XML Validator

XML validator përdoret për të kontrolluar sintaksën e XML dokumentit.

Well Formed XML Dokumentet

Një XML dokument me sintaksën korrekte quhet "Well Formed". Rregullat sintaksore janë përshkruar në ushtrimin e kaluar:

- XML dokumentet duhet të kenë një element rrënjë
- XML elementet duhet të kenë një etiketë mbyllëse
- XML etiketat janë case-sensitive
- XML elementet duhet të jenë të mbivendosura (ndërfutura) në mënyrën e duhur
- XML vlerat e atributeve duhet të jenë të vendosura në thonjëza.

Erroret në XML do të ndalin aplikacionin. Me XML, erroret nuk lejohen.

Valid XML Dokumentet

Një "well formed" XML dokument nuk është i njëjtë si një "valid" XML dokument. Një "valid" XML dokument duhet të jetë well formed. Përveç kësaj, duhet të jetë konform me një *document type definition*. Janë dy definime të ndryshme të llojit të dokumentit që mund të përdoren me XML:

- **DTD** - Document Type Definition
- **XML Schema** - Një DTD alternativ i bazuar në XML

Një DTD definon rregullat dhe elementet dhe atributet legale për një XML dokument.

XML DTD

DTD qëndron për **Document Type Definition**. Një DTD definon strukturën dhe elementet dhe atributet legale të një XML dokumenti.

Një "Valid" XML dokument është "Well Formed", si dhe është konfrom me rregullat e një DTD:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note SYSTEM "Note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

Deklarimi DOCTYPE përmban një referencë në një DTD fajll. Përmbajtja e DTD fajllit tregohet dhe shpjegohet në vazhdim.

XML DTD

Qëllimi i një DTD është të definojmë strukturën dhe elementet dhe atributet legale të një XML dokumenti.:

Note.dtd:

```
<!DOCTYPE note
[
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
```

DTD i mësipërm interpretohet si vijon:

- !DOCTYPE note - Definon që elementi rrënjë i dokumentit është note
- !ELEMENT note - Definon që elementi note duhet të përmbajë elementet: “to, from, heading, body”
- !ELEMENT to - Definon elementin to të jetë e llojit “#PCDATA”
- !ELEMENT from- Definon elementin from të jetë e llojit “#PCDATA”
- !ELEMENT heading - Definon elementin heading të jetë e llojit “#PCDATA”
- !ELEMENT body - Definon elementin body të jetë e llojit “#PCDATA”

#PCDATA nënkupton parseable character data.

Kjo përfaqëson një deklarim eksternal DTD, prandaj definimi <!DOCTYPE> duhet të përmbajë një referencë në DTD fajllin.

Nëse DTD është deklaruar brenda XML fajllit, duhet të jetë në definimin <!DOCTYPE>:

```
<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
```

```
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body>
</note>
```

DTD - XML Building Blocks

Blloqet kryesore të ndërtimit të dokumenteve në XML dhe HTML janë elementet. Duke shikuar nga ana e DTD, të gjithë elementet në XML janë të përbërë nga blloqet vijuese ndërtuese:

- Elementet
- Atributet
- Entitetet
- PCDATA
- CDATA

PCDATA nënkupton parsed character data. Mendoni për character data si një tekst që gjendet ndërmjet etiketës së fillimit dhe të mbarimit të një XML elementi. Parsed character data nuk duhet të përmbajnë karateret me kuptim special në XML. PCDATA është e dhëna që analizohet nga parser-i. Teskti do të kontrollohet nga parseri për entitete dhe markup.

CDATA nënkupton character data. CDATA është tekst që nuk do të analizohet parser-i. Etiketat brenda tekstit nuk do të trajtohen si markup dhe entitete.

DTD - Elements

XML elementet në DTD deklarohen si në vijim:

```
<!ELEMENT element-name category>
or
<!ELEMENT element-name (element-content)>
```

Elementet e zbrazëta:

```
<!ELEMENT element-name EMPTY>
```

```
Example:
<!ELEMENT br EMPTY>
```

```
XML example:
<br />
```

Elementet vetëm me parsed character data deklarohen me #PCDATA brenda kllapave:

```
<!ELEMENT element-name (#PCDATA)>
```

```
Example:
<!ELEMENT from (#PCDATA)>
```

Elementet e deklaruara me fjalën kyçe ANY, mund të përmbajnë ndonjë kombinim të të dhënave të parsueshme (të analizueshme).

```
<!ELEMENT element-name ANY>
```

```
Example:
<!ELEMENT note ANY>
```

Elementet me një ose më shumë fëmijë deklarohen me emrin e elementeve fëmijë brenda kllapave:

```
<!ELEMENT element-name (child1)>
or
<!ELEMENT element-name (child1,child2,...)>
```

```
Example:
<!ELEMENT note (to,from,heading,body)>
```

Kur fëmijët deklarohen në një sekuencë të ndarë me presje, fëmijët duhet të paraqiten në të njëjtën sekuencë në dokumentin. Në deklarimin e plotë, fëmijë gjithashtu duhet të deklarohen, dhe fëmijët mund gjithashtu të kenë fëmijë. Deklarimi i plotë i elementit note:

```
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

Deklarimi i vetëm një paraqitjeje të një elementi

```
<!ELEMENT element-name (child-name)>
```

```
Example:
<!ELEMENT note (message)>
```

Ky shembull deklaron se elementi fëmijë “message” duhet të paraqitet vetëm njëherë, dhe vetëm njëherë brenda elementit “note”.

Deklarimi i minimum një paraqitjeje të një elementi

```
<!ELEMENT element-name (child-name+)>
```

```
Example:
<!ELEMENT note (message+)>
```

Shenja + në këtë shembull deklaron se elementi fëmijë “message” duhet të paraqitet një ose më shumë herë brenda elementit “note”.

Deklarimi i zero ose më shumë paraqitjeve të një elementi

```
<!ELEMENT element-name (child-name*)>
Example:
<!ELEMENT note (message*)>
```

Shenja * në këtë shembull deklaron se elementi fëmijë “message” mund të paraqitet zero ose më shumë herë brenda elementit “note”.

Deklarimi i zero ose një paraqitjeje të një elementi

```
<!ELEMENT element-name (child-name?)>

Example:
<!ELEMENT note (message?)>
```

Shenja ? në këtë shembull deklaron se elementi fëmijë “message” mund të paraqitet zero ose njëherë brenda elementit “note”.

Deklarimi i një elementi ose tjetrit

```
<!ELEMENT note (to,from,header,(message|body))>
```

Shembulli mësipër deklaron se elementi “note” duhet të përmbajë një element “to”, një element “from”, një element “header”, dhe një element “message” ose “body”..

Deklarimi i përmbajtjes së përzier

```
<!ELEMENT note (#PCDATA|to|from|header|message)*>
```

Shembulli i mësipërm deklaron se elementi “note” mund të përmbajë zero ose më shumë paraqitje të parsed character data, elementet “to”, “from”, “header”, ose “message”.

Deklarimi i Atributeve

Deklarimi i atributit ka këtë sintaksën vijuese:

```
<!ATTLIST element-name attribute-name attribute-type attribute-value>

DTD example:
```

```
<!ATTLIST payment type CDATA "check">
```

XML example:

```
<payment type="check" />
```

Attribute-type mund të jetë një nga vijueset:

Lloji	**Përshkrimi**
CDATA	Vlera është character data
(en1|en2|..)	Vlera duhet të jetë një nga lista e numëruar
ID	Vlera është një id unike
IDREF	Vlera është id e elementit tjetër
IDREFS	Vlera është një listë e id-ve tjera
NMTOKEN	Vlera është një emër valid i XML
NMTOKENS	Vlera është një listë e emrave validë të XML
ENTITY	Vlera është një entitet
ENTITIES	Vlera është një listë entitetesh
NOTATION	Vlera është një emër i një shënimi
xml:	Vlera është një vlerë e paradefinuar xml

Attribute-value mund të jetë një nga vijueset:

Vlera	**Shpjegimi**
value	Vlera default e atributit (në thonjëza)
#REQUIRED	Atributi është required
#IMPLIED	Atributi është opsional
#FIXED value	Atributi është fiks

Një atribut me vlerë default

DTD:

```
<!ELEMENT square EMPTY>
<!ATTLIST square width CDATA "0">
```

Valid XML:

```
<square width="100" />
```

Në shembullin më sipër, elementi “square” është definuar të jetë një element i zbrazët me një atribut “width” të llojit CDATA. Nëse nuk është specifikuar atributi width, ai ka një vlerë default 0.

#REQUIRED

Sintaksa

```
<!ATTLIST element-name attribute-name attribute-type #REQUIRED>
```

Shembull

DTD:

```
<!ATTLIST person number CDATA #REQUIRED>
```

Valid XML:

```
<person number="5677" />
```

Invalid XML:

```
<person />
```

Përdorni #REQUIRED nëse nuk keni një opsion për një vlerë default, por ende dëshironi që atributi të jetë i pranishëm.

#IMPLIED

Sintaksa

```
<!ATTLIST element-name attribute-name attribute-type #IMPLIED>
```

Shembull

DTD:

```
<!ATTLIST contact fax CDATA #IMPLIED>
```

Valid XML:

```
<contact fax="555-667788" />
```

Valid XML:

```
<contact />
```

Përdorni #IMPLIED nëse nuk dëshironi të detyroni që autori të përfshijë një atribut, dhe ju nuk keni një opsion për një vlerë default.

#FIXED

Sintaksa

```
<!ATTLIST element-name attribute-name attribute-type #FIXED "value">
```

Shembull

DTD:

```
<!ATTLIST sender company CDATA #FIXED "Microsoft">
```

Valid XML:

```
<sender company="Microsoft" />
```

Invalid XML:

```
<sender company="W3Schools" />
```

Përdorni #FIXED kur dëshironi që një atribut të ketë një vlerë fikse pa lejuar autorin që ta ndryshojë atë. Nëse autori përfshin një tjetër vlerë, XML parser-i do të gjenerojë një gabim.

Vlerat e numëruara të atributeve

Sintaksa

```
<!ATTLIST element-name attribute-name (en1|en2|..) default-value>
```

Shembull

```
DTD:
<!ATTLIST payment type (check|cash) "cash">

XML example:
<payment type="check" />
or
<payment type="cash" />
```

Përdorni vlerat e numëruara të atributeve kur dëshironi që vlera e atributit të jetë një nga një bashkësi fikse e vlerave legale.

Përdorimi i DTD për deklarimin e entiteteve

Entitetet përdoren për të definuar shkurtesa për karakteret speciale. Entitetet mund të deklarohen internale ose eksternale.

Deklarimi i entitetit internal

Sintaksa

```
<!ENTITY entity-name "entity-value">
```

Shembull

```
DTD Example:
<!ENTITY writer "Donald Duck.">
<!ENTITY copyright "Copyright W3Schools.">

XML example:
<author>&writer;&copyright;</author>
```

Deklarimi i entitetit eksteral

Sintaksa

```
<!ENTITY entity-name SYSTEM "URI/URL">
```

Shembull

```
DTD Example:
```

```
<!ENTITY writer SYSTEM "https://www.w3schools.com/entities.dtd">
<!ENTITY copyright SYSTEM "https://www.w3schools.com/entities.dtd">
```

XML example:

```
<author>&writer;&copyright;</author>
```

DOCTYPE deklarimi mund gjithashtu të përdoret për të definuar karakteret speciale ose përdoren në dokument:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE note [
<!ENTITY nbsp " ">
<!ENTITY writer "Writer: Donald Duck.">
<!ENTITY copyright "Copyright: W3Schools.">
]>

<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
<footer>&writer; &copyright;</footer>
</note>
```

Një *entitet* ka tre pjesë: fillon me një simbol (&), pastaj vjen emri i entitetit, dhe mbaron me një pikëpresje (;).

Kur të përdorim DTD?

Me një DTD, grupet e pavarura të njerëzve mund të pajtohen të përdorin një standard DTD për shkëmbim të të dhënave.
Me një DTD, ju mund të verifikoni që e dhëna që e pranoni nga bota e jashtme është valide.
Mund gjithashtu të përdorni DTD për të verifikuar të dhënat e juaja.

Kur të mos përdorim DTD?

XML nuk kërkojnë një DTD.
Kur jeni duke eksperimentuar me XML, ose kur jeni duke punuar me fajlla të vegjël XML, krijimi i DTD-së mund të jetë konsum kohor.

Shembuj & Detyra

1. XML Dokumenti përmban të dhënat për një gazetë. Elementi rrënjë i XML dokumentit është vetë gazeta. Elementi gazeta përmban një ose më shumë artikuj. Artikujt përmbajnë disa elemente fëmijë, kryetitullin, kryesuesin, trupin, dhe shënimet. Këto elemente duhet të paraqiten vetëm nga njëherë brenda një artikulli. Secili element fëmijë i artikullit përmban parsed character data. Elementi artikulli përmban disa atribute, autori, editori, data, dhe botimi. Të gjitha atributet janë të llojit character data, secili atribut është opsional përveqç atributit autori që patjetër duhet të jetë i pranishëm. Dokumenti poashtu përmban disa entitete. Gazeta është entitet me vlerën "Koha ditore", publikuesi me vlerën e entitetit "Botimet Koha" dhe copyyright e drejta e autorit me vlerën "Copyright 2000 Botimet Koha".

2. XML Dokumenti përmban të dhëna për një bankë. Elementi rrënjë i XML dokumentit është vetë banka. Elementi banka përmban një ose më shumë llogari bankare. Llogaritë bankare përmbajnë disa elemente fëmijë, numrin e xhirollogarisë bankare, data e hapjes së llogarisë bankare, bilanci, dhe mbajtësi i llogarisë bankare. Këto elemente duhet të paraqiten vetëm njëherë brenda elementit llogaria bankare. Elementi llogaria bankare përmban disa atribute, lloji (kategoria) dhe statusi. Lloji i llogarisë mund të jetë "debi" ose "kredi", ndërsa statusi mund të jetë ose "aktiv" ose "pasiv". Atributi statusi është opsional ndërsa lloji i llogarisë bankare duhet patjetër të jetë i pranishëm. Elementet numri i llogarisë, data e hapjes së llogarisë bankare, dhe bilanci përmbajnë parsed character data. Elementi mbajtësi i llogarisë bankare përmban dy elemente fëmijë, emri dhe mbiemri i personit mbajtës së llogarisë bankare, të cilat duhet të jenë të vetme. Të dy elementet fëmijë përmbajnë parsed character data.