

24

XML

1. Дадре-структурнизиране XML

1.1 Основни концепции

- XML таг - маркер (таг) с име на елемент, отделен с `< >`; `<name></name>` отварящ замварящ
- XML елемент - означава се с отварящ и замварящ таг и съдържанието между тях
- XML таг - състои се от таговете на документа
- XML документ - съдържа текст под формата на 1 или повече елементи
- XML атрибут - звънка им/стойност, асоцииран с елемент, добавя се в `<name value="5"></name>` отварящ замварящ таг, стойността се заправя с `" / "`
- Инструкции - `<? ... ?>` предоставят инфо, необходима на бъдещи приложения
- XML декларация - `<?xml version="1.0" encoding="UTF-16" standalone="yes"?>` указва версията на XML спецификацията на док., единична ли и да ли е асоциирана с бъдещо DTD, в началото на документа

1.2 XML йерархия

- Представяне - иерархично
деревидно - връзките са XML елементи
- всеки док. има 1 root елемент, който съдържа всички останали
- Структура на XML док.:
 - Граф - декларация на док, стилове и тип на док;
 - Екземпляр (Instance) - йерархията от елементи
 - Допълнения - коментари, CDATA секции и инструкции за обработка

1.3 Синтаксични правила

- всички отварящи таг трябва да имат съвпадащи замварящи (или да е самозат.)
- маркети не трябва да се заимстват, ил. трябва да подхожда библиотека
- XML root док. трябва да има точно 1 root ил.
- имената на ил. трябва да спазват XML конвенциите
 - да започват с лявна/лявка буква или '-' и да не почват с XML, xml, Xm1
- XML е case-sensitive
- XML запазва празните пространства в PCDATA (Parsed Character Data)
- Всеки атрибут трябва да има стойност, гори и да е правилен
- не е разрешено пълнотрида на западните XML символи
 - те се заместват с предварително деф. entities или с поставят в `<![[CDATA[...]]>` секция

- 1.4. Търсопранства от имена (namespaces)
- абстрактни номенклатури за група от имена
 - чрез уникатността на XML елементите поддържан от URI
 - за привисците означаващи пространство от имена, се използват URIs
 - деф. се с атрибути xmlns и namespace за елементите в програмата, в която са дефинирани: <pers:person xmlns:pers="http://..."/>
 - един док. може да има мн-во пространства от имена
 - използват се за разрешаване на конфликти

2. XML валидация чрез Document Type Definition (DTD)

2.1. Част на валидиранието

- всички док. се очаква да съдържа определен език / речник и са необходими методи за определяне на нивото на съответствие
- XML док. е валиден, ако съдържанието му отговаря на деф. на посочените елементи, атрибути и др. членки
- DTD док. очаква правилата за структуриране на XML док., изразяващи то ет., възможни типове атрибути и съдържиме им по подразбиране

2.2. DTD структура

- DTD е съхранявана в:
 - вече видима форма
 - в XML док. фалата
 - и във файла
- Състои се от:
 - Document Type definition (DOCTYPE) - информация за парсера, че е док. и съдържимо (DTD), задължително е в началото на док.
 - Текст - съдържима декларации на ед., атрибути, entities и номенклатури;
 - DTD декларацията се завършва с ">"

Пример:

```
<!DOCTYPE nome [
  <!ELEMENT nome (first, last)>
  <!ELEMENT first (#PCDATA)>
    —————— last ——————
]>
```

2.3 Синтаксис

- Document Type - <!DOCTYPE ... >
- Character data - <!CDATA[...]>
- Entities - <!ENTITIES ... >
- Notation - <!NOTATION ... >
- Element - <!Element {name} {EMPTY|ANY|type|children, ...} >
- Attributes - <!ATTLIST element-name attribute-name type value>

- conditional section

- <! [INCLUDE] >, <! [IGNORE] >

3. XML валидация чрез XML Schema

3.1. Спецификация

- XML док., който описва структурата на друг XML док.

- наследник на DTD, но предоставя по-страница спецификация

3.2. Типове данни

- Възможни

- простиви - string, boolean, float, ID...

- язоводни - language, IDREF, long, short...

- Генерирани от компютър

- чрез дефиниране от съществуващи типове - чрез определение на

- чрез промяна на множество: фрази

- derivedBy="extension разширява с атрибути

- derivedBy="restriction - ограничава генерираният тип

- <Simple Type> декларации

3.3. Фрази

Определят аспект на проектирането от стойността на даден тип

Две бури:

указани

- Уравнителни - equal, order, bounds, cardinality, numeric

- Ограничавани - length, pattern, maxInclusive, maxExclusive, minLength, precision, enumeration, scale и пр.

3.4. Структури

- Def. на типове - <simpleType>, <complexType>

- Декларации на ел. - загавам има и допълнителни ограничения за елемент
<element name="..." type="..." minOccurs="..." maxOccurs="...">

- Декларация на атрибути (всички идат отDef. на елем.)
<attribute name="..." type="..." fixed="..." ...>

- Def. на групи от атрибути:

<attributeGroup name="...">

attrDeclaration

</attributeGroup>

definиции - с права на група, която

указва наименование в основата

Def. - с права на група, която уз

се назовава в идентификацията

- Def. на групи

<group name="...">

models

</group>

- Идент на ограничение - <sequence>, <choice>, <all>, <element> и пр.

- Декларации на нотации - <notations>

- Анонти - <annotation>

3.5. Сравнение с DTD

- XSD поддържа XML syntax, DTD дефиницията от欠缺
- XSD поддържа namespaces и нови типове + да създават нови
- XSD поддържа възможности за по-добри ограничения
- XSD има Entity
- XSD е винаги в отделен файл

XSD - XML schema definition

4. Извличане на XSLT

4.1. XSLT - (extensible Stylesheet language transformation)

- използва се за трансформиране на XML док. с помощта на стилове и форм.
- най-често за трансф. XML → HTML/text file/ друг XML док.
- ползва се за разделяне на представяне и логика от модела на данни

Възможности:

- структурни манипулации на съдържание (CREATE, sort)
- дърв. на модели, която ще се приложи към всички XML док.

XSLT елементи в .xsl файл

- <xsl:stylesheet> - загава обектом на .xsl. XSLT док. и конф. параметри
- <xsl:template> - определящата как се използва обектом от обработката
 - какво и че бъде добавено в изх. док
 - използва кога в модела възникне ел. възможност за attr match
- <xsl:template match = "/>
- <xsl:apply-templates> - община решение как се използва обработката
- <xsl:value-of> - излиза стойността на ел.
- <xsl:for-each> - групова обработка на ел.
- <xsl:sort> - сортира изх. ел.
- <xsl:if>; <xsl:choose>; <xsl:when> и др. - условна обработка

4.2 XPath XPath = //tagname[@Attribute = 'Value']

- за адресиране и манипулиране на секции от XML документ
- ползва се от основните XML спецификации (XSLT, XPointer, XQL)
- локациите се задават чрез URL
- В XPath 1.0 съставните лайми на XML док. са представени като възли,
 - Коренчето е същото като XML док.
 - XPath връща мн-во от възли от дербом, като XML документ
 - XPath израз е низ, чрез който се изброява мн-во от възли на дербом

- XPath процесорът изпълнява XPath израз, зададен във върхом, като се „придвижва по дървото, до указаната възел
Контекстен възел - възелът, в който XPath израз е наименуван
- Базови XPath изрази:
 - корен
 - елемент
 - текст
 - атрибут и др.

- Чрез XPath израз са изброяни елемент, атрибут, инструкции или метод, - дълъга, или - ба от базови, чиста, структурни
- location path - изрази, определящи или - ба от базови

 - relative - спрямо контекстен възел
 - absolute - спрямо корен

5. DOM (Document Object Model) и SAX (Simple API for XML)

5.1 Основни интерфейси и начин на използването им

3 основни стъпки в използването на XML парсер:

- Създаване на парсер обект
- предаване на XML документ на парсера
- Обработка на резултатите

Общият назив за - DOM

Управяванието на съдимия парсери - SAX

DOM - дефинира логическо дърво, представяще първичния документ

DOM core - или - ба от основни интерфейси

• Базови интерфейси - задълж. за всичка DOM имп.

- Node - базов тип единици в DOM - може да е el., текст и др.

- Document - представлява целият XML документ

- Element - елемент в рамките на XML документ

- Attr - атрибут на ел.

- Text - Съдържание на атр. или ел.

• Първи - само за DOM имплементации работещи с XML

- CDATASection

- DocumentType - списък от придр. с - ба на документ

- Notation - декларация в DTD язиката

- Entity

- EntityReference

- Processing Instruction

SAX

② SAX - дер. манипулатори (Handlers), съдържащи методи за разбор на XML док.

SAX 1 интерфейс - DocumentHandler - управление на съдържанието за съдържанието на док. Deprecated

SAX2 - подобен на първия, но с namespace - ContentHandler

- ContentHandler - получава съдържания от парсера
- XMLReader - четещия XML док. използвати на Callbacks
- ErrorHandler - осигурява инфо за възникнали грешки
 - 3 типа - warning, error, FatalError
- DTDHandler - методи за използвани на DTD съдържания
- EntityResolver - определя поведението на парсера при преобразуване на XML Filter

5.2 Удавление

- DOM разчита над XML като логическо дърво, докато SAX е базиран на съдържанието
- DOM е по-добър при добавка на ин.ел. и структурата не е променена XML
- DOM - II - при многократен достъп на документа
 - не чете на ново всеки път
- SAX е по-ефикасен за анализ на голями XML док. и извлечение само на нужни ел.
 - този начин не чете целият
- SAX е идеален за прости операции над XML