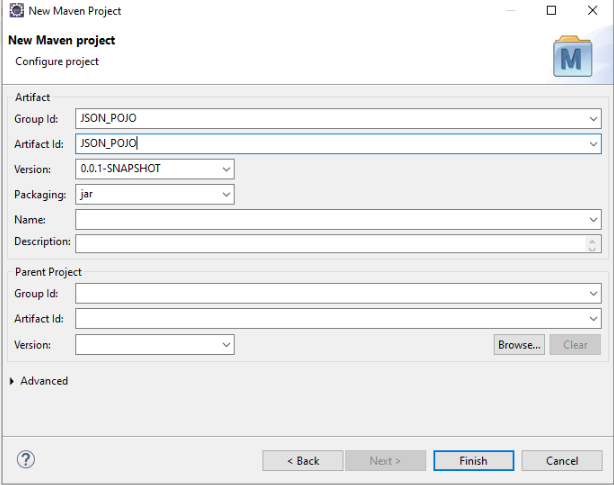
**SBA 8**

**Serialization & Deserialization**

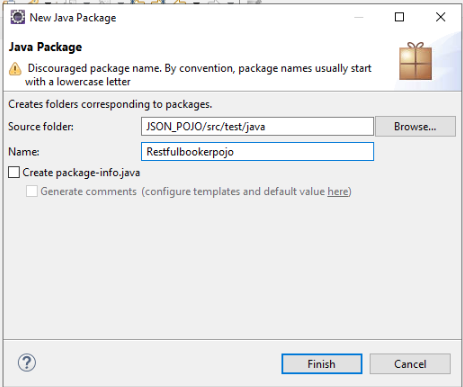
**Serialization** 🡪 converting POJO to json (here we are converting Employee class object to JSON representation or object.

**Deserialization 🡪** Reverse of serialization (convert a JSON object to Employee class object).

* Create a maven project.



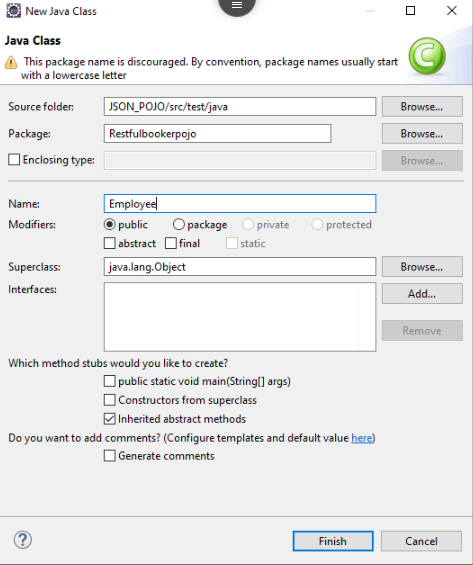
* Create a Java Package.



* Create two java classes

Employee class.

EmployeeSerializationDeserialization class.



* **Employee class**

package Restfulbookerpojo;

public class Employee {

// private variables or data members of pojo class

private boolean married;

private String firstName;

private String lastName;

private String gender;

private int age;

private double salary;

public Employee () {

}

public Employee(String firstName, String lastName, String gender, int age, double salary, boolean married) {

super();

this.firstName = firstName;

this.lastName = lastName;

this.gender = gender;

this.age = age;

this.salary = salary;

this.married = married;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public String getGender() {

return gender;

}

public void setGender(String gender) {

this.gender = gender;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

this.salary = salary;

}

public void setMarried(boolean married) {

this.married = married;

}

public boolean getMarried() {

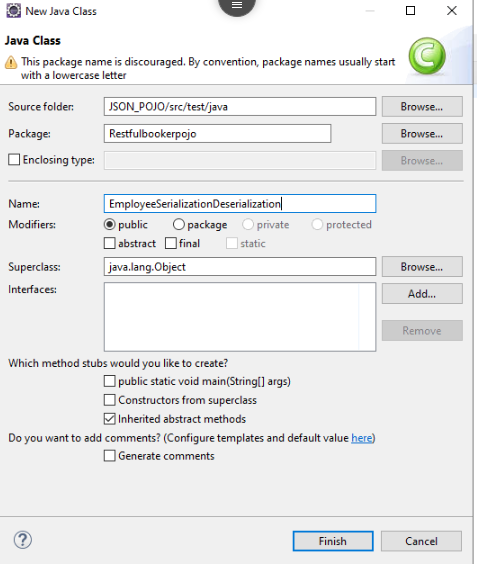
// TODO Auto-generated method stub

return this.married;

}

}

* **EmployeeSerializationDeserialization**



package Restfulbookerpojo;

import java.io.IOException;

import org.testng.annotations.Test;

import com.fasterxml.jackson.core.JsonProcessingException;

import com.fasterxml.jackson.databind.ObjectMapper;

public class EmployeeSerializationDeserialization {

@Test

public void createEmployeeJSONFromEmployeePOJOClass() throws JsonProcessingException

{

// Just create an object of Pojo class

Employee employee = new Employee("Amod","Mahajan","Male",29,3434343,false);

// Converting a Java class object to a JSON payload as string

ObjectMapper objectMapper = new ObjectMapper();

String employeeJson = objectMapper.writerWithDefaultPrettyPrinter().writeValueAsString(employee);

System.out.println(employeeJson);

}

@Test

public void getPojoFromEmployeeObject() throws JsonProcessingException,IOException

{

// Just create an object of Pojo class

Employee employee = new Employee();

// Set value as you wish

employee.setFirstName("Amod");

employee.setLastName("Mahajan");

employee.setAge(29);

employee.setGender("Male");

employee.setSalary(3434343);

employee.setMarried(false);

// Converting a Java class object to a JSON payload as string

ObjectMapper objectMapper = new ObjectMapper();

String employeeJson = objectMapper.writerWithDefaultPrettyPrinter().writeValueAsString(employee);

// Converting EMployee json string to Employee class object

Employee employee2 = objectMapper.readValue(employeeJson, Employee.class);

System.out.println("First Name of employee : "+employee2.getFirstName());

System.out.println("Last Name of employee : "+employee2.getLastName());

System.out.println("Age of employee : "+employee2.getAge());

System.out.println("Gender of employee : "+employee2.getGender());

System.out.println("Salary of employee : "+employee2.getSalary());

System.out.println("Marital status of employee : "+employee2.getMarried());

}

}

* **Add the dependencies in the POM file**

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>JSON\_POJO</groupId>

<artifactId>JSON\_POJO</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured -->

<dependency>

<groupId>io.rest-assured</groupId>

<artifactId>rest-assured</artifactId>

<version>4.4.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>io.rest-assured</groupId>

<artifactId>json-path</artifactId>

<version>4.2.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>io.rest-assured</groupId>

<artifactId>xml-path</artifactId>

<version>4.2.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>io.rest-assured</groupId>

<artifactId>json-schema-validator</artifactId>

<version>4.2.0</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/org.testng/testng -->

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.4.0</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>com.googlecode.json-simple</groupId>

<artifactId>json-simple</artifactId>

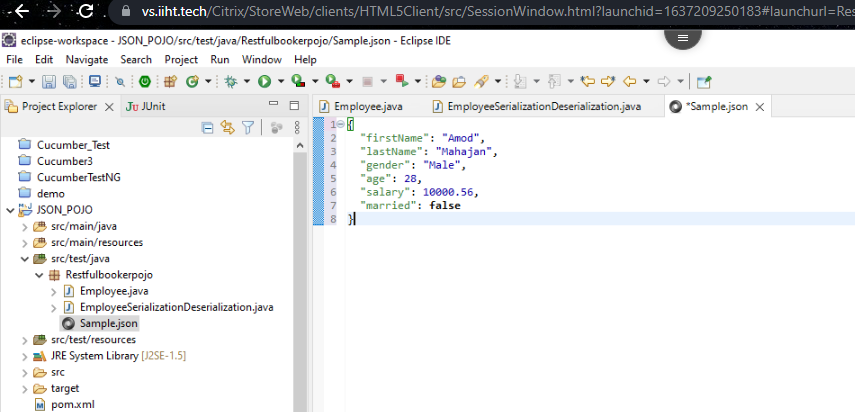
<version>1.1.1</version>

</dependency>

</dependencies>

</project>

* **Then create a json file (Sample.json).**



* **Serialization**

@Test

public void createEmployeeJSONFromEmployeePOJOClass() throws JsonProcessingException

{

// Just create an object of Pojo class

Employee employee = new Employee("Amod","Mahajan","Male",29,3434343,false);

// Converting a Java class object to a JSON payload as string

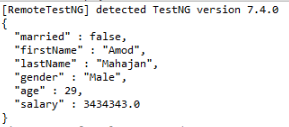
ObjectMapper objectMapper = new ObjectMapper();

String employeeJson = objectMapper.writerWithDefaultPrettyPrinter().writeValueAsString(employee);

System.out.println(employeeJson);

}

* **output**



* **Deserialization**

@Test

public void getPojoFromEmployeeObject() throws JsonProcessingException,IOException

{

// Just create an object of Pojo class

Employee employee = new Employee();

// Set value as you wish

employee.setFirstName("Amod");

employee.setLastName("Mahajan");

employee.setAge(29);

employee.setGender("Male");

employee.setSalary(3434343);

employee.setMarried(false);

// Converting a Java class object to a JSON payload as string

ObjectMapper objectMapper = new ObjectMapper();

String employeeJson = objectMapper.writerWithDefaultPrettyPrinter().writeValueAsString(employee);

// Converting EMployee json string to Employee class object

Employee employee2 = objectMapper.readValue(employeeJson, Employee.class);

System.out.println("First Name of employee : "+employee2.getFirstName());

System.out.println("Last Name of employee : "+employee2.getLastName());

System.out.println("Age of employee : "+employee2.getAge());

System.out.println("Gender of employee : "+employee2.getGender());

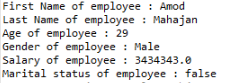
System.out.println("Salary of employee : "+employee2.getSalary());

System.out.println("Marital status of employee : "+employee2.getMarried());

}

}

**Output**



* **Final Output**

