

PROJET

Informatique Fondamentale

IMA2A4

Juin 2019

Etudiant

SECK Malick

Encadrant

J. DEQUIDT

1. Introduction

L'idée majeure de ce projet est de réaliser un programme capable d'analyser les fichiers sous le format JSON afin d'en déduire la validité ou non. Après cette première étape, le programme doit pouvoir convertir les fichiers JSON sous le format YAML.

Cela sera possible en mettant en pratique les notions vues en cours concernant les analyses lexicale et syntaxique grâce à l'outil ANTLR .

2. Travail demandé

2.1 Cahier des charges

Le travail à réaliser est le suivant:

- Définir les éléments lexicaux et syntaxiques qui permettent de vérifier qu'un fichier vérifie la syntaxe JSON
- Implémenter ces éléments avec ANTLR et obtenir l'arbre syntaxique abstrait (AST)
- Effectuer en Java les opérations nécessaires sur l'AST de manière à le convertir en un autre format de fichier. Dans un premier temps, vous ferez une conversion en YAML, s'il vous reste du temps vous ferez la conversion en XML

2.2 Le format JSON

Le format de fichier JSON (*JavaScript Object Notation*) est un standard ouvert pour représenter des objets en *Javascript*. Il permet de représenter de l'information structurée tout en étant lisible par un humain. Sa syntaxe est réduite et non-extensible et comprend:

- {...} : les accolades définissent un objet.
- "lexer" : "ANTLR" : Les guillemets (double-quotes) et les double-points définissent un couple clé/valeur (on parle de membre).
- [...] : Les crochets définissent un tableau.
- Les virgules permettent de séparer les membres d'un tableau ou d'un objet comme dans l'exemple suivant.

```
{  
  "menu": {
```

```

    "id": "file",
    "value": "File",
    "popup": {
        "menuitem": [
            { "value": "New", "onclick": "CreateNewDoc()" },
            { "value": "Open", "onclick": "OpenDoc()" },
            { "value": "Close", "onclick": "CloseDoc()" }
        ]
    }
}

```

3. Réalisations

3.1 Démarche

Tout d'abord, j'ai commencé par regarder les différents éléments contenus dans un fichier json (comme défini dans la partie 2.2) afin de faire des règles correspondantes dans mon fichier antlr.

Ainsi, j'arrive à détecter les objets d'une manière générale ainsi que les tableaux, les nombres et les chaînes de caractères.

Concernant ces dernières, je détecte simplement les combinaisons de lettres en majuscules et en minuscules ainsi que des parenthèses pour simplifier.

Le programme est capable de détecter des nombres écrits sous forme décimale mais aussi en notation scientifique.

Le programme peut être compilé et exécuté automatiquement grâce à un script nommé antlr.sh (avec les droits requis) dans le répertoire JSON.

3.2 Résultat

Comme le montre la figure ci-dessous on peut voir que le programme marche et est capable de traduire un fichier sous le format JSON en AST.

Cependant, ayant effectué les tests avec un nombre de fichier limité, je ne pourrai garantir son efficacité.

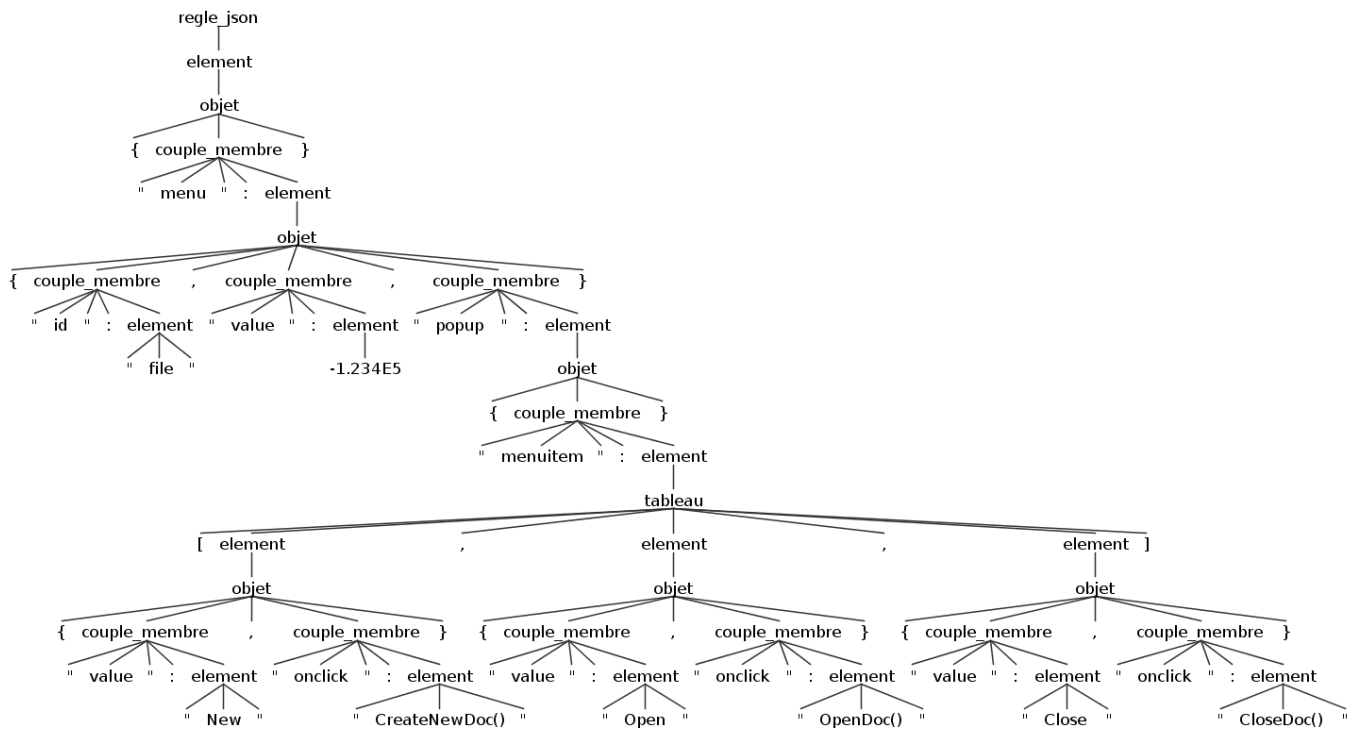


Figure 1: arbre syntaxique

3.3 Manquements

En ce qui concerne l'analyse lexicale et sémantique, je n'ai pas pu gérer les caractères unicode. De plus, je me suis limité à faire le parser et le lexer par souci de temps. En effet, je ne suis pas arrivé à traduire mon arbre syntaxique en fichier de format YAML ou XML en java.