

JAVA 2



XML

Teme

- XML
- DTD
- XSD
- JAXP
- XML parsers
- JAXB

XML

- ***Well-formed***

- XML elementi moraju biti pravilno zatvoreni
- XML oznake razlikuju mala i velika slova
- XML oznake moraju biti pravilno ugniježdene
- XML dokument mora imati korijenski element
- sve vrijednosti atributa moraju biti unutar navodnika
- prazna su mjesta sačuvana unutar navodnika
- znakovi `<`, `>`, `&`, `'`, `"` su specijalni pa *lome* dokument
 - zamjena sa referencama entiteta: `<`, `>`, `&`, `'`, `"`;
 - *wrappanje* u `<![CDATA[< > " ' &]]>` - ne parsiraju se jer su *character data*

- ***Valid – DTD, XSD***

DTD

- **Document Type Definition**
- *internal* (unutar xml dokumenta) i *external* (poseban dokument)
- privatni (*SYSTEM*) i javni (*PUBLIC*)
- opisuje XML strukturu i služi za njegovu validaciju
 - forsira redoslijed pojavljivanja elemenata
 - opcionalni i obavezni elementi
 - opcionalni i obavezni atributi sa *default* vrijednostima
- elementi, atributi i referencijalni entiteti
- brojni nedostaci – ne omogućuju strogu tipiziranost, ograničavanja vrijednosti...

DTD

Type	Description
CDATA	The value is character data
(<i>en1</i> <i>en2</i> ..)	The value must be one from an enumerated list
ID	The value is a unique id
IDREF	The value is the id of another element
IDREFS	The value is a list of other ids
NMTOKEN	The value is a valid XML name
NMTOKENS	The value is a list of valid XML names
ENTITY	The value is an entity
ENTITIES	The value is a list of entities
NOTATION	The value is a name of a notation
xml:	The value is a predefined xml value

The **attribute-value** can be one of the following:

Value	Explanation
<i>value</i>	The default value of the attribute
#REQUIRED	The attribute is required
#IMPLIED	The attribute is optional
#FIXED <i>value</i>	The attribute value is fixed

XSD

- **XML Schema Definition**
- napisan u XML
- podrška za imenske prostore (*namespaces*) – izbjegavanje imenskih konflikata
- superioran DTD, puno veća kontrola
 - stroga tipiziranost, ugrađeni osnovni tipovi
 - nasljednost (definiranje vlastitih kompleksnih i jednostavnih tipova)
 - ograničavanje vrijednosti
 - može olabaviti pravila redoslijeda pojavljivanja elemenata
 - . . .

XML Parsers

SAX - Simple **API** for **XML**

- *push model* - parsira liniju po liniju i obaviještava pretplatnike o događajima (otvaranje, zatvaranje tag-a, čitanje teksta...)

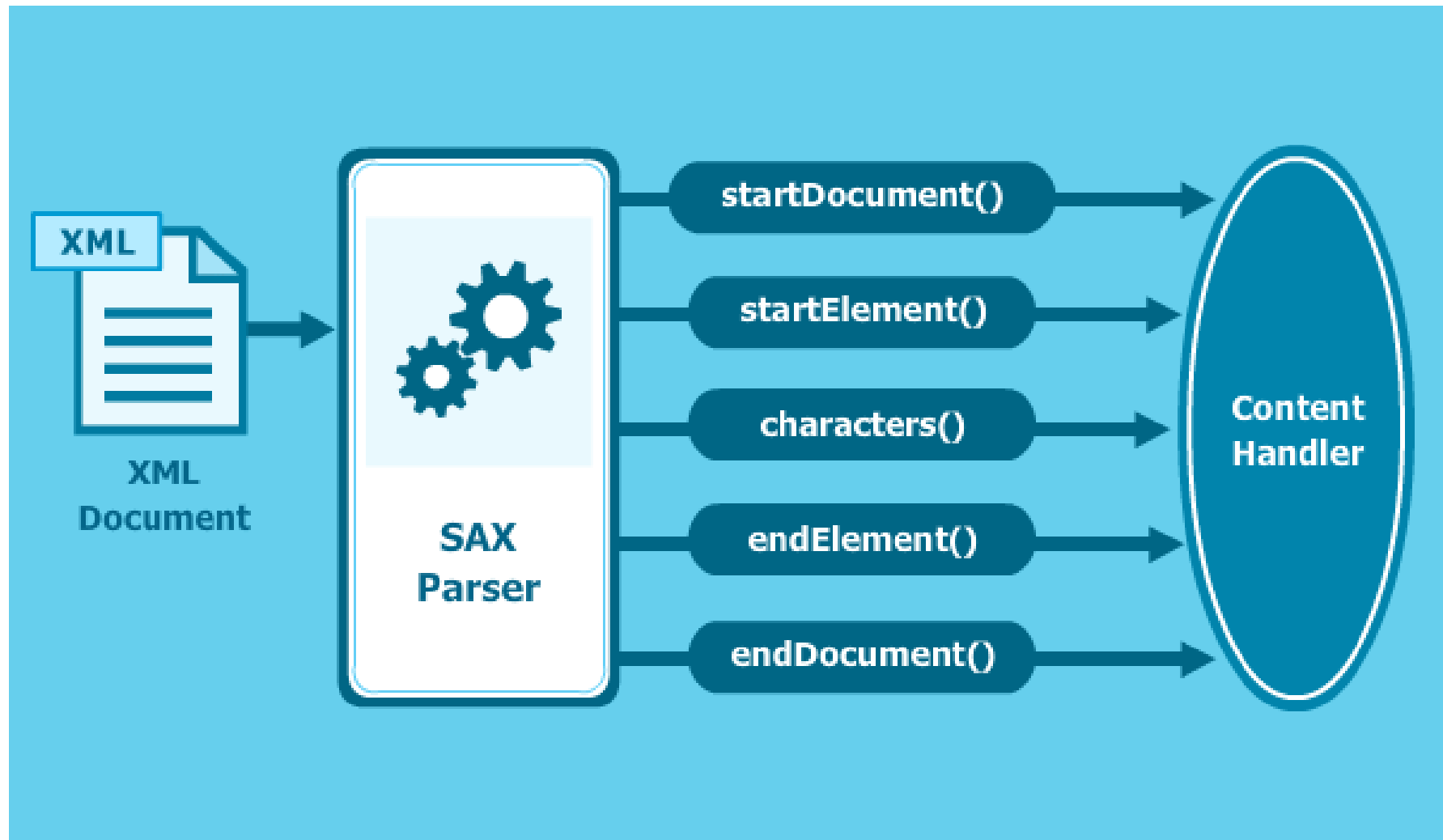
StAX - Streaming **API** for **XML**

- *pull model* – klijent sam *povlači* događaje

DOM - Document **Object Model**

- učitava cijeli xml u memoriju i omogućuje prolazak po čvorovima

Sax



Izvor:<https://www.edureka.co/blog/parsing-xml-file-using-sax-parser/>

Stax



Izvor: <https://www.developerfusion.com/article/84523/stax-the-odds-with-woodstox/>

Dom

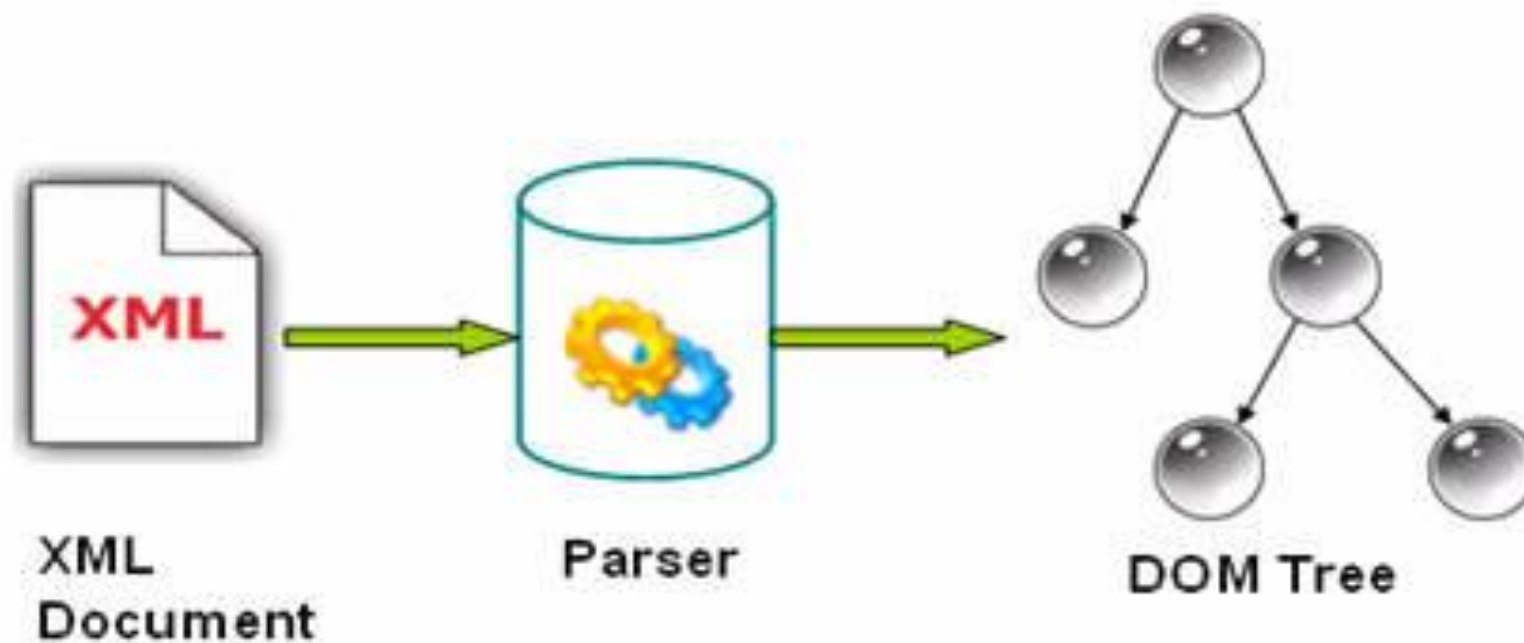


Fig a. Document Object Model

Izvor: <https://www.topjavatutorial.com/java/java-programs/parsing-xml-dom-java/>

JAXB

- **J**ava **A**rchitecture for **X**ML **B**inding
- *bind-a*, povezuje Java kod i xml tagove
- anotacije – mapiranje klasa sa njihovim xml reprezentacijama
- lagana prilagodba – imena, atributi, liste
- *marshall* – omogućuje nizanje klasa u xml
- *unmarshall* – obrnut postupak za kreiranje klasa iz xml
- odradili smo na Java 1 kolegiju

Demo

- Project



Izvor:<http://www.jnhsolutions.com/contact-us/request-a-demo/>

Hvala na pažnji!

