StAX

XML i web servisi

Stevan Gostojić

Fakultet tehničkih nauka, Novi Sad

5. april 2016.

Agenda

1. XML Parsers
2. StAX

XML Parsers

- Moguće je napraviti XML parser koristeći tehnike sa kojima ste se upoznali na programskim prevodiocima (za dobro formirane ili validne XML dokumente)
- U većini slučajeva je bolje koristiti postojeće XML parsere
  - manje posla
  - manja verovatnoća greške

XML Parsers

Postoji nekoliko programskih modela za parsiranje XML dokumenata

- streaming parseri
  - pull (StAX parseri)
  - push (SAX parseri)
- DOM parseri
- XSL-T parseri

Streaming vs. DOM parseri

	streaming	DOM
+	koristi malo resursa (procesor, memorija)	slučajan pristup, lakši za korišćenje
-	sekvencijalni pristup, teži za korišćenje	koristi puno resursa (procesor, memorija)

Table 1: Streaming vs. DOM parseri

Push i pull parseri

- Push parseri implementiraju programski model u kome XML parser šalje (gura) podatke programu koji ga koristi nailazeći na elemente XML informacionog skupa (elemente, atribute, 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 vs. SAX vs. DOM vs. XSLT parseri

	StAX	SAX	DOM	TrAX
Tip	pull, streaming	push, streaming	in memory tree	XSLT rules
lakoća upotrebe	lako	srednje	lako	srednje
XPath	ne	ne	da	da
CPU/RAM	niski zahtevi	niski zahtevi	visoki za-htevi	visoki za-htevi
jednosmerni	da	da	ne	ne
čita XML	da	da	da	da
piše XML	da	ne	da	da
CRUD	ne	ne	da	ne

Table 2: StAX vs. SAX vs. DOM vs. XSLT parseri

Agenda

1 XML Parsers

2 StAX

StAX API

- StAX API implementira pull koncept što znači da program traži podatke od StAX parsra onda kada su mu potrebni
- Parser čita XML dokument od početka do kraju i prepoznaje elemente XML informacionog skupa (tokene koji čine dobro formiran XML dokument)
- StAX API može da se koristi i za čitanje i za pisanje XML dokumenata (za razliku od SAX API-a)
- XML dokumente može da parsira korišćenjem iteratora (iterira kroz listu događaja da bi dobio podatke) ili kursora (podatke dobija preko pokazivača na XML čvorove)

Cursor API vs. Iterator API

- Cursor API koristi kursor koji se kreće od početka do kraja XML dokumenta i pokazuje na elemente XML informacionog skupa
- Iterator API predstavlja XML dokument kao niz diskretnih događaja (koji odgovaraju XML informacionom skupu) koji se mogu obraditi

StAX API

Klasa/Interfejs	Opis
XMLEventReader	obezbeđuje iterator nad događajima koji se koriste za parsiranje XML dokumenta
XMLEventWriter	sadrži metode za pravljenje događaja
XMLStreamReader	obezbeđuje kursor koji se koristi za parsiranje XML dokumenta
XMLStreamWriter	sadrži metode za pravljenje čvorova

Table 3: Bitni elementi StAX API-a

XMLEventReader

Klasa	Opis
XMLEvent nextEvent()	vraća sledeći događaj
public boolean hasNext()	proverava da li postoji još događaja

Table 4: Metode XMLEventReader klase

XMLEvent

Klasa	Opis
StartDocument	sadrži xml version, encoding i standalone svojstva
StartElement	sadrži lokalno ime, deklaracije prostora imena i atribute
EndElement	kraj elementa
Characters	sadrži tekstualni sadrža (uključujući i CDATA, ignorable whitespace)

Table 5: Naslednice XMLEvent klase

XMLEventReader

```
1  XMLEventReader eventReader = factory.createXMLEventReader(reader);
2
3  while(eventReader.hasNext()) {
4    XMLEvent event = eventReader.nextEvent();
5
6    switch(event.getEventType()) {
7    case XMLStreamConstants.START_ELEMENT:
8      StartElement startElement = event.asStartElement();
9      // ...
10     break;
11   case XMLStreamConstants.CHARACTERS:
12     Characters characters = event.asCharacters();
13     // ...
14     break;
15   case XMLStreamConstants.END_ELEMENT:
16     EndElement endElement = event.asEndElement();
17     // ...
18     break;
19   default:
20   }
21 }
```

XMLEventWriter

Klasa	Opis
add(Event event)	dodaje događaj koji predstavlja element

Table 6: Metode XMLEventWriter klase

XMLEventWriter

```
1  XMLEventWriter eventWriter = factory.createXMLEventWriter(writer);
2
3  XMLEvent event = eventFactory.createStartDocument();
4  eventWriter.add(event);
5
6  event = eventFactory.createStartElement("prefix", "namespaceUri", "localName");
7  eventWriter.add(event);
8
9  event = eventFactory.createNamespace("prefix", "namespaceUri");
10 eventWriter.add(event);
11
12 event = eventFactory.createAttribute("attributeName", "attributeValue");
13 eventWriter.add(event);
14
15 event = eventFactory.createCharacters("text");
16 eventWriter.add(event);
17
18 event = eventFactory.createEndElement("prefix", "namespaceUri", "localName");
19 eventWriter.add(event);
20
21 eventWriter.flush();
22 eventWriter.close();
```

XMLStreamReader

Metoda	Opis
int next()	vraća sledeći čvor
boolean hasNext()	proverava da li postoji još čvorova
String getText()	vraća tekstualni sadržaj elementa
String getLocalName()	vraća ime elementa
...	..

Table 7: Metode XMLStreamReader klase

XMLStreamReader

```
1  XMLStreamReader streamReader = factory.createXMLStreamReader(reader);
2
3  while(streamReader.hasNext()) {
4    streamReader.next();
5
6    switch(streamReader.getEventType()) {
7    case XMLStreamConstants.START_ELEMENT:
8      // ...
9      break;
10   case XMLStreamConstants.CHARACTERS:
11     // ...
12     break;
13   case XMLStreamConstants.END_ELEMENT:
14     // ...
15     break;
16   default:
17   }
18 }
```

XMLStreamWriter

Metoda	Opis
writeStartElement(String localName)	dodaje početni tag
writeEndElement(String localName)	dodaje krajnji tag
writeAttribute(String localName, String value)	dodaje atribut
...	...

Table 8: Metode XMLStreamWriter klase

XMLStreamWriter

```
1  XMLStreamWriter streamWriter = factory.createXMLStreamWriter(writer);
2
3  streamWriter.writeStartDocument();
4  streamWriter.writeStartElement("localName");
5  streamWriter.writeStartElement("localName");
6  streamWriter.writeAttribute("attributeName", "attributeValue");
7  streamWriter.writeCharacters("text");
8  streamWriter.writeEndElement();
9  streamWriter.writeEndElement();
10 streamWriter.writeEndDocument();
11
12 writer.flush();
13 writer.close();
```

Zaključak

- Upoznali smo se sa tipovima XML parsera
- Naučili smo kako se pravi XML parser koji koristi StAX API (kursor ili iterator)

Reference

- StAX API, https://goo.gl/wfoT0J

Hvala na pažnji!
Pitanja?