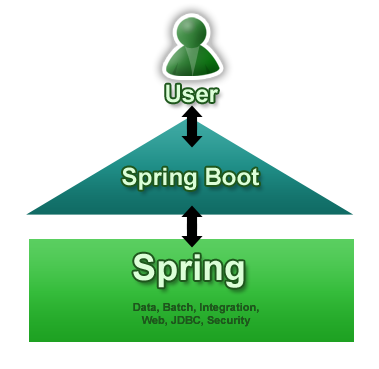
**Spring Boot**

**Spring Boot** je organziran na način da pojednostavni stvaranje, preko platforme **Spring**, aplikacije i servise na stručnoj razini s minimalnom razinom složenosti. Spring Boot također omogućuje da postojeći i novi korisnici vrlo lako koriste segmente Spring-a koji su im potrebni. Može se koristiti u svrhu stvaranja samostalnih (stand-alone) Java aplikacija. SB pruža i 'command line tool' preko kojeg se vrlo lako pokreću Spring skripte.



Spring Boot ne generira nikakav dodatni kod. Nije postoji niti potreba za XML postavkama za rad! Spring skrpite se pišu jezikom **Groovy**, sintaksa vrlo nalik na Javu, ali je pojednostavnjena s obzirom na Javu. Groovy je vrlo intuitivan pri pisanju, stoga je moguće logički deducirati što tuđi kod znači s vrlo malo razmišljanja. Primjer jednostavne skripte:

@Controller

class ThisWillActuallyRun {

@RequestMapping("/")

@ResponseBody

String home() {return "Hello World!"}

}

Preko Spring Boot CLI-a s komandom **‘spring run webapp.groovy’** se pokreće ova jednostavna skripta  
Vise informacija se nalazi na [spring.io](https://spring.io/blog/2013/08/06/spring-boot-simplifying-spring-for-everyone/)

**Spring Tool Suite**

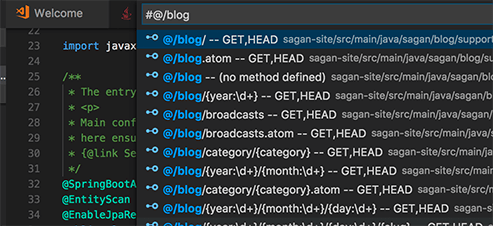
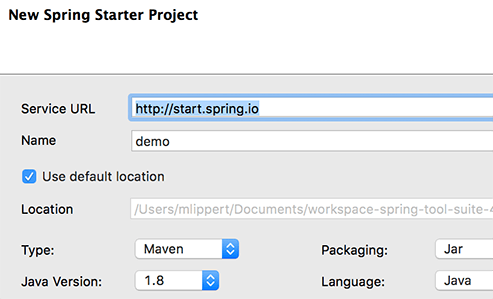
**Spring Tool Suite (STS)** pruža među najboljim okruženjenjima za stvaranje i razvoj aplikacija koje rade preko **Spring platforme**. **STS** pruza mnogo alata za rad s najnovijim izdanjima Jave i Springa te je najčešće na vrhu liste Eclipse-ovih alata. STS podržava razvojni tip **Pivotal tc Server**-a, dogovorena zamjena za Apache Tomcat, vrlo dobro optimiziran za rad u Spring-u.

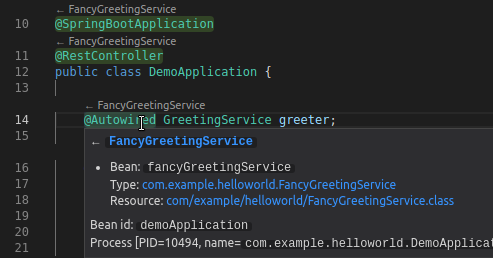
**Pivotal Tc Server** pruža graficki (real-time) prikaz od performansi aplikacija, što omogućuje softver programerima lakše uočavanje problema u aplikaciji te ubrzava cijeli razvojni proces projekta.

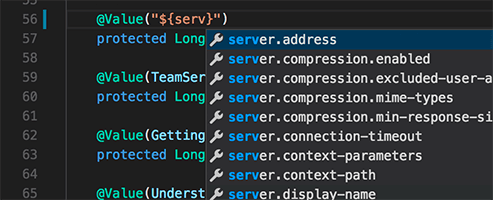
Aktualni STS je **Spring Tool Suite 4**:  
 **-** Najnovija generacija Spring alata za svoje radno okruženje  
 **-** STS4 je većinski napravljen ispočetka  
 **-** Pruža svjetsku razinu podrške za rad nad Spring aplikacijama  
 **-** STS4 se može koristiti u više razvojnih okruženja: **Eclipse, Visual Studio Code, Atom IDE**

**Spring aware**Stvoren za rad sa Spring platformomi Spring Boot-om**,** vrlo lako detektira bilo kakvu Spring aplikaciju i pruža mnoštvo alata i preporuka za pisanje koda.

**IDE diagnostic**Koristi se u svim inačicama STS-a kroz sva razvojna okruženja koja imaju njegovu podršku.  
Služi za dodavanje dodatnih sučelja i uključivanje različitih inačica alata iz STS-a koje bi programeru omogućile lakši razvoj aplikacije.



Intuitivno stvaranje **Spring** podržane aplikacije Olakšane pretrage svih izvornih kodova   
kroz par jednostavnih koraka. za 'importanje' u svoju aplikaciju.



IDE automatski prati što pišete u kodu te odmah  
traži sve moguće nadopune tako da se izbjegnu Uz to pruža dijagnostiku svih djelova vaše  
pogreške na razini tipkanja samog aplikacije čak dok je u pogonu, u svrhu koda. pronalaženja mogućih grešaka

**GitHub**

**GitHub** je distribuirani sustav za upravljenje izvornim kodom nastao 2005. godine iz nezadovoljstva razvijatelja Linux operacijskoga sustava karakteristikama postojećeg sustava za upravljanje izvornim kôdom ("BitKeeper").

**Svojstva GitHuba:**

* **Distribuiranost**, za razliku od ranijih sustava za upravljanje izvornim kôdom (CVS,SVN) GitHub je distribuiran, dakle nakon inicijalnog "kloniranja" spremišta/repozitorija nije nužno za svaki *commit* odnosno inačicu softvera spajati se na središnje spremište, moguće je pohraniti inačicu u lokalno spremište i sinkronizirati spremišta izvornog kôda onda kad je to prikladno
* **Usklađenost s postojećim protokolima**, GitHub podržava HTTP, FTP, rsync, git ili ssh internetske protokole
* **Efikasnost u radu s velikim projektima**, npr. Linux Kernel

Jednostavnost stvaranja repozitorija se jasno vidi na primjeru:

**moj-prvi-projekt** i stvoriti novi repozitorij u njemu:

$ mkdir moj-prvi-projekt // stvara se direktorij na računalu **moj-prvi-projekt**

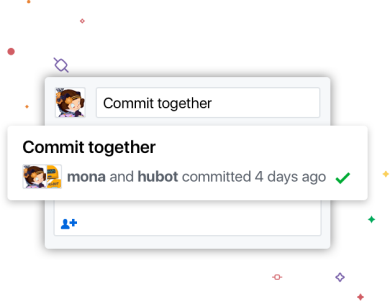
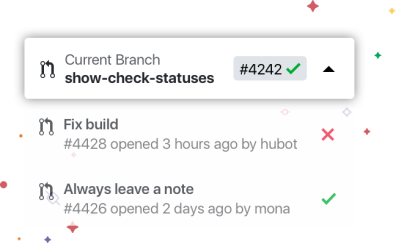
$ cd moj-prvi-projekt // lokaliziramo terminal na novostvoreni direktorij

$ git init // git init stvara repozitorij s imenom izvorne mape

Initialized empty Git repository in /home/user/moj-prvi-projekt/.git/

$

Git naredbe imaju strog tip formatiranja: $ git <naredba> <opcija1> <opcija2> ...  
  
**GitHub Desktop**

Stvoren u svrhu brzog i jednostavnog pristupa  
GitHubu i korištenja većina njegovih funkcija   
bez potrebe korištenja Git naredbi preko CLI-a.

Vrlo lako se vidi tko je kada napravio 'commit'  
na GitHub.   
Također je vrlo fleksibilan za   
preuzimanje podataka iz jedne ili više grana.

Vrlo lako se primjete izmjene nad kodom  
sa obojanim **'diff'**-ovima.  
GitHub desktop kao i GitHub podržava   
razlikovanje vrsta jezika kojim je bilo koji  
**'push'**-an kod napisan.

**Google Cloud Platform**

**Google Cloud Platforma** je pružana od strane Googlea, sadrži skupinu servisa za programiranje na cloudu. Neki od pruženih servisa ubrajaju i podršku za spremanje podataka, analizu podataka, strojno učenje i slično. Za registraciju na GCP je potrebna **kreditna kartica** ili **bankovni račun**.  
  
**Sadržaj GCP**

**Compute**App Engine – Platforma za pokretanje Java, PHP, Python, C#, .NET, Ruby, Node.js i Go aplikacija  
Compute Engine – Infrastruktura za pokretanje MS Windows i Linux virtualnih mašina  
Cloud Functions – Funkicje kao servis za pokretanje 'event-driven' kod baziranog nad Node.js

**Pohrana i Baze Podataka**Cloud Spremište – Objektno spremište za nestrukturirane podatke.  
Cloud SQL – Baza podataka kao servis baziran nad MySQL i PostgreSQL.  
Cloud BigTable – Rukovođen NoSQL servis za baze podataka  
Cloud Spanner – Horizontalno skalabilan, snažno konzistentna, Relacijska Baza Podataka  
Persistent Disk – Block spremište za Compute Engine virtualne mašine

**Networking**VPC – Virtualni privatni cloud za upravljanje nad softver definiranim networkom cloud resursa  
Cloud Load Balancing – Softverski definirani servis za upravljanje stresom nad prometom podataka  
Cloud Armor – pruža sigurnost protiv napada 'hakera' npr. DDos napadi.

**Alati za Upravljanje**Stackdriver – promatranje, zabilježavanje te dijagnostika za aplikacije na GCP i AWS  
Cloud Deployment Manager – Alat za pokretanje GCP resursa koji su definirani na gotovim platformama  
Cloud Console – Web konzola za upravljanje GCP resursima  
Cloud Shell – Preko browsera (Firefox i slični) CLI za pristupanje GCP resursima.  
Cloud Console Mobile App – Android i iOS aplikacija za pristup GCP resursima.

**Identitet i Sigurnost**Cloud Identity – Single sign on (SSO) servis baziran na SAML 2.0 i OpenID.  
Cloud IAM – Identitiy & Access Managment (IAM) servis za definiranje postavki baziranih na   
role-based access control  
Cloud Identity-Aware Proxy – Servis za kontrolni pristup cloud aplikacijama koje rade na GCP bez korištenja VPN-a  
Cloud Data Loss Prevention API – Servis za automatizirano otkrivanje, klasificiranje, i raščlanjenje osjetljivih podataka  
Security Key Enforcement – Two-Step verification servis baziran na sigurnosnom ključu.

**IoT**Cloud IoT Core – Sigurna konekcija nad uređajem i upravljanje servisa za IoT  
Cloud IoT Edge – Dovodi AI (artifitial inteligence) u sloj za 'edge computing'.

**Google App Engine**

**Google App Engine** je web framework platforma za razvoj i pružanje web aplikacija u Google-upravljanim podatkovnim centrima. Aplikacije su prilagođene za rad preko mnoštvo servera.

App engine pruža automatsku skalabilnost podatkovnih resursa za aplikaciju. Što je potražnja za aplikacijom veća, to će se više resursa memorije pridodijeliti aplikaciji kako bi mogla brzo i neometano raditi na svakom klijentu bez kompromisa.

Google App Engine je besplatna platforma do određenih granica korištenja. Cijena za korištenje je diktirana potrebom za više memorije spremišta podataka, brzine konekcije, ili sati neometanog rada aplikacije.

Podržani pogramski jezici uključuju sve standardne razvojne programske jezike današnjeg svijeta. A to su uglavnom :

* Python
* Java
* JRuby
* Go
* PHP
* Node.js
* C#

Google App Engine sadrži mnoštvo Java i Python standarda i frameworkova,što uključuje  
Pyramid, Flask, web2py za Python i za Javine standarde je ključan servlet 2.5 tehnologija.

Preko integrirane baze podataka, Google Cloud Datastore, možda nepoznata mnoštvu programera, njoj se pristupa preko jednostavne Low-level API te ima podršku od JPA i JDO.

Spring platforma također može raditi preko Google App Engine.

**Pouzdanost i Podrška**App engine je dizajniran na način da može podnijeti više podatkovnih čvorova bez padova ili smetnja. Ova otpornost na gubljenje informacija je pokazana statistički da GAE ima 'down time' od 0%!  
Plaćena podrška je omogućena za motrenje aplikacija na dubljoj i sigurnijoj razini nego besplatna varijanta korištenja GAE.  
Besplatna podrška omogućuje sve standarne načine potražnje pomoći, Stack Overflow,GitHub, Server Fault itd.

POPIS TEHNOLOGIJA

* GOOGLE COMPUTE ENGINE

Google compute engine (GCE) je infrastruktura kao usluga (IaaS) komponenta Google cloud platforme koja je građena kao globalna infrastruktura koja pokreće Google search engine, Gmail,Youtube i ostale servise.

GCE omogučava klijentima keiranje i pokretanje vlastitih programa na Google fizičkom hardveru.

GCE nam pruža veći broj skalabilnih virtualnih mašina(VMs) koje služe kao računalni cluster za navedenu svrhu.

GCE-om se može upravljati preko RESTful API-a, CLI-a(Command line interface) ili preko Web konzole.

VMs se mogu pokreniti preko standardne slike (system images) ili prilagođene slike kreirane od strane korisnika.

GCE omogučava administratoru da izabere regiju i zonu gdje će podatkovni resursi biti pohranjeni i korišteni. Do sada na raspolganju stoji United States, Europa i Azija.

GCE također pruža paket alata koji omogučava administratorima izgradnju napredne mreže na regionalnom nivou.

Podržava Linux i windows slike (system images) koje su nužne za pokretanje VMs.

Uporaba GCE virtualnih mašina nije besplatna, već se nakon 10-minutnog rada naplačuje po minuti.

* GOOGLE CLOUD SHELL

Google cloud shell nam pruža Command-line pristup našim cloud resursima iz samog browsera.

Omogučava nam da upravljamo našim projektima i resursima bez instaliranja Google Cloud SDK-a ili drugih alata. Uz pomoć Cloud shell-a, Cloud SDK gcloud command-line alati su nam uvijek dostupni i ažurirani.

Cloud shell pruža command-line pristup instanci virtualne mašine (VM) u terminalskom prozori koji se otvara u web konzoli.

Cloud shell tako pruža sigurnost pristupa projektima i resursima korištenjem autorizacije.

Cloud shell također pruža široki izbor već instaliranih i ažuriranih command-line alata iz basha,sh-a,emacs-a,vim-a, kao i administrativnih alata kao što su mysql client i kubernetes.

Osim command-line alata Cloud shell također pruža široki izbor razvojnih alata kao što su Java,Python,Go,node.js, PHP i uby.

Cloud shell pruža 5GB memorije na persistent disku za pohranu podataka koji je mountan kao $HOME direktorij na Cloud shell instanci.

* WEB SERVICE SECURITY (WSS)

WSS je SOAP ekstenzija koja uvodi sigurnosni sloj kod web usluga.

U 2002, Microsoft, IBM, i Verisign su izdali WS-Security specifikaciju jer SSL/TLS nije bio dovoljan kada SOAP poruka prolazi kroz posredne točke.

U 2003, WS-Security je predložen kao OASIS norma, i u 2004 je izdan kao OASIS norma.

WS-Security specificira SOAP sigurnosne ekstenzije koje omogućavaju tajnost uporabom XML Encryption i integritet podataka uporabom XML Signature.

WS-Security također uključuje profil koji specificira kako umetnuti različite tipove binarnih i XML sigurnosnih tokena u WS-Security zaglavljima za provjeru autentičnosti i autorizacijske namjere:

* Korisničko ime s opcionalnim pregledom lozinke (definira kako korisnik Web usluge može dostaviti korisničko ime kao ovlaštenje za provjeru autentičnosti; uz korisničko ime može biti priložen *hashed* lozinka).
* X.509 certifikat (potpisana struktura podataka dizajnirana za slanje javnog ključa stranki koja prima).
* Kerberos etiketa (token za provjeru autentičnosti i sesiju).
* Security Assertion Markup Language (SAML) sigurnosni tokeni (više detalja o SAML kasnije u ovom dokumentu).
* REL dokument (*rights expression language* (REL) odobreni tokeni umetnuti u WS-Security zaglavlja se rabe za autorizaciju).
* XCBF dokument (definira kako uporabiti XML Common Biometric Format (XCBF) jezik za provjeru autentičnosti s WS-Security specifikacijom).

* XML DIGITAL SIGNATURE

Definira sintaksu i pravila za kreiranje digitalnog potpis XML sadržaja.

XML (Extensible Markup Language) je vrlo jednostavan, fleksibilan tekstualni format razvijen iz SGML (Standard Generalized Markup Language) jezika.

SGML je ISO (International Organization for Standardization) standard za definiranje markup jezika za dokumente.

Digitalni potpis je elektronička zamjena za rukom pisani potpis, a služi istoj funkciji. Uz to pruža i potvrdu autentikacije, integriteta, te povjerljivosti izvora i podataka koji se prenose.

digitalni potpis nije kopija ručno pisanog potpisa.

U tehničkom smislu, digitalni potpis stvara i provjerava posebna aplikacija koja generira kriptografske poruke. Kako bi digitalni potpis funkcionirao, stvaraju se dva različita ključa. Prvi, zvan javni ključ, kreira digitalni potpis transformacijom podataka u nerazumljiv kod. Drugi, zvan privatni ključ, provjerava digitalni potpis te dešifrira transformiranu poruku u izvorni oblik. Ovaj sustav je siguran dok je privatni ključ zadržan tajnim.

XML digitalni potpis je u principu digitalni potpis dizajniran za uporabu u XML transakcijama, a može se koristiti za potpisivanje bilo kojeg tipa podataka.

Dobra strana XML digitalnih potpisa je bolja fleksibilnost od digitalnih potpisa kao što su PGP (Pretty Good Privacy) i CMS12 (Cryptographic Message Syntax), jer ne radi na binarnim podacima nego na XML informacijskom skupu.

XML digitalni potpie koristi koncept kanonizacije što omogućuje potpisivanje samo važnog dijela, te uklanjanje manje važnih podataka poput praznih razmaka i završetka retka.

Nepravilno rukovanje XML digitalnim potpisima može dovesti do pojave sigurnosnih ranjivosti. Jedno od važnih obilježja ovih potpisa je da se potpisani XML elementi, zajedno s potpisom, mogu kopirati iz jednog dokumenta u drugi uz zadržavanje mogućnosti za provjeru potpisa.

* XPATH

Xpath je jezik za specifkaciju upita nad XML podacima (kao sql za upite nad relacijskim bazama).

Služi za lociranje dijela XML dokumenta, pronalaženje elemenata I atributa koji odgovaraju određenom kriteriju I omogučava traženje u bilo kojem smjeru (unaprijed, unatrag, u oba smjera).

Rezultat XPath izraza može bit skup čvorova ili atomarnih vrijednost (sadržaj), odnosno bilo koji slijed u podatkovnom modelu.

Zadnja verzija Xpatha je Xpath 3.1(W3C preporuka 21.03.2017) i u usporedbi s Xpathom 1.0(prvom verzijom, W3C preporuka 16.11.1999) podržava bogatiji skup podatkovnih tipova, mapa i nizova, iskorištava tip informacije kod validacije putem XML Scheme i uvodi podršku za JSON.

**Soap web service – Simple Object Access Protocol**

* Protokol za slanje poruka za izmjenu informacija u implementaciji web servisa i računalnih mreža
* Koristit XML
* Oslanja se na protokole aplikacijskog sloja – HTTP ili SMTP
* Envelope(omotnica), definira strukturu poruke I kako ju koristit
* Skup pravila za izražavanje instanci aplikacijskog sloja
* Konvencija za prikaz procedura poziva I odgovora

Pravila za sintaksu

* A SOAP poruka MORA biti kodirana u XML-u
* A SOAP poruka MORA koristiti SOAP Envelope namespace
* A SOAP poruka MORA koristit SOAP Encoding namespace
* A SOAP poruka NE SMIJE sadržavati DTD reference
* A SOAP poruka NE SMIJE sadržavati XML Processing Instructions

ELEMENTI

Envelope(required) –korijenski element koji enkapsulira cijelu informaciju

Header(Optional) − sadržava određene attribute koji pomažu u obradi poruke

Body(required) − sadržava XML kod koji nas zanima

Fault(optional) – sadrži informacije o pogreškama

<https://www.w3schools.com/xml/xml_soap.asp>

**WSDL**

* WSDL - Web Services Description Language
* WSDL se koristi za opisivanje web servisa
* Koristi XML

|  |  |
| --- | --- |
| Element | Opis |
| <types> | Definira tipove koje koristi web servis |
| <message> | Definira podatkovne elemente za svaku operaciju |
| <portType> | Opisuje operacije koje mogu biti izvršene I poruke koje su uključene |
| <binding> | Definiria protocol I tip podataka za svaki port |

## WSDL Binding to SOAP

 binding element ima 2 atributa – ime I tip.

Ime (bilo koje se može koristiti) definira ime binding-a, I tip atributa koji pokazuje na port za bindanje

soap:binding element ima 2 atributa - style i transport.

Style atribut može biti "rpc" or "document". Transport atribut definira SOAP protocol koji se koristi(HTTP ili SMTP)

operation element definira svaku operaciju koju portType otkriva

Za svaku operaciju odgovarajuća SOAP akcija moa biti definirana. Također, potrebno je navesti ulaz I izlaz koji su kodirani

<https://www.w3schools.com/xml/xml_wsdl.asp>

**XML**

XML - eXtensible Markup Language.

XML se koristi za pohranjivanje I transport podataka

XML je dizajniran da bude čitljiv I čovjeku I stroju

XML ne radi ništa, on je samo informacija zapakirana u tagove

XML I HTML su dizajnirani s različitim ciljevima

* XML je dizajniran da prenosi podatke – s fokusom na ono što podaci jesu
* HTML je dizajniran da pokazuje podatke – s fokusom kako podaci izgledaju
* XML tagovi nisu predefinirani kao u HTML-u

Kada se prikazuju podaci u HTML-u, ne treba uređivati HTML datoteke kad se podaci promijene, s XML-om se podaci mogu spremiti u odvojene XML datoteke

Čitanje I uređivanje XML datoteke se može jednostavno napraviti s JavaScipt-om.

<https://www.w3schools.com/xml/xml_usedfor.asp>

**XSD**

XML Schema language se često zove I XML Schema Definition (XSD)

## XML sheme podržavaju tipove podataka

## XML sheme koriste XML sintaksu

## Ukratko, XML, samo što postoje tipovi podataka i onda definiramo u toj shemi što je koji tip podatka.. To se u zaglavlju određuje

## https://www.w3schools.com/xml/schema\_schema.asp

**Rest web service - REpresentational State Transfer**

REST je web-standard arhitektura I koristi HTTP Protocol. Svaka komoponenta je nekakav resurs I svakom resursu se pristupa preko njegovog sučelja koristeći standardne HTTP metode.

HTTP methods:

* GET − omogućava read-only određenog resursa
* POST – koristi se za kreiranje novog resursa
* DELETE – za brisanje resursa
* PUT – za kreiranje novog resursa ili update postojećeg

REST web servis koristi JSON iako može i XML-om

dok SOAP koristi isklučivo XML.

https://en.wikipedia.org/wiki/Representational\_state\_transfer

**JSON**

JSON: JavaScript Object Notation.

JSON je sintaksa za pohranu I izmjenu podataka.

JSON je tekst, pisan sa JavaScript objektnom notacijom.

JSON is text, and we can convert any JavaScript object into JSON, and send JSON to the server.

Budući je JSON text, možemo kovertati bilo koji JavaScript object u JSON I poslati ga na server.

JavaScript ima ugrađene funkcije za pretvorbu stringa iz JSON-a u JS objekt

JSON.parse()

JSON vrijednosti moraju biti neki od ovih tipova

* a string
* a number
* an object (JSON object)
* an array
* a boolean
* null

## JSON koristi JavaScript sintaksu

I JSON I XML se mogu koristit za primanje podataka sa web server.

**JSON i XML sličnosti**

* oboje su “self describing" (čitljivi ljudima)
* imaju istu hijerarhiju
* oboje mogu biti parsirani I korišteni od više programskih jezika
* oboje mogu biti dohvaćeni sa XMLHttpRequest

**JSON i XML razlike**

* JSON ne koristi završni tag
* JSON je kraći
* JSON je brži za čitanje I pisanje
* JSON can use arrays

Najveća razlika:

 XML mora biti parsiran sa XML parserom. JSON može biti parsran sa standardnom JS funkcijom

**Zašto je JSON bolji**

XML je teže parsirati nego JSON.  
JSON je parsiran u JavaScript objekt.

Using XML

* dohvati XML document
* koristi XML DOM za proći kroz dokument
* izvuci vrijednosti I spremi u varijable

Using JSON

* dohvari JSON string
* JSON.Parse se koristi za izvlačenj podataka  
  https://www.w3schools.com/js/js\_json\_intro.asp