# 1. Thư viện

Example: http://websystique.com/java/json/jackson-json-annotations-example/

Muốn sử dụng Json Annotation phải thêm thư viện:

\_ jackson-annotations-2.6.3

**import** com.fasterxml.jackson.annotation.\*;

\_ jackson-databind-2.6.3

**import** com.fasterxml.jackson.databind.annotation.JsonSerialize;

\_ jackson-core-2.6.3.jar

**import** com.fasterxml.jackson.core.JsonGenerator;

**import** com.fasterxml.jackson.core.JsonProcessingException;

# 2. Demo

## Class Car:

**package** com.jsonannotations;

**import** java.util.\*;

**import** com.fasterxml.jackson.annotation.\*;

**import** com.fasterxml.jackson.databind.annotation.JsonSerialize;

@JsonIgnoreProperties({"ignoreme1","ignoreme2"})

**public** **class** Car {

@JsonProperty

**private** String name;

@JsonProperty("carModel")

**private** **int** model;

@JsonProperty

**private** String price;

**private** String ignoreme1;

**private** String ignoreme2;

@JsonProperty

**private** List<String> colors = **new** ArrayList<>();

@JsonProperty

@JsonSerialize(using = CustomDateSerializer.**class**)

**private** Date promoDate;

**private** Map<String, Object> otherProperties = **new** HashMap<>();

@JsonAnyGetter

**public** Map<String, Object> getOtherProperties() {

**return** otherProperties;

}

@JsonAnySetter

**public** **void** setOtherProperties(Map<String, Object> otherProperties) {

**this**.otherProperties = otherProperties;

}

@JsonAnySetter

**public** **void** set(String name, Object value){

otherProperties.put(name, value);

}

@Override

**public** String toString() {

**return** "Car [name=" + name + ", model=" + model + ", price=" + price

+ ", ignoreme1=" + ignoreme1 + ", ignoreme2=" + ignoreme2

+ ", colors=" + colors + ", promoDate=" + promoDate

+ ", otherProperties=" + otherProperties + "]";

}

}

## Class CustomDateSerializer:

**package** com.jsonannotations;

**import** java.io.IOException;

**import** java.text.SimpleDateFormat;

**import** java.util.\*;

**import** com.fasterxml.jackson.core.JsonGenerator;

**import** com.fasterxml.jackson.core.JsonProcessingException;

**import** com.fasterxml.jackson.databind.JsonSerializer;

**import** com.fasterxml.jackson.databind.SerializerProvider;

**public** **class** CustomDateSerializer **extends** JsonSerializer<Date>{

**private** **static** **final** SimpleDateFormat ***dateFormat*** = **new** SimpleDateFormat("dd/MM/yyyy");

@Override

**public** **void** serialize(Date date, JsonGenerator generator, SerializerProvider provider)

**throws** IOException, JsonProcessingException {

String formattedDate = ***dateFormat***.format(date);

generator.writeString(formattedDate);

}

}

## Class Demo:

**package** com.jsonannotations;

**import** java.io.File;

**import** java.io.IOException;

**import** com.fasterxml.jackson.core.JsonParseException;

**import** com.fasterxml.jackson.databind.JsonMappingException;

**import** com.fasterxml.jackson.databind.ObjectMapper;

**public** **class** JsonJacksonAnnotationsDemo {

**public** **static** **void** main(String[] args) **throws** JsonParseException, JsonMappingException, IOException {

ObjectMapper mapper = **new** ObjectMapper();

Car car = mapper.readValue(**new** File("inputfile.json"), Car.**class**);

mapper.writeValue(**new** File("outputfile.json"), car);

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

}

}

## Inputfile.json:

{

"name": "AUDI",

"carModel": 2014,

"price": 30000,

"foo": "bar",

"tom": "jerry",

"ignoreme1": "ig1",

"ignoreme2": "ig2",

"colors": [

"GRAY",

"BLACK",

"WHITE"

],

"promoDate": 1399647200000

}

## Result:

Ta được file Outputfile.json như sau:

{

"name": "AUDI",

"price": "30000",

"colors": [

"GRAY",

"BLACK",

"WHITE"

],

"promoDate": "09/05/2014",

"carModel": 2014,

"tom": "jerry",

"foo": "bar"

}

Và kết quả xuất ra màn hình console sẽ khác:



## Giải thích:

Cái này để sau đi. Học annotation trước cái đã

# 3. Annotations trong java

<http://o7planning.org/vi/10197/huong-dan-su-dung-java-annotation>

<http://www.vanthuong.com/2016/03/java-annotation.html>

<http://laptrinh.vn/d/4710-annotation-trong-spring-mvc.html>

**Annotation** (Chú thích) được sử dụng để cung cấp thông tin dữ liệu cho mã Java của bạn. Là thông tin dữ liệu, các **Annotation** không trực tiếp ảnh hưởng đến việc thực hiện các mã của bạn, mặc dù một số loại chú thích thực sự có thể được sử dụng cho mục đích đó. **Annotation** đã được thêm vào Java từ Java 5.

## **1. @Deprecated**

Thông báo tự động của Eclipse hoặc của trình biên dịch, thường là thông báo cái method mình sử dụng đã lỗi thời, nên dùng method khác. Eclipse sẽ gạch ngang mehod lỗi thời.

**package** com.jsonannotations;

**import** java.util.Date;

**public** **class** DeprecatedDemo {

/\*\*

\* **@deprecated** replaced by {@link #todo(String,Date)}

\*/

@Deprecated

**public** **void** ~~todoJob~~(String jobName) {

System.***out***.println("Todo " + jobName);

}

**public** **void** todo(String jobName, Date atTime) {

System.***out***.println("Todo " + jobName + " at " + atTime);

}

**public** **void** todoNothing() {

System.***out***.println("Todo Nothing");

}

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

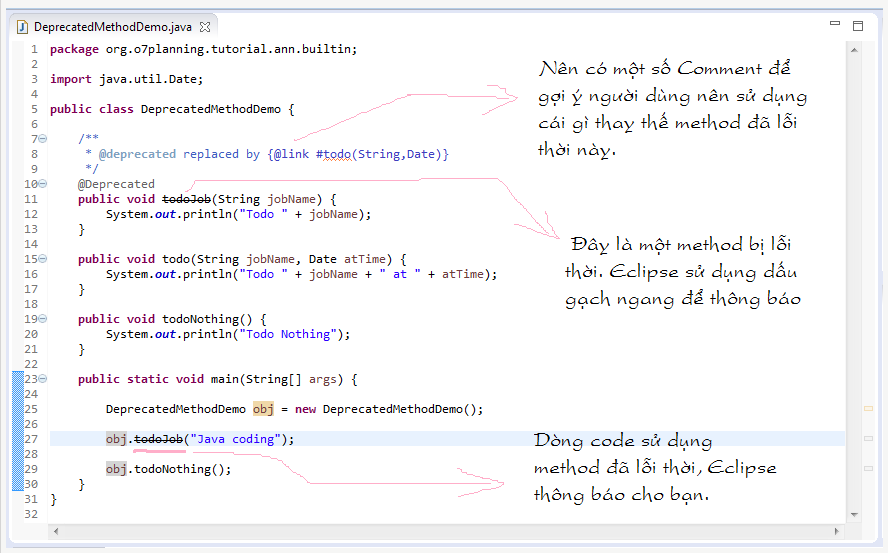
DeprecatedDemo obj = **new** DeprecatedDemo();

obj.~~todoJob~~("Java coding");

obj.todoNothing();

}

}



## 2. @Override

Phương thức ghi đè quen thuộc. Nếu nó là method k phải ghi đè mà để @Override thì sẽ báo lỗi ngay.

## **3. @SuppressWarnings**

Để tắt cảnh báo của trình biên dịch như: thông báo lỗi thời, ép kiểu k an toàn, khai báo biến k sử dụng.