

Job Interview Questions Series

HIBERNATE, SPRING & STRUTS

**INTERVIEW QUESTIONS
YOU'LL MOST LIKELY BE ASKED**

378

Interview Questions



VIBRANT
PUBLISHERS

Hibernate, Spring & Struts

Interview Questions

You'll Most Likely Be Asked

© 2021, By Vibrant Publishers, USA. All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior permission of the publisher.

ISBN-10: 1-946383-02-3

ISBN-13: 978-1-946383-02-0

Library of Congress Control Number: 2011920203

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. The author has made every effort in the preparation of this book to ensure the accuracy of the information. However, information in this book is sold without warranty either expressed or implied. The Author or the Publisher will not be liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Vibrant Publishers books are available at special quantity discount for sales promotions, or for use in corporate training programs. For more information please write to **bulkorders@vibrantpublishers.com**

Please email feedback / corrections (technical, grammatical or spelling) to **spellerrors@vibrantpublishers.com**

To access the complete catalogue of Vibrant Publishers, visit **www.vibrantpublishers.com**

Table of Contents

Hibernate

| | | |
|------------------|-----------------------------------|----|
| chapter 1 | Hibernate Interfaces | 9 |
| chapter 2 | Hibernate Configuration | 13 |
| chapter 3 | Criteria Queries | 21 |
| chapter 4 | Persistent Classes | 25 |
| chapter 5 | Object States | 29 |
| chapter 6 | O/R Mapping | 35 |
| chapter 7 | Hibernate Inheritance | 41 |
| chapter 8 | Hibernate Caches | 45 |
| chapter 9 | Hibernate Interceptor and Filters | 55 |

Spring

| | | |
|-------------------|------------------------|----|
| chapter 10 | Spring Modules | 61 |
| chapter 11 | IoC Container | 65 |
| chapter 12 | Beans | 69 |
| chapter 13 | Dependency Injection | 73 |
| chapter 14 | Importing XML Resource | 77 |
| chapter 15 | Idref Element | 81 |
| chapter 16 | Bean Collaborators | 87 |
| chapter 17 | Collections | 91 |

| | | | |
|---------|-----------|----------------------------------|-----|
| chapter | 18 | Depends-on Attribute | 97 |
| chapter | 19 | Lazy Initialized Bean | 101 |
| chapter | 20 | Autowiring | 105 |
| chapter | 21 | Bean Scopes | 109 |
| chapter | 22 | Bean Lifecycle Methods | 115 |
| chapter | 23 | Inheritance (Normal Inheritance) | 121 |
| chapter | 24 | Inheritance with Abstract | 125 |
| chapter | 25 | Annotations Based Configuration | 129 |
| chapter | 26 | Application Context | 133 |
| chapter | 27 | Spring and Hibernate Integration | 137 |
| chapter | 28 | Spring AOP | 141 |

Struts

| | | | |
|---------|-----------|-------------------------------------|-----|
| chapter | 29 | Configuration | 153 |
| chapter | 30 | Main Classes | 161 |
| chapter | 31 | Exception Handling | 177 |
| chapter | 32 | Validation and Validator Framework | 183 |
| chapter | 33 | Logic Tags | 191 |
| chapter | 34 | Internationalization / Localization | 195 |
| chapter | 35 | Integrate with other Frameworks | 199 |
| chapter | 36 | Action and ActionForms | 203 |
| | | HR Questions | 211 |
| | | Index | 241 |

Dear Reader,

Thank you for purchasing **Hibernate, Spring & Struts Interview Questions You'll Most Likely Be Asked**. We are committed to publishing books that are content-rich, concise and approachable enabling more readers to read and make the fullest use of them. We hope this book provides the most enriching learning experience as you prepare for your interview.

Should you have any questions or suggestions, feel free to email us at reachus@vibrantpublishers.com

Thanks again for your purchase. Good luck with your interview!

- Vibrant Publishers Team



This page is intentionally left blank

Hibernate, **Spring** & **Struts** Interview Questions

Review these typical interview questions and think about how you would answer them. Read the answers listed; you will find best possible answers along with strategies and suggestions.

This page is intentionally left blank.

Hibernate

This page is intentionally left blank.

Chapter 1

Hibernate Interfaces

1: Explain Database Transaction management using Transaction Interface.

Answer:

In Hibernate, every transaction is handled by a session object. It controls the transactions and hides it from the outer world. To make sure the transaction completes successfully and is committed upon success and is rolled back upon failure, the entire transaction happens within a try – catch block. Within the try block, the session begins the transaction and continues with the processes and finally commits it. If anything goes wrong, the try block throws an error and under the exception handling part, a rollback is issued for the transaction. And finally, the session is closed. This is an efficient way of handling database transactions using Hibernate.

2: What are the core interfaces available in hibernate?

Answer:

The core interfaces available in hibernate are:

- a) **Session**: used to store and retrieve the objects and they are not thread-safe
- b) **SessionFactory**: one object being created per application
- c) **Criteria**: used to provide a conditional search over resultset
- d) **Query**: obtained by invoking createQuery() method of Session
- e) **Configuration**: used to specify the hibernate mapping file's location
- f) **Transaction**: used to perform several database operations

3: What is SessionFactory? Is it thread-safe object?

Answer:

SessionFactory is an interface creating session instances. Threads servicing the client requests obtain the session instances from the SessionFactory. SessionFactory is created only once for each application. More than one thread can access the SessionFactory concurrently since it is thread-safe object.

4: How will you create SessionFactory object?

Answer:

SessionFactory object can be created using buildSessionFactory() method.

```
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
```

```
SessionFactory sf = new  
Configuration().configure().buildSessionFactory();
```

5: What is Session? Is it a thread safe object?**Answer:**

Session is obtained from SessionFactory and not thread-safe, meaning that it cannot be shared between threads. It is a single unit of work with the database. To avoid creating multiple sessions, ThreadLocal can be used. Get current session from the ThreadLocal object using get(). If no sessions are available, get a new session from SessionFactory object using openSession() method and set it in ThreadLocal.

6: Explain about Criteria in hibernate.**Answer:**

- a) Criteria in hibernate are used to create dynamic queries to execute
- b) Criteria instances are obtained from session object using createQuery() method
- c) Criteria mycriteria = session1.createCriteria(Empl.class)

This page is intentionally left blank.

Chapter 2

Hibernate Configuration

7: How do you configure hibernate?

Answer:

Hibernate could be configured in the following methods:

- a) Create hibernate configuration file “hibernate.cfg.xml” to provide hibernate configuration properties
- b) Create hibernate mapping file say “emp.hbm.xml” to provide hibernate mapping details i.e., mapping tables and java objects
- c) Configuration class uses these two files to create SessionFactory which in turn creates session instances

8: What are the important tags of hibernate configuration file (hibernate.cfg.xml)?

Answer:

The important tags of hibernate configuration file are:


```

<! DOCTYPE ... >
<hibernate-configuration>
  <session-factory>
    <property name="..."> ... </property>
    ...
  </session-factory>
</hibernate-configuration>

```

- a) **DTD: doctype**
- b) **Configuration of JDBC connection:** driver class, url, username, password
- c) **Dialect:** specify the type of sql to be generated
- d) **Size of Connection Pool**
- e) **Specify hbm2ddl.auto:** automatic generation of database schema
- f) **Map hbm.xml files:** include Hibernate Mapping files

9: Why column attribute is required if the property name is "date"?

Answer:

If the column attribute is not given hibernate will map the column-name as the property name. However if "date" is given as property name, column should be explicitly given since date is a keyword.

```
<property name="date" column="created_date"/>
```

10: How will you get hibernate statistics?

Answer:

Hibernate statistics could be obtained by:

- a) using `getStatistics()` method of `SessionFactory`
- b) `SessionFactory.getStatistics();`

11: How will you make generated sql to be displayed in console?

Answer:

Generated sql could be displayed in console by setting “show_sql” property to “true” in the hibernate configuration file.

```
<property name = "show_sql" > true </property>
```

12: How are the columns of the database mapped with the java class properties in hibernate?

Answer:

The columns of the database are mapped with the java class properties in hibernate as follows:

- a) Write POJO (getters and setters) class
- b) Create hibernate mapping file “hibernate.cfg.xml” where mapping between table columns and the class properties are specified

```
<hibernate-mapping>
    <class name="empl" table="empl_tabl">
        <property name="name" column="empl_name">
        <property name="age" column="empl_age"/>
        <many-to-one name="dept" cascade="all"
        column="dept_Id"/>
    </class>
</hibernate-mapping>
```

13: If you want to insert data for few columns into a large table with hundreds of columns, hibernate will generate the insert sql query at run time containing all the table columns which will create performance issue. How will you make the hibernate to generate dynamically generated sql queries containing only the necessary columns? For instance, it should not include the null columns / property values.

Answer:

We can make the hibernate to generate dynamically generated sql queries using **dynamic-insert="true"** attribute in the class mapping. Default option is "false".

```
<class name=".." table="UserDetails" catalog="ks" dynamic-  
insert="true" >
```

...

```
</class>
```

14: What is the flow of hibernate communication with database?

Answer:

The flow of hibernate communication with database is as follows:

- a) First, load hibernate configuration file and create configuration object. Using configuration object load all hibernate mapping files.
- b) Create SessionFactory from the Configuration object.
- c) Create Session from the SessionFactory object.
- d) Create the HQL query.
- e) Run the query which retrieves list of java objects.

15: How will you configure Sequence generated primary key?

Answer:

We can configure sequence generated primary key by using <generator> tag.

```
<id column="emp_Id" name="empId" type="java.lang.Long">  
    <generator class="sequence">  
        <param name="myseq"> emp_id_seq </param>  
    </generator>  
</id>
```

16: How will you change one relational database to another database without code changes?**Answer:**

We can change a relational database to another using hibernate SQL dialect. Hibernate will generate sql queries based on the dialect defined in the configuration file.

```
<property name="dialect"> org.hibernate.dialect.MySQLDialect  
</property>
```

17: What is dynamic-insert and dynamic-update option in the class mapping?**Answer:**

Dynamic-insert and dynamic-update option in the class-mapping are:

- a) **dynamic-insert="true/false"**: Used to decide whether to include the columns having null properties, in the dynamically generated sql INSERT statement
- b) **dynamic-update="true/false"**: Used to decide whether to include the null columns, in the dynamically generated sql

UPDATE statement

18: How will you configure Hibernate to access the instance variables directly without using setter method?

Answer:

We can configure hibernate to access the instance variables directly without using setter method as follows:

- a) Using `access="field"` attribute in the `<property>` tag (mapping file)
- b) `<property name="uname" access="field"/>`

19: What is Automatic Dirty checking in hibernate?

Answer:

Automatic Dirty checking in hibernate:

- a) means that Hibernate persists the data (or updates the database) automatically if the state of the object is modified inside a transaction when the transaction is committed and the session is opened
- b) means no need for explicit update statemen

20: Write down a sample code for Automatic Dirty checking.

Answer:

Automatic Dirty checking Sample Code:

```
Session session1 = sessionFactory1.openSession();
Transaction tx1 = session1.beginTransaction();
User user = (User)session1.get(User.class, 5L);
user.setUname("Shobhana");
tx1.commit(); //data is synchronized with database
```

```
session1.close();
```

21: How hibernate is database independent and what are the changes required?**Answer:**

Hibernate is database independent as by changing the below properties, databases can be replaced without much changes.

```
<property name="hibernate.dialect">  
    org.hibernate.dialect.Oracle9Dialect  
</property>
```

```
<property name="hibernate.connection.driver_class">  
    oracle.jdbc.driver.OracleDriver  
</property>
```

22: How will you include hibernate mapping file in the hibernate configuration file?**Answer:**

We can include hibernate mapping file in the hibernate configuration file using <mapping> tag and resource attribute.

```
<mapping resource="user.hbm.xml" />  
<mapping resource="addr.hbm.xml" />
```

This page is intentionally left blank.

Job Interview Questions Series

HIBERNATE, SPRING & STRUTS

INTERVIEW QUESTIONS
YOU'LL MOST LIKELY BE ASKED



Hibernate, Spring & Struts Interview Questions You'll Most Likely Be Asked is a perfect companion to stand ahead above the rest in today's competitive job market. Rather than going through comprehensive, textbook-sized reference guides, this book includes only the information required immediately for job search to build an IT career. This book puts the interviewee in the driver's seat and helps them steer their way to impress the interviewer.

The following is included in this book:

- 300 Hibernate, Spring & Struts Interview Questions, Answers and proven strategies for getting hired as an IT professional
- Dozens of examples to respond to interview questions
- 78 HR Questions with Answers and proven strategies to give specific, impressive, answers that help nail the interviews
- 2 Aptitude Tests download available on www.vibrantpublishers.com



www.vibrantpublishers.com

