

Úvod do XML Schema

(zdroj: <http://w3schools.com>)

XML Schema je na XML založená alternatíva k DTD (Document Type Definition).

XML Schema popisuje **štruktúru XML dokumentu**.

XML Schema jazyk je tiež označovaný ako **XML Schema Definition (XSD)**.

Čo je XML Schema?

Účelom jazyka XML Schema a aj DTD je definovať legálne (povolené) stavebné bloky XML dokumentu.

XML Schema definuje:

- elementy, ktoré sa môžu objaviť v XML dokumente
 - atribúty, ktoré sa môžu objaviť v XML dokumente
 - ktoré elementy sú detskými elementmi
 - poradie detských elementov
 - počet detských elementov
 - či je element prázdny alebo či môže obsahovať text
 - dátové typy elementov a atribútov
 - defaultové a pevné hodnoty elementov a atribútov
-

XML Schémy sú následníci DTD

W3C predpokladá, že čoskoro budú XML Schémy použité vo väčšine webových aplikácií ako náhrada za DTD, z týchto dôvodov:

- XML Schémy sú rozšíriteľné budúcimi rozšíreniami (nastavbami)
 - XML Schémy sú bohatšie a výkonnejšie ako DTD
 - XML Schémy sú napísané v XML
 - XML Schémy podporujú dátové typy
 - XML Schémy podporujú priestory názvov (menné priestory)
-

XML Schema je W3C Recommendation (odporúčaním)

XML Schema sa stala W3C Recommendation (odporúčaním) 2 mája 2001.

Ako vytvoriť XSD?

XML dokumenty môžu mať referenciu na DTD alebo na XML Schema.

Jednoduchý XML dokument

Tento XML dokument je nazvaný "note.xml".

```
<?xml version="1.0"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

XML Schema

Nasledujúci príklad je súbor "note.xsd" vytvorený v XML Schema jazyku, ktorý definuje elementy vyššieho XML dokumentu ("note.xml").

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">

  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

note element je komplexného typu (complexType), pretože obsahuje iné elementy (to, from, heading, body), ktoré sú jednoduchého typu (simpleType), pretože oni neobsahujú iné elementy.

Referencia na XML Schemu

Tento XML dokument má referenciu na XML Schemu.

```
<?xml version="1.0"?>

<note
xmlns="http://www.w3schools.com"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3schools.com note.xsd">
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

XSD <schema> element

<schema> element je koreňovým elementom každej XML Schemy.

<schema> element

```
<?xml version="1.0"?>
```

```
<xs:schema>
```

```
...
```

```
...
```

```
</xs:schema>
```

<schema> element môže obsahovať atribúty. Deklarácia schémy je často nasledujúca:

```
<?xml version="1.0"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">
...
...
</xs:schema>
```

Nasledujúci fragment

```
xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

indikuje, že elementy a dátové typy použité v schéme pochádzajú z priestoru mien (názvov) "http://www.w3.org/2001/XMLSchema".

Taktiež indikuje, že elementy a dátové typy, ktoré pochádzajú z priestoru mien "http://www.w3.org/2001/XMLSchema" by mali mať prefix **xs**:

Tento fragment

```
targetNamespace="http://www.w3schools.com"
```

indikuje, že elementy definované touto schémou (note, to, from, heading, body.) pochádzajú z priestoru mien "http://www.w3schools.com".

Tento fragment

```
xmlns="http://www.w3schools.com"
```

indikuje, že defaultový priestor mien je "http://www.w3schools.com".

Tento fragment

```
elementFormDefault="qualified"
```

indikuje, že hocikáký element použitý XML inštanciou tohto dokumentu, ktorý bol deklarovaný touto schémou, musí byť kvalifikovaný priestorom mien.

Jednoduchý XSD element

XML Schémy definujú elementy XML súborov.

Jednoduchý element je XML element, ktorý **obsahuje len text**. **Nemôže obsahovať iné elementy** alebo **atribúty**.

Čo je jednoduchý element?

Jednoduchý element je XML element, ktorý **môže obsahovať len text**. **Nemôže obsahovať iné elementy** alebo **atribúty**.

Avšak textu môže byť veľa rôznych typov zahrnutých v XML Schema definícii (boolean, string, date atď.) alebo to môže používateľom definovaný typ textu. Na dátový typ elementu môžeme uplatniť reštrikcie, pomocou ktorých môžeme limitovať obsah elementu alebo môžeme požadovať obsah elementu priradený špecifikovaným vzorom (šablónou).

Definovanie jednoduchého elementu

Syntax pre jeho definíciu je nasledovná:

```
<xs:element name="xxx" type="yyy"/>
```

, kde xxx je názov elementu a yyy je jeho dátový typ.

XML Schema má veľa vstavaných dátových typov. Najobecnejšie typy sú:

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date

- xs:time

Príklad

Majme nasledujúce XML elementy:

```
<lastname>Refsnes</lastname>
<age>36</age>
<dateborn>1970-03-27</dateborn>
```

Tu sú im zodpovedajúce definície jednoduchých elementov:

```
<xs:element name="lastname" type="xs:string"/>
<xs:element name="age" type="xs:integer"/>
<xs:element name="dateborn" type="xs:date"/>
```

XSD atribúty

Všetky atribúty sú deklarované ako jednoduché typy.

Čo je atribút?

Jednoduché elementy nemôžu mať atribúty. Ak element má atribúty, tak je elementom komplexného typu, ale atribút samotný je vždy deklarovaný ako jednoduchý typ.

Ako definovať atribút?

Syntax definície atribútu je nasledovná:

```
<xs:attribute name="xxx" type="yyy"/>
```

, kde xxx je názov atribútu a yyy je jeho dátový typ.

XML Schema má veľa vstavaných dátových typov. Najobecnejšie typy sú:

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date
- xs:time

Príklad

Tu je XML element s atribútom:

```
<lastname lang="EN">Smith</lastname>
```

a tu je zodpovedajúca definícia atribútu:

```
<xs:attribute name="lang" type="xs:string"/>
```

Reštrikcie obsahu

Definovaním dátového typu XML elementu alebo atribútu je zároveň daná reštrikcia ich obsahu. Napr. ak je XML element dátového typu "xs:date" a obsahuje reťazec "Hello World", tak takýto element nebude validný (platný).

Pomocou XML Schém môžeme pridávať svoje vlastné reštrikcie na XML elementy a atribúty. Tieto reštrikcie sa nazývajú aspekty (fazety).

XSD reštrikcie/aspekty (fazety)

Reštrikcie sa používajú na definovanie akceptovateľných hodnôt pre XML elementy a atribúty. Retrikcie na obsah XML elementov sa nazývajú aspekty (fazety).

Reštrikcie na hodnoty

Nasledujúci príklad definuje element "age" s reštrikciou. Hodnota elementu "age" nemôže byť menšia ako 0 alebo väčšia ako 120.

```
<xs:element name="age">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="120"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Reštrikcie na sadu hodnôt

Pre limitáciu obsahu XML elementu sadou akceptovateľných hodnôt môžeme použiť enumeration obmedzenie.

Nasledujúci príklad definuje element "car" s reštrikciou. Jedinými akceptovateľnými hodnotami tohto elementu sú nasledujúce: Audi, Golf alebo BMW.

```
<xs:element name="car">
  <xs:simpleType>
```

```

<xs:restriction base="xs:string">
  <xs:enumeration value="Audi"/>
  <xs:enumeration value="Golf"/>
  <xs:enumeration value="BMW"/>
</xs:restriction>
</xs:simpleType>
</xs:element>

```

Uvedený príklad by mohol byť napísaný aj takto:

```

<xs:element name="car" type="carType"/>

<xs:simpleType name="carType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Audi"/>
    <xs:enumeration value="Golf"/>
    <xs:enumeration value="BMW"/>
  </xs:restriction>
</xs:simpleType>

```

Pozn.: V tomto prípade typ "carType" môže byť použitý na definovanie iných elementov, pretože nie je časťou "car" elementu.

Reštrikcie na sériu hodnôt

Obsah XML elementu môžeme limitovať definovaním série čísiel alebo písmen (zakov), ktoré budú použité ako šablónové obmedzenie.

Nasledujúci príklad definuje element "letter" s reštrikciou. Jedinou akceptovateľnou hodnotou tohto elementu je **JEDEN malý znak zo série znakov a až z**.

```

<xs:element name="letter">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-z]"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

```

Ďalší príklad definuje element "initials" s reštrikciou. Jedinými akceptovateľnými hodnotami tohto elementu sú **TRI veľké znaky, každý zo série znakov A až Z**.

```

<xs:element name="initials">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Z][A-Z][A-Z]"/>
    </xs:restriction>
  </xs:simpleType>

```

```
</xs:element>
```

Ďalší príklad tiež definuje element "initials" s reštrikciou. Jedinými akceptovateľnými hodnotami tohto elementu sú **TRI veľké** alebo **malé znaky**, **každý zo série znakov a až z** alebo **A až Z**.

```
<xs:element name="initials">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z][a-zA-Z][a-zA-Z]" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Ďalší príklad definuje element "choice" s reštrikciou. Jedinou akceptovateľnou hodnotou tohto elementu je **JEDEN z nasledujúcich znakov x, y** alebo **z**.

```
<xs:element name="choice">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[xyz]" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Ďalší príklad definuje element "prodid" s reštrikciou. Jedinými akceptovateľnými hodnotami tohto elementu je **PÄŤ číslic v sekvencii**, pričom **každá číslica musí byť z rozsahu 0 až 9**.

```
<xs:element name="prodid">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:pattern value="[0-9][0-9][0-9][0-9][0-9]" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Reštrikcie na dĺžku

Pre limitovanie dĺžky hodnoty v elemente môžeme použiť **length**, **maxLength** a **minLength** obmedzenia.

Nasledujúci príklad definuje element "password" s reštrikciou. Jeho hodnota musí byť presne 8-znaková.

```
<xs:element name="password">
  <xs:simpleType>
    <xs:restriction base="xs:string">
```



```
<xs:length value="8"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
```

Nasledujúci príklad definuje element "password" s reštrikciou. Jeho hodnota musí byť minimálne 5-znaková a maximálne 8-znaková.

```
<xs:element name="password">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:minLength value="5"/>
      <xs:maxLength value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Reštrikcie na dátové typy (tabuľka s popismi v angličtine)

Constraint	Description
enumeration	Defines a list of acceptable values
fractionDigits	Specifies the maximum number of decimal places allowed. Must be equal to or greater than zero
length	Specifies the exact number of characters or list items allowed. Must be equal to or greater than zero
maxExclusive	Specifies the upper bounds for numeric values (the value must be less than this value)
maxInclusive	Specifies the upper bounds for numeric values (the value must be less than or equal to this value)
maxLength	Specifies the maximum number of characters or list items allowed. Must be equal to or greater than zero
minExclusive	Specifies the lower bounds for numeric values (the value must be greater than this value)
minInclusive	Specifies the lower bounds for numeric values (the value must be greater than or equal to this value)
minLength	Specifies the minimum number of characters or list items allowed. Must be equal to or greater than zero
pattern	Defines the exact sequence of characters that are acceptable
totalDigits	Specifies the exact number of digits allowed. Must be greater than zero
whiteSpace	Specifies how white space (line feeds, tabs, spaces, and carriage returns) is handled

Komplexné XSD elementy

Komplexný element obsahuje iné elementy a/alebo atribúty.

Čo je komplexný element?

Komplexný element je XML element, ktorý **obsahuje iné elementy a/alebo atribúty**.

Poznáme 4 druhy komplexných elementov:

- prázdne elementy
- elementy obsahujúce len iné elementy
- elementy obsahujúce len text
- elementy obsahujúce oboje, čiže iné elementy a text

Pozn.: Každý z týchto elementov môže obsahovať atribúty.

Príklady komplexných elementov

Komplexný XML element "product", ktorý je **prázdny**:

```
<product pid="1345"/>
```

Komplexný XML element "employee", ktorý **obsahuje len iné elementy**:

```
<employee>
  <firstname>John</firstname>
  <lastname>Smith</lastname>
</employee>
```

Komplexný XML element "food", ktorý **obsahuje len text**:

```
<food type="dessert">Ice cream</food>
```

Komplexný XML element "description", ktorý **obsahuje oboje, elementy a text**:

```
<description>
  It happened on<date lang="norwegian">03.03.99</date> ....
</description>
```

Ako definovať komplexný element

Nasledujúci komplexný XML element "employee", ktorý **obsahuje len iné elementy**:

```
<employee>
  <firstname>John</firstname>
  <lastname>Smith</lastname>
</employee>
```

Tento komplexný element môžeme v XML Schema definovať dvomi spôsobmi:

1. "employee" element môže byť deklarovaný priamo pomenovaním elementu:

```

<xs:element name="employee">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Ak použijeme uvedenú metódu, tak **jediný "employee" element môže použiť špecifikovaný komplexný typ**. Jeho detské elementy "firstname" and "lastname" sú ohraničené <sequence> indikátorom. To znamená, že tieto detské elementy sa musia v elemente "employee" objaviť v takto deklarovanom poradí.

2. "employee" element môže mať **atribút type**, ktorý odkazuje na názov použitého komplexného typu.

```

<xs:element name="employee" type="personinfo"/>

<xs:complexType name="personinfo">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

```

Ak použijeme ostatnú uvedenú metódu, tak **viacero elementov sa môže odkazovať na rovnaký komplexný typ**.

```

<xs:element name="employee" type="personinfo"/>
<xs:element name="student" type="personinfo"/>
<xs:element name="member" type="personinfo"/>

<xs:complexType name="personinfo">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

```

Taktiež môžeme komplexný element založiť na existujúcom komplexnom elemente a pridávať do neho ďalšie elementy.

```

<xs:element name="employee" type="fullpersoninfo"/>

<xs:complexType name="personinfo">
  <xs:sequence>

```

```

<xs:element name="firstname" type="xs:string"/>
<xs:element name="lastname" type="xs:string"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="fullpersoninfo">
  <xs:complexContent>
    <xs:extension base="personinfo">
      <xs:sequence>
        <xs:element name="address" type="xs:string"/>
        <xs:element name="city" type="xs:string"/>
        <xs:element name="country" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

XSD Indikátory

pomocou nich môžeme riadiť **ako** a **koľko elementov** je použitých v dokumentoch.

Indikátory

Poznáme 7 indikátorov:

Indikátory poradia:

- All
- Choice
- Sequence

Indikátory výskytu:

- maxOccurs
- minOccurs

Skupinové indikátory:

- Group name
- attributeGroup name

Indikátory poradia

sa používajú na definovanie poradia elementov.

All indikátor

<all> indikátor špecifikuje, že detské elementy sa môžu objaviť v ľubovoľnom poradí a že každý detský element sa musí vyskytnúť len jeden krát:

```

<xs:element name="person">
  <xs:complexType>

```

```
<xs:all>
  <xs:element name="firstname" type="xs:string"/>
  <xs:element name="lastname" type="xs:string"/>
</xs:all>
</xs:complexType>
</xs:element>
```

Pozn.: Keď používame <all> indikátor, tak <minOccurs> indikátor môžeme nastaviť na 0 alebo 1 a <maxOccurs> indikátor môže byť nastavený len na 1.

Choice indikátor

<choice> indikátor špecifikuje, že buď jeden alebo iný detský element sa môže vyskytnúť:

```
<xs:element name="person">
  <xs:complexType>
    <xs:choice>
      <xs:element name="employee" type="employee"/>
      <xs:element name="member" type="member"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

Sequence indikátor

<sequence> indikátor špecifikuje, že detské elementy sa musia objaviť v špecifikovanom poradí:

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Indikátory výskytu

sa používajú na definovanie počtu výskytov elementu.

Pozn.: Pre všetky indikátory poradia a skupinové indikátory (any, all, choice, sequence, group name, a group reference) je defaultová hodnota pre maxOccurs a minOccurs indikátory 1.

maxOccurs indikátor

<maxOccurs> indikátor špecifikuje maximálny počet výskytov elementu:

```

<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="full_name" type="xs:string"/>
      <xs:element name="child_name" type="xs:string" maxOccurs="10"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Uvedený príklad použitia tohto indikátora indikuje, že detský element "child_name" sa môže vyskytnúť minimálne 1-krát (defaultová hodnota minOccurs je 1) a maximálne 10-krát v elemente "person".

minOccurs indikátor

<minOccurs> indikátor špecifikuje minimálny počet výskytov elementu:

```

<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="full_name" type="xs:string"/>
      <xs:element name="child_name" type="xs:string" maxOccurs="10" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Uvedený príklad použitia tohto indikátora indikuje, že detský element "child_name" sa môže vyskytnúť minimálne 0-krát (defaultová hodnota minOccurs je 1) a maximálne 10-krát v elemente "person".

Tip: Ak chceme umožniť elementu nelimitovaný počet výskytov, tak použijeme nasledujúci príkaz: maxOccurs="unbounded"

Funkčný príklad:

Majme XML súbor "Myfamily.xml" s nasledujúcim obsahom:

```

<?xml version="1.0" encoding="ISO-8859-1"?>

<persons xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="family.xsd">

  <person>
    <full_name>Hege Refsnes</full_name>
    <child_name>Cecilie</child_name>
  </person>

  <person>
    <full_name>Tove Refsnes</full_name>

```

```

    <child_name>Hege</child_name>
    <child_name>Stale</child_name>
    <child_name>Jim</child_name>
    <child_name>Borge</child_name>
  </person>

  <person>
    <full_name>Stale Refsnes</full_name>
  </person>
</persons>

```

Tento XML súbor obsahuje koreňový element "persons". V ňom sú definované 3 "person" elementy. Každý "person" element **musí** obsahovať detský element "full_name" a **môže** obsahovať až 5 detských elementov "child_name".

Tu je schémový súbor "family.xsd" k tomuto XML súboru, definujúci požadované reštrikcie:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <xs:element name="persons">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="person" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="full_name" type="xs:string"/>
              <xs:element name="child_name" type="xs:string" minOccurs="0" maxOccurs="5"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>

```