

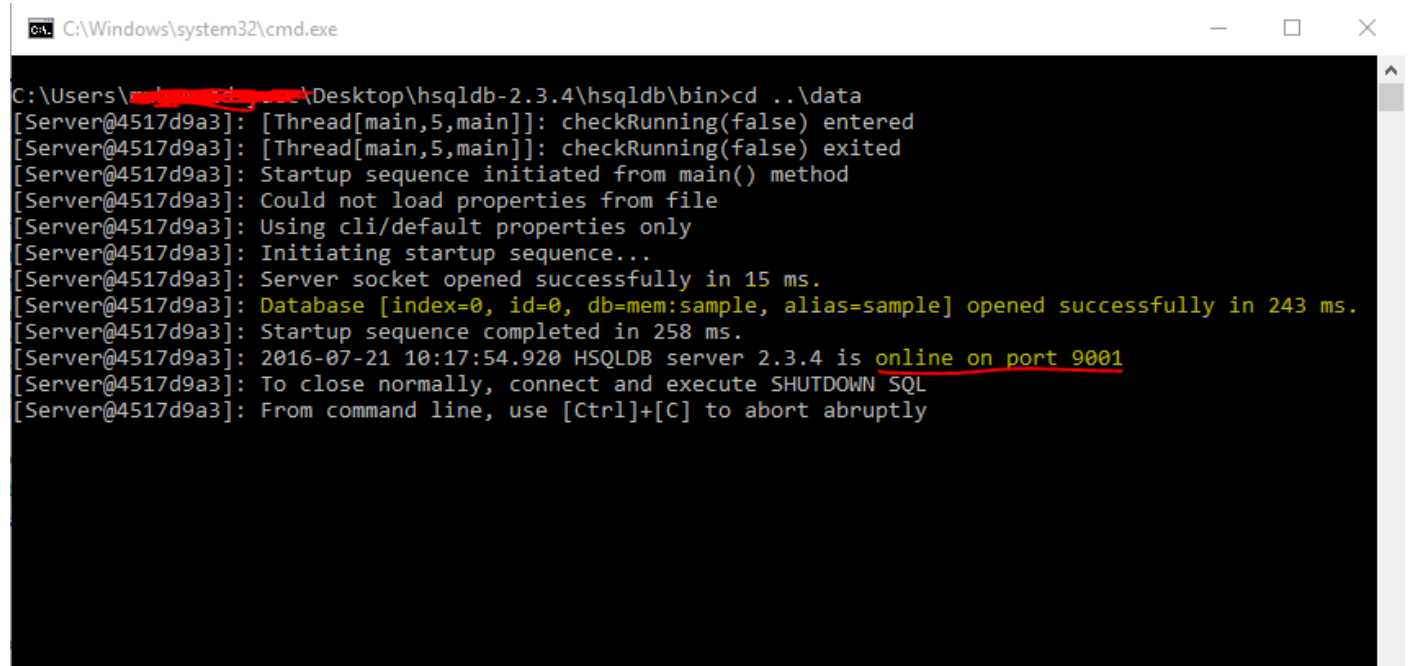
JUnit 4 Testing with DBUnit in Spring Hibernate Application

Hsqldb Connection

1. Download hsqldb.zip file from [here](#)
2. Extract zip file
3. Open bin folder in the hsqldb folder
4. Edit **runServer.bat** file as follows:

```
1 cd ..\data
2 @java -classpath ../lib/hsqldb.jar org.hsqldb.server.Server -database.0 mem:sample -dbname.0
   sample
```

6. After running this file you should see the command prompt as follows:



```
C:\Users\...\Desktop\hsqldb-2.3.4\hsqldb\bin>cd ..\data
[Server@4517d9a3]: [Thread[main,5,main]]: checkRunning(false) entered
[Server@4517d9a3]: [Thread[main,5,main]]: checkRunning(false) exited
[Server@4517d9a3]: Startup sequence initiated from main() method
[Server@4517d9a3]: Could not load properties from file
[Server@4517d9a3]: Using cli/default properties only
[Server@4517d9a3]: Initiating startup sequence...
[Server@4517d9a3]: Server socket opened successfully in 15 ms.
[Server@4517d9a3]: Database [index=0, id=0, db=mem:sample, alias=sample] opened successfully in 243 ms.
[Server@4517d9a3]: Startup sequence completed in 258 ms.
[Server@4517d9a3]: 2016-07-21 10:17:54.920 HSQLDB server 2.3.4 is online on port 9001
[Server@4517d9a3]: To close normally, connect and execute SHUTDOWN SQL
[Server@4517d9a3]: From command line, use [Ctrl]+[C] to abort abruptly
```

Spring applicationContext-test.xml File

```
1 <beans xmlns="http://www.springframework.org/schema/beans"
2     xmlns:context="http://www.springframework.org/schema/context"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://www.springframework.org/schema/beans
5         http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
6         http://www.springframework.org/schema/context
7         http://www.springframework.org/schema/context/spring-context-3.0.xsd">
8
9     <context:component-scan base-package="com.codesenior.telif.service.mail">
10         <context:exclude-filter type="annotation"
11             expression="org.springframework.stereotype.Controller"/>
12     </context:component-scan>
13
14     <bean id="dataSource"
```

```

15     class="com.mchange.v2.c3p0.ComboPooledDataSource" destroy-method="close">
16     <property name="driverClass" value="org.hsqldb.jdbcDriver"/>
17     <property name="jdbcUrl" value="jdbc:hsqldb:hsq://localhost/sample"/>
18     <property name="user" value="sa"/>
19     <property name="password" value=""/>
20     <property name="initialPoolSize" value="1"/>
21     <property name="maxPoolSize" value="1"/>
22     <property name="minPoolSize" value="1"/>
23 </bean>
24 <bean id="transactionManager"
25     class="org.springframework.orm.hibernate5.HibernateTransactionManager">
26     <property name="sessionFactory" ref="sessionFactory"/>
27 </bean>
28
29 <bean id="sessionFactory"
30     class="org.springframework.orm.hibernate5.LocalSessionFactoryBean">
31     <property name="dataSource" ref="dataSource"/>
32     <property name="packagesToScan" value="com.codesenior.telif.service.mail.model"/>
33     <property name="hibernateProperties">
34         <props>
35             <prop key="hibernate.hbm2ddl.auto">create</prop>
36             <prop key="hibernate.hbm2ddl.import_files">import.sql</prop>
37             <prop key="hibernate.dialect">org.hibernate.dialect.H2Dialect</prop>
38             <prop key="hibernate.show_sql">true</prop>
39             <prop key="hibernate.connection.CharSet">utf8</prop>
40             <prop key="hibernate.connection.characterEncoding">utf8</prop>
41             <prop key="hibernate.connection.useUnicode">true</prop>
42             <prop key="hibernate.show_sql">true</prop>
43             <prop key="hibernate.format_sql">true</prop>
44             <!--with hibernate 5, new_generator_mappings should be false-->
45             <prop key="hibernate.id.new_generator_mappings">false</prop>
46             <prop key="hibernate.cache.use_second_level_cache">true</prop>
47             <prop key="hibernate.cache.use_query_cache">true</prop>
48             <prop key="hibernate.generate_statistics">false</prop>
49             <prop key="javax.persistence.validation.mode">none</prop>
50             <prop key="hibernate.cache.region.factory_class">
51                 org.hibernate.cache.ehcache.SingletonEhCacheRegionFactory
52             </prop>
53         </props>
54     </property>
55 </bean>
56 </beans>

```

Spring JUnit4 Example

```

1  import org.dbunit.database.DatabaseDataSourceConnection;
2  import org.dbunit.database.IDatabaseConnection;
3  import org.dbunit.dataset.IDataset;
4  import org.dbunit.dataset.xml.FlatXmlDataSetBuilder;
5  import org.dbunit.operation.DatabaseOperation;
6  import org.junit.After;
7  import org.junit.Before;
8  import org.junit.Test;
9  import org.junit.runner.RunWith;
10 import org.springframework.beans.factory.annotation.Autowired;
11 import org.springframework.test.context.ContextConfiguration;
12 import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
13
14 import javax.sql.DataSource;
15 import java.io.InputStream;
16
17 /**
18  * DBUnitExample

```

```

19  */
20  @RunWith(SpringJUnit4ClassRunner.class)
21  @ContextConfiguration(locations={"classpath:applicationContext-test.xml"})
22  public class DBUnitExampleTest {
23      @Autowired
24      private DataSource dataSource;
25
26      @Before
27      public void init() throws Exception{
28          IDatabaseConnection connection=getConnection();
29          try{
30              DatabaseOperation.CLEAN_INSERT.execute(getConnection(), getDataSet());
31          }finally {
32              connection.close();
33          }
34      }
35
36      @After
37      public void after() throws Exception{
38          IDatabaseConnection connection=getConnection();
39          try{
40              DatabaseOperation.DELETE_ALL.execute(getConnection(), getDataSet());
41          }finally {
42              connection.close();
43          }
44      }
45      private IDatabaseConnection getConnection() throws Exception{
46
47          return new DatabaseDataSourceConnection(dataSource);
48      }
49
50      private IDataset getDataSet() throws Exception {
51          InputStream in = getClass().getResourceAsStream("/data/full.xml");
52          return new FlatXmlDataSetBuilder().build(in);
53      }
54      @Test
55      public void testHello() throws Exception{
56          System.out.println("hello");
57      }
58      @Test
59      public void testHello1() throws Exception{
60          System.out.println("hello1");
61      }
62  }

```