

XML 2010

Erik Tjong Kim Sang
erikt(at)xs4all.nl
18 oktober 2010

XML 2010

College 7

Planning

Inmiddels werken jullie aan de eindopdracht voor XML

De tweede quiz staat inmiddels op Nestor. De derde quiz is op dinsdag 19 oktober beschikbaar

We moeten nu deadlines vaststellen voor het afronden van de eindopdracht en de quizen

Opmerkingen:

- de tentamens beginnen op 25 oktober
- de tweede helft van het eerste semester begint op 8 november

1

XML 2010

College 7

Over opdracht 5.10

5.10 Hoeveel zinnen in jouw bestand bevatten twee keer of vaker het woord de (in kleine letters)?

De beoogde XPath-oplossing was:

```
count(//node-000[@word='de']/following::node-000[@word='de'])
```

Deze oplossing werkt niet voor bestanden waar meerdere zinnen instaan. De tweede `node-000[@word='de']` zoekt dan ook in de andere zinnen

Op de website staat een werkende oplossing met XSLT en XPath

2

XML 2010

College 7

Vorige week

Opslaan van XML-data in databases:

- opslag van XML-data in tekstvelden
- omzetten van XML-data naar tabellen (XML-shredding)
- zoeken in databases met XPath en SQL

3

XML 2010

College 7

Deze week

Gebruik van XML voor web services: RSS, SOAP en Ajax

4

XML 2010

College 7

Wat is RSS?

RSS (*Really Simple Syndication*) is een standaard voor het samenvatten van de inhoud van websites

Via RSS kan een website op een gestandaardiseerde manier worden gepresenteerd voor verwerking door software

Een RSS-bestand dat een website beschrijft heet een *feed*

RSS-bestanden zijn XML-bestanden; zij worden gelezen met feedreaders

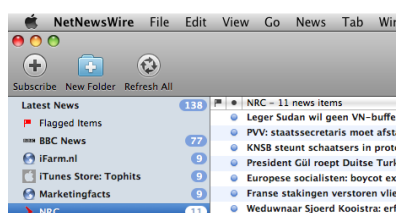
Organisaties die RSS-feeds van verschillende websites verzamelen en aanbieden heten aggregators

5

XML 2010

College 7

Voorbeeld van een feedreader



6

XML 2010

College 7

Geschiedenis van RSS

1999 RSS versie 0.9 van Netscape (R.V. Guha)

2000 RSS versie 1.0 van RSS-DEV (Guha)

2000 RSS versie 0.92 van Userland Software (Dave Winer)

2002 RSS versie 2.0 van Userland Software (Winer)

2005 Atom versie 1.0 van de Internet Engineering Task Force

Alle grote nieuwsorganisaties gebruiken anno 2010 RSS 2.0

7

RSS 1.0 en RDF

RSS 1.0 maakt gebruik van RDF: *Resource Description Framework*

RDF is een methode om concepten te beschrijven

Het gebruikt daarbij drie velden:

- de naam van het concept, bijvoorbeeld boek1234
- de eigenschap van het concept, bijvoorbeeld titel
- de waarde van de eigenschap, bijvoorbeeld Beginning XML

De drie velden samen vormen RDF-triplets

8

Een voorbeeld van RDF in RSS 1.0

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <channel rdf:about="http://meerkat.oreillynet.com/?_fl=rss1.0">
    <item rdf:about="http://c.moreover.com/click/here.pl?r123">
      <title>XML: A Disruptive Technology</title>
      <link>Xhttp://c.moreover.com/click/here.pl?r123</link>
      <dc:publisher>XThe O'Reilly Network</dc:publisher>
      <dc:description>
        XXML is placing increasingly heavy loads on the...
```

RDF-triplets in kleur: naam, eigenschap en waarde

9

Dublin Core

De in het RSS1.0-voorbeeld gebruikte namespace dc: staat voor Dublin Core

Dit is een verzameling van 15 metadata-elementen die digitale en fysieke data beschrijven in RDF-bestanden

De 15 elementen gebruikt in Dublin Core zijn:

title, creator, subject, description, publisher, contributor, date, type, format, identifier, source, language, relation, coverage en rights

10

Standaardelementen in RSS 2.0

- channel: beschrijving van de feed
- item: artikel in de feed
- title: titel van artikel of feed
- link: adres van artikel of feed
- description: beschrijving van artikel of feed
- pubDate: publicatiedatum van artikel (optioneel)
- guid: identificator van artikel (optioneel)

Een RSS2.0-feed kan nog diverse andere optionele elementen bevatten

11

Voorbeeld van een RSS2.0-feed

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="/d/styles/css2full.xml"?>
<?xml-stylesheet type="text/css" href="/d/styles/itemcontent.css"?>
<rss version="2.0">
  <channel>
    <title>NRC Handelsblad | Voorpagina</title>
    <description>Nieuws en achtergronden van NRC Handelsblad</description>
    <link>http://www.nrc.nl</link>
    <item>
      <title>Leger Sudan wil geen VN-bufferzone tussen noord en zuid</title>
      <link>http://feedproxy.google.com/~r/NrcHandelsbladVoorpagin...</link>
      <description>Khartoem, 16 okt. De Verenigde Naties ku...</description>
      <guid>http://www.nrc.nl/buitenland/article2632552.ece/Leger_...</guid>
      <pubDate>Sat, 16 Oct 2010 16:30:58 +0200</pubDate>
    </item>
  </channel>
</rss>
```

12

Zelf een feedreader maken

Als je kunt programmeren in Perl of in Java dan kan je zelf een feedreader maken: schrijf dan een programma dat:

1. feedbestanden ophaalt
2. met XML-parsing de gegevens uit de bestanden haalt
3. eventueel gegevens selecteert en andere verwijdert
4. de gegevens laat zien op het scherm (of opslaat)

Voor de XML-parsing is voor beide programmeertalen de module SAX (Simple API for XML) beschikbaar

13

Wat zijn webservices?

Webservices zijn taken die door servers worden uitgevoerd op verzoek van andere computers (de clients)

Het leveren van een webpagina wordt meestal niet als webservice gezien

Een meer algemene naam voor zo'n service is *remote procedure call* (RPC)

Voor de communicatie bij webservices wordt vaak XML gebruikt

14

HTTP

HTTP (Hypertext Transfer Protocol) is het voor webservices gebruikt communicatieprotocol tussen computers:

1. de client opent een verbinding met de server
2. de client stuurt een verzoek naar de server
3. de server voert de taak van de client uit
4. de server stuurt een antwoord naar de client
5. de verbinding wordt verbroken

Het protocol kan worden getest met het Linuxcommando `telnet`

15

Webservices via GET

De belangrijkste taken die HTTP kan uitvoeren zijn GET en POST

GET wordt gebruikt voor het ophalen van webpagina's maar het kan ook gebruikt worden voor webservices

De parameters van de taak moeten dan in het adres worden verwerkt:

```
$ telnet urd.let.rug.nl 80
GET /erikt/bin/qademo.cgi?TEXT=wie+is+madonna
```

De vraag aan het systeem wordt in een variabele TEXT meegegeven in de url na een vraagteken (plussen staan voor spaties)

16

Webservices via POST

Bij gebruik van het commando POST wordt de taak voor de webserver vermeld in de body van het bericht van de client:

```
POST /bin/getDatabaseData
Content-Type: text/xml
Content-Length: 68

<dbcommand>
  SELECT * FROM nodes WHERE lemma="fiets";
</dbcommand>
```

Het verzoek van de webserver is nu gecodeerd met XML

17

Wat is SOAP?

SOAP (ooit *Simple Object Access Protocol*) is een standaardformaat voor berichten voor webservices

Het definieert een algemene manier om meta-informatie over verzuurde berichten te coderen, bijvoorbeeld authenticatie en foutafhandeling

SOAP-berichten zijn gecodeerd in XML

De laatste versie, 1.2 (2007), is een W3C-recommendation

18

Voorbeeld van een SOAP-bericht

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap...">
  <soap:Header>...</soap:Header>
  <soap:Body>
    <soap:Fault>
      <soap:Code><soap:Value>Sender</soap:Value></soap:Code>
      <soap:Reason><soap:Text>Fout!</soap:Text></soap:Reason>
    </soap:Fault>
    <myData><dbcommand>SELECT...</dbcommand></myData>
  </soap:Body>
</soap:Envelope>
```

De elementen Envelope, Header en Body zijn verplicht
Het element Fault specificeert de foutafhandeling

19

Elementen in SOAP-berichten

- Envelope: root-element (verplicht)
- Body: bevat webservicebericht (verplicht)
- Header: bevat meta-informatie over bericht (verplicht)
- Fault: bevat verwachte reacties op fouten

Voorbeelden van mogelijke foutcodes:

Sender: fout in formaat SOAP-bericht
Receiver: SOAP-bericht kon niet worden verwerkt
MustUnderstand: verplicht headerveld kon niet worden verwerkt

20

Het definiëren van webservices

De structuur van een webservice kan, net als de structuur van XML-bestanden, formeel worden gedefinieerd

Hiervoor wordt de taal WSDL gebruikt: *Web Service Description Language*

WSDL bevat onderdelen voor het definiëren van berichtenstructuur (element types), datatypes (message), relaties tussen invoer en uitvoer (portType en binding) en adres (service)

21

Voorbeeld van een WSDL-bestand

```
<definitions name="ABC" ...namespaces...>
  <types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
      ...schema-definitie van webserviceberichtstructuur...
    </xsd:schema>
  </types>
  <message>...typedefinitie van objecten in berichten</message>
  <portType>
    <operation><input message="type"/><output.../></operation>
  </portType>
  <binding>...formaat berichten en reacties (soap)...</binding>
  <service>...adres van webservice...</service>
</definitions>
```

22

AJAX

AJAX (Asynchronous JavaScript and XML) is een verzameling van technieken om asynchroon data op te halen van websites

De term *asynchroon* slaat op het feit dat de data-uitwisseling niet synchroon hoeft te gaan met het ophalen van een webpagina

Met AJAX kan een gedeelte van een webpagina worden verversd zonder dat de hele pagina opnieuw van de webserver hoeft te worden gehaald

23

Gebruik van AJAX

Een bekend toepassing van AJAX is *autocompletion*

Terwijl een gebruiker een zoekopdracht intikt, worden de ingetikte letters al naar de server gestuurd die mogelijke aanvullingen terugstuurt

Voorbeelden van websites die deze techniek gebruiken zijn
<http://www.google.com/webhp> en <http://www.youtube.com>

Een andere toepassing van AJAX is interactieve manipulatie van kaarten, zie bijvoorbeeld <http://maps.google.com/>

24

AJAX gebruikt XML

AJAX is vooral een programmeertechniek; een uitgebreide behandeling past beter in een college over webprogrammeren (onderwerpen: Javascript en de module XMLHttpRequest)

Voor ons is belangrijk dat AJAX voor de berichten tussen client en server vaak XML gebruikt, zie bijvoorbeeld:

[http://google.com/complete/search?output=toolbar&q="](http://google.com/complete/search?output=toolbar&q=)

Het gebruik van XML is nog niet universeel, zie:

<http://en.wikipedia.org/w/api.php?action=opensearch&search=test>

25

Achtergrondliteratuur

David Hunter, Jeff Rafter, Joe Fawcett and Eric van der Vlist, *Beginning XML (Programmer to Programmer)*. John Wiley & Sons, 2004 (chapters 13–16: RSS, Web Services, SOAP and Ajax)

W3schools, *Ajax Tutorial*, <http://www.w3schools.com/ajax>

W3schools, *RSS Tutorial*, <http://www.w3schools.com/rss>

W3schools, *SOAP Tutorial*, <http://www.w3schools.com/soap>

W3schools, *Web Services Tutorial*, <http://www.w3schools.com/webservices>

26

Slotwoord

We hebben in zeven weken diverse onderwerpen bekeken die iets met XML te maken hadden: DTDs, CSS, XSLT, XPath, XML-databases en webservices

Met de meeste onderwerpen hebben we ook praktische oefeningen gemaakt

Een mogelijk vervolg op dit college is Semantic Web Technology (LIX002M10)

XML-technologie is nog steeds in ontwikkeling; ook de komende jaren zijn komen er ongetwijfeld nieuwe technieken ter beschikking

27

SUCCES MET DE VERDERE STUDIE!