

# Compiladors: Examen de laboratori de ANTLR4

13 d'abril de 2018

**IMPORTANT:** Cal entregar l'examen en un fitxer *.tgz* pujat al Racó. Llegiu les instruccions del final de l'enunciat per veure com generar-lo.

JSON es un format de representació de dades creat per a serialitzar objectes JavaScript, que s'ha extès com un estàndard d'intercanvi d'informació entre serveis i clients web.

Un objecte JSON pot ser:

- Un *string*, entre cometes dobles.
- Una llista, delimitada per [ i ], que conté elements separats per comes. Cada element de la llista pot ser un objecte JSON qualsevol.
- Un diccionari, delimitat per { i }, que conté els elements separats per comes. Cada element d'un diccionari és un parell *clau:valor* on la clau és un *string* (entre cometes dobles) i el valor pot ser qualsevol objecte JSON.

Escriviu una gramàtica i el corresponent *Listener* en ANTLR4 que llegeixi qualsevol objecte en JSON i el converteixi a XML, amb les següents condicions:

- Tot el XML de sortida s'inclou en un tag XML <OBJECT>.
- Les llistes JSON es converteixen en el tag <LIST>, el qual conté un tag <ITEM> per cada element de la llista. Si un element de la llista és un *string*, es genera en el format <ITEM value="string" />. Altrament, es genera un tag <ITEM> que conté l'XML corresponent a l'element de la llista.
- Els diccionaris JSON es converteixen en el tag <DICT>, el qual conté un tag <ENTRY> per a cada parell del diccionari. La clau del parell es posa com a atribut, en el format <ENTRY key="clau">. Si el valor del parell és un *string*, es posa també com un atribut: <ENTRY key="clau" value="valor">. Altrament, el valor s'expandeix en XML dins del tag <ENTRY>.

Per exemple, per al JSON:

```
{ "colors" : ["green", "yellow", "black", "red", "blue", "white" ],
  "shapes" : { "square" : { "side" : "23", "fill" : "true" },
               "rectangle" : { "side1" : "23", "side2" : "44" },
               "circle" : { "radius" : "12" }
            },
  "friends" : [ {"name" : "John", "phone" : "12345"},
                 {"name" : "Mary", "phone" : "54321"},
                 {"name" : "Peter", "phone" : "45123"},
                 {"name" : "Sally", "phone" : "32154"},
                 "a string may be here too",
                 ["or", "another", "list"]
            ]
}
```

Cal generar el següent XML:

```

<OBJECT>
  <DICT>
    <ENTRY key="colors">
      <LIST>
        <ITEM value="green" />
        <ITEM value="yellow" />
        <ITEM value="black" />
        <ITEM value="red" />
        <ITEM value="blue" />
        <ITEM value="white" />
      </LIST>
    </ENTRY>
    <ENTRY key="shapes">
      <DICT>
        <ENTRY key="square">
          <DICT>
            <ENTRY key="side" value="23" />
            <ENTRY key="fill" value="true" />
          </DICT>
        </ENTRY>
        <ENTRY key="rectangle">
          <DICT>
            <ENTRY key="side1" value="23" />
            <ENTRY key="side2" value="44" />
          </DICT>
        </ENTRY>
        <ENTRY key="circle">
          <DICT>
            <ENTRY key="radius" value="12" />
          </DICT>
        </ENTRY>
      </DICT>
    </ENTRY>
    <ENTRY key="friends">
      <LIST>
        <ITEM>
          <DICT>
            <ENTRY key="name" value="John" />
            <ENTRY key="phone" value="12345" />
          </DICT>
        </ITEM>
        <ITEM>
          <DICT>
            <ENTRY key="name" value="Mary" />
            <ENTRY key="phone" value="54321" />
          </DICT>
        </ITEM>
        <ITEM>
          <DICT>
            <ENTRY key="name" value="Peter" />
            <ENTRY key="phone" value="45123" />
          </DICT>
        </ITEM>
        <ITEM>
          <DICT>
            <ENTRY key="name" value="Sally" />
            <ENTRY key="phone" value="32154" />
          </DICT>
        </ITEM>
        <ITEM value="a string may be here too" />
        <ITEM>
          <LIST>
            <ITEM value="or" />
            <ITEM value="another" />
            <ITEM value="list" />
          </LIST>
        </ITEM>
      </LIST>
    </ENTRY>
  </DICT>
</OBJECT>

```

```

        </LIST>
      </ITEM>
    </LIST>
  </ENTRY>
</DICT>
</OBJECT>

```

**ATENCIÓ:** No és necessari que genereu la indentació del XML, o que us preocupeu d'on posar espais o salts de linia. Simplement genereu XML en un string, i useu una eina externa per formatjar la sortida, p.ex:

```
./json2xml input.json | xmllint -format -encode utf8 -
```

**CODI PER A L'EXAMEN:** Al Racó (examens.fib.upc.edu) trobareu un fitxer `examen.tgz` amb el següent contingut:

- Codi esquelet del conversor:
  - `main.cpp` : Programa principal, no l'heu de modificar.
  - `json.g4` : Gramàtica ANTLR4 que no fa pràcticament res. Cal completar-la.
  - `j2xmlListener.*` : *Listener* per a fer la conversió. Cal completar-lo amb els mètodes necessaris, segons els noms que poseu a les regles de la vostra gramàtica. El codi conté comentaris amb algunes pistes, llegiu-los amb atenció.
- `Makefile` : Makefile, no l'heu de modificar.
- `jps`: Subdirectori amb jocs de proves (`jpXX.json`), i la seva corresponent sortida esperada (`jpXX.out`).
- `evaluator.sh`: Script que executa tots els jocs de proves, formatja la sortida, i els compara amb la sortida esperada.
- `empaqueta.sh`: Script que crea un fitxer `examen-nom.cognom.tgz` amb la vostra solució. Aquest és el fitxer que cal pujar al Racó.