**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: **J**ava**S**cript **O**bject **N**otation.

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