Tutoriel Android
Comment lire et écrire un fichier XML en utilisant DOM

Romain Raveaux

Préambule

Il existe plusieurs manières de lire un fichier XML, deux API sont souvent utilisées : DOM et SAX. Il existe un bon tutoriel pour lire un fichier XML en utilisant SAX : http://thibault-koprowski.fr/2010/10/15/tutoriel-parsing-xml-sous-android/
Pour ma part, je vais illustrer comment lire et écrire un fichier XML en utilisant DOM.
Ce tutoriel est à but pédagogique, il a vocation à faire travailler les étudiants. Pour cette raison, le tutoriel n'est pas « clé en main » cependant vous y trouverez suffisamment d'éléments pour progresser.

Prérequis

Etre familier avec les documents XML, les langages à balises et le vocabulaire associé (XML, DTD, XSD).
Savoir créer une application Android.

SAX vs DOM

Une comparaison rapide :
DOM : Nécessite de charger en mémoire tout le document XML. DOM est donc gourmant en mémoire (heap). DOM est souvent lent car charger le document XML prend du temps. Il parfois même impossible de charger en mémoire tout le document XML (base de connaissance WikiPedia). Par contre accéder à l'information est plus rapide car l'arbre XML est stocké en mémoire.

SAX : Mode pas à pas, le fichier XML est traité ligne pas ligne. Le fichier XML est interprété ligne par ligne. Peu gourmant en mémoire mais SAX ne permet pas de filtrer ou d'accéder à l'information de manière « intelligente ».

Lecture

Le but est d'obtenir l'élément racine du document XML (élément racine = la balise racine). Dans le code ci-dessous, File est une chaine de caractères contenant le chemin du fichier XML à lire.

```
DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

        DocumentBuilder builder = factory.newDocumentBuilder();

        Document dom = builder.parse(new FileInputStream(File));
Element root = dom.getDocumentElement();
```

A partir de cette étape, on peut récupérer les nœuds du document XML et les attributs

Liste de noeuds

Par exemple dans un document XML, nous voulons récupérer toutes les balises « etape »
NodeList items = root.getElementsByTagName("etape");

Les attributs

Chaque balise « etape » possède des attributs dont un attribut « id ». Pour récupérer la valeur de cet attribut, il possible de procéder ainsi :

```
for (int i = 0; i < items.getLength(); i++) {
    Node item = items.item(i);
    String id = item.getAttributes().getNamedItem("id")
            .getNodeValue();
    …
```

Ecriture

Création du fichier

La méthode « Environment.getExternalStorageDirectory() » retourne le chemin de la SDCARD (souvent /mnt/sdcard)

```
File newxmlfile = new File(Environment.getExternalStorageDirectory()+"/fichier.xml");

try{

    newxmlfile.createNewFile();

}catch(IOException e){

    Log.e("IOException", "exception in createNewFile() method");

}
```

Création du flux

```
FileOutputStream fileos = null;

try{

    fileos = new FileOutputStream(newxmlfile);

}catch(FileNotFoundException e){

    Log.e("FileNotFoundException", "can't create FileOutputStream");

}
```

Ecriture dans le fichier

```
//On crée un XmlSerializer pour écrire les données

        XmlSerializer serializer = Xml.newSerializer();

        try {

            //On fixe le FileOutputStream comme sortie pour le serializer, utilisant
l'encodage UTF-8
            serializer.setOutput(fileos, "UTF-8");

            //Ecrit la déclaration XML <?xml avec le type d'encodange (if encodage
est non null) et le flag standalone (si standalone est non null)

            serializer.startDocument(null, Boolean.valueOf(true));

            //on fixe les options d'indentation
            serializer.setFeature("http://xmlpull.org/v1/doc/features.html#indent-
output", true);

            //On écrit le nœud racine appelé "root"
            serializer.startTag(null, "monxml");
            //ensuite on écrit une balise «etape »
            serializer.startTag(null, "etape");
            //on ecrit un attribut à la balise
            serializer.attribute(null, "id","01") ;
            //on écrit du texte à l'intérieur
            serializer.text(«valeur »);
            //puis on ferme la balise « étape »
            serializer.endTag(null, " etape ");
```

Finir le document

```
serializer.endDocument();
serializer.flush();
fileos.close();
```