## ****WHAT IS JSON?****

JSON stands for **JavaScript Object Notation**.

JSON is a light-weight, language independent, English text-based data exchange technique which is convenient to read and write the data by both, humans and machines.



### ****WHAT IS THE STRUCTURE OF THE JSON FILE?****

JSON file has extension “.json”. It is represented in two structures one is Object and another one is the Array. JSON Object has the unordered collection of name-value pairs, whereas, the JSON Array has the unordered sequence of values.  These values could be anything like String, Integer, Boolean, Null, etc.

**SIGNIFICANCE OF MAINTAINING TEST DATA IN JSON FORMAT**

* Ease to maintain the test data
* Sometimes the excel sheet gets corrupted. We don’t face such issue with JSON
* Easy to understand
* Lightweight
* JSON data parsing is faster
* It is maintained in notepad or any other text file

### ****HOW TO IMPLEMENT JSON DATA PARSING IN THE SELENIUM PROJECT?****

There is a JAVA API which sets the environment for JSON data parsing, data generation, data transformation, and data query.  Read complete architecture of [JSON Simple here](https://code.google.com/archive/p/json-simple/).

Either you can [download the JSON simple jar file](http://www.java2s.com/Code/Jar/j/Downloadjsonsimple11jar.htm) or just put this dependency in your Maven project (Assuming you are using Maven as your build framework).

<!-- https://mvnrepository.com/artifact/com.googlecode.json-simple/json-simple -->

<dependency>

<groupId>com.googlecode.json-simple</groupId>

<artifactId>json-simple</artifactId>

<version>1.1.1</version>

</dependency>

[Or]

<http://www.java2s.com/Code/Jar/j/Downloadjsonsimple11jar.htm>

### ****EXPLANATION OF THE JSON SIMPLE API****

JSON Simple API has the object and array implementation as JSONObject and JSONArray respectively for JSON data parsing. JSONObject has the Map view implementation of zero or more data in key-value pairs, whereas, JSONArray has the List implementation of the data.

### ****JSON DATA PARSING COMMAND TO READ TEST DATA****

Below is the basic code which gives a command to read the data from the JSON file.

JSONParser parser = new JSONParser();

JSONArray jsonArr = (JSONArray)parser.parse(JSONArray jsonArr = (JSONArray)parser.parse(reader));

Object obj = jsonArr;

JSONObject jo = (JSONObject)obj;

String n = (String)jo.get(keyword);

**Storing Simple JSON Data**

* JSON stands for **J**ava**S**cript **O**bject **N**otation.
* It stores the data in a text format in an organized way which is easy to access.
* We create an object that we can access using a variable “**details**” (Refer to the code snippet below).
* The variable value is an object that contains multiple properties using a “name:value” pair.

var details = {

"Age" : "24",

"Hometown" : "Chennai, India",

"Gender" : "Male"

};

### ****Storing JSON data in Arrays****

* We can also have multiple sets of data in JSON. To do this, we can store multiple objects in square brackets. We call this as a JSON Array.

var empdetails = [{

"Name" : "John",

"Designation" : "Software Engineer",

"Gender" : "Male"

},

{

"Name" : "Peter",

"Designation" : "Manager",

"Gender" : "Male"

}];

To access this information, we need to access the array index of the empdetails variable we wish to access.

**package** com.google.tests;

**import** java.io.FileNotFoundException;

**import** java.io.FileReader;

**import** java.io.IOException;

**import** org.json.simple.JSONArray;

**import** org.json.simple.JSONObject;

**import** org.json.simple.parser.JSONParser;

**import** org.json.simple.parser.ParseException;

**public** **class** JSONDataExample {

String n;

**public** String readJSONData(String keyword){

JSONParser parser = **new** JSONParser();

FileReader reader;

**try** {

reader = **new** FileReader("G:\\Selenium\\eclipse-workspace\\FirstDemo\\Testdata.json");

JSONArray jsonArr = (JSONArray)parser.parse(reader);

**for**(Object obj : jsonArr){

JSONObject jo = (JSONObject)obj;

n = (String)jo.get(keyword);

}

} **catch** (FileNotFoundException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}**catch** (ParseException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}**catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** n;

}

**public** **static** **void** main(String[] args){

JSONDataExample e = **new** JSONDataExample();

String a;

a = e.readJSONData("first name");

System.***out***.println(a);

}

}

**Testdata.json file**

[

{

"first name":"Avinash",

"last name":"Mishra",

"Blog URL":"http://www.theavinashmishra.com"

}

]

**Output:**

Avinash

Reference links:

<http://www.pavantestingtools.com/p/selenium.html> -- basics

# How to Change Eclipse Java Version for Compiler and JRE

<https://www.concretepage.com/ide/eclipse/how-to-change-eclipse-java-version-for-compiler-and-jre>