Gestion des données XML

HEI 2019 / 2020

Introduction

- Après avoir modélisé nos données et construit une base de données à partir de cette modélisation, on peut réaliser un logiciel qui manipule et affiche ces données.

- Un autre gros besoin est de pouvoir échanger des données entre plusieurs logiciels de manière fiable.
  - Pour échanger des données, il va falloir mettre en place un langage commun d'échange.

XML

- XML (e**X**tensible **M**arkup **L**anguage) est un méta-langage permettant de définir des formats de document

- Il est caractérisé par son utilisation de balises comme HTML.

```
<?xml version="1.0" encoding="UTF-8"?>
<person>
    <name type="birth">Dupont</name>
    <name type="common">Martin</name>
    <firstname>Aurélie</firstname>
    <birthdate>2001-05-09</birthdate>
    <address>
        <street>1 rue de Paris</street>
        <zipcode>59000</zipcode>
        <city>Lille</city>
    </address>
</person>
```

Syntaxe

Généralités

- Le langage XML va permettre de créer des documents. Chaque document consiste en du texte, lisible par un être humain.

- Normalement, Unicode est le codage de caractère préconisé par XML. On peut donc représenter des informations pour pratiquement toutes les langues du monde.

Éléments

- Un document est composé d'un nombre variable d'élément. Chaque élément est représenté par une balise entre **<** et **>**.

<mon-élément>

- Il n'existe pas de liste d'éléments en XML, chaque application peut définir les siens.
- La seule limite est l'interdiction de certains caractères réservés :

!"#$%&'()*+,/;<=>?@[\]^`{|}~

Éléments

- Comme en HTML, les balises sont la plupart du temps présentes par paire : une balise ouvrante et une balise fermante.
  - La balise fermante commence par le caractère **/**.

```
<mon-élément>contenu</mon-élément>
```

- Les balises vides **doivent** être indiquée avec un caractère final **/**.

```
<mon-élément-vide />
```

Hiérarchie des éléments

- Les éléments du document XML peuvent être hiérarchisés. On représente donc un ensemble de données modélisé par un arbre.

```
<mon-élément>
    contenu
    <un-autre-élément>
        <mon-élément>autre contenu</mon-élément>
        <mon-élément>encore du contenu</mon-élément>
    </un-autre-élément>
    <un-élément-vide />
</mon-élément>
```

Racine du document

- Un document XML possèdent un nœud racine, **l'élément document**, qui contient tous les autres.

- Il ne peut y avoir qu'une et une seule racine à un document XML.

Attributs

- Un élément peut avoir un ou des attributs. On les renseigne au niveau de la balise ouvrante de l'élément.

```
<mon-élément un-attribut="sa valeur">
   contenu
</mon-élément>
```

- En XML, un attribut a toujours une valeur.
- On ne peut pas répéter plusieurs fois un attribut sur un même élément.

Commentaires

- Les commentaires s’écrivent comme en HTML.

```
<!-- Ceci est un commentaire -->
```

Prologue

- Un document XML doit commencer par une ligne de prologue :

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

Document bien formé

- Les règles de syntaxe du XML sont assez simples et permettent de représenter ce que l'ont veut.

- Si les contraintes de syntaxe sont respectées, un document est dit **bien formé**.

Sémantique

Du méta-langage au langage

- Maintenant que la syntaxe est connu, il va falloir définir un ensemble de règles pour permettre la communication entre deux systèmes informatiques :
  - Lister les éléments et leurs attributs possibles
  - Donner des contraintes dans l'organisation du document

Schémas

- Un **schéma** est la matérialisation informatique de l'ensemble de règles auxquelles doit se conformer le fichier XML.
- Un document XML qui respecte son schéma est dit **valide**.
- Il existe un grand nombre de normes pour déclarer des schémas XML. Différents logiciels vont utiliser différents types de schéma suivant leur implémentation.

Document Type Definition (DTD)

- Les DTD sont le type de schémas le plus ancien, hérité du SGML.

```
<!DOCTYPE person [
        <!ELEMENT person (name+, firstname, birthdate, address)>

        <!ELEMENT name (#PCDATA)>
        <!ATTLIST name TYPE CDATA #REQUIRED>

        <!ELEMENT firstname (#PCDATA)>
        <!ELEMENT birthdate (#PCDATA)>

        <!ELEMENT address (street, zipcode, city)>
        <!ELEMENT street (#PCDATA)>
        <!ELEMENT zipcode (#PCDATA)>
        <!ELEMENT city (#PCDATA)>
]>
```

Lien entre un document et sa DTD

- Comme en HTML, on peut lier un document XML avec sa DTD avec l'instruction **DOCTYPE**.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE person SYSTEM person.dtd>
<person >
    <name type="birth">Dupont</name>
    <name type="common">Martin</name>
(…)
```

Fichier contenant la DTD

W3C XML Schema (XSD)

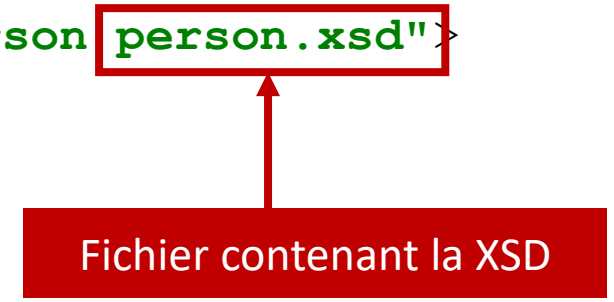
- Les XSD sont l'un des nombreux langages de schéma en XML. C'est l'un des plus répandu.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="person">
        <xs:complexType>
            <xs:choice>
                <xs:element name="name" type="name" maxOccurs="2"/>
                <xs:element name="firstname" type="xs:string"/>
                <xs:element name="birthdate" type="xs:string" minOccurs="0"/>
                <xs:element name="address" type="address" minOccurs="0"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>

    (…)
</xs:schema>
```

Lien entre un document et sa XSD

- L'attribut **schemaLocation** permet de lier un document XML avec son schéma.

```
<?xml version="1.0" encoding="UTF-8"?>
<person xmlns="http://donnees.chticod.eu/person"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://donnees.chticod.eu/person person.xsd">

    <name type="birth">Dupont</name>
    <name type="common">Martin</name>
(…)
```



Espace de nommage

- Dans les exemples précédents, on a pu voir l'utilisation de l'attribut **xmlns**. Cet attribut sert à introduire un espace de nommage (Name space).

- En assignant un ou des espaces de nommage à un document XML, il est possible d'utiliser les éléments de différents ensemble de règles (définis ou non par des schémas).

Attribut xmlns

- L'attribut xmlns permet de déclarer un espace de nommage. Il s'écrit de la façon suivante :

- Le préfixe permet d'avoir plusieurs espaces de nommage dans un même document.
- L'identifiant doit être sous la forme d'une URL mais c'est juste un identifiant, il peut n'y avoir rien derrière l'URL.

Document avec plusieurs espaces de nommage

- Un espace de nommage par défaut se déclare sans préciser de préfixe.
- Le préfixe est ensuite repris dans le nom des éléments des attributs pour les lier au bon espace de nommage.

Exemple

```xml
<?xml version="1.0" encoding="UTF-8"?>
<family xmlns="http://donnees.chticod.eu/family"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:p="http://donnees.chticod.eu/person" xsi:schemaLocation="http://donnees.chticod.eu/person person.xsd"
        xmlns:a="http://donnees.chticod.eu/animal">

    <p:person>
        <p:name p:type="birth">Dupont</p:name>
        <p:name p:type="common">Martin</p:name>
        <p:firstname>Aurélie</p:firstname>
        <p:birthdate>2001-05-09</p:birthdate>
    </p:person>

    <pets>
        <a:dog>
            <a:name>Médor</a:name>
            <a:race>Corgi</a:race>
        </a:dog>
        <a:cat>
            <a:name>Nala</a:name>
            <a:coat-type>tortoiseshell</a:coat-type>
        </a:cat>
        <a:cat>
            <a:name>Simba</a:name>
            <a:coat-type>bicolor</a:coat-type>
            <a:coat-color>black</a:coat-color>
            <a:coat-color>white</a:coat-color>
        </a:cat>
    </pets>
</family>
```

Exemples de langages XML

- XHTML : version XML de HTML
- RSS / ATOM : Syndication de contenu web
- XSL-T : Transformation de fichiers XML
- SOAP : Protocole d’échange pour les web services
- SVG : Image vectorielle
- Office Open XML : documents de la suite office
- Maven POM : configuration de projets java
- Thymeleaf : moteur de template java