

# INTERNET TEHNOLOGIJE I VEB DIZAJN

Upotreba AJAX-a iz jQuery biblioteke

predavač: dr Miloš Antonijević

# Uvod u JSON

- **JSON** je skraćenica od JavaScript Object Notation, odnosno, predstavlja model za predstavljanje strukture podataka nezavisan od platforme i programskog jezika.
- Predstavlja svojevrsnu alternativu XML-u.
- Obzirom da se u JSON-u može opisati svaka struktura podataka, pogodan je za razmenu informacija između klijenta i server-a.
- Svi moderni programski jezici (serverski i klijentski) imaju funkcije za konvertovanje JSON oblika u određenu formu podataka i obrnuto.
- Podaci se u JSON-u predstavljaju u vidu parova **name:value**, pri čemu su objekti označeni sa vitičastim zagradama { }, a nizovi sa [ ].
- Podatak tipa String se piše pod navodnicima, a podatak tipa broj se zahteva navodnike.

# Primer u JSON

```
{  
  studenti: [  
    { "ime": "Petar", "prezime": "Petrovic", "godiste": 2000 }  
    { "ime": "Jovan", "prezime": "Jovanovic", "godiste": 2002 }  
    { "ime": "Milos", "prezime": "Milosevic", "godiste": 2002 }  
  ],  
  ustanova: { "univerzitet": "Singidunum", "fakultet": "tf" },  
  broj_telefona: "011/123-456-789"  
}
```

# JSON vs. XML

Sličnosti	Razlike
JSON i XML su u formi običnog teksta.	JSON ne poseduje tagove u svojoj sintaksi
JSON i XML poseduju hijerarhijsku strukturu	JSON je kraći od XML-a i brže se čita i upisuje
JSON i XML su nezavisni od programskog jezika	JSON može da opisuje nizove podataka
JSON i XML se mogu koristiti za AJAX komunikaciju	JSON ne koristi rezervisane reči

# XML Primer

```
<?xml version='1.0' encoding='UTF-8'?>
<card>
  <fullname>Sean Kelly</fullname>
  <org>SK Consulting</org>
  <emailaddr>
    <address type='work'>kelly@seankelly.biz</address>
    <address type='home'>kelly@seankelly.tv</address>
  </emailaddr>
  <telephones>
    <tel type='work'>+1 214 555 1212</tel>
    <tel type='fax'>+1 214 555 1213</tel>
    <tel type='mobile'>+1 214 555 1214</tel>
  </telephones>
  <addresses>
    <address type='work' format='us'>1234 Main StSpringfield</address>
    <address type='home' format='us'>5678 Main StSpringfield</address>
  </addresses>
</card>
```

# JSON Primer

```
{  
  "fullname": "Sean Kelly",  
  "org": "SK Consulting",  
  "emailaddrs": [  
    {"type": "work", "value": "kelly@seankelly.biz"},  
    {"type": "home", "pref": 1, "value": "kelly@seankelly.tv"}  
  ],  
  "telephones": [  
    {"type": "work", "pref": 1, "value": "+1 214 555 1212"},  
    {"type": "fax", "value": "+1 214 555 1213"},  
    {"type": "mobile", "value": "+1 214 555 1214"}  
  ],  
  "addresses": [  
    {"type": "work", "format": "us", "value": "1234 Main StnSpringfield"},  
    {"type": "home", "format": "us", "value": "5678 Main StnSpringfield"}  
  ]  
}
```

# Pristup podacima JSON objekta

- Svojstvima objekta kreiranog u JavaScript-u pristupa se na način da se naziv objekta i naziv svojstva odvajaju tačkom.

```
var student = { "ime":"Jovan", "prezime":"Jovanovic", "godiste":2002 };
```

```
alert("ime studenta je: " + student.ime);
```

```
var fakulteti = { [ "naziv":"singi", "podaci": { "lokacija":"",  
"telefon":"+381111234567" } ] };
```

```
alert("telefon fakulteta je: " + fakulteti[0].podaci.telefon);
```

- Svojstvo se može ukloniti iz objekta korišćenjem operatora **delete**.

```
var employee = { firstname: 'John', lastname: 'Doe' };
```

```
console.log(Employee.firstname);
```

```
delete employee.firstname;
```

```
console.log(Employee.firstname);
```

# Korišćenje JSON-a za komunikaciju sa serverom

- HTTP protokolom se podaci razmenjuju u tekstualnom formatu. Iz toga razloga, objekat definisan u JS-u se pre slanja konvertuje u JSON oblik.
- Za potrebe konverzije JS objekta u JSON oblik koristi se funkcija **stringify()** klase **JSON**.
- Obrnuti proces, konverzija String-a u JS objekat obavlja se upotrebom funkcije **parse()** klase **JSON**.

```
var fakulteti = { [ "naziv":"singi", "podaci": { "lokacija":"",  
"telefon":"+381111234567" } ] };  
var stringovano = JSON.stringify(fakulteti);  
console.log(stringovano);  
var kopija_fakulteti = JSON.parse(stringovano);  
console.log(kopija_fakulteti.naziv);
```



# JSON Primer tabele

```
<script>
    function prikaz_tabele() {
        studenti= [
            { "ime":"Petar", "prezime":"Petrovic", "godiste":2000 },
            { "ime":"Jovan", "prezime":"Jovanovic", "godiste":2002 },
            { "ime":"Milos", "prezime":"Milosevic", "godiste":2002 }
        ];
        var tabela="<table border=1>";
        for (var i=0; i<studenti.length; i++) {
            tabela+="<tr><td>" + studenti[i].ime + "</td><td>" +
                studenti[i].prezime + "</td><td>" +
                studenti[i].godiste + "</td></tr>";
        }
        tabela+="<table>";
        document.getElementById("tabela").innerHTML=tabela;
    }
</script>
```

# Korišćenje JSON-a za AJAX komunikaciju sa serverom iz jQuery biblioteke

- AJAX se iz jQuery-ja može koristiti na dva načina:
  - Upotrebom **\$.get()** i **\$.post()** metoda:

```
$.get("app/list.php", { "id": "1" }, function(data) { //obrada podataka });
```

- Upotrebom **\$.ajax()** metode:

```
$.ajax({  
    type: "POST",  
    url: "app/add.php",  
    data: json_obj,  
    dataType: "json",  
    timeout: 600000,  
    success: function (data) { // podatak je prenet serveru i server nije vratio  
gresku },  
    error: function (e) {      // server je vratio gresku alert(e.responseText); }  
});
```

# Primer direktnog pribavljanja JSON fajla

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $.getJSON("neki_json.js", function(data){
            $.each(data, function(i, field){
                $("div").append(field + " ");
            });
        });
    });
});
</script>
<body>
    <button>Dohvati JSON podatke</button>
    <div></div>
</body>
```

**PITANJA?**