that ensures asynchronous interaction with the server.

How does it work?

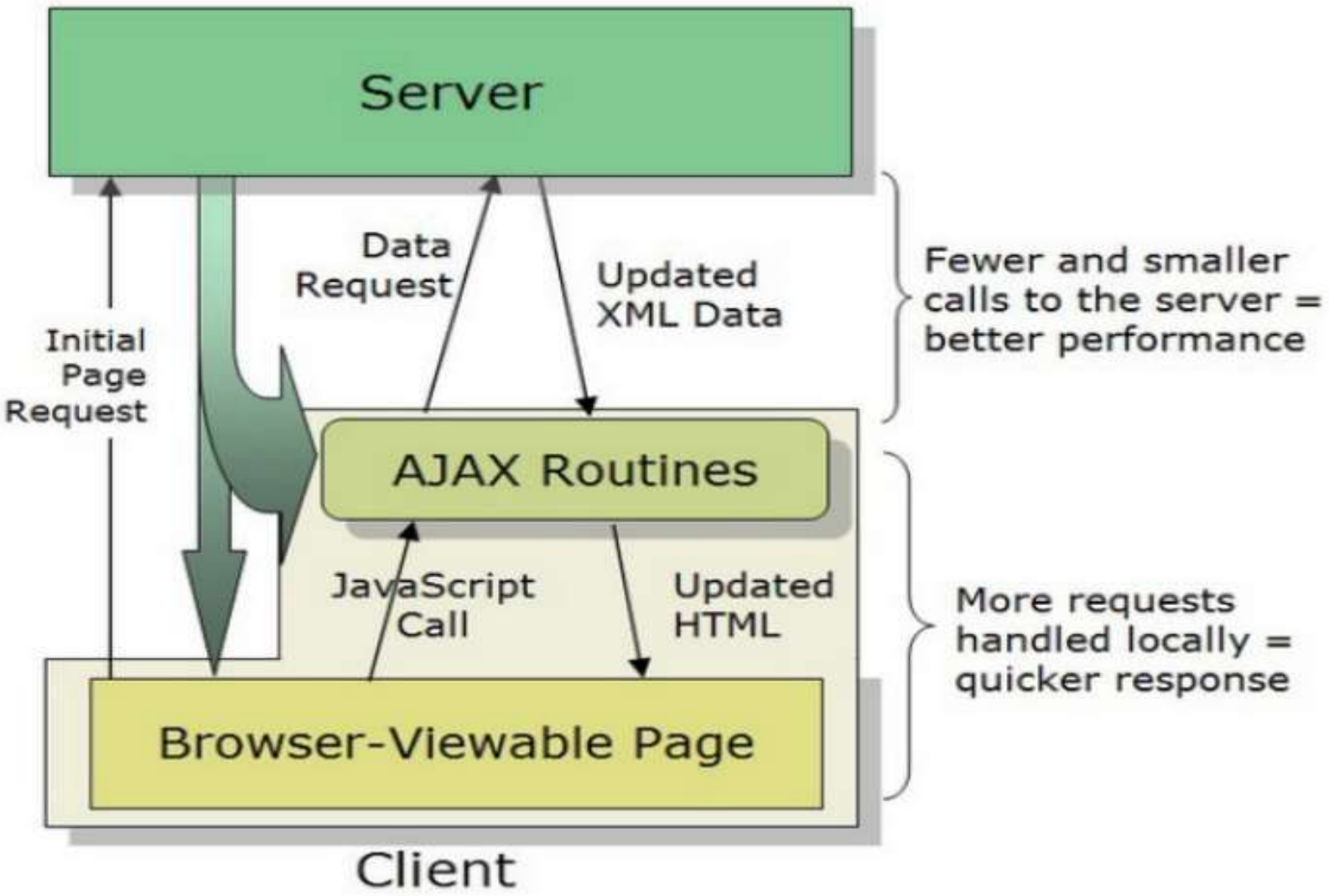
- **An AJAX call typically unfolds in three steps:**

1. **Asynchronous request** to the server within a JavaScript function triggered by an event:
   - An **XMLHttpRequest** object (Mozilla) or,
   - an *ActiveX* **XMLHTTP** control (IE) is used.

2. **Asynchronous transfer** of data in XML or JSON format.

3. **Dynamic client-side processing** for display (inclusion in the HTML document).

How does it work?



Communication between Client and Server

- **Client side:**
  - **Sending the request:**
    1. Creation of the request object (XMLHttpRequest).
    2. Specification of request elements (URL, method, HTTP headers, parameters).
    3. Association with an event handler.
    4. Sending the object.
  - **Receiving the response:**
    1. Upon any change in the request's state: check if in the *"ready"* state.
    2. Processing the received data (*Updating the user inteface*).
- **Server side:**
  - Defining actions for handling asychronous AJAX requests.
  - Specifying the format and content of the response to be sent to the client.

XMLHTTPRequest

- **JavaScript Object**
  - Microsoft :
    - ActiveXObject("Msxml2.XMLHTTP")
    - ActiveXObject("Microsoft.XMLHTTP")
  - The others:
    - XMLHttpRequest()
- **Supported by all modern browsers**
  - Mozilla™, Firefox, Safari, and Opera, including on Smartphones,
- **Communicate with the server via GET/POST HTTP standards,**
- **The object XMLHttpRequest is used in background to handle asynchronous communications.**

Attributs of the XMLHTTP object

Attribute	Description	Possible Values
Onreadystatechange	Event handler for state changes	Function
readyState	Represents the state of the XMLHttpRequest	0: uninitialized, 1:open, 2:sent, 3:receiving, 4:loaded
responseText	Returns the response as a string	String
responseXML	Returns the response as an XML document	XML document
status	Returns the status code of the HTTP response	HTTP status codes (e.g., 200 for OK)

Methods of the XMLHTTPRequest object

Method	Parameters
Open(method,url,async)	Indicates the demand type Method: demand type: GET or POST url: server (file) 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)
abort	Cancel the current HTTP request

Using XMLHTTPRequest

1. Create an object
2. Send a request to a server
   - **Method, URL**
   - **Asynchronous ?**
3. Associate a function to process the result of the request.
4. Process the result
   - Text ?
   - XML / JSON ?

Reminder

I have several friends

Mohamed, Ali, Souad are my friends.

Ana is my best friend.

```
1  <friendslist>
2    <friend>
3      <name>Mohammed</name>
4    </friend>
5    <friend>
6      <name>Ali</name>
7    </friend>
8    <friend>
9      <name>Souad</name>
10   </friend>
11   <bestfriend>
12     <name>Ana</name>
13   </bestfriend>
14 </friendslist>
```

Reminder

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE personInfo SYSTEM "personInfo.dtd">
<personInfo>
   <person>
      <name>John Doe</name>
      <age>30</age>
      <city>New York</city>
      <email>john.doe@example.com</email>
   </person>
</personInfo>
```

It looks like HTML where:

- Its own tags are used,
- The syntax is strict,
- This document cannot be displayed directly,
- Tools are used to manipulate the information in this document.

Exploiting data in XML format

```
objXML = XMLHTTPRequest.responseXML
objXML.getElementsByTagName(n)
     → Collection of nodes
Collection of nodes:
length   → number of elements
[x]   → access to the xth element
Node:
firstChild   → First child
childNodes   → Collection of children
hasChildNodes()   → Has children?
nodeValue   → Node value
nodeName   → Node name
getAttribute(a)   → Value of the attribute
```

JSON (JavaScript Object Notation)

- JSON: a generic data format that enables the transmission of structured information between different technologies.
- It uses the syntax of JavaScript objects and is highly effective technology for AJAX development.
- PHP has supported JSON since version 5.2.
- JSON is widely used in the field of smartphones due to its lightweight nature.
- Website: http://www.json.org/
- It is recommended to work with JSON in UTF-8 encoding, especially when dealing with the transport of accented characters.

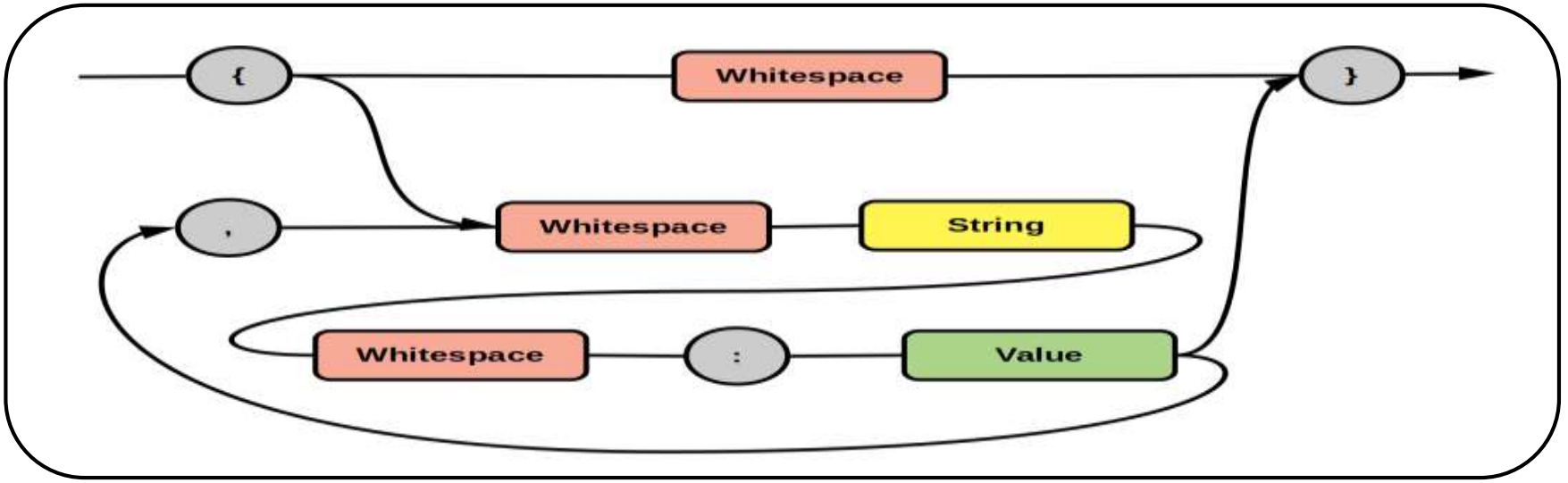
JSON

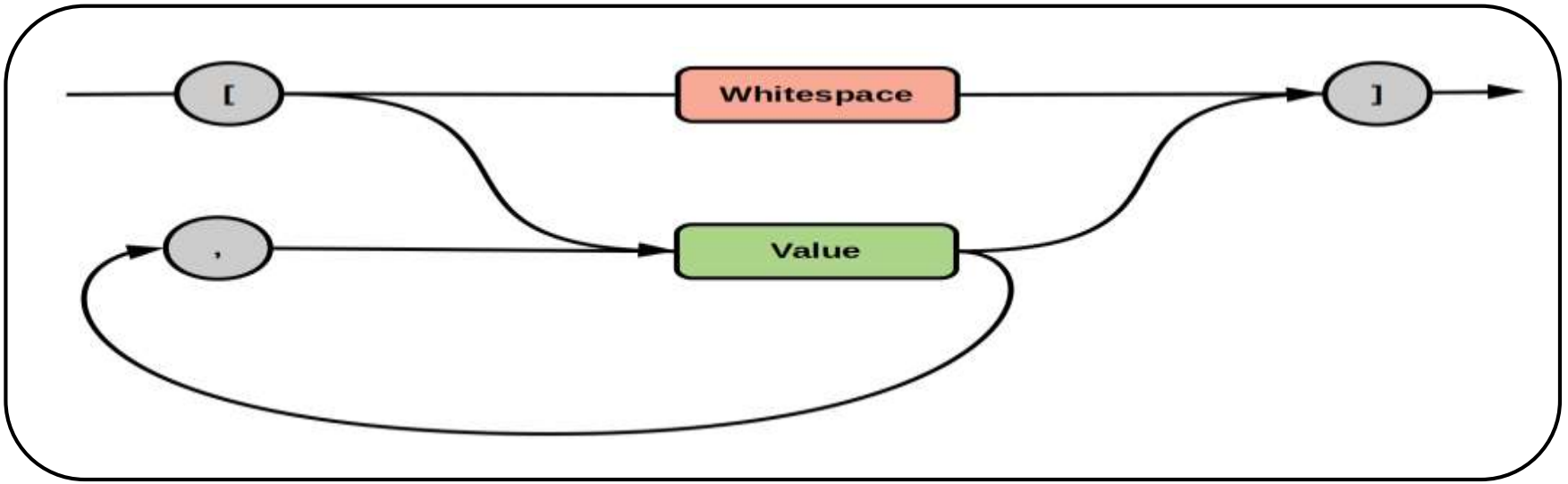
- **Two structures:**

1. **Object**
   - **{}**
   - **{string : value}**
   - **{string : value, string : value, …}**
2. **Array**
   - **[]**
   - **[value]**
   - **[value, value, …]**

- **Value :**

string|number|object|array|true|false|null


{
Whitespace
}
,
Whitespace
String
Whitespace
:
Value


[
Whitespace
]
,
Value

- **Sample JSON array of objects:**

```
[
    {
        "id": 4,
        "body": "First Item",
        "severity": 1,
        "status": 0
    },
    {
        "id": 6,
        "body": "Second Item",
        "severity": 2,
        "status": 1
    }
]
```

JSON vs XML

JSON Format	XML Format
{ "menu": { "id": "file", "value": "File", "popup": { "menuitem": [{ "value": "New", "onclick": "CreateNewDoc()" }, { "value": "Open", "onclick": "OpenDoc()" }, { "value": "Close", "onclick": "CloseDoc()" }] } } }	<menu id="file" value="File"> <popup> <menuitem value="New" onclick="CreateNewDoc()"/> <menuitem value="Open" onclick="OpenDoc()"/> <menuitem value="Close" onclick="CloseDoc()"/> </popup> </menu>

JSON vs XML

JSON Advantages:

- Processing speed.
- Ease of implementation.
- No need to parse an XML file to extract information over internet because JSON is natively recognized by JavaScript.

XML Advantages:

- XML is extensible in terms of language; one create can formats such as RSS.
- It is widely used and recognized by all programming languages.
- It is easier to read.

Read JSON file in JavaScript

- **Initial idea:**
  - JavaScript has **eval()** function that executes JavaScript code contained in a string.
  - Using eval() to parse JSON:

```
var data = '{ "company": "XYZ Corp", "employees": [{ "name": "John", "age": 30 }, {
"name": "Alice", "age": 25 }] }';
eval("b = " + data);

console.log(b);
```

- Problem: security vulnerability!
- **Never use eval().**
- **Use the JSON library.**

JSON Library

- **JSON.parse** : JSON string→ JavaScript value

```
var data = '{ "company": "XYZ Corp", "employees": [{ "name":
"John", "age": 30 }, { "name": "Alice", "age": 25 }] }';
var b = JSON.parse(data);

console.log(b);
```

→ **Resulting Object b:**

```
{
    company: 'XYZ Corp',
    employees: [
      { name: 'John', age: 30 },
      { name: 'Alice', age: 25 }
    ]
}
```

JSON Library

- **JSON.stringify** : JavaScript value → JSON string

```
var obj = { name: "John", age: 30 };
var jsonString = JSON.stringify(obj);
console.log(jsonString);
// Output: {"name":"John","age":30}
```

Exploiting data in JSON format

```
{
 "menu": "Fichier",
 "commandes": [
    {
       "title": "Nouveau",
       "action":"CreateDoc"
    },
    {
       "title": "Ouvrir",
       "action": "OpenDoc"
    },
    {
       "title": "Fermer",
       "action": "CloseDoc"
    }
  ]
}
```

Example

File,json

Exploiting data in JSON format

Example:

```
{
    "category": "Products",
    "items": [
      {
        "name": "Laptop",
        "price": 1200,
        "description": "Powerful laptop with high performance."
      },
      {
        "name": "Smartphone",
        "price": 600,
        "description": "Latest smartphone with advanced features."
      },
      {
        "name": "Headphones",
        "price": 150,
        "description": "Wireless headphones for an immersive audio experience."
      }
    ]
  }
```

- **Step 1: Creation of the Request object**

XMLHttpRequest object creation is done using ActiveXObject (IE) method or XMLHttpRequest JavaScript object.

– **Two options to test compatibility with the browsers**

```
var req = null ;
function getRequest ( ) {
    if ( window . XMLHttpRequest ) {
        req = new XMLHttpRequest ( ) ;
    } else if ( typeof ActiveXObject != " undefined") {
        req = new ActiveXObject (" Microsoft. XMLHTTP") ;
    }
    return req;
}
```

- **Step 2: Asynchronous data loading**

The operation related to the response are included in a function assigned to the **'onreadystatechange'** attribute of the created XMLHttpRequest object.

```
function processXMLData ( ) {
   if(req != null) {
        req.onreadystatechange = function() {
            // readyState 4 = requête terminée
            if (req.readyState == 4) {
                // status 200 = page requêtée trouvée
                if (req.status == 200) {
                    // ... traitement en cas de succès
                } // Page non trouvée
                else {
                    // ... traitement en cas d'échec
                }
            }
        };
   }
}
```

- **Step 3: Request execution**
  - AJAX request is executed throught two XMLHttpRequest methods 'open' and 'send‘.

JavaScript code to initiate a request for reading a document from the server:

```
function executeRequest ( ) {
   req = getRequest ( );
   processXMLData ( );
   req.open('GET', 'URL/fichier', true);
   //req.setRequestHeader is called only if the method is POST and not GET
   req.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
   req.send(null);
}
```

AJAX Advantages and disadvantages

Avantages:

- Increased client-side interactivity.
- **Faster** response.
- **Reduced client/server transactions** (retreival of scripts and style sheets only once).
- Separation of methods for information transmission and formats used to represent information.

Disadvantages:

- No reccording in the browser history of dynamically modified pages.

Solution: modifying the anchor part (#) of the URL.

- No possible indexing of pages by search engines.
- If a browser does not support JavaScript and AJAX, the page becomes unusable.

References

Zimmermann, M. Usage of the Web for Various Prototyping Scenarios. *Bettina Conradi, Doris Hausen, Fabian Hennecke, Max-Emanuel Maurer*, 19.

http://www.asp.net/ajax/
http://prototypejs.org/learn/introduction-to-ajax
https://www.codeproject.com/Articles/16272/Optimizing-Web-Applications-with-AJAX-and-Spread
https://starship-knowledge.com/tag/json