



AJAX – Asynchronous JavaScript And XML (...mali delić WEB 2.0 standarda)

Tutorijal za osnovno korišćenje AJAX-a - Vladica Savić

Šta je zapravo AJAX?

AJAX je skraćenica od Asynchronous JavaScript And XML.

Kratko rečeno AJAX je zasnovan na JavaScript i HTTP zahtevima i njegova glavna osobina je da omogući da se na nekom sajtu šalje/prima sadržaj bez ponovnog učitavanja stranice.

Šta bi već trebalo da znate pre nego počnete sa AJAX-om?

- HTML / XHTML
- JavaScript
- Server-side scripting

AJAX nije novi jezik programiranja, već tehnika za kreiranje boljih, bržih i interaktivnijih web aplikacija. Sa AJAX-om, vaš JavaScript može da komunicira direktno sa serverom, koristeći JavaScript XMLHttpRequest objekat. Pomoću ovog objekta, vaš JavaScript može da razmenjuje podatke sa web serverom, bez ponovnog učitavanja stranice.

AJAX koristi asinhroni transfer podataka (HTTP zahtevi) između pretraživača i servera, omogućujući web aplikaciji da obrađuje samo one podatke koji su zaista potrebni umesto čitavih stranica.

AJAX je zasnovan na sledećim web standardima:

- JavaScript
- XML
- HTML
- CSS

Web standardi koji se koriste u AJAX-u su dobro definisani i podržani od svih velikih web browsera.

AJAX prednosti

Web aplikacije imaju mnogo prednosti nad desktop aplikacijama, osnovna je da one lako mogu da dopru do više korisnika, lakše su za instaliranje i obezbeđivanje podrške, i lakše za razvijanje.

Sa AJAX-om, internet aplikacije mogu da budu bogatije i brže za korišćenje nego što su bile do sada.

AJAX koristi HTTP zahteve

U "klasicnom" programiranju, ako želite da dobijete bilo kakvu informaciju iz baze podataka ili fajla nekog koji se nalazi na serveru, ili pošaljete serveru informacije o korisniku, vi trebate da napravite HTML formu i preko GET ili POST metode dobijete ili pošaljete podatke serveru. Sto bi značilo da nakon sto se pošalje/dobije informacija od servera da se onda učita opet cela strana kako bi rezultati bili vidljivi i samim tim pošto server vraća novu stranicu svaki put kada korisnik pošalje neke podatke, tradicionalne web aplikacije znaju da budu spore.

Pomocu AJAX-a JavaScript komunicira direktno sa serverom, kroz JavaScript XMLHttpRequest objekat. Pomoću HTTP zahteva, web stranica šalje zahtev serveru, i dobije odgovor od njega - bez ponovnog učitavanja web stranice, i korisnik gotovo da i neće primetiti da je skript poslao zahtev za stranicom, ili da je u pozadini poslao podatke na server.

XMLHttpRequest Objekat

Koristeći XMLHttpRequest objekat, web programer može da ažurira stranicu sa podacima sa servera nakon što se stranica učitala sto je opet jedan veliki plus!

Napomena

Kada radimo sa **XMLHttpRequest**-om trebamo voditi računa i koji browser je u pitanju. Veliki broj browsera podržava **XMLHttpRequest** (Internet Explorer (...sve verzije novije od 5.0 verzije), Firefox, Opera (sto bi ukratko značilo skoro svi)) ali je malo drugacije slanje **XMLHttpRequest-a** svakom od njih.

A sad, da počnemo nešto konkretnije nakon što smo se upoznali teorijski sa AJAX-om...

Pokazaću vam kako AJAX nije neka babaroga koje se treba plašiti, i da je zapravo relativno lako koristiti ga.

Napravićemo malu web aplikaciju koja neće imati neki kompleksan zadatak, već će biti mali primer kako jednostavno da pošaljete serveru nešto na obradu i da server vrati obrađen podatak, i to bez ponovnog učitavanja (osvežavanja) cele strane...

Napravićemo HTML formu u kojoj ćemo imati jedno tekstualno polje u koje ćemo unositi tekst, neko ime recimo i skriptu koja će nam vratiti podatak o tome šta smo uneli u to tekstualno polje, i **md5()**-ovan oblik toga recimo...vi naravno možete staviti bilo šta drugo, ali ovde sam hteo da samo pokažem kako lako sve to implementirati.

Dakle, napravićemo HTML fajl koji će za pocetak izgledati ovako:

```
<html>
<body>
<form name="moja_forma">
Unesi svoje ime:
<input type="text" onKeyUp="ajaxFunction();" name="ime" />
Prikaz nakon obrade AJAX-om:
<span id="prikaz" />
</form>
</body>
</html>
```

Primitićete da se na "onKeyUp" događaj poziva JavaScript funkcija pod imenom "ajaxFunction()", ona izgleda ovako:

```
function ajaxFunction()
{
var xmlhttp;
try
{
// Firefox, Opera 8.0+, Safari i ostali "NORMALNI" browseri
xmlhttp=new XMLHttpRequest();
}
catch (e)
{
// IE - (Imbecil Engine (Internet Explorer))
try
{
xmlhttp=new ActiveXObject("Msxml2.XMLHTTP");
}
catch (e)
{
try
{
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
catch (e)
{
alert("Tvoj browser ne podrzava AJAX!");
return false;
}
}
}
xmlhttp.onreadystatechange=function()
{
if(xmlhttp.readyState==4)
{
document.getElementById("prikaz").innerHTML = xmlhttp.responseText;
}
}
var sta_kucamo = (document.moja_forma.ime.value);
var url="obradi.php?tekst="+sta_kucamo;
xmlhttp.open("GET",url,true);
xmlhttp.send(null);
}
```

Šta možemo da primetimo u ovoj skripti? Pa, najpre određujemo koji browser korisnik ima i da li njegov browser podržava AJAX i na osnovu toga kreiramo **XMLHttpRequest** objekat, i da nakon što to uradimo, ukoliko browser podržava AJAX sledi provera stanja XMLHttpRequest objekta:

```
xmlHttpRequest.onreadystatechange=function()
{
  if(xmlHttpRequest.readyState==4)
  {
    document.getElementById("prikaz").innerHTML = xmlHttpRequest.responseText;
  }
}
```

xmlHttpRequest.onreadystatechange - čuva funkciju koja će da procesuirati podatke koje dobija od servera.

xmlHttpRequest.readyState==4 - stanje 4 predstavlja stanje kada je završena obrada poslatih podataka preko ajax-a. U našem slučaju dodelićemo našem HTML elementu čiji je id "**prikaz**" tekst koji će nam vratiti skripta koja obrađuje željene podatke.

Inače, moguće vrednosti readyState XMLHttpRequest-a su:

- 0 – Zahtev nije inicijalizovan
- 1 – Zahtev je podešen
- 2 – Zahtev je poslat
- 3 – Zahtev se obrađuje
- 4 – Zahtev je završen

U promenljivoj **sta_kucamo** smeštamo tekst koji je u textbox-u pod imenom "**ime**" i to prosleđujemo preko GET metode našoj skripti na serveru, u mom slučaju pod imenom "**obradi.php**" (u promenljivoj "**url**").

Da bi poslali zahtev na server mi koristimo `open()` i `send()` metode.

xmlHttpRequest.open("GET",url,true);

Metoda `open()` ima tri argumenta, prvi argument definiše koji metod slanja podataka koristimo (GET ili POST), drugi argument predstavlja URL skripte na serveru koja obrađuje podatke koje šaljemo, i treći argument kaže da zahtev bude obavljen asinhrono.

xmlHttpRequest.send(null);

Da bi zahtev za obradu serveru konačno bio poslat moramo da iskoristimo i `send()` metodu.

I znači naša **demo.php** strana izgleda ovako:

```
<html>
<body>
<script type="text/javascript">
function ajaxFunction()
{
  var xmlHttpRequest;
  try
  {
    // Firefox, Opera 8.0+, Safari i ostali "NORMALNI" browseri
    xmlHttpRequest=new XMLHttpRequest();
  }
  catch (e)
  {
    // IE - (Imbecil Engine (Internet Explorer))
    try
    {
      {
        xmlHttpRequest=new ActiveXObject("Msxml2.XMLHTTP");
      }
    }
    catch (e)
    {
      try
      {
        {
          xmlHttpRequest=new ActiveXObject("Microsoft.XMLHTTP");
        }
      }
    }
    catch (e)
    {
      alert("Tvoj browser ne podržava AJAX!");
      return false;
    }
  }
}
```

```

xmlHttpRequest.onreadystatechange=function()
{
  if(xmlHttpRequest.readyState==4)
  {
    document.getElementById("prikaz").innerHTML = xmlHttpRequest.responseText;
  }
}
var sta_kucamo = (document.moja_forma.ime.value);
var url="obradi.php?tekst="+sta_kucamo;
xmlHttpRequest.open("GET",url,true);
xmlHttpRequest.send(null);
}
</script>
<form name="moja_forma">
Unesi svoje ime:
<input type="text" onKeyUp="ajaxFunction();" name="ime" />
Prikaz nakon obrade AJAX-om:
<span id="prikaz" />
</form>
</body>
</html>

```

Na početku smo rekli da će naša skripta na serveru da obradi podatke koje joj pošaljemo i vrati nam ih da vidimo kako ona može da izgleda:

```

<?php
echo "MD5-ovan oblik gore unetog teksta je: ". md5($_GET["tekst"])."";
?>

```

Ovu php skriptu ćemo sačuvati pod imenom "obradi.php" i postaviti u istom folderu gde i gore pomenutu skriptu demo.php i to je to, kad završite ovako bi trebalo da izgleda:

<http://vladicavac.iz.rs/ajax/demo.php>

Ukoliko imate neki komentar pošaljite možete me kontaktirati na vladicavac@yahoo.com ili <http://vladicavac.iz.rs>
 Pozdrav svima koji stignu do dna ove strane, Vladica Savić.