

[JAVA TUTORIAL](#)[#INDEX POSTS](#)[#INTERVIEW QUESTIONS](#)[RESOURCES](#)[HIRE ME](#)[DOWNLOAD ANDROID APP](#)[CONTRIBUTE](#)**Subscribe to Download Java Design Patterns eBook**[DOWNLOAD NOW](#)[HOME](#) » [SPRING](#) » [SPRING DATASOURCE JNDI WITH TOMCAT EXAMPLE](#)

# Spring DataSource JNDI with Tomcat Example

APRIL 2, 2018 BY [PANKAJ](#) — [22 COMMENTS](#)

Welcome to the Spring DataSource JNDI Tomcat Example Tutorial. Earlier we saw how to implement database operations using [Spring JDBC](#) integration. However most of the time enterprise applications are deployed in a servlet container such as Tomcat, JBoss etc.

## Table of Contents [\[hide\]](#)

- 1 [Spring DataSource](#)
  - 1.1 [Database Setup](#)
  - 1.2 [Spring DataSource MVC Project](#)
  - 1.3 [Spring JDBC and Jackson Dependencies](#)
  - 1.4 [Model Class](#)
  - 1.5 [Spring Controller Class](#)
  - 1.6 [Spring Bean Configuration](#)
  - 1.7 [Tomcat DataSource JNDI Configuration](#)
  - 1.8 [Running the Spring DataSource JNDI Sample Project](#)

## Spring DataSource

We know that [DataSource with JNDI](#) is the preferred way to achieve connection pooling and get benefits of container implementations. Today we will look how we can configure a Spring Web Application to use JNDI connections provided by Tomcat.

For my example, I will use MySQL database server and create a simple table with some rows. We will create a Spring Rest web service that will return the JSON response with list of all the data in the table.

## Database Setup

```
CREATE TABLE `Employee` (  
  `id` int(11) unsigned NOT NULL,  
  `name` varchar(20) DEFAULT NULL,  
  `role` varchar(20) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

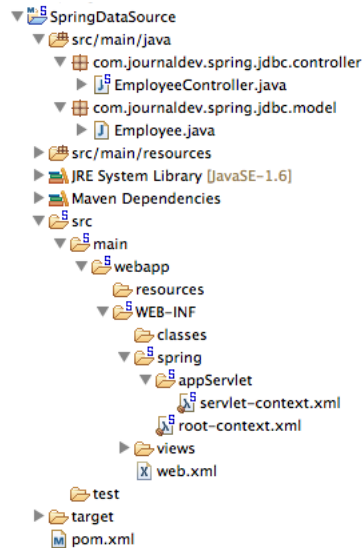
```
INSERT INTO `Employee` (`id`, `name`, `role`)
VALUES
    (1, 'Pankaj', 'CEO'),
    (2, 'David', 'Manager');

commit;
```

## Spring DataSource MVC Project

Create a Spring MVC Project in the Spring Tool Suite so that our spring application skeleton code is ready.

Once we will be done with our implementation, our project structure will look like below image.



## Spring JDBC and Jackson Dependencies

We will have to add Spring JDBC, **Jackson** and MySQL Database driver as the dependencies in the pom.xml file. My final pom.xml file looks like below.

```
<?xml version="1.0" encoding="UTF-8"?>
<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/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.journaldev.spring</groupId>
  <artifactId>SpringDataSource</artifactId>
  <name>SpringDataSource</name>
  <packaging>war</packaging>
  <version>1.0.0-BUILD-SNAPSHOT</version>
  <properties>
    <java-version>1.6</java-version>
    <org.springframework-version>4.0.2.RELEASE</org.springframework-
version>
    <org.aspectj-version>1.7.4</org.aspectj-version>
    <org.slf4j-version>1.7.5</org.slf4j-version>
    <jackson.databind-version>2.2.3</jackson.databind-version>
  </properties>
  <dependencies>
    <!-- Spring JDBC Support -->
    <dependency>
```

If you are not familiar with Rest in Spring, please read [Spring Restful Webservice Example](#).

## Model Class

Our Employee bean that is modeled after Employee table, looks like below.

```
package com.journaldev.spring.jdbc.model;

import java.io.Serializable;

public class Employee implements Serializable{

    private static final long serialVersionUID = -7788619177798333712L;

    private int id;
    private String name;
    private String role;

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
```

## Spring Controller Class

Our simple controller class looks like below.

```
package com.journaldev.spring.jdbc.controller;

import java.util.ArrayList;
import java.util.List;
import java.util.Map;

import javax.sql.DataSource;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;

import com.journaldev.spring.jdbc.model.Employee;

/**
 * Handles requests for the Employee JDBC Service.
```

Important Points about the Controller class are:

- DataSource will be wired by Spring Bean configuration with name dbDataSource.
- We are using JdbcTemplate to avoid common errors such as resource leak and remove JDBC boiler plate code.
- URI to retrieve the list of Employee will be http://[host]:[port]/SpringDataSource/rest/emps
- We are using @ResponseBody to send the list of Employee objects as response, Spring will take care of converting it to JSON.

## Spring Bean Configuration

There are two ways through which we can JNDI lookup and wire it to the Controller DataSource, my spring bean configuration file contains both of them but one of them is commented. You can switch between these and the response will be the same.

1. Using jee namespace tag to perform the JNDI lookup and configure it as a Spring Bean. We also need to include jee namespace and schema definition in this case.
2. Creating a bean of type `org.springframework.jndi.JndiObjectFactoryBean` by passing the JNDI context name. `jndiName` is a required parameter for this configuration.

My spring bean configuration file looks like below.

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

    <!-- DispatcherServlet Context: defines this servlet's request-processing
    infrastructure -->

    <!-- Enables the Spring MVC @Controller programming model -->
    <annotation-driven />

    <!-- Handles HTTP GET requests for /resources/** by efficiently serving
```

## Tomcat DataSource JNDI Configuration

Now that we are done with our project, the final part is to do the JNDI configuration in Tomcat container to create the JNDI resource.

```
<Resource name="jdbc/TestDB"
```

```
global="jdbc/TestDB"
auth="Container"
type="javax.sql.DataSource"
driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:3306/TestDB"
username="pankaj"
password="pankaj123"

maxActive="100"
maxIdle="20"
minIdle="5"
maxWait="10000"/>
```

Add above configuration in the GlobalNamingResources section of the server.xml file.

```
<ResourceLink name="jdbc/MyLocalDB"
              global="jdbc/TestDB"
              auth="Container"
              type="javax.sql.DataSource" />
```

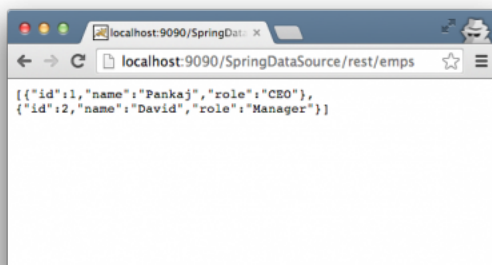
We also need to create the Resource Link to use the JNDI configuration in our application, best way to add it in the server context.xml file.

Notice that ResourceLink name should be matching with the JNDI context name we are using in our application. Also make sure MySQL jar is present in the tomcat lib directory, otherwise tomcat will not be able to create the MySQL database connection pool.

## Running the Spring DataSource JNDI Sample Project

Our project and server configuration is done and we are ready to test it. Export the project as WAR file and place it in the tomcat deployment directory.

The JSON response for the Rest call is shown in the below image.



That's all for the Spring integration with servlet container JNDI context, download the sample project from the below link and play around with it to learn more.

[Download Spring DataSource JNDI Project](#)

[« PREVIOUS](#)[Spring JDBC Example](#)[NEXT »](#)[Spring Transaction Management Example JDBC](#)

#### 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: [SPRING](#)

## Comments

#### Navin says

JANUARY 10, 2018 AT 9:10 PM

Dude, always i see jndi url...will it be only in the resources file|i mean application server's configuration side?]

=====

it won't be there in the properties file? and won't they get added in the application? as i see nothing is been passed to the configuration file

=====

Is it like this...since it is a sample application you have avoided to provide it?..In real scenarios will it be the same like this?

[Reply](#)

**Pankaj says**

JANUARY 10, 2018 AT 9:54 PM

I have provided JNDI configuration in "Tomcat DataSource JNDI Configuration" section, please check.

[Reply](#)**Venkateswarlu says**

JULY 20, 2017 AT 7:35 AM

Hi Pankaj Kumar,

I tried spring integration with Hibernate JNDI connection pooling but facing exceptions

Could you please help me.

Tomcat version-7.0.23

This is my pom.xml

4.0.0

com.webappdemo

mywebtest

war

1.0-SNAPSHOT

mywebtest Maven Webapp

<http://maven.apache.org>

4.2.0.RELEASE

4.3.10.Final

5.1.31

org.springframework

spring-core

org.springframework

spring-web

org.springframework

spring-webmvc

org.springframework

spring-tx

org.springframework

spring-orm

org.springframework

spring-context

org.hibernate

hibernate-core

org.hibernate

validation-api

org.hibernate

hibernate-validator

org.hibernate

mysql

mysql-connector-java

```

$mysql.connector.version
javax.servlet
javax.servlet-api
3.1.0
javax.servlet.jsp
javax.servlet.jsp-api
2.3.1
javax.servlet
jstl
1.2
org.apache.maven.plugins
maven-compiler-plugin
3.2
1.7
1.7
org.apache.maven.plugins
maven-war-plugin
2.4
src/main/webapp
mywebtest
false
mywebtest
Please find below exception
Jul 20, 2017 7:42:04 PM org.apache.catalina.core.AprLifecycleListener init
INFO: The APR based Apache Tomcat Native library which allows optimal performance in production
environments was not found on the java.library.path: C:\Program
Files\Java\jdk1.8.0_111\bin;C:\Windows\Sun\Java\bin;C:\Windows\system32;C:\Windows;C:/Program
Files/Java/jdk1.8.0_111/bin/.../jre/bin/server;C:/Program
Files/Java/jdk1.8.0_111/bin/.../jre/bin;C:/Program
Files/Java/jdk1.8.0_111/bin/.../jre/lib/amd64;C:\oracle\product\11.2.0\server\bin;C:\Windows\system32;C:\Windows;C:\Win
Files\Broadcom\Broadcom 802.11\Driver;C:\Program Files (x86)\Common
Files\lenovo\easyplussdk\bin;C:\Program Files (x86)\Skype\Phone\;C:\Program
Files\Java\jdk1.8.0_111\bin;;D:\java softwares\jars&zip\apache-maven-
3.2.3\bin;;;C:\Users\Venky\Desktop\eclipse;;
Jul 20, 2017 7:42:04 PM org.apache.tomcat.util.digester.SetPropertiesRule begin
WARNING: [SetPropertiesRule][Server/Service/Engine/Host/Context] Setting property 'source' to
'org.eclipse.jst.jee.server:mywebtest' did not find a matching property.
Jul 20, 2017 7:42:04 PM org.apache.coyote.AbstractProtocol init
INFO: Initializing ProtocolHandler ["http-bio-8081"]
Jul 20, 2017 7:42:04 PM org.apache.coyote.AbstractProtocol init
INFO: Initializing ProtocolHandler ["ajp-bio-8009"]
Jul 20, 2017 7:42:04 PM org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 911 ms
Jul 20, 2017 7:42:04 PM org.apache.catalina.core.StandardService startInternal
INFO: Starting service Catalina
Jul 20, 2017 7:42:04 PM org.apache.catalina.core.StandardEngine startInternal
INFO: Starting Servlet Engine: Apache Tomcat/7.0.23
Jul 20, 2017 7:42:05 PM org.apache.catalina.util.SessionIdGenerator createSecureRandom
INFO: Creation of SecureRandom instance for session ID generation using [SHA1PRNG] took [186]
milliseconds.
Jul 20, 2017 7:42:05 PM org.apache.catalina.loader.WebappClassLoader validateJarFile
INFO:
validateJarFile(F:\workspace\spring\metadata\plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\mywebtest\WEB-
INF\lib\javax.servlet-api-3.1.0.jar) - jar not loaded. See Servlet Spec 2.3, section 9.7.2. Offending class:
javax/servlet/Servlet.class
Jul 20, 2017 7:42:08 PM org.apache.catalina.core.ContainerBase startInternal

```



```
SEVERE: A child container failed during start
java.util.concurrent.ExecutionException: org.apache.catalina.LifecycleException: Failed to start
component [StandardEngine[Catalina].StandardHost[localhost].StandardContext[/mywebtest]]
at java.util.concurrent.FutureTask.report(FutureTask.java:122)
at java.util.concurrent.FutureTask.get(FutureTask.java:192)
at org.apache.catalina.core.ContainerBase.startInternal(ContainerBase.java:1100)
at org.apache.catalina.core.StandardHost.startInternal(StandardHost.java:782)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150)
at org.apache.catalina.core.ContainerBase$StartChild.call(ContainerBase.java:1525)
at org.apache.catalina.core.ContainerBase$StartChild.call(ContainerBase.java:1515)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:745)
Caused by: org.apache.catalina.LifecycleException: Failed to start component
[StandardEngine[Catalina].StandardHost[localhost].StandardContext[/mywebtest]]
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:154)
... 6 more
Caused by: org.apache.tomcat.util.bcel.classfile.ClassFormatException: Invalid byte tag in constant pool:
18
at org.apache.tomcat.util.bcel.classfile.Constant.readConstant(Constant.java:131)
at org.apache.tomcat.util.bcel.classfile.ConstantPool.(ConstantPool.java:60)
at org.apache.tomcat.util.bcel.classfile.ClassParser.readConstantPool(ClassParser.java:209)
at org.apache.tomcat.util.bcel.classfile.ClassParser.parse(ClassParser.java:119)
at org.apache.catalina.startup.ContextConfig.processAnnotationsStream(ContextConfig.java:1949)
at org.apache.catalina.startup.ContextConfig.processAnnotationsJar(ContextConfig.java:1840)
at org.apache.catalina.startup.ContextConfig.processAnnotationsUrl(ContextConfig.java:1808)
at org.apache.catalina.startup.ContextConfig.processAnnotations(ContextConfig.java:1794)
at org.apache.catalina.startup.ContextConfig.webConfig(ContextConfig.java:1214)
at org.apache.catalina.startup.ContextConfig.configureStart(ContextConfig.java:828)
at org.apache.catalina.startup.ContextConfig.lifecycleEvent(ContextConfig.java:302)
at org.apache.catalina.util.LifecycleSupport.fireLifecycleEvent(LifecycleSupport.java:119)
at org.apache.catalina.util.LifecycleBase.fireLifecycleEvent(LifecycleBase.java:90)
at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5148)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150)
... 6 more
Jul 20, 2017 7:42:08 PM org.apache.catalina.core.ContainerBase startInternal
SEVERE: A child container failed during start
java.util.concurrent.ExecutionException: org.apache.catalina.LifecycleException: Failed to start
component [StandardEngine[Catalina].StandardHost[localhost]]
at java.util.concurrent.FutureTask.report(FutureTask.java:122)
at java.util.concurrent.FutureTask.get(FutureTask.java:192)
at org.apache.catalina.core.ContainerBase.startInternal(ContainerBase.java:1100)
at org.apache.catalina.core.StandardEngine.startInternal(StandardEngine.java:291)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150)
at org.apache.catalina.core.StandardService.startInternal(StandardService.java:443)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150)
at org.apache.catalina.core.StandardServer.startInternal(StandardServer.java:727)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150)
at org.apache.catalina.startup.Catalina.start(Catalina.java:624)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at 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 org.apache.catalina.startup.Bootstrap.start(Bootstrap.java:322)
at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:450)
```

```
Caused by: org.apache.catalina.LifecycleException: Failed to start component
[StandardEngine[Catalina].StandardHost[localhost]]
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:154)
at org.apache.catalina.core.ContainerBase$StartChild.call(ContainerBase.java:1525)
at org.apache.catalina.core.ContainerBase$StartChild.call(ContainerBase.java:1515)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:745)
Caused by: org.apache.catalina.LifecycleException: A child container failed during start
at org.apache.catalina.core.ContainerBase.startInternal(ContainerBase.java:1108)
at org.apache.catalina.core.StandardHost.startInternal(StandardHost.java:782)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150)
... 6 more
Jul 20, 2017 7:42:08 PM org.apache.catalina.startup.Catalina start
SEVERE: Catalina.start:
org.apache.catalina.LifecycleException: Failed to start component [StandardServer[8005]]
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:154)
at org.apache.catalina.startup.Catalina.start(Catalina.java:624)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at 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 org.apache.catalina.startup.Bootstrap.start(Bootstrap.java:322)
at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:450)
Caused by: org.apache.catalina.LifecycleException: Failed to start component [StandardService[Catalina]]
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:154)
at org.apache.catalina.core.StandardServer.startInternal(StandardServer.java:727)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150)
... 7 more
Caused by: org.apache.catalina.LifecycleException: Failed to start component [StandardEngine[Catalina]]
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:154)
at org.apache.catalina.core.StandardService.startInternal(StandardService.java:443)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150)
... 9 more
Caused by: org.apache.catalina.LifecycleException: A child container failed during start
at org.apache.catalina.core.ContainerBase.startInternal(ContainerBase.java:1108)
at org.apache.catalina.core.StandardEngine.startInternal(StandardEngine.java:291)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150)
... 11 more
Jul 20, 2017 7:42:08 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 3491 ms
Reply
```

**mike says**

SEPTEMBER 29, 2016 AT 10:01 PM

Thanks. This is what I was looking for. its working out well. Am having a problem with the internalviewresolver and also the resolver for static resources.

[Reply](#)**Sourabh says**

JANUARY 5, 2016 AT 6:06 AM

Hi Pankaj,  
Thanks for the help and information.Steps mentioned are quite self-explanatory.  
[Reply](#)

**Anon says**

OCTOBER 13, 2015 AT 12:51 PM

In some places you have  
jdbc/TestDB  
while in other places you have  
jdbc/MyLocalDB  
Is this not a mistake?  
[Reply](#)

**Pankaj says**

OCTOBER 14, 2015 AT 5:21 AM

jdbc/TestDB is the Resource and defined in server.xml whereas jdbc/MyLocalDB is resource link and defined in context.xml. There is no mistake, you can keep any names for these, in code we use resource link configured.  
[Reply](#)

**David says**

AUGUST 20, 2015 AT 7:41 PM

I can run the downloaded project without any issue except the json data is not shown on the browser. Instead,it prompts me the location for saving the data. May I ask what could go wrong in my project ?  
Your prompt reply is greatly appreciated !  
[Reply](#)

**Kanika Sud says**

AUGUST 19, 2015 AT 4:01 AM

Could you tell us the configuration details for oracle database? in JNDI?  
[Reply](#)

**dilhar says**

JUNE 28, 2015 AT 3:12 AM

hi sir,  
Why are you using "setDataSource" here? It is not mandatory right?  
I'm bit confused about spring autowiring feature.Can spring autowire/inject non-component objects(like javax.sql.DataSource) ?  
please can you explain? thanks..  
[Reply](#)

**Vipin says**

NOVEMBER 27, 2014 AT 1:33 PM

Hi Pankaj,

First of all thanks for your great tutorials, really helpful.

I have one particular question, and want to know if that is possible to do or not. I want to use Spring MVC framework in Tomcat. In Tomcat i would be configuring the DataSource (and this data source would be in web.xml , context.xml). Now, using @Resource annotation, can't it be injected by Spring framework? Do we still need to use the JNDI lookup? In other words, is it possible to use the DataSource that we define in Tomcat to be used by Spring MVC framework without doing JNDI lookup?

Please clarify.

If you want me to provide the source code as to what I have done, please let me know.

Regards,

Vipin

[Reply](#)

**kim young Choi says**

NOVEMBER 10, 2014 AT 5:07 PM

I want to use encrypted username and password in server.xml

How can I change username and password in my FactoryBean?

for example

```
public class MyJndiObjectFactoryBean extend JndiObjectFactoryBean {  
    // username and password has encrypted..  
    // so It will be decryption  
    // how ??  
}
```

Thank you.

[Reply](#)

**Gibran Castillo says**

OCTOBER 20, 2014 AT 11:53 AM

Really good. Do you have a tutorial similar like this one, with the addition of Hibernate and JTA. Replacing Spring JDBC with JTA and Hibernate.

[Reply](#)

**mhewedy says**

OCTOBER 19, 2014 AT 5:47 AM

Why not using DI here??

```
JdbcTemplate jdbcTemplate = new JdbcTemplate(dataSource);
```

[Reply](#)

**rajan says**

SEPTEMBER 2, 2014 AT 8:16 PM

hello sir,

please help me this error coming, I did as per all the configuration

ERROR: org.springframework.web.servlet.DispatcherServlet – Context initialization failed

org.springframework.beans.factory.BeanCreationException: Error creating bean with name

'dbDataSource' defined in ServletContext resource [/WEB-INF/spring/appServlet/servlet-context.xml]:

Invocation of init method failed; nested exception is javax.naming.NameNotFoundException: Name jdbc

is not bound in this Context

```
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:304)
at
org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:228)
at
org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:300)
at
org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:195)
at
org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.java:681)
at
org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:760)
at
org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:482)
at
org.springframework.web.servlet.FrameworkServlet.configureAndRefreshWebApplicationContext(FrameworkServlet.java:658)
at
org.springframework.web.servlet.FrameworkServlet.createWebApplicationContext(FrameworkServlet.java:624)
at
org.springframework.web.servlet.FrameworkServlet.createWebApplicationContext(FrameworkServlet.java:672)
at
org.springframework.web.servlet.FrameworkServlet.initWebApplicationContext(FrameworkServlet.java:543)
at org.springframework.web.servlet.FrameworkServlet.initServletBean(FrameworkServlet.java:484)
at org.springframework.web.servlet.HttpServletBean.init(HttpServletBean.java:136)
at javax.servlet.GenericServlet.init(GenericServlet.java:158)
at org.apache.catalina.core.StandardWrapper.initServlet(StandardWrapper.java:1108)
at org.apache.catalina.core.StandardWrapper.loadServlet(StandardWrapper.java:1062)
at org.apache.catalina.core.StandardWrapper.load(StandardWrapper.java:971)
at org.apache.catalina.core.StandardContext.loadOnStartup(StandardContext.java:4562)
at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:4845)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:140)
at org.apache.catalina.core.ContainerBase.startInternal(ContainerBase.java:1028)
at org.apache.catalina.core.StandardHost.startInternal(StandardHost.java:773)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:140)
at org.apache.catalina.core.ContainerBase.startInternal(ContainerBase.java:1028)
at org.apache.catalina.core.StandardEngine.startInternal(StandardEngine.java:278)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:140)
at org.apache.catalina.core.StandardService.startInternal(StandardService.java:429)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:140)
at org.apache.catalina.core.StandardServer.startInternal(StandardServer.java:662)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:140)
at org.apache.catalina.startup.Catalina.start(Catalina.java:592)
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.apache.catalina.startup.Bootstrap.start(Bootstrap.java:290)
at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:418)
Caused by: javax.naming.NameNotFoundException: Name jdbc is not bound in this Context
at org.apache.naming.NamingContext.lookup(NamingContext.java:775)
```

```
at org.apache.naming.NamingContext.lookup(NamingContext.java:145)
at org.apache.naming.NamingContext.lookup(NamingContext.java:786)
at org.apache.naming.NamingContext.lookup(NamingContext.java:145)
at org.apache.naming.NamingContext.lookup(NamingContext.java:786)
at org.apache.naming.NamingContext.lookup(NamingContext.java:158)
at org.apache.naming.SelectorContext.lookup(SelectorContext.java:161)
at javax.naming.InitialContext.lookup(Unknown Source)
at org.springframework.jndi.JndiTemplate$1.doInContext(JndiTemplate.java:155)
at org.springframework.jndi.JndiTemplate.execute(JndiTemplate.java:87)
at org.springframework.jndi.JndiTemplate.lookup(JndiTemplate.java:152)
at org.springframework.jndi.JndiTemplate.lookup(JndiTemplate.java:179)
at org.springframework.jndi.JndiLocatorSupport.lookup(JndiLocatorSupport.java:95)
at org.springframework.jndi.JndiObjectLocator.lookup(JndiObjectLocator.java:106)
at org.springframework.jndi.JndiObjectFactoryBean.lookupWithFallback(JndiObjectFactoryBean.java:231)
at org.springframework.jndi.JndiObjectFactoryBean.afterPropertiesSet(JndiObjectFactoryBean.java:217)
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.invokeInitMethods(AbstractAutowireCapableBeanFactory.java:155)
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initializeBean(AbstractAutowireCapableBeanFactory.java:155)
... 39 more
Sep 03, 2014 8:39:29 AM org.apache.catalina.core.ApplicationContext log
SEVERE: StandardWrapper.Throwable
org.springframework.beans.factory.BeanCreationException: Error creating bean with name
'dbDataSource' defined in ServletContext resource [/WEB-INF/spring/appServlet/servlet-context.xml]:
Invocation of init method failed; nested exception is javax.naming.NameNotFoundException: Name jdbc
is not bound in this Context
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initializeBean(AbstractAutowireCapableBeanFactory.java:155)
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:534)
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:475)
at
org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:304)
at
org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:228)
at
org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:300)
at
org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:195)
at
org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.java:681)
at
org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:760)
at
org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:482)
at
org.springframework.web.servlet.FrameworkServlet.configureAndRefreshWebApplicationContext(FrameworkServlet.java:658)
at
org.springframework.web.servlet.FrameworkServlet.createWebApplicationContext(FrameworkServlet.java:624)
at
org.springframework.web.servlet.FrameworkServlet.createWebApplicationContext(FrameworkServlet.java:672)
at
org.springframework.web.servlet.FrameworkServlet.initWebApplicationContext(FrameworkServlet.java:543)
at org.springframework.web.servlet.FrameworkServlet.initServletBean(FrameworkServlet.java:484)
at org.springframework.web.servlet.HttpServletBean.init(HttpServletBean.java:136)
```

```

at javax.servlet.GenericServlet.init(GenericServlet.java:158)
at org.apache.catalina.core.StandardWrapper.initServlet(StandardWrapper.java:1108)
at org.apache.catalina.core.StandardWrapper.loadServlet(StandardWrapper.java:1062)
at org.apache.catalina.core.StandardWrapper.load(StandardWrapper.java:971)
at org.apache.catalina.core.StandardContext.loadOnStartup(StandardContext.java:4562)
at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:4845)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:140)
at org.apache.catalina.core.ContainerBase.startInternal(ContainerBase.java:1028)
at org.apache.catalina.core.StandardHost.startInternal(StandardHost.java:773)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:140)
at org.apache.catalina.core.ContainerBase.startInternal(ContainerBase.java:1028)
at org.apache.catalina.core.StandardEngine.startInternal(StandardEngine.java:278)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:140)
at org.apache.catalina.core.StandardService.startInternal(StandardService.java:429)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:140)
at org.apache.catalina.core.StandardServer.startInternal(StandardServer.java:662)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:140)
at org.apache.catalina.startup.Catalina.start(Catalina.java:592)
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.apache.catalina.startup.Bootstrap.start(Bootstrap.java:290)
at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:418)
Caused by: javax.naming.NameNotFoundException: Name jdbc is not bound in this Context
at org.apache.naming.NamingContext.lookup(NamingContext.java:775)
at org.apache.naming.NamingContext.lookup(NamingContext.java:145)
at org.apache.naming.NamingContext.lookup(NamingContext.java:786)
at org.apache.naming.NamingContext.lookup(NamingContext.java:145)
at org.apache.naming.NamingContext.lookup(NamingContext.java:786)
at org.apache.naming.NamingContext.lookup(NamingContext.java:158)
at org.apache.naming.SelectorContext.lookup(SelectorContext.java:161)
at javax.naming.InitialContext.lookup(Unknown Source)
at org.springframework.jndi.JndiTemplate$1.doInContext(JndiTemplate.java:155)
at org.springframework.jndi.JndiTemplate.execute(JndiTemplate.java:87)
at org.springframework.jndi.JndiTemplate.lookup(JndiTemplate.java:152)
at org.springframework.jndi.JndiTemplate.lookup(JndiTemplate.java:179)
at org.springframework.jndi.JndiLocatorSupport.lookup(JndiLocatorSupport.java:95)
at org.springframework.jndi.JndiObjectLocator.lookup(JndiObjectLocator.java:106)
at org.springframework.jndi.JndiObjectFactoryBean.lookupWithFallback(JndiObjectFactoryBean.java:231)
at org.springframework.jndi.JndiObjectFactoryBean.afterPropertiesSet(JndiObjectFactoryBean.java:217)
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.invokeInitMethods(AbstractAutowireCapableBeanFactory.java:15
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initializeBean(AbstractAutowireCapableBeanFactory.java:15
... 39 more
Sep 03, 2014 8:39:29 AM org.apache.catalina.core.StandardContext loadOnStartup
SEVERE: Servlet /SpringDataSource threw load() exception
javax.naming.NameNotFoundException: Name jdbc is not bound in this Context
at org.apache.naming.NamingContext.lookup(NamingContext.java:775)
at org.apache.naming.NamingContext.lookup(NamingContext.java:145)
at org.apache.naming.NamingContext.lookup(NamingContext.java:786)
at org.apache.naming.NamingContext.lookup(NamingContext.java:145)
at org.apache.naming.NamingContext.lookup(NamingContext.java:786)
at org.apache.naming.NamingContext.lookup(NamingContext.java:158)
at org.apache.naming.SelectorContext.lookup(SelectorContext.java:161)

```

```
at javax.naming.InitialContext.lookup(Unknown Source)
at org.springframework.jndi.JndiTemplate$1.doInContext(JndiTemplate.java:155)
at org.springframework.jndi.JndiTemplate.execute(JndiTemplate.java:87)
at org.springframework.jndi.JndiTemplate.lookup(JndiTemplate.java:152)
at org.springframework.jndi.JndiTemplate.lookup(JndiTemplate.java:179)
at org.springframework.jndi.JndiLocatorSupport.lookup(JndiLocatorSupport.java:95)
at org.springframework.jndi.JndiObjectLocator.lookup(JndiObjectLocator.java:106)
at org.springframework.jndi.JndiObjectFactoryBean.lookupWithFallback(JndiObjectFactoryBean.java:231)
at org.springframework.jndi.JndiObjectFactoryBean.afterPropertiesSet(JndiObjectFactoryBean.java:217)
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.invokeInitMethods(AbstractAutowireCapableBeanFactory.java:155)
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initializeBean(AbstractAutowireCapableBeanFactory.java:155)
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:534)
at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:475)
at
org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:304)
at
org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:228)
at
org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:300)
at
org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:195)
at
org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.java:681)
at
org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:760)
at
org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:482)
at
org.springframework.web.servlet.FrameworkServlet.configureAndRefreshWebApplicationContext(FrameworkServlet.java:658)
at
org.springframework.web.servlet.FrameworkServlet.createWebApplicationContext(FrameworkServlet.java:624)
at
org.springframework.web.servlet.FrameworkServlet.createWebApplicationContext(FrameworkServlet.java:672)
at
org.springframework.web.servlet.FrameworkServlet.initWebApplicationContext(FrameworkServlet.java:543)
at org.springframework.web.servlet.FrameworkServlet.initServletBean(FrameworkServlet.java:484)
at org.springframework.web.servlet.HttpServletBean.init(HttpServletBean.java:136)
at javax.servlet.GenericServlet.init(GenericServlet.java:158)
at org.apache.catalina.core.StandardWrapper.initServlet(StandardWrapper.java:1108)
at org.apache.catalina.core.StandardWrapper.loadServlet(StandardWrapper.java:1062)
at org.apache.catalina.core.StandardWrapper.load(StandardWrapper.java:971)
at org.apache.catalina.core.StandardContext.loadOnStartup(StandardContext.java:4562)
at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:4845)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:140)
at org.apache.catalina.core.ContainerBase.startInternal(ContainerBase.java:1028)
at org.apache.catalina.core.StandardHost.startInternal(StandardHost.java:773)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:140)
at org.apache.catalina.core.ContainerBase.startInternal(ContainerBase.java:1028)
at org.apache.catalina.core.StandardEngine.startInternal(StandardEngine.java:278)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:140)
at org.apache.catalina.core.StandardService.startInternal(StandardService.java:429)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:140)
```



```

at org.apache.catalina.core.StandardServer.startInternal(StandardServer.java:662)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:140)
at org.apache.catalina.startup.Catalina.start(Catalina.java:592)
at sun.reflect.NativeMethodAccessorImpl.invoke(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.apache.catalina.startup.Bootstrap.start(Bootstrap.java:290)
at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:418)

```

[Reply](#)

**Pankaj says**

SEPTEMBER 2, 2014 AT 8:50 PM

Please check your JNDI configurations in the server, for this example it should be "jdbc/MyLocalDB" in context.xml file in Tomcat. If you are using some other container, there may be different way to do it other than "java:comp/env/jdbc/MyLocalDB".

Check your server logs to see if there is any error in creating the JNDI DataSource? Make sure you have Database driver jar in the server lib directory.

[Reply](#)

**rajan says**

SEPTEMBER 2, 2014 AT 11:39 AM

iam getting this error

```

org.springframework.beans.factory.BeanCreationException: Error creating bean with name
'dbDataSource' defined in ServletContext resource [/WEB-INF/spring/appServlet/servlet-context.xml]:
Invocation of init method failed; nested exception is javax.naming.NameNotFoundException: Name jdbc
is not bound in this Context
Caused by: javax.naming.NameNotFoundException: Name jdbc is not bound in this Context
org.springframework.web.servlet.DispatcherServlet - Context initialization failed
org.springframework.beans.factory.BeanCreationException: Error creating bean with name
'dbDataSource' defined in ServletContext resource [/WEB-INF/spring/appServlet/servlet-context.xml]:
Invocation of init method failed; nested exception is javax.naming.NameNotFoundException: Name jdbc
is not bound in this Context

```

[Reply](#)

**Sahil Taneja says**

MAY 27, 2014 AT 12:06 PM

```

INFO: Initializing Spring FrameworkServlet 'appServlet'
INFO : org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'appServlet': initialization
started
INFO : org.springframework.web.context.support.XmlWebApplicationContext - Refreshing
WebApplicationContext for namespace 'appServlet-servlet': startup date [Wed May 28 00:11:48 IST 2014];
parent: Root WebApplicationContext
INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions
from ServletContext resource [/WEB-INF/spring/appServlet/servlet-context.xml]
INFO : org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor - JSR-330
'javax.inject.Inject' annotation found and supported for autowiring
INFO : org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping -
Mapped "[[/rest/emps],methods=[GET],params=[],headers=[],consumes=[],produces=[],custom=[]]" onto

```

```
public java.util.List com.spring.jdbc.EmployeeController.getAllEmployees()
INFO : org.springframework.web.servlet.handler.SimpleUrlHandlerMapping - Mapped URL path
[/resources/**] onto handler
'org.springframework.web.servlet.resource.ResourceHttpRequestHandler#0'
ERROR: org.springframework.web.servlet.DispatcherServlet - Context initialization failed
org.springframework.beans.factory.BeanCreationException: Error creating bean with name
'dbDataSource' defined in ServletContext resource [/WEB-INF/spring/appServlet/servlet-context.xml]:
Invocation of init method failed; nested exception is javax.naming.NameNotFoundException: Name
[jdbc/test] is not bound in this Context. Unable to find [jdbc].
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:304)
at
org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:228)
at
org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:300)
at
org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:195)
```

[Reply](#)**Sahil Taneja says**

MAY 27, 2014 AT 12:27 PM

Server.xml  
servlet-context.xml  
context.xml  
please check why I'm getting exception...

[Reply](#)**Pankaj says**

MAY 27, 2014 AT 12:29 PM

From the logs, it looks like the issue with your JNDI settings for Database connection pool.  
Please check <https://www.journaldev.com/2513/tomcat-datasource-jndi-example-for-servlet-web-application>

[Reply](#)**Stephen Spalding says**

APRIL 24, 2014 AT 10:19 AM

Greetings!

I just wanted you to know that this is exactly what I needed today! Excellent post!

I used your example to connect to a SQL Server db using the net.sourceforge.jtds.jdbc.Driver class, and the one extra step that I had to do to get this to work in my STS is copy the jar file for this class, jtds-1.3.1.jar, in the folder:

C:\eclipse\springsource\vfabric-tc-server-developer-2.9.3.RELEASE\tomcat-7.0.42.A.RELEASE\lib

I had originally, as per your instructions of "make sure MySQL jar is present in the tomcat lib directory", copied the jtds-1.3.1.jar file to the folder:

C:\eclipse\springsource\vfabric-tc-server-developer-2.9.3.RELEASE\base-instance\lib

That didn't work, but copying the jar file to the \tomcat-7.0.42.A.RELEASE\lib folder did.

Thank you very much for this and all of your excellent posts!

-Stephen Spalding

St. Louis, MO

[Reply](#)

**Pankaj says**

APRIL 24, 2014 AT 11:57 AM

I am glad that it worked out well for 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

DOWNLOAD ANDROID APP



## 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 Interceptor](#)
- > [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 Initializr Web](#)
- > [Spring Boot Initializr IDE](#)
- > [Spring Boot Initializr CLI](#)
- > [Spring Boot Initializr 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](#)

---

### 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](#)

**Oracle Database Tools**

---

**Employer Interview Questions**

---

**Java Programming Courses**

---

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

**Learn Programming Online**

---

**Web Design Tutorials**

---

**Learn Java in 7 Days**

---

**Create Your Own Website**

---

**JSP Interview Questions**

