

JACKSON - FULL DATA BINDING

http://www.tutorialspoint.com/jackson/jackson_full_data_binding.htm

Copyright © tutorialspoint.com

Full data binding refers to mapping of JSON to any Java Object.

```
//Create an ObjectMapper instance
ObjectMapper mapper = new ObjectMapper();

//map JSON content to Student object
Student student = mapper.readValue(new File("student.json"), Student.class);

//map Student object to JSON content
mapper.writeValue(new File("student.json"), student);
```

Full Data Binding Example

Let us take a simple example to understand full data binding in detail. In the following example, we will map a Java Object directly to JSON and vice versa.

Create a Java class file named JacksonTester in **C:\>Jackson_WORKSPACE**.

File: JacksonTester.java

```
import java.io.File;
import java.io.IOException;

import org.codehaus.jackson.JsonGenerationException;
import org.codehaus.jackson.JsonParseException;
import org.codehaus.jackson.map.JsonMappingException;
import org.codehaus.jackson.map.ObjectMapper;

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

        JacksonTester tester = new JacksonTester();

        try {
            Student student = new Student();
            student.setAge(10);
            student.setName("Mahesh");
            tester.writeJSON(student);

            Student student1 = tester.readJSON();
            System.out.println(student1);

        }
        catch (JsonParseException e) { e.printStackTrace(); }
        catch (JsonMappingException e) { e.printStackTrace(); }
        catch (IOException e) { e.printStackTrace(); }

    }

    private void writeJSON(Student student) throws JsonGenerationException,
    JsonMappingException, IOException{
        ObjectMapper mapper = new ObjectMapper();
        mapper.writeValue(new File("student.json"), student);
    }

    private Student readJSON() throws JsonParseException, JsonMappingException,
    IOException{
        ObjectMapper mapper = new ObjectMapper();
        Student student = mapper.readValue(new File("student.json"), Student.class);
        return student;
    }
}
```

```
class Student {
    private String name;
    private int age;

    public Student(){}

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String toString(){
        return "Student [ name: "+name+", age: "+ age+ " ]";
    }
}
```

Verify the Result

Compile the classes using **javac** compiler as follows –

```
C:\Jackson_WORKSPACE>javac JacksonTester.java
```

Now execute the jacksonTester to see the result.

```
C:\Jackson_WORKSPACE>java JacksonTester
```

Verify the Output –

```
Student [ name: Mahesh, age: 10 ]
```