# Resources

Source: <http://json.org/>

Driver:

* Oracle JSON-P: <https://javaee.github.io/jsonp/>
* Google GSON: <http://repo1.maven.org/maven2/com/google/code/gson/gson/2.8.1/>

Sample: <http://www.oracle.com/technetwork/articles/java/json-1973242.html>

Online Json document:

<http://jsonstudio.com/resources/>

<http://geojson.io>

<http://mysafeinfo.com/api/data?list=states&format=json&abbreviate=false>

<http://mysafeinfo.com/api/data?list=presidents&format=jsonp&abbreviate=false>

# Programming

## Exercise 1

Dùng JSON-P chuyển đổi đối tượng sang json và ngược lại

|  |
| --- |
| **package** session03.ex01;  **public** **class** Employee {  **private** **long** id;  **private** String name;  **private** **double** salary;    **public** Employee(**long** id, String name, **double** salary) {  **this**.id = id;  **this**.name = name;  **this**.salary = salary;  }  **public** Employee() {  }  **public** **long** getId() {  **return** id;  }  **public** **void** setId(**long** id) {  **this**.id = id;  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  **public** **double** getSalary() {  **return** salary;  }  **public** **void** setSalary(**double** salary) {  **this**.salary = salary;  }  @Override  **public** String toString() {  **return** "Employee [id=" + id + ", name=" + name + ", salary=" + salary + "]";  }  } |
| **import** java.io.FileOutputStream;  **import** java.io.PrintWriter;  **import** javax.json.Json;  **import** javax.json.JsonObject;  **import** javax.json.JsonObjectBuilder;  **public** **class** EncodeJson {  **public** **static** **void** main(String[] args) **throws** Exception {  EncodeJson ej=**new** EncodeJson();  Employee e=**new** Employee(10001, "Thân Thị Đẹt", 10000d);  String js=ej.genjson(e);  ej.export("json/emp.json", js);  }  **public** String genjson(Employee e) {  JsonObjectBuilder builder=Json.*createObjectBuilder*();  builder.add("id", e.getId());  builder.add("name", e.getName());  builder.add("salary", e.getSalary());  JsonObject jo=builder.build();  **return** jo.toString();  }    **public** **void** export(String filePath, String json)**throws** Exception{  PrintWriter out=**new** PrintWriter(**new** FileOutputStream(filePath),**true**);  out.println(json);  out.close();  }  } |
| **import** java.io.FileInputStream;  **import** java.io.InputStream;  **import** javax.json.Json;  **import** javax.json.JsonNumber;  **import** javax.json.JsonObject;  **import** javax.json.JsonReader;  **import** session03.ex01.Employee;  **public** **class** DecodeJson {  **public** **static** **void** main(String[] args) **throws** Exception{  InputStream in=**new** FileInputStream("json/emp.json");  JsonReader reader=Json.*createReader*(in);    JsonObject jo = reader.readObject();  JsonNumber id = jo.getJsonNumber("id");  String name = jo.getString("name");  JsonNumber sal = jo.getJsonNumber("salary");  Employee emp=**new** Employee(  id.longValue(),  name,  sal.doubleValue()  );  System.***out***.println(emp);  }  } |

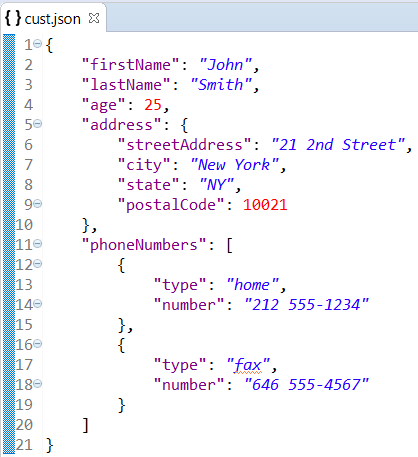
Sử dụng JSON-P Streaming API

|  |
| --- |
| **import** java.io.StringReader;  **import** javax.json.Json;  **import** javax.json.stream.JsonParser;  **import** javax.json.stream.JsonParser.Event; |
| **public** **static** **void** main(String[] args) {  **final** String result = "{\"name\":\"Falco\",\"age\":3,\"bitable\":false}";  **final** JsonParser parser = Json.*createParser*(**new** StringReader(result));  String key = **null**;  String value = **null**;  **while** (parser.hasNext()) {  **final** Event event = parser.next();  **switch** (event) {  **case** ***KEY\_NAME***:  key = parser.getString();  System.***out***.println(key);  **break**;  **case** ***VALUE\_STRING***:  value = parser.getString();  System.***out***.println(value);  **break**;  **case** ***END\_ARRAY***:  **break**;  **case** ***END\_OBJECT***:  **break**;  **case** ***START\_ARRAY***:  **break**;  **case** ***START\_OBJECT***:  **break**;  **case** ***VALUE\_FALSE***:  **break**;  **case** ***VALUE\_NULL***:  **break**;  **case** ***VALUE\_NUMBER***:  **break**;  **case** ***VALUE\_TRUE***:  **break**;  **default**:  **break**;  }  }  parser.close();  } |

## Exercise 2

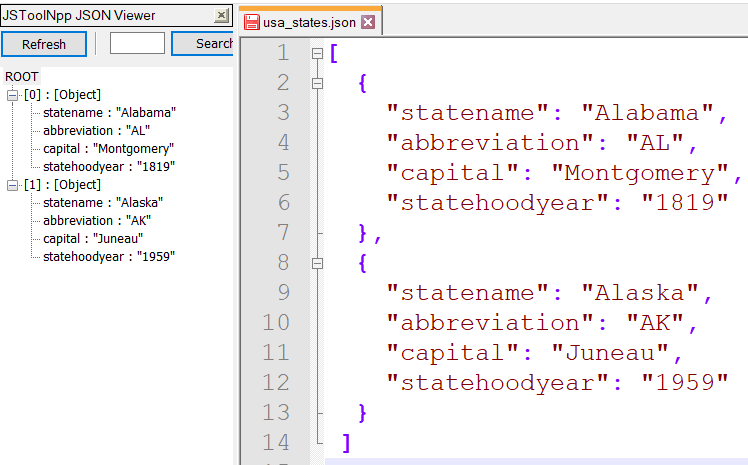
Cho tài liệu JSON có cấu trúc như sau:

Viết các lớp đối tượng. Sau đó, dùng JSON-P chuyển đổi đối tượng sang json và ngược lại.



## Exercise 3

Cho tài liệu JSON có cấu trúc như sau:



Tài liệu đầy đủ có tại: <http://mysafeinfo.com/api/data?list=states&format=json&abbreviate=false>

1. Tạo các lớp với các phương thức thực hiện các công việc sau
   1. Tìm kiếm đối tượng State theo abbreviation, trả về null nếu không tồn tại

**public State findByAb(String abb)**

* 1. Tìm kiếm các đối tượng State có state-hood-year(ie. năm thành lập) trước một năm được cung cấp

**public ArrayList<State>findByYear(int year)**

1. Tạo các unit test-cases để test các phương thức trên

## Exercise 4

Sử dụng Gson

Cho lớp Employee như sau:

|  |
| --- |
| **package** session02.jsonps.gen;  **public** **class** Employee {  **private** **long** id;  **private** String name;  **public** Employee() {  }  **public** Employee(**long** id, String name) {  **super**();  **this**.id = id;  **this**.name = name;  }  **public** **long** getId() {  **return** id;  }  **public** **void** setId(**long** id) {  **this**.id = id;  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  @Override  **public** String toString() {  **return** id+", "+name;  }  } |

1. Convert Java object to JSON

|  |
| --- |
| **public** **void** convertObj2JSON(Employee emp) **throws** Exception{  Gson gson = **new** Gson();  // 1. Java object to JSON, and save into a file  gson.toJson(emp, **new** FileWriter("D:\\file.json"));  // 2. Java object to JSON, and assign to a String  String jsonInString = gson.toJson(emp);  System.***out***.println(jsonInString);  } |

1. Convert JSON to Java object

|  |
| --- |
| **public** **void** convertJSON2Object() **throws** Exception{  Gson gson = **new** Gson();  // 1. JSON to Java object, read it from a file.  Employee emp1 = gson.fromJson(**new** FileReader("json/file.json"), Employee.**class**);  // 2. JSON to Java object, read it from a Json String.  String jsonInString = "{\"id\":1001,\"name\":\"Than thi det\"}";  Employee emp2 = gson.fromJson(jsonInString, Employee.**class**);  // 3. JSON to JsonElement, convert to String later.  JsonElement json = gson.fromJson(**new** FileReader("json/file.json"), JsonElement.**class**);  String result = gson.toJson(json);    System.***out***.println(emp1);  System.***out***.println(emp2);  System.***out***.println(result);  } |

## Exercise 5

Viết phương thức tạo một đối tượng JSONObject từ một Map<String,Object>

## Exercise 6

\*\* Viết chương trình kết nối cơ sở dữ liệu đọc và hiển thị danh sách các bảng sau đó cho phép chọn bảng để export ra định dạng JSON của bảng dữ liệu

Read more

GSON

<https://www.mkyong.com/java/how-do-convert-java-object-to-from-json-format-gson-api/>

<https://www.mkyong.com/java/gson-streaming-to-read-and-write-json/>