



Données Semi Structurées

Chapitre 3: Galaxie XML Partie 1

S. Bella
Bella.salyma@gmail.com

L3 – Systèmes Informatiques

2021/2022

Plan

- 01 Les chemins : XPath
- 02 Les feuilles de style et traitement : XSLT
- 03 Applications XML : RDF, SVG
- 04 Les pointeurs : XPointer
- 05 Les liens : XLink

Les chemins : XPath

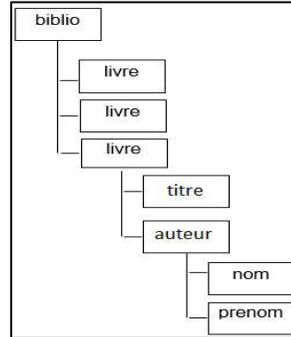
Langage de requête XPath

- XPath (**XML Path Language**) est utilisé pour **parcourir** les éléments et les attributs d'un document XML.
- Une **expression XPath** est une **requête** qui sert principalement à extraire telle ou telle partie de l'arborescence XML.
- XPath contient plus de 200 **fonctions intégrées**.
- Les expressions XPath peuvent également être utilisées en JavaScript, Java, schéma XML, PHP, Python, C, C++, ...
- XPath est un **élément majeur de norme XSLT**.

Chemin de localisation

- XPath utilise des expressions de chemin pour **identifier et localiser** des parties d'un document XML et **sélectionner** des nœuds.
- Les expressions de chemin ressemblent beaucoup aux expressions de chemin utilisées avec les systèmes de fichiers informatique traditionnelles.

```
<?xml version='1.0'?>
<biblio>
<livre lang='fr'>
<titre> Les Misérables </titre>
<auteur>
<nom> Hugo </nom>
<prenom> Victor </prenom>
</auteur>
</livre>
<livre lang='ar'>
.....
</livre>
<livre lang='en'>
....
</livre>
</biblio>
```



5

Chemin de localisation

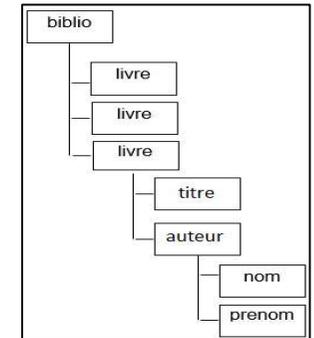
Il existe deux types de chemins :

- Chemin absolu**: Si le point de départ est la racine du document.

`/biblio/livre/auteur/nom`

- Chemin relatif**: Si le point de départ est un autre nœud différent de la racine du document.

`auteur/nom`



6

XPath: Nœud

- Les documents XML sont traités comme des arbres de nœuds. L'élément le plus haut de l'arbre est appelé l'élément racine.
- Dans XPath, il existe 7 types de nœuds : **éléments**, **attributs**, **texte**, **espace de nom**, **instructions de traitement**, **commentaires** et **document**.

```
<?xml version="1.0"?>
<biblio xmlns="http://www.mabiblio.com">
<!-- Livre 1-->
<livre lang='fr'>
<titre> Les Misérables </titre>
<auteur>
<nom> Hugo </nom>
<prenom> Victor </prenom>
</auteur>
</livre>
</biblio>
```

7

XPath: Nœud

Il existe 5 types de relation entre les nœuds dans XPath :

- PARENT** : chaque élément et attribut a un parent.
- CHILDREN** : les nœuds d'éléments peuvent avoir zéro, un ou plusieurs enfants.
- SIBLINGS** : les nœuds ayant le même parent.
- ANCESTORS**: ou ancêtres en français, sont le parent d'un nœud, parent d'un parent, etc.
- DESCENDANTS** (descendance) : les enfants d'un nœud, les enfants des enfants,... etc.

```
<?xml version="1.0"?>
<biblio>
<livre lang='fr'>
<titre> Les Misérables </titre>
<auteur>
<nom> Hugo </nom>
<prenom> Victor </prenom>
</auteur>
</livre>
</biblio>
```

8

XPath: Désignation (Syntaxe)

- XPath utilise des expressions de chemin pour **sélectionner** ou **désigner** des nœuds ou des ensembles de nœuds dans un document XML.
- La désignation peut se faire de plusieurs manières par :
 - ✓ La sélection des nœuds (par le chemin);
 - ✓ La propriété (prédicat le qualifiant);
 - ✓ La sélection des nœuds inconnus;
 - ✓ La sélection de plusieurs chemins.

9

XPath: Sélection des nœuds

Le nœud est sélectionné en suivant **un chemin** ou des **étapes**. Les expressions de chemin les plus utilisées sont représentées dans le tableau suivant :

Expression XPath	Description
node_name	Sélectionner tous les nœuds avec le nom «node_name».
/	Sélectionner à partir du nœud racine.
//	Sélectionne les nœuds du document à partir du nœud actuel qui correspondent à la sélection, peu importe où ils se trouvent.
.	Sélectionner le nœud actuel.
..	Sélectionner le parent du nœud actuel.
@	Sélectionner les attributs.

10

XPath: Sélection des nœuds

Doc XML	Expression XPath	Description (Résultat)
<pre><?xml version="1.0" encoding="iso-8859-1" ?> <etudiants> <etudiant num_inscription="102154" > <note module="1" > 12.5 </note> <note module="2" > 8 </note> <nom> mustapha </nom> <prenom> mohamed </prenom> <groupe> 5 </groupe> </etudiant> <etudiant num_inscription="102168" > <note module="1" > 19 </note> <note module="2" > 15 </note> <nom> omar </nom> <prenom> ali </prenom> <groupe> 3 </groupe> </etudiant> </etudiants></pre>	/etudiants/etudiant	Sélectionner tous les éléments etudiant qui sont des enfants de l'élément racine etudiants .
	//etudiant	Sélectionner tous les éléments etudiant peu importe où ils se trouvent dans le document.
	//@module	Sélectionner tous les attributs nommé module .

11

XPath: Désignation par le prédicat

- Les prédicats sont utilisés pour **rechercher un nœud spécifique** ou un nœud contenant une **valeur spécifique** (exprimer une condition). Les prédicats sont toujours intégrés entre **crochets [..]**

Expression XPath	Description (Résultat)
/etudiants/etudiant[1]	Sélectionner le premier élément etudiant qui est l'enfant de l'élément etudiants .
/etudiants/etudiant[last()]	Sélectionner le dernier élément etudiant qui est l'enfant de l'élément racine etudiants .
/etudiants/etudiant[last()-1]	Sélectionner l'avant dernier élément etudiant qui est l'enfant de l'élément racine etudiants .
/etudiants/etudiant[position()<10]	Sélectionner les 9 premiers éléments etudiant qui sont les enfants de l'élément racine etudiants .

12

XPath: Désignation par le prédicat

Expression XPath	Description (Résultat)
<code>//etudiant[@num_inscription]</code>	Sélectionner tous les éléments etudiant qui ont un attribut nommé num_inscription .
<code>//etudiant[@num_inscription="12354"]</code>	Sélectionner tous les éléments etudiant qui ont un attribut nommé num_inscription avec la valeur 12354.
<code>//etudiant[note>18]</code>	Sélectionner tous les éléments etudiant qui ont une note supérieure à 18.
<code>//etudiant[note>12]/nom</code>	Sélectionner tous les éléments nom des éléments etudiant qui ont un élément note avec une valeur supérieure à 12.

13

XPath: Sélection des nœuds inconnus

- Les **caractères génériques** XPath peuvent être utilisés pour sélectionner des nœuds XML inconnus.

Caractères génériques XPath	Description (Résultat)
<code>*</code>	Correspond à n'importe quel nœud d'élément.
<code>@*</code>	Correspond à n'importe quel nœud d'attributs.
<code>node()</code>	Correspond à n'importe quel nœud de toute sorte.

Expression XPath	Description (Résultat)
<code>//etudiant/*</code>	Sélectionner tous les éléments fils d'élément etudiant .
<code>//note[@*]</code>	Sélectionner tous les éléments note qui ont au moins un attribut de toute nature.

14

XPath: Sélection de plusieurs chemins

- Le caractère « `|` » est utilisé dans une expression XPath pour **sélectionner plusieurs chemins**.
- Exemple:

Expression XPath	Description (Résultat)
<code>//note //nom</code>	Sélectionner tous les éléments note et nom dans le document XML.

15

XPath: Opérateurs

- Les **opérateurs** qui peuvent être utilisées dans les expressions XPath sont les suivantes :

Type	Opérateurs
Booléen	and, or
Logique	!=, >=, >, <, <=
Opérations	+, -, *, div, mod

Expression XPath	Description
<code>//note[@module != "1"]</code>	Sélectionner tous les éléments note qui ont un attribut module ayant une valeur différente de 1.
<code>//etudiant[nom and @num_inscription]</code>	Sélectionner tous les éléments etudiant qui ont un élément enfant nom et un attribut num_inscription .

16

XPath: Exemples

....

Merci pour votre Attention

Vos Questions !!

....

XPath: Exemple

Expression XPath	Description	Exemple	Résultat (tous)	Résultat (data)
/AAA	Sélectionner l'élément racine AAA	<pre><AAA> <BBB/> <CCC/> <BBB/> <DDD> <BBB/> </DDD> <CCC/> </AAA></pre>	<ol style="list-style-type: none"> <AAA> <BBB/> <CCC/> <BBB/> <DDD> <BBB/> </DDD> <CCC/> </AAA> 	<ol style="list-style-type: none"> (vide)
/AAA/CCC	Sélectionner tous les éléments CCC qui sont enfants de l'élément racine AAA	<pre><AAA> <BBB/> <CCC> salam </CCC> <BBB/> <DDD> <BBB/> </DDD> <CCC/> </AAA></pre>	<ol style="list-style-type: none"> <CCC> salam </CCC> <CCC/> 	<ol style="list-style-type: none"> salam (vide)
/AAA/DDD/BBB	Sélectionne tous les éléments BBB qui sont enfants des éléments DDD , qui sont enfants de l'élément racine AAA	<pre><AAA> <BBB/> <CCC/> <BBB/> <DDD> <BBB/> </DDD> <CCC/> </AAA></pre>	<ol style="list-style-type: none"> <BBB/> 	<ol style="list-style-type: none"> (vide)
/AAA/BBB[1] ou /AAA/BBB[position()=1]	Sélectionner le premier élément BBB , fils de l'élément racine AAA	<pre><AAA> <BBB/> <BBB/> <BBB/> </AAA></pre>	<ol style="list-style-type: none"> <BBB/> 	<ol style="list-style-type: none"> (vide)
/AAA/BBB[last()]	Sélectionner le dernier élément BBB , fils de l'élément racine AAA	<pre><AAA> <BBB/> <BBB/> <BBB/> </AAA></pre>	<ol style="list-style-type: none"> <BBB/> 	<ol style="list-style-type: none"> (vide)
//@id	Sélectionner tous les attributs id	<pre><AAA> <BBB id="b1"/> <BBB id="b2"/> <BBB name="bbb"/> <BBB/> </AAA></pre>	<ol style="list-style-type: none"> <BBB id="b1"/> <BBB id="b2"/> 	<ol style="list-style-type: none"> b1 b2
//BBB[@id]	Sélectionner tous les éléments BBB	<pre><AAA> <BBB id="b1"/> <BBB id="b2"/></pre>	<ol style="list-style-type: none"> <BBB id="b1"/> 	<ol style="list-style-type: none"> (vide)

XPath: Exemple

	qui ont un attribut id	Salam </BBB> <BBB name="bbb"/> <BBB/> </AAA>	2. <BBB id="b2"> Salam </BBB>	2. Salam
//BBB[@*]	Sélectionner tous les éléments BBB qui ont un attribut quelconque	<AAA> <BBB id="b1"/> <BBB id="b2"/> <BBB name="bbb" /> <BBB/> </AAA>	1. <BBB id="b1"/> 2. <BBB id="b2"/> 3. <BBB name="bbb"/>	1. (vide) 2. (vide) 3. (vide)
//BBB[@name]	Sélectionner tous les éléments BBB qui ont un attribut name	<AAA> <BBB id="b1"/> <BBB id="b2"/> <BBB name="bbb"/> <BBB/> </AAA>	1. <BBB name="bbb"/>	1. (vide)
//BBB[@name='bbb']	Sélectionner tous les éléments BBB ayant un attribut name dont la valeur est bbb	<AAA> <BBB id="b1"/> <BBB name="bbb"/> <BBB name="bbb"/> <BBB/> </AAA>	1. <BBB name="bbb"/>	1. (vide)
//BBB[normalize-space(@name)='bbb']	Sélectionner tous les éléments BBB ayant un attribut name dont la valeur est bbb . Les espaces de début et fin sont supprimés avant la comparaison.	<AAA> <BBB id="b1"/> <BBB name="bbb"/> <BBB name="bbb"/> <BBB/> </AAA>	1. <BBB name="bbb"/> 2. <BBB name="bbb"/>	1. (vide) 2. (vide)
//BBB[not(@*)]	Sélectionner tous les éléments BBB qui n'ont pas d'attributs	<AAA> <BBB id="b1"/> <BBB id="b2"/> <BBB name="bbb"/> <BBB/> </AAA>	1. <BBB/>	1. (vide)
//*[count(BBB)=2]	Sélectionner tous les éléments ayant deux enfants BBB	<AAA> <CCC> <BBB/> <BBB/> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> <EEE>	1. <DDD> <BBB/> <BBB/> </DDD>	1 (vide)

XPath: Exemple

		<pre> <CCC/> <DDD/> </EEE> </AAA> </pre>		
<pre> //*[count(*)=3] </pre>	<p>Sélectionner tous les éléments ayant trois enfants</p>	<pre> <AAA> <CCC> <BBB/> <BBB/> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> <EEE> <CCC/> <DDD/> </EEE> </AAA> </pre>	<ol style="list-style-type: none"> 1. <AAA> <CCC> <BBB/> <BBB/> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> <EEE> <CCC/> <DDD/> </EEE> </AAA> 2. <CCC> <BBB/> <BBB/> <BBB/> </CCC> 	<ol style="list-style-type: none"> 1. (vide) 2. (vide)
<pre> //*[name()='BBB'] </pre>	<p>Sélectionner tous les éléments nommés BBB</p>	<pre> <AAA> <BCC> <BBB/> <BBB/> <BBB/> </BCC> <DDB> <BBB/> <BBB/> </DDB> <BEC> <CCC/> <DBD/> </BEC> </AAA> </pre>	<ol style="list-style-type: none"> 1. <BBB/> 2. <BBB/> 3. <BBB/> 4. <BBB/> 5. <BBB/> 	(vide)
<pre> //*[starts-with(name(),'B')] </pre>	<p>Sélectionner tous les éléments dont leur nom commence par B</p>	<pre> <AAA> <BCC> <BBB/> <BBB/> <BBB/> </BCC> <DDB> <BBB/> <BBB/> </pre>	<ol style="list-style-type: none"> 1. <BCC> <BBB/> <BBB/> <BBB/> </BCC> 2. <BBB/> 3. <BBB/> 4. <BBB/> 5. <BBB/> 	

XPath: Exemple

		<pre> </DDB> <BEC> <CCC/> <DBD/> </BEC> </AAA> </pre>	<pre> 6. <BBB/> 7. <BEC> <CCC/> <DBD/> </BEC> </pre>	
//*[contains(name(),'C')]	Sélectionner tous les éléments dont leur nom contient la lettre C	<pre> <AAA> <BCC> <BBB/> <BBB/> <BBB/> </BCC> <DDB> <BBB/> <BBB/> </DDB> <BEC> <CCC/> <DBD/> </BEC> </AAA> </pre>	<pre> 1. <BCC> <BBB/> <BBB/> <BBB/> </BCC> 2. <BEC> <CCC/> <DBD/> </BEC> 3. <CCC/> </pre>	
//*[string-length(name())=3]	Sélectionner tous les éléments dont leur longueur de nom est 3	<pre> <AAA> <Q/> <BB/> <CCC/> <DDDDDDDD/> <EEEE/> </AAA> </pre>	<pre> 1. <AAA> <Q/> <BB/> <CCC/> <DDDDDDDD/> <EEEE/> </AAA> 2. <CCC/> </pre>	
//*[string-length(name())<4]	Sélectionner tous les éléments dont leur longueur de nom < 4	<pre> <AAA> <Q/> <BB/> <CCC/> <DDDDDDDD/> <EEEE/> </AAA> </pre>	<pre> 1. <AAA> <Q/> <BB/> <CCC/> <DDDDDDDD/> > <EEEE/> </AAA> 2. <Q/> 3. <BB/> 4. <CCC/> </pre>	
/AAA/EEE //BBB	Sélectionner tous les éléments EEE qui sont enfants de l'élément racine AAA et tous les éléments BBB	<pre> <AAA> <CCC> <BBB/> <BBB/> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> </pre>	<pre> 1. <BBB/> 2. <BBB/> 3. <BBB/> 4. <BBB/> 5. <BBB/> 6. <EEE> <CCC/> <DDD/> </EEE> </pre>	

XPath: Exemple

		<pre> <EEE> <CCC/> <DDD/> </EEE> </AAA> </pre>		
//DDD/parent::*	Sélectionner tous les parents de DDD	<pre> <AAA> <CCC> <BBB/> <BBB/> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> <EEE> <CCC/> <DDD/> </EEE> </AAA> </pre>	<ol style="list-style-type: none"> <pre> <AAA> <CCC> <BBB/> <BBB/> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> <EEE> <CCC/> <DDD/> </EEE> </AAA> </pre> <pre> <EEE> <CCC/> <DDD/> </EEE> </pre> 	
/AAA/CCC/BBB/BEC/ancestor::*	Sélectionner tous les ancêtres des éléments BEC selon ce chemin absolu.	<pre> <AAA> <CCC> <BBB/> <BBB> <BEC/> </BBB> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> <EEE> <CCC/> <DDD/> </EEE> </AAA> </pre>	<ol style="list-style-type: none"> <pre> <AAA> <CCC> <BBB/> <BBB> <BEC/> </BBB> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> <EEE> <CCC/> <DDD/> </EEE> </AAA> </pre> <pre> <CCC> <BBB/> <BBB> <BEC/> </BBB> <BBB/> </CCC> </pre> 	

XPath: Exemple

			3. <BBB> <BEC/> </BBB>	
//BBB/ancestor::*	Sélectionner tous les ancêtres des éléments BBB	<pre> <AAA> <CCC> <BBB/> <BBB> <BEC/> </BBB> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> <EEE> <CCC/> <DDD/> </EEE> </AAA> </pre>	1. <AAA> <CCC> <BBB/> <BBB> <BEC/> </BBB> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> <EEE> <CCC/> <DDD/> </EEE> </AAA> 2. <CCC> <BBB/> <BBB> <BEC/> </BBB> <BBB/> </CCC> 3. <DDD> <BBB/> <BBB/> </DDD>	
//BBB/descendant::*	Sélectionner tous les descendant des éléments BBB	<pre> <AAA> <CCC> <BBB/> <BBB> <BEC> <hhh/> </BEC> </BBB> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> </AAA> </pre>	1. <BEC> <hhh/> </BEC> 2. <hhh/>	

XPath: Exemple

<p>//BBB/child::*</p>	<p>Sélectionner tous les enfants des éléments BBB</p>	<pre><AAA> <CCC> <BBB/> <BBB> <BEC> <hhh/> </BEC> </BBB> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> </AAA></pre>	<p>1. <BEC> <hhh/> </BEC></p>	
<p>//CCC/following-sibling::*</p>	<p>Sélectionner les éléments frères et sœurs après l'élément CCC</p>	<pre><AAA> <CCC> <BBB/> <BBB> <BEC> <hhh/> </BEC> </BBB> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> </AAA></pre>	<p>1. <DDD> <BBB/> <BBB/> </DDD></p>	
<p>//BBB[3]/preceding-sibling::*</p>	<p>Sélectionner les éléments frères et sœurs avant de troisième élément BBB.</p>	<pre><AAA> <CCC> <BBB/> <BBB> <BEC> <hhh/> </BEC> </BBB> <BBB/> </CCC> <DDD> <BBB/> <BBB/> </DDD> </AAA></pre>	<p>1. <BBB/> 2. <BBB> <BEC> <hhh/> </BEC> </BBB></p>	