

XML Schema

16 janvier 2014

emeline.dorey@gmail.com

Sommaire

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

- 1 Introduction
- 2 Structure
- 3 Commentaires
- 4 Association XML Schema
- 5 Les types
- 6 Types simples
- 7 Types composés

Introduction

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

- DTD limitées \Rightarrow création en 2001, par le W3C, du langage XML Schema, pour corriger les défauts des DTD.
 - Nouveau langage de schéma.
 - Il est écrit en XML.
 - Il est plus complet et plus puissant que celui des DTD.
 - Il offre un meilleur contrôle du contenu des documents XML.

Introduction

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

- Le but d'un schéma est de décrire :
 - les autorisations d'imbrications,
 - l'ordre d'apparition des éléments et de leurs attributs.
- Définition des éléments XML :
 - DTD \Rightarrow utilisation d'un type unique,
 - XML Schema \Rightarrow utilisation de différents types.

Structure

XML Schema

Introduction

Structure

Commentaires

Association XML Schema

Les types

Types simples

Types composés

Attributs

- Un XML Schema est un document XML, donc il doit :
 - débuter par un prologue,
 - avoir un élément racine **schema**,
 - respecter la syntaxe XML.
- Ce schéma est sauvegardé dans un fichier ayant l'extension **.xsd**

Structure

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

L'élément racine doit contenir l'espace de noms du W3C, contenant tous les éléments et les types de données de XML Schema.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema
  ">
  <!-- declarations d'elements, d'attributs... -->
</xs:schema>
```

A retenir : Tout élément d'un schéma doit débiter par le préfixe **xs** pour distinguer les éléments appartenant au langage à proprement parlé, des éléments et attributs définis par un schéma donné.

Commentaires

XML Schema

Introduction

Structure

Commentaires

Association XML Schema

Les types

Types simples

Types composés

Attributs

- Il est possible de placer dans un XML Schema des commentaires XML, entre les balises `<!-- -->`, mais ils ne sont pas transmis au cours des traitements.
- XML Schema introduit des commentaires (nommés *annotations*) pouvant être analysés et traités \Rightarrow ce sont eux-mêmes des éléments.

Annotation/Commentaire

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

```
...  
<xs:annotation>  
  <xs:documentation>  
    L'annotation permet de documenter le schema et d'  
    en faciliter la maintenance.  
  </xs:documentation>  
</xs:annotation>  
...
```


Association XML Schema

XML Schema

Introduction

Structure

Commentaires

Association XML Schema

Les types

Types simples

Types composés

Attributs

Pour valider un document XML par rapport à un XML Schema, il faut ajouter son emplacement à la déclaration de l'élément racine du document XML :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<biblio
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance"
  xsi:noNamespaceSchemaLocation="monFichier.xsd">
  ...
</biblio>
```

Les types

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

Les éléments XML peuvent être définis avec XML Schema selon deux types :

- Type simple \Rightarrow élément XML qui ne contient que du "texte".
- Type composé \Rightarrow élément XML qui contient des attributs et/ou des éléments fils.

Types simples

XML Schema

Introduction

Structure

Commentaires

Association XML Schema

Les types

Types simples

Types composés

Attributs

- Les types simples ne contiennent pas d'attribut, ni d'élément fils.
- Ils contiennent une seule valeur (date, nombre, chaîne de caractères...).
- De nombreux types sont prédéfinis dans la bibliothèque de types intégrés de XML Schema.

Types simples

XML Schema

Introduction

Structure

Commentaires

Association XML Schema

Les types

Types simples

Types composés

Attributs

La déclaration de ces éléments est de la forme :

```
<xs:element name="nom_elt" type="type_elt" />
```

- **nom_elt** ⇒ nom de l'élément défini
- **type_elt** ⇒ type prédéfinis dans XML Schema

Types simples

Exemples de types prédéfinis

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

- **xs:string** ⇒ l'élément doit contenir une chaîne de caractères
- **xs:decimal** ⇒ l'élément doit contenir un nombre décimal, positif ou négatif
- **xs:integer** ⇒ l'élément doit contenir un nombre entier, positif ou négatif
- **xs:boolean** ⇒ l'élément doit contenir **true** ou **false**
- **xs:date** ⇒ l'élément doit contenir une date au format *AAAA-MM-JJ*
- **xs:time** ⇒ l'élément doit contenir une heure au format *hh:mm:ss*

Types simples

Types simple prédéfinis, présentés sur le site du W3C en arbre

XML Schema

Introduction

Structure

Commentaires

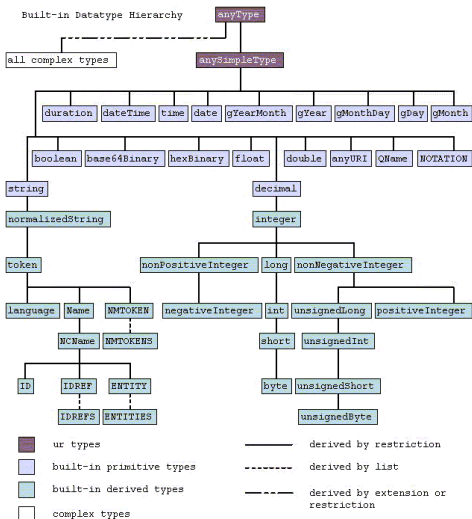
Association XML Schema

Les types

Types simples

Types composés

Attributs



Types simples

Prédéfinir le contenu d'un élément

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

Il est possible de prédéfinir le contenu d'un élément en ajoutant à la déclaration de l'élément :

- **fixed="valeur"** \Rightarrow valeur imposée. L'élément courant ne sera valide que si son contenu est égal à *valeur*.
- **default="valeur"** \Rightarrow valeur par défaut utilisée si l'élément courant est vide ou omis.

Il n'est pas possible d'utiliser ces deux attributs pour le même élément (ils se contredisent).

Types simples personnalisés

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

- XML Schema permet la création de types simples personnalisés grâce à la redéfinition de ceux prédéfinis
⇒ dérivation de types.
- Deux techniques de dérivation de types :
 - dérivation par restriction du type ancêtre,
 - dérivation par extension du type ancêtre,
 - utiliser un type prédéfini,
 - dériver l'un de ces types pour créer un type simple personnalisé.

Types simples personnalisés

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

Déclaration des types simples personnalisés :

```
<xs:element name="nom_elt">  
  <xs:simpleType>  
    ...  
  </xs:simpleType>  
</xs:element>
```

L'élément **<xs:simpleType>** contient la définition du type simple dérivé.

Types simples personnalisés

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

- La dérivation des types simples prédéfinis s'effectuent en spécifiant certaines caractéristiques et en positionnant des contraintes.
- Ces restrictions sont des **facettes**.

Types simples personnalisés

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

Par exemple, le type prédéfini **string** dispose de six facettes modifiables :

- **length** ⇒ longueur de la chaîne.
- **minLength** ⇒ longueur minimale.
- **maxLength** ⇒ longueur maximale.
- **pattern** ⇒ expression régulière.
- **enumeration** ⇒ énumération de valeurs possibles.
- **whitespace** ⇒ gestion des espaces.

Types simples personnalisés

Exemple

XML Schema

Introduction

Structure

Commentaires

Association XML Schema

Les types

Types simples

Types composés

Attributs

```
<xs:element name="resume">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="220" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

L'élément **resume** est une chaîne de caractères dont la longueur est limitée à 220 caractères.

Types simples personnalisés

- Type simple personnalisé créé : peut être défini une unique fois et utilisé par plusieurs éléments du XML Schema \Rightarrow définition d'un type personnalisé nommé.
- Donner un nom au nouveau type \Rightarrow ajout de l'attribut **name** à la déclaration de la balise `<xs:simpleType>`.

Exemple de déclaration :

```
<xs:simpleType name="type_isbn">  
  ...  
</xs:simpleType>
```

Exemple d'utilisation :

```
<xs:element name="isbn" type="type_isbn" />
```

Types composés

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

- Les éléments de types composés peuvent posséder des éléments fils et/ou des attributs
- Quatre types composés :
 - Texte \Rightarrow contenu simple (texte et attributs)
 - Élément \Rightarrow contenu composé (éléments fils et attributs)
 - Élément vide \Rightarrow contenu composé (attributs)
 - Contenu mixte \Rightarrow contenu simple et composé (texte, éléments fils et attributs)

Types composés

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

Déclaration des types composés :

```
<xs:element name="nom_elt">  
  <xs:complexType>  
    ...  
  </xs:complexType>  
</xs:element>
```

L'élément **<xs:complexType>** contient la définition du type composé.

Types composés

Séquences d'éléments

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

DTD \Rightarrow `<!ELEMENT livre (titre, auteur, editeur)>`

- Déclaration de l'élément **livre**, contenant les éléments **titre**, **auteur** et **editeur** dans cet ordre fixe.

Pour faire de même avec XML Schema il faut utiliser les séquences :

```
<xs:element name="livre">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="titre" type="xs:string" />
      <xs:element name="auteur" type="xs:string" />
      <xs:element name="editeur" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```


Types composés

Choix d'élément

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

DTD \Rightarrow `<!ELEMENT livre (titre | auteur | editeur)>`

- Déclaration de l'élément **livre** comme pouvant contenir soit l'élément **titre**, soit l'élément **auteur**, soit l'élément **editeur**.

Pour faire de même avec XML Schema il faut utiliser la notion de choix :

```
<xs:element name="livre">
  <xs:complexType>
    <xs:choice>
      <xs:element name="titre" type="xs:string" />
      <xs:element name="auteur" type="xs:string" />
      <xs:element name="editeur" type="xs:string" />
    </xs:choice>
  </xs:complexType>
</xs:element>
```

Types composés

Exemple

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

DTD \Rightarrow `<!ELEMENT livre (titre, (auteur | editeur))>`

- L'élément **livre** doit contenir un **titre** et un **auteur** ou un **editeur**.

Déclaration XML Schema :

```
<xs:element name="livre">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="titre" type="xs:string" />
      <xs:choice>
        <xs:element name="auteur" type="xs:string" />
        <xs:element name="editeur" type="xs:string" /
          >
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Types composés

Définition non ordonnées

XML Schema

Introduction

Structure

Commentaires

Association XML Schema

Les types

Types simples

Types composés

Attributs

- En XML Schema il est possible d'indiquer que les éléments enfants doivent apparaître une fois ou pas du tout, dans n'importe quel ordre \Rightarrow `<xs :all>`
- `<xs :all>` doit être un enfant direct de l'élément `<xs :complexType>`

Types composés

Définition non ordonnées

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

```
<xs:element name="livre">
  <xs:complexType>
    <xs:all>
      <xs:element name="titre" type="xs:string" />
      <xs:element name="auteur" type="xs:string" />
      <xs:element name="editeur" type="xs:string" />
    </xs:all>
  </xs:complexType>
</xs:element>
```

Dans un élément **livre**, les éléments **titre**, **auteur**, et **editeur** peuvent apparaître une fois ou pas du tout, dans un ordre quelconque.

Types composés

Contenu mixte

XML Schema

Introduction

Structure

Commentaires

Association XML Schema

Les types

Types simples

Types composés

Attributs

- Contenu mixte \Rightarrow lorsqu'un élément contient à la fois des éléments fils et du texte.
- Définition :

```
<xs:complexType mixed="true">  
  ...  
</xs:complexType>
```

L'attribut **mixed** permet d'indiquer que du texte peut se glisser entre les sous-éléments autorisés.

Types composés

Définition globale

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

Il est possible de déclarer un élément globalement et y faire référence ailleurs dans le fichier XML Schema.

```
<xs:element name="nom_elt">  
  <xs:complexType>  
    ...  
  </xs:complexType>  
</xs:element>
```

Appel :

```
<xs:element ref="nom_elt" />
```

Types composés

Définition du nombre d'occurrences

XML Schema

Introduction

Structure

Commentaires

Association XML Schema

Les types

Types simples

Types composés

Attributs

- Le nombre d'occurrences possibles dans un document XML peut être contrôlé avec XML Schema :
 - **minOccurs="n"** ⇒ nombre minimal d'occurrences de l'élément courant.
 - Valeur par défaut de cet attribut ⇒ **1**.
 - Valeur possible ⇒ entier positif ou nul.
 - Associé à `<xs :all>`, cet attribut ne peut valoir que 0 ou 1.
 - **maxOccurs="n"** ⇒ nombre maximal d'occurrences de l'élément courant.
 - Valeur par défaut de cet attribut ⇒ **1**.
 - Valeur possible ⇒ entier positif ou nul, ou le mot **unbounded** pour ne pas fixer de limite.
 - Associé à `<xs :all>`, cet attribut vaut obligatoirement 1.

Types composés

Définition du nombre d'occurrences

XML Schema

Ces attributs ne peuvent pas être utilisés dans la définition d'un élément global \Rightarrow ils n'ont de sens que localement.

```
<xs:element name="livre">
  <xs:complexType>
    <xs:choice>
      <xs:element name="titre" type="xs:string"
        minOccurs="1" maxOccurs="3" />
      <xs:element name="auteur" type="xs:string"
        minOccurs="1" maxOccurs="unbounded"/>
      <xs:element name="editeur" type="xs:string"
        minOccurs="0" maxOccurs="1" />
    </xs:choice>
  </xs:complexType>
</xs:element>
```


Attributs

XML Schema

Introduction

Structure

Commentaires

Association XML
Schema

Les types

Types simples

Types composés

Attributs

Attributs :

- Éléments de type simple (ne contiennent pas d'élément fils ni d'attribut).
- Ne peuvent apparaître que dans des éléments de type composé.
- Déclaration avant la balise fermante `</xs :complexType>`.

Attributs

XML Schema

Introduction

Structure

Commentaires

Association XML Schema

Les types

Types simples

Types composés

Attributs

Définition d'un attribut :

```
<xs:attribute name="nom_attr" type="type_attr" />
```

- **nom_attr** \Rightarrow nom de l'attribut
- **type_attr** \Rightarrow type simple prédéfini ou nommé

Attributs

XML Schema

Introduction

Structure

Commentaires

Association XML Schema

Les types

Types simples

Types composés

Attributs

- Par défaut un attribut est toujours facultatif.
- Forcer la présence ou l'absence d'un attribut grâce à l'attribut **use** :
 - **use="required"** ⇒ attribut obligatoire
 - **use="prohibited"** ⇒ attribut absent
- L'attribut **value="valeur"** peut être associé à l'attribut, où **valeur** est la seule valeur acceptable pour cet attribut.