

Document Object Model - DOM

Document Object Model

- W3C standard
- standard za objektno-orijentisanu reprezentaciju dokumenata sa hijerarhijskom strukturom - stablo
- varijante za različite jezike
 - Java
 - C++
 - JavaScript
 - ...
- varijante za različite dokumente
 - XML
 - HTML

Document Object Model

- koncept rada sa DOM parserom razlikuje se od SAX-a
- rezultat parsiranja je stablo objekata
 - stalno prisutno u memoriji, za ceo dokument
 - redosled obrade elemenata ne mora se poklapati sa redosledom u dokumentu
 - možemo serijalizovati stablo nazad u XML
 - problem sa zauzećem memorije
 - za velike dokumente
 - za serverske aplikacije (puno istovremeno procesiranih dokumenata)

DOM + Java

- rad sa XML dokumentima preko objektno reprezentacije stabla dokumenta
- od Java 1.4
- paketi
 - javax.xml.*
 - org.w3c.dom.*
 - org.xml.sax.*

...DOM parseri se po pravilu oslanjaju na SAX parsere

Parsiranje dokumenta

- fabrika parsera
`DocumentBuilderFactory factory =
 DocumentBuilderFactory.newInstance();`
- napravimo parser
`DocumentBuilder builder = factory.newDocumentBuilder();`
- parsiramo dokument
`Document doc = builder.parse(xml);`

→primer1

DocumentBuilderFactory

- kreira parsere
- moguće je podešavati osobina generisanih parsera
 - setValidating()
 - setNamespaceAware()
 - setIgnoringElementContentWhitespace()
 - setIgnoringComments()
 - setCoalescing() - da li se CDATA čvorovi pretvaraju u text čvorove i spajaju sa okolnim text čvorovima
 - setAttribute() - postavljanje dodatnih atributa (npr. validacija pomoću šeme)

DocumentBuilder

- parsira postojeći XML dokument u DOM stablo
 - `parse(File f)`
 - `parse(InputSource is)`
- kreira novo DOM stablo
 - `newDocument()`
- obrada grešaka
 - `setErrorHandler()` - ako se ne postavi, koristi se default
 - obrada grešaka istovetna kao i kod SAX parsera

Kretanje kroz DOM stablo

- DOM stablo reprezentuju interfejsi
 - Node - apstraktni čvor stabla, nasleđuju ga
 - Document - čitav XML dokument
 - Element
 - Attr
 - Text - tekstualni sadržaj
 - Comment
 - CDATASection
 - Entity
 - Notation
 - ProcessingInstruction

Kretanje kroz DOM stablo

- Node interfejs
 - pristup karakteristikama samog čvora
 - nodeName
 - nodeType
 - nodeValue
 - attributes

Kretanje kroz DOM stablo

- Node interfejs
 - pristup povezanim čvorovima
 - getChildNodes
 - getFirstChild
 - getLastChild
 - parentNode
 - getNextSibling

Kretanje kroz DOM stablo

- Node interfejs
 - metode za ažuriranje
 - appendChild
 - insertBefore
 - replaceChild
 - setNodeValue
 - removeChild

Kretanje kroz DOM stablo

- Document interfejs
 - getElementElement - pristup korenskom elementu dokumenta
 - getElementById - pristup elementu sa datim ID-em
 - getElementsByTagName - lista elemenata sa datim nazivom
 - sve metode vraćaju samo Element čvorove!
-
- primer2 (konstruiše JTree od DOM-a)
 - primer3 (filtrira prikazane čvorove)
 - primer4 (prikazuje i sadržaj elemenata)

Kretanje kroz DOM stablo

- tipovi čvorova

Interface	nodeName	nodeValue	attributes
Attr	name of attribute	value of attribute	null
CDATASection	"#cdata-section"	content of the CDATA Section	null
Comment	"#comment"	content of the comment	null
Document	"#document"	null	null
DocumentFragment	"#document-fragment"	null	null
DocumentType	document type name	null	null
Element	tag name	null	NamedNodeMap
Entity	entity name	null	null
EntityReference	name of entity referenced	null	null
Notation	notation name	null	null
ProcessingInstruction	target	entire content excluding the target	null
Text	"#text"	content of the text node	null

Kreiranje DOM stabla

- kreiranje praznog dokumenta

```
DocumentBuilderFactory factory =  
    DocumentBuilderFactory.newInstance();
```

```
DocumentBuilder builder = factory.newDocumentBuilder();
```

```
Document doc = builder.newDocument();
```

Kreiranje DOM stabla

- primer - programski kreiramo stablo koje odgovara dokumentu

```
<team name="Sacramento">  
  <player num="21">  
    Vlade Divac  
  </player>  
</team>
```

Kreiranje DOM stabla

- kreiramo root element

```
// kreiramo root element
```

```
Element root = (Element)doc.createElement("team");
```

```
// kreiramo cvor za atribut i dodajemo ga root-u
```

```
root.setAttributeNode(doc.createAttribute("name"));
```

```
// postavljamo vrednost atributa
```

```
root.setAttribute("name", "Sacramento");
```

```
// jedino dete Document čvora je root
```

```
doc.appendChild(root);
```


Kreiranje DOM stabla

- dodavanje elementa

```
Element player = (Element)doc.createElement("player");  
player.setAttributeNode(doc.createAttribute("num"));  
player.setAttribute("num", "21");  
root.appendChild(player);
```

Kreiranje DOM stabla

- dodavanje tekstualnog čvora

```
Text name = (Text)doc.createTextNode("Vlade Divac");  
player.appendChild(name);
```

→primer5 (programski se kreira dokument)
→primer6 (dokument se i normalizuje)

Validacija pomoću šeme

- slično kao i za SAX

```
DocumentBuilderFactory factory =  
    DocumentBuilderFactory.newInstance();  
  
factory.setNamespaceAware(true);  
factory.setValidating(true);  
  
factory.setAttribute(  
    "http://java.sun.com/xml/jaxp/properties/schemaLanguage",  
    "http://www.w3.org/2001/XMLSchema");
```

Povezivanje dokumenta sa šemom

- na dva načina
 - šema deklaracija u dokumentu
 - programsko povezivanje

```
factory.setAttribute(  
    "http://java.sun.com/xml/jaxp/properties/schemaSource",  
    new File("mySchema.xsd"));
```

Povezivanje dokumenta sa šemom

- dokument sa više namespace-ova

```
factory.setAttribute(  
    "http://java.sun.com/xml/jaxp/properties/schemaSource",  
    new String[] {  
        "nikon.xsd",  
        "pentax.xsd",  
        "olympus.xsd"});
```