



xajax
PHP



Izradio: Davor Cihlar

Ajax - uvod

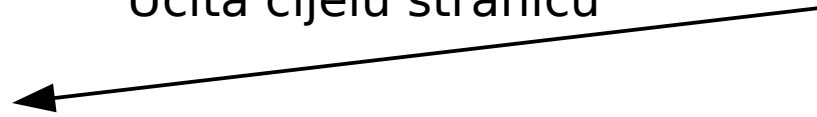
- ▶ Kratica od: Asynchronous JavaScript and XML (ali sad se ta tehnologija podržava i sinkrone pozive i ne samo XML)
- ▶ Omogućuje izradu web aplikacija
- ▶ Skripta na **klijentu** traži od servera samo dodatne podatke – ne osvježava se cijela stranica!
- ▶ Primjeri: Gmail, AHyCo, Facebook, ...
- ▶ [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))



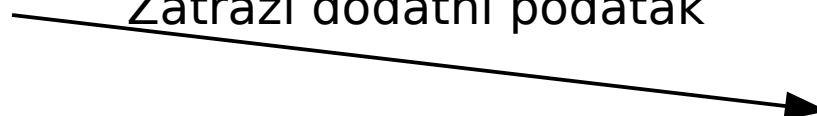
Ajax



Učita cijelu stranicu



Zatraži dodatni podatak



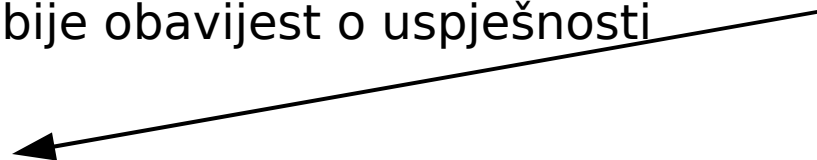
Dobije samo traženi podatak



Zatraži obnovu podataka u bazi



Dobije obavijest o uspješnosti



Firebug



The screenshot shows the Firebug XHR panel with a list of 15 network requests. The requests are to mail.google.com and include various GET and POST methods. A waterfall chart on the right shows the timing of each request, with bars indicating the duration of each. The status for all requests is 200 OK.

Method	URL	Status	Size	Time
GET	mail?ui=2&view=ss&ver=8ext	200 OK	23 KB	838ms
GET	mail?ui=2&ik=b386e7f61b&vi	200 OK	207 B	2.41s
POST	mail?ui=2&ik=b386e7f61b&v	200 OK	4 KB	1.94s
GET	mail?ui=2&view=jsm&name=l	200 OK	?	689ms
POST	mail?ui=2&ik=b386e7f61b&v	200 OK	829 B	661ms
POST	mail?ui=2&ik=b386e7f61b&v	200 OK	205 B	533ms
GET	mail?ui=2&view=jsm&name=c	200 OK	?	565ms
POST	mail?ui=2&ik=b386e7f61b&v	200 OK	2 KB	152ms
GET	mail?ui=2&view=jsm&name=c	200 OK	?	164ms
POST	mail?ui=2&ik=b386e7f61b&v	200 OK	970 B	377ms
POST	mail?ui=2&ik=b386e7f61b&v	200 OK	355 B	315ms
GET	mail?ui=2&view=jsm&name=c	200 OK	?	479ms
POST	mail?ui=2&ik=b386e7f61b&v	200 OK	2 KB	1.22s
GET	mail?ui=2&view=jsm&name=r	200 OK	?	744ms
GET	mail?ui=2&view=jsm&name=c	200 OK	?	641ms
GET	test?ui=2&ik=b386e7f61b&v	200 OK	?	1.06s

The screenshot shows the Firebug Headers, Post, and Response tabs for a POST request to multiply.server.php. The status is 200 OK and the domain is xajaxproject.org. The response is an XML document.

POST multiply.server.php 200 OK xajaxproject.org ?

Headers Post Response

```
<?xml version="1.0" encoding="utf-8" ?><xjx><cmd cmd="as" id="z" prop="value">N6</cmd></xjx>
```



Ajax biblioteke

- ▶ Osim direktnog korišćenja XMLHttpRequest iz Javascripta, imamo već gotove biblioteke koje nam uvelike olakšavaju razvoj:
 - ▶ jQuery
 - ▶ Dojo Toolkit
 - ▶ xajax
 - ▶ ...
- ▶ Detaljnije na [Wikipediji](#).



xajax

- ▶ xajax je biblioteka za PHP koja znatno olakšava korištenje Ajax tehnologije
 - ▶ Objektno orijentiran
 - ▶ Automatski generira Javascript
 - ▶ Pretpostavljeni encoding je utf-8
 - ▶ Moguće je transparentno pozivanje PHP funkcija iz web preglednika
- Primjerice, klikom na gumb, moguće je pozvati neku PHP funkciju na serveru, a bez da server generira ponovno kompletnu stranicu
- ▶ <http://xajaxproject.org/>



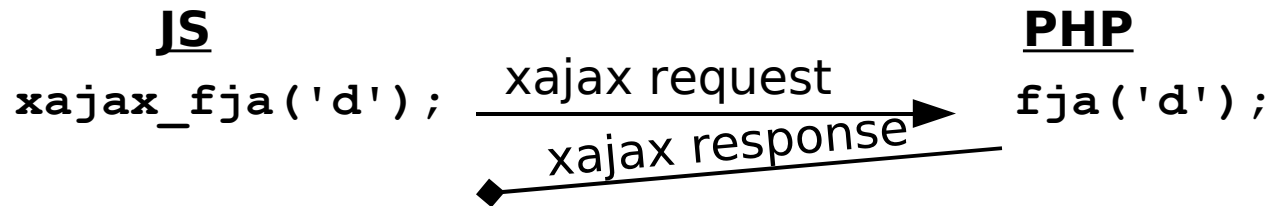
xajax – funkcije

- ▶ xajaxu odredimo koje PHP funkcije želimo omogućiti klijentu
- ▶ Za sve njih xajax generira Javascript *wrapper* funkcije koje zatim transparentno pozivaju svoju PHP funkciju

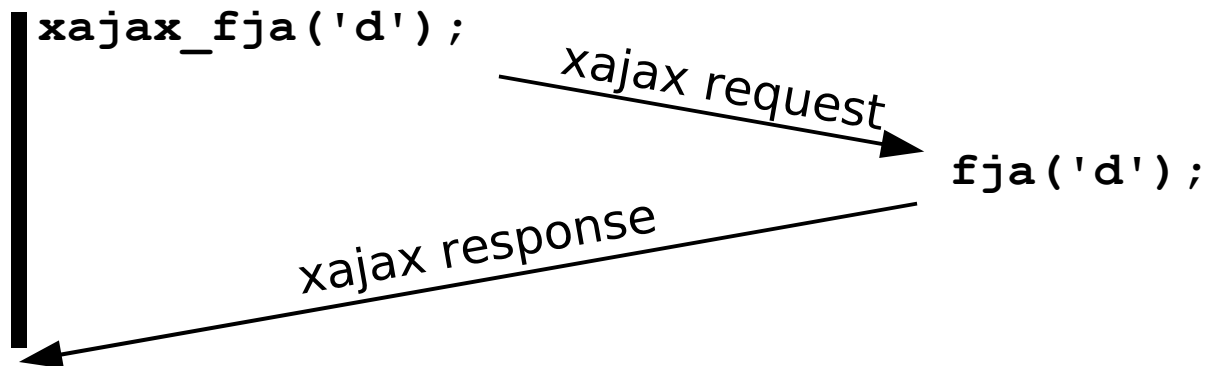


Tipovi funkcija

- ▶ Asinkrone funkcije:



- ▶ Sinkrone funkcije:
(klijentska funkcija čeka da se udaljena izvrši)



Pozivanje Javascript funkcija

- ▶ Javascript funkciju korisnik može pozvati na više načina:

- ▶ Linkom:

```
<a href="#" onclick="alert('hello');">
```

```
  Say hello!</a>
```

```
<a href="javascript:alert('hello');
```

```
  void(0);">Say hello!</a>
```

- ▶ Submit gumbom:

```
<input type="submit"
```

```
  onclick="alert('hello'); return false;">
```

```
(return false sprječava submit obrasca!)
```



Minimalni primjer

```
<?
require_once (' ./xajax/xajax_core/xajax.inc.php' );

function fja() { ... }

$xjx = new xajax();
$xjx->configure('javascript URI', './xajax/');
$xjx->register(XAJAX_FUNCTION, 'fja');
$xjx->processRequest();
?>
<html>
<head><? $xjx->printJavascript(); ?></head>
<body>
<input id="inp" type="submit"
        onclick="xajax_fja('d'); return false;">
</body></html>
```



Minimalni primjer

- ▶ Funkcija koju će klijent pozivati:

```
function fja() {  
    $resp = new xajaxResponse();  
  
    // akcije koje će se izvršiti na klijentu  
    $resp->assign('inp', 'value', 'Kliknut!');  
    $resp->...  
    ...  
  
    return $resp;  
}
```



Funkcije

- ▶ **new xajax()**
 - ▶ Instanciranje novog xajax objekta
- ▶ **configure('javascript URI', './xajax/')**
 - ▶ Odredimo gdje je instaliran kako bi znao generirati točnu referencu na javascript
- ▶ **register(XAJAX_FUNCTION, 'fja')**
 - ▶ Kažemo xajaxu da registriira funkciju *xajax_fja* na klijentskoj strani. Pozivom te Javascript funkcije, transparentno se poziva funkcija *fja* na serveru
 - ▶ Asinkrone funkcije su pretpostavljene!
 - ▶ Za sinkrone je potrebno dodati još jedan argument:
array('mode'=>'synchronous')



Funkcije

- ▶ **processRequest ()**

- ▶ Ako je došlo do xajax zahtjeva za pozivom neke funkcije, a ne do običnog učitavanja stranice, ova xajax funkcija će pozvati traženu funkciju i **završiti izvršavanje skripte!**

- ▶ **printJavascript ()**

- ▶ Uključi xajax Javascript biblioteku
- ▶ Generira Javascript koji definira sve prethodno registrirane funkcije



Funkcije za odgovor

- ▶ **new XMLHttpRequest()**

- ▶ Instanciranje novog odgovora na XMLHttpRequest zahtjev
- ▶ Koristi se samo unutar registrirane funkcije

- ▶ **assign(ID, ATRIBUT, VRIJEDNOST)**

- ▶ Pronađe HTML element sa oznakom id=ID, te mu postavi ATRIBUT na zadanu VRIJEDNOST
- ▶ Atribut može biti:
 - ▶ innerHTML
 - ▶ value
 - ▶ src
 - ▶ ...



Funkcije za odgovor

- ▶ **alert ('poruka')**
 - ▶ Prikaže informativni prozorčić s 'poruka' na klijentu
 - ▶ Korisno za debug
- ▶ **redirect (URL, ČEKANJE)**
 - ▶ Preusmjeri klijenta na URL nakon ČEKANJE sekundi
 - ▶ ČEKANJE je opcionalni argument. Pretpostavljena vrijednost je 0
- ▶ **setReturnValue (VRIJEDNOST)**
 - ▶ Vrijedi samo za sinkrone funkcije!



Debug

- ▶ Veliki problem za traženje grešaka predstavlja transparentnost poziva PHP funkcija
- ▶ Ne možemo vidjeti što se dogodilo na serveru
- ▶ Ako dođe do nekog ispisa unutar funkcije za xajax odgovor, klijent neće ništa vidjeti
- ▶ Kako bi smo prikazali detaljnije informacije što se događa s xajaxom, trebamo omogućiti debug način rada:

```
$xjx->configure( 'debug' , true );
```



Instalacija

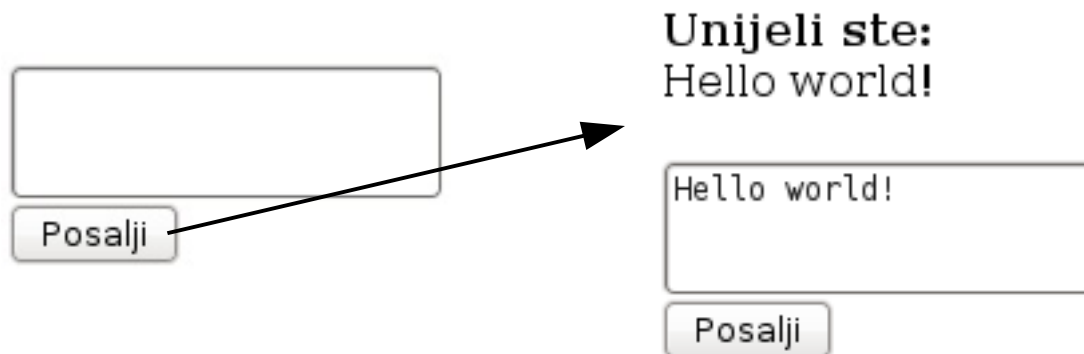
- ▶ Sa stranica <http://xajaxproject.org> preuzmite standardnu verziju
- ▶ Arhivu raspakirajte, te sadržaj kopirajte pomoću ftp-a u direktorij u kojem ćete vježbati (U Windows Exploreru napišite kao URL: `ftp://username@osnovephp.kset.org`)
- ▶ Npr. neka struktura direktorija izgleda ovako:
 - ▶ `/xvjezba/xajax` (biblioteka)
 - ▶ `/xvjezba/primjer.php`
 - ▶ `/xvjezba/zad1.php`
 - ▶ ...



Zadatak 1



- ▶ Napišite web aplikaciju u koju unesete HTML
- ▶ Kada korisnik klikne na gumb, na početak unesenog teksta ubacite podebljani redak "Unijeli ste:", te podatke zatim prikažite u span elementu pomoću innerHTML atributa
- ▶ Koristite sljedeći kod kako bi dohvatili podatak iz input elementa pomoću Javascripta:
`document.getElementById('broj').value`



Zadatak 2



- ▶ Koristeći isključivo xajax tehnologiju napišite web aplikaciju koja računa sumu brojeva
- ▶ Cijela lista brojeva mora biti stalno prikazana (Koristite innerHTML atribut i span element)
- ▶ Brojevi se moraju moći dodavati i brisati (klikom na broj koji želimo izbrisati)
- ▶ Kao spremište podataka koristite sjednice

1 2 4

Suma: 7

Dodaj



Zadatak 2 - kostur

```
...
<body>
<span id="brojke">
  <!-- automatski generirano: -->
  <a POZOVI BRISANJE NULTOG ELEMENTA>15</a>
  <a POZOVI BRISANJE PRVOG ELEMENTA>25</a>
  <!-- end of automatski generirano -->
</span><br>

<b>Suma:</b><span id="suma"><!-- automatski generirana suma:
40 --></span><br>

<input type="text" id="broj">
<input type="submit" value="Dodaj" onclick="POZOVI DODAVANJE
BROJA POHRANJENOG U ELEMENTU S ID-EM broj">

</body>
```



Automatsko generiranje JavaScript-a

- ▶ **register** funkcija razreda **xajax** vraća primjerak razreda **xajaxRequest** pomoću kojega možemo odrediti koje argumente će primiti funkcija za slanje zahtjeva, te generirati JavaScript poziv

- ▶

```
$rf =& $xjx->register(XAJAX_FUNCTION,  
                    'fja');  
$rf->printScript(); // ili getScript()  
▶ xajax_fja()
```



xajaxRequest::addParameter

- ▶ Pomoću te funkcije možemo odrediti koje će argumente imati poziv na JavaScript funkciju
- ▶ Postoji više tipova argumenata:
 - ▶ XAJAX_QUOTED_VALUE - string (konstanta)
 - ▶ XAJAX_INPUT_VALUE - podatak iz input elementa
 - ▶ XAJAX_CHECKED_VALUE - true ako je označen checkbox ili radio
 - ▶ XAJAX_FORM_VALUES - cijeli obrazac
 - ▶ XAJAX_JS_VALUE - JavaScript kod, numeričke konstante
- ▶ Poziv: **addParameter** (*TIP*, *STRING*)



xajaxRequest::useSingleQuote

- ▶ Ako se koristi bilo koji tip argumenta osim XAJAX_JS_VALUE, potrebno je odrediti da se koriste jednostruki navodnici!
 - ▶ U protivnom će se sukobiti dvostruki navodnici od HTML atributa s navodnicima od argumenta JavaScript funkcije
 - ▶ Funkciju je potrebno pozvati **prije** dodavanja argumenata

 - ▶ `$rf->useSingleQuote () ;`
-



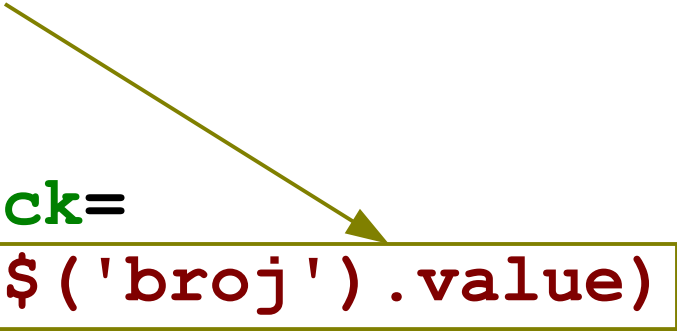
XAJAX_INPUT_VALUE

- ▶ Pogledajmo što radi sljedeći primjer:

```
$rf =& $xjx->register('fja');  
$rf->addParameter( XAJAX_INPUT_VALUE,  
                  'broj');  
  
$rf->useSingleQuote();  
echo '<a href="#" onclick='',  
      $rf->getScript(), '>dodaj</a>';
```

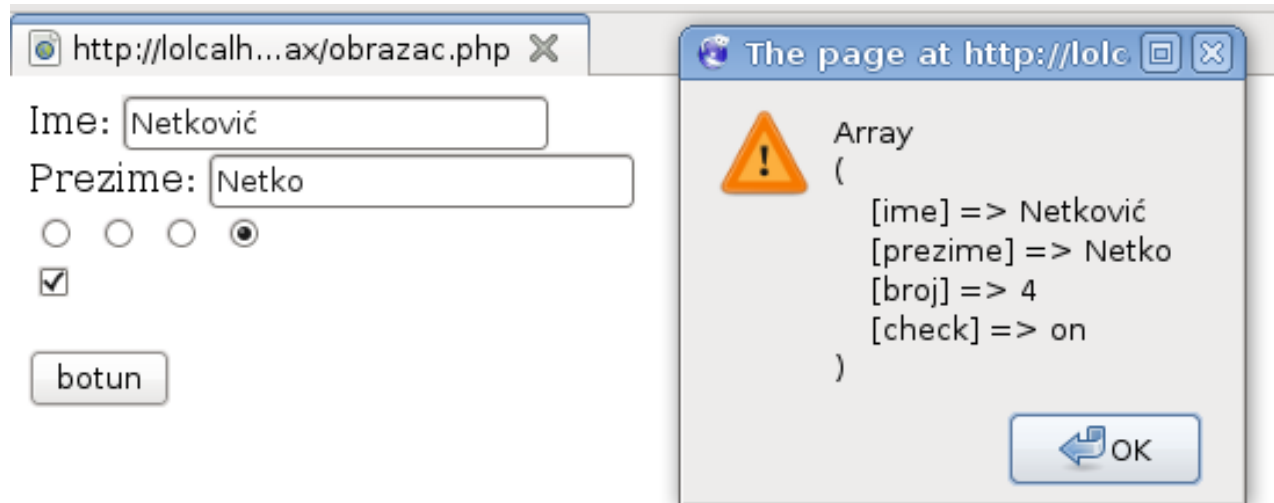
- ▶ Ispisat će:

```
<a href="#" onclick=  
"xajax_fja(xajax.$('broj').value)">...
```



XAJAX_FORM_VALUES

- ▶ Moguće je napraviti xajax zahtjev koji će sadržaj cijelog obrasca poslati na server kao samo jedno asocijativno polje
- ▶ `addParameter (XAJAX_FORM_VALUES , 'id_obrasca') ;`



Zabrana slanja obrasca



- ▶ form element ima akciju "onsubmit"
- ▶ Ako u toj akciji završimo s JavaScript-om "return false;", obrazac se neće poslati
- ▶ To nam omogućuje provjeru podataka prije slanja podataka!
- ▶ Primjer:
(xajax_check mora biti sinkrona funkcija!)
`<form method="POST" onsubmit="return xajax_check(sadržaj obrasca);">`



Odvajanje skripte za obradu zahtjeva

- ▶ Kod većih projekata možemo imati jako puno registriranih xajax funkcija
- ▶ Kako bi spriječili gomilanje funkcija, možemo odvojiti glavnu skriptu od skripte za obradu zahtjeva
- ▶ Podijelimo skriptu na 3 datoteke:
 - ▶ glavna.php – skripta sa samom stranicom
 - ▶ glavna.common.php – konfiguracija xajax-a
 - ▶ glavna.server.php – skripta za obradu zahtjeva



Odvajanje skripte za obradu zahtjeva

- ▶ xajax-u je potrebno javiti da mu se funkcije za obradu zahtjeva nalaze na nekom drugom mjestu:

```
$xjx = new xajax('stranica.server.php');
```

- ▶ Više primjera možete naći na <http://xajaxproject.org/en/docs-tutorials/>



Zadatak 3



- ▶ Napišite web aplikaciju za dijeljenje 2 broja
- ▶ Rezultat prikažite unutar span elementa
- ▶ Koristite tip argumenta `XAJAX_INPUT_VALUE`

- ▶ **Za napredne:**
Ako je korisnik kao drugi broj unio 0, ispišite mu poruku o grešci, te izbrišite drugi input.

- ▶ **Za vježbu:**
Isti zadatak riješite koristeći `XAJAX_FORM_VALUES`



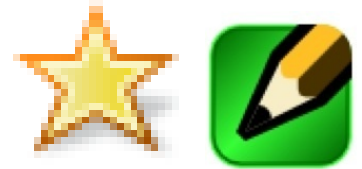
Zadatak 4



- ▶ Prepravite obrazac za prijavu videoteke
- ▶ Napišite asinkronu funkciju za provjeru prijave
 - ▶ Ako provjera nije uspješna, ispišite poruku o grešci (pomoću alert-a ili span elementa unutar obrasca)
 - ▶ U protivnom, postavite potrebne varijable sjednice, te preusmjerite stranicu na `index.php`



Zadatak 5



- ▶ Prepravite obrazac za dodavanje novog korisnika (administratora ili običnog)
 - ▶ Ako je uneseno korisničko ime već postojećeg korisnika, potrebno je zabraniti slanje obrasca, te ispisati poruku o grešci
 - ▶ U protivnom dopustiti slanje obrasca

