

Asinhroni JavaScript http zahtevi

Advanced JavaScript

ITAcademy

Asinhrono upravljanje serverskim resursima pomoću JavaScript-a

- Može se ostvratiti na minimum tri načina
 - **Asynchronous JavaScript And XML**
 - Long Pooling
 - WebSocket

Šta je AJAX?

- AJAX je skraćenica od: Asynchronous JavaScript and XML
- Ovo je nešto novija tehnologija kreiranja boljih, bržih i interaktivnijih web aplikacija.
- AJAX koristi HTML za sadržaj, CSS za stilizaciju kao i DOM i JavaScript za dinamičko upravljanje sadržajem.
- Konvencionalne web aplikacije razmenjuju informacije sa serverom koristeći se sinhronim zahtevima.
- Uz pomoć AJAX-a moguće je izvršiti komunikaciju sa severom tako da klijent ne primeti da je do bilo kakve komunikacije došlo (submit).

Šta je AJAX?

- XML je često korišćen tip podatka koji se dobavlja od servera, ali ne i jedini. Pored XML-a koristi se JSON, HTML pa čak i regularan tekst.
- AJAX je tehnologija koja je nezavisna od softwera na serveru.
- Korisnik može da nastavi sa korišćenjem aplikacije dok AJAX vrši komunikaciju sa serverom.
- Aktivacija AJAX-a može biti pokrenuta nakon registracije događaja koji je prirodan i intuitivan za korisnika.

AJAX

- Standard u JavaScriptu danas
- Koristi se za preuzimanje podataka sa servera na zahtev klijenta
- Najčešće se dovodi u blisku vezu sa klasom **XMLHttpRequest**

Rich Internet Application Technology

- AJAX je najkorišćenija Rich Internet Application (RIA) tehnologija.
- Trendovi razvoja veoma složenih web aplikacija sve više zahtevaju upotrebu tehnologija asinhronone komunikacije sa serverom te su tako i primeri upotrebe AJAX-a veoma česti u praksi.
- Sa druge strane, AJAX nekada može biti nekompatibilan sa starijim pretraživačima i zahteva podršku JavaScript-a što su njegove najveće slabosti.

AJAX i Open Standards

- AJAX je baziran na otvorenim standardima (Open Standards):
 - Prezentacija podataka je bazirana na samom pretraživaču i upotrebi HTML-a i CSS-a.
 - Podaci su smešteni u XML (JSON, HTML, text) formatu i prikupljeni sa servera.
 - Podaci se pribavljaju sa servera u pozadini uz pomoć **XMLHttpRequest** objekta.
 - Čitav proces pokreće JavaScript

Podrška za AJAX

- Napomena: AJAX ne podržavaju svi pretraživači.
- Lista najpoznatijih WEB pretraživača koji podržavaju AJAX:
 - Mozilla Firefox 1.0 i noviji.
 - Netscape version 7.1 i noviji.
 - Apple Safari 1.2 i noviji.
 - Microsoft Internet Explorer 5 i noviji.
 - Konqueror.
 - Opera 7.6 i noviji.
- Prilikom kreiranja aplikacija, imati u vidu pretraživače koji ne podržavaju AJAX.
- Kada se kaže da pretraživač ne podržava AJAX to znači da pretraživač jednostavno nije u stanju da kreira **XMLHttpRequest** objekat.

jQuery i AJAX

- jQuery u sebi ima metode za efikasno pozivanje asinhronone komunikacije bez potrebe da programer ima bilo kakvo znanje o **XMLHttpRequest** objektu
- Ne zavisi od browsera. jQuery ima u sebi ugrađene algoritme koji prepoznaju browser i kreiraju odgovarajući **XMLHttpRequest** objekat

Koraci

- Klijent registruje događaj.
- jQuery kreira XMLHttpRequest objekat.
- jQuery konfiguriše objekat.
- jQuery upućuje asinhroni zahtev ka Web serveru.
- Web server procesira zahtev i vraća rezultat.
- jQuery poziva callback() funkciju i procesira rezultat.
- HTML DOM se ažurira novim podacima.

jQuery metode

- .load()
- .get()
- .post()

.load()

- Osnovna sintaksa
 - `$("#div1").load("test.txt");`
 - `$("#div1").load("test.html #p1");`
- Proširena sintaksa
 - `$("#div1").load("test.html", function(responseTxt, statusTxt, xhr)){ }`

Primer

test.txt

Ovo je tekst u tekstualnoj datoteci

index.html

```
<html><head>
<script src="jquery-3.2.0.js "></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").load("test.txt");
    });
});
</script>
</head>
<body>
<div id="div1"><h2>jQuery i AJAX ce promeniti ovaj tekst</h2></div>
<button>Procitaj sadrzaj spoljne datoteke</button>
</body></html>
```

ITAcademy

Primer

test.html

```
<h2>jQuery i AJAX!!!</h2>
<p id="p1">Ovo je tekst u paragrafu.</p>
```

index.html

```
<html><head>
<script src="jquery-3.2.0.js "></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").load("test.html #p1");
    });
});
</script>
</head>
<body>
<div id="div1"><h2>jQuery i AJAX ce promeniti ovaj tekst</h2></div>
<button>Procitaj sadrzaj spoljne HTML datoteke</button>
</body></html>
```

ITAcademy

Primer

```
<html><head>
<script src="jquery-3.2.0.js "></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").load("test.html", function(odgovor, status, xhr){
            if(status == "success")
                alert("Ucitano!");
            if(status == "error")
                alert("Greska: " + xhr.status + ": " + xhr.statusText);
        });
    });
});
</script>
</head><body>
<div id="div1"><h2>jQuery i AJAX ce promeniti ovaj tekst</h2></div>
<button>Procitaj sadrzaj spoljne datoteke</button>
</body></html>
```

jQuery i PHP

ajax.php

```
<?php echo " <h2>Ovo je iz php-a</h2>";?>
```

index.html

```
<html><head>
<script src="jquery-3.2.0.js "></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").load("ajax.php");
    });
});
</script>
</head>
<body>
<div id="div1"><h2>jQuery i AJAX ce promeniti ovaj tekst koriscenjem PHP-a</h2></div>
<button>Procitaj sadrzaj spoljne datoteke</button>
</body></html>
```


Callback funkcije

- Problem sa jQuery-jem je vreme izvršavanja instrukcija
- Kada radimo sa animacijom i AJAX-om, postoji mogućnost da se prethodna instrukcija ne završi, a sledeća već da počne
- Rešenje tih problema su Callback funkcije
- Callback funkcija se dodaje iza parametara

Callback funkcije

Ne valja

```
$ ("p") .hide(2000);  
$ ("p") .show(2000);
```

Valja

```
$ ("p") .hide(2000, function() {  
    $ ("p") .show(2000);  
});
```

Callback funkcije

- Ista priča važi i za AJAX, mora se obezbediti da se AJAX zahtev završi pre nego što se nastavi sa izvršavanjem programa

AJAX GET metoda

- GET metoda, pored URL-a dokumenta koji treba da se izvrši, šalje i GET parametre
- Parametre čita skript jezik na serveru (u našem slučaju PHP), obrađuje ih i vraća rezultat jQuery-ju
- jQuery smešta rezultat na HTML stranicu

Sintaksa

- **`$.get(URL,data,function(odgovor,status,xhr),dataType)`**
 - *URL : putanja do datoteke koju pozivamo (obavezan)*
 - *data : podaci koje šaljemo (opcioni)*
 - *function(odgovor,status,xhr) : callback funkcija (opcioni)*
 - *dataType : tip odgovora (opcioni)*

Sintaksa

- Postoji nekoliko načina
 - `$.get("test.php?ime=Bosko&grad=Beograd");`
 - `$.get("test.php", { ime:"Bosko", grad:"Beograd" });`
 - `$.get("test.php", function(odgovor){
 alert("PHP je vratio: " + odgovor);
});`

Primeri

ajax2.php

```
<?php echo "ovo je iz PHP-a"?>
```

index2.html

```
<html>
<head>
  <script src="jquery-3.2.0.js"></script>
  <script>
    $(document).ready(function() {
      $("button").click(function() {
        $.get("ajax2.php", function(data, status) {
          $("div").html("Data: <b>" + data + "</b><br>Status: " + status);
        });
      });
    });
  </script>
</head>
<body>
  <button>Slanje HTTP GET zahteva</button>
  <div></div>
</body>
</html>
```

ITAcademy

Primeri

ajax3.php

```
<?php
    $ime=$_GET['ime'];
    $grad=$_GET['grad'];
    echo "Dobrodosao $ime, kako je u $grad?";
?>
```

index3.html

```
<html>
<head>
    <script src="jquery-3.2.0.js"></script>
    <script>
        $(document).ready(function() {
            $("button").click(function() {
                $.get("ajax3.php?ime=Bosko&grad=Beograd", function(data) {
                    $("div").html("<h2>" + data + "</h2><br>");
                });
            });
        });
    </script>
</head>
<body>
<button>Slanje HTTP GET zahteva sa parametrima</button>
<div></div>
</body>
</html>
```


Primeri

ajax3.php

```
<?php
    $ime=$_GET['ime'];
    $grad=$_GET['grad'];
    echo "Dobrodosao $ime, kako je u $grad?";
?>
```

index4.html

```
<html>
<head>
    <script src="jquery-3.2.0.js"></script>
    <script>
        $(document).ready(function() {
            $("button").click(function() {
                $.get("ajax3.php", {"ime": "Bosko", "grad": "Beograd"}, function(data) {
                    $("div").html("<h2>" + data + "</h2><br>");
                });
            });
        });
    </script>
</head>
<body>
<button>Slanje HTTP GET zahteva sa parametrima</button>
<div></div>
</body>
</html>
```

ITAcademy

Primeri

ajax3.php

```
<?php
    $ime=$_GET['ime'];
    $grad=$_GET['grad'];
    echo "Dobrodosao $ime, kako je u $grad?";
?>
```

index5.html

```
<html>
<head>
    <script src="jquery-3.2.0.js"></script>
    <script>
        $(document).ready(function() {
            $("button").click(function() {
                $.get("ajax3.php", { "ime": $("#tekst1").val(), "grad": $("#tekst2").val() }, function(data) {
                    $("div").html("<h2>" + data + "</h2><br>").fadeIn(2000);
                });
            });
        });
    </script>
</head>
<body>
<input type="text" id="tekst1" placeholder="Unesite ime"/><br>
<input type="text" id="tekst2" placeholder="Unesite grad"/><br>
<button>Slanje HTTP GET zahteva sa parametrizacijom iz text box-a</button>
<div style="display:none"></div>
</body>
</html>
```

ITAcademy

AJAX POST metoda

- POST metoda, pored URL-a dokumenta koji treba da se izvrši, šalje i POST parametre
- Parametre čita skript jezik na serveru (u našem slučaju PHP), obrađuje ih i vraća rezultat jQuery-ju
- jQuery smešta rezultat na HTML stranicu

Sintaksa

- **`$.post(URL,data,function(odgovor,status,xhr));`**
 - *URL : putanja do datoteke koju pozivamo(obavezan)*
 - *data : podaci koje šaljemo (opcionni)*
 - *function(odgovor,status,xhr) : callback funkcija (opcionni)*

Sintaksa

- Najčešći način
 - `$.post("ajax6.php",{"ime":"Bosko", "grad":"Beograd"}, function(data, status){....});`

Primeri

ajax6.php

```
<?php
    $ime=$_POST['ime'];
    $grad=$_POST['grad'];
    echo "Dobrodosao $ime, kako je u $grad?";
?>
```

index6.html

```
<html>
<head>
    <script src="jquery-3.2.0.js"></script>
    <script>
        $(document).ready(function() {
            $("button").click(function() {
                $.post("ajax6.php", {"ime": "Bosko", "grad": "Beograd"}, function(data, status) {
                    $("div").html("<h2>" + data + "</h2>Status: " + status);
                });
            });
        });
    </script>
</head>
<body>
<button>Slanje HTTP POST zahteva sa parametrima</button>
<div></div>
</body>
</html>
```

ITAcademy

ajax6.php

Primeri

```
<?php
    $ime=$_POST['ime'];
    $grad=$_POST['grad'];
    echo "Dobro dosao $ime, kako je u $grad?";
?>
```

index7.html

```
<html>
<head>
    <script src="jquery-3.2.0.js"></script>
    <script>
        $(document).ready(function() {
            $("button").click(function() {
                $.post("ajax6.php", {"ime":$("#tekst1").val(), "grad":$("#tekst2").val() }, function(data) {
                    $("div").html("<h2>" + data + "</h2><br>").fadeIn(2000);
                });
            });
        });
    </script>
</head>
<body>
<input type="text" id="tekst1" placeholder="Unesite ime"/><br>
<input type="text" id="tekst2" placeholder="Unesite grad"/><br>
<button>Slanje HTTP POST zahteva sa parametrizacijom iz text box-a</button>
<div style="display:none"></div>
</body>
</html>
```

JSON

- JavaScript **O**bject **N**otification
- Način prikaza objekata u tekstualnom formatu
- Ne zavisi od programskog jezika, platforme ili operativnog sistema

JSON

- Razmena podataka između klijenta i servera može biti samo tekst
- JSON je tekst i pomoću njega se može konvertovati bilo koji objekat u tekst i bilo koji JSON tekst u objekat korišćenjem JSON metoda

JSON metode (JS)

- Pretvaranje u JSON tekst

```
var obj = { "ime":"Bosko", "godine":41, "grad":"Beograd" };  
var myJSON = JSON.stringify(obj);  
alert(myJSON);
```

- Pretvaranje JSON teksta u objekat

```
var myJSON = '{ "name":"John", "age":31, "city":"New York" }';  
var obj = JSON.parse(myJSON);  
document.getElementById("demo").innerHTML = obj.name;
```

JSON metode (PHP)

- Pretvaranje u JSON tekst

```
<?php
$niz = array("Pera", "Laza", "Mika", "Joca");
$myJSON = json_encode($niz);
echo $myJSON;
?>
```

- Pretvaranje objekta iz JSON teksta

```
<?php
$niz=json_decode($myJSON);
echo $niz[0];
?>
```

jQuery, AJAX i JSON

- Jedan od načina komunikacije sa serverom je korišćenjem stringa u JSON formatu
- JSON je izuzetno koristan u ovakvim slučajevima jer je nezavisan od programskog jezika, a omogućava slanje objekata

Primeri

ajax8.php

```
<?php
$niz=["Pera", "Laza", "Mika", "Joca"];
$myJSON=json_encode($niz);
echo $myJSON;
?>
```

index8.html

```
<html>
<head>
  <script src="jquery-3.2.0.js"></script>
  <script>
    $(document).ready(function() {
      $("button").click(function() {
        $.get("ajax8.php", function(data) {
          alert(data);
          var niz=JSON.parse(data);
          $("div").html("<h2>" + niz[0] + "</h2><br>");
        });
      });
    });
  </script>
</head>
<body>
<button>Slanje HTTP GET zahteva i obrada JSON odgovora</button>
<div></div>
</body>
</html>
```

ITAcademy

Primeri

- ajax9.php

```
<?php
```

```
$db=mysqli_connect("localhost", "root", "");  
mysqli_select_db($db, "test");  
$sql="SELECT * FROM korisnici ORDER BY id asc;";  
$rez=mysqli_query($db, $sql);  
$niz=[];  
$niz=mysqli_fetch_all($rez, MYSQLI_ASSOC);  
$myJSON=json_encode($niz);  
echo $myJSON;  
?>
```

ITAcademy

Primeri

- index9.html

```
<html>
<head>
  <script src="jquery-3.2.0.js"></script>
  <script>
    $(document).ready(function() {
      $("button").click(function() {
        $.get("ajax9.php", function(data) {
          var niz=JSON.parse(data);
          var ispis="";
          for(i in niz)
            ispis+="id: "+niz[i]['id']+", ime="+niz[i]['ime']+"<br>";
          $("div").html(ispis);
        });
      });
    });
  </script>
</head>
<body>
<button>Slanje HTTP GET zahteva i obrada JSON odgovora iz baze podataka</button>
<div></div>
</body>
</html>
```