

# AJAX

# ŠTO JE AJAX?

- Skup tehnologija, akronim od Asynchronous JavaScript and XML (2005.)
- Nedostatak JavaScripta bila je nemogućnost slanja podataka prema poslužitelju i primanje podataka s njega, nije se moglo spojiti na bazu podataka ili izvršiti kompleksne izračune; morale su se primjenjivati metode POST i GET – nakon što bi korisnik unio ili zatražio neke podatke, morao bi kliknuti na submit i nakon toga bi se stranica ponovo učitavala i prikazala odgovor sa poslužitelja, što je moglo potrajati...
- AJAX tehnologija pokušava zaobići i ublažiti ovaj problem tako da omogući komunikaciju s poslužiteljem pomoću posebnog JavaScript objekta XMLHttpRequest koji dopušta **pokupiti podatke ili informacije s poslužitelja, a da se stranica ne učitava ispočetka**

# HTML OBRAZAC NA KLIJENTSKOJ STRANI

```
<html>
<head>.....</head>
<body>
<form id="frmTime" method="POST" action="">
Vrijeme: <br />
<input type="text" name="txt_time" id="txt_time" value="">
</form>
</body>
</html>
```

Nema gumba submit, nego se korisniku kad klikne na poveznicu, u tekstualnom polju prikažu trenutni datum i sat, a da se pritom ne mora ponovno učitavati cijela stranica.

# XMLHTTPREQUEST OBJEKT ZA KOMUNIKACIJU

- Svi preglednici ne podržavaju isti način upotrebe AJAX-a
- Kako stvoriti osnovni AJAX objekt za Explorer, Operu, Firefox i Safari

```
<html><head>
<script type="text/javascript">// u datoteci index.html
function ajax()
{
var ajaxRequest;
try
{
// Opera, Firefox, Safari
ajaxRequest = new XMLHttpRequest();
}
catch (e)
{
try
{
// Internet Explorer
ajaxRequest = new ActiveXObject('Microsoft.XMLHTTP');
}
catch (e)
{
alert('Problem u AJAX komunikaciji');
return false;
}
}
}
// ovdje slijedi nastavak kôda... ajaxRequest.onreadystatechange=function()...
</script>
```

# SLANJE ZAHTJEVA POSLUŽITELJU I OBRADA ODGOVORA

- Da bismo mogli poslati podatke poslužitelju, treba stvoriti funkciju koja će moći primiti informacije koje će poslužitelj vratiti
- Objekt XMLHttpRequest ima svojstvo onreadystatechange koje u sebi sadrži funkciju koja će obraditi podatke poslane s poslužitelja, koristimo ju odmah nakon stvaranja variable ajaxRequest
- Drugo važno svojstvo je readyState (status poslužitelja: obrađuje zahtjev, preuzima zahtjev, u stanju završne obrade...). Svaki put kada se svojstvo readyState promjeni, izvršava se funkcija onreadystatechange
- Važno je stanje kada je server završio s obradom i vratio neke podatke, vrijednost svojstva readyState tada je 4

```
ajaxRequest.onreadystatechange = function()
{//programski kod
if(ajaxRequest.readyState == 4)
{//dohvat podataka
var element = document.getElementById('txt_time');
element.value = ajaxRequest.responseText; //sadržaj koji je vratio poslužitelj pospremili smo u tekstualno polje
}
}
```

# AJAX I PHP

- Slanje zahtjeva prema poslužitelju (php skripta vrijeme.php) i čekanje rezultata:
- <?php echo date(„d.m.Y. H:i:s”);?>
- Uz pretpostavku da su HTML kod i PHP skripta u istoj mapi na poslužitelju moguće je napisati kod za slanje zahtjeva (nakon do sada napisanog koda):

```
ajaxRequest.open('GET', 'vrijeme.php', true);
ajaxRequest.send(null); (nakon toga ide })</script>
```
- Kod ima dva koraka: specifikaciju naziva skripte koja se koristi i pozivanje metode send kojom se zahtjev šalje poslužitelju, nakon toga treba pozvati JavaScript funkciju ajax u href atributu poveznice s forme (kao što je navedeno) ili se koristi mogućnosti funkcije onclick
- <a href="" onclick=„ajax()”>Prikaži trenutno vrijeme</a>

# OPIS

- Kad se stranica prvi puta učita, prikazat će se jednostavna forma i poveznica
- Klikom na poveznicu aktivira sa JavaScript funkcija koja stvara novi objekt XMLHttpRequest
- Nakon uspješno kreiranog objekta šalje se zahtjev poslužitelju te se čeka njegov odgovor
- Nakon što stigne odgovor, rezultat se upisuje u tekstualno polje
- Sve se odvija na strani korisnika i pritom unutar preglednika ne dolazi do osvježavanja stranice

# SLANJE PODATAKA NA SERVER POMOĆU AJAX-A

- Navođenjem unutar URL-a u pozivu PHP skripte u metodi open objekta XMLHttpRequest

```
ajaxRequest.open("GET", "obrada.php?ime=Hrvoje&prezime=Horvat", true);
```

- Skripta obrada.php podatke iz URL-a prihvata pomoću superglobalne varijable \$GET:

```
<?php  
$ime = $_GET['ime'];  
$prez = $_GET['prezime'];  
// daljnja obrada na serveru  
?>
```

- Unutar URL-a ne smiju se upotrebljavati bilo koji znakovi, pa je podatke prije slanja preko URL-a potrebno enkodirati pomoću JavaScript funkcije escape:

```
var ime_prezime = escape('Hrvoje Horvat');  
ajaxRequest.open("GET", "obrada.php?ime_pres=" + ime_presime, true);  
<?php  
$ime_pres = $_GET['ime_pres'];  
// daljnja obrada na serveru  
?>
```

# PRIMJER OBRADE PODATAKA POMOĆU AJAX-A

- Slanje podataka s klijenta na server, te zatim obrađenih podataka nazad klijentu – unos imena u tekstualno polje. Klikom na gumb dodaj uneseno se ime šalje na server i tamo pretvara u velika slova. Obrađeno ime dodaje se kao novi redak u elementu divDodano

```
<html>
<head>
<title>ajax</title>
<script type="text/javascript">
function dodaj()
{
    xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function()
    {
        if (xmlhttp.readyState == 4)
        {
            document.getElementById('divDodano').innerHTML+=
xmlhttp.responseText + '<br>';
        }
    }
}
```

```
    }
};

ime = document.getElementById('txtIme').value;
ime = escape(ime);
xmlhttp.open('GET', 'obrada.php?ime=' + ime, true);
xmlhttp.send();
} <?php echo strtoupper($_GET['ime']);?>
</script>
</head>
<body>
<input type="text" id="txtIme"/>
<input type="button" onclick="dodaj()" value="Dodaj"/>
<div id="divDodano" style="background-color: Beige;
width: 200px;"></div>
</body>
</html>
```

# DOHVAĆANJE PODATAKA IZ MYSQL BAZE UPOTREBOM AJAX-A

```
<html><head></head><body>

<a href="javascript:dohvatiUcenike(1983)">1983</a>
<a href="javascript:dohvatiUcenike(1984)">1984</a>
<a href="javascript:dohvatiUcenike(1985)">1985</a>

<div id="divRezultat"></div>

</body></html>
```

Ideja: kada se klikne na određenu godinu, dohvate se sva imena učenika koji su te godine rođeni

Php skripta:

```
include("db_connection.php");

$god = $_GET["god"];

$query = "SELECT ime, prezime ";
$query .= "FROM ucenici ";
$query .= "WHERE datumRod >= '$god-01-01' ";
$query .= "AND datumRod <= '$god-12-31'";
$result = mysql_query($query) or die ("Greska u SQLu");

while($row = mysql_fetch_array($result)){
echo $row["ime"]." ".$row["prezime"];
echo '<br />';}
```

PHP kod datoteke db\_connection.php uključen u skriptu dohvatiUcenike.php

sadržava naredbe za povezivanje s MySQL bazom podataka:

```
$server = 'ime servera';
$username = 'korisničko ime';
```

```
$password = 'lozinka';
$database = 'ime baze';
$db = mysql_connect($server, $username, $password);
if($db){
$db_selected = mysql_select_db($database, $db);
if(!$db_selected)
echo 'Doslo je do pogreske kod odabira baze';
else
echo 'Doslo je do pogreske prilikom spajanja';
PHP skriptu dohvatiUcenike.php trebamo pozvati preko AJAX-a. JavaScript funkcija dohvatiUcenike zadužena za taj AJAX poziv mogla bi izgledati ovako:
function dohvatiUcenike(god){
xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function(){
if (xmlhttp.readyState == 4){
document.getElementById('divRezultat').innerHTML = xmlhttp.responseText + '<br>';
}};
xmlhttp.open("GET", "dohvatiUcenike.php?god=" + god,
true);
xmlhttp.send(null);}

Parametar god proslijeden je u poziv php skripte preko URL-a pomoću superglobalne varijable $_GET
ajaxRequest.open("GET", "dohvatiUcenike.php?god=" + god, true);
```