

# JSON Parser

ying dai

July 1, 2017

## Contents

<b>1</b>	<b>1</b>
<b>2 JSON</b>	<b>1</b>
2.1 JSON . . . . .	1
<b>3</b>	<b>4</b>
<b>4</b>	<b>7</b>

---

## 1

JSON

## 2 JSON

JSON(JavaScript Object Notation)

### 2.1 JSON

```
““ /””“ {” “}”””””””” /” ” “”  
value“[”]”””  
valuestring(number)truefalse nullobjectarray  
stringUnicodecharactercharacter string  
stringCJava  
numberCJava
```

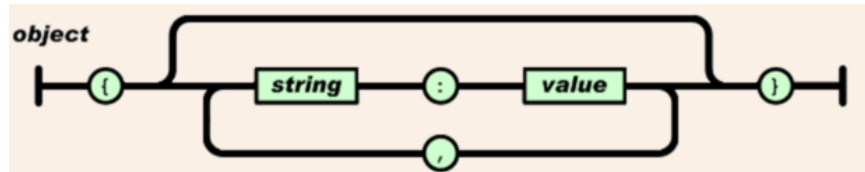


Figure 1: object

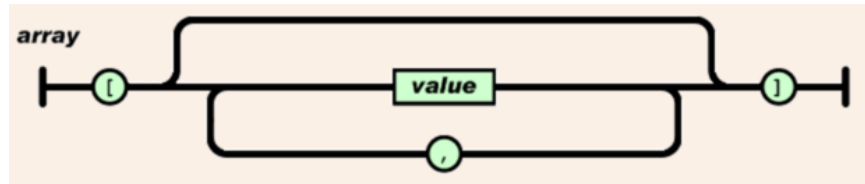


Figure 2: array

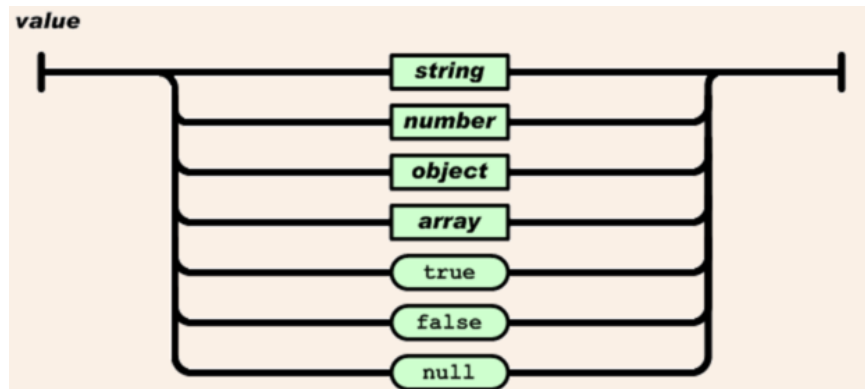


Figure 3: value

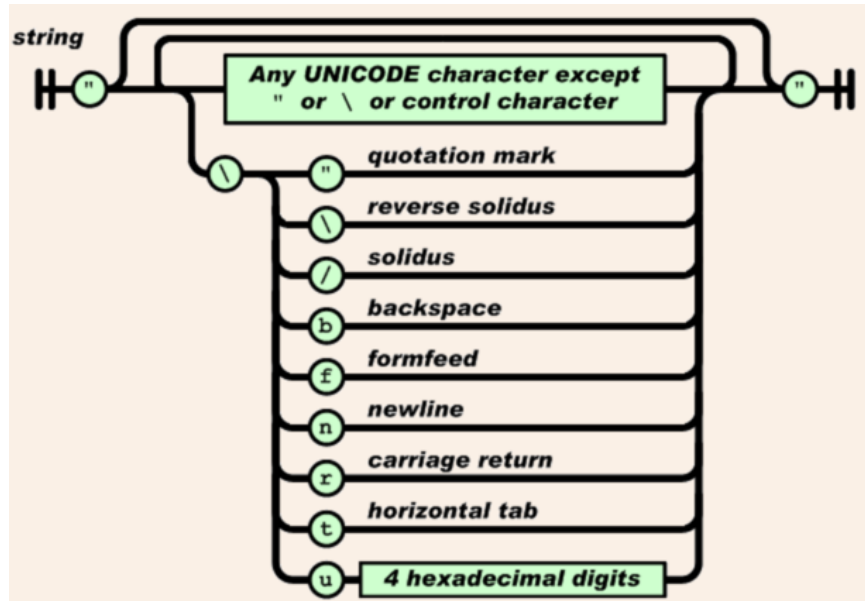


Figure 4: string

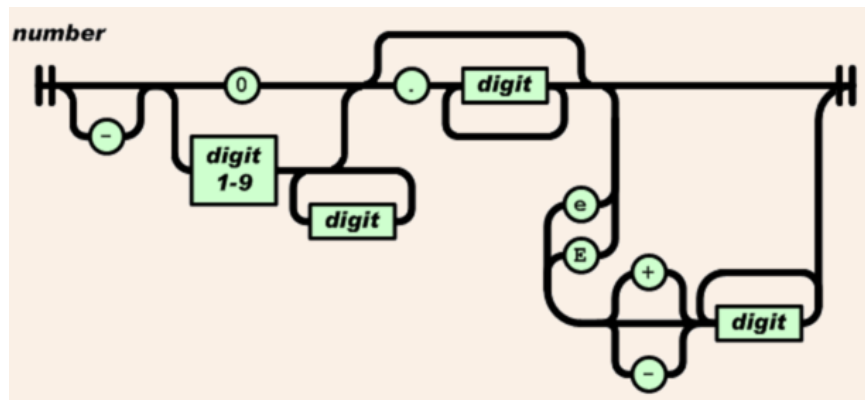


Figure 5: number

### 3

JSON JSON ANTLR  
ANTLR  
OS X

```
cd /usr/local/lib
sudo curl -O http://www.antlr.org/download/antlr-4.7-complete.jar
export CLASSPATH=".:usr/local/lib/antlr-4.7-complete.jar:$CLASSPATH"
alias antlr4='java -jar /usr/local/lib/antlr-4.7-complete.jar'
alias grun='java org.antlr.v4.gui.TestRig'
```

JSON.g4

```
/** Taken from "The Definitive ANTLR 4 Reference" by Terence Parr */
```

```
// Derived from http://json.org
grammar JSON;
```

```
json
    : value
    ;
```

```
obj
    : '{' pair (',' pair)* '}'
    | '{' '}'
    ;
```

```
pair
    : STRING ':' value
    ;
```

```
array
    : '[' value (',' value)* ']'
    | '[' ']'
    ;
```

```
value
    : STRING
    | NUMBER
    | obj
```

```

    | array
    | 'true'
    | 'false'
    | 'null'
    ;

STRING
    : '"' (ESC | ~ ["\\])* '"'
    ;
fragment ESC
    : '\\' (["\\/bfnrt] | UNICODE)
    ;
fragment UNICODE
    : 'u' HEX HEX HEX HEX
    ;
fragment HEX
    : [0-9a-fA-F]
    ;
NUMBER
    : '-'? INT '.' [0-9] + EXP? | '-'? INT EXP | '-'? INT
    ;
fragment INT
    : '0' | [1-9] [0-9]*
    ;
// no leading zeros
fragment EXP
    : [Ee] [+\\-]? INT
    ;
// \\- since - means "range" inside [...]
WS
    : [ \\t\\n\\r] + -> skip
    ;

import java.io.File;
import java.util.Scanner;

import org.antlr.v4.runtime.ANTLRInputStream;
import org.antlr.v4.runtime.CommonTokenStream;
import org.antlr.v4.runtime.tree.ParseTree;

```

```

public class JSONParserTest
{
    public static void main(String[] args) throws Exception
    {
        String content = new Scanner(new File("jsonExample.txt")).useDelimiter("\\Z").next();
        System.out.println( "JSON File:\n" + content + "\n\n");

        ANTLRInputStream input = new ANTLRInputStream( content );

        JSONLexer lexer = new JSONLexer(input);

        CommonTokenStream tokens = new CommonTokenStream(lexer);

        JSONParser parser = new JSONParser(tokens);

        ParseTree tree = parser.json();

        System.out.println( "ParseTree:\n" + tree.toStringTree( parser ) + "\n");
    }
}

jsonExample.txt
{
    "student":
    {
        "id" : "12345678",
        "prename" : "John",
        "surname" : "Doe",
        "address" :
        {
            "street" : "Johndoestreet",
            "postcode" : "99999"
        },
        "email" : "johndoe@doe.com"
    }
}

antlr4 JSON.g4

```

```
javac JSON*.java JSONParserTest.java
java JSONParserTest
```

```
(json (value (obj { (pair "student" : (value (obj { (pair "id" : (value "12345678")) ,
```

**4**

Python Demo, JSON GitHub scanner decoder JSON JSON parser