Subscribe to Download Java Design Patterns eBook

Full name

name@example.com

DOWNLOAD NOW

Spring Hibernate Integration Example Tutorial (Spring 4 + Hibernate 3 and Hibernate 4)

APRIL 2, 2018 BY PANKAJ — 69 COMMENTS

Spring is one of the most used Java EE Framework and **Hibernate** is the most popular ORM framework. That's why **Spring Hibernate** combination is used a lot in enterprise applications. Recently I have written a lot for Spring Tutorial and Hibernate Tutorial, so a post for **spring hibernate integration** was due for long time.

Table of Contents [hide]

Spring Hibernate

Today in this tutorial, we will use Spring 4 and integrate it with Hibernate 3 and then update the same project to use Hibernate 4. Since there are a lot of versions for Spring and Hibernate both and Spring ORM artifact supports both Hibernate 3 and Hibernate 4, it's good that I list all the dependencies I have used in my project.

Note that I have noticed that all spring and hibernate versions are not compatible, below versions have worked for me so I think they are compatible. If you are using some other versions and getting `java.lang.NoClassDefFoundError`, then it means that they are not compatible. Mostly it's because Hibernate classes are moved from one package to another causing this error. For example `org.hibernate.engine.FilterDefinition` class is moved to `org.hibernate.engine.spi.FilterDefinition` in latest hibernate versions.

- Spring Framework Version: 4.0.3.RELEASE
- Hibernate Core and Hibernate EntityManager Version: 3.6.9.Final and 4.3.5.Final
- Spring ORM Version: 4.0.3.RELEASE

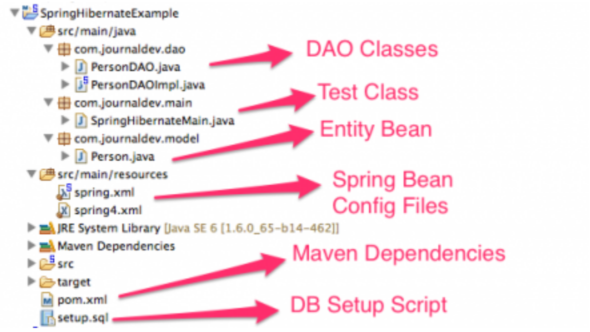
Database Setup

I am using MySQL database for my project, so below setup.sql script will create the necessary table for this example.

```
CREATE TABLE `Person` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(20) NOT NULL DEFAULT '',
  `country` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;
commit;
```

Spring Hibernate Integration Example Project Structure

Below image shows the final project structure, we will go through each of the components one by one.



Maven Dependencies

We will first look into our pom.xml file for all the required dependencies and their versions.

```
<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
http://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>
        <groupId>org.springframework.samples</groupId>
        <artifactId>SpringHibernateExample</artifactId>
        <version>0.0.1-SNAPSHOT</version>
        <properties>

                <!-- Generic properties -->
                <java.version>1.6</java.version>
                <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
                <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

                <!-- Spring -->
                <spring-framework.version>4.0.3.RELEASE</spring-framework.version>

                <!-- Hibernate / JPA -->
```

Important Dependencies for Spring and Hibernate Integration Project are:

- spring-context and spring-tx for core Spring functionalities. Notice that I am using version 4.0.3.RELEASE.
- spring-orm dependency for Spring ORM support, it's required for hibernate integration in our spring project.
- hibernate-entitymanager and hibernate-core dependencies for Hibernate framework. Notice that version is 3.6.9.Final, for using Hibernate 4 all we need is to change it to 4.3.5.Final as commented in above pom.xml file.
- mysql-connector-java for MySQL driver for database connection.

Model Class or Entity Bean

We can use Hibernate XML based mapping as well as JPA annotation based mapping. Here I am using JPA annotations for mapping because hibernate provides JPA implementation.

```
package com.journaldev.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

/**
 * Entity bean with JPA annotations
 * Hibernate provides JPA implementation
 * @author pankaj
 *
 */
@Entity
@Table(name="Person")
public class Person {

        @Id
        @Column(name="id")
        @GeneratedValue(strategy=GenerationType.IDENTITY)
```

DAO Classes

We will implement two methods in our DAO classes, first to save the Person object into table and second that will fetch all the records from the table and returns the list of Persons.

```
package com.journaldev.dao;

import java.util.List;

import com.journaldev.model.Person;

public interface PersonDAO {

        public void save(Person p);

        public List<Person> list();

}
```

Above DAO class implementation would be like below.

```
package com.journaldev.dao;

import java.util.List;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;

import com.journaldev.model.Person;

public class PersonDAOImpl implements PersonDAO {

        private SessionFactory sessionFactory;

    public void setSessionFactory(SessionFactory sessionFactory) {
        this.sessionFactory = sessionFactory;
    }

        @Override
        public void save(Person p) {
                Session session = this.sessionFactory.openSession();
                Transaction tx = session.beginTransaction();
```

Notice that this is the only place where we are using Hibernate related classes. This pattern makes our implementation flexible and easy to migrate from one technology to another. For example, if we want to use iBatis ORM framework, all we need is to provide a DAO implementation for iBatis and then change the spring bean configuration file.

In above example, I am using Hibernate session transaction management. But we can also use Spring declarative transaction management using `@Transactional` annotation, read more at Spring Transaction Management.

Spring Bean Configuration File for Hibernate 3 Integration

Let's first look at the spring bean configurations we need for Hibernate 3 integration, we will look into detail later on.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:aop="http://www.springframework.org/schema/aop"
        xmlns:tx="http://www.springframework.org/schema/tx"
        xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
                http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-4.0.xsd
                http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-4.0.xsd">

        <bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"
                destroy-method="close">
                <property name="driverClassName" value="com.mysql.jdbc.Driver" />
                <property name="url" value="jdbc:mysql://localhost:3306/TestDB" />
                <property name="username" value="pankaj" />
                <property name="password" value="pankaj123" />
        </bean>

<!-- Hibernate 3 XML SessionFactory Bean definition-->
<!--    <bean id="hibernate3SessionFactory"
```

There are two ways we can provide database connection details to Hibernate, first by passing everything in `hibernateProperties` and second by creating a DataSource and then passing it to hibernate. I prefer the second approach, that's why we have Apache Commons DBCP dependency to create a `BasicDataSource` by setting database connection properties.

For Spring and Hibernate 3 integration, **Spring ORM** provides two classes – `org.springframework.orm.hibernate3.LocalSessionFactoryBean` when hibernate mappings are XML based and `org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean` for annotations based mapping. I have provided simple bean configuration of `LocalSessionFactoryBean` in comments, if you are using XML based mappings. `AnnotationSessionFactoryBean` extends `LocalSessionFactoryBean` class, so it has all the basic properties for hibernate integration.

The properties are self understood and mostly hibernate related, so I will not go into much detail for them. But if you are wondering from where **hibernateProperties**, **annotatedClasses** are coming, you need to look into the bean class source code.

Notice the bean definition of **personDAO**, like I said earlier if we have to switch to some other ORM framework, we need to change the implementation class here and set any other properties we need.

Spring 4 Hibernate 3 Test Program

Our setup is ready now, let's write a simple program to test our application.

```
package com.journaldev.main;

import java.util.List;

import org.springframework.context.support.ClassPathXmlApplicationContext;
import com.journaldev.dao.PersonDAO;
import com.journaldev.model.Person;

public class SpringHibernateMain {

        public static void main(String[] args) {

                ClassPathXmlApplicationContext context = new
ClassPathXmlApplicationContext("spring.xml");

                PersonDAO personDAO = context.getBean(PersonDAO.class);

                Person person = new Person();
                person.setName("Pankaj"); person.setCountry("India");

                personDAO.save(person);
```

When we execute above program, we get a lot of output related to Hibernate because I haven't setup logging properly, but that's out of scope of this tutorial. However we get following output generated by our program.

```
Person::id=3, name=Pankaj, country=India
Person List::id=1, name=Pankaj, country=India
Person List::id=2, name=Pankaj, country=India
Person List::id=3, name=Pankaj, country=India
```

Spring 4 Hibernate 4 integration changes

Now let's change our application to use Hibernate 4 instead of Hibernate 3. For this migration, we need to make only following configuration changes.

1. Change the hibernate version to 4.3.5.Final in the pom.xml file, as shown in comments above.
2. Change the spring bean configuration file, till now you must have figured out that Spring bean configuration file is the key for integration of spring and hibernate framework. Below spring bean configuration file will work for Spring 4 and Hibernate 4 versions.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:aop="http://www.springframework.org/schema/aop"
        xmlns:tx="http://www.springframework.org/schema/tx"
        xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
                http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-4.0.xsd
                http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-4.0.xsd">

        <bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"
                destroy-method="close">
                <property name="driverClassName" value="com.mysql.jdbc.Driver"
/>
                <property name="url" value="jdbc:mysql://localhost:3306/TestDB"
/>
                <property name="username" value="pankaj" />
```

```
            <property name="password" value="pankaj123" />
        </bean>
```

> For hibernate 4, we need to use `org.springframework.orm.hibernate4.LocalSessionFactoryBean` for SessionFactory bean, Spring ORM has merged both the classes for Hibernate 3 and there is a single class now, this is good to avoid confusion. All the other configurations are same as before.

That's it, our project is successfully migrated to Hibernate 4, neat isn't it. Just change the `SpringHibernateMain` class to use `spring4.xml` for beans configuration and it will work fine, you will get same output as before.

You can download the final project from below link and play around with more configurations to learn more.

Download Spring Hibernate Integration Project

« PREVIOUS
Hibernate SessionFactory

NEXT »
Spring MVC Hibernate MySQL Integration CRUD Example Tutorial

About Pankaj
If you have come this far, it means that you liked what you are reading. Why not reach little more and connect with me directly on **Google Plus**, **Facebook** or **Twitter**. I would love to hear your thoughts and opinions on my articles directly.

Recently I started creating video tutorials too, so do check out my videos on **Youtube**.

FILED UNDER: HIBERNATE, SPRING

Comments

shivani says

JANUARY 15, 2018 AT 7:39 PM

why web.xml is not required in this example?

Reply

Naseem says

FEBRUARY 26, 2018 AT 6:04 AM

Because its not dealing with views.

Reply

Jam says

DECEMBER 4, 2017 AT 6:15 PM

Hi, I'm new to Spring. In the Main function there's this line:
PersonDAO personDAO = context.getBean(PersonDAO.class);
Is this the same as:
PersonDAO personDAO = context.getBean("personDAO");
?
Thanks in advance for your reply.

Reply

Nagendra says

MARCH 28, 2018 AT 3:07 AM

Yes, It is same. You are passing 'personDAO' which is bean id. You should typecast it to persondao class like below. [why beacuse getBean(String beanname) returns Object.]
PersonDAO personDAO = (PersonDAO) context.getBean("personDAO");

Reply

Ilya says

SEPTEMBER 18, 2017 AT 3:16 PM

Just "PERSON"! Not "Person" table name.
When I changed it, all stay to work) thx for guide

Reply

Samet says

JUNE 22, 2017 AT 6:48 AM

Hello,
i have one question,
first of all i want to apoligize to you for my bad english.
doesn't occur SessionFactory for each dao object when we wire it with xml bean approach?
does it need to create once if thats right?
I hope i could express myself.

Reply

vishal says

MAY 22, 2017 AT 3:21 AM

Getting java.lang.IllegalArgumentException at personDAO.save(person) line in springhibernatemail class

Reply

sumit says

FEBRUARY 9, 2017 AT 12:30 AM

Nice tutorial! thanks………..

Reply

Akouri ammar says

NOVEMBER 15, 2016 AT 2:26 PM

Thank you so much

Reply

maryetta heimann says

OCTOBER 5, 2016 AT 1:37 AM

Excellent post . Speaking of which , if anyone requires a Jamaica defence force application form , my kids filled a fillable version here `https://goo.gl/TNmQbl`

Reply

viruskimera says

OCTOBER 4, 2016 AT 8:18 AM

Do you have an FTP inbound/outbound adapters example? with Maven. Please reply.

Reply

Rosa says

JULY 27, 2016 AT 4:00 AM

Hello PANKAJ,
Thanks for this tutorial.
I'm trying to create CRUD application with both Spring and Hibernate: Only I found a problem with Update.

******************In PersonDAOImpl.java, I made:

```
public void updatePerson(int id) {
Session session = this.sessionFactory.openSession();
Transaction tx = session.beginTransaction();
Person p = (Person) session.load(Person.class, new Integer(id));
session.update(p);
tx.commit();
session.close();
}
```

******************In SpringHibernateMain, I made:

```
Person person = personDAO.getPersonById(1);
person3.setName("Lisa");
```

person3.setCountry("UK");
personDAO.updatePerson(person.getId());
———> I don't have any error on Console but there's any update on database.
Can You lease tell me what's the error or what I missed.
Regards.
Reply

Rosa says
JULY 27, 2016 AT 4:36 AM
Hello PANKAJ,
It was solved.
The problem was the use of Logger on PersonDAOImpl.
Thanks.
Reply

ALPER KOPUZ says
JULY 12, 2016 AT 12:20 AM
WARN : org.springframework.web.servlet.PageNotFound – No mapping found for HTTP request with URI [/SpringMVCHibernate/] in DispatcherServlet with name 'appServlet'
What I need to do for the warning ? As far as I see , I could not see an exact solution for that. I will be appreciated if you can help me.
Reply

Stryder Jermaine says
JUNE 20, 2016 AT 4:31 AM
Hi please I'm getting this error when I try to build the project in NetBeans.
Plugin org.apache.maven.plugins:maven-jar-plugin:2.3.2 or one of its dependencies could not be resolved: Failed to read artifact descriptor for org.apache.maven.plugins:maven-jar-plugin:jar:2.3.2: Could not transfer artifact org.apache.maven.plugins:maven-jar-plugin:pom:2.3.2 from/to central (http://repo.maven.apache.org/maven2): This is usually a temporary error during hostname resolution and means that the local server did not receive a response from an authoritative server -> [Help 1]
Please help
Reply

Pankaj says
JUNE 20, 2016 AT 5:12 AM
check your internet connections, it seems required jars are not getting downloaded. Are you behind some firewall?
Reply

Imran says
MARCH 12, 2016 AT 6:39 AM
Thanks a lot,
Great job !!!
Reply

Chen says

MARCH 8, 2016 AT 4:40 AM

Hi,
Can anybody tell me what's the function of the toString() method in Person.java ?
I'm so confused.

Reply

Nishant says

MARCH 11, 2016 AT 11:54 AM

toString() method of Person class is used wjen i simply write system.out.println(person). Same goes with any class you define. you just need to provide this method and whenever you sysout something it will automatically call this function.

Reply

Mokshh says

MARCH 7, 2017 AT 3:06 AM

Hi Chen,
Just to represent Person class object into String representation. As if you don't override toString() method then while printing the person object reference, it will call Object class toString() method which will return getClass().getName() + "@" + Integer.toHexString(hashCode()) representation.
If your are using Eclipse then I would recommended to use shortcut to generate toString() method with class attributes by entering "alt+shift+s+s"

Reply

Nagendra says

MARCH 28, 2018 AT 3:12 AM

Awesome Explanation

Reply

Nilvandro Muianga says

FEBRUARY 12, 2016 AT 5:32 AM

Hi Pankaj, this tutorial was so helpful for me. I'm so thankful.

Reply

Divy Kumar says

JANUARY 30, 2016 AT 12:00 PM

I am getting following warning and after that I am not able to access the webpage.May you please help.WARN : org.springframework.web.servlet.PageNotFound – No mapping found for HTTP request with URI [/] in DispatcherServlet with name 'appServlet'

Reply

Gunns says

JANUARY 22, 2016 AT 12:50 AM

SEVERE: Servlet.service() for servlet [rest] in context with path [/WebRestDemo] threw exception [Handler processing failed; nested exception is java.lang.NoSuchMethodError: org.hibernate.SessionFactory.openSession()Lorg/hibernate/Session;] with root cause java.lang.NoSuchMethodError: org.hibernate.SessionFactory.openSession()Lorg/hibernate/Session;

Reply

Enrico says

NOVEMBER 15, 2015 AT 11:29 AM

Thanks a lot.
Simple tutorial but containing all the required steps to have the example working.
Just one hint, instead of creating the table manually with the SQL command I'd rather go with
update
in the spring configuration file.

Reply

Alex says

NOVEMBER 3, 2015 AT 4:20 PM

Hi Pankaj thank you for this intro.
I am getting this error when trying to run the JAR file

```
Exception in thread "main" java.lang.NoClassDefFoundError:
org/springframework/context/support/ClassPathXmlApplicationContext
at com.journaldev.main.springHibernateMain.main(springHibernateMain.java:13)
Caused by: java.lang.ClassNotFoundException:
org.springframework.context.support.ClassPathXmlApplicationContext
at java.net.URLClassLoader$1.run(URLClassLoader.java:366)
at java.net.URLClassLoader$1.run(URLClassLoader.java:355)
at java.security.AccessController.doPrivileged(Native Method)
at java.net.URLClassLoader.findClass(URLClassLoader.java:354)
at java.lang.ClassLoader.loadClass(ClassLoader.java:425)
at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:308)
at java.lang.ClassLoader.loadClass(ClassLoader.java:358)
```

The line with the error is
ClassPathXmlApplicationContext context=new ClassPathXmlApplicationContext("spring4.xml");
Any ideas?

Reply

> **Shiva says**
>
> FEBRUARY 22, 2016 AT 9:25 PM
>
> Its
> ApplicationContext context = new ClassPathXmlApplicationContext("spring4.xml");
>
> Reply

janis says

OCTOBER 30, 2015 AT 5:13 AM

Not good example many mistakes

Reply

PAblo says

DECEMBER 14, 2015 AT 6:38 PM

Janis, could you please clarify ?

Reply

Someone else. says

JULY 24, 2016 AT 3:15 AM

I'm not Janis, but I'll give my 2 cents. After I downloaded the sourcecode and imported it to my IDE, I'm getting few errors. IDE (STS) says e.g:
"The method list() of type PersonDAOImpl must override a superclass method"
and
"The method save(Person) of type PersonDAOImpl must override a superclass method"
Also, see ClassPathXmlApplicationContext discussion in these comments.
I'm not saying this article is bad. I'm saying Java environment setup for programming is often so complicated that if one does not manage the whole setup process, there will be problems. If one wants to find out what problems there are with this tutorial, then take a blank computer in front of you and follow this tutorial (don't do any setup which is reguired, but missing from this tutorial, or use your knowledge which people are trying to learn using this tutorial.). You will soon find problems, because this tutorial was written for certain setup, which is impossible to duplicate with these instructions.
Also, I could not get the tutorial sourcecode to work. After deducting I have to rename spring4.xml to spring.xml to use hibernate 4, I'm still stuck, mysql connection doesn't work.
Next, I'll go to some other site to find a next tutorial to try. After I find one which I can get to work with decent work, I'll really investigate how it works and start learning from it. I don't use tutorials which I have to fix for them to be usable. Usually if I can fix them easily, I don't need them.

Reply

Surya says

SEPTEMBER 20, 2015 AT 8:08 AM

Hi All,
I am getting this below error after trying to do theexample in eclipse

```
11:08:01,052 [ INFO] ClassPathXmlApplicationContext:510 – Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@1e1c66a: startup date [Sun Sep 20 11:08:01 EDT 2015]; root of context hierarchy
11:08:01,146 [ INFO] XmlBeanDefinitionReader:317 – Loading XML bean definitions from class path resource [spring.xml]
11:08:01,474 [ WARN] ClassPathXmlApplicationContext:487 – Exception encountered during context initialization – cancelling refresh attempt
org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'hibernate3AnnotatedSessionFactory' defined in class path resource [spring.xml]: Initialization of bean failed; nested exception is java.lang.NoClassDefFoundError: [Lorg/hibernate/engine/FilterDefinition;
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:5
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:476
at
```

```
org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:303)
at
org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:230)
at
org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:299)
at
org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:194)
at
org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.java:743)
at
org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:757)
at
org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:480)
at org.springframework.context.support.ClassPathXmlApplicationContext.
(ClassPathXmlApplicationContext.java:139)
at org.springframework.context.support.ClassPathXmlApplicationContext.
(ClassPathXmlApplicationContext.java:83)
at rugal.sample.core.service.impl.SpringHibernateMain.main(SpringHibernateMain.java:15)
Caused by: java.lang.NoClassDefFoundError: [Lorg/hibernate/engine/FilterDefinition;
at java.lang.Class.getDeclaredMethods0(Native Method)
at java.lang.Class.privateGetDeclaredMethods(Unknown Source)
at java.lang.Class.privateGetPublicMethods(Unknown Source)
at java.lang.Class.privateGetPublicMethods(Unknown Source)
at java.lang.Class.getMethods(Unknown Source)
at org.springframework.beans.ExtendedBeanInfoFactory.supports(ExtendedBeanInfoFactory.java:54)
at org.springframework.beans.ExtendedBeanInfoFactory.getBeanInfo(ExtendedBeanInfoFactory.java:46)
at org.springframework.beans.CachedIntrospectionResults.(CachedIntrospectionResults.java:270)
at org.springframework.beans.CachedIntrospectionResults.forClass(CachedIntrospectionResults.java:189)
at
org.springframework.beans.BeanWrapperImpl.getCachedIntrospectionResults(BeanWrapperImpl.java:327)
at
org.springframework.beans.BeanWrapperImpl.getPropertyDescriptorInternal(BeanWrapperImpl.java:359)
at org.springframework.beans.BeanWrapperImpl.isWritableProperty(BeanWrapperImpl.java:439)
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.applyPropertyValues(AbstractAutowireCapableBeanFactor
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.populateBean(AbstractAutowireCapableBeanFactory.java:1
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:5
… 11 more
Caused by: java.lang.ClassNotFoundException: org.hibernate.engine.FilterDefinition
at java.net.URLClassLoader$1.run(Unknown Source)
at java.net.URLClassLoader$1.run(Unknown Source)
at java.security.AccessController.doPrivileged(Native Method)
at java.net.URLClassLoader.findClass(Unknown Source)
at java.lang.ClassLoader.loadClass(Unknown Source)
at sun.misc.Launcher$AppClassLoader.loadClass(Unknown Source)
at java.lang.ClassLoader.loadClass(Unknown Source)
… 26 more
Exception in thread “main” org.springframework.beans.factory.BeanCreationException: Error creating
bean with name ‘hibernate3AnnotatedSessionFactory’ defined in class path resource [spring.xml]:
Initialization of bean failed; nested exception is java.lang.NoClassDefFoundError:
[Lorg/hibernate/engine/FilterDefinition;
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:5
at
```

```
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:476
at
org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:303)
at
org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:230)
at
org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:299)
at
org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:194)
at
org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.java:743)
at
org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:757)
at
org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:480)
at org.springframework.context.support.ClassPathXmlApplicationContext.
(ClassPathXmlApplicationContext.java:139)
at org.springframework.context.support.ClassPathXmlApplicationContext.
(ClassPathXmlApplicationContext.java:83)
at rugal.sample.core.service.impl.SpringHibernateMain.main(SpringHibernateMain.java:15)
Caused by: java.lang.NoClassDefFoundError: [Lorg/hibernate/engine/FilterDefinition;
at java.lang.Class.getDeclaredMethods0(Native Method)
at java.lang.Class.privateGetDeclaredMethods(Unknown Source)
at java.lang.Class.privateGetPublicMethods(Unknown Source)
at java.lang.Class.privateGetPublicMethods(Unknown Source)
at java.lang.Class.getMethods(Unknown Source)
at org.springframework.beans.ExtendedBeanInfoFactory.supports(ExtendedBeanInfoFactory.java:54)
at org.springframework.beans.ExtendedBeanInfoFactory.getBeanInfo(ExtendedBeanInfoFactory.java:46)
at org.springframework.beans.CachedIntrospectionResults.(CachedIntrospectionResults.java:270)
at org.springframework.beans.CachedIntrospectionResults.forClass(CachedIntrospectionResults.java:189)
at
org.springframework.beans.BeanWrapperImpl.getCachedIntrospectionResults(BeanWrapperImpl.java:327)
at
org.springframework.beans.BeanWrapperImpl.getPropertyDescriptorInternal(BeanWrapperImpl.java:359)
at org.springframework.beans.BeanWrapperImpl.isWritableProperty(BeanWrapperImpl.java:439)
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.applyPropertyValues(AbstractAutowireCapableBeanFactor
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.populateBean(AbstractAutowireCapableBeanFactory.java:1
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:
… 11 more
Caused by: java.lang.ClassNotFoundException: org.hibernate.engine.FilterDefinition
at java.net.URLClassLoader$1.run(Unknown Source)
at java.net.URLClassLoader$1.run(Unknown Source)
at java.security.AccessController.doPrivileged(Native Method)
at java.net.URLClassLoader.findClass(Unknown Source)
at java.lang.ClassLoader.loadClass(Unknown Source)
at sun.misc.Launcher$AppClassLoader.loadClass(Unknown Source)
at java.lang.ClassLoader.loadClass(Unknown Source)
… 26 more
```

Regards
Surya

Reply

pape says

OCTOBER 22, 2015 AT 8:49 AM

you have to vrified the location of the spring config file and refenced it in your web.xml

Reply

Deepak says

SEPTEMBER 12, 2015 AT 11:43 PM

Hi Pankaj Sir ,

Really nice example . But you have not defined HibernateTransactionManager in this , So my question is that if the TransactionManagers of spring are required in the case where we are using declarative transaction management of spring and the application is deployed in any j2ee container ?

Reply

Deepak says

SEPTEMBER 12, 2015 AT 11:37 PM

Hi Pankaj Sir , Really good example this is .

There are different transaction managers in spring. And you have not used hibernate transaction manager in this example . So my question is for the j2ee container application only where we are doing declarative transaction management the transaction managers are required ? Suppose I do not want to use any of them and my application is not residing inside any j2ee container .Then in this case if we really need to define the transaction managers ?

Reply

Subramanya says

SEPTEMBER 11, 2015 AT 11:48 PM

Also add to servlet-context.xml if you are facing the error as WARN : org.springframework.web.servlet.PageNotFound – No mapping found for HTTP request with URI [/] in DispatcherServlet with name 'appServlet'

Reply

Hicham says

SEPTEMBER 11, 2015 AT 9:18 AM

Can you help me please,

I use spring with hibernate3, when i test my program with JUnit, i have this message : org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'org.springframework.context.annotation.internalRequiredAnnotationProcessor': Initialization of bean failed; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'org.springframework.transaction.config.internalTransactionAdvisor': Cannot create inner bean '(inner bean)' of type [org.springframework.transaction.interceptor.TransactionInterceptor] while setting bean property 'transactionInterceptor'; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name '(inner bean)': Cannot resolve reference to bean 'transactionManager' while setting bean property 'transactionManager'; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'transactionManager' defined in class path resource [application-context.xml]: Cannot resolve reference to bean 'sessionFactory' while setting bean property 'sessionFactory'; nested exception is

```
org.springframework.beans.factory.BeanCreationException: Error creating bean with name
'sessionFactory' defined in class path resource [application-context.xml]: Cannot resolve reference to
bean 'dataSource' while setting bean property 'dataSource'; nested exception is
org.springframework.beans.factory.NoSuchBeanDefinitionException: No bean named 'dataSource' is
defined
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:5
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory$1.run(AbstractAutowireCapableBeanFactory.java:485)
at java.security.AccessController.doPrivileged(Native Method)
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:455
at
org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:251)
at
org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:169)
at
org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:248)
at
org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:170)
at
org.springframework.context.support.AbstractApplicationContext.getBean(AbstractApplicationContext.java:883)
at
org.springframework.context.support.AbstractApplicationContext.registerBeanPostProcessors(AbstractApplicationContext.java:593)
at
org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:351)
at org.springframework.context.support.ClassPathXmlApplicationContext.
(ClassPathXmlApplicationContext.java:122)
at org.springframework.context.support.ClassPathXmlApplicationContext.
(ClassPathXmlApplicationContext.java:66)
at com.university.service.CourseServiceTest.setUpBeforeClass(CourseServiceTest.java:17)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
at java.lang.reflect.Method.invoke(Unknown Source)
at org.junit.runners.model.FrameworkMethod$1.runReflectiveCall(FrameworkMethod.java:45)
at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:15)
at org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:42)
at org.junit.internal.runners.statements.RunBefores.evaluate(RunBefores.java:27)
at org.junit.internal.runners.statements.RunAfters.evaluate(RunAfters.java:30)
at org.junit.runners.ParentRunner.run(ParentRunner.java:300)
at org.eclipse.jdt.internal.junit4.runner.JUnit4TestReference.run(JUnit4TestReference.java:50)
at org.eclipse.jdt.internal.junit.runner.TestExecution.run(TestExecution.java:38)
at org.eclipse.jdt.internal.junit.runner.RemoteTestRunner.runTests(RemoteTestRunner.java:459)
at org.eclipse.jdt.internal.junit.runner.RemoteTestRunner.runTests(RemoteTestRunner.java:675)
at org.eclipse.jdt.internal.junit.runner.RemoteTestRunner.run(RemoteTestRunner.java:382)
at org.eclipse.jdt.internal.junit.runner.RemoteTestRunner.main(RemoteTestRunner.java:192)
```

application-context.xml :

```
com.university.model.Course
com.university.model.Student
com.university.model.Teacher
org.hibernate.dialect.MySQLInnoDBDialect
```

Reply

Muhammad Asif says

JULY 28, 2015 AT 6:50 AM

Excellent job, really interested and informative.

Reply

Umesh Annegirikar says

JULY 8, 2015 AT 2:02 AM

@Column(name="id")
@GeneratedValue(strategy=GenerationType.IDENTITY) not working for me i have used @Column (name="id",nullable=false) @GeneratedValue(strategy=GenerationType.AUTO) then it working fine cod e is explaining excellent

Reply

El. says

JUNE 4, 2015 AT 10:22 AM

Very good tutorial. Thank you so much for that work. have a nice Day. EL..

Reply

Derick Daniek says

APRIL 27, 2015 AT 9:39 AM

Hey pankaj, in spring4.xml you have used annotation mapping, and in class name you have given "localSessionFactoryBean", if i am not wrong "localSessionFactoryBean" is used for XML mapping, if so then why have you used it with "hibernate4AnnotatedSessionFactory".
Correct me if i am wrong, as i am new to Spring+Hibernate integration

Reply

Krishna Srinivasan says

APRIL 27, 2015 AT 9:20 AM

Good article!

Reply

Vinod says

APRIL 20, 2015 AT 1:14 PM

Hi Pankaj,
could you please confirm which maven archetype you have used to develop above mentioned application.

Reply

Surendra says

APRIL 12, 2015 AT 12:10 AM

Very well written example. Thoroughly understood. I know Hibernate and Spring Injection. Never worked on this integration. Now I'm able to do it on my own. Much appreciated work.

Reply

Dave says

FEBRUARY 28, 2015 AT 4:09 AM

This is a well put together tutorial – thanks

Reply

Dheeraj Kumar says

JANUARY 28, 2015 AT 4:40 AM

I'm integrating spring3.1 with hibernate4.0 i I'm getting blew error
ava.lang.NoClassDefFoundError: Lorg/hibernate/cache/CacheProvider;
at java.lang.Class.getDeclaredFields0(Native Method)
at java.lang.Class.privateGetDeclaredFields(Class.java:2291)
at java.lang.Class.getDeclaredFields(Class.java:1743)
at
org.springframework.orm.jpa.support.PersistenceAnnotationBeanPostProcessor.findPersistenceMetadata(PersistenceAnnotationBeanPostPro
at
org.springframework.orm.jpa.support.PersistenceAnnotationBeanPostProcessor.postProcessMergedBeanDefinition(PersistenceAnnotationBe
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.applyMergedBeanDefinitionPostProcessors(AbstractAutow
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:4
HIbernate Configuration xml
org.hibernate.dialect.Oracle10gDialect
true
10
true
true
thread
true
com.dcs.employee.model.Employee

Reply

Chandra Bhanu Rastogi says

MARCH 15, 2015 AT 10:31 PM

It seems compatibility issue. You are using hibernate 4.x but somewhere at runtime it's trying to use spring classes which are hibernate 3 compatible. Check all the beans defined in configration xml and make sure you don't pass hibernate 3 compatible classes of spring.

Reply

Prem Tiwari says

JUNE 13, 2015 AT 10:09 PM

Hi Dheeraj,
Please replace bleow in your pom.xml and you are done :).
4.3.5.Final

Reply

papepula says

JANUARY 9, 2015 AT 6:11 AM

Hi, when i start my app with string boot run then it works.
But i get this excpetion when i wanna deploy the war file on a tomcat 7.
I did exactly the same but my tomcat doesnt know the hibernate.dialect.
I also tried to remove the hibernate.dialect because it isn't necessary to declared it
and the app should select automatically one. I tried so much but nothing works…
Hope someone has an idea.

```
org.apache.catalina.LifecycleException: Failed to start component
[StandardEngine[Catalina].StandardHost[localhost].StandardContext[/myapp-0.0.1]]
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:154)
at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:901)
at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:877)
at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:632)
at org.apache.catalina.startup.HostConfig.deployWAR(HostConfig.java:1073)
at org.apache.catalina.startup.HostConfig.deployApps(HostConfig.java:553)
at org.apache.catalina.startup.HostConfig.check(HostConfig.java:1648)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:606)
at org.apache.tomcat.util.modeler.BaseModelMBean.invoke(BaseModelMBean.java:301)
at
com.sun.jmx.interceptor.DefaultMBeanServerInterceptor.invoke(DefaultMBeanServerInterceptor.java:819)
at com.sun.jmx.mbeanserver.JmxMBeanServer.invoke(JmxMBeanServer.java:801)
at org.apache.catalina.manager.ManagerServlet.check(ManagerServlet.java:1445)
at org.apache.catalina.manager.HTMLManagerServlet.upload(HTMLManagerServlet.java:284)
at org.apache.catalina.manager.HTMLManagerServlet.doPost(HTMLManagerServlet.java:206)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:646)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:727)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:303)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:208)
at org.apache.catalina.filters.CsrfPreventionFilter.doFilter(CsrfPreventionFilter.java:213)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:241)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:208)
at org.apache.catalina.filters.SetCharacterEncodingFilter.doFilter(SetCharacterEncodingFilter.java:108)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:241)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:208)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:220)
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:122)
at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:610)
at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:170)
at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:98)
at org.apache.catalina.valves.AccessLogValve.invoke(AccessLogValve.java:950)
at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:116)
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:408)
at org.apache.coyote.http11.AbstractHttp11Processor.process(AbstractHttp11Processor.java:1040)
at org.apache.coyote.AbstractProtocol$AbstractConnectionHandler.process(AbstractProtocol.java:607)
at org.apache.tomcat.util.net.JIoEndpoint$SocketProcessor.run(JIoEndpoint.java:313)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)
at java.lang.Thread.run(Thread.java:745)
Caused by: org.springframework.beans.factory.BeanCreationException: Error creating bean with name
'entityManagerFactory' defined in class path resource
[org/springframework/boot/autoconfigure/orm/jpa/HibernateJpaAutoConfiguration.class]: Invocation
```

```
of init method failed; nested exception is org.hibernate.HibernateException: Access to
DialectResolutionInfo cannot be null when 'hibernate.dialect' not set
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initializeBean(AbstractAutowireCapableBeanFactory.java:15
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:5
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:475
at
org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:302)
at
org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:228)
at
org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:298)
at
org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:193)
at
org.springframework.context.support.AbstractApplicationContext.getBean(AbstractApplicationContext.java:975)
at
org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:752)
at
org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:482)
at
org.springframework.boot.context.embedded.EmbeddedWebApplicationContext.refresh(EmbeddedWebApplicationContext.java:109)
at org.springframework.boot.SpringApplication.refresh(SpringApplication.java:691)
at org.springframework.boot.SpringApplication.run(SpringApplication.java:320)
at org.springframework.boot.builder.SpringApplicationBuilder.run(SpringApplicationBuilder.java:142)
at
org.springframework.boot.context.web.SpringBootServletInitializer.createRootApplicationContext(SpringBootServletInitializer.java:89)
at
org.springframework.boot.context.web.SpringBootServletInitializer.onStartup(SpringBootServletInitializer.java:51)
at
org.springframework.web.SpringServletContainerInitializer.onStartup(SpringServletContainerInitializer.java:175)
at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5456)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150)
… 40 more
Caused by: org.hibernate.HibernateException: Access to DialectResolutionInfo cannot be null when
'hibernate.dialect' not set
at
org.hibernate.engine.jdbc.dialect.internal.DialectFactoryImpl.determineDialect(DialectFactoryImpl.java:104)
at org.hibernate.engine.jdbc.dialect.internal.DialectFactoryImpl.buildDialect(DialectFactoryImpl.java:71)
at org.hibernate.engine.jdbc.internal.JdbcServicesImpl.configure(JdbcServicesImpl.java:205)
at
org.hibernate.boot.registry.internal.StandardServiceRegistryImpl.configureService(StandardServiceRegistryImpl.java:111)
at
org.hibernate.service.internal.AbstractServiceRegistryImpl.initializeService(AbstractServiceRegistryImpl.java:234)
at
org.hibernate.service.internal.AbstractServiceRegistryImpl.getService(AbstractServiceRegistryImpl.java:206)
at org.hibernate.cfg.Configuration.buildTypeRegistrations(Configuration.java:1885)
at org.hibernate.cfg.Configuration.buildSessionFactory(Configuration.java:1843)
at
org.hibernate.jpa.boot.internal.EntityManagerFactoryBuilderImpl$4.perform(EntityManagerFactoryBuilderImpl.java:850)
at
org.hibernate.jpa.boot.internal.EntityManagerFactoryBuilderImpl$4.perform(EntityManagerFactoryBuilderImpl.java:843)
at
org.hibernate.boot.registry.classloading.internal.ClassLoaderServiceImpl.withTccl(ClassLoaderServiceImpl.java:397)
```

```
at
org.hibernate.jpa.boot.internal.EntityManagerFactoryBuilderImpl.build(EntityManagerFactoryBuilderImpl.java:842)
at
org.hibernate.jpa.HibernatePersistenceProvider.createContainerEntityManagerFactory(HibernatePersistenceProvider.java:152)
at
org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean.createNativeEntityManagerFactory(LocalContainerEntityManagerFact
at
org.springframework.orm.jpa.AbstractEntityManagerFactoryBean.afterPropertiesSet(AbstractEntityManagerFactoryBean.java:318)
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.invokeInitMethods(AbstractAutowireCapableBeanFactory.ja
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initializeBean(AbstractAutowireCapableBeanFactory.java:15
… 58 more
```

Reply

Bhawani Shanker says

DECEMBER 21, 2014 AT 11:47 PM

I need Spring 4 MVC 1-on-1 training starting right from setting up web apps development environment (JDK8, Spring 4, Gradle etc) supports.
Who can help?

Reply

prasad says

DECEMBER 30, 2014 AT 9:08 PM

sure I can…contact on kulkarni.prasad98@gmail.com

Reply

Alex says

DECEMBER 17, 2014 AT 4:45 AM

this was very helpful to understand the spring hibernate integration

Reply

Vijay Ambekar says

DECEMBER 9, 2014 AT 9:13 AM

Great work!!! Thanks. It worked for me. □

Reply

Ruby says

DECEMBER 8, 2014 AT 6:32 PM

Thank so much. It is very helpful for beginer like me.

Reply

Aarti says

DECEMBER 1, 2014 AT 10:55 PM

All of your tutorial are really helpful… Good Work & Thank You!!!!

Reply

Biplab says

NOVEMBER 29, 2014 AT 8:48 PM

Hi Pankaj,
Great tutorial for spring beginners. I just wanted to know if we could use oracle.jdbc.pool.OracleDataSource as a data source instead of org.apache.commons.dbcp.BasicDataSource. I am using oracle 10g xe.
Thanks !!

Reply

Pankaj says

NOVEMBER 29, 2014 AT 9:18 PM

Yes we can use any of the DataSource implementation.

Reply

Biplab says

NOVEMBER 30, 2014 AT 6:58 AM

Thanks for the reply. I will try that one.

Reply

Deevaker Bhatia says

NOVEMBER 18, 2014 AT 10:06 AM

sir i want to create an application employee management i want to use STS Hibernate maven …..can you help me out in this webapp

Reply

aditya says

OCTOBER 26, 2014 AT 7:46 AM

in create table u say `Person` but in entity the table name is 'PERSON' its different and making error

Reply

Pankaj says

OCTOBER 26, 2014 AT 2:03 PM

Usually database table name is not case sensitive. Looks like not the case with your database.

Reply

lib says

NOVEMBER 22, 2014 AT 6:44 AM

Just wanted to add that I also got a SQLException saying testdb.PERSON doesn't exist when my table name was Person. I'm using mysql 5.5.40

Reply

Pankaj says

NOVEMBER 22, 2014 AT 10:16 AM

I got the root cause of the issue. I am using Mac OS X as my development machine and I believe you guys are using some other form of Unix. According to MySQL Reference manual for case sensitivity:

Consequently, the case sensitivity of the underlying operating system plays a part in the case sensitivity of database and table names. This means database and table names are not case sensitive in Windows, and case sensitive in most varieties of Unix. One notable exception is Mac OS X, which is Unix-based but uses a default file system type (HFS+) that is not case sensitive.

Reference: http://dev.mysql.com/doc/refman/5.0/en/identifier-case-sensitivity.html

NOTE: To avoid this issue for other users, I have updated the post accordingly.

Reply

Ivan says

OCTOBER 21, 2014 AT 2:11 AM

Hello, I have couple question regarding to your post.

1. Why don't you use Spring @Autowired in your DAO and why don't use this instead?

```
public void save(Person person)
{
this.sessionFactory.getCurrentSession().save(person);
}
```

Because I don't know whether you use Transaction for any benefits?

2. Could you give more example if you create another class to wrap PersonDAOImpl to it gonna be like:

```
@Service
@Transactional(readOnly = true)
public class PersonServiceImpl implements PersonService
{
@Autowired
protected PersonDao personDao;
public List getPersons()
{
return this.personDao.getAll();
}
@Transactional(readOnly = false)
public void save(Person person)
{
this.personDao.save(person);
}
}
```

Because #2 don't work for me, it said that I've configured more than 1 autowired? I'm new to hibernate 4 and Spring 4, so I need some answers for this updates. And also please bring it with your spring-config.xml in case mapping for this service class I assume would be better for hierarchical software achitecture.

Thank you for your assistance.

Reply

Muhammed AbdElFatah says

OCTOBER 20, 2014 AT 1:32 AM

I am newbie in hibernate with spring and i am really thank you for this tutorial .
I tested hibernate v3 with spring v4 and it works good , but when i changed it with hibernate v4
it gives to me this error >>>

```
Exception in thread “main” java.lang.UnsupportedOperationException: The user must supply a JDBC
connection
at
org.hibernate.connection.UserSuppliedConnectionProvider.getConnection(UserSuppliedConnectionProvider.java:54)
at org.hibernate.jdbc.ConnectionManager.openConnection(ConnectionManager.java:446)
at org.hibernate.jdbc.ConnectionManager.getConnection(ConnectionManager.java:167)
at org.hibernate.jdbc.JDBCContext.connection(JDBCContext.java:160)
at org.hibernate.transaction.JDBCTransaction.begin(JDBCTransaction.java:81)
at org.hibernate.impl.SessionImpl.beginTransaction(SessionImpl.java:1473)
at db.dao.EmployeeDAOImpl.saveEmployee(EmployeeDAOImpl.java:22)
at db.main.SpringHibernateMain.main(SpringHibernateMain.java:23)
```

i searched a lot about this error and cannot find any solution.
thank you ^_^

Reply

Andrei Anpilogov says

SEPTEMBER 24, 2014 AT 12:04 AM

Big thanks!

Reply

right_now says

OCTOBER 14, 2014 AT 12:28 PM

Nice tutorial! There’s a great RAD tool out called Jigy Generator that automatically spits you out a fully configured spring project which can already connect to your database, authenticate users, handle file uploads, etc. It even creates dao’s, domain objects and validators in your project by reverse engineering your database. You can download the project at http://www.getjigy.com

Reply

Sunil says

SEPTEMBER 21, 2014 AT 6:43 PM

Thank you so much . Its very helpful tutorials

Reply

Rizwan Mursaleen says

SEPTEMBER 15, 2014 AT 3:02 AM

It was very helpful and very clear. Thank you.

Reply

Suresh says

JULY 28, 2014 AT 1:26 AM

Hi pankaj, do you have any spring and hibernate real time project. If it's there I will pay you

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

☐

Save my name, email, and website in this browser for the next time I comment.

POST COMMENT

Search for tutorials...

DOWNLOAD ANDROID APP

GET IT ON Google Play

SPRING FRAMEWORK

Spring Tutorial

Spring Core

- Spring Framework
- Spring Dependency Injection
- Spring IoC and Bean
- Spring Bean Life Cycle
- Spring REST
- Spring REST XML
- Spring RestTemplate
- Spring AOP

- Spring AOP Method Profiling
- Spring Annotations
- Spring @Autowired
- Spring @RequestMapping

Spring MVC

- Spring MVC Example
- Spring MVC Tutorial
- Spring MVC Exception Handling
- Spring MVC Validator
- Spring MVC Intercepter
- Spring MVC File Upload
- Spring MVC i18n
- Spring MVC Hibernate MqSQL

Spring ORM

- Spring ORM
- Spring ORM JPA
- Spring Data JPA
- Spring Transaction
- Spring JdbcTemplate

Spring Security

- Spring Security Overview
- Spring Security Example Tutorial
- Spring Security UserDetailsService
- Spring MVC Login Logout
- Spring Security Roles

Spring Boot

- Spring Boot Tutorial
- Spring Boot Components
- Spring Boot CLI Hello World
- Spring Boot Initilizr Web
- Spring Boot Initilizr IDE
- Spring Boot Initilizr CLI
- Spring Boot Initilizr Tools
- Spring Boot MongoDB
- Spring Boot Redis Cache
- Spring Boot Interview Questions

Spring Batch

- Spring Batch
- Spring Batch Example

Spring AMQP

- Spring AMQP
- Spring RabbitMQ
- Spring AMQP RabbitMQ
- Apache ActiveMQ
- Spring ActiveMQ Tutorial
- Spring ActiveMQ Example

Spring Integrations

- Spring JDBC
- Spring DataSource JNDI
- Spring Hibernate
- Spring Primefaces JPA
- Spring Primefaces MongoDB
- Spring Primefaces Hibernate
- SpringRoo Primefaces Hibernate
- Spring JSF
- Spring JDF Hibernate

Miscellaneous

- Spring Data MongoDB

- Spring Interview Questions

HIBERNATE FRAMEWORK

Hibernate Tutorial

- Hibernate Example
- Hibernate SessionFactory
- Hibernate Session get load
- Hibernate Session save
- HQL Example
- Hibernate Criteria
- Hibernate SQL
- Hibernate Named Query
- Hibernate Log4J
- Hibernate Validator
- Hibernate Tomcat DataSource

Hibernate Mapping

- Hibernate One to One Mapping
- Hibernate One to Many Mapping
- Hibernate Many to Many Join Tables

Hibernate Caching

- Hibernate Cache
- Hibernate EHCache

Hibernate Integrations

- Hibernate Spring
- Hibernate Spring MVC
- Hibernate Struts 2
- Hibernate Primefaces
- Hibernate Primefaces Spring
- Hibernate SpringRoo Primefaces
- Hibernate JSF Spring

Miscellaneous

- Hibernate Tools Eclipse Plugin
- Hibernate Configuration Offline
- [Solved] No identifier specified
- Hibernate Program Not Terminating
- Access to DialectResolutionInfo
- get is not valid
- No CurrentSessionContext configured
- Hibernate Interview Questions

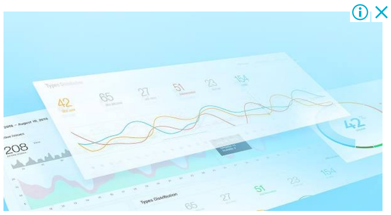
RECOMMENDED TUTORIALS

Java Tutorials

- Java IO
- Java Regular Expressions
- Multithreading in Java
- Java Logging
- Java Annotations
- Java XML
- Collections in Java
- Java Generics
- Exception Handling in Java
- Java Reflection
- Java Design Patterns
- JDBC Tutorial

Java EE Tutorials

- Servlet JSP Tutorial
- Struts2 Tutorial
- Spring Tutorial
- Hibernate Tutorial
- Primefaces Tutorial
- Apache Axis 2
- JAX-RS
- Memcached Tutorial


Progress
Kendo UI

© 2018 · Privacy Policy · Don't copy, it's Bad Karma · Powered by WordPress