

jQuery – Rad sa AJAX osobinama

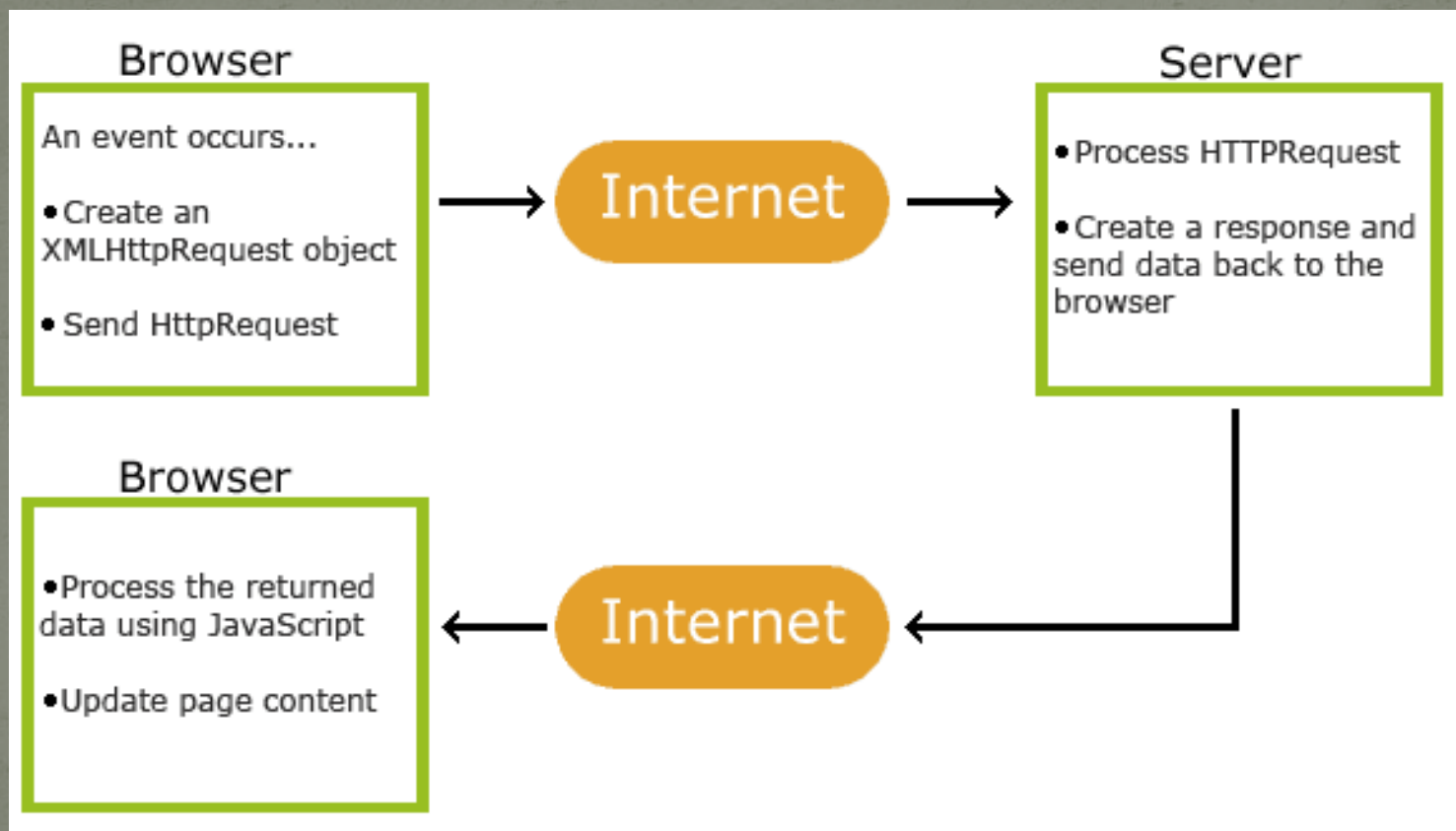
- AJAX predstavlja jednu od najvažnijih osobina koja je predstavljena u okviru browseru u poslednjoj deceniji
- **AJAX** – **A**synchronous **J**avaScript **A**nd **X**ML
- Omogućava integraciju podataka sa servera u naše aplikacije bez potrebe za osvežavanjem stranice , čak i samo delova stranice.
- AJAX nije programski jezik već skup web tehnologija na klijent strani

jQuery – Rad sa AJAX osobinama

- Koristi kombinaciju :
 - Ugradjenog **XMLHttpRequest** objekta u browseru - za dohvaćanje podataka sa servera (>IE7 , IE6 koristi ActiveX kontrolu)
 - **JavaScript** i **HTML DOM** – za prikaz i korišćenje podataka
- Sa AJAX-om možemo da radimo i sa XML i sa JSON formatima podataka

jQuery – Rad sa AJAX osobinama

- Kako AJAX radi :



jQuery – Rad sa AJAX osobinama

- Mi ćemo da koristimo jQuery funkciju za rad sa AJAX-om
- jQuery podržava rad sa XML, JSON i HTML
- jQuery vodi računa o verziji browsera i pozivu XMLHttpRequest objekta ili ActiveX kontrole
- Pogledajmo funkcije koje ćemo učiti :
 - 1.) `$(selektor).load()` – učitava HTML podatke sa servera
 - 2.) `$.get()` i `$.post()` – dohvata podatke sa servera
 - 3.) `$.getJSON()` – vraća JSON format podataka (get/post)
 - 4.) `$.ajax()` – dohvata podatke sa servera ali možemo više parametara da podešavamo

jQuery – Rad sa AJAX osobinama

1. Učitavanje HTML sadržaja sa servera - load() funkcija

- jQuery za dohvaćanje HTML podataka sa servera i dodavanje u DOM objekat koristi **load()** funkciju .

Sintaksa load funkcije :

Samo prvi parametar je obavezan

`$(selector).load(url, data, callback)`

- Load funkcija prima tri parametra , **url-putanju**, **data-podatke** koje šaljemo i **callback funkciju** – rezultat nakon slanja podataka (radimo samo sa prvim parametrom primer)

Primer :

```
$(document).ready(function() {  
    $('#HelpDugme').click(function() {  
        $('#IspisiHelp').load('HelpDetalji.html');  
    });  
});
```

Load uvek vraća html

Poziv funkcije koja učitava
HTML stranicu HelpDetalji

jQuery – Rad sa AJAX osobinama

- U okviru load() funkcije možemo da koristimo i selektor (kao filter) da dohvatimo samo deo HTML stranice definisane selektorom.

Primer :

```
$(document).ready(function() {  
    $('#HelpDugme').click(function() {  
        $('#IspisiHelp').load('HelpDetalji.html #Poglavlje1');  
    });  
});
```

PRIMER: [*jQuery-load.html*](#)



Učitavamo deo
HTML stranice HelpDetalji i
to samo deo stranice sa ID Poglavlje1

jQuery – Rad sa AJAX osobinama

ZADATAK:

Koristeći prethodni primer jQuery-load.html prikazati kursnu listu NBS na današnji dan .Koristiti sledeću adresu

<http://www.nbs.rs/kursnaListaModul/srednjiKurs.faces?lang=cir>

Dodati jedno dugme i nazvati ga **Napomena** sa id mojeDugme1 i na klik dugmeta prikazati deo stranice gore navedene i to samo klasu **form_napomena**

jQuery – Rad s AJAX osobinama

2. get() funkcija

- Za razliku od load funkcije koja vraća samo HTML tip sa get funkcijom možemo da vratimo JSON,XML,HTML,skript i string tipove podataka sa servera. Ona ustvari radi GET request ka serveru
- get() funkcija ima 4 parametra od kojih su prva tri ista kao i kod load funkcije. 4 parametar je tip podatka koji se očekuje od servera.
- Sintaksa get funkcije :
`$.get(url,data,callback,datatype)`
- get funkcija prima četiri parametra , url-putanju, data-podatke koje šaljemo serveru ,callback – rezultat nakon slanja podataka (sadrži podatke,status i xhr – XMLHttpRequest objekt) i tip podatka koji se očekuje od servera

jQuery – Rad sa AJAX osobinama

Pogledajmo primer get funkcije koji dohvata podatke html stranice (slično kao kod load funkcije) :

```
$.get('HelpDetalji.html', function(data) {  
    $('#OutputDiv').html(data);  
});
```

Koristimo prvi i treći parametar

PRIMER : *jQuery-getKaoLoad.html*

- Funkcija get() koristeći **drugi parametar** vrši slanje podataka serveru kao kod submit forme.
- **Treći parametar tj. callback** funkcija služi za prijem povratnih vrednosti od servera i ima tri parametra – podatke, status i xhr objekat.
- Dobra praksa je da se koriste **metode done() i fail()** koje se aktiviraju ukoliko je status dobar tj. **success** odnosno ako ne status je **error**.

jQuery – Rad sa AJAX osobinama

- pogledajmo primer get funkcije sa prosleđivanjem podataka serveru

PRIMER : *jQuery-get.html*

- Ukoliko želimo da koristimo get funkciju samo za dobijanje JSON podataka od servera možemo da koristimo funkciju `getJSON()` koja vraća upravo ovakve podatke.

- Sintaksa funkcije `getJSON()`:

`$.getJSON(url, data, callback)`

- Za razliku od `get()` funkcije `getJSON` nema četvrti parametar to je jedina razlika

PRIMER 2 : *jQuery-getJSON.html*

jQuery – Rad sa AJAX osobinama

3. post() funkcija

- Nedostaci GET request je keširanje podataka u browseru kao i količina podataka koje možemo da pošaljemo ka serveru.
- Ove nedostatke rešava POST request , koji ne kešira podatke u browser i omogućava slanje veće količine podataka kroz forme.
- Zato ćemo i koristiti jQuery funkciju post() koja upravo radi POST request ka serveru
- Sintaksa post() funkcije:
`$.post (url, data, callback, datatype)`
- Broj , raspored i namena parametara je ista kao kod get() funkcije

jQuery – Rad sa AJAX osobinama

3. post() funkcija (2)

- Pogledajmo primer post() funkcije:

PRIMER: *[jQuery-post.html](#)*

jQuery – Rad sa AJAX osobinama

4. ajax() funkcija

- U suštini load,get,getJSON i post funkcije su ustvari AJAX funkcije napisane da rade samo određen deo funkcionalnosti
- AJAX funkciju koristimo da bi obezbedili dodatnu kontrolu kada koristimo ajax pozive ka serveru
- Za razliku od ostalih funkcija ima dodatne proprietije koje možemo da koristimo prilikom poziva AJAX funkcije.

Sintaksa AJAX funkcije :

`$.ajax(url [,JSON properties]);`

jQuery – Rad sa AJAX osobinama

- JSON svojstvi :

- contentType - koji tip prosleđujemo serveru (boolean ili String)
(default: 'application/x-www-form-urlencoded; charset=UTF-8'
može biti multipart/form-data, ili text/plain
- data - podaci koji će biti poslani serveru (objekat , String ili niz)
- dataType – tip podatka koji očekujemo da se vrati od servera
(xml,json,html) – (String)
- error – ako se desi greška poziva se funkcija koja razrešava grešku.
Ova funkcija prima 3 argumenta (jqXHR, status, error) . Status je
String tipa i može da bude "timeout", "error" , "abort" i
"parsererror". (funkcija)
Kada se desi HTTP greška error sadrži HTTP status "Not found" ili
"Internal Server Error"

jQuery – Rad sa AJAX osobinama

- JSON propertiji :

- **success** – funkcija koja će biti pozvana ako je zahtev bio uspešan. Prosleđuje tri argumenta (data, textStatus, jqXHR). data su podaci koje vrati server, status i jqXHR objekat

Primer (od v 3.0 success, error i complete funkcije su izbačene):

```
$.ajax({  
  url: '../CustomerService.svc/InsertCustomer',  
  data: customer,  
  dataType: 'json',  
  success: function(data, status, xhr){  
    alert("Unesi status:" + data.d.Status + '\n' + data.d.Message);  
  },  
  error: function(xhr, status, error){  
    alert('Error occurred: ' + status);  
  }  
});
```

← JSON objekat – key:value

← REST API, web servis,

← Šaljemo podatke serveru – objekat customer

← Server vraća podatke u JSON formatu

← Kod servisa svc WFC ceo data se umota u d

← Ako server vrati success uzimamo podatke

← Ako server vrati error callback ispisujemo grešku

jQuery – Rad sa AJAX osobinama

Od verzije 1.5 uvodi se **promise interfejs** i dodaju se sve metode i svojstvima. Ove metode uzimaju jedan ili više argumenata funkcije koje su pozvane nakon što se \$.ajax() zahtev završi. Ovo omogućava dodelu više callbackova po jednom zahtevu.

Metode koje se koriste umesto success, error i complete su done, fail i always

Primer :

```
$.ajax({  
    url: '../CustomerService.svc/InsertCustomer',  
    data: customer,  
    dataType: 'json'  
}).done(function(data, status, xhr) {  
    alert("Unesi status:" + data.d.Status + '\n' + data.d.Message);  
}).fail(function(xhr, status, error){  
    alert( "error" );  
});
```

Metod done i fail se dodaju nakon ajax funkcije

jQuery – Rad sa AJAX osobinama

- U sledećem primeru pozivamo WFC servis (.NET servis) i prosleđujemo mu objekat Customer i očekujemo potvrdu da je podatak iz forme uspešno unet

PRIMER : *[jQuery-ajax.html](#)*

- U sledećem primeru koristeći AJAX funkciju pozivamo API strukturu sa get tipom bez prosleđivanja podataka
- Pre toga prikažimo API servis sa podacima i izanalizirajmo ga :

<https://api.meetup.com/2/cities>

PRIMER 2 : *[jQuery-ajax-api.html](#)*

jQuery – Rad sa AJAX osobinama

PRIMER ZA KRAJ 😊 : [jQuery-ajax-puppies.html](http://jquery-ajax-puppies.html)