

# Documents XML et API SaX et DOM

## Objectifs :

- Comprendre XML et les DTD
- Savoir utiliser SaX
- Savoir utiliser DOM

## Exercice 1 : Description du fichier /etc/interfaces

Le fichier /etc/network/interfaces décrit les interfaces réseau disponibles ainsi que la valeur de certains paramètres de configuration de ces interfaces. Le listing 1 est un exemple d'un tel fichier.

Listing 1 – Exemple de fichier /etc/network/interfaces

```
1 # /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)
2
3 # The loopback interface
4 auto lo
5 iface lo inet loopback
6
7 auto eth0 eth1
8
9 iface eth0 inet dhcp
10
11 iface eth1 inet static
12     address 147.127.18.49
13     netmask 255.255.240.0
14     gateway 147.127.18.200
```

Le listing 2 propose une DTD (Document Type Declaration) pour définir la structure des informations du fichier /etc/network/interfaces.

**1.1** Rappeler la signification des éléments constitutifs de la DTD du listing 2 et expliquer la signification de cette DTD.

**1.2** Écrire un document XML valide par rapport à la DTD du listing 2 qui décrit les informations contenues dans le listing 1. Sur machine, on peut utiliser `xmllint` pour vérifier qu'un document XML est bien formé et ajouter l'option `--valid` pour vérifier qu'il est valide par rapport à sa DTD. *Il est conseillé de traduire ligne à ligne le fichier d'exemple donné et de le vérifier au fur et à mesure<sup>1</sup> avec `xmllint`.*

**1.3** Dessiner l'arbre correspondant à ce document.

---

1. C'est un réflexe qui devrait déjà être acquis !

Listing 2 – DTD des utilisateurs d'un réseau informatique

```
1 <?xml version="1.0" encoding="ISO-8859-1" ?>
2
3 <!ELEMENT interfaces ((auto | iface)*)>
4 <!ELEMENT auto (name+)>
5 <!ELEMENT name EMPTY>
6 <!ENTITY % addressIP "#PCDATA">
7 <!ATTLIST name value CDATA #REQUIRED>
8
9 <!ELEMENT iface (inet)>
10 <!ATTLIST iface name CDATA #REQUIRED>
11
12 <!ELEMENT inet (loopback | dhcp | static)>
13 <!ELEMENT loopback EMPTY>
14
15 <!ELEMENT dhcp EMPTY>
16 <!ATTLIST dhcp hostname CDATA #IMPLIED>
17
18 <!ELEMENT static (address, netmask, gateway?)>
19 <!ELEMENT address (%addressIP;)>
20 <!ELEMENT netmask (%addressIP;)>
21 <!ELEMENT gateway (%addressIP;)>
```

**1.4** Indiquer comment modifier la DTD pour garantir que les noms utilisés après `iface` sont uniques (on ne peut pas donner deux définitions d'une même interface) et que les noms après `auto` correspondent à des noms d'interfaces définies<sup>2</sup>.

**1.5** Écrire un programme Java s'appuyant sur DOM ou JDOM pour construire ce document XML.

### Exercice 2 : Exploiter cette DTD

On souhaite répondre aux questions suivantes à partir d'un document XML valide par rapport à la DTD de l'exercice 1.

1. Combien y a-t-il d'interfaces « auto » ?
2. Combien y a-t-il d'interfaces définies (grâce à une ligne « iface ») ?
3. Quels sont les noms des interfaces « auto » ?
4. Y a-t-il des interfaces définies et non « auto » ? Quel est le nom de ces interfaces ?
5. Quelles sont les adresses des serveurs dhcp utilisés ?
6. Quelles sont les interfaces qui utilisent la passerelle 147.127.18.200 ?
7. Donner la liste des paramètres des `address` et `netmask` des interfaces définies **static** ?

Répondre aux questions précédentes pour le fichier `/etc/interfaces` proposé au listing 1. On expliquera comment extraire de la structure arborescente les informations nécessaires.

---

2. Ceci est vrai pour les configurations simples où l'on n'utilise que des noms physiques. Ce n'est pas le cas si on ajoute les « mappings » qui font le lien entre un nom physique (utilisé sur les lignes `auto`) et les noms logiques (utilisés sur les lignes `iface`).

**Exercice 3 : Exploiter un document XML avec DOM**

Utiliser l'API DOM pour répondre aux questions de l'exercice 2.

**Exercice 4 : Exploiter un document XML avec JDOM**

Utiliser l'API JDOM pour répondre aux questions de l'exercice 2.

**Exercice 5 : Exploiter un document XML avec SaX**

Utiliser l'API SaX pour répondre aux questions de l'exercice 2.

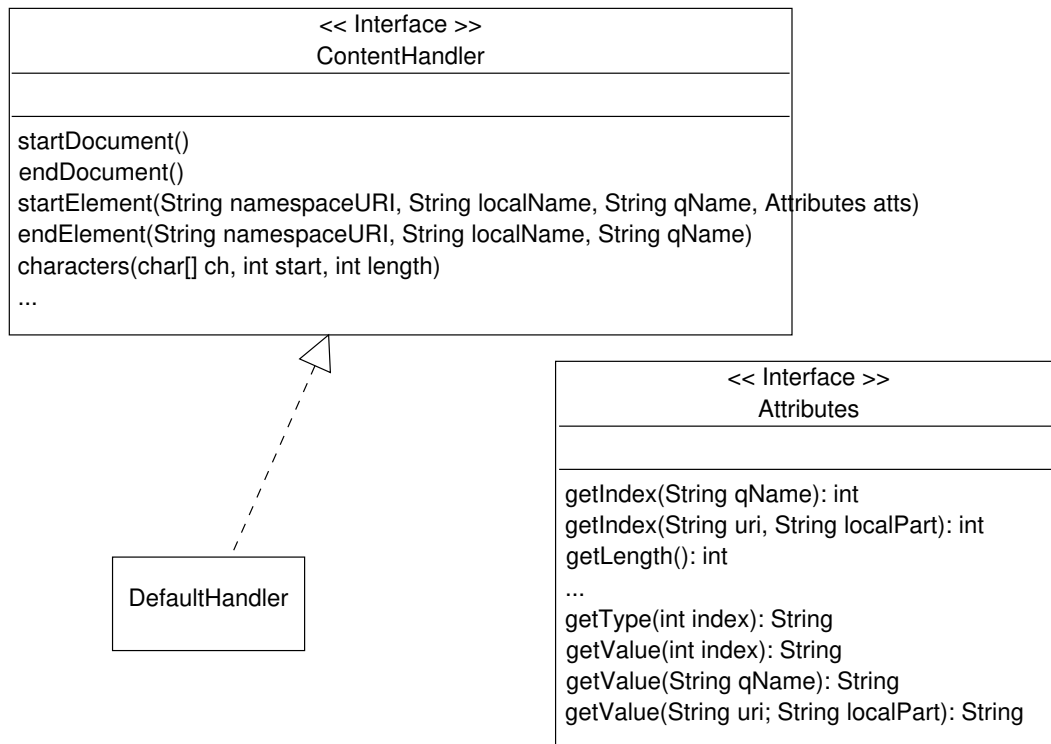


FIGURE 1 – Diagramme de classes de l'API SaX

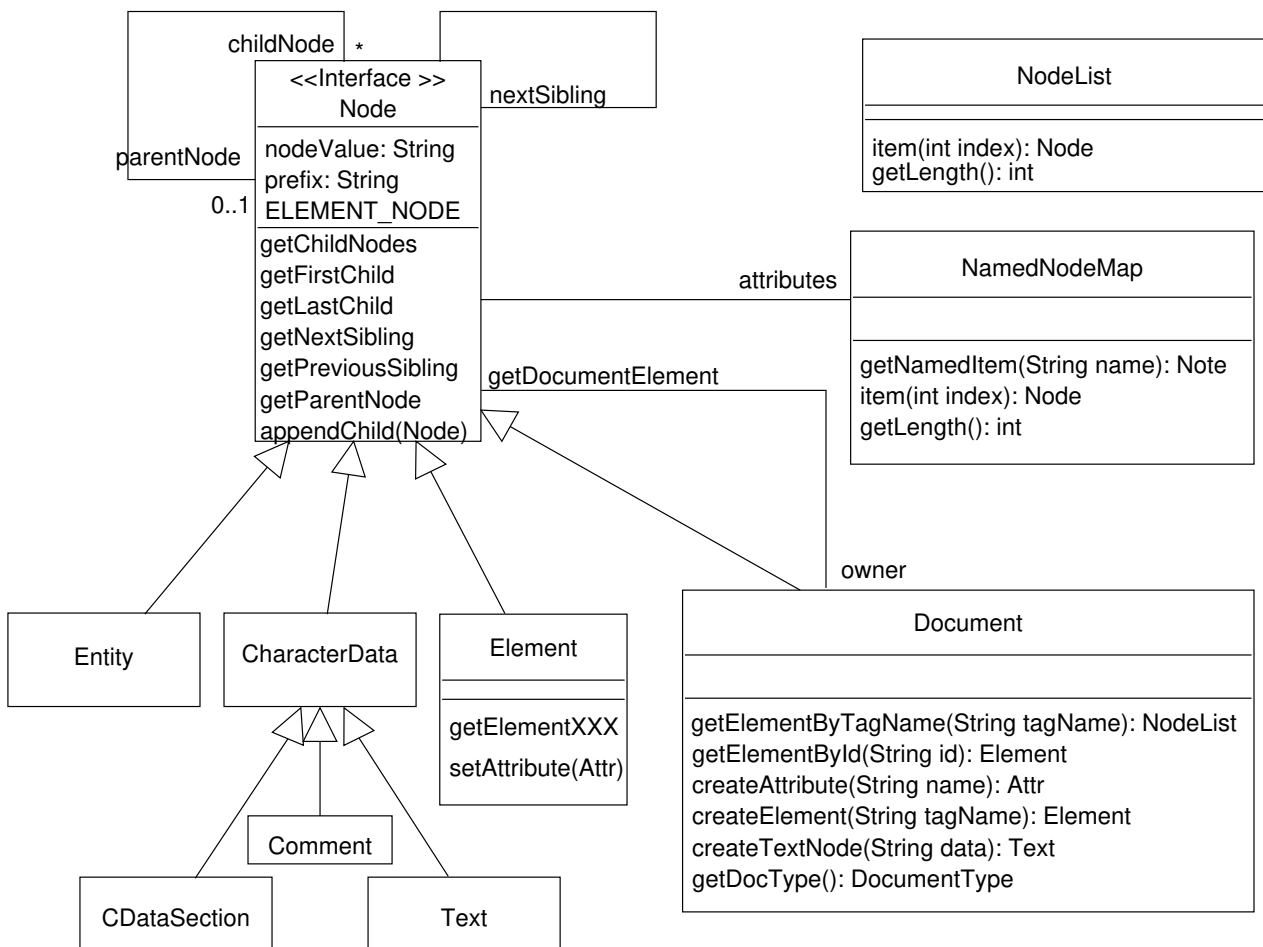


FIGURE 2 – Diagramme de classes de l’API DOM