

XML usmeni - vodič za preživljavanje

1, 7, 24, 27 i 28. prezentacija ne trebaju

(1. je uvod, 7. je na engleskom, 27. i 28. je profa rekao da ne trebaju, a 24. sam čuo od nekoga)

(mada nikoga nije pitao ni SOAP - ispravite ako jeste)

Redni broj je broj prezentacije u kojoj se nalazi odgovor :)

1. Šta je dokument

Osnovni nosilac značenja, osnovna jedinica razmjene informacija, ima 3 dimenzije (karakteristike): struktura, sadržaj, prezentacija

Dokument je fiksna i strukturirana količina informacija kojom se upravlja kao jedinicom i koja se razmenjuje kao jedinica između korisnika i sistema.

Dokumenti predstavljaju jedinične nosioce informacija između elemenata informacionog sistema, imaju životni ciklus, mogu imati strukturu

Mogu biti transakcioni (jako strukturirani) i narativni (bez strukture)

Sadržaj vs reprezentacija dokumenta (prikazan HTML vs HTML kod)

1. Šta je resurs

Resurs može biti bilo šta što ima identitet:

- elektronski dokument

- slika

- servis

- kolekcija resursa

Informacioni resursi su resursi čije se bitne karakteristike mogu prenijeti u poruci

Resurs može biti dostupan putem Interneta (npr. fajl), ali i ne mora (npr. autor fajla)

1. Razlika između URL i URN

URI - Uniform Resource Identifier - niz znakova za identifikovanje resursa. Može biti formiran kao URL ili URN

URL - Uniform Resource Locator - protokol:putanja-do-resursa - podskup URI-ja koji identifikuje resurse preko reprezentacije njihovog primarnog mehanizma pristupa (npr. njihove lokacije na mreži)

URN - Uniform Resource Name - urn:nmspid:nss - podskup URI-ja koji mora ostati globalno jedinstven i perzistentan čak i ako resurs prestane da postoji ili postane nedostupan.

- + za URL - omogućava pristup resursu

S + za URN - omogućava identifikaciju resursa koji čak više i ne postoje ili nisu dostupni

1.6. Koja je razlika između validnog i dobro formiranog dokumenta

Dobro formiran dokument:

XML dokument mora biti dobro formiran da bi mogao da se mašinski obradi

- specijalni znaci mogu da se nau u dokumentu samo u svojoj ulozi
- moraju se poštovati pravila imenovanja elemenata i atributa
- dokument mora imati jedan korenski element
- elementi se mogu ugnježdavati ali ne i preklapati
- vrednost atributa mora biti unutar navodnika (jednostrukih ili dvostrukih)
- element ne može imati dva atributa sa istim imenom
- komentari i procesne instrukcije ne mogu se nalaziti unutar taga

Validan dokument - validan u odnosu na DTD ili Schemu - poštuje sva pravila (strukturu) propisana DTDom

5. XML namespaces

Svako može da definiše sopstevnu XML gramatiku, pa tako može doći do preklapanja imena elemenata. Da bismo učinili nazive elemenata jedinstvenim, dodaćemo im prefiks.

Prostor imena je skup elemenata sa istim prefiksom.

Identifikacija prostora imena vrši se nazivom prostora imena koji je u URI formatu.

Atributi generalno ne nasleđuju namespace od elementa, ali mogu, ako aplikacija tako interpretira.

5. Šta je defaultni namespace

Podrazumijevani namespace je namespace čiji se elementi unutar dokumenta (i pod-dokumenata) navode bez prefiksa.

6. Nasledjivanje prostih i slozenih tipova u semi

C Prosti tipovi: ne mogu sadržati elemente i attribute, mogu nastati restrikcijom ugrađenih tipova

Složeni tipovi: mogu imati podelemente i attribute

Svaka definicija novog tipa se oslanja na neki od postojećih (prethodno definisanih) tipova.

Mehanizmi za kreiranje prostog tipa: (nasleđivanje prostog tipa)

- Restrikcija: navođenje ograničenja na vrijednost novog tipa (skup vrijednosti novog tipa je podskup skupa vrijednosti osnovnog tipa)
- Lista: tip koji definišemo predstavlja listu vrijednosti osnovnog (polaznog) tipa. elementi liste su razdvojene razmacima
- Unija: kombinovanje vrijednosti više različitih osnovnih tipova

Definisanje složenih tipova:

- Element sa prostim sadržajem i atributima - element -> complexType -> simpleContent -> extension -> attribute
- Element sa podelementima - koristi jedan od modela sadržaja: sekvenca podelemenata, izbor između podelemenata, neuređen skup podelemenata
- Element sa miješanim sadržajem - može imati i prost sadržaj i elemente.

Novi složeni tip na osnovu postojećeg (nasleđivanje složenog tipa)

- proširivanje: complexType -> complexContent -> extension -> sequence/choice..

- restrikcija: complexType -> complexContent -> restriction -> ponovo se navode naslijeđeni podelementi sa željenim restrikcijama

6. Jedinstvenost i ključevi u XML Schema

(Isto kao kod relacionih baza podataka)

U DTD-u je bilo moguće osigurati da će vrijednost atributa biti jedinstvena upotrebom tipa ID, dok XML Schema ima veće mogućnosti:

- sadržaj elementa može biti jedinstven (unique)
- vrijednosti atributa koji nisu tipa ID mogu biti jedinstvene
- kombinacija sadržaja elementa i vrijednosti atributa može biti jedinstvena
- može se definisati opseg na kojem se provjerava jedinstvenost
- pravi se razlika između jedinstvenosti (unique) i ključa (key)

Ključ je element, atribut ili kombinacija elemenata i atributa koja:

- a. uvijek postoji (minOccurs > 0)
- b. ne može biti nill (nillable="False")
- c. jeste jedinstvena po vrijednosti (unique="True")

6. Razlika između lokalnih i globalnih atributa u xml šemi

Definicija lokalnih atributa je ugnježdjena u definiciju nekog elementa ili anonimnog tipa i ne može biti korišćena na drugom mjestu u šemi, dok je globalni definisan na nivou cijele šeme i može biti referenciran bilo gdje unutar šeme ili u drugim šemama koje uključuju datu šemu.

Globalni elementi su u namespace-u, lokalni elementi i atributi su povezani sa globalnim elementima (ili lokalnim elementima koji su povezani sa globalnim elementima)

6. XML prostori simbola

Imena komponenti koje se nalaze u šemi (elementi, atributi, tipovi) su smješteni u prostor simbola (symbol space). Ovo (dodatno) rješava problem konflikta imena (pored namespace-ova).

Poseban symbol space za: definicije tipova, deklaracije elemenata, deklaracije atributa

Poseban symbol space za: svaki definisani tip, globalne komponente šeme

Ovo nam dozvoljava korišćenje istih imena ne samo u različitim namespace-ovima nego i različitim symbol space-ovima unutar istog namespace-a. Npr možemo imati isto ime za tip, element i atribut.

8.9. Razlika između stax i sax parsera

Stream parseri: StAX (pull), SAX (push)

Push parseri implementiraju programski model u kome XML parser šalje (gura) podatke programu koji ga koristi nailazeći na elemente XML informacionog skupa (elemente, attribute, tekst, itd.)

Pull parseri implementiraju programski model u kome programi koji ih koriste pozivaju metode XML parsera (vuku podatke) kada im treba element XML informacionog skupa (element, atribut, tekst, itd.)

	StAX	SAX
tip	pull	push
lakoća upotrebe	lako	srednje
XPath	ne	ne
čita XML	da	da
piše XML	da	ne
CPU/RAM	niski zahtjevi	

StAX implementira PULL koncept, program traži podatke kad su mu potrebni. Parser čita XML dokument od početka do kraja i prepoznaje elemente XML informacionog skupa. Može da parsira XML dokumente korišćenjem iteratora (iterira kroz listu događaja da bi dobio podatke - XMLStreamReader/XMLStreamWriter) ili kursora (podatke dobija preko pokazivača na XML čvorove - XMLStreamReader/XMLStreamWriter).

SAX parser - event driven parsiranje. Parser tokom parsiranja generiše događaje (počeo dokument, počeo element, završio se element). Programski kod definiše event handlera koji obrađuju događaje, a pozivaju se iz parsera kao callback.

10. Kakav je to DOM parser

DOM parser podržava slučajan pristup XML čvorovima, jednostavan je za korišćenje, omogućava CRUD operacije, ali zahtijeva veće procesorske i memorijske resurse.

Rezultat parsiranja je stablo objekata (in memory tree) koje je stalno prisutno u memoriji za cijeli dokument. Parser je dvosmjernan jer stablo objekata možemo serijalizovati nazad u XML.

Kretanje kroz DOM stablo -DOM stablo je predstavljeno interfejsima:

Node - apstraktni čvor stabla koga nasleđuju: Document, Element, Attr, Text, Comment, CDATASection, Entity... Node omogućava:

- pristup karaktistikama čvora (get node name, type, value, attributes)
- pristup povezanim čvorovima (get child nodes, first child, last child, parent, next sibling)
- ažuriranje (append child, insert before, replace child, remove child)

11. Šta je JAXB

JAXB (Java Architecture for XML Binding) je framework za generisanje Java klasa na osnovu DTD ili XML Schema šema (i obrnuto) i iz transformaciju XML dokumenata u graf Java objekata (i obrnuto). Ne zahtijeva poznavanje XML-a za obradu XML dokumenata jer sve mapira na Java objekte (POJO, Bean). JAXB je API.

10,11. Koja je razlika između DOM i JAXB

JAXB radi na višem nivou apstrakcije od DOM parsera. U DOM parseru elementi XML informacionog skupa su predstavljeni ugrađenim interfejsima (generičkim čvorovima - Node, Document, Element, Attribute), dok JAXB enkapsulira rad sa XMLom i kao rezultat daje reprezentujuće Java klase iz određenog domena.

Zbog ovih činjenica, upotreba ovih parsera se razlikuje. JAXB parseri se koriste za deserijalizaciju transakcionih dokumenata (dokumenata sa jakom strukturom) koji se lako mapiraju na stablo objekata, dok se parseri nižeg nivoa (DOM, StAX, SAX) koriste za parsiranje narativnih dokumenata (slabije strukture).

12. XPath

XPath je jezik za označavanje dijelova XML dokumenta koji se zasniva na konceptu navigacije kroz stablo dokumenta.

XML dokument se, sa stanovišta XPath-a, posmatra kao stablo čiji su čvorovi tipa: element, atribut, tekst, namespace, komentar, dokument (korijen)...

Koncept tekućeg čvora ("gdje se trenutno nalazim"), ekvivalent "working directory-ju" u terminalu/cmd-u.

Korak putanje se sastoji iz ose kretanja(skup čvorova u odnosu na tekući čvor), testa čvora, nula ili više predikata(koriste se za pronalaženje čvora koji zadovoljava dati uslov).

`osa::test[predikat]`

12. Apsolutna vs relativna putanja u XPath-u

Apsolutna putanja počinje sa / i kreće od čvora Dokument, a relativne putanje ne počinju sa / i kreću od čvora u kome se trenutno nalazimo, tj tekućeg čvora.

13. xquery

Standardni upitni jezik za XML, ekvivalent SQL-u za XML dokumente.

Rezultat XQuery upita može biti atomička vrijednost ili XML sadržaj.

svaka XQuery vrednost je sekvenca koja sadrži 0 ili više elemenata(singleton).

svaki singleton ima svoj tip izveden iz item().

Svaki XQuery izraz ima statički tip(compile-time) dinamički tip-tip dobijenog rezultata, vrednost rezultata je instanca tog tipa(run-time)

XQuery prolog- nije obavezan, definiše kontekst za upit (compile-time)-namespace, funkcije, importovani schema tipovi, importovani tipovi, promenljive.

Promenljive - sa znakom \$ ispred imena(ne može se menjati vrednost)

- FLWOR izrazi (for, let, where, order by, return)
- FLOWR join (dekartov proizvod, inner join-one to one, inner join-many to many, outer join -left outer join,self join
- Poređenje: egzistencijalno poređenje. poređenje član-po-član, univerzalno poređenje - uslov je zadovoljen za svaki element

15. Kako uključiti css fajl u xml i šta još treba u css fajlu dodati

CSS je tehnologija za vizuelizaciju HTML dokumenta, adaptirana i za XML. Služi kako bi se razdvojila logička struktura dokumenta od njegove prezentacije.

CSS se dodaje u XML dokument korišćenjem procesne instrukcije:

```
<?xml-stylesheet type="text/css" href="fajl.css"?>
```

U css fajlu se ne dodaje ništa ----> document {display:block;} u css

16. xslt transformacije

XSLT je jezik za opis transformacija XML dokumenta u druge XML dokumente koji za navigaciju koristi XPath izraze. Između ostalog, koristi se i za vizuelizaciju XML dokumenta (ako je rezultat transformacije XHTML).

XML dokument se transformiše u drugi transformacijom opisanom XSLT dokumentom. Transformaciju vrši XSLT procesor.

XSLT sadrži definicije šablona koji opisuju transformacije dijelova polaznog dokumenta (identifikovanih XPath izrazom).

16. Koji je osnovni element u transformacijama

valjda: <xsl:template match="xpath"> ... </xsl:template>

(<xsl:stylesheet version="1.0"> ... </xsl:stylesheet> (korenski element))

20. Šta je RDF

RDF (Resource Description Framework) je graf bazirani model za opisivanje stvari (resursa, objekata) i odnosa među njima.

To je standardni model za razmenu podataka na web-u.

Proširuje strukturu web-a tako što koristi URI za imenovanje stvari i relacija između stvari.

Ova struktura formira usmereni označeni graf (čvorovi su stvari, a grane relacije između stvari)

Graf se može predstaviti kao skup iskaza oblika: <subjekat> <predikat> <objekat>

20. Linked Data principi

Linked data je skup principa za objavljivanje, pronalaženje i pregledanje podataka u RDF formatu

Linked Data predstavlja način objavljivanja podataka na (semantičkom) webu koji: ohrabruje ponovno korišćenje podataka

smanjuje redundantnost

maksimizira (pravu i potencijalnu) povezanost između podataka

omogućava stvaranje "mrežnog efekta" i dodaje vrednost podacima

Principi:

- +
 - Koristiti URI za imena stvari
 - Koristiti HTTP URI da bi stvari mogle da se pronađu
 - Kada neko traži resurs, ponuditi odgovor u RDF formatu
 - U odgovor uključiti RDF iskaze koji povezuju traženi resurs sa drugim resursima da bi mogle da se pronađu srodne stvari

20. RDF formati (turtle, rdf/xml, RDFa)

Turtle - kompaktna i lako čitljiva tekstualna serijalizacija RDF grafa.

```
<subject> <predicate> <object>;  
    <predicate> <object>.
```

RDF/XML - tekstualna XML serijalizacija RDF grafa koja obezbjeđuje maksimalnu interoperabilnost

```
<rdf:RDF>  
    <rdf:description about="subject">  
        <pref:predicate1>object1</pref:predicate1>  
        <pref:predicate2>object2</pref:predicate2>  
    </rdf:description>  
</rdf:RDF>
```

RDFa (RDF in attributes) - omogućava ugrađivanje RDF iskaza u XML dokumente pomoću standardizovanih atributa.

about - subjekat, typeof - tip subjekta

rel, property - predikat

href, src, resource - objekat, content - objekat (literal)

```
<link rel="..." href="...">  
<span about="urn:ISBN:0091808189" typeof="bibo:Book"  
property="dc:title">Canteen Cuisine</span>
```

GRDDL je specifikacija koja olakšava ekstrakciju RDF iskaza iz XML dokumenta. Najčešće se RDFa iskazi transformišu u RDF/XML upotrebom XSLT transformacija.

22. sparql

Upitni jezik za postavljanje upita nad podacima u RDF formatu, a ujedno i protokol za postavljanje upita nad udaljenim RDF skladištima preko HTTP protokola.

SPARQL omogućava

- ekstrakciju podataka iz strukturiranih i polustrukturiranih izvora
- transformaciju podataka u RDF formatu iz jednog rječnika u drugi
- Izvršavanje složenih join operacija u jednom upitu.

SELECT ?title

WHERE {

<urn:isbn:9780743477123> dc:title ?title

}

Vrste SPARQL upita:

SELECT - vraća promjenljive u vidu tabele

ASK - vraća logičku vrijednost koja označava da li bi ekvivalenta SELECT upit vratio rezultat

CONSTRUCT - vraća RDF graf konstruisan na osnovu promjenljivih

DESCRIBE - vraća RDF graf koji opisuje resurs

Struktura RDF upita:

deklaracija prefiksa

PREFIX ...

definicija izvora podataka (rdf grafova nad kojima će se vršiti upit)

FROM ...

klauzula rezultata - specifikacija podataka koje će upit vratiti

SELECT ...

obrazac grafa - specifikacija restrikcija nad podacima (where ...)

WHERE ...

modifikatori upita - sortiranje, grupisanje, isijecanje

GROUP BY, SORT, LIMIT, OFFSET

22. Šta je sparql endpoint

SPARQL endpoint je nešto kao API, interfejs, za izvršavanje sparql upita. Omogućavaju pristup podacima u RDF formatu. Prima zahtjeve i vraća odgovore preko HTTP protokola (SPARQL protokol ili RESTful).

Podaci nad kojima se vrše upiti mogu biti skladišteni triplestore-u ili u drugim vrstama skladišta sa kojim se komunicira preko middleware-a.

22. Razlika između triple store-a i sparql endpoint-a

RDF skladišta (triplestores) skladište podatke u RDF obliku kao skup trojki ili RDF graf. Omogućavaju izvršavanje upita nad podacima koje skladište, a indeksiranjem su optimizovani za vršenje upita nad grafovima.

24. REST principi

REpresentational State Transfer - definiše principe softverske arhitekture za web

Principi:

- RESURS - svaki entitet na webu je resurs (html strana, web servis, xml dokument)
- ADRESA resura - svaki resurs je identifikovan svojim URI-jem nad resursima se obavljaju jednostavne operacije (CRUD) (HTTP metode: GET, POST, PUT, DELETE)
- REST nije standard nego stimuliše upotrebu standarda:
 - HTTP (protokol za komunikaciju)
 - URL (standard za lokaciju i identifikaciju resursa)
 - HTML, XML, JSON, JPEG (standardi za reprezentaciju resursa)

Klijent se obraća resursu putem URI-ja, dobija reprezentaciju resursa, ta reprezentacija pomera klijenta u novo stanje, klijent se zatim obraća drugom resursu, itd.

Seoba klijenta iz stanja u stanje = transfer