BenchResources.Net

Java, Collection, JDBC, Spring, Web Services, Maven, Android, Oracle SOA-OSB & Open Source

JAVA ~ TOOLS V HOME SPRING ~ WEB SERVICES ~ ORACLE SOA V CLOUD ~ ANDROID INTERVIEW Q **JOBS**

How to stop Serialization in Java

O November 12, 2016 A SJ 🗁 Serialization 🔘 0

In this article, we will discuss how to stop serialization to happen for a serializable class.

One would wonder why we need to stop a serialization that too for a class which is marked as serializable by implementing java.io.Serializable interface

I had same thinking, after interviewer asked me this question?

Let us move on and discuss how to stop serialization

Override writeObject() method and throw IOException and similarly we can write logic for readObject() method for de-serialization process

SEARCH TUTORIALS

SEARCH ...

SUBSCRIBE VIA EMAIL

Join 194 other subscribers

Email Address

SUBSCRIBE

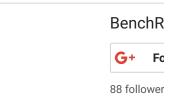
POPULAR ARTICLES

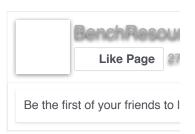
JDBC: An example to connect MS Access database in Java 8

Customer.java

```
package in.bench.resources.stop.serialization;
import java.io.IOException;
import java.io.ObjectOutputStream;
import java.io.Serializable;
class Customer implements Serializable {
        // member variables for Customer
        transient int customerId;
        String customerName;
        transient String customerSSN;
        // 3-arg parameterized constructor for
Customer
        public Customer(int customerId, String
customerName,
                        String customerSSN) {
                super();
                this.customerId = customerId;
                this.customerName =
customerName;
                this.customerSSN = customerSSN;
        }
        /**
         * this method invoked automatically
during serialization process
         * @param objectOutputStream
         * @throws Exception
         * /
        private void
writeObject(ObjectOutputStream
objectOutputStream)
                        throws Exception {
                // don't provide implementation
details here
                throw new IOException("No
serialization not allowed");
```

Java JDBC: An example to connect MS Access database
Spring JDBC: An example on JdbcTemplate using
Annotation
Oracle OSB 12c: Service
Callout and Routing Table example
Oracle OSB 12c: Hello
World service with both
Business and Proxy
Service





Test class for serialization

SerializeCustomer.java

```
package in.bench.resources.stop.serialization;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectOutputStream;
public class SerializeCustomer {
        public static void main(String[] args) {
                // create a customer object
using 3-arg parametrized constructor
                Customer customer = new
Customer(101, "SJ", "SSN-101919");
                // creating output stream
variables
                FileOutputStream fos = null;
                ObjectOutputStream oos = null;
                try {
                        // for writing or saving
binary data
                        fos = new
FileOutputStream("Customer.ser");
                        // converting java-
object to binary-format
                        oos = new
ObjectOutputStream(fos);
                        // writing or saving
customer object's value to stream
oos.writeObject(customer);
```

Output:

```
java.io.IOException: No serialization not
allowed
        at.
in.bench.resources.stop.serialization.Customer.w
riteObject(
Customer.java:33)
        at
sun.reflect.NativeMethodAccessorImpl.invoke0(
Native Method)
sun.reflect.NativeMethodAccessorImpl.invoke(
NativeMethodAccessorImpl.java:62)
        at
sun.reflect.DelegatingMethodAccessorImpl.invoke(
DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(
Method.java:498)
        at
java.io.ObjectStreamClass.invokeWriteObject(
ObjectStreamClass.java:1028)
java.io.ObjectOutputStream.writeSerialData(
ObjectOutputStream.java:1496)
        at
java.io.ObjectOutputStream.writeOrdinaryObject(
ObjectOutputStream.java:1432)
        at
```

```
java.io.ObjectOutputStream.writeObject0(
ObjectOutputStream.java:1178)
          at
java.io.ObjectOutputStream.writeObject(
ObjectOutputStream.java:348)
          at
in.bench.resources.stop.serialization.SerializeC
ustomer.main(
SerializeCustomer.java:27)
```

This example is to test whether we know about internal details of serialization and de-serialization process

- Controlling serialization by overriding writeObject() method
- Controlling de-serialization by overriding readObject() method

References:

https://docs.oracle.com/javase/7/docs/api/java/io/Serializable.html

https://docs.oracle.com/javase/7/docs/platform/serialization/spec/serial-arch.html

https://docs.oracle.com/javase/7/docs/api/java/io/ObjectOutputStream.html

https://docs.oracle.com/javase/7/docs/api/java/io/ObjectInput Stream.html

https://docs.oracle.com/javase/7/docs/api/java/io/FileOutputS tream.html

https://docs.oracle.com/javase/7/docs/api/java/io/FileInputStream.html

http://docs.oracle.com/javase/specs/jls/se7/html/jls-8.html#jls-8.3.1.3

Read Also:

Java Serialization and De-Serialization Tutorial Index

- Serialization and De-Serialization in Java
- Serializable interface
- Transient keyword with Serialization in Java
- Transient keyword with static variable in Serialization
- Transient keyword with final variable in Serialization
- Serializing a variable with transient modifier or keyword
- Order of Serialization and De-Serialization
- Serialization with Aggregation
- Serialization with Inheritance
- Externalizable interface with example
- Serializable v/s Externalizable
- Importance of SerialVersionUID in Serialization
- Singleton Design pattern with Serialization
- How to construct a singleton class in a multi-threaded environment in Java
- How to serialize and de-serialize ArrayList in Java

Happy Coding!! Happy Learning!!



How to construct a singleton class in a multi-threaded environment in Java

Singleton Design pattern with Serialization

Related Posts:

- 1 Importance of SerialVersionUID in Serialization
- 2. Singleton Design pattern with Serialization
- 3. Serialization interview question and answer in Java
- 4. Serialization with Aggregation

DESERIALIZATION JAVA SERIALIZABLE

SERIALIZATION SERIALVERSIONUID



Start the discussion...

Be the first to comment.

ALSO ON BENCHRESOURCES.NET

5 important keywords in Java **Exception handling**

1 comment • 2 years ago

Saurabh Naik — Various possible combinations of try catch finally 1) try{} catch(x x)

Java features version-wise

2 comments • a year ago Waseem Siddiqi — Thanks

Java: StringBuffer length() method

4 comments • a year ago

BenchResources.Net - Glad that you found it very useful!!

Oracle OSB 12c: Hello World service with both Business and

12 comments • 2 years ago

Neerav Chaudhary — Hi, I am new to SOA, need help in terms of what is required to expose a



Proudly powered by Tuto WordPress theme from MH Themes