

JSON Data Binding with Jackson

JSON is

JSON (JavaScript Object Notation) is a lightweight data-interchange format. JSON is a text format that is completely language independent, JSON is "self-describing" and easy to understand.

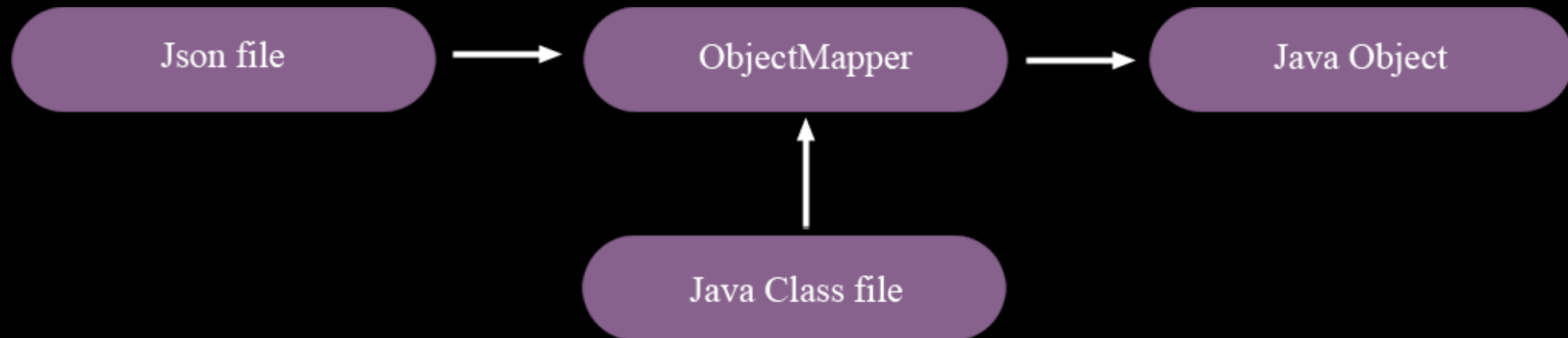
```
Person.json
{
  "id": 12,
  "name": "Alan Smith",
  "age": 41,
  "salary": 250
}
```

Reading JSON – Data Binding in Jackson

Data binding is a JSON processing model that allows for seamless conversion between JSON data and Java objects. With data binding, you create POJOs following JavaBeans convention with properties corresponding to the JSON data. The Jackson ObjectMapper is responsible for mapping the JSON data to the POJOs.

```
public class Person {  
    private int id;  
    private String name;  
    private int age;  
    private BigDecimal salary;  
    /*Getter and Setter Methods*/  
    //  
}
```

When Jackson maps JSON to POJOs, it inspects the setter methods. Jackson, by default, maps a key for the JSON field with the setter method name.



ObjectMapper example

In the ObjectMapperExample class above, we created an ObjectMapper object and called its overloaded readValue() method passing two parameters. We passed a File object representing the JSON file as the first parameter, and Person.class as the target to map the JSON values as the second parameter. The readValue() method returns an Person object populated with the data read from the JSON file.

```
public class ObjectMapperExample {  
    public Person readJsonWithObjectMapper() throws IOException {  
        ObjectMapper objectMapper = new ObjectMapper();  
        Person person = objectMapper.readValue(new File("Person.json"), Person.class);  
        return person;  
    }  
}
```

@JsonProperty

@JsonProperty is a marker annotation to define logical property.

@JsonProperty can be annotated at non-static setter or getter method or non-static object field. The logical property is used in serialization and de-serialization of JSON.

1. Using object field.

```
@JsonProperty("Name")  
private String name;
```

2. Using Getter method.

```
@JsonProperty("Name ")  
public String getName () {  
    return Name;  
}
```

3. Using Setter method.

```
@JsonProperty("Name ")  
public void setName(String name) {  
    this.name = name;  
}
```

Core data types mapping

Simple data binding refers to mapping of JSON to JAVA Core data types.

Sr. No.	JSON Type	Java Type
1	object	LinkedHashMap<String,Object>
2	array	ArrayList<Object>
3	string	String
4	complete number	Integer, Long or BigInteger
5	fractional number	Double / BigDecimal
6	true false	Boolean
7	null	null