BenchResources.Net

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

Serializable v/s Externalizable

November 9, 2016 SJ Serialization 0

In this article, we will discuss difference between ***Serialization*** and ***Externalization*** in detail i.e.; Serializable v/s Externalizable

The ***main difference between these serializing concepts*** is

- Serialization helps in ***serializing complete object*** whereas with Externalization we can ***serialize either complete or partial object***
- ***Performance-wise*** Externalization is far ***better*** than Serialization as only ***partial required member variables*** of an ***object*** is considered while serializing to file storage
- And in ***serialization complete object*** is ***compulsorily*** considered while serializing to file storage, even if some of the ***member variables not required***

Let's move and differentiate these two serializing principle in more detail in ***tabular form;***

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

Serializable v/s Externalizable:

Serializable	Externalizable
Serializable is a marker interface which doesn't contain any methods and JVM provides special ability during serialization process	Externalizable is a sub-interface of Serializable interface and contains 2 methods viz.; readExternal() and writeExternal()
During Serialization process, all member variables of an object is serialized (even if some of the variables not required to be serialized)	But in Externalization, programmer has to provide serialization logic
That's why, it is referred as default serialization	This is referred as custom serialization, as programmer has to write custom logic for serialization to happen
From above stated points, it is clear that JVM takes the complete control over serialization process	Programmer has complete control over serialization process (to write custom logic for the required variables to be serialized)
Performance-wise, Serializable is relatively low as complete object need to be serialized, even if we require only partial object	Performance is high in extenalizable, as programmer design what all required variable need to be serialized
Doesn't require any public no-argument constructor for serializable	Public no-argument constructor is very must in externalizable, otherwise InvalidClassException is thrownThis is mainly required during readExternal() method; that is while restoring object back to heap memory from file storage
For variable that needn't to be serialized use transient modifier (but still its default	Variable with transient modifier not required; as programmer can write custom logic to ignore

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

value is stored into file)Transient modifier play a very important role in serializable	those variables which is not requiredSo, transient modifier doesn't play any important role in externaizable
This is the best suit; when whole or complete object required to be serialized to the file storage	This is the best suit; when partial object or few of the member variables of an object need to be serialized to the file storage

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

- Importance of SerialVersionUID in Serialization
- Singleton Design pattern with Serialization
- How to stop Serialization in Java
- 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 !!

« Importance of SerialVersionUID in Serialization

Externalizable interface with example »

Related Posts:

1. **Serialization and De-Serialization in Java**
2. **Serializable interface**
3. **Serialization with Inheritance**
4. **Serializing a variable with transient modifier or keyword**

EXTERNALIZABLE EXTERNALIZATION JAVA
SERIALIZABLE SERIALIZATION

0 Comments BenchResources.Net

Ava

♡ Recommend Share Sort by Best

Start the discussion…

Be the first to comment.

ALSO ON **BENCHRESOURCES.NET**

final v/s finally v/s finalize

1 comment • 2 years ago

Sachin Kamble — Hi, Thanks for your effort to make this so simple to undestand.

Interview Question and Answers on final keyword in

2 comments • 2 years ago

BenchResources.Net — Anurag,You are most welcome !!There are various other Java

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

18 comments • 2 years ago

Daroga Jee — then tell us.. Where to go and connect to MS-Excel..

Java JDBC: An example to connect MS Access database

2 comments • 2 years ago

BenchResources.Net — Buket,You need to include hsqldb-2.4.0.jar in your project