

Java JSON

Using JSON.simple library



json-simple

JSON.simple - A simple Java toolkit for JSON

 Search projects[Project Home](#)[Downloads](#)[Wiki](#)[Issues](#)[Source](#)[Export to GitHub](#)

READ-ONLY: This project has been [archived](#). For more information see [this post](#).

[Summary](#) [People](#)

Project Information

[Project feeds](#)**Code license**[Apache License 2.0](#)**Labels**

JSON, JSON.simple, Java, parser, encode, decode, marshal, unmarshal, json_simple, ajax, library, tutorial, SAX-like, stoppable, streaming

**Members**[fangyid...@gmail.com](#)[5 committers](#)

Featured

**Downloads**[json-simple-1.1.1.jar](#)[json_simple-1.1-all.zip](#)[json_simple-1.1.jar](#)[Show all »](#)**Wiki pages**[ChangeLog](#)[Show all »](#)

GitHub

<https://github.com/fangyidong/json-simple>

Overview

JSON.simple is a simple Java toolkit for JSON. You can use JSON.simple to encode or decode JSON text.

Features

- Full compliance with [JSON specification](#) (RFC4627) and reliable (see [compliance testing](#))
- Provides multiple functionalities such as encode, decode/parse and escape JSON text while keeping the library lightweight
- Flexible, simple and easy to use by reusing Map and List interfaces
- Supports streaming output of JSON text
- Stoppable SAX-like interface for streaming input of JSON text (learn [more](#))
- Heap based parser
- High performance (see [performance testing](#))
- No dependency on external libraries
- Both of the source code and the binary are JDK1.2 compatible

Getting Started

Note: You need to put the latest [json-simple-1.1.1.jar](#) in your CLASSPATH before compiling and running the example codes.

Write JSON to file (imports)

```
import java.io.FileWriter;  
import java.io.IOException;  
import org.json.simple.JSONArray;  
import org.json.simple.JSONObject;
```

Write JSON to file (objects and arrays)

```
JSONObject obj = new JSONObject();  
obj.put("name", "talepis");  
obj.put("age", new Integer(100));
```

```
JSONArray list = new JSONArray();  
list.add("mymsg 1");  
list.add("mymsg 2");  
list.add("mymsg 3");  
obj.put("messages", list);
```

Write JSON to file (write to file)

```
try {  
    FileWriter file = new FileWriter("c:\\myjavaprogs\\test1.json");  
    file.write(obj.toJSONString());  
    file.flush();  
    file.close();  
} catch (IOException e) {  
    e.printStackTrace();  
}
```



```
import java.io.FileWriter;
import java.io.IOException;
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;


public class Json1 {
    public static void main(String[] args) {

        JSONObject obj = new JSONObject();
        obj.put("name", "talepis");
        obj.put("age", new Integer(100));

        JSONArray list = new JSONArray();
        list.add("mymsg 1");
        list.add("mymsg 2");
        list.add("mymsg 3");
        obj.put("messages", list);

        try {
            FileWriter file = new FileWriter("c:\\myjavaprogs\\test1.json");
            file.write(obj.toJSONString());
            file.flush();
            file.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
        System.out.print(obj);
    }
}
```

Compile and Run in Cmd



```
Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\timmy>cd..

C:\Users>cd..

C:\>cd myjavaprogs

C:\myjavaprogs>javac Json1.java -cp json-simple-1.1.1.jar
Note: Json1.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

C:\myjavaprogs>java -cp .;json-simple-1.1.1.jar Json1
{"name":"talepis","messages":["mymsg 1","mymsg 2","mymsg 3"],"age":100}
C:\myjavaprogs>
```

Read JSON from file (imports)

```
import java.io.FileNotFoundException;  
import java.io.FileReader;  
import java.io.IOException;  
import java.util.Iterator;  
import org.json.simple.JSONArray;  
import org.json.simple.JSONObject;  
import org.json.simple.parser.JSONParser;  
import org.json.simple.parser.ParseException;
```


Read JSON from file (read json file)

```
Object obj = parser.parse(new FileReader("c:\\myjavaprogs\\test1.json"));
JSONObject jsonObject = (JSONObject) obj;
String name = (String) jsonObject.get("name");
System.out.println(name);
long age = (Long) jsonObject.get("age");
System.out.println(age);
```

Read JSON from file (iterate through array)

```
JSONArray msg = (JSONArray) jsonObject.get("messages");
Iterator<String> iterator = msg.iterator();
while (iterator.hasNext()) {
    System.out.println(iterator.next());
}
```

Compile and Run in Cmd



```
Command Prompt
C:\myjavaprogs>javac Json2.java -cp json-simple-1.1.1.jar
Note: Json2.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

C:\myjavaprogs>java -cp .;json-simple-1.1.1.jar Json2
talepis
100
mymsg 1
mymsg 2
mymsg 3

C:\myjavaprogs>_
```


Get a JSON file from Openweathermap And use it!

```
{"coord":{"lon":23.62,"lat":37.96},"weather":[{"id":802,"main":"Clouds","description":"scattered clouds","icon":"03d"}],"base":"stations","main":{"temp":298.17,"pressure":1011,"humidity":60,"temp_min":295.93,"temp_max":300.93},"visibility":10000,"wind":{"speed":1},"clouds":{"all":40},"dt":1445425154,"sys":{"type":1,"id":5673,"message":0.0067,"country":"GR","sunrise":1445402389,"sunset":1445441991},"id":260204,"name":"Keratsini","cod":200}
```