

Notes XML

Table des matières

Notes XML	1
Les espaces de noms	2
Espace de nom XML	2
Espaces de noms communément utilisés	2
Les schémas XML (XSD)	3
Les types simples	3
Les chaînes de caractères	3
Plus en détails	4
Les dates.....	6
Plus en détails	6
Numériques.....	8
Plus en détails	8
Autres types	10
Plus en détails	11
Héritage	12
Héritage par restriction	12
La norme XPath.....	14
Nœuds d'un document.....	14
Introduction aux services web	16
Présentation des services web	16
Services web et SOA	17
Avantages	17



Les normes	17
SOAP	17
WDSL	17
UDDI.....	18
XML-RPC	18
Plateformes à services web	18
SOAP : principes généraux.....	18
Structure d'un message SOAP.....	20
Une réponse contenant une SOAP fault.....	20

Les espaces de noms

Dublin Core

<https://www.dublincore.org/specifications/dublin-core/dcmi-terms/>

DocBook

<https://tdg.docbook.org/tdg/5.1/> .

Xinclude

<https://www.w3.org/TR/xinclude/>

Espace de nom XML

Le préfixe xml est toujours implicitement lié à l'espace de noms XML identifié par l'URI <http://www.w3.org/XML/1998/namespace> . Cet espace de noms n'a pas besoin d'être déclaré. Les quatre attributs particuliers xml:lang, xml:space, xml:base et xml:id font partie de cet espace de noms.

Ces quatre attributs sont déclarés par le schéma XML qui se trouve à l'URL <http://www.w3.org/2001/xml.xsd> . Ce schéma peut être importé par un autre schéma pour ajouter certains de ces attributs à des éléments.

Espaces de noms communément utilisés

XML



DAWAN – REPRODUCTION INTERDITE

<http://www.w3.org/XML/1998/namespace>
 XInclude
<http://www.w3.org/2001/XInclude>
 XLink
<http://www.w3.org/1999/xlink>
 MathML
<http://www.w3.org/1998/Math/MathML>
 XHTML
<http://www.w3.org/1999/xhtml>
 SVG
<http://www.w3.org/2000/svg>
 Schémas
<http://www.w3.org/2001/XMLSchema>
 Instances de schémas
<http://www.w3.org/2001/XMLSchema-instance>
 Schematron
<http://purl.oclc.org/dsdl/schematron>
 XSLT
<http://www.w3.org/1999/XSL/Transform>
 XSL-FO
<http://www.w3.org/1999/XSL/Format>
 DocBook
<http://docbook.org/ns/docbook>
 Dublin Core
<http://purl.org/dc/elements/1.1/>

Les schémas XML (XSD)

Les types simples

Les chaînes de caractères

Type	Description	Commentaire
string	représente une chaîne de caractères	attention aux caractères spéciaux
normalizedString	représente une chaîne de caractères normalisée	basé sur le type string
token	représente une chaîne de caractères normalisée sans espace au début et à la fin	basé sur le type normalizedString
language	représente le code d'une langue	basé sur le type token
NMTOKEN	représente une chaîne de caractères "simple"	basé sur le type token applicable uniquement aux attributs
NMTOKENS	représente une liste de NMTOKEN	applicable uniquement aux attributs
Name	représente un nom XML	basé sur le type token

Type	Description	Commentaire
NCName	représente un nom XML sans le caractère :	basé sur le type Name
ID	représente un identifiant unique	basé sur le type NCName applicable uniquement aux attributs
IDREF	référence à un identifiant	basé sur le type NCName applicable uniquement aux attributs
IDREFS	référence une liste d'identifiants	applicable uniquement aux attributs
ENTITY	représente une entité d'un document DTD	basé sur le type NCName applicable uniquement aux attributs
ENTITIES	représente une liste d'entités	applicable uniquement aux attributs

Plus en détails

Le type string

Il représente une chaîne de caractères et peut donc contenir un peu tout et n'importe quoi. Il est cependant important de noter que certains caractères spéciaux comme le & doivent être écrits avec leur notation HTML.

Le type normalizedString

Basé sur le type string, représente une chaîne de caractères normalisée, c'est-à-dire, une chaîne de caractères qui peut contenir tout et n'importe quoi à l'exception de tabulations, de sauts de ligne et de retours chariot.

Puisque le type normalizedString est basé sur le type string, toutes les règles du type string s'appliquent également au type normalizedString. Ainsi, les caractères spéciaux comme le & doivent être écrits avec leur notation HTML.

Un type hérite toujours de toutes les règles du type sur lequel il se base.

Le type token

Basé sur le type normalizedString et représente une chaîne de caractères normalisée sans espace au début ni à la fin.

Le type language

Basé sur le type token et représente une langue.

Identifiée par 2 lettres (norme ISO 639).



DAWAN – REPRODUCTION INTERDITE

Les 2 caractères peuvent éventuellement être suivi d'un code pays (ISO 3166).

Le type NMTOKEN

Basé sur le type token et représente une chaîne de caractères "simple", chaîne sans espace qui ne contient que les symboles suivants :

- Des lettres.
- Des chiffres.
- Les caractères spéciaux .-_ et :

Pour une compatibilité entre les Schémas XML et les DTD, n'utiliser le type NMTOKEN que pour un attribut.

Le type Name

Basé sur le type token et représente un nom XML, une chaîne de caractères sans espace qui ne contient que les symboles suivants :

- Des lettres.
- Des chiffres.
- Les caractères spéciaux .-_ et :

Différent du type NMTOKEN car une chaîne de caractères de type Name commence obligatoirement par une lettre, ou l'un des 2 caractères spéciaux suivants : _ et :

Le type NCName

Basé sur le type Name. hérite des règles du type Name plus une nouvelle règle : le type NCName ne peut pas contenir le caractère spécial :

Le type ID

Basé sur le type NCName, hérite des règles de ce type. Représente un identifiant. Contient des valeurs uniques. Impossible de définir une valeur fixe ou par défaut.

Un ID ne doit être utilisé qu'avec les attributs pour assurer une compatibilité entre les Schémas XML et les DTD.

Le type IDREF

Fait référence à un ID existant dans le document XML. Basé sur le type NCName et hérite des règles de ce type. IDREF n'est utilisable qu'avec des attributs.

Le type IDREFS

NMTOKENS représente une liste de NMTOKEN séparés par un espace, IDREFS représente lui une liste de IDREF séparés par un espace.



DAWAN – REPRODUCTION INTERDITE

N'utiliser le type IDREFS que pour un attribut pour une compatibilité entre les Schémas XML et les DTD.

Le type ENTITY

Permet de faire référence à une entité le plus souvent non XML et déclaré dans des fichiers DTD. Basé sur le type NCName, hérite de toutes ses règles.

Pour une compatibilité entre les Schémas XML et les DTD, n'utiliser le type ENTITY que pour un attribut.

Le type ENTITIES

Fait référence à une liste d'ENTITY séparés par un espace.

ENTITIES ne doit être utilisé qu'avec un attribut.

Les dates

Type	Description
duration	représente une durée
date	représente une date
time	représente une heure
dateTime	représente une date et un temps
gYear	représente une année
gYearMonth	représente une année et un mois
gMonth	représente un mois
gMonthDay	représente un mois et un jour
gDay	représente un jour

Plus en détails

Le type duration

Représente une durée. Exprimé en nombre d'années, de mois, de jours, d'heures, de minutes et de secondes selon l'expression suivante : PnYnMnDTnHnMnS.

- P : début de l'expression.
- nY : nombre d'années (year), où n est un nombre entier.
- nM : nombre de mois (month) où n est un nombre entier.
- nD : nombre de jours (day) où n est un nombre entier.
- T : pour séparer la partie date de la partie heure.
- nH : nombre d'heures (hour) où n est un nombre entier.
- nM : nombre de minutes (minute) où n est un nombre entier.
- nS : nombre de secondes (second) où n est un nombre entier ou décimal.



DAWAN – REPRODUCTION INTERDITE

L'expression peut être précédée du signe - dans le cas où l'on souhaite exprimer une durée négative. Tous les champs ne doivent pas être renseignés. Possible de ne renseigner que les heures, les minutes, etc. Dans le cas d'une date, le symbole T ne doit plus figurer.

Le type date

Exprime une date. Selon une expression spécifique : YYYY-MM-DD.

- YYYY : année (year) sur 4 chiffres ou plus.
- MM : mois (month) sur 2 chiffres.
- DD : jour (day) également sur 2 chiffres.

Dans le cas où l'on souhaite exprimer une date avant Jésus-Christ, un signe - peut-être placé devant l'expression.

Le type time

Exprime une heure. Selon une expression spécifique : hh:mm:ss.

- hh : heures (hour) sur 2 chiffres.
- mm : minutes (minute) sur 2 chiffres.
- ss : secondes (second) sur 2 chiffres entiers ou à virgule.

Le type dateTime

Mélange entre le type date et le type time. Représente une date ET une heure. Selon une expression particulière : YYYY-MM-DDThh:mm:ss.

Le type gYear

Représente une année sur 4 chiffres ou plus. Dans le cas où l'on souhaite exprimer une année avant Jésus-Christ, un signe - peut-être placé devant l'expression.

Le type gYearMonth

Représente une année et un mois. Selon l'expression : YYYY-MM.

Le type gMonth

Représente un mois sur 2 chiffres précédés du symbole --.

Le type gMonthDay

Représente un mois et un jour. Selon l'expression : --MM--DD.

Le type gDay

Représente un jour sur 2 chiffres précédés du symbole --.



DAWAN – REPRODUCTION INTERDITE

Numériques.

Type	Description	Commentaire
float	représente un nombre flottant sur 32 bits conforme à la norme IEEE 754	
double	représente un nombre flottant sur 64 bits conforme à la norme IEEE 754	
decimal	représente un nombre décimal	
integer	représente un nombre entier	basé sur le type decimal
long	représente un nombre entier	basé sur le type integer
int	représente un nombre entier	basé sur le type long
short	représente un nombre entier	basé sur le type int
byte	représente un nombre entier	basé sur le type short
nonPositiveInteger	représente un nombre entier non positif	basé sur le type integer
negativeInteger	représente un nombre entier négatif	basé sur le type nonPositiveInteger
nonNegativeInteger	représente un nombre entier non négatif	basé sur le type integer
positiveInteger	représente un nombre entier positif	basé sur le type nonNegativeInteger
unsignedLong	représente un nombre entier positif	basé sur le type nonNegativeInteger
unsignedInt	représente un nombre entier positif	basé sur le type unsignedLong
unsignedShort	représente un nombre entier positif	basé sur le type unsignedInt
unsignedByte	représente un nombre entier positif	basé sur le type unsignedShort

Plus en détails

Le type float

Représente un nombre flottant sur 32 bits, un nombre entier ou décimal, se trouvant entre les valeurs 3.4×10^{-38} et 3.4×10^{38} .

A cette plage de valeurs, 3 autres peuvent être ajoutées :

- -INF pour moins l'infini.
- +INF pour plus l'infini.
- NaN pour Not a Number, c'est-à-dire pour désigner une valeur non numérique.

Un type float avec des exposants. notation E ou e.



DAWAN – REPRODUCTION INTERDITE

Le type double

Représente un nombre flottant sur 64 bits, un entier ou décimal entre les valeurs 1.7×10^{-308} et 1.7×10^{308} .

Comme pour le type float, les 3 valeurs suivantes peuvent être ajoutées à la liste :

- -INF pour moins l'infini.
- +INF pour plus l'infini.
- NaN pour Not a Number, c'est-à-dire pour désigner une valeur non numérique.

L'exposant s'exprime comme pour float

Le type decimal

Représente un nombre décimal, c'est-à-dire un nombre qui peut-être entier ou à virgule. Ce nombre peut-être positif ou négatif et donc être précédé du symbole + ou -. Dans le cas d'un nombre où la partie entière est égale à zéro, il n'est pas obligatoire de l'écrire.

Le type integer

Basé sur le type decimal et représente un nombre entier, c'est-à-dire un nombre sans virgule. Comme pour le type décimal, un nombre de type integer peut être précédé par le symbole + ou -.

Le type long

Basé sur le type integer mais forcément être compris entre les valeurs -9 223 372 036 854 775 808 et 9 223 372 036 854 775 807.

Le type int

Basé sur le type long mais forcément être compris entre les valeurs -2 147 483 648 et 2 147 483 647.

Le type short

Basé sur le type int mais forcément être compris entre les valeurs -32 768 et 32 767.

Le type byte

Basé sur le type short mais forcément être compris entre les valeurs -128 et 127.

Le type nonPositiveInteger



DAWAN – REPRODUCTION INTERDITE

Basé sur le type `integer`, le type `nonPositiveInteger` représente un nombre entier qui n'est pas positif. Concrètement, cela correspond à un nombre négatif ou au nombre zéro.

Le type `negativeInteger`

Basé sur le type `nonPositiveInteger`, le type `negativeInteger` représente un nombre entier strictement négatif, c'est-à-dire strictement inférieur à zéro.

Le type `nonNegativeInteger`

Basé sur le type `integer`, le type `nonNegativeInteger` représente un nombre entier qui n'est pas négatif, c'est-à-dire un nombre supérieur ou égal à zéro.

Le type `positiveInteger`

Basé sur le type `nonNegativeInteger`, le type `positiveInteger` représente un nombre entier strictement positif, c'est-à-dire strictement supérieur à zéro.

Le type `unsignedLong`

Basé sur le type `nonNegativeInteger` et représente un entier compris entre les valeurs 0 et 18 446 744 073 709 551 615.

Le type `unsignedInt`

Basé sur le type `unsignedLong` et représente un entier compris entre les valeurs 0 et 4 294 967 295.

Le type `unsignedShort`

Basé sur le type `unsignedInt` et représente un entier compris entre les valeurs 0 et 65 535.

Le type `unsignedByte`

Basé sur le type `unsignedShort` et représente un entier compris entre les valeurs 0 et 255.

Autres types

Type	Description
<code>boolean</code>	représente l'état vrai ou faux
<code>QName</code>	représente un nom qualifié
<code>NOTATION</code>	représente une notation
<code>anyURI</code>	représente une URI
<code>base64Binary</code>	représente une donnée binaire au format Base64



DAWAN – REPRODUCTION INTERDITE

Type	Description
hexBinary	représente une donnée binaire au format hexadecimal

Plus en détails

Le type boolean

Représente un booléen. Vrai ou faux

2 états possibles, 4 valeurs acceptées :

- true qui représente l'état vrai.
- false qui représente l'état faux.
- 1 qui représente l'état vrai.
- 0 qui représente l'état faux.

Le type QName

Représente un nom qualifié (espaces de nom).

Le type NOTATION

Identifie et décrit du contenu XML ou non, par exemple une image.

Le type anyURI

Représente une URI (Uniform Resource Identifier).

L'URI est une chaîne de caractères qui identifie une ressource.

On distingue généralement 2 types d'URI :

- Les URL (Uniform Resource Locator) est probablement la forme d'URI la plus connue et je suis sûr que ce nom vous parle. En effet, les URL sont utilisées pour décrire l'adresse d'une ressource sur un réseau.
- Les URN (Uniform Resource Name) sont quant à eux utilisés pour identifier une ressource dans un espace de noms. Je ne vais pas m'attarder sur les URN car nous reviendrons plus tard dans ce cours sur la notion des espaces de nom.

Une URI permet d'identifier une ressource de manière relative ou absolue.

Le type base64Binary

Représente une donnée binaire au format Base64.

Plusieurs règles :



DAWAN – REPRODUCTION INTERDITE

- Seules les lettres (majuscules ou minuscules), les chiffres et les symboles + / et = sont autorisés.
- Le nombre de caractères qui composent la chaîne doit être un multiple de 4.

Dans le cas où le symbole = est utilisé, de nouvelles règles doivent être respectées :

- Il ne peut apparaître qu'en fin de chaîne, une fois ou deux.
- Dans le cas où il est utilisé qu'une seule fois, il doit forcément être précédé des caractères A Q g ou w.
- Dans le cas où il est utilisé 2 fois, il doit forcément être précédé des caractères A E I M Q U Y c g k o s w 0 (zéro) 4 ou 8.

Le type hexBinary

Représente une donnée binaire au format hexadécimal.

Seuls les lettres entre A et F (majuscules ou minuscules), ainsi que les chiffres sont autorisés.

Héritage

Héritage par restriction

Type_de_base

Type	Description
minExclusive	permet de définir une valeur minimale exclusive
minInclusive	permet de définir une valeur minimale inclusive
maxExclusive	permet de définir une valeur maximale exclusive
maxInclusive	permet de définir une valeur maximale inclusive
totalDigits	permet de définir le nombre exact de chiffres qui composent un nombre
fractionDigits	permet de définir le nombre de chiffres autorisés après la virgule
length	permet de définir le nombre exact de caractères d'une chaîne
minLength	permet de définir le nombre minimum de caractères d'une chaîne
maxLength	permet de définir le nombre maximum de caractères d'une chaîne
enumeration	permet d'énumérer la liste des valeurs possibles
whiteSpace	permet de déterminer le comportement à adopter avec les espaces
pattern	permet de définir des expressions rationnelles

La restriction minExclusive

S'applique à un élément de type numérique et permet de définir sa valeur minimale. La valeur indiquée est exclue des valeurs que peut prendre l'élément.



DAWAN – REPRODUCTION INTERDITE

La restriction minInclusive

Comme **minExclusive** mais la valeur indiquée peut-être prise par l'élément.

La restriction maxExclusive

S'applique à un élément de type numérique et permet de définir sa valeur maximale. La valeur indiquée est exclue des valeurs que peut prendre l'élément.

La restriction maxInclusive

Comme **maxExclusive** mais la valeur indiquée peut-être prise par l'élément.

La restriction totalDigits

S'applique à un élément de type numérique et définit le nombre exact de chiffres qui composent le nombre. Obligatoirement supérieure à zéro.

La restriction fractionDigits

S'applique à un élément de type numérique et permet de définir le nombre maximal de chiffres qui composent une décimale. Obligatoirement supérieure ou égale à zéro.

La restriction length

Définit le nombre exact de caractères d'une chaîne. Obligatoirement supérieure ou égale à zéro.

La restriction minLength

Définit le nombre minimum de caractères d'une chaîne. Obligatoirement supérieure ou égale à zéro.

La restriction maxLength

Définit le nombre maximum de caractères d'une chaîne. Obligatoirement supérieure ou égale à zéro.

La restriction enumeration

Permet d'énumérer la liste des valeurs possibles pour un élément ou un attribut.

La restriction whiteSpace

Spécifie le comportement à adopter par un élément lorsqu'il contient des espaces (tabulations, les retours à la ligne, etc).

3 valeurs possibles :



DAWAN – REPRODUCTION INTERDITE

Preserve : préserve tous les espaces.

Replace : remplace tous les espaces (tabulations, retour à la ligne, etc.) par des espaces "simples".

Collapse : supprime les espaces en début et fin de chaîne, de remplacer les tabulations et les retours à la ligne par un espace "simple" et de remplacer les espaces multiples par un espace "simple".

La restriction pattern

Défini des expressions rationnelles (ou expressions régulières). C'est un motif permettant de décrire le contenu attendu.

Les **expressions rationnelles** sont un langage à part entière.

La norme XPath

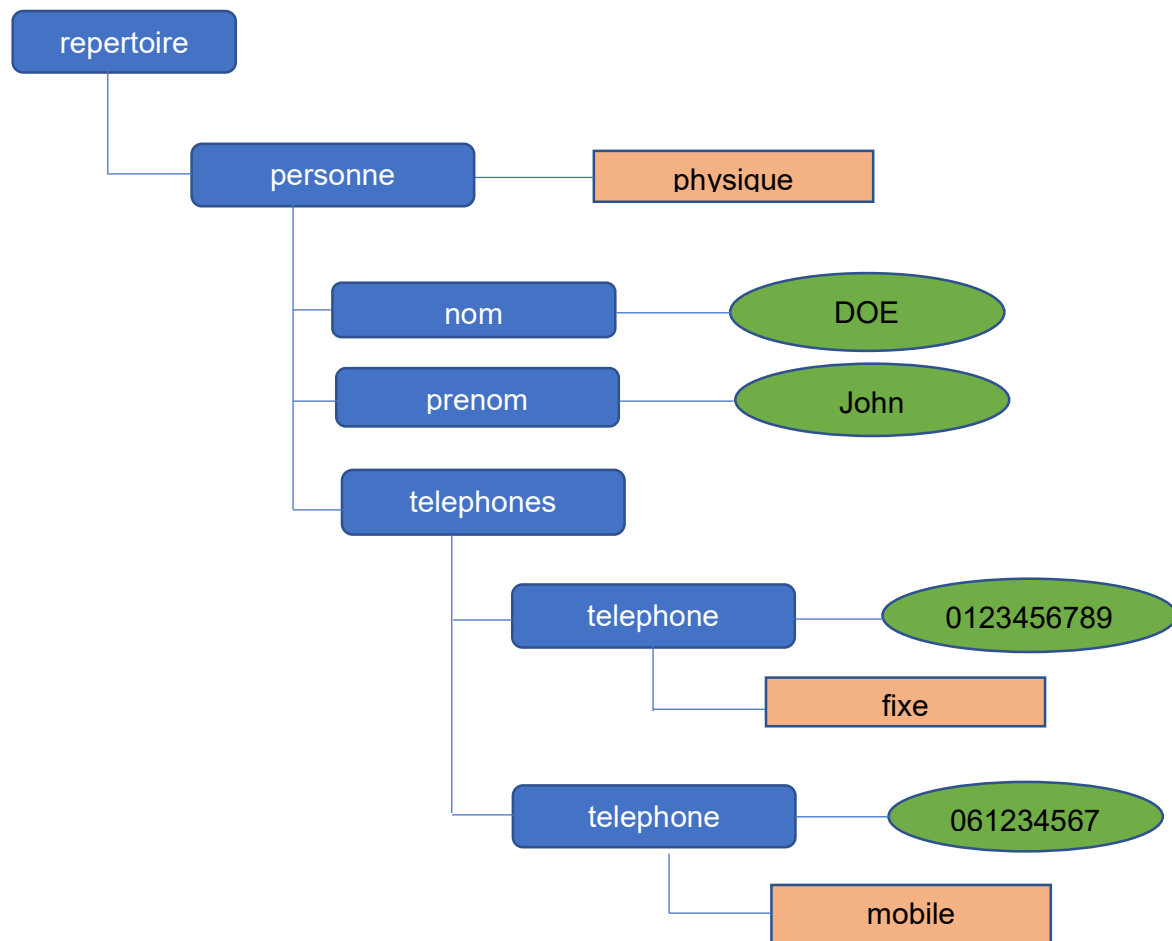
Nœuds d'un document

Les nœuds sont les embranchements que représente l'encapsulation des éléments dans d'autre élément, on appelle un arbre l'ensemble des embranchement.

Ce xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<repertoire>
  <!-- John DOE -->
  <personne type="physique">
    <nom>DOE</nom>
    <prenom>John</prenom>
    <telephones>
      <telephone type="fixe">01 02 03 04 05</telephone>
      <telephone type="portable">06 07 08 09 10</telephone>
    </telephones>
  </personne>
</repertoire>
```

Est traduit par l'arbre suivant :



Les axes

Nom de l'axe	Description
ancestor	oriente la recherche vers les ancêtres du nœud courant
ancestor-or-self	oriente la recherche vers le nœud courant et ses ancêtres
attribute	oriente la recherche vers les attributs du nœud courant
child	oriente la recherche vers les enfants du nœud courant
descendant	oriente la recherche vers les descendants du nœud courant
descendant-or-self	oriente la recherche vers le nœud courant et ses descendants
following	oriente la recherche vers les nœuds suivant le nœud courant
following-sibling	oriente la recherche vers les frères suivants du nœud courant
parent	oriente la recherche vers le père du nœud courant
preceding	oriente la recherche vers les nœuds précédant le nœud courant
preceding-sibling	oriente la recherche vers les frères précédents du nœud courant

Nom de l'axe	Description
self	oriente la recherche vers le nœud courant

Il existe des raccourcis, par exemple l'axe par défaut est child, donc il n'est pas utile de le préciser.

Il existe un axe namespace mais il est vu plus tard

Les nœuds

Nom	Description
nom du nœud	oriente la recherche vers le nœud dont le nom a explicitement été spécifié
*	oriente la recherche vers tous les nœuds
node()	oriente la recherche vers tous les types de nœuds (éléments, commentaires, attributs, etc.)
text()	oriente la recherche vers les nœuds de type texte
comment()	oriente la recherche vers les nœuds de type commentaire

Les prédicats

Nom du prédicat	Description
attribute	permet d'affiner la recherche en fonction d'un attribut
count()	permet de compter le nombre de nœuds
last()	permet de sélectionner le dernier nœud d'une liste
position()	permet d'affiner la recherche en fonction de la position d'un nœud
not()	Le nœud ou élément n'est pas ...

Introduction aux services web

Présentation des services web

Permet la communication et l'échange de données entre applications et systèmes hétérogènes.

Ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine (synchrone ou asynchrone).

En résumé un service web c'est :

Un agent qui fournit le service, un demandeur qui avec l'aide d'un agent de service utilise le service.



DAWAN – REPRODUCTION INTERDITE

Services web et SOA

SOA : Service Oriented Architecture ou en Fr Architecture Orientée Service (AOS)

Une forme d'architecture de médiation en un modèle d'interaction applicative mettant en œuvre des services.

Avantages

Les services web permettent l'interopérabilité, c'est-à-dire qu'ils sont dans le principe un code générique pour toutes les applications sur les différentes plateformes.

Les services web gèrent aussi l'authentification et les autorisations

Si j'ai un compte sur un site, je suis traité comme utilisateur authentifié du site. Si vous avez le droit de voir ce que je publie sur le site, vous êtes un utilisateur authentifié, donc en combinant ces deux informations :

Tous les utilisateurs authentifiés ayant accès aux ressources sont donc des utilisateurs autorisés pour ses ressources.

Donc pour les services web c'est la même chose, les identifiant et mot de passe sont utilisés pour l'authentification (un jeton ou token) et ce jeton sert aussi aux autorisations sur les ressources du service.

Les interfaces sont dynamiques (changements) sans devoir obligatoirement modifier la logique client lors de son interaction avec le service. On peut plus facilement et efficacement intégrer plusieurs logiciels.

Les normes

SOAP

Les services Web utilisent le protocole SOAP (Simple Object Access Protocol) qui utilise XML comme charge utile ou corps de requête. C'est un protocole avec état car il n'y a pas de méthode indépendante pour le type d'opération spécifique.

Toutes les demandes et réponses sont transmises en même temps via XML et aucune méthode indépendante comme GET, PUT, POST ou DELETE n'est explicitement fournie.

WDSL

Cette requête SOAP utilise Langage de description de services Web (WSDL) qui est un composant très utile du service Web.



DAWAN – REPRODUCTION INTERDITE

Cela définit où réside réellement le service Web et également le type de service Web à récupérer pour une demande spécifique. Cela utilise un fichier XML qui décrit la fonctionnalité du service Web.

UDDI

Un autre élément utile est UDDI . Cela signifie Universal Description Discovery and Integration. Il existe un fournisseur de services qui fournit le service Web. Par conséquent, pour un fournisseur de services particulier, cet UDDI est utilisé pour décrire, découvrir et publier ces services Web.

UDDI est chargé de laisser un client découvrir (UDDI fournit un référentiel pour WSDL) où se trouve le fichier XML du WSDL. C'est ainsi qu'un service Web est défini et décrit.

XML-RPC

Il signifie Extensible Markup Language - Remote Procedure. Un autre composant très important du service Web est XML - RPC qui est responsable de l'envoi de messages à travers les systèmes. Les demandes et les réponses sont sous forme de XML et sont envoyées / reçues via HTTP POST.

La meilleure caractéristique de XML-RPC est qu'une application client résidant sur une plate-forme différente peut communiquer avec un serveur différent. Il y a quelque chose appelé JSON-RPC qui a été expliqué dans la dernière partie de l'article car il ne forme pas un composant d'un service Web.

Plateformes à services web

SOAP : principes généraux

SOAP : Simplify Object Access Protocol

Un service SOAP est un service avec des états, qui utilise le langage XML pour former une enveloppe. Une enveloppe SOAP peut être décrite en deux parties, une est l'entête et le corps SOAP, l'autre le protocole pour envoyer les messages SOAP.

Dans l'entête nous avons l'authentification et l'autorisation, le corps fait partie de la section payload de la requête qui utilise WDSL pour décrire le service web, le protocole est principalement HTTP.

Les services SOAP disposent d'une sécurité (SSL – Secure Socket Layer) qui est chargée d'éviter la fuite de données pendant la transmission (cryptage et décryptage).

Les services web ont besoin d'un réseau pour fonctionner, ils doivent assurer la sécurité, ils possèdent pour cela trois entités importantes pour cela :



DAWAN – REPRODUCTION INTERDITE

Authentification et autorisation

Confidentialité, qui dépend du SSL

Sécurité internet, extraire toutes les réponses SOAP et XML-RPC

SOAP est une recommandation pour l'échange de messages au format XML entre un expéditeur (SOAP sender) et un destinataire (SOAP receiver). La structure des messages SOAP est identique qu'elle provienne de l'expéditeur (requête) ou du destinataire (réponse). Un message est constitué par une enveloppe (SOAP envelope) qui peut contenir un en-tête (SOAP header) et ensuite une charge utile (SOAP body).

Pour une utilisation avec le protocole HTTP, le message doit être envoyé avec la méthode POST à l'URL du service (appelée aussi endpoint)

Exemple d'un message SOAP

```
<!-- l'enveloppe du message -->
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"

  <!-- l'en-tête du message -->
  <soapenv:Header>
    <!-- les éventuelles méta-informations -->
  </soapenv:Header>

  <!-- le corps de la requête -->
  <soapenv:Body>
    <!-- le message (payload) -->
  </soapenv:Body>
</soapenv:Envelope>

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:m="http://www.mymeteo.com/meteo">
  <soapenv:Header>
  </soapenv:Header>
  <soapenv:Body>
    <m:releve>
      <m:temperature>
        <m:valeur>6.0</m:valeur>
        <m:unite>celsius</m:unite>
      </m:temperature>
      <m:lieu>
        <m:ville>Bordeaux</m:ville>
      </m:lieu>
    </m:releve>
  </soapenv:Body>
```



DAWAN – REPRODUCTION INTERDITE

```
</soapenv:Envelope>
```

Dans l'exemple ci-dessus le message contient de toute évidence un relevé météorologique pour la ville de bordeaux. Ce message peut aussi bien être une requête d'un expéditeur qu'une réponse d'un destinataire (la structure des messages est la même).

Structure d'un message SOAP

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:m="http://www.mymeteo.com/meteo">
  <soapenv:Header>
  </soapenv:Header>
  <soapenv:Body>
    <m:releve>
      <m:temperature>
        <m:valeur>6.0</m:valeur>
        <m:unite>celsius</m:unite>
      </m:temperature>
      <m:lieu>
        <m:ville>Bordeaux</m:ville>
      </m:lieu>
    </m:releve>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP définit un format pour un type de réponse particulière : une SOAP fault. Une SOAP fault signale un problème lors du traitement du service.

Une réponse contenant une SOAP fault

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>Internal server error</faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

Pour l'élément faultcode, on peut utiliser les valeurs :

- {http://schemas.xmlsoap.org/soap/envelope/}:Server pour signaler que l'origine de l'incident provient du traitement du serveur



DAWAN – REPRODUCTION INTERDITE

- {http://schemas.xmlsoap.org/soap/envelope/}:Client pour signaler que l'origine de l'incident provient de la requête du client



DAWAN – REPRODUCTION INTERDITE