Advanced Web Technologies

Iosif Polenakis

PhD Candidate,
Department of Computer Science and Engineering,
University of Ioannina.

email: ipolenak@cs.uoi.gr, tel.: 2651008831

AJAX INTRO

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ Create asynchronous Web Applications on the client side using a set of Web development techniques.
  - ✓ Web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behavior of the existing page.
  - ✓ decoupling the data interchange layer from the presentation layer, pages and applications are allowed to change content dynamically without the need to reload the entire page
  - ✓ JSON → native to JavaScript.

AJAX INTRO

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ AJAX has come to represent a variety of Web Technologies used to implement a Web application that communicates with a server in the background, without interfering with the current page's state.
    - HTML & CSS
      - Presentation
    - Document Object Model (DOM)
      - dynamic display and interaction with data
    - JSON / XML
      - data interchange (XSLT manipulation)
    - The XMLHttpRequest Object
      - asynchronous communication
    - JavaScript
      - interconnection of the above

AJAX INTRO

❑ AJAX: Asynchronous JavaScript And XML

- ✓ AJAX has come to represent a variety of Web Technologies used to implement a Web application that communicates with a server in the background, without interfering with the current page's state.
  - HTML & CSS
    - Presentation
  - Document Object Model (DOM)
    - dynamic display and interaction with data
  - JSON / XML
    - data interchange (XSLT manipulation)
  - The XMLHttpRequest Object
    - asynchronous communication
  - JavaScript
    - interconnection of the above

eXtensible Stylesheet Language Transofrmation

AJAX INTRO

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ AJAX has come to represent a variety of Web Technologies used to implement a Web application that communicates with a server in the background, without interfering with the current page's state.
    - Standards-based presentation using HTML and CSS;
    - Dynamic display and interaction using the Document Object Model;
    - Data interchange and manipulation using JSON, XML and XSLT;
    - Asynchronous data retrieval using XMLHttpRequest;
    - JavaScript binding everything together.

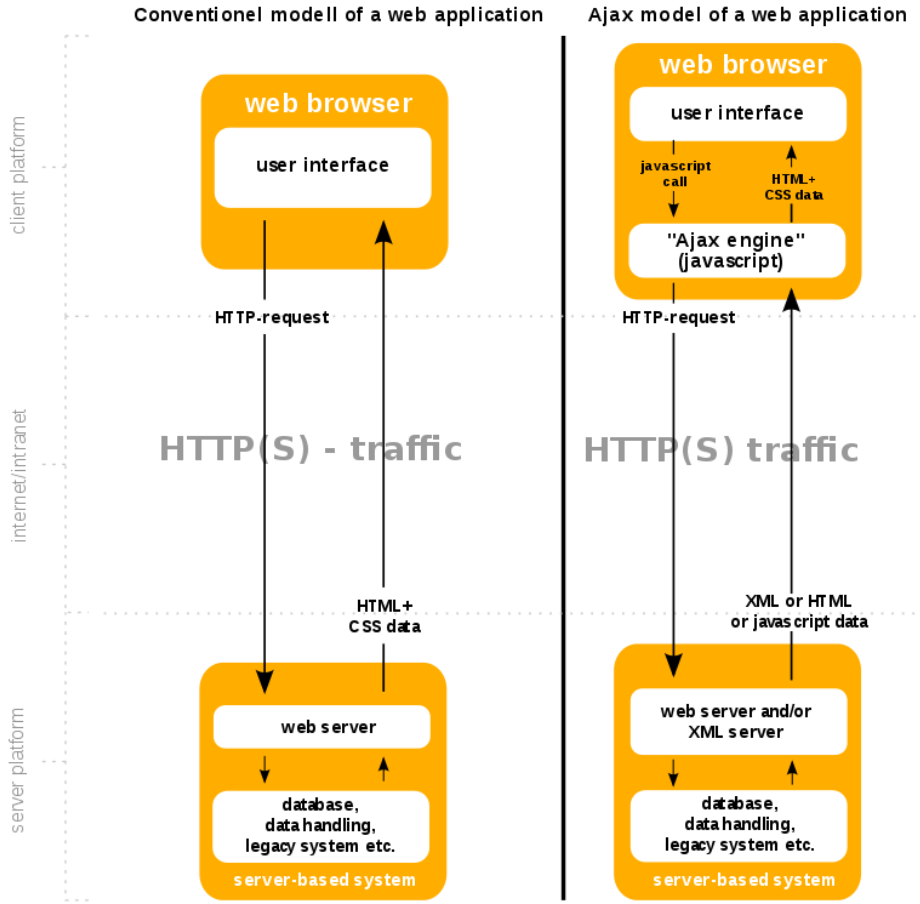
AJAX INTRO

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ AJAX is not a programming language.
  - ✓ AJAX just uses a combination of:
    - A browser built-in XMLHttpRequest object (request data from server)
    - JavaScript and HTML DOM (display/use the data)
  - ✓ AJAX applications might use XML to transport data, but it is equally common to transport data as plain text or JSON text.
  - ✓ AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes.
    - This means that it is possible to update parts of a page, without reloading the whole page.

Ajax Basics

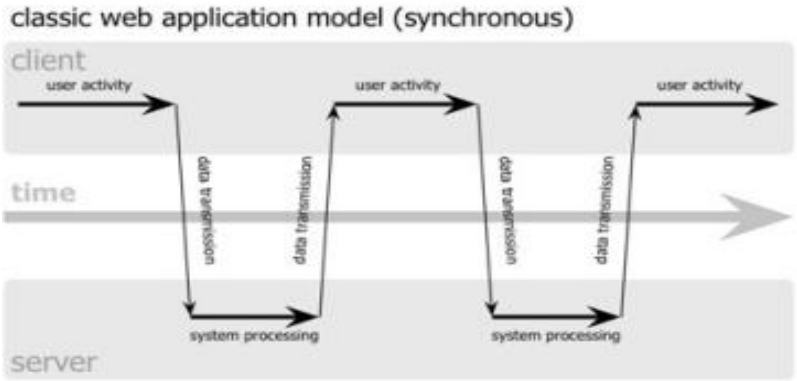
❑ AJAX: Asynchronous JavaScript And XML

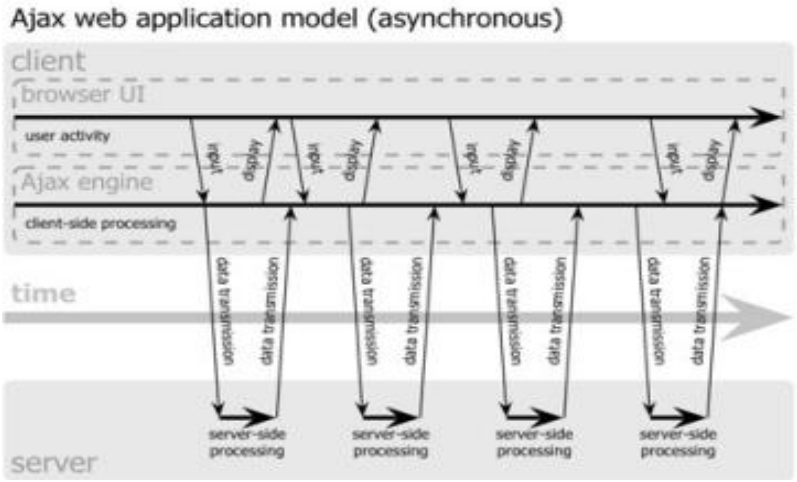
✓ AJAX – How it works...



Ajax Basics

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ AJAX – How it works...





AJAX BASICS

❑ AJAX: Asynchronous JavaScript And XML

✓ AJAX – How it works...

- An AJAX application places an intermediary between the user and the server called AJAX Engine (also known as JavaScript part of a web page).
- It seems like adding a layer to the application would make it less responsive, but the opposite is true.
- Instead of loading a webpage, at the start of the session, the browser loads an AJAX engine-written in JavaScript and usually tucked away in a hidden frame.
- This engine is responsible for both rendering the interface the user sees and communicating with the server on the user's behalf.
- The AJAX engine allows the user's interaction with the application to happen asynchronously, independent of communication with the server. So the user is never staring at a blank browser window and an hourglass icon, waiting around for the server to do something.

AJAX BASICS

❑ AJAX: Asynchronous JavaScript And XML

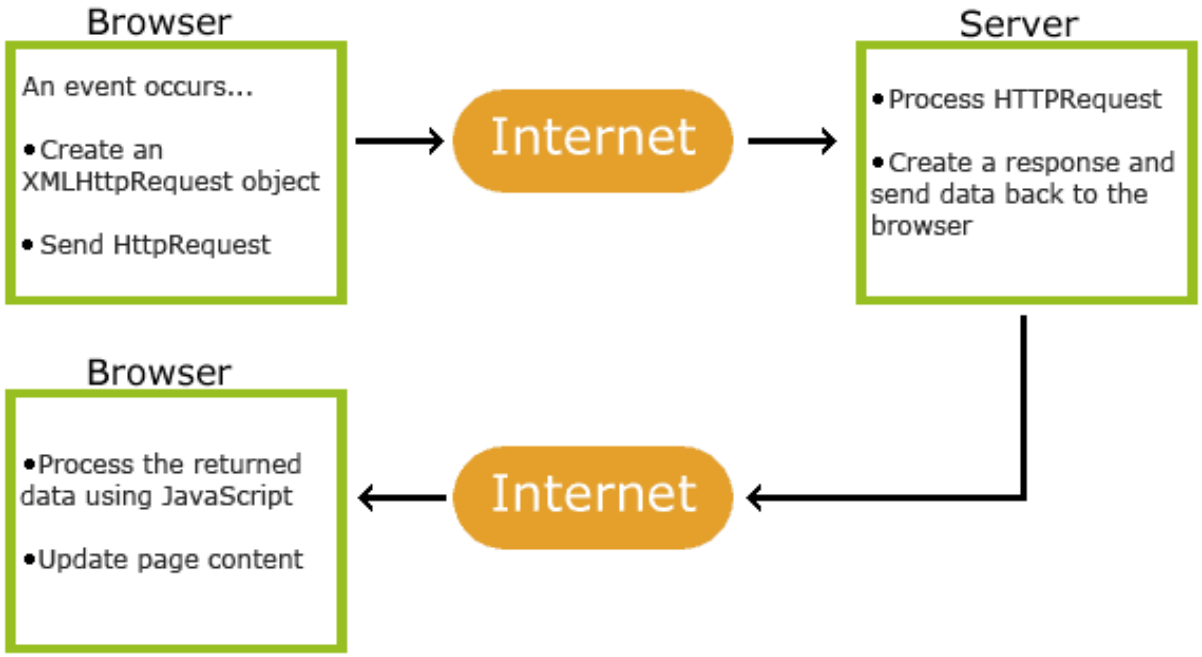
✓ AJAX – How it works...

- Every user action that normally would generate an HTTP request takes the form of a JavaScript call to the AJAX Engine instead.
- Any response to a user action that doesn't require a trip back to the server, such as simple data validation, editing data in memory, and even some navigation - the engine handles on its own.
- If the engine needs something from the server in order to respond if it's submitting data for processing, loading additional interface code, or retrieving new data, the engine makes those requests asynchronously, usually using XML, without stalling a user's interaction with the application.

AJAX BASICS

❑ AJAX: Asynchronous JavaScript And XML

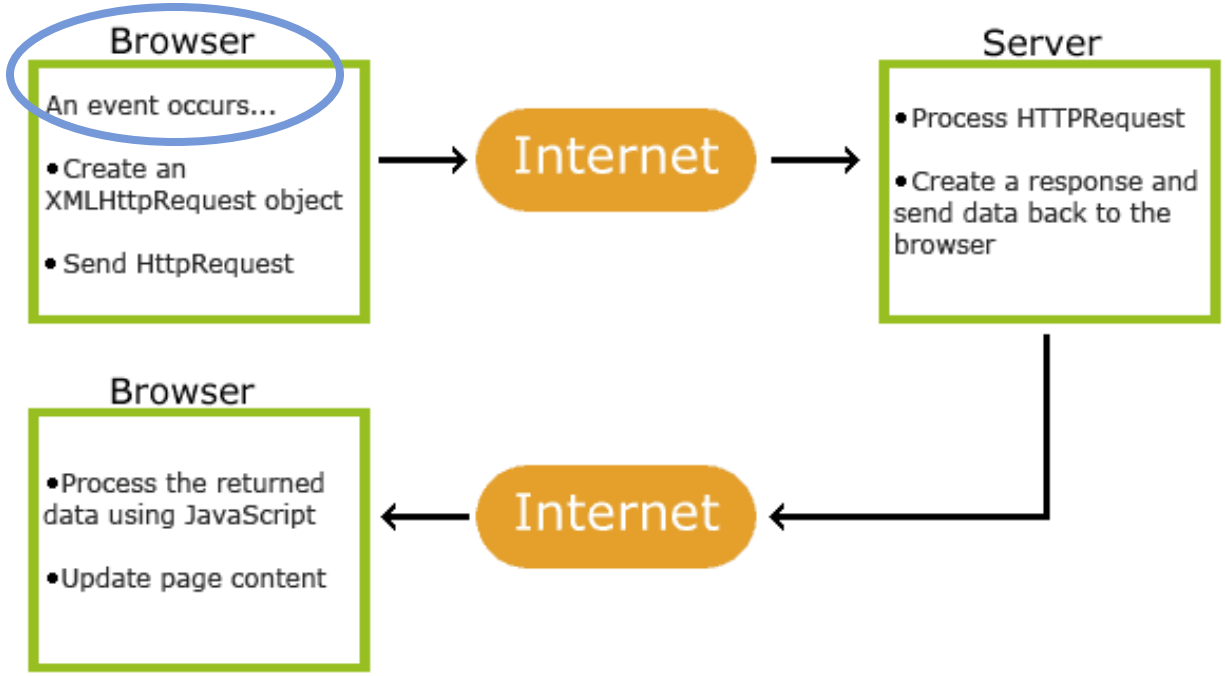
✓ AJAX – How it works...



AJAX BASICS

❑ AJAX: Asynchronous JavaScript And XML

✓ AJAX – How it works...

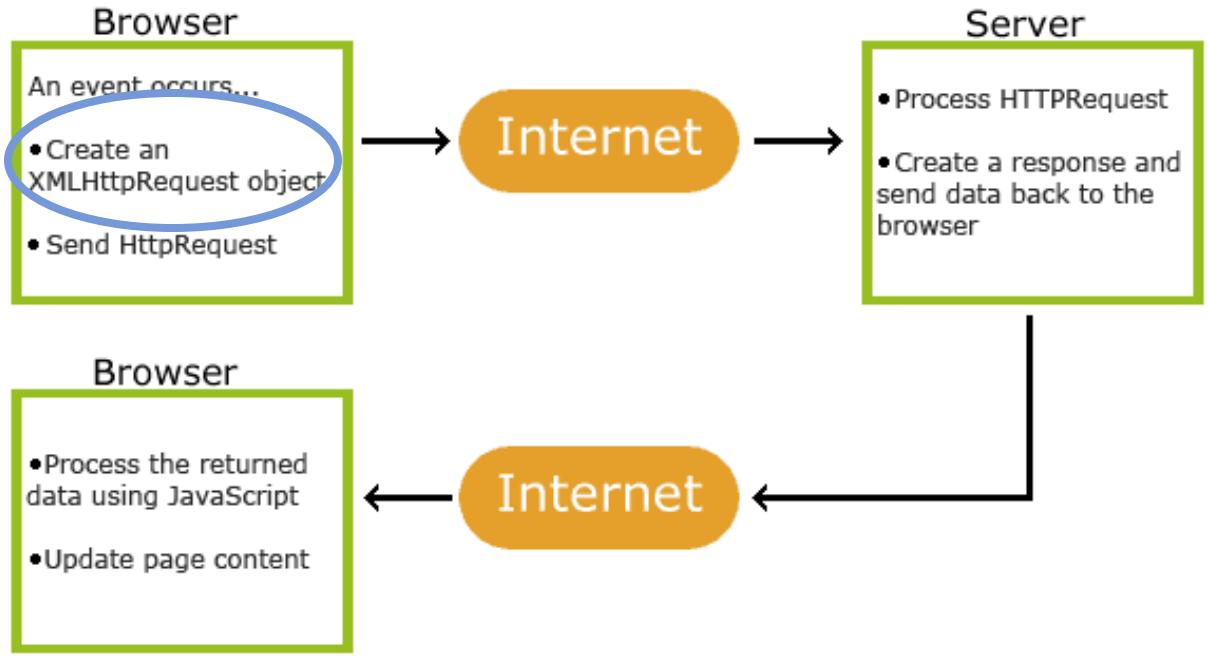


An event occurs in a web page
(the page is loaded, a button is clicked)

AJAX BASICS

❑ AJAX: Asynchronous JavaScript And XML

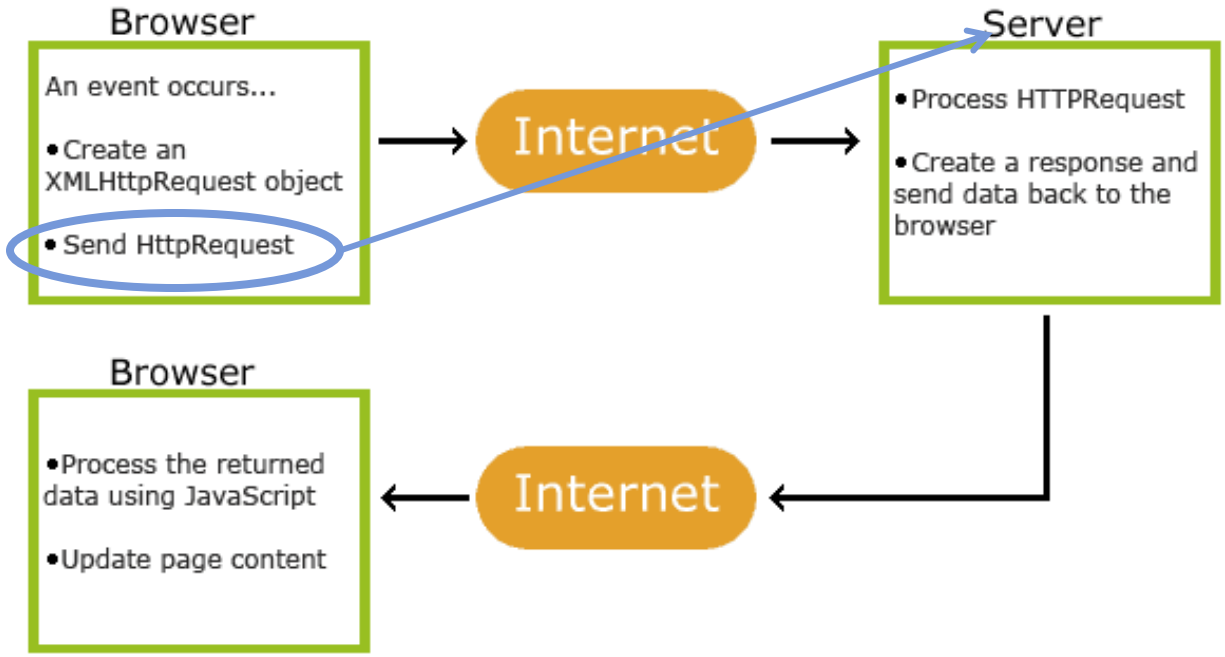
✓ AJAX – How it works...



An XMLHttpRequest object is created by JavaScript

AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ AJAX – How it works...

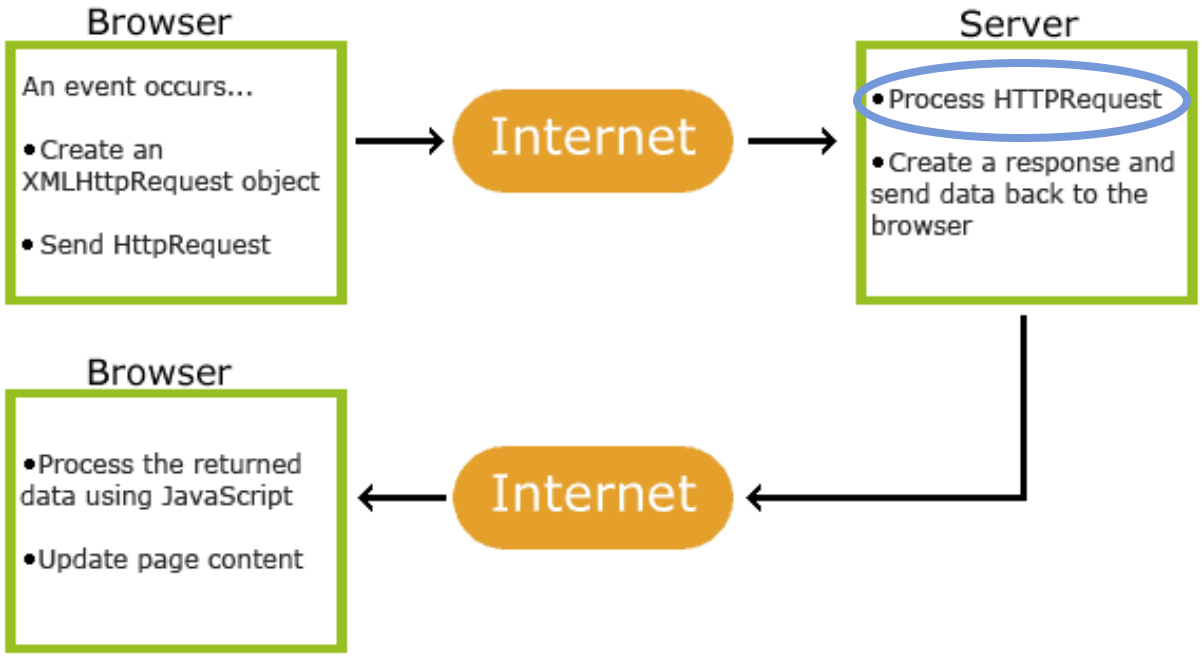


The XMLHttpRequest object sends a request to a web server

AJAX BASICS

❑ AJAX: Asynchronous JavaScript And XML

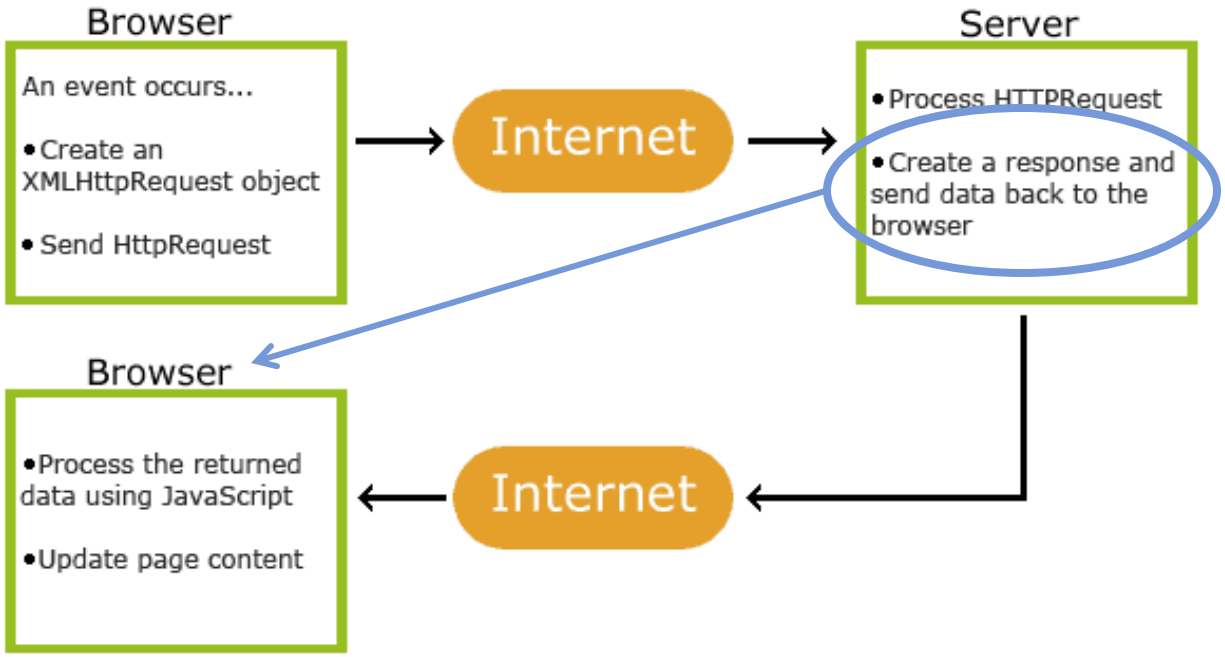
✓ AJAX – How it works...



The server processes the request

AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ AJAX – How it works...

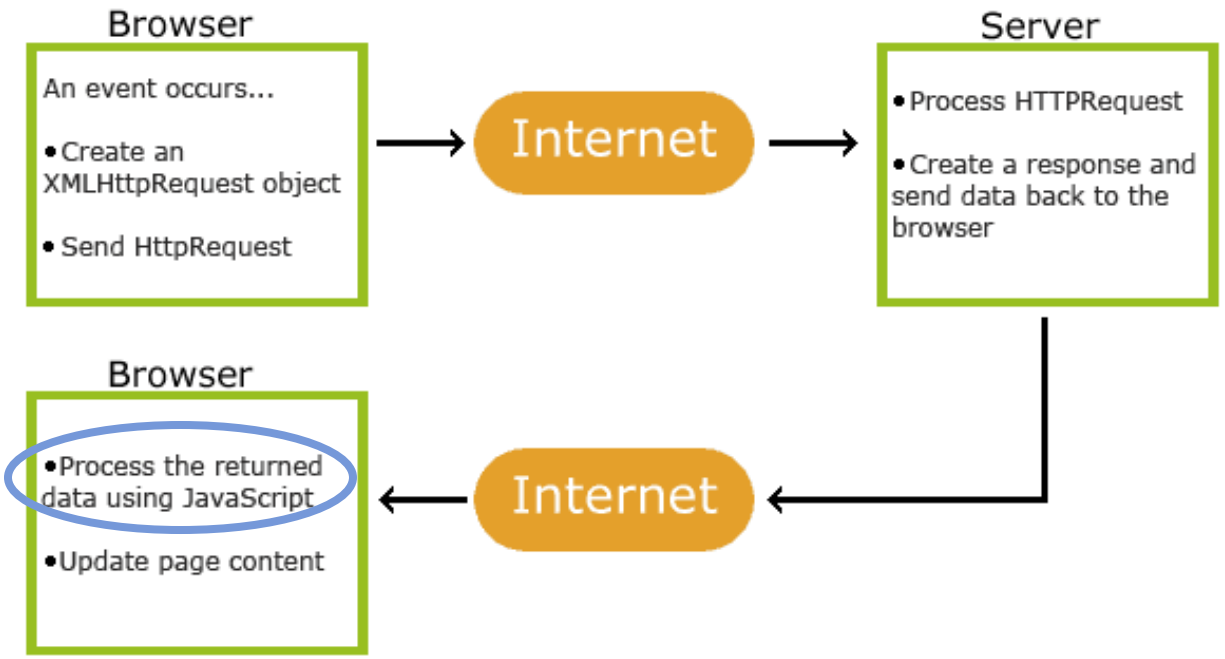


The server sends a response back to the web page

AJAX BASICS

❑ AJAX: Asynchronous JavaScript And XML

✓ AJAX – How it works...

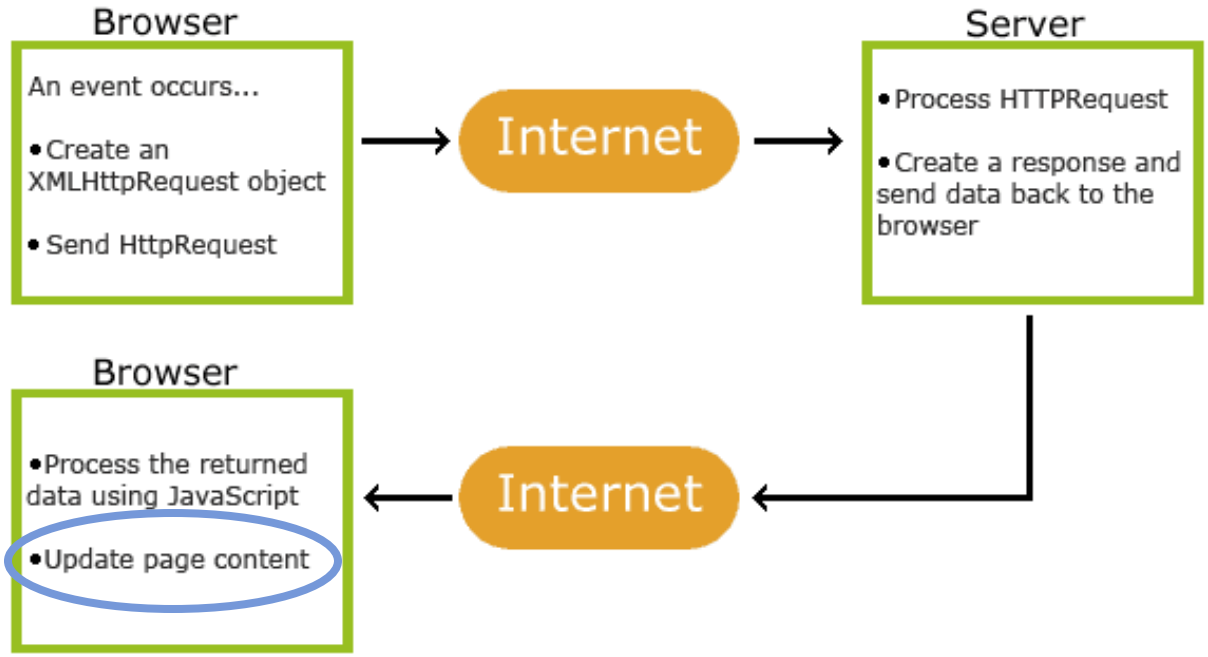


The response is read by JavaScript

AJAX BASICS

❑ AJAX: Asynchronous JavaScript And XML

✓ AJAX – How it works...



Proper action (like page update) is performed by JavaScript

AJAX BASICS

❑ AJAX: Asynchronous JavaScript And XML

✓ AJAX – How it works...

- **JavaScript**
  - JavaScript function is called when an event in a page occurs …
- **DOM**
  - API for accessing and manipulating structured documents, representing the structure of XML and HTML documents.
- **CSS**
  - Allows for a clear separation of the presentation style from the content and may be changed programmatically by JavaScript.
- **XMLHttpRequest**
  - JS Object that performs asynchronous interaction with the server.

AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ So Far So Good, but what is an XMLHttpRequest?
    - The XMLHttpRequest object can be used to request data from a web server.
    - The XMLHttpRequest object is a "tool", because you can:
      - Update a web page without reloading the page
      - Request / Receive data from a server - after the page has loaded
      - Send data to a server - in the background

AJAX BASICS

❑ AJAX: Asynchronous JavaScript And XML

✓ So Far So Good, but what is an XMLHttpRequest?

- The XMLHttpRequest object can be used to request data from a web server.
- The XMLHttpRequest object is a "tool", because you can:
  - Update a web page without reloading the page
  - Request / Receive data from a server - after the page has loaded
  - Send data to a server - in the background

AJAX BASICS

❑ AJAX: Asynchronous JavaScript And XML

✓ Use Case – Brief AJAX Example

```
<!DOCTYPE html>
<html>
<body>

<h2>Using the XMLHttpRequest Object</h2>

<div id="demo">
<button type="button" onclick="loadXMLDoc()">Change Content</button>
</div>

<script>
function loadXMLDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "xmlhttp_info.txt", true);
  xhttp.send();
}
</script>

</body>
</html>
```

AJAX BASICS

❑ AJAX: Asynchronous JavaScript And XML

✓ Use Case – Brief AJAX Example

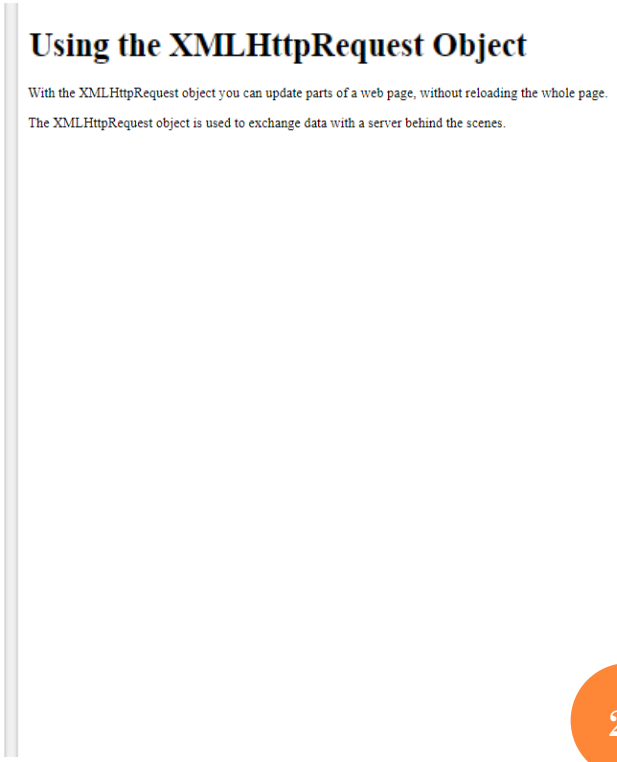
```
<!DOCTYPE html>
<html>
<body>

<h2>Using the XMLHttpRequest Object</h2>

<div id="demo">
<button type="button" onclick="loadXMLDoc()">Change Content</button>
</div>

<script>
function loadXMLDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "xmlhttp_info.txt", true);
  xhttp.send();
}
</script>

</body>
</html>
```



AJAX BASICS

❑ AJAX: Asynchronous JavaScript And XML

✓ Use Case – Brief AJAX Example

```
<!DOCTYPE html>
<html>
<body>

<div id="demo">
<h1>The XMLHttpRequest Object</h1>
<button type="button" onclick="loadDoc()">Change Content</button>
</div>

<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
</script>

</body>
</html>
```

AJAX BASICS

❑ AJAX: Asynchronous JavaScript And XML

✓ Use Case – Brief AJAX Example

```
<!DOCTYPE html>
<html>
<body>

<div id="demo">
<h1>The XMLHttpRequest Object</h1>
<button type="button" onclick="loadDoc()">Change Content</button>
</div>

<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
</script>

</body>
</html>
```

AJAX

AJAX is not a programming language.

AJAX is a technique for accessing web servers from a web page.

AJAX stands for Asynchronous JavaScript And XML.

AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ XMLHttpRequest – Properties (open)
    - For the given xhttp variable the following open properties apply:

      xhttp.open(method, URL, asynchronous) OR xhttp.open(method,URL)
      - The method is usually 'GET' or 'POST'
      - The URL is where you are sending the data
        - If using a 'GET', data is appended to the URL
        - If using a 'POST', data is added in a later step
      - If asynchronous is true, the browser does not wait for a response

AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ XMLHttpRequest – Properties (send)
    - For the given xhttp variable the following send properties apply:

      xhttp.send(null) with GET OR xhttp.send(content) with POST
      - The content has the same syntax / suffix of both cases (GET/POST)
      - POST requests are used less frequently than GET requests
      - Example POST: xhttp.send('var1=' + value1 + '&var2=' + value2);

AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ XMLHttpRequest – Properties (other methods)
    - For the given xhttp variable the following set of properties apply:
      - abort → xhttp.abort();
        - Terminates current request
      - getAllResponseHeaders → xhttp.getAllResponseHeaders();
        - Returns all labels/values as a string
      - getResponseHeader → xhttp.getResponseHeader("header");
        - Returns value of a given header
      - setRequestHeader → xhttp.setRequestHeader("label","value");
        - Sets Request Headers before sending

AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ XMLHttpRequest – Properties (other methods)
    - For the given xhttp variable the following set of properties apply:
      - onreadystatechange → xhttp.onreadystatechange;
        - Set with an JavaScript event handler that fires at each state change
      - readyState → xhttp.readyState;
        - Returns the current status of request
      - status → xhttp.status;
        - HTTP Status returned from server: 200 = OK
      - responseText → xhttp.responseText;
        - String version of data returned from the server
      - responseXML → xhttp.responseXML;
        - XML document of data returned from the server
      - statusText→ xhttp.statusText;
        - Status text returned from server

AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ XMLHttpRequest – Properties (other methods)
    - For the given xhttp variable the following set of properties apply:
    - The XMLHttpRequest object (readyState and status Properites).

```
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
       // Typical action to be performed when the document is ready:
       document.getElementById("demo").innerHTML = xhttp.responseText;
    }
};
xhttp.open("GET", "filename", true);
xhttp.send();
```

AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ XMLHttpRequest – Properties (other methods)
    - For the given xhttp variable the following set of properties apply:
    - The XMLHttpRequest object (readyState and status Properites).

```
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
     document.getElementById("demo").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
```

AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ XMLHttpRequest – Properties (other methods)
    - For the given xhttp variable the following set of properties apply:
    - The XMLHttpRequest object (readyState and status Properites).

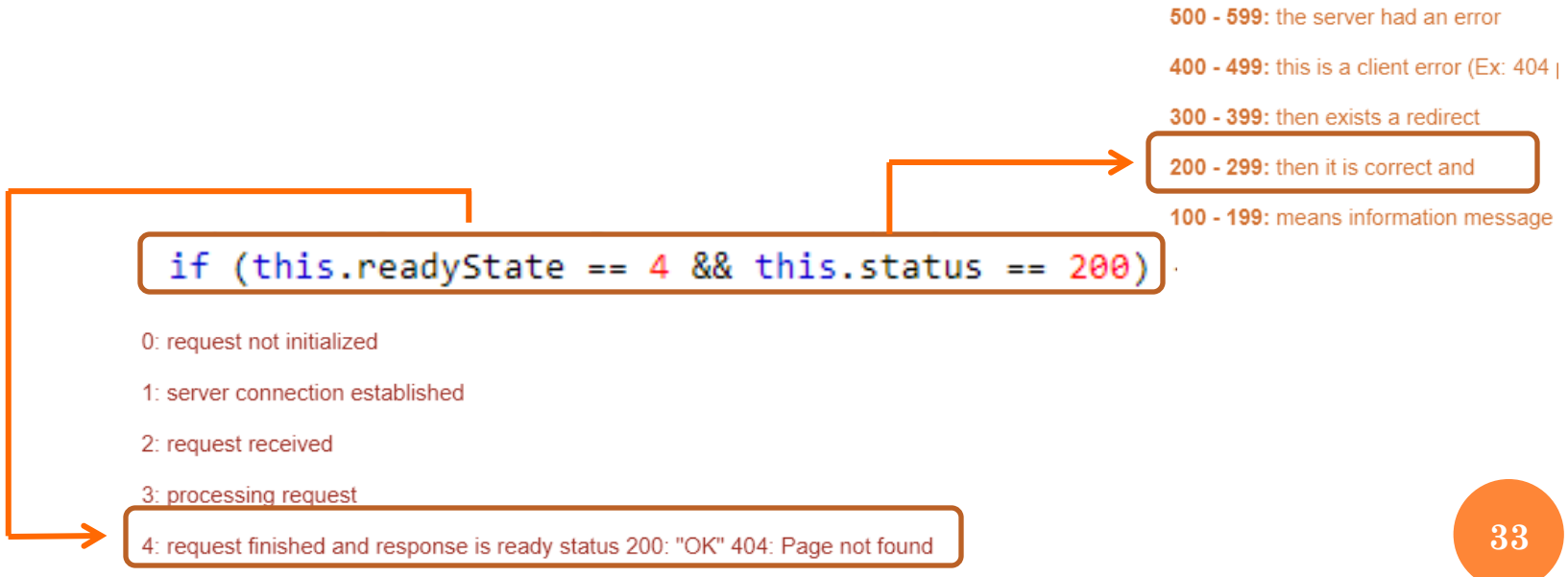
```
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
```

Ajax Basics

❑ AJAX: Asynchronous JavaScript And XML

✓ XMLHttpRequest – Properties (other methods)

- For the given xhttp variable the following set of properties apply:
- The XMLHttpRequest object (readyState and status Properites).



AJAX BASICS

❑ AJAX: Asynchronous JavaScript And XML

✓ XMLHttpRequest – Properties (other methods)

- For the given xhttp variable the following set of properties apply:
- The XMLHttpRequest object (readyState and status Properites).

readyState Property:

The readyState property defines the current state of the XMLHttpRequest object with one of the following four codes.

1.	after you have called the open() method, and before you have called send().
2.	after you have called send().
3.	after the browser has established a communication with the server, but before the server has completed the response.
4.	after the request has been completed, and the response data have been completely received from the server.

AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ XMLHttpRequest – Properties (other methods)
    - For the given xhttp variable the following set of properties apply:
    - The XMLHttpRequest object (readyState and status Properites).

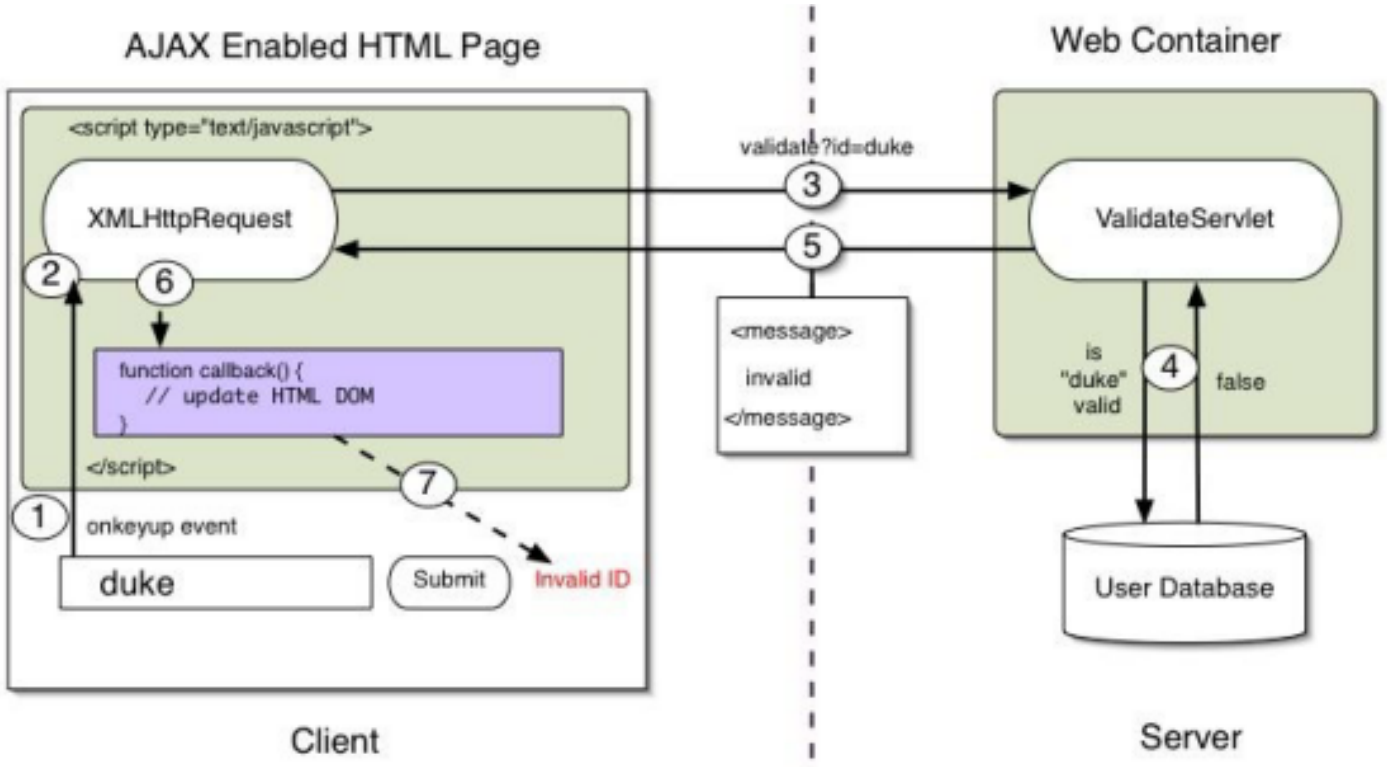
      status Property:

      The status property status indicates if server response is valid and in practice the following codes apply:

500 – 599	the server had an error
400 – 499	this is a client error (Ex: 404 page not found)
300 – 399	then exists a redirect
200 – 299	then it is correct and
100 – 199	means information message

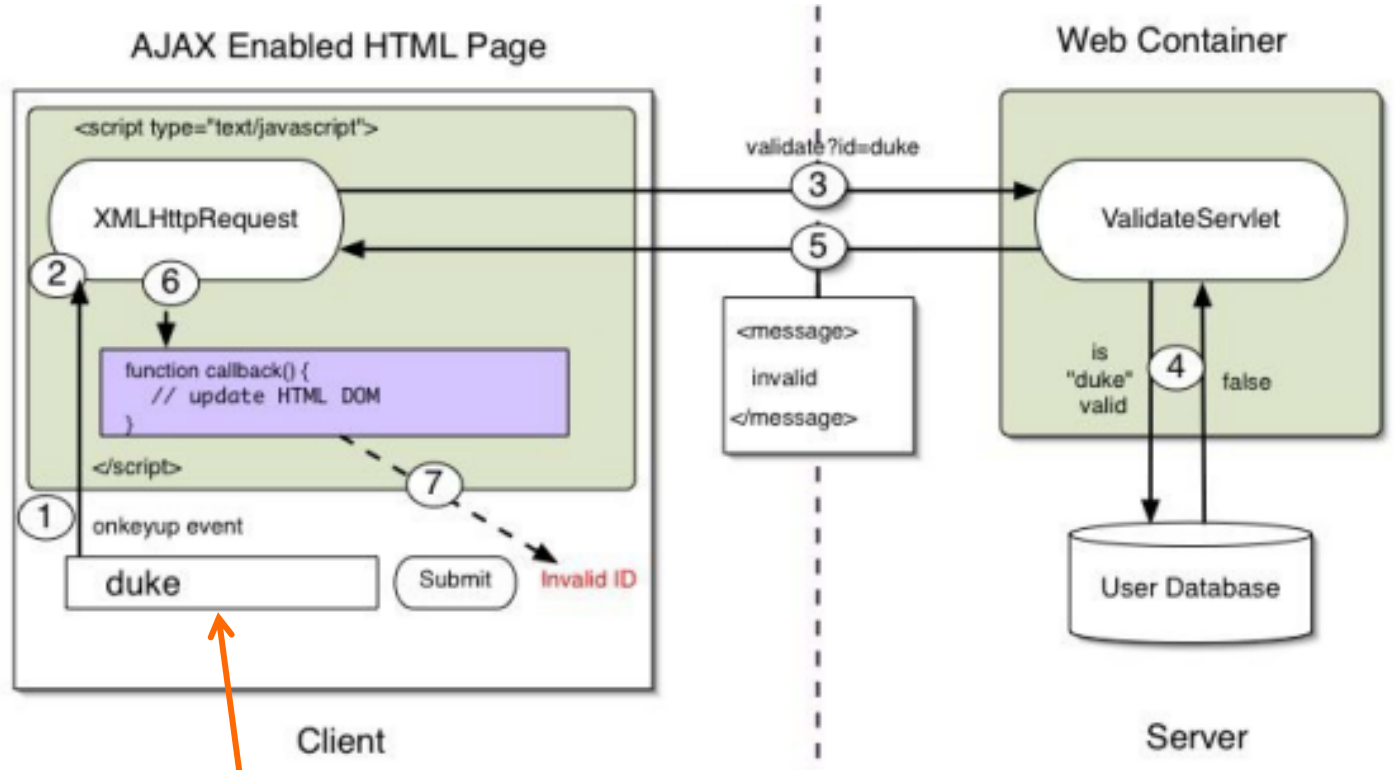
AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ AJAX – How it works...



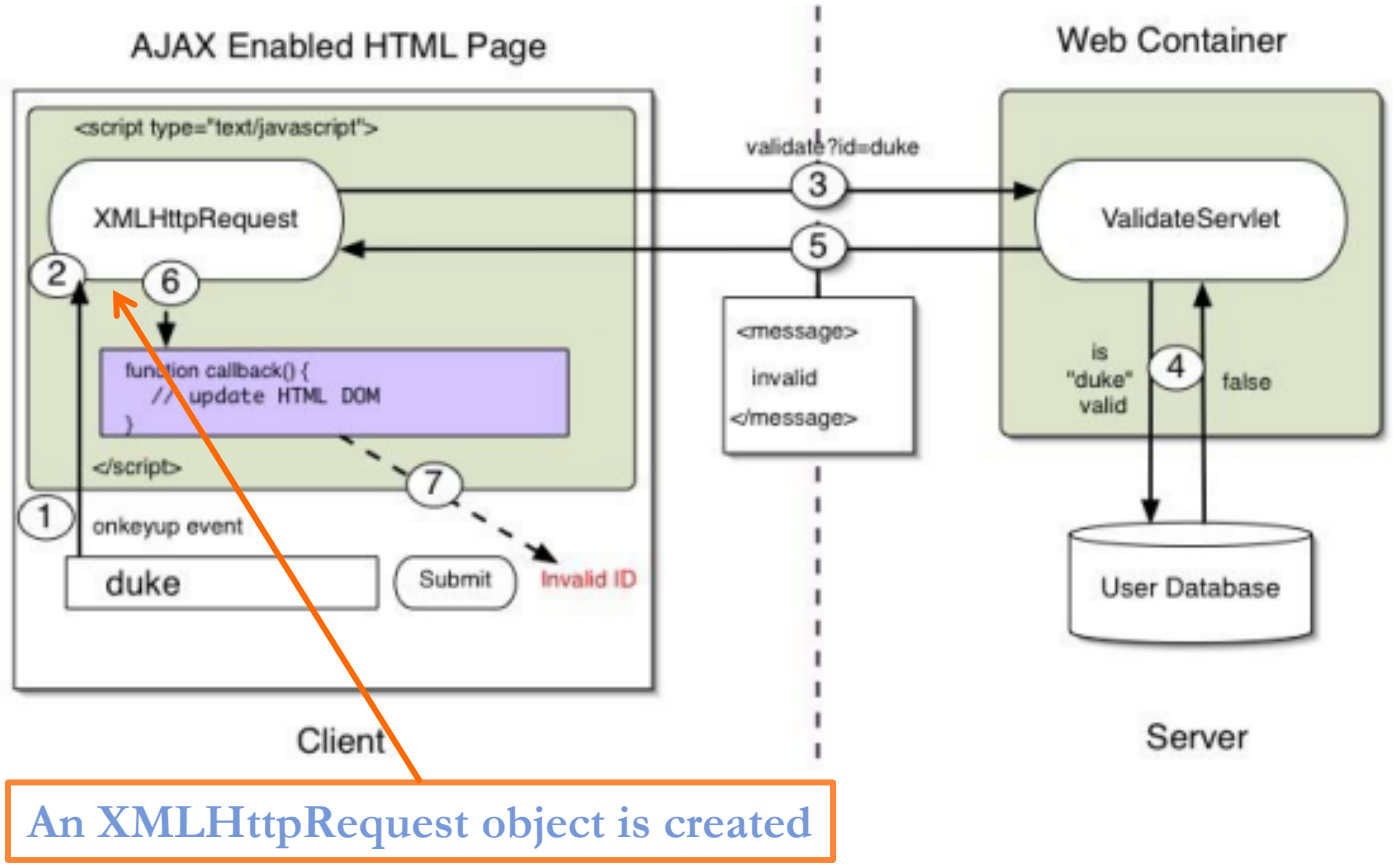
AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ AJAX – How it works...



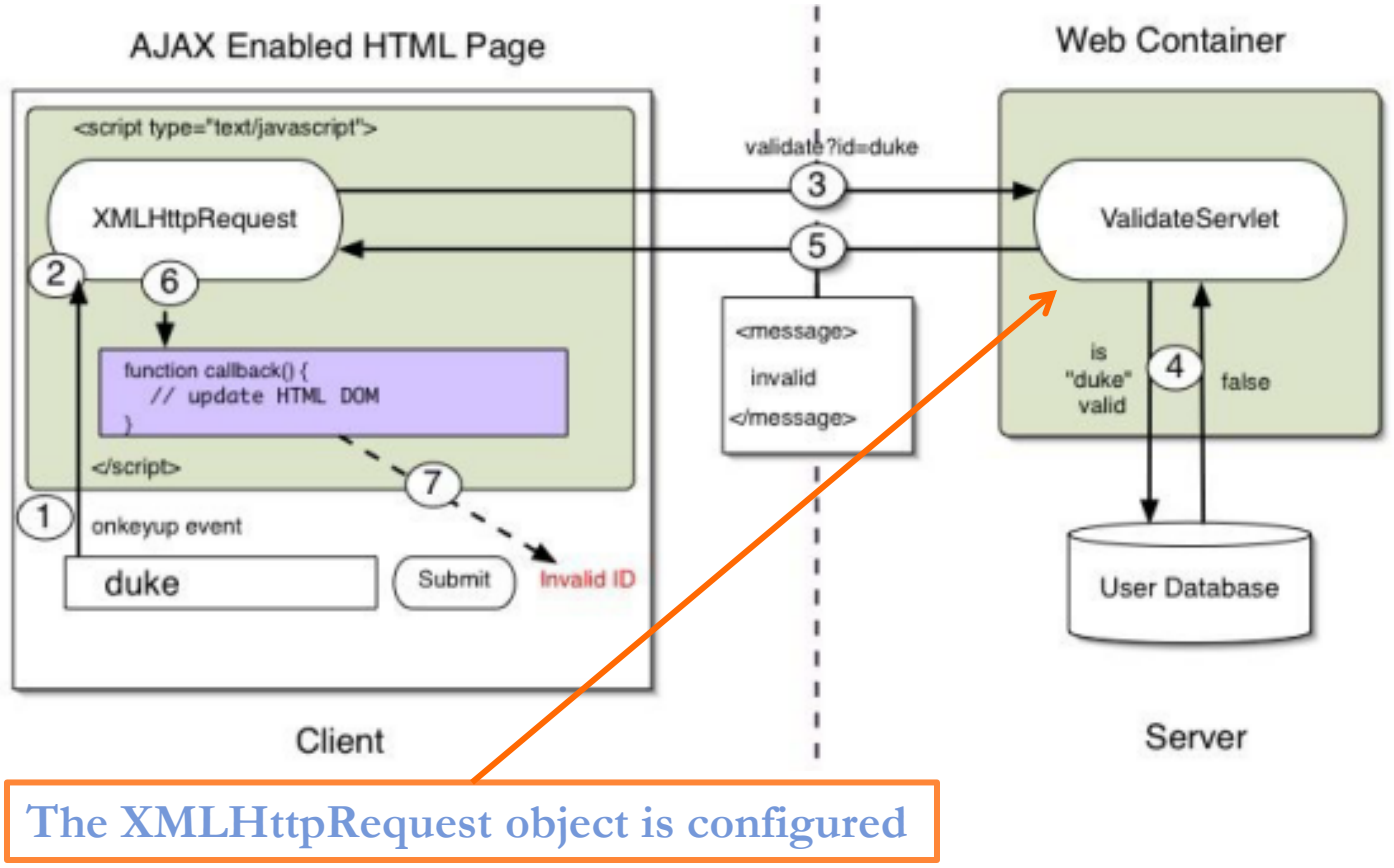
AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ AJAX – How it works...



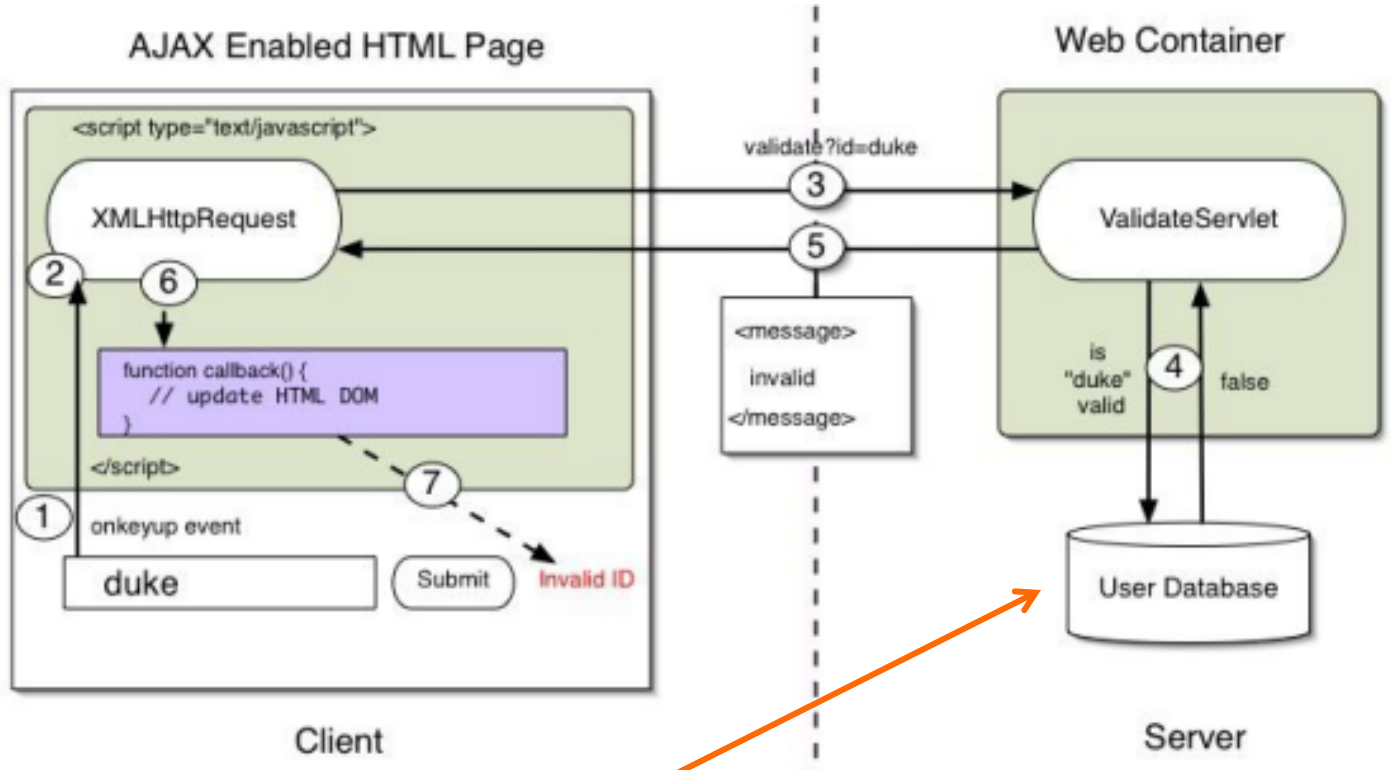
AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ AJAX – How it works...



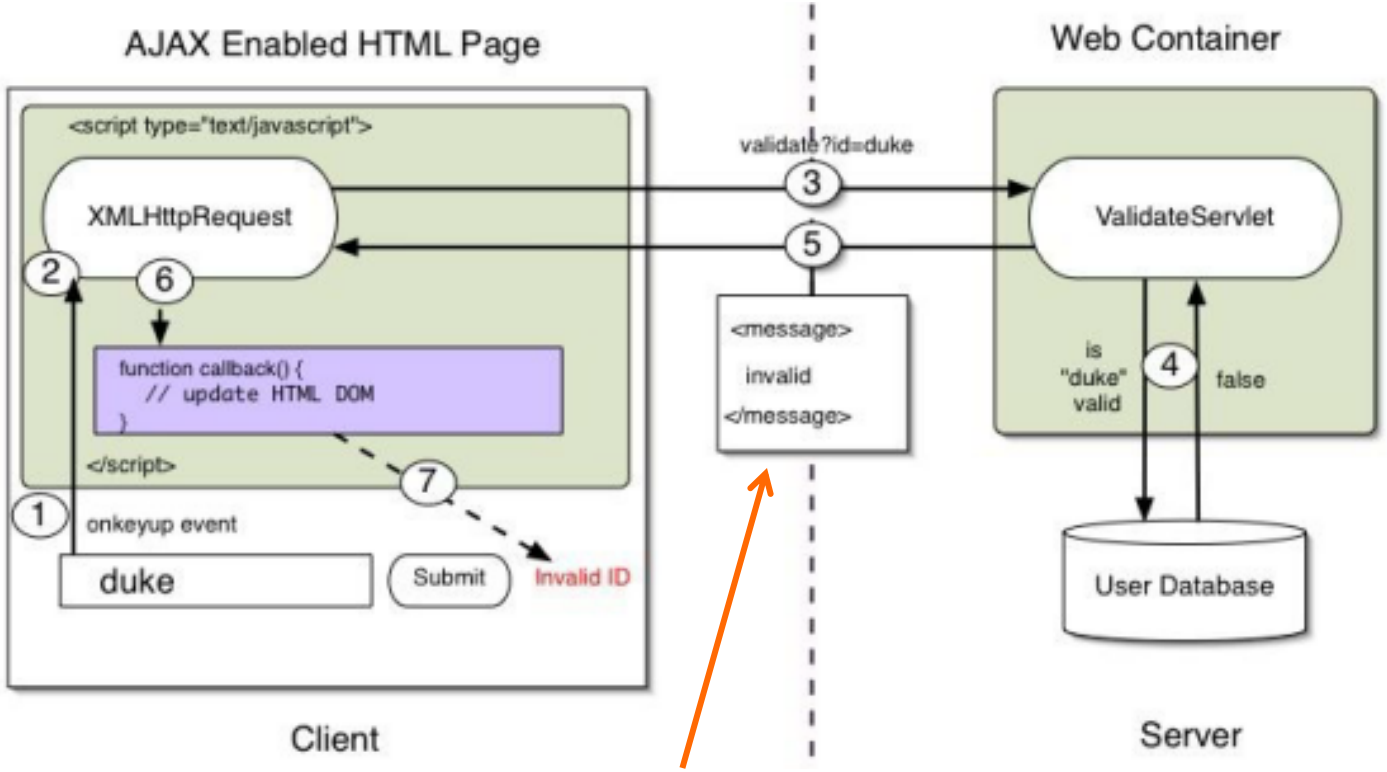
AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ AJAX – How it works...



AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ AJAX – How it works...

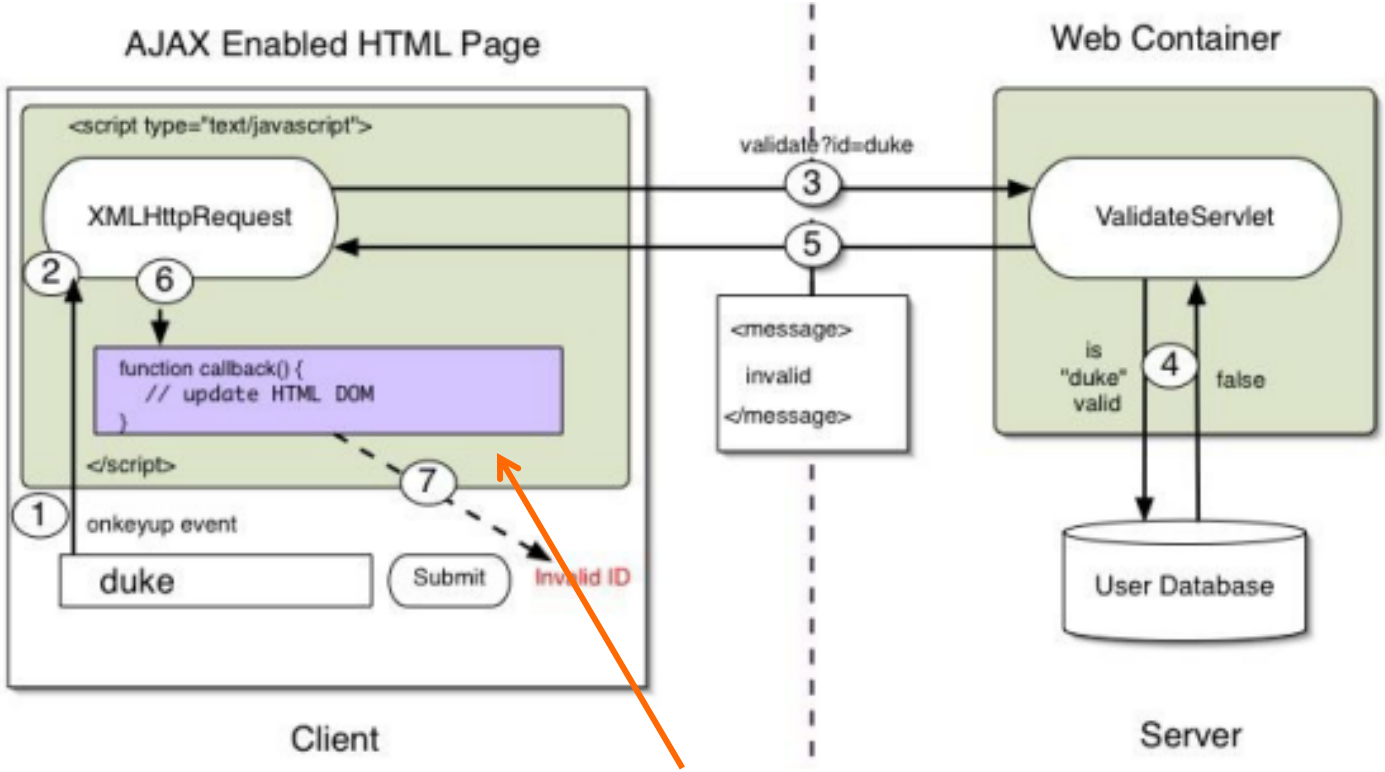


The ValidateServlet returns an XML document containing the result

AJAX BASICS

❑ AJAX: Asynchronous JavaScript And XML

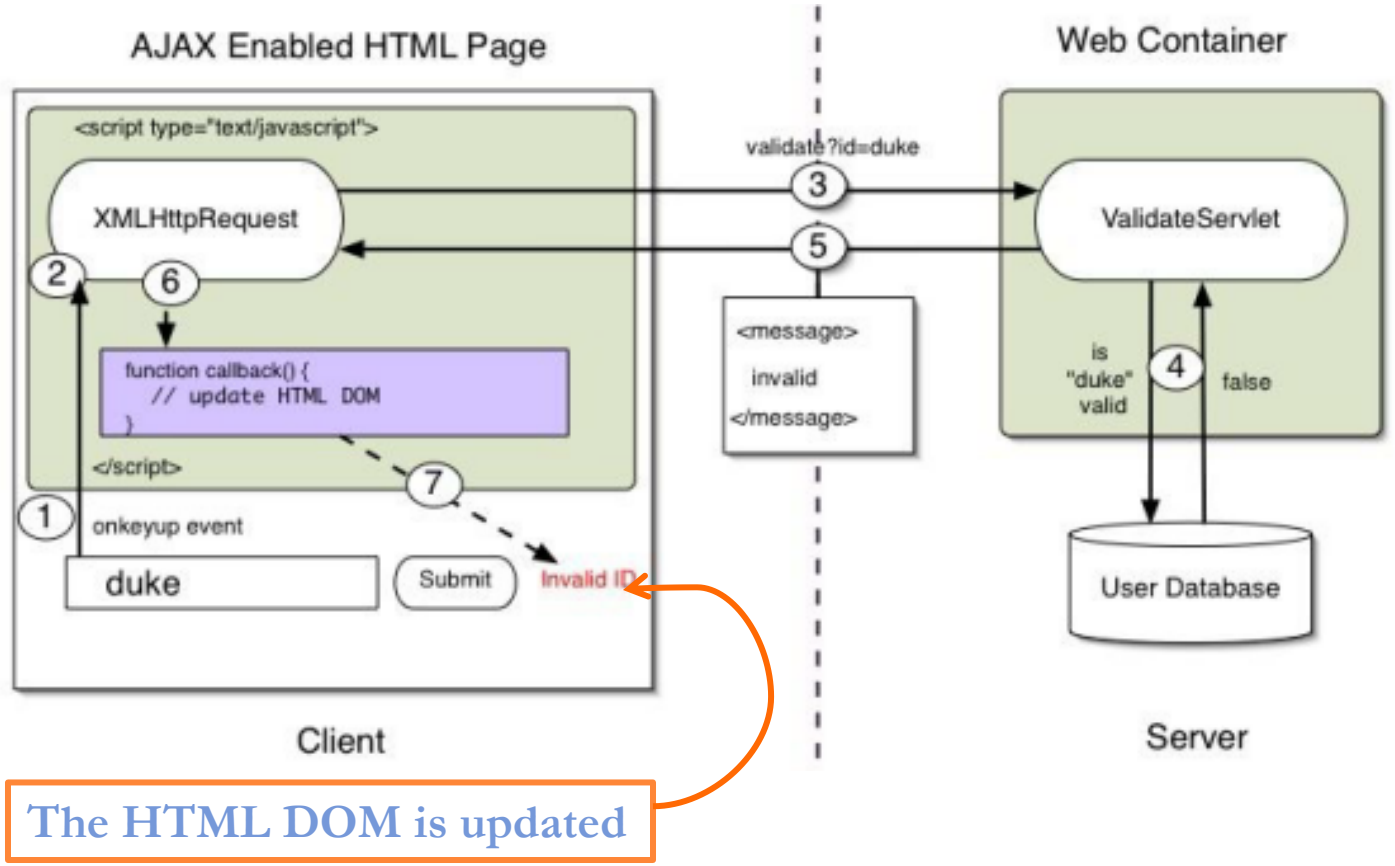
✓ AJAX – How it works...



The XMLHttpRequest object calls the callback() function and processes the result

AJAX BASICS

- **AJAX: Asynchronous JavaScript And XML**
  - ✓ AJAX – How it works...




AJAX
AMSTERDAM