

JACKSON - DATA BINDING WITH GENERICS

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

Copyright © tutorialspoint.com

In simple data binding, we have used Map class which uses String as key and Object as a value object. Instead, we can have a concrete Java object and type cast it to use it in JSON binding.

Consider the following example with a class UserData, a class to hold user-specific data.

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

File: JacksonTester.java

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

import java.util.Arrays;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

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

public class JacksonTester {
    public static void main(String args[]){
        JacksonTester tester = new JacksonTester();
        try {
            ObjectMapper mapper = new ObjectMapper();

            Map<String, UserData> userDataMap = new HashMap<String, UserData>();
            UserData studentData = new UserData();
            int[] marks = {1, 2, 3};

            Student student = new Student();
            student.setAge(10);
            student.setName("Mahesh");

            // JAVA Object
            studentData.setStudent(student);

            // JAVA String
            studentData.setName("Mahesh Kumar");

            // JAVA Boolean
            studentData.setVerified(Boolean.FALSE);

            // Array
            studentData.setMarks(marks);
            TypeReference ref = new TypeReference<Map<String, UserData>>() { };

            userDataMap.put("studentData1", studentData);
            mapper.writeValue( new File("student.json"), userDataMap );

            // {
            //     "studentData1":
            //     {
            //         "student":
            //         {
            //             "name": "Mahesh",
            //             "age": 10
            //         },
            //         "name": "Mahesh Kumar",
            //         "verified": false,
```

```

        //      "marks":[1,2,3]
        //      }
        // }

        userDataMap = mapper.readValue(new File("student.json"), ref);

        System.out.println( userDataMap.get("studentData1").getStudent() );
        System.out.println( userDataMap.get("studentData1").getName() );
        System.out.println( userDataMap.get("studentData1").getVerified() );
        System.out.println( Arrays.toString(userDataMap.get("studentData1").getMarks()) );

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

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+ " ]";
    }
}

class UserData {
    private Student student;
    private String name;
    private Boolean verified;
    private int[] marks;

    public UserData(){}

    public Student getStudent() {
        return student;
    }

    public void setStudent(Student student) {
        this.student = student;
    }

    public String getName() {
        return name;
    }

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

    public Boolean getVerified() {

```

```
        return verified;
    }

    public void setVerified(Boolean verified) {
        this.verified = verified;
    }
    public int[] getMarks() {
        return marks;
    }

    public void setMarks(int[] marks) {
        this.marks = marks;
    }
}
```

Verify the Result

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

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

Now run the jacksonTester to see the result –

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

Verify the Output –

```
Student [ name: Mahesh, age: 10 ]
Mahesh Kumar
false
[1, 2, 3]
```