Java Serialization

BY CHAITANYA SINGH | FILED UNDER: JAVA TUTORIALS

Here we are gonna discuss how to serialize and de-serialize an object and what is the use of it.

What is Java Serialization?

Serialization is a mechanism to convert an object into stream of bytes so that it can be written into a file, transported through a network or stored into database. De-serialization is just a vice versa. In simple words serialization is converting an object to stream of bytes and de-serialization is rebuilding the object from stream of bytes. Java Serialiation API provides the features to perform seralization & de-serialization. A class must implement java.io.Serializable interface to be eligible for serialization.

Lets take an example to understand the concepts better:

Example
This class implements Serializable interface which means it can be serialized. All the fields of this class can be written to a file after being converted to stream of bytes, except those fields that are declared **transient**. In the below example we have two transient fields, these fields will not take part in serialization.
Student.java

```
public class Student implements java.io.Serializable{
  private int stuRollNum;
  private int stuAge;
  private String stuName;
  private transient String stuAddress;
  private transient int stuHeight;

  public Student(int roll, int age, String name,
  String address, int height) {
    this.stuRollNum = roll;
    this.stuAge = age;
    this.stuName = name;
    this.stuAddress = address;
    this.stuHeight = height;
  }

  public int getStuRollNum() {
    return stuRollNum;
  }
  public void setStuRollNum(int stuRollNum) {
    this.stuRollNum = stuRollNum;
  }
  public int getStuAge() {
    return stuAge;
  }
  public void setStuAge(int stuAge) {
    this.stuAge = stuAge;
  }
  public String getStuName() {
    return stuName;
  }
  public void setStuName(String stuName) {
    this.stuName = stuName;
  }
  public String getStuAddress() {
    return stuAddress;
  }
  public void setStuAddress(String stuAddress) {
    this.stuAddress = stuAddress;
  }
  public int getStuHeight() {
    return stuHeight;
  }
  public void setStuHeight(int stuHeight) {
    this.stuHeight = stuHeight;
  }
}
```

Serialization of Object

This class is writing an object of Student class to the Student.ser file. We are using FileOutputStream and ObjectOutputStream to write the object to File.

Note: As per the best practices of Java Serialization, the file name should have .ser extension.

```
import java.io.FileOutputStream;
import java.io.ObjectOutputStream;
import java.io.IOException;
public class SendClass
{
  public static void main(String args[])
  {
    Student obj = new Student(101, 25, "Chaitanya", "Agra", 6);
    try{
      FileOutputStream fos = new FileOutputStream("Student.ser");
      ObjectOutputStream oos = new ObjectOutputStream(fos);
      oos.writeObject(obj);
      oos.close();
      fos.close();
      System.out.println("Serialzation Done!!");
   }catch(IOException ioe){
      System.out.println(ioe);
    }
  }
}
```

Output:

```
Serialzation Done!!
```

De-serialization of Object

This class would rebuilt the object of Student class after reading the stream of bytes from the file. Observe the output of this class, student address and student height fields are having null & 0 values consecutively. This is because these fields were declared transient in the Student class.

```
import java.io.FileInputStream;
import java.io.ObjectInputStream;
import java.io.IOException;
public class AcceptClass {

 public static void main(String args[])
 {
    Student o=null;
    try{
      FileInputStream fis = new FileInputStream("Student.ser");
      ObjectInputStream ois = new ObjectInputStream(fis);
      o = (Student)ois.readObject();
      ois.close();
      fis.close();
    }
    catch(IOException ioe)
    {
       ioe.printStackTrace();
       return;
    }catch(ClassNotFoundException cnfe)
     {
       System.out.println("Student Class is not found.");
       cnfe.printStackTrace();
       return;
     }
    System.out.println("Student Name:"+o.getStuName());
    System.out.println("Student Age:"+o.getStuAge());
    System.out.println("Student Roll No:"+o.getStuRollNum());
    System.out.println("Student Address:"+o.getStuAddress());
    System.out.println("Student Height:"+o.getStuHeight());
 }
}
```

Output:

```
Student Name:Chaitanya
Student Age:25
Student Roll No:101
Student Address:null
Student Height:0
```