



G5b: Informazio-trukea eta prozesamendua: (XML) Ajax

Rosa Arruabarrena, Jose Ángel Vadillo
LSI, UPV/EHU

1



Ajax

- **Asynchronous JavaScript And XML**
- [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))
- <http://es.wikipedia.org/wiki/AJAX>
- <http://www.w3.org/TR/XMLHttpRequest/>
- <http://www.w3schools.com/ajax/>

2



Helburua

- AJAXen garapen ereduaren oinarritzko kontzeptuen azalpena
- Orain arte irakasgaian agertu diren kontzeptu eta erreminta batzuen eta AJAXen arteko erlazioa ezarri
- AJAX erabiltzearen aukeran sakontzea eta hausnartzea honek zertan lagun gaitzakeen Web-guneen garapenen hobekuntzan

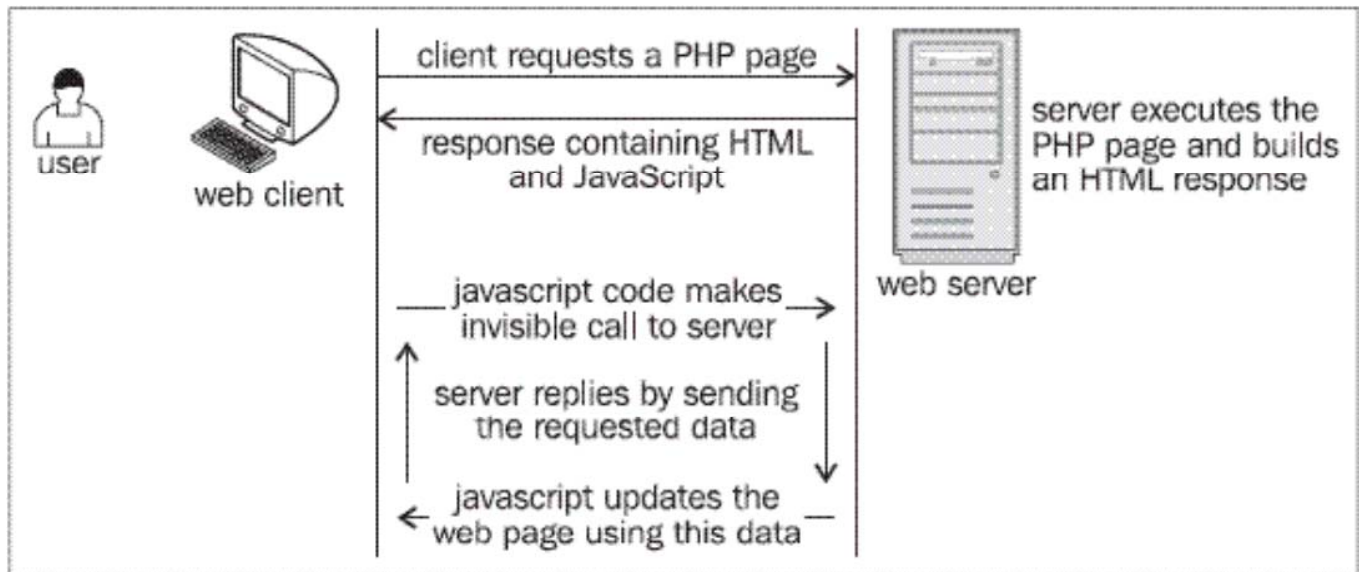
3



Ajax *Asynchronous JavaScript and XML*

- 2005ean asmatutako terminoa, Jesse James Garrett-en artikuluan (www.adaptivepath.com/publications/essays/archives/000385.php)
- *Ajax, edo JavaScript asinkronoa eta XML, web aplikazio interaktiboak garatzeko teknika bat da.*
- **Helburua:**
 - Web aplikazio sentikorrak egitea, mahai gaineko aplikazioen antzekoak.
- **Modua:**
 - Zerbitzariaren eta bezeroaren artean datu-transferentziak ezkutuan eta modu asinkronoan eginez.
- **Aldiro web-orri osoak transferitu beharrean, orri-zatiak transferitzeko aukera lantzen da.**
- Horrela web-orrien interaktibitatea, abiadura eta erabilgarritasuna hobetzen da.

4



Zertan datza Ajax teknologia



- Indarrean dauden web teknologiak konbinatzen dira.
- Hortaz, Ajax EZ da izatez teknologia berri bat.
 - Ajax teknikak honako osagaien konbinazioa erabiltzen du:
 - XHTML (edo HTML) eta CSS, web orrien edukiak markatzeko eta estilo-informazioa jartzeko.
 - Bezeroaren aldeko script-gintza lengoaia (JavaScript, normalean), DOM (Document Object Model) atzitzeko aukera ematen duena. Osagai hau funtsezkoa da.
 - **XMLHttpRequest** objektua, web zerbitzariarekin datu-transferentzia **asinkronoa** egiteko.
 - JavaScript-eko objektua da. Modu errazean egiten dira konexioak, su-hesien arazorik gabe, eta bien bitartean erabiltzaileak lanean jarrai dezake.
 - Zerbitzariaren aldeko script-gintza lengoaia (PHP edo) datuak dinamikoki sortzeko.
 - XML aipatzen da gehienetan datu-transferentziarako formatu bezala, nahiz eta beste zenbait formatuk ere balio dezaketen.



Ajax

- AJAX eredua jarraitzen duten aplikazioak **bezeroaren aldean** egikaritzen dira. Bezero eta web zerbitzarien arteko datu-transferentzia asinkronoa erabiltzen dute, honek ahalbidetzen du **web-orriek informazio zati txikiak eska ahal izatea** zerbitzariari, orri osoak eskatu orde. Eta ezaugarri honek,
 - **web aplikazioen azkartasuna handitzen du**
- Teknologia hau bezeroaren aldekoa da, eta ez dago web-zerbitzariaren softwarearen menpe
- JavaScript, XML, HTML eta CSSen oinarritzen da AJAX
 - Hurbilpen sakona egin nahi bada, komenigarria da HTMLeko DOM eta XML ezagutzea, XSLT eta SOAP

7

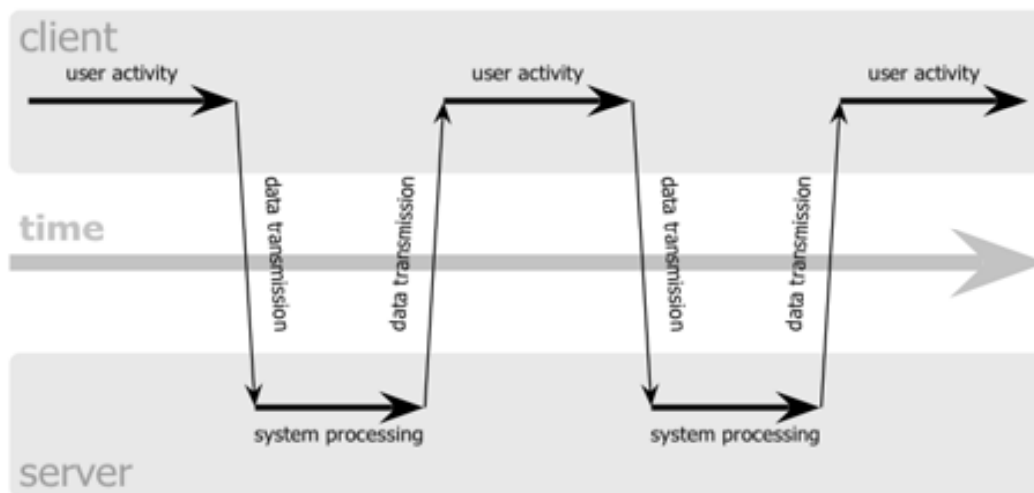


Aplikazio ezagunen adibideak

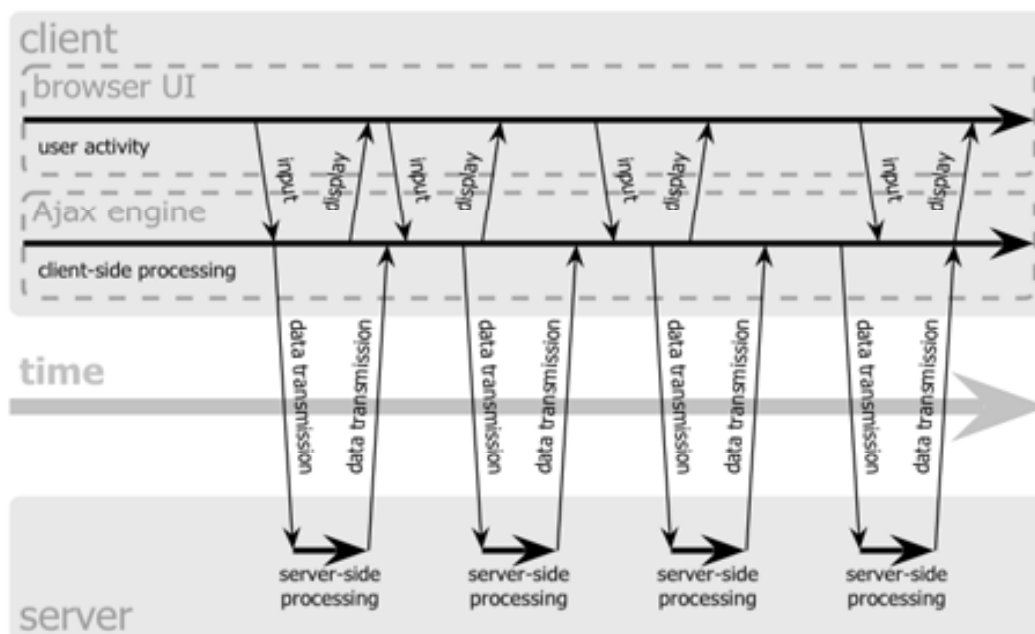
- Google Suggest:
 - <http://www.google.com/webhp?complete=1>
- Gmail
- Flickr
- Google Maps:
 - <http://maps.google.com/>
- Google docs:
 - <http://docs.google.com/>



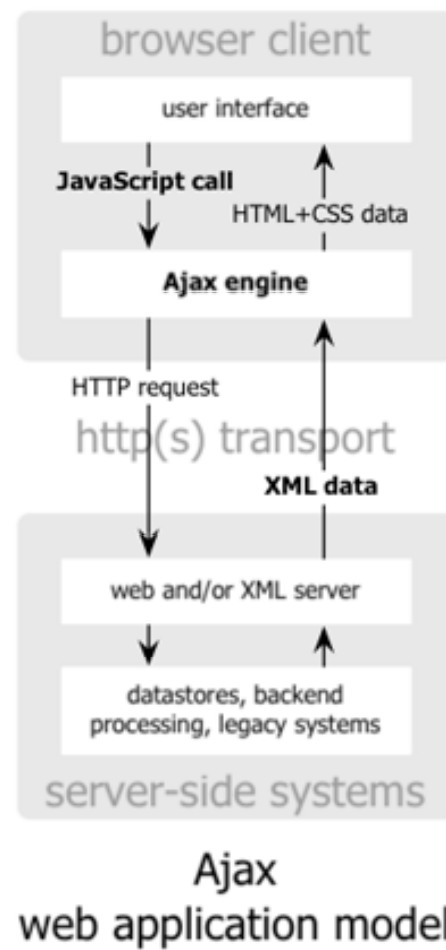
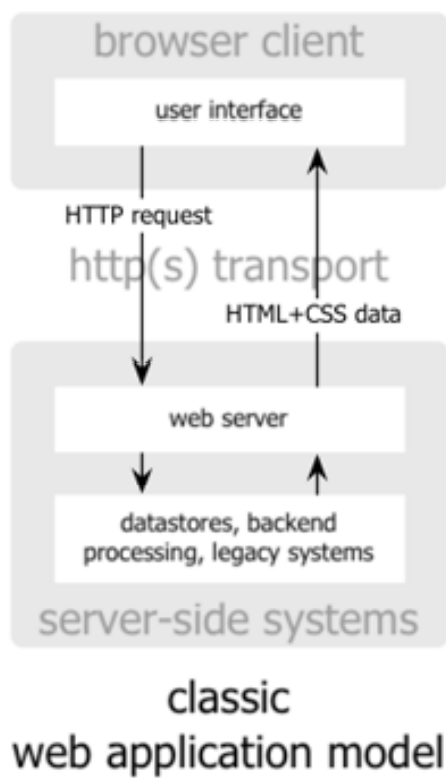
classic web application model (synchronous)



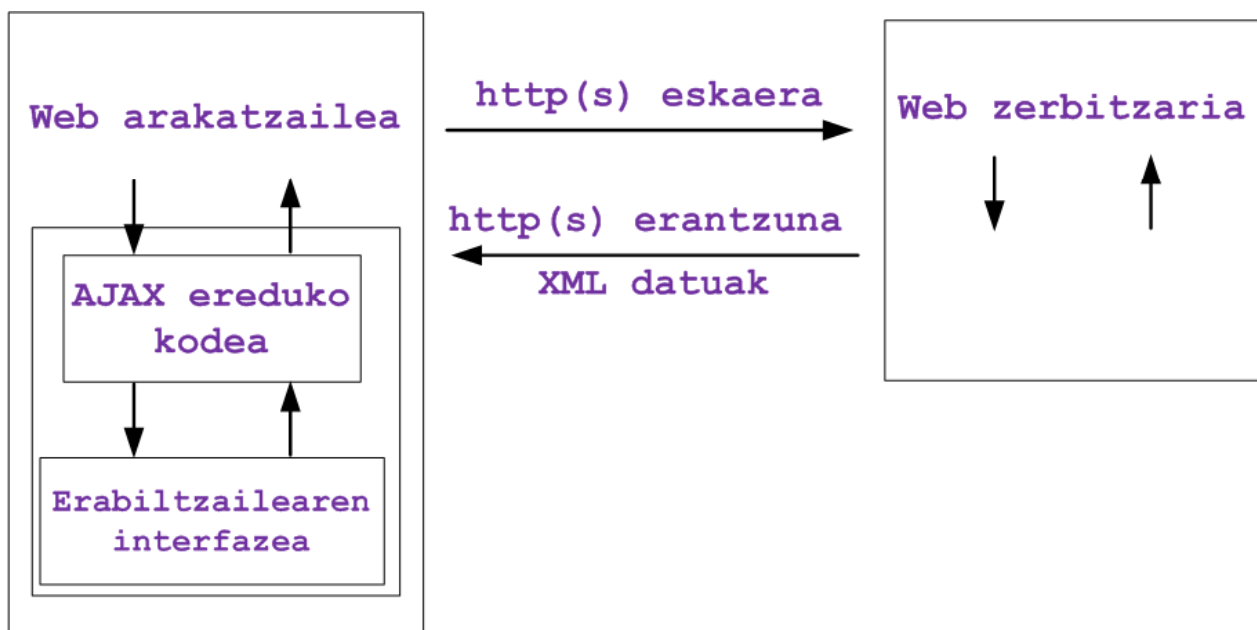
Ajax web application model (asynchronous)



Ajax teknologia asinkronoa da: zerbitzariari eskatzen zaion informazio gehigarria iristean bigarren planoan kargatzen da, orriaren bistaratzeari eragin gabe, ez eta haren portaeran ere.



11



12

AJAX-ek XML eta XMLHttpRequest erabiltzen ditu



- Web aplikazio konbentzionalek HTTP formularioak erabiltzen dituzte sarrera-informazioa igortzeko web zerbitzariari, honek informazioa prozesatzen du eta orri oso bat itzultzen dio nabigatzaileari (eta azkenik erabiltzaileari)
 - Informazio nahaste honek web aplikazioen abiaduran eragina du bai eta erabiltzailearen inguruneen adiskigarritasunean
- AJAXi esker, web aplikazioek datuak bidali eta jaso ditzakete orri osoa birkargatu gabe
 - **XMLHttpRequest** objektua erabiliz zerbitzariari datu eskaera igortzen zaio eta objektu bera erabiliz, datuak jasotzen dira zerbitzaritik, normalki, XML formatuan
 - JavaScript/DOM erabiliz, informazioa erabili/erakutsi web orriko parteak bakarrik aldatuz

13

XMLHttpRequest objektua



- Ajax-en implementazioak *XmlHttpRequest* objektuaren menpekoak dira.
 - Objektu hau erabiliz trukutzen da informazio zerbitzariarekin (bigarren plano batean)
- XMLHttpRequest (XHR) API bat da, bezeroko script-gintza lengoaiak erabil dezaketeena, web zerbitzariarekin HTTP bidez XML datuak transferitzeko.
- Nahiz XMLHttpRequest deitzen den, objektu hori XML ez den beste formatu batzuetarako ere erabil daiteke (testu soila, esate baterako).

14



XMLHttpRequest objektua

- Eraikitzailea

```
// XMLHttpRequest objektua sortzen du
var r = new XMLHttpRequest();

(IE5 eta IE6 bertsio zaharkituentzat)
var r = new ActiveXObject("Microsoft.XMLHTTP");
```

15

XMLHttpRequest objektua Propietateak



Propietatea	Deskribapena
readyState	Objektuaren egoera itzultzen du: 0 = sortua, baina hasieratu gabe 1 = hasieratuta: kargatzen 2 = eskaera bidalia: header-ak kargatzen 3 = erantzuna jasotzen: interakzionatzen 4 = erantzuna jasota..
responseText	Erantzuna string moduan itzultzen du
responseXML	Erantzuna itzultzen du XML moduan, hots, DOM zuhaitz baten moduan. DOM eko metodoak eta propietateak erabiliz atzigarria Itzulitako XML dokumentua W3C DOM APIaren bidez landu daiteke.
status	Osoko bat. "HTTP status code" itzultzen du zenbaki moduan (adibidez, 404: "Not Found" eta 200: "OK")
statusText	Statusa itzultzen du string moduan (adibidez, "Not Found" edo "OK").

16

XMLHttpRequest objektua

Metodoak



Metodoa	Deskribapena
<code>abort()</code>	Uneke eskaera bertan behera utzi.
<code>getAllResponseHeaders()</code>	Erantzuneko HTTP goiburuko balio guztiak string moduan itzuli.
<code>getResponseHeader(headerName)</code>	Erantzuneko HTTP goiburutik adierazitako balioa itzuli.
<code>open(method, URL)</code> <code>open(method, URL, async)</code> <code>open(method, URL, async, userName)</code> <code>open(method, URL, async, userName, password)</code>	<p>Eskaera batek behar dituen datuak hasieratzen dira:</p> <p>method: "GET", "POST", "HEAD", "PUT", "DELETE",...</p> <p>URL: Eskaeraren URLa. Inbokatzen den baliabidearen url-a.</p> <p>async: eskaera asinkronoa izango den ala ez. "true" denean, deiaaren ostean exekutatzeko jarraituko du, bestela erantzunaren zain geratuko da.</p> <p>userName: erabiltzaile-izena</p> <p>password: pasahitza</p>
<code>send(content)</code>	Eskaera bidaltzen du; content parametroan 'POST' eskaeretako edukia pasatzen zaio
<code>setRequestHeader(label, value)</code>	Bidaliko den HTTP goiburukoari label/value bikote bat gehitu.

17

http 1.1-ek 5 mota gehitzen ditu: OPTIONS, PUT, DELETE, TRACE eta CONNECT

XMLHttpRequest objektua

Gertaerak



Gertaera	Deskribapena
<code>onreadystatechange</code>	Objektuaren egoera aldatzen den bakoitzean aktibatzen den gertaera-maneeiatzailea
<code>Onabort,</code> <code>onload,</code> <code>onloadstart,</code> <code>onprogress,</code> <code>ontimeout</code>	Some events, in the new specification: level 2

18



XMLHttpRequest. Erantzuna

- Erantzuna *XMLHttpRequest* objektuaren *responseText* propietatetik irakurtzen da
- Iristen den erantzuna uneko HTML dokumentuan txertatzen da
- Erantzuna XML denean
 - XML erantzunak egituratu(ago)ak dira, eta egokiagoak edukien kudeaketa landua egiteko
 - Erantzuna XMLn bueltatzeko, zerbitzariko script-ak "Content-type" goiburukoan 'text/xml' balioa izan behar du
 - Erantzuna *XMLHttpRequest* objektuaren *responseXML* propietatetik irakurtzen da
 - Erantzuna DOM zuhaitza da

19



Laburbilduz: nola programatu

- Gertaera zehatz bat sortzen duen egoerari Javascript funtzio bat erlazionatzen zaio
- Funtzioaren kodean orri dinamiko bat (asp, php, hsp,...) inbokatzen duen (maiz modu asinkronoan) kodea dago, eta XML emaitza itzuliko du (beste testuz formatua izan liteke)
- Zerbitzariak bidalitako informazioa (javascript kodeak) tratatzen eta erakusten du, HTML orrian dagokion objektuko propietateak atzitzuz.

20



Adib1: Testu fitxategi baten edukiaren bistaratzea

```
<body>
<h1>AJAX erabiliz datuen azalpena</h1>
<form>
<input type="button" value="Mezua erakutsi" onclick="datuakEskatu()">
</form>
<div id="emaitza" style="background-color:#99FF66;">
<p>Hemen testua azalduko da</p>
</div>
</body>
</html>
```

21



Adib1: Testu fitxategi baten edukiaren bistaratzea

```
<script language = "javascript">
  //(IE7+, Firefox, Chrome, Safari, and Opera)
  xmlhttp = new XMLHttpRequest();

  xmlhttp.onreadystatechange = function(){
    alert (xmlhttp.readyState);
    if (xmlhttp.readyState==4)
    {var obj = document.getElementById('emaitza');
    obj.innerHTML = xmlhttp.responseText;}
  }
  function datuakEskatu(){
    xmlhttp.open("GET", 'datuak.txt');
    xmlhttp.send(null);
  }
</script>
```

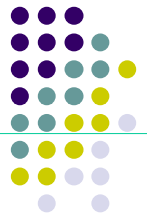
22



Adib2: XML dokumentu bat maneiatzen

```
<body>
<h1>AJAX erabiliz datuen azalpena</h1>
<form>
<input type="button" value="Pelikula erakutsi"
  onclick="datuakEskatu()">
</form>
<div id="emaitza" style="background-color:#99FF66;">
<p>Pelikula baten titulua agertuko da</p>
</div>
</body>
</html>
```

23



Adib2: XML dokumentu bat maneiatzen

```
<script language = "javascript">
  xhttp = new XMLHttpRequest();

  xhttp.onreadystatechange = function(){
    if (xhttp.readyState==4)
      { alert(xhttp.responseText); // xml dokumentua string moduan bistaratu
        var erantzuna = xhttp.responseXML;
        var obj = document.getElementById('emaitza');
        var x = erantzuna.getElementsByTagName('titulo')[0];
        var y = x.childNodes[0];
        obj.innerHTML = y.nodeValue;
      }
  };

  function datuakEskatu(){
    xhttp.open("GET", 'pelis2.xml');
    xhttp.send(null);
  }
</script>
```

24

Adib3: DB bat atzitzen duen PHP programa bati deia



```
<!doctype HTML>
<html> <head>
<script type="text/javascript" language = "javascript">
    xhttp = new XMLHttpRequest();

    xhttp.onreadystatechange = function(){
        alert(xhttp.readyState);
        if ((xhttp.readyState==4)&&(xhttp.status==200 ))
        {    document.getElementById("txtHint").innerHTML= xhttp.responseText;
        }
    };
function izenakIkusi(){
    xhttp.open("GET","datuakIkusi.php", true);
    xhttp.send();
}
</script></head>
<body>

 <br>
<form>
<input type="button" name="kideakIkusi" value="Kideak ikusi"
    onclick="izenakIkusi()">
</form>
<div id="txtHint" style="background-color:#99FF66;">
<p>Nire kideak hemen agertuko dira...</p>
</div>
</body></html>
```

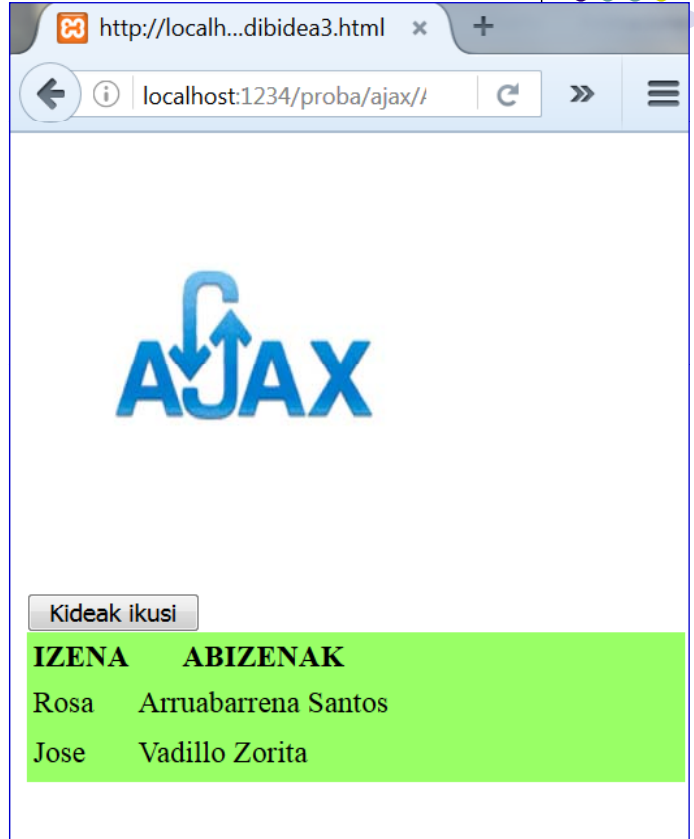
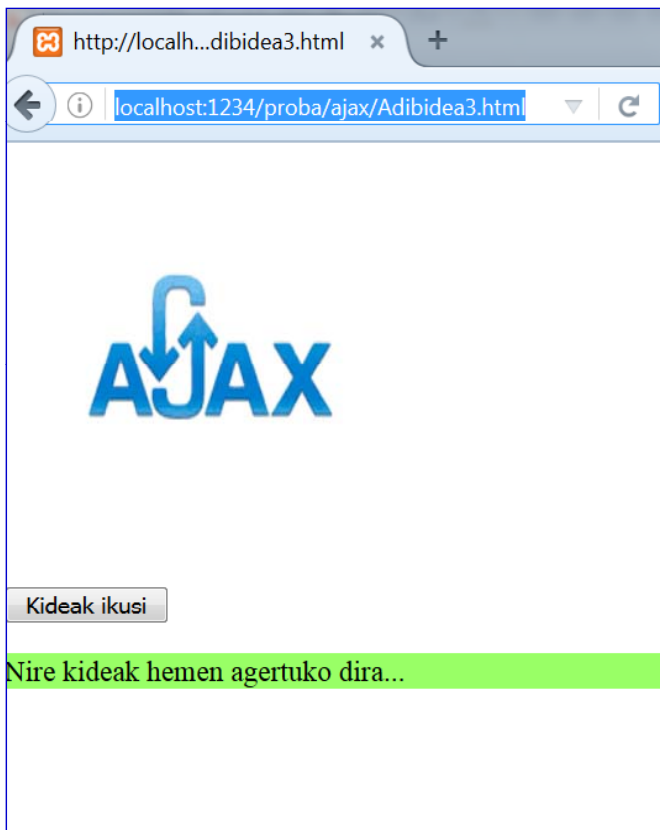
25

Adib.3: DB bat atzitzen duen PHP programa bati deia



```
1 <?php
2 $linki = new mysqli ("localhost","root","","quiz");
3 if ($linki->connect_errno){
4     echo "Error while connecting to MySQL".$linki->connect_error;
5 }
6 else {
7     $sqli="SELECT * FROM erabiltzaileak";
8     $result= $linki->query($sqli);
9     if (!$result){
10         echo "Error in the query".$result->connect_error; $linki->close();
11     } else{
12         echo '<table><tr><th>IZENA</th><th>ABIZENAK</th></tr>';
13         while ($row = $result->fetch_assoc()){
14             echo '<tr><td>'. $row['izena'].'</td><td>'. $row['abizenak'].'</td><tr>';
15         }
16         echo '</table>';
17     }
18 }
19 >
```

26



27



Adib4: PHPri deia parametroak pasatuz GETen

```

<!doctype HTML>
<html>
<head>
<script type="text/javascript" language = "javascript">
xmlhttp = new XMLHttpRequest();

xmlhttp.onreadystatechange=function(){
    if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {document.getElementById("txtHint").innerHTML=xmlhttp.responseText; }
}

function showPeli(str){
if(str=="") {
    document.getElementById("txtHint").innerHTML="";
    return;
}
xmlhttp.open("GET", "pelikulakoDatuak.php?q="+str, true);
xmlhttp.send();
}

</script>

```

28



Adib4: PHPri deia parametroak pasatuz GETen

```
<body>
<br>
<form>
Titulu bat aukera ezazu:
<select name="pelis" onchange="showPeli(this.value)">
  <option value="">.....</option>
  <option value="La vida de los otros">La vida de los otros</option>
  <option value="Milenium I">Milenium I</option>
</select>
</form>
<div id="txtHint"><b>Pelikulako datuen bistaratzea hemen ...</b></div>
</body>
```

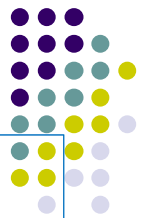
29

```
1 <?php
2 $irteera="";
3 sleep(2);
4 $q=$_GET["q"];
5 $peliz=simplexml_load_file('pelis2.xml');
6 foreach($peliz->xpath('//pelicula') as $pelikula){
7   if ($pelikula->titulo==$q){
8     $irteera="Argumentua: <br>";
9     foreach($pelikula->personajes->personaje as $pertsonaia){
10      $irteera .= $pertsonaia->nombre. "==" . $pertsonaia->actor. "<br>";
11    }
12  }
13 }
14 if ($irteera==""){echo "Pelikulak ez du existitzen";}
15 {echo $irteera;}
16 ?>
```



30

Adib5: PHPri deia parametroak pasatuz POSTen



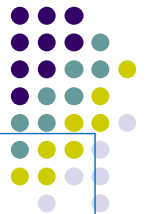
```
function showPeli(str)
...
xmlhttp.open("POST", "pelikulakoDatuakPOST.php", true);
xmlhttp.setRequestHeader("Content-type",
    "application/x-www-form-urlencoded");
xmlhttp.send("titulo="+str);
}
```

```
<?php
$irteera="";
sleep(2);
$q=$_POST["titulo"];
$peliz=simplexml_load_file('pelis2.xml');
foreach($peliz->xpath('//pelicula') as $pelikula){
    if ($pelikula->titulo==$q){
        $irteera="Argumentua: <br>";
        foreach($pelikula->personajes->personaje as $pertsonaia){
            $irteera .= $pertsonaia->nombre. "==">" . $pertsonaia->actor.
        }
    }
}
if ($irteera==""){echo "Pelikulak ez du existitzen";}
{echo $irteera;}
?>
```



31

Adib6: Erantzunaren zain egonik, mezuen erakusketa



```
function showPeli(str) {
if (xmlhttp){
    xmlhttp.onreadystatechange=eskaerarenEgoera;
    xmlhttp.open("GET", "pelikulakoDatuak.php?q="+str,true);
    xmlhttp.send(null);}
}

function eskaerarenEgoera() {
switch(xmlhttp.readyState) { //Eskaeraren egoeraren arabera Testu bat
    itzultzen da
case 0: document.getElementById('estado').innerHTML =
    "Hasi gabe..."; break;
case 1: document.getElementById('estado').innerHTML =
    "<b>Kargatzen...</b>"; break;
case 2: document.getElementById('estado').innerHTML =
    "<b>Kargatzen...</b>"; break;
case 3: document.getElementById('estado').innerHTML =
    "Elkarrekintza..."; break;
case 4: document.getElementById('estado').innerHTML =
    "<b>Amaitua...</b>";

//Eskaera amaitu bada, gainera informazioa itzultzen da.
document.getElementById('txtHint').innerHTML=xmlhttp.responseText; break;
}
}
```

32