

## LAB JSON

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

Driver:

- Oracle JSON-P: <https://javaee.github.io/jsonp/>
- Google GSON: <https://mvnrepository.com/artifact/com.google.code.gson/gson>

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

Online Json document:

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

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

## PROGRAMMING

### BÀI TẬP 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

```

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();
}

```

## BÀI TẬP 2

Cho tài liệu JSON có cấu trúc như sau:

```

{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": {

```

```

    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": 10021
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "fax",
      "number": "646 555-4567"
    }
  ]
}

```

Viết các lớp đối tượng. Sau đó,

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

→ Kết quả xuất ra màn hình hoặc đọc ghi file tùy ý (tạo 2 project riêng).

```

<!-- https://mvnrepository.com/artifact/javax.json/javax.json-api -->
<dependency>
  <groupId>javax.json</groupId>
  <artifactId>javax.json-api</artifactId>
  <version>1.1.4</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.glassfish/javax.json -->
<dependency>
  <groupId>org.glassfish</groupId>
  <artifactId>javax.json</artifactId>
  <version>1.1.4</version>
</dependency>

<dependency>
  <groupId>com.google.code.gson</groupId>
  <artifactId>gson</artifactId>
  <version>2.8.5</version>
</dependency>

```

### BÀI TẬP 3

Cho tài liệu JSON có cấu trúc như sau:

```

[
  {
    "StateName": "Alabama",
    "Abbreviation": "AL",
    "Capital": "Montgomery",
    "Statehood": 1819,
    "ID": 1
  }
]

```

```

    },
    {
      "StateName": "Alaska",
      "Abbreviation": "AK",
      "Capital": "Juneau",
      "Statehood": 1959,
      "ID": 2
    }
  ]

```

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 (*dùng object model và streaming APIs*)
  - a. Tìm kiếm đối tượng State theo abbreviation, trả về null nếu không tồn tại  
     + *findByAb(abb:String): State*
  - b. Tìm kiếm các đối tượng State có state-hood-year(ie. năm thành lập) trước một năm được cung cấp  
     + *findByYear(year:int): List<State>*
2. Tạo các unit test-cases để test các phương thức trên

#### BÀI TẬP 4

Cho tài liệu JSON có cấu trúc như sau

```

[
  {
    "sku": 43900,
    "name": "Duracell - AAA Batteries (4-Pack)",
    "type": "HardGood",
    "price": 5.49,
    "upc": "041333424019",
    "category": [
      {
        "id": "pcmcat312300050015",
        "name": "Connected Home & Housewares"
      },
      {
        "id": "pcmcat248700050021",
        "name": "Housewares"
      },
      {
        "id": "pcmcat303600050001",
        "name": "Household Batteries"
      },
      {
        "id": "abcat0208002",
        "name": "Alkaline Batteries"
      }
    ],
    "shipping": 5.49,
    "description": "Compatible with select electronic devices; AAA size; DURALOCK Power Preserve technology; 4-pack",
    "manufacturer": "Duracell",
    "model": "MN2400B4Z",
  }
]

```

```

        "url": "http://www.bestbuy.com/site/duracell-aaa-batteries-4-pack/43900.p?id=1051384074145&skuId=43900&cmp=RMXCC",
        "image": "http://img.bbystatic.com/BestBuy_US/images/products/4390/43900_sa.jpg"
    },
    {
        "sku": 48530,
        "name": "Duracell - AA 1.5V CopperTop Batteries (4-Pack)",
        "type": "HardGood",
        "price": 5.49,
        "upc": "041333415017",
        "category": [
            {
                "id": "pcmcat312300050015",
                "name": "Connected Home & Housewares"
            },
            {
                "id": "pcmcat248700050021",
                "name": "Housewares"
            },
            {
                "id": "pcmcat303600050001",
                "name": "Household Batteries"
            },
            {
                "id": "abcat0208002",
                "name": "Alkaline Batteries"
            }
        ],
        "shipping": 5.49,
        "description": "Long-lasting energy; DURALOCK Power Preserve technology; for toys, clocks, radios, games, remotes, PDAs and more",
        "manufacturer": "Duracell",
        "model": "MN1500B4Z",
        "url": "http://www.bestbuy.com/site/duracell-aa-1-5v-coppertop-batteries-4-pack/48530.p?id=1099385268988&skuId=48530&cmp=RMXCC",
        "image": "http://img.bbystatic.com/BestBuy_US/images/products/4853/48530_sa.jpg"
    }
]

```

Tài liệu đầy đủ có tại: <https://github.com/BestBuyAPIs/open-data-set/blob/master/products.json>

- a. Tạo các lớp với các phương thức thực hiện các công việc sau
  - a. Tìm kiếm đối tượng Product theo sku, trả về null nếu không tồn tại  
+ *findBySku(sku:int): Product*
  - b. Tìm kiếm các đối tượng Product theo price (tìm trong khoảng từ .. đến)  
+ *findByPrice(from:double, to:double): List<Product>*
  - c. Tìm kiếm theo các tiêu chí khác
- b. Tạo các unit test-cases để test các phương thức trên

## BÀI TẬP 5

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

## **BÀI TẬP 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/>