

Revision MLBDA XML

DTD

- Une DTD est composée d'une suite de déclarations pour tous les composants utilisés dans un document XML :
 - Éléments :
 - Attributs d'éléments :
 - Entités :
 - Notations :
- **Déclaration d'Eléments**
 - Chaque nom d'élément est une expression régulière
 - Si E1 et E2 sont des expressions régulières, alors
 - E1, E2 : concaténation
 - E1|E2 : choix
 - **E1*** : fermeture avec zéro ou plusieurs occurrences
 - **E1+** : fermeture avec au moins une occurrence
 - **E1 ?** : option (0 ou 1 fois)
 - #PCDATA Parsed Character Data :
 - Contenu texte
 - Ex:
 - Modèle de contenu mixte :
 - (#PCDATA | élément1 | élément2 | ... | élémentn)*
 - Pour mêler contenu textuel et éléments
 - Modèle de contenu vide : EMPTY
 - Modèle de contenu libre : ANY
- **declaration des attributs**
 - Les attributs ne sont pas ordonne mais les elements oui
 - Type-attribut :
 - CDATA : la valeur de l'attribut est une chaîne de caractères
 - ID : identificateur d'élément,
 - IDREF(S) : renvoi vers un (des) ID
 - NMTOKEN(S) : un ou des noms symboliques (sans blanc)
 - (a | b | c...) : type énumération de valeurs possibles
 - ENTITY(IES) : entités externes non XML
 - Mode-défaut :
 - Valeur par défaut
 - Valeur constante : #FIXED
 - Présence obligatoire : #REQUIRED
 - Présence facultative : #IMPLIEDe sont pas ordonnés alors que les éléments le sont
- exemple dtd

```

<!ELEMENT base ((restaurant | ville)+)>
<!ELEMENT restaurant (fermeture?, menu,menu+)>
<!ATTLIST restaurant
    nom CDATA #REQUIRED
    etoile (0|1|2|3) #REQUIRED
    ville IDREF #REQUIRED
>

<!ELEMENT menu EMPTY>
<!ATTLIST menu
    nom CDATA #REQUIRED
    prix CDATA #REQUIRED
>

<!ELEMENT ville (plusBeauMonument?)>

<!ATTLIST ville
    nom ID #REQUIRED
    departement CDATA #REQUIRED
>

<!ELEMENT plusBeauMonument EMPTY>
<!ATTLIST plusBeauMonument
    nom CDATA #REQUIRED
    tarif CDATA #REQUIRED
>
<!ELEMENT fermeture (#PCDATA)>

```

XS Schema

Un schéma XML est un document XML avec un élément racine **xs:schema** (xs designe l'espace de nom de XML Schema).

- **declaration des element**

- simple :
 - `<xs:element name="date" type="xs:string">`
 - default : `<xs:element name="couleur" type="xs:string" default="bleu"/>`
 - fixes : `<xs:element name="couleur" type="xs:string" fixed="bleu"/>`
- complexe : types complexes: `<sequence>`, `<choice>`, `<all>`
 - `<xs:sequence>` : séquence ordonnée d'éléments : (A,B,C)
 - `<xs:choice>` : choix parmi un ensemble d'éléments : (A|B|C)
 - `<xs:all>` : séquence non ordonnée

```

<xs:element name="ident">
  <xs:complexType>
    <xs:choice>
      <xs:element name="raisonsociale" type="xs:string"/>
      <xs:element name="identité" type="xs:string"/>
    </xs:choice>
  </xs:complexType>
</xs:element>

```

```

    </xs:choice>
  </xs:complexType>
</xs:element>

```

- **declaration des attributs:** -syntaxe `<xs:attribute name="langue" type="xs:string">`
toujours de type simple
 - contrainte :
 - Optionalité : use="optional" | "required" | "prohibited"
 - Valeur par défaut : default="string"
 - Valeur constante : fixed="10"
- **declaration des type simple par nous meme**
- par inclusion

```

<xs:simpleType name="T_etoile">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0" />
    <xs:maxInclusive value="3" />
  </xs:restriction>
</xs:simpleType>

```

- enumeration :

```

<xs:simpleType name="forme">
  <xs:restriction base="xs:string">
    <xs:enumeration value="cercle"/>
    <xs:enumeration value="triangle"/>
    <xs:enumeration value="carre"/>
  </xs:restriction>
</xs:simpleType>

```

- **declaration des type complexe par nous meme**

```

<xs:complexType name='T_Restaurant'>
  <xs:sequence>
    <xs:element name='fermeture' type='xs:string' minOccurs="0" />
    <xs:element name='menu' type='T_Menu' minOccurs="2"
maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name='nom' type='xs:string' use='required' />
  <xs:attribute name='etoile' type='T_etoile' use='required' />
  <xs:attribute name='ville' type='xs:string' use='required' />
</xs:complexType>

<xs:complexType name="T_Menu">

```

```

    <xs:attribute name="nom" type="xs:string" />
    <xs:attribute name="prix" type="xs:string" />
</xs:complexType>

```

- **declaration des clés**
- a mettre la ou on a ensemble des elements et non la ou on a qu'un seul element

```

<xs:key name="cleVille">
    <xs:selector xpath="ville" /> ici on met le chemin pour arriver
    <xs:field xpath="@nom" /> l'attribut pour avoir la cle
</xs:key>

```

```

<xs:keyref name="refVille" refer="cleVille">
    <xs:selector xpath="restaurant" />
    <xs:field xpath="ville" /> dans key ref pas la peine @
</xs:keyref>

```

exemple TME

```

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

<!DOCTYPE xs:schema SYSTEM "XMLSchema.dtd">

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <!-- à compléter :-->
    <xs:element name="base">
        <xs:complexType>
            <xs:choice maxOccurs="unbounded">
                <xs:element name="restaurant" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="fermeture" type="xs:string"
minOccurs="0" />
                            <xs:element name="menu" minOccurs="2"
maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:attribute name="nom"
type="xs:string" />
                                    <xs:attribute name="prix"
type="xs:integer" />
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                        <xs:attribute name="nom" type="xs:string"
use="required" />
                        <xs:attribute name="etoile" type="T-etoile"
use="required" />
                    </xs:complexType>
                </xs:element>
            </xs:choice>
        </xs:complexType>
    </xs:element>

```

```

        <xs:attribute name="ville" type="xs:string"
use="required" />
    </xs:complexType>

</xs:element>
<xs:element name="ville" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="plusBeauMonument" type="T-
plusBeauMonument" minOccurs="0" maxOccurs="1"></xs:element>
        </xs:sequence>
        <xs:attribute name="nom" type="xs:string"
use="required" />
        <xs:attribute name="departement" type="xs:integer"
use="required" />
    </xs:complexType>

</xs:element>
</xs:choice>
</xs:complexType>
<xs:key name="cleVille">
    <xs:selector xpath="ville" />
    <xs:field xpath="@nom" />
</xs:key>
<xs:keyref name="refVille" refer="cleVille">
    <xs:selector xpath="restaurant" />
    <xs:field xpath="ville" />
</xs:keyref>
</xs:element>
<xs:complexType name="T-plusBeauMonument">
    <xs:attribute name="nom" type="xs:string" use="required" />
    <xs:attribute name="tarif" type="xs:integer" use="required" />
</xs:complexType>
<xs:simpleType name="T-etoile">
    <xs:restriction base="xs:integer">
        <xs:minInclusive value="0"></xs:minInclusive>
        <xs:maxInclusive value="3"></xs:maxInclusive>
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

exemple 2

```

?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE xs:schema SYSTEM "XMLSchema.dtd">

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name='base'>
        <xs:complexType>
            <xs:choice minOccurs="1" maxOccurs="unbounded">

```

```

        <xs:element name='restaurant' type='T_Restaurant' />
        <xs:element name='ville' type='T_Ville' />
    </xs:choice>
</xs:complexType>
<xs:key name="villeKey">
    <xs:selector xpath="ville" />
    <xs:field xpath="@nom" />
</xs:key>
<xs:keyref name="villeRef" refer="villeKey">
    <xs:selector xpath='restaurant' />
    <xs:field xpath='@ville' />
</xs:keyref>
</xs:element>
<xs:complexType name='T_Restaurant'>
    <xs:sequence>
        <xs:element name='fermeture' type='xs:string' minOccurs="0" />
        <xs:element name='menu' type='T_Menu' minOccurs="2"
maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name='nom' type='xs:string' use='required' />
    <xs:attribute name='etoile' type='T_etoile' use='required' />
    <xs:attribute name='ville' type='xs:string' use='required' />
</xs:complexType>

<xs:simpleType name="T_etoile">
    <xs:restriction base="xs:integer">
        <xs:minInclusive value="0" />
        <xs:maxInclusive value="3" />
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="T_Menu">
    <xs:attribute name="nom" type="xs:string" />
    <xs:attribute name="prix" type="xs:string" />
</xs:complexType>
<xs:complexType name='T_Ville'>
    <xs:sequence minOccurs="0">
        <xs:element name='plusBeauMonument'>
            <xs:complexType>
                <xs:attribute name='nom' type='xs:string'
use="required" />
                <xs:attribute name='tarif' type='xs:integer'
use="required" />
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="nom" type='xs:string' use="required" />
    <xs:attribute name="departement" type="xs:integer" use="required"
/>
</xs:complexType>
</xs:schema>

```