

```

1 package com.niit.config;
2
3 import java.util.Properties;
4 import javax.sql.DataSource;
5
6 import org.hibernate.SessionFactory;
7 import org.springframework.context.annotation.Bean;
8 import org.springframework.context.annotation.ComponentScan;
9 import org.springframework.context.annotation.Configuration;
10
11 import org.springframework.jdbc.datasource.DriverManagerDataSource;
12 import org.springframework.orm.hibernate4.LocalSessionFactoryBuilder;
13
14 @Configuration
15 @ComponentScan("com.niit")
16 public class DBConfig
17 {
18     public DataSource getH2DataSource()
19     {
20         DriverManagerDataSource dataSource = new DriverManagerDataSource();
21
22         dataSource.setDriverClassName("org.h2.Driver");
23         dataSource.setUrl("jdbc:h2:tcp://localhost/~Saurabh");
24         dataSource.setUsername("dteja");
25         dataSource.setPassword("dteja");
26
27         System.out.println("-----> Data Source is now operational <-----");
28         return dataSource;
29     }
30
31     @Bean(name="sessionFactory")
32     public SessionFactory getSessionFactory()
33     {
34         Properties hibernateProp = new Properties();
35
36         hibernateProp.setProperty("hibernate.hbm2ddl.auto","update");
37         hibernateProp.put("hibernate.dialect","org.hibernate.dialect.H2Dialect");
38
39         LocalSessionFactoryBuilder factoryBuilder = new LocalSessionFactoryBuilder(getH2DataSource());
40         factoryBuilder.addProperties(hibernateProp);
41
42         // The LocalSessionFactoryBuilder object which, is responsible for creating a bean object of the entire
43         // active session between the application's external user interface and the H2 Database in the back-end, is taking
44         // into its body, the H2 DataSource and the Hibernate orm properties of hbm auto update and the orm H2 dialect
45         // that will translate a java based Spring bean data object in the dialect of H2 Database via orm of Hibernate
46         // during an ongoing session in real-time!!
47
48         System.out.println(" * * * * * Session Factory Bean has just been initialized * * * * * ");
49         return factoryBuilder.buildSessionFactory();
50     }
51 }
52

```