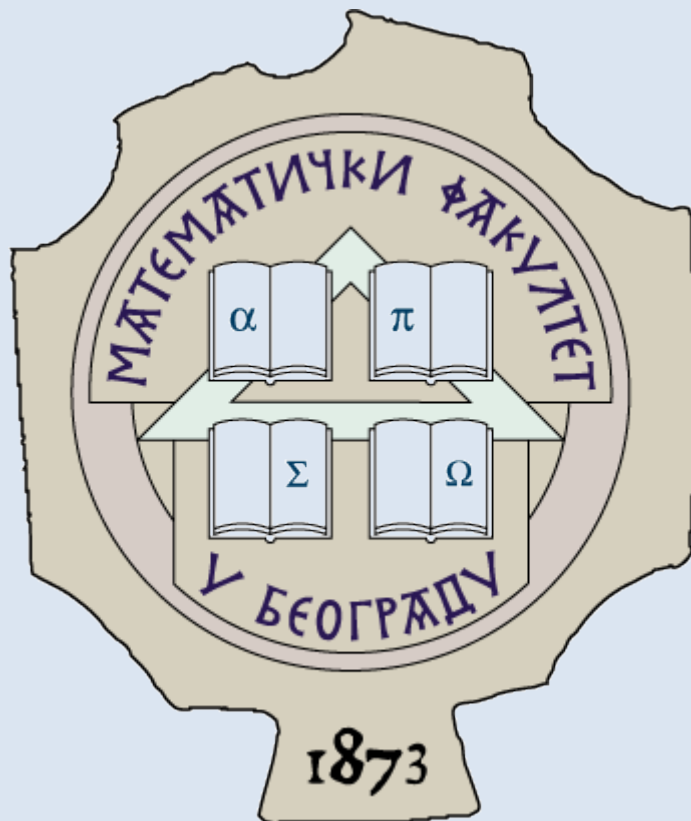


JSON

СЕМИНАРСКИ РАД



МАТЕМАТИЧКИ ФАКУЛТЕТ
УНИВЕРЗИТЕТ У БЕОГРАДУ

СТУДЕНТ: АТАНАСОВСКИ ПЕТАР

БРОЈ ИНДЕКСА: 1032/2012

ПРЕДМЕТ: МЕТОДОЛОГИЈА СТРУЧНОГ И НАУЧНОГ РАДА

ШКОЛСКА ГОДИНА: 2012/2013

ПРОФЕСОР: ДР ВЛАДИМИР ФИЛИПОВИЋ

САДРЖАЈ

Садржај	2
JSON - увод	3
Историја	6
JSON схема	7
Internet media type	8
JSON-RPC	8
JSON и Ајах	10
Поређења са другим форматима	11
YAML	11
XML	11
Примери	11
JSON пример	11
YAML пример	12
XML примери	12
Литература	13

JSON - УВОД

JSON (JavaScript Object Notation) је формат за размену података који троши мало ресурса. Једноставан је за обраду, тачније за људе да га читају и пишу, и за машине да га парсирају и генеришу. Базира се на подскупу JavaScript програмског језика, стандард ECMA-262 треће издање – децембар 1999.



JSON је текстуални формат који је у потпуности језички независан, али користи правила која су блиска програмерима С фамилије језика, што укључује С, С++, С#, Java, JavaScript, Perl, Python, али и многе друге. Ова својства чине JSON идеалним језиком за размену података.

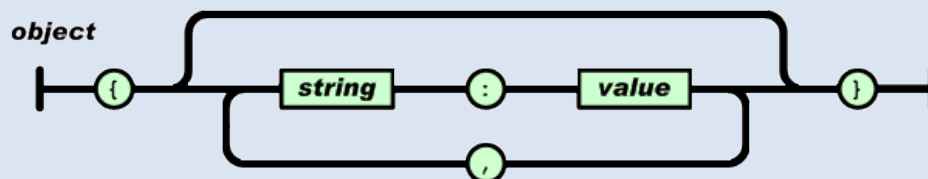
JSON се базира на две структуре:

- Колекција парова имена/вредности. У различитим језицима, ово се реализује као објекат, слог, структура, речник, хеш табела, листа кључева, или асоцијативни низ.
- Уређена листа вредности. У највећем броју језика, ово се реализује као низ, вектор, листа, или секвенца.

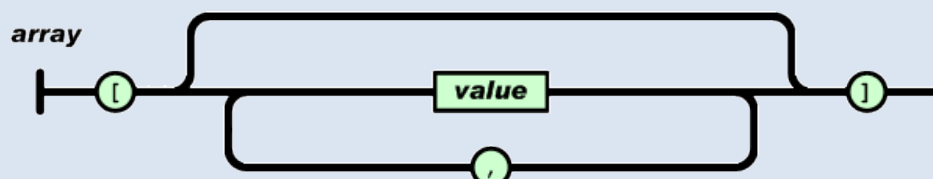
Ово су универзалне структуре података. Сви модерни програмски језици подржавају их у некој форми. Управо из тог разлога и има смисла да формат за размену података буде базиран на тим структурама.

У JSON-у, они су у овим облицима:

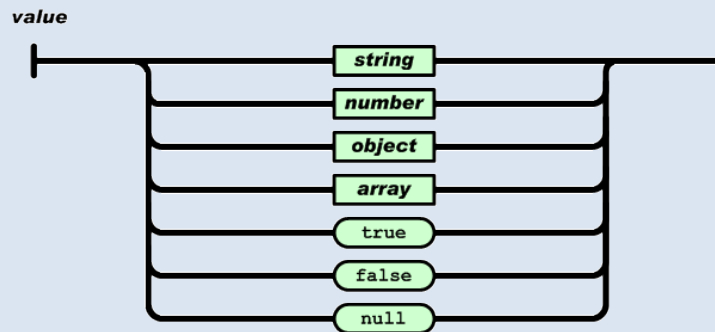
- **Објекат** је несређени скуп име/вредност парова. Објекат почиње са „{“ и завршава се са „}“. Свако име следи „:“. Име/вредност парови су раздвојени зарезом.



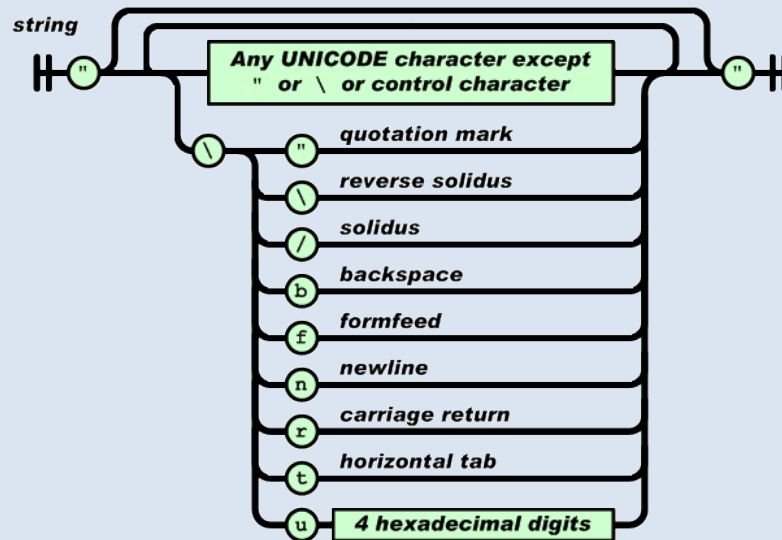
- **Низ** је уређена колекција вредности. Низ почиње са „[“, а завршава се са „]“. Вредности су раздвојене зарезом, као што је случај код објеката.



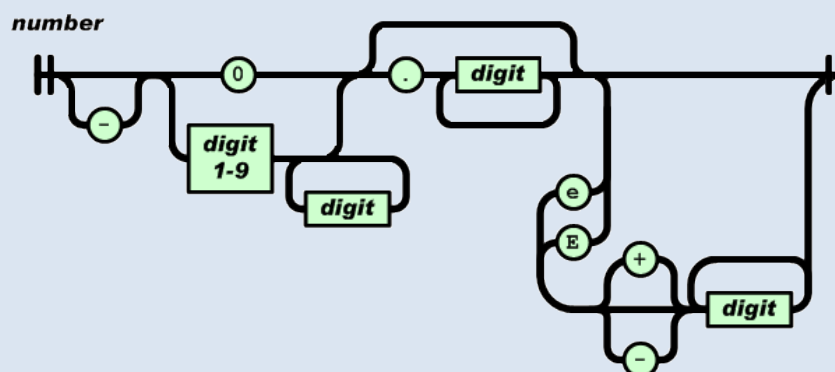
- **Вредност** може бити ниска у наводницима, или број или true или false или null или објекат или низ. Те структуре могу бити и угњежђене.



- **Ниска** је низ од нула или више Unicode знакова, ограђена двоструким наводницима, и користи се обрнута коса црта за изузетке. Један знак се представља као ниска од једног знака. Ниска је веома сличана ниски у језицима C или Java.



- **Број** веома личи на број у језицима C или Java, осим што се октални и хексадецимални формат не користи.



- **Белине** могу да се уметну између било ког пара симбола.

Наредни пример показује JSON репрезентацију објекта који описује особу. Објекат има ниску карактера за име и презиме, број за године, објекат који представља адресу особе и низ објеката за број телефона.

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": 10021
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "fax",
      "number": "646 555-4567"
    }
  ]
}
```

Потенцијална замка слободне природе JSON-а долази уз могућност да се бројеви представљају као нумерички литерали, или ниска између наводника. Како је JSON готово подскуп JavaScript-а, не препоручује се, али је могуће парсирати JSON текст у објекат позивањем функције `eval()`.

```
var p = eval("(" + contact + ")");
```

Променљива `contact` мора бити у загради да би се избегла двосмисленост синтаксе JavaScript-а. Препоручује се коришћење JSON парсера. Исправно имплементиран JSON парсер прихвата само валидан код, избегавајући потенцијални злонамерни део да буде извршен.

```
var p = JSON.parse(contact);
```

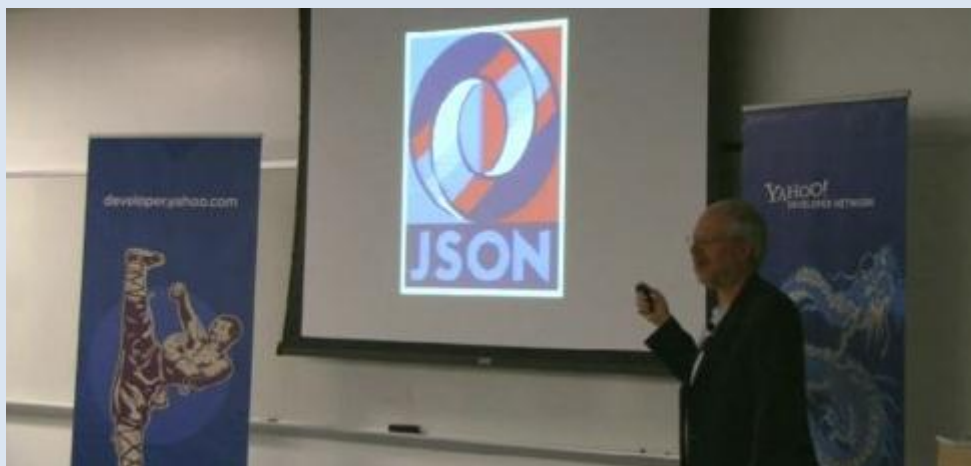
Новији претраживачи укључују специјалне функције за парсирање JSON-а. Подршка претраживача је далеко сигурнији и ефикаснији начин од функције `eval()`.

ИСТОРИЈА

Прва особа која је заслужна за дефинисање и популаризацију JSON формата је Даглас Крокфорд, амерички програмер који је постао познат по константном доприносу развоја језика JavaScript. Поред самог JSON формата развио је и многе друге алате, од којих су најпознатији JSLint и JSMIn. Тренутно је сениор JavaScript програмер у PayPal-у.

JSON се користио у State Software Inc., компанији чији је сувласник Крокфорд, покренутој у априлу 2001., и основане од стране Tesla Ventures. Када је фирма основана почетком 2001. од стране шест бивших запослених Communities.com, сложили су се да изграде систем који ће користити могућности стандардног прегледача и омогућити апстрактни слој за Веб програмере да креирају Веб апликације са перзистентним дуплекс конекцијама на Веб сервер одржавајући две http конекције отвореним и бришући их пре него што прегледач пређе у стање паузе уколико се више подаци не размењују. Идеја State Application Framework-а је развијена од стране Чип Морнингстара у State Software-у. Коришћена је као део пројекта у Communities.com за Cartoon Network, који је користио plug-in са адекватним форматом порука за манипулисање DHTML елементима. Након открића раних AJAX могућности, digiGroups, Noosh, и други су користили оквири за пренос информација у визуално поље корисничког прегледача без освежавања садржаја Веб апликације, користећи само стандарде HTTP, HTML и могућности JavaScript-а у Netscape 4.0.5+ и IE 5+. Даглас Крокфорд је затим открио да би JavaScript могао да се користи као објектно оријентисан формат за размену порука на таквом систему. Систем је продат компанијама Sun Microsystems, Amazon.com и EDS. JSON.org сајт је покренут већ 2002. У децембру 2005. Yahoo! Почине да нуди неке од својих Веб сервиса у JSON-у. Google то чини JSON за свој GData веб протокол у децембру 2006.

Иако је JSON оригинално осмишљен као не тако стриктни подскуп JavaScript језика (тачније, Standard ECMA-262 о коме је већ било речи) и често се користио са овим језиком, представља језички независан формат података. Код за парзирање и генерисање JSON података је доступан за велики број програмских језика. JSON сајт пружа обимни списак постојећих JSON библиотеке, распоређених по програмским језицима.



Даглас Крокфорд на једном од својих предавања

JSON СХЕМА

JSON схема је спецификација за формат базиран на JSON-у и дефинише структуру JSON података. Ова схема обезбеђује правила шта JSON подаци морају поседовати за дату апликацију и како се могу мењати. Схема тежи обезбеђивању валидације, документације, и контролу интеракције између података. Базира се на концепту XML схеме, RelaxNG, и Kwalify.

JSON схема је писана као Internet-Draft, тренутна верзија је 4. Тренутно постоји неколико валидатора за различите програмске језике, сваки са засебним нивоом усаглашености.

Пример JSON схеме:

```
{
  "name": "Product",
  "properties": {
    "id": {
      "type": "number",
      "description": "Product identifier",
      "required": true
    },
    "name": {
      "type": "string",
      "description": "Name of the product",
      "required": true
    },
    "price": {
      "type": "number",
      "minimum": 0,
      "required": true
    },
    "tags": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "stock": {
      "type": "object",
      "properties": {
        "warehouse": {
          "type": "number"
        },
        "retail": {
          "type": "number"
        }
      }
    }
  }
}
```

Ова схема се може користити за валидацију следећег кода:

```
{
  "id": 1,
  "name": "Foo",
  "price": 123,
  "tags": [ "Bar", "Eek" ],
  "stock": {
    "warehouse": 300,
    "retail": 20
  }
}
```

INTERNET MEDIA TYPE

Internet media type представља двострани идентификатор за формате докумената на интернету. Идентификатори су били примарно дефинисани у оквиру RFC 2046 за коришћење приликом слања мејлова кроз SMTP, али се њихова употреба проширила и на друге протоколе, као што су HTTP, RTP и SIP. Користио се назив MIME типови (Multipurpose Internet Mail Extensions).

Званични назив MIME типа за JSON је **"application/json"**.

JSON-RPC

Remote procedure call (RPC) је комуникација између процеса која омогућава програму да изазове потпрограму или процедуру која ће се извршавати у другом адресном простору (често и на другом рачунару на дељеној мрежи), без нужне потребе да се програмирају детаљи овако удаљене интеракције. Тачније, програмер пише исти код у сваком случају.

JSON-RPC је remote procedure call (RPC) протокол програмиран у JSON-у. Представља веома једноставан протокол (и веома је сличан XML-RPC), дефинишући само неколико типова података и команди. JSON-RPC подржава нотификације (информације послате серверу које не захтевају одговор) и вишеструке позиве да се шаљу серверу са могућношћу за одговор мимо редоследа.

У наставку су одређени примери коришћења JSON-RPC 2.0 (актуелна верзија).

Процедура позвана са позиционим параметрима:

```
--> {"jsonrpc": "2.0", "method": "subtract", "params": [42, 23], "id": 1}
```

```
<-- {"jsonrpc": "2.0", "result": 19, "id": 1}
```

```
--> {"jsonrpc": "2.0", "method": "subtract", "params": [23, 42], "id": 2}
```

```
<-- {"jsonrpc": "2.0", "result": -19, "id": 2}
```


Процедура позвана са именованим параметрима:

```
--> {"jsonrpc": "2.0", "method": "subtract", "params": {"subtrahend": 23, "minuend": 42}, "id": 3}
<-- {"jsonrpc": "2.0", "result": 19, "id": 3}
```

```
--> {"jsonrpc": "2.0", "method": "subtract", "params": {"minuend": 42, "subtrahend": 23}, "id": 4}
<-- {"jsonrpc": "2.0", "result": 19, "id": 4}
```

Нотификације:

```
--> {"jsonrpc": "2.0", "method": "update", "params": [1,2,3,4,5]}
```

```
--> {"jsonrpc": "2.0", "method": "foobar"}
```

Позив непостојеће процедуре:

```
--> {"jsonrpc": "2.0", "method": "foobar", "id": 10}
<-- {"jsonrpc": "2.0", "error": {"code": -32601, "message": "Procedure not found."}, "id": 10}
```

Позив неисправног JSON-а:

```
--> {"jsonrpc": "2.0", "method": "foobar", "params": "bar", "baz"}
<-- {"jsonrpc": "2.0", "error": {"code": -32700, "message": "Parse error"}, "id": null}
```

Позив неисправног JSON-RPC-а:

```
--> [1,2,3]
<-- {"jsonrpc": "2.0", "error": {"code": -32600, "message": "Invalid JSON-RPC."}, "id": null}
```

```
--> {"jsonrpc": "2.0", "method": 1, "params": "bar"}
<-- {"jsonrpc": "2.0", "error": {"code": -32600, "message": "Invalid JSON-RPC."}, "id": null}
```

JSON и АЈАХ

Ајах представља могућност веб странице да тражи нове податке након учитавања странице прегледача, најчешће као одговор на акције корисника у оквиру саме странице. JSON се често користи уз Ајах.

Као део Ајах модела, нови подаци се обично припајају у кориснички интерфејс динамично како пристижу са сервера. Пример овога у пракси би био да корисник куца у поље за претрагу, клијентска страна шаље шта је откуцано до тада серверу који одговара са листом могућности које се поклапају са претрагом у бази података. Ово се може представити у падајућем мениу, те корисник може зауставити своје куцање и изабрати нешто од понуђеног. Када се појавио средином 2000-тих, Ајах се најчешће ослањао на XML као формат за размену података, али су многи програмери користили управо JSON за размену Ајах ажурирања између сервера и клијента.

Наредни JavaScript код је пример клијентског коришћења XMLHttpRequest захтева у JSON формату од сервера.

```
var my_JSON_object = {};  
var http_request = new XMLHttpRequest();  
http_request.open("GET", url, true);  
http_request.onreadystatechange = function () {  
    var done = 4, ok = 200;  
    if (http_request.readyState == done && http_request.status ==  
ok) {  
        my_JSON_object = JSON.parse(http_request.responseText);  
    }  
};  
http_request.send(null);
```

ПОРЕЂЕЊА СА ДРУГИМ ФОРМАТИМА

JSON је често представљен као једноставна алтернатива за XML док оба формата поседују широку могућност за стварање, читање и декодирање у реалном окружењу где се често користе. Поред XML-а, примери би могли да укључе рецимо OGD, YAML и CSV. Такође, Google Protocol Buffers би могао да испуни ову улогу, иако није језик за размену података.

YAML

YAML је скоро, али не у потпуности, надскуп JSON-а. На пример, избегавање косе црте (/) коришћењем обрнуте косе црте (\) је валидно у JSON-а, али не и у YAML-а. Ово је чест случај да када се убацује JSON код у HTML за заштиту напада од стране блиских сајтова. Ипак, многи YAML парсери могу у потпуности да парсирају излаз из многих JSON енкодера.

XML

XML је коришћен за опис структурираних података и серијализацију објеката. Различити протоколи који се ослањају на XML постоје ради представљања исте врсте структуре података и за исту сврху. Када се податак кодира у XML-у, резултат је обично дужи од истог кодирања у JSON-у, највећим делом из разлога што XML има и етикете за затварање. Ипак, уколико се подаци компресују коришћењем одређеног алгорита као што је gzip веома су мале разлике јер је компресија добра за чување простора када се патерн понавља.

XML вредности немају специфичну врсту података.

ПРИМЕРИ

JSON ПРИМЕР

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021"
  },
  "phoneNumber": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "fax",
      "number": "646 555-4567"
    }
  ]
}
```

Оба наредна примера обухватају исте информације као и горе наведени, на различите начине.

YAML ПРИМЕР

Иако је горе наведени код у потпуности исправан, YAML нуди алтернативну синтаксу која је разумљивија и прихватљивија за људе, замењујући заграде и наводнике структурираним белинама.

```
--
firstName: John
lastName: Smith
age: 25
address:
  streetAddress: 21 2nd Street
  city: New York
  state: NY
  postalCode: 10021

phoneNumber:
  -
    type: home
    number: 212 555-1234
  -
    type: fax
    number: 646 555-4567
```

XML ПРИМЕРИ

```
<person>
  <firstName>John</firstName>
  <lastName>Smith</lastName>
  <age>25</age>
  <address>
    <streetAddress>21 2nd Street</streetAddress>
    <city>New York</city>
    <state>NY</state>
    <postalCode>10021</postalCode>
  </address>
  <phoneNumbers>
    <phoneNumber type="home">212 555-1234</phoneNumber>
    <phoneNumber type="fax">646 555-4567</phoneNumber>
  </phoneNumbers>
</person>
```

```
<person firstName="John" lastName="Smith" age="25">
  <address streetAddress="21 2nd Street" city="New York" state="NY"
postalCode="10021" />
  <phoneNumbers>
    <phoneNumber type="home" number="212 555-1234"/>
    <phoneNumber type="fax" number="646 555-4567"/>
  </phoneNumbers>
</person>
```

ЛИТЕРАТУРА

1. Александра М. Јанковић, мастер рад, „AJAX технологија“, Математички факултет, Београд, јул 2008.
2. Douglas Crockford , „JSON The x in Ajax“, Yahoo! Inc.
3. Званична интернет презентација <http://www.json.org>
4. Слободна енциклопедија, Wikipedia <http://www.wikipedia.org>
5. JSON Tutorial <http://www.w3schools.com/json/>