

RASSUS odgovori na pitanja iz prezentacije (3. predavanje)

B25

22. studenoga 2022.

1. Što je SOAP i čemu služi?

SOAP (*Simple Object Access Protocol*) omogućuje komunikaciju s web-uslugom. Postoje dva načina rada:

- **poziv udaljenih procedura (RPC)**. Služi za prijenos serijaliziranih parametara i rezultata. Posljedice:
 - dobro definirana sučelja i tipovi podataka
 - prilagodni kod može biti generiran automatski
- **razmjene dokumenata/poruka**. Koriste se XML dokumenti u razmjeni poruka.

2. Što je WSDL i čemu služi?

WSDL (*Web Services Description Language*) je jezik za opis web-usluga koji koristi XML. Također se koristi za lociranje web-usluga u kombinaciji s UDDI-om.

3. Kakva je struktura WSDL-a i čemu služe pojedini elementi?

WSDL se sastoji od **apstraktnog** i **konkretnog** opisa.

Apstraktni opis - platformski i jezično neovisni.

Konkretni opis - specifičan za implementacijske usluge.

ELEMENTI:

- **definitions** - osnovni element WSDL dokumenta
- **types** - opis podataka pomoću XML Scheme
- **message**
 - opis jednosmjernе poruke
 - definira ime poruke
 - svaka poruka se sastoji od dijelova (*part*)
 - svaki dio se referencira na vrste podataka iz dijela *types*
- **portType**
 - definira operacije
 - svaka operacija se sastoji od poruka
 - poruke mogu biti **parametri** (ulazne poruke) ili **rezultati** (podaci koji se vraćaju)

- **binding**
 - definira kako će se poruke iz informacije prenositi
 - definira koje će se protokole koristiti (HTTP, SOAP...)
 - stil (*style*) definira vrstu čitave poruke
 - * **rpc** - zahtjev ima omotač u kojem piše naziv funkcije
 - * **document** - zahtjev i odgovori imaju obične XML dokumente
 - vrste pakiranja poruka
 - * **literal** - onako kako definira schema
 - * **encoded** - kodirano prema pravilima u SOAP-u
- **services** - definira lokaciju usluge
- **import** - uključivanje drugih WSDL ili XML dokumenata

4. Što je UDDI i čemu služi?

UDDI (*Universal Description, Discovery and Integration*) je protokol koji definira na koji način se mogu objaviti i otkriti web-usluge. Registar je poslužitelj čija je svrha čuvanje podataka i meta-podataka o web-uslugama.

Sastoji se od 3 dijela:

- **imenik** - sprema adrese, kontakte i identifikatore
- **poslovni imenik** - ima kategorizaciju područja, usluga i proizvoda te lokacije
- **tehničke informacije** - sadrži tehničke informacije o uslugama (dokumenti u WSDL-u)

5. Koja je razlika između web-usluga RPC i usluga temeljenih na dokumentima?

- **RPC** se temelji na pozivu udaljene procedure koju čini operacija deklarirana i opisana WSDL-om.
- **Na dokumentima** - osnovna jedinica komunikacije je poruka, a ne operacija.

6. Kakve su web-usluge temeljene na prijenosu prikaza stanja resursa i što ih karakterizira?

To su usluge koje koriste HTTP i slične protokole. Ovdje je fokus na interakciji s resursima koji imaju stanje, a ne na porukama ili procedurama.

Bez stanja su i imaju priručni spremnik.

Resursi su predstavjeni URL-ovima.

7. Koje su razlike između JSON-a i XML-a?

- **JSON**
 - ZA
 - * format
 - * ugrađen u preglednike
 - PROTIV
 - * nema ugrađene poveznice
 - * nema načina definiranja scheme
 - * ograničen skup podataka
 - * problem binarnih podataka

- **XML**
 - ZA
 - * mogu se pisati složene strukture
 - * ima standardne poveznice
 - * dobar skup alata za obradu
 - PROTIV
 - * brzina procesiranja
 - * problem binarnih podataka
 - * ograničenja DTD-a
 - * puno standarda koji ga kompliciraju

8. Koja su svojstva metoda HTTP-a i objasnite ih?

- **Svojstva**
 - Sigurna - bez posljedica za podatke
 - Idempotentna - izvršavanje više puta
 - Može se prijevremeno pripremiti odgovor
- **Metode**
 - GET - sigurna, idempotentna, *cacheable*
 - PUT - idempotentna
 - DELETE - idempotentna
 - HEAD - sigurna, idempotentna
 - POST - ništa
 - PATCH - ništa, ali se može napraviti idempotentna

9. Objasnite model zrelosti web-usluga?

Ima 4 razine:

- **0. razina**
 - pozivanje usluga je većinom pozivanje udaljenih procedura
 - jedan URI jedna HTTP metoda
- **1. razina**
 - koriste više resursa, ali su nazivi metoda enkodirani u URL
 - više URI-ja, jedna HTTP metoda (GET)
- **2. razina**
 - koriste više resursa i HTTP metoda te statusne kodove
 - ne koriste svoju shemu (vrste podataka koji se koriste)
 - npr. Amazon S3
- **3. razina**
 - isto kao 2., ali koristi hipermedijske vrste
 - resurs opisuje svoje mogućnosti i veze

10. Koje su dvije vrste web-aplikacija i koje su razlike između njih?

- koje izgledaju kao normalne web-stranice (npr portali)

- koje izgledaju kao "normalne", ali imaju bogato korisničko sučelje

11. Kakva je arhitektura web-aplikacije i svrha svakog sloja?

- **Sloj prezentacije**
 - prikaz informacija, GUI
 - obrada HTTP zahtjeva
- **Poslovni sloj**
 - obrada podataka
 - glavni dio sustava koji radi ono za što je sustav namijenjen
- **Sloj pristupa podacima**
 - komunicira s BP
 - brine se o transakcijama
 - brine se o pohrani podataka

12. Koja je razlika između nove i stare arhitekture web-aplikacija?

U novoj arhitekturi se **sloj prezentacije** nalazi na klijentskoj strani. Inicijalno se dohvati web-stranica, a kasnije korisnik interakcijom pokreće dijelove programa u JS i mijenja korisničko sučelje.

13. Čemu služe skripte na klijentu?

Služe za dinamičko stvaranje sadržaja, interakciju s korisnikom, provjeru obrazaca, komunikaciju s poslužiteljem.

14. Što je i kako radi AJAX?

AJAX (*Asynchronous JavaScript and XML*) je tehnika koja se koristi za dinamičko dohvaćanje podataka s poslužitelja. On omogućuje da se stranice mogu ažurirati asinkorono razmjenjujući podatke.

Ako su potrebni podaci s poslužitelja koriste se objekti za AJAX koji šalju HTTP zahtjev na poslužitelj i kada dođe odgovor obavijesti se funkcija zadužena za taj zahtjev

15. Usporedite dva načina slanja događaja s poslužitelja klijentu?

- **Prozivanje poslužitelja** - klijent periodički šalje zahtjeve na poslužitelj i prima odgovore. U slučaju da nema nikakvog događaja, prima prazan odgovor. Kada na poslužitelj stigne poruka od drugog klijenta onda prilikom sljedećeg upita rvi klijent u odgovoru prima i poruku. Opterećivanje mreže i servera nepotrebnim zahtjevima.
- **Dugo prozivanje poslužitelja** - klijent šalje zahtjev, ali server čeka dok se ne pojavi događaj i tek onda šalje odgovor. Nakon toga klijent primi odgovor i odmah šalje novi zahtjev i opet se čeka. Problem kod ovog pristupa je taj što i klijent i poslužitelj moraju stalno imati otvorenu vezu. Pogotovo poslužitelj koji mora imati puno otvorenih veza.

16. Što su i čemu služe web-usluge?

Web-usluga je aplikacija identificirana URI-jem, čija se sučelja i veze mogu definirati, opisati i otkriti XML-om i koja podržava direktnu interakciju s drugim aplikacijama putem internetskih protokola koristeći poruke temeljene na XML-u.

Program koji:

- je identificiran URI-jem

- komunicira klijentskim programima putem weba
- ima API opisan standardima web-usluga
- ima omogućeno korištenje neovisno o platformama i program. jez.

17. Objasnite dva načina kako web-aplikacija koristi web usluge.

- **Dio web-aplikacije na poslužitelju** koristi web-usluge - kada klijent pošalje zahtjev na web-poslužitelj onda server u obradi tog zahtjeva može pozvati drugu web-uslugu koja može biti na istom ili nekom drugom web-poslužitelju. Uglavnom u poslovnim okruženjima gdje se želi zabraniti korištenje usluga izvan poslovne mreže.
- **Dio web-aplikacija na klijentu** koristi web-uslugu - preglednik kroz pozive iz JS-a koristi web-uslugu na poslužitelju. Omogućuje se trećim entitetima da koriste usluge i integriraju ih u svoje web-aplikacije. Često su temeljene na resursima.

18. Koja je razlika između Weba 1.0 i Weba 2.0?

WEB 1.0

- čitanje informacija
- klijent-poslužitelj
- HTML
- portali
- parsiranje informacija
- posjedovanje
- direktorij
- modemska veza i HW
- npr. Netscape

WEB 2.0

- pisanje informacija
- P2P
- XML, JSON, HTML5
- usluge
- REST API sučelja
- dijeljenje
- oznake
- širokopojasna veza i SW
- npr. Google, Facebook,....

19. Koje su 3 vrste web-usluga i razlike između njih?

- **Usluge temeljene na pozivu udaljene procedure** - poziv udaljene procedure (RPC) koju čini operacija deklarirana i opisana WSDL-om.
- **Usluge temeljene na porukama/dokumentima** - osnovna jedinica komunikacije je poruka. To je temelj uslužno-orijentirane arhitekture (SOA). Fokus na "ugovorima" koji su propisani WSDL-om pa ne ovise o implementacijskom jeziku.
- **Usluge temeljene na prijenosu prikaza stanja resursa** - HTTP i slični protokoli. Fokus na resursima koji imaju stanjem, a ne na procedurama i porukama.

20. Na što treba paziti kod dizajniranja REST usluge?

Ne znam...

21. Objasniti postupak dizajniranja REST usluge.

Ne znam...