**Spring MVC and Spring JDBC Transaction Tutorial**

1- [Instroduction](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial" \l "a142298)

2- [Script to create tables](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial#a143131)

3- [Create Project](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial#a142301)

4- [Configure Maven](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial#a142305)

5- [Configure Spring](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial#a142303)

6- [Java Classes](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial#a167841)

7- [Views](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial#a167884)

8- [Run Application](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial#a242038)

9- [Appendix](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial#a167938)

1- Instroduction [▴](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial" \l "top)

The document is based on:

* **Spring MVC 4**
* **Spring JDBC**

2- Script to create tables [▴](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial" \l "top)

ORACLE:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | create table DEPARTMENT (        DEPT\_ID number(10,0) not null,        DEPT\_NAME varchar2(255 char) not null,        DEPT\_NO varchar2(20 char) not null unique,        LOCATION varchar2(255 char),        primary key (DEPT\_ID)    ); |

MySQL:

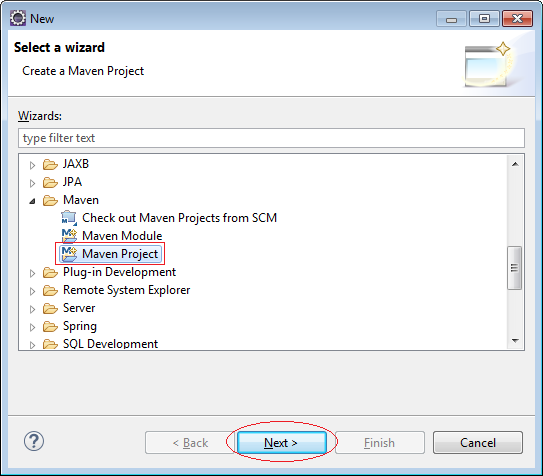
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | create table DEPARTMENT (    DEPT\_ID integer not null,    DEPT\_NAME varchar(255) not null,    DEPT\_NO varchar(20) not null,    LOCATION varchar(255),    primary key (DEPT\_ID),    unique (DEPT\_NO)  ); |

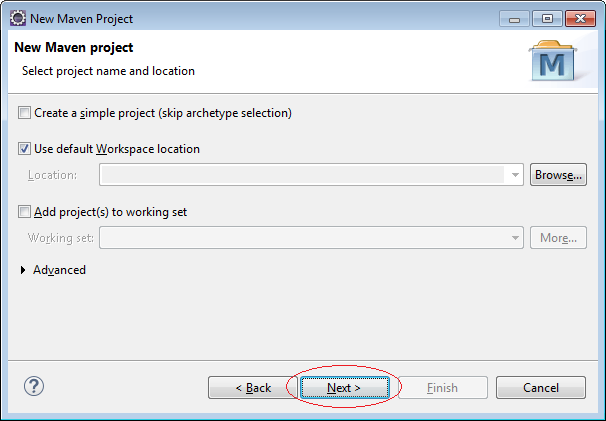
SQLServer:

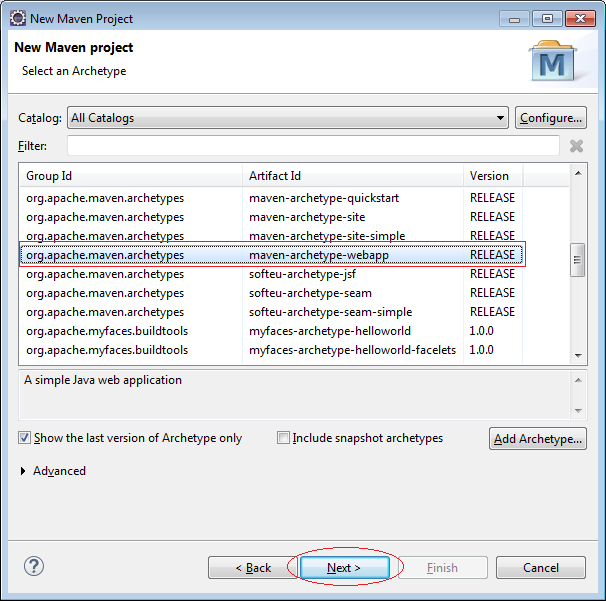
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | Create table DEPARTMENT (     DEPT\_ID int not null,     DEPT\_NAME varchar(255) not null,     DEPT\_NO varchar(20) not null,     LOCATION varchar(255),     primary key (DEPT\_ID),     unique (DEPT\_NO)  ); |

3- Create Project [▴](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial" \l "top)

* **File/New/Other..**

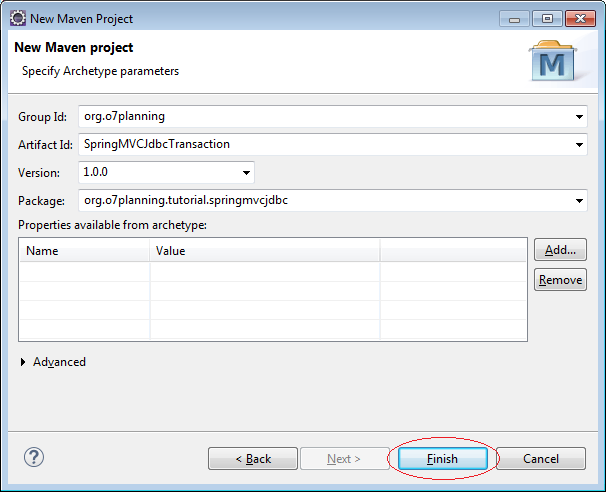




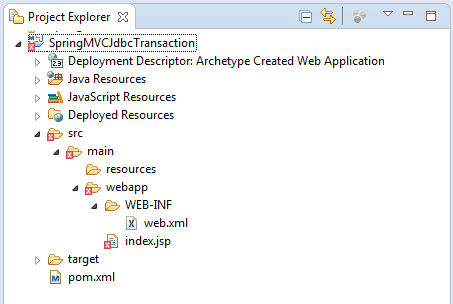


Enter:

* **Group Id**: org.o7planning
* **Artifact Id**: SpringMVCJdbcTransaction
* **Package**: org.o7planning.tutorial.springmvcjdbc

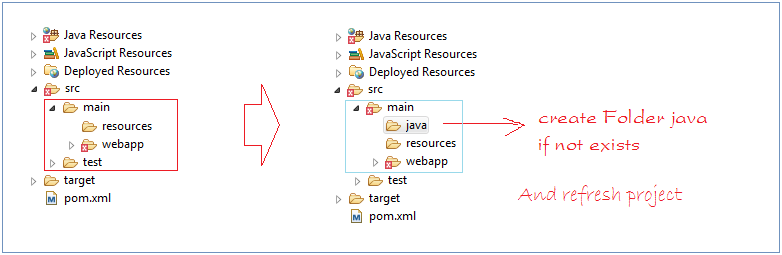


Project was created:



Do not worry with the error message when Project has been created. The reason is that you have not declared Servlet library.

**Note:**  
**Eclipse 4.4 (Luna)** create **Maven**project structure may be wrong. You need to check.

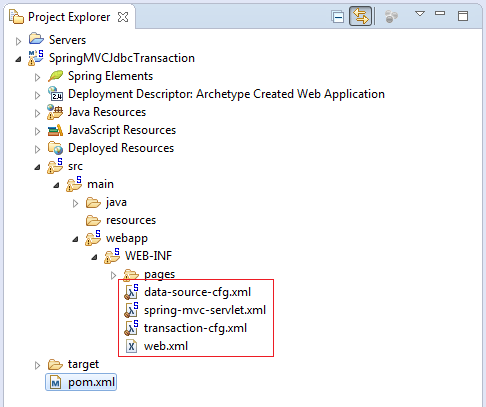


4- Configure Maven [▴](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial" \l "top)

* pom.xml

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136 | <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>org.o7planning</groupId>      <artifactId>SpringMVCJdbcTransaction</artifactId>      <packaging>war</packaging>      <version>1.0.0</version>      <name>SpringMVCJdbcTransaction Maven Webapp</name>      <url>[http://maven.apache.org](http://maven.apache.org/)</url>        <properties>          <java-version>1.7</java-version>      </properties>        <repositories>          <!-- Repository for ORACLE JDBC Driver -->          <repository>              <id>codelds</id>              <url><https://code.lds.org/nexus/content/groups/main-repo></url>          </repository>      </repositories>        <dependencies>            <dependency>              <groupId>junit</groupId>              <artifactId>junit</artifactId>              <version>3.8.1</version>              <scope>test</scope>          </dependency>            <!-- Servlet API -->          <!-- <http://mvnrepository.com/artifact/javax.servlet/javax.servlet-api> -->          <dependency>              <groupId>javax.servlet</groupId>              <artifactId>javax.servlet-api</artifactId>              <version>3.1.0</version>              <scope>provided</scope>          </dependency>            <!-- Jstl for jsp page -->          <!-- <http://mvnrepository.com/artifact/javax.servlet/jstl> -->          <dependency>              <groupId>javax.servlet</groupId>              <artifactId>jstl</artifactId>              <version>1.2</version>          </dependency>              <!-- JSP API -->          <!-- <http://mvnrepository.com/artifact/javax.servlet.jsp/jsp-api> -->          <dependency>              <groupId>javax.servlet.jsp</groupId>              <artifactId>jsp-api</artifactId>              <version>2.2</version>              <scope>provided</scope>          </dependency>            <!-- Spring dependencies -->          <!-- <http://mvnrepository.com/artifact/org.springframework/spring-core> -->          <dependency>              <groupId>org.springframework</groupId>              <artifactId>spring-core</artifactId>              <version>4.1.4.RELEASE</version>          </dependency>            <!-- <http://mvnrepository.com/artifact/org.springframework/spring-web> -->          <dependency>              <groupId>org.springframework</groupId>              <artifactId>spring-web</artifactId>              <version>4.1.4.RELEASE</version>          </dependency>            <!-- <http://mvnrepository.com/artifact/org.springframework/spring-webmvc> -->          <dependency>              <groupId>org.springframework</groupId>              <artifactId>spring-webmvc</artifactId>              <version>4.1.4.RELEASE</version>          </dependency>              <!-- <http://mvnrepository.com/artifact/org.springframework/spring-jdbc> -->          <dependency>              <groupId>org.springframework</groupId>              <artifactId>spring-jdbc</artifactId>              <version>4.1.4.RELEASE</version>          </dependency>            <!-- <http://mvnrepository.com/artifact/org.springframework/spring-tx> -->          <dependency>              <groupId>org.springframework</groupId>              <artifactId>spring-tx</artifactId>              <version>4.1.4.RELEASE</version>          </dependency>            <!-- MySQL JDBC driver -->          <!-- <http://mvnrepository.com/artifact/mysql/mysql-connector-java> -->          <dependency>              <groupId>mysql</groupId>              <artifactId>mysql-connector-java</artifactId>              <version>5.1.34</version>          </dependency>            <!-- Oracle JDBC driver -->          <dependency>              <groupId>com.oracle</groupId>              <artifactId>ojdbc6</artifactId>              <version>11.2.0.3</version>          </dependency>            <!-- SQLServer JDBC driver (JTDS) -->          <!-- <http://mvnrepository.com/artifact/net.sourceforge.jtds/jtds> -->          <dependency>              <groupId>net.sourceforge.jtds</groupId>              <artifactId>jtds</artifactId>              <version>1.3.1</version>          </dependency>       </dependencies>       <build>          <finalName>SpringMVCJdbcTransaction</finalName>          <plugins>                <!-- Config: Maven Tomcat Plugin -->              <!-- <http://mvnrepository.com/artifact/org.apache.tomcat.maven/tomcat7-maven-plugin> -->              <plugin>                  <groupId>org.apache.tomcat.maven</groupId>                  <artifactId>tomcat7-maven-plugin</artifactId>                  <version>2.2</version>                  <!-- Config: contextPath and Port (Default - /SpringMVCJdbcTransaction,8080) -->                  <!--                  <configuration>                      <path>/</path>                      <port>8899</port>                  </configuration>                  -->              </plugin>          </plugins>      </build>    </project> |

5- Configure Spring [▴](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial" \l "top)



* **web.xml**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | <web-app xmlns="<http://java.sun.com/xml/ns/javaee>"     xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"     xsi:schemaLocation="<http://java.sun.com/xml/ns/javaee>  <http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd>"     version="3.0">       <display-name>Archetype Created Web Application</display-name>     <servlet>         <servlet-name>spring-mvc</servlet-name>         <servlet-class>             org.springframework.web.servlet.DispatcherServlet         </servlet-class>         <load-on-startup>1</load-on-startup>     </servlet>       <servlet-mapping>         <servlet-name>spring-mvc</servlet-name>         <url-pattern>/</url-pattern>     </servlet-mapping>       <!-- Other XML Configuration -->     <!-- Load by Spring ContextLoaderListener -->     <context-param>         <param-name>contextConfigLocation</param-name>         <param-value>            /WEB-INF/data-source-cfg.xml,            /WEB-INF/transaction-cfg.xml          </param-value>     </context-param>         <!-- Spring ContextLoaderListener -->     <listener>     <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>     </listener>    </web-app> |

* **spring-mvc-servlet.xml**

|  |  |
| --- | --- |
| [?](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial)1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | <?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:p="<http://www.springframework.org/schema/p>"     xmlns:context="<http://www.springframework.org/schema/context>"     xmlns:mvc="<http://www.springframework.org/schema/mvc>"     xsi:schemaLocation="<http://www.springframework.org/schema/beans>  [http://www.springframework.org/schema/beans/spring-beans-4.1.xsd](http://www.springframework.org/schema/beans/spring-beans-4.1.xsd%C2%A0)  <http://www.springframework.org/schema/context>  [http://www.springframework.org/schema/context/spring-context-4.1.xsd](http://www.springframework.org/schema/context/spring-context-4.1.xsd%C2%A0)  <http://www.springframework.org/schema/mvc>  <http://www.springframework.org/schema/mvc/spring-mvc-4.1.xsd>">       <context:component-scan base-package="org.o7planning.tutorial.springmvcjdbc.\*" />       <context:annotation-config />       <bean         class="org.springframework.web.servlet.view.InternalResourceViewResolver">           <property name="prefix">             <value>/WEB-INF/pages/</value>         </property>           <property name="suffix">             <value>.jsp</value>         </property>       </bean>    </beans> |

Here I use the Oracle database. In case you use another database (**MySQL**, **SQLServer**) you can see the configuration in the appendix.

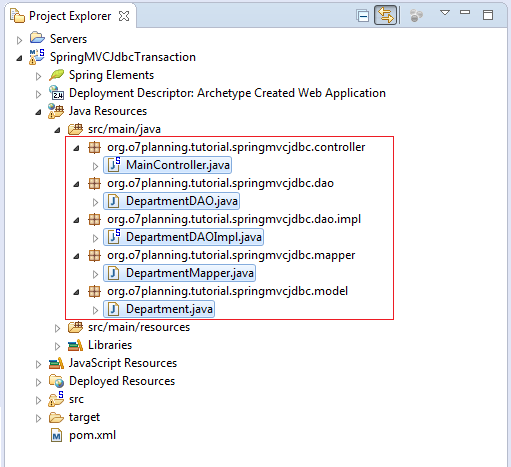
* **data-source-cfg.xml**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="<http://www.springframework.org/schema/beans>"      xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"      xsi:schemaLocation="<http://www.springframework.org/schema/beans>  <http://www.springframework.org/schema/beans/spring-beans.xsd>">     <bean id="myDataSource"          class="org.springframework.jdbc.datasource.DriverManagerDataSource">      <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver" />    <property name="url" value="jdbc:oracle:thin:@localhost:1521:db11g" />    <property name="username" value="simplehr" />    <property name="password" value="12345" />   </bean>    </beans> |

* **transaction-cfg.xml**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | <?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:tx="<http://www.springframework.org/schema/tx>"      xmlns:aop="<http://www.springframework.org/schema/aop>"      xsi:schemaLocation="<http://www.springframework.org/schema/beans>  <http://www.springframework.org/schema/beans/spring-beans-4.1.xsd>  <http://www.springframework.org/schema/aop>  [http://www.springframework.org/schema/aop/spring-aop-4.1.xsd](http://www.springframework.org/schema/aop/spring-aop-4.1.xsd%C2%A0%C2%A0%C2%A0%C2%A0%C2%A0%C2%A0%C2%A0)  <http://www.springframework.org/schema/tx>  <http://www.springframework.org/schema/tx/spring-tx-4.1.xsd>">     <import resource="data-source-cfg.xml"/>    <tx:annotation-driven proxy-target-class="true" transaction-manager="txManager"/>    <bean id="txManager"    class="org.springframework.jdbc.datasource.DataSourceTransactionManager">          <property name="dataSource" ref="myDataSource" />      </bean>    </beans> |

6- Java Classes [▴](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial" \l "top)



* **Department.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46 | package org.o7planning.tutorial.springmvcjdbc.model;    public class Department {       private Integer deptId;     private String deptNo;     private String deptName;     private String location;       public Department() {       }       public Department(Integer deptId, String deptNo, String deptName,             String location) {         this.deptId = deptId;         this.deptNo = deptNo;         this.deptName = deptName;         this.location = location;     }       public Integer getDeptId() {         return deptId;     }       public void setDeptId(Integer deptId) {         this.deptId = deptId;     }       public String getDeptNo() {         return deptNo;     }       public void setDeptNo(String deptNo) {         this.deptNo = deptNo;     }       public String getDeptName() {         return deptName;     }       public void setDeptName(String deptName) {         this.deptName = deptName;     }       public String getLocation() {         return location;     }       public void setLocation(String location) {         this.location = location;     }  } |

* **DepartmentMapper.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | package org.o7planning.tutorial.springmvcjdbc.mapper;    import java.sql.ResultSet;  import java.sql.SQLException;    import org.o7planning.tutorial.springmvcjdbc.model.Department;  import org.springframework.jdbc.core.RowMapper;    public class DepartmentMapper implements RowMapper<Department> {        @Override      public Department mapRow(ResultSet rs, int rowNum) throws SQLException {          Integer deptId = rs.getInt("dept\_id");          String deptNo = rs.getString("dept\_no");          String deptName = rs.getString("dept\_name");          String location = rs.getString("location");            return new Department(deptId, deptNo, deptName, location);      }    } |

* **DepartmentDAO.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | package org.o7planning.tutorial.springmvcjdbc.dao;    import java.util.List;    import org.o7planning.tutorial.springmvcjdbc.model.Department;    public interface DepartmentDAO {        public Department findDepartment(String deptNo);        public void insertDepartment(String deptName, String location);        public List<Department> listDepartment();  } |

* **DepartmentDAOImpl.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61 | package org.o7planning.tutorial.springmvcjdbc.dao.impl;    import java.util.List;    import javax.sql.DataSource;    import org.o7planning.tutorial.springmvcjdbc.dao.DepartmentDAO;  import org.o7planning.tutorial.springmvcjdbc.mapper.DepartmentMapper;  import org.o7planning.tutorial.springmvcjdbc.model.Department;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.jdbc.core.support.JdbcDaoSupport;  import org.springframework.stereotype.Service;  import org.springframework.transaction.annotation.Transactional;    @Service  @Transactional  public class DepartmentDAOImpl extends JdbcDaoSupport implements DepartmentDAO {        @Autowired      public DepartmentDAOImpl(DataSource dataSource) {          this.setDataSource(dataSource);      }        private int getMaxDeptId() {          String sql = "Select max(d.dept\_id) from Department d";            Integer value = this.getJdbcTemplate().queryForObject(sql,                  Integer.class);          if (value == null) {              return 0;          }          return value;      }        @Override      public Department findDepartment(String deptNo) {          String sql = "Select d.dept\_id,d.dept\_no,d.dept\_name,d.location "                  + " from Department d where d.dept\_no = ?";            Object[] params = new Object[] { deptNo };          DepartmentMapper mapper = new DepartmentMapper();            Department dept = this.getJdbcTemplate().queryForObject(sql, params,                  mapper);          return dept;      }        @Override      public List<Department> listDepartment() {          String sql = "Select d.dept\_id,d.dept\_no,d.dept\_name,d.location "                  + " from Department d";            Object[] params = new Object[] {};          DepartmentMapper mapper = new DepartmentMapper();            List<Department> list = this.getJdbcTemplate().query(sql, params,                  mapper);          return list;      }        @Override      public void insertDepartment(String deptName, String location) {          String sql = "Insert into Department (dept\_id,dept\_no,dept\_name,location) "                  + " values (?,?,?,?) ";          int deptId = this.getMaxDeptId() + 1;          String deptNo = "D" + deptId;          Object[] params = new Object[] { deptId, deptNo, deptName, location };          this.getJdbcTemplate().update(sql, params);      }    } |

**@Transactional** can annotate on the method, or on the class, it will work on all methods of class.

* **MainController.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | package org.o7planning.tutorial.springmvcjdbc.controller;    import java.util.List;    import org.o7planning.tutorial.springmvcjdbc.dao.DepartmentDAO;  import org.o7planning.tutorial.springmvcjdbc.model.Department;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Controller;  import org.springframework.ui.Model;  import org.springframework.web.bind.annotation.RequestMapping;  import org.springframework.web.bind.annotation.RequestMethod;    @Controller  public class MainController {        @Autowired      private DepartmentDAO departmentDAO;        @RequestMapping(value = { "/" }, method = RequestMethod.GET)      public String welcomePage(Model model) {          departmentDAO.insertDepartment("HR", "Chicago");          departmentDAO.insertDepartment("INV", "Hanoi");          List<Department> list = departmentDAO.listDepartment();          model.addAttribute("departments", list);          return "index";      }  } |

7- Views [▴](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial" \l "top)

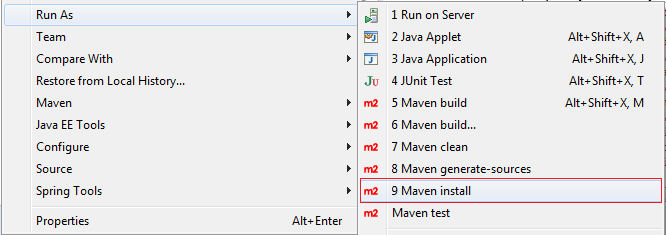
* **index.jsp**

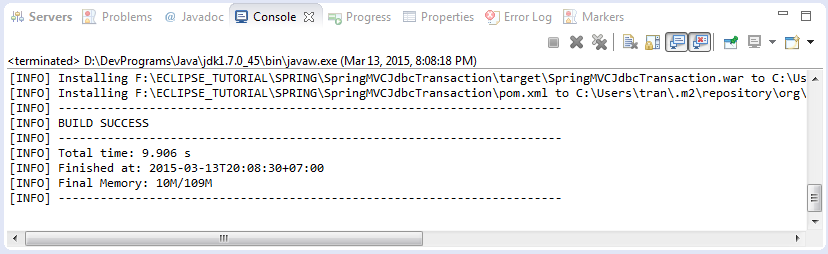
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | <%@page contentType="text/html" pageEncoding="UTF-8"%>  <%@ taglib uri="<http://java.sun.com/jsp/jstl/core>" prefix="c" %>    <html>    <head>        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">        <title>Department List</title>    </head>    <body>        <div align="center">            <h1>Department List</h1>            <table border="1">                <tr>                  <th>No</th>                  <th>Dept No</th>                  <th>Dept Name</th>                  <th>Location</th>                </tr>                <c:forEach var="dept" items="${departments}" varStatus="status">                <tr>                    <td>${status.index + 1}</td>                    <td>${dept.deptNo}</td>                    <td>${dept.deptName}</td>                    <td>${dept.location}</td>                </tr>                </c:forEach>            </table>        </div>    </body>  </html> |

8- Run Application [▴](http://o7planning.org/web/fe/default/en/document/142158/spring-mvc-and-spring-jdbc-transaction-tutorial" \l "top)

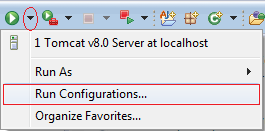
In the first, before running the application you need to build the entire project.

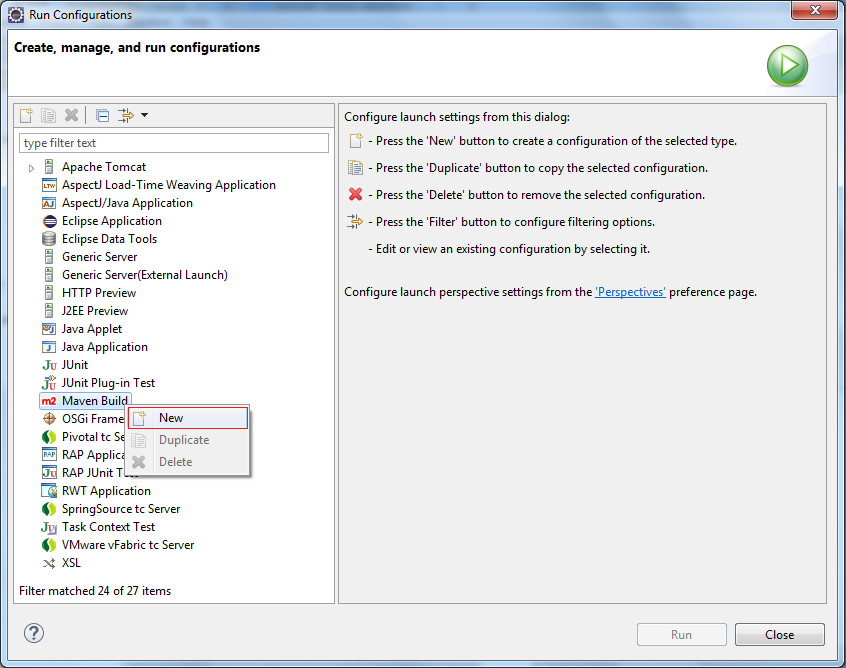
Right-click the project and select:





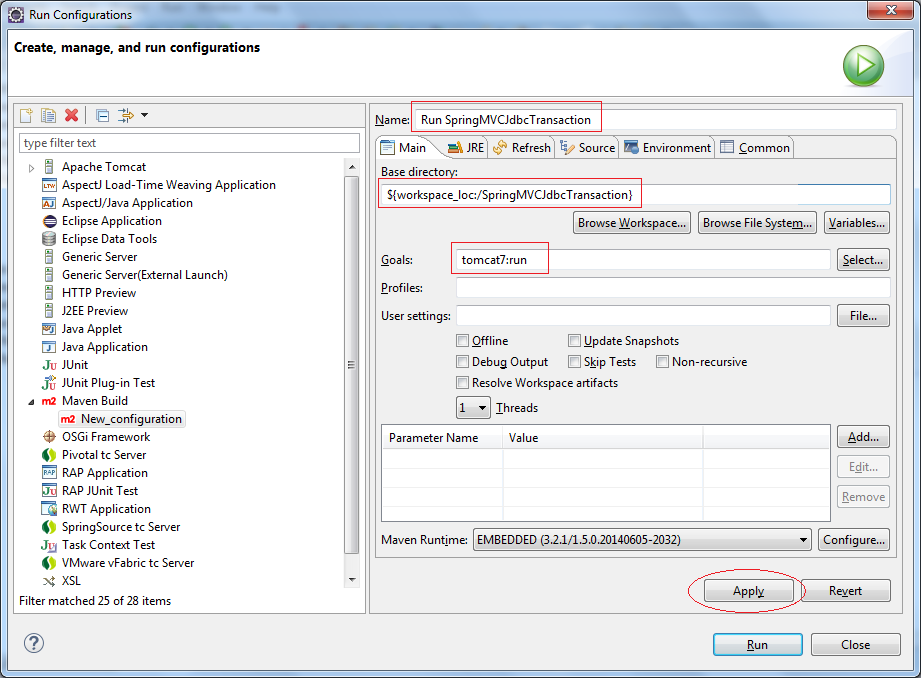
**Run Configurations:**



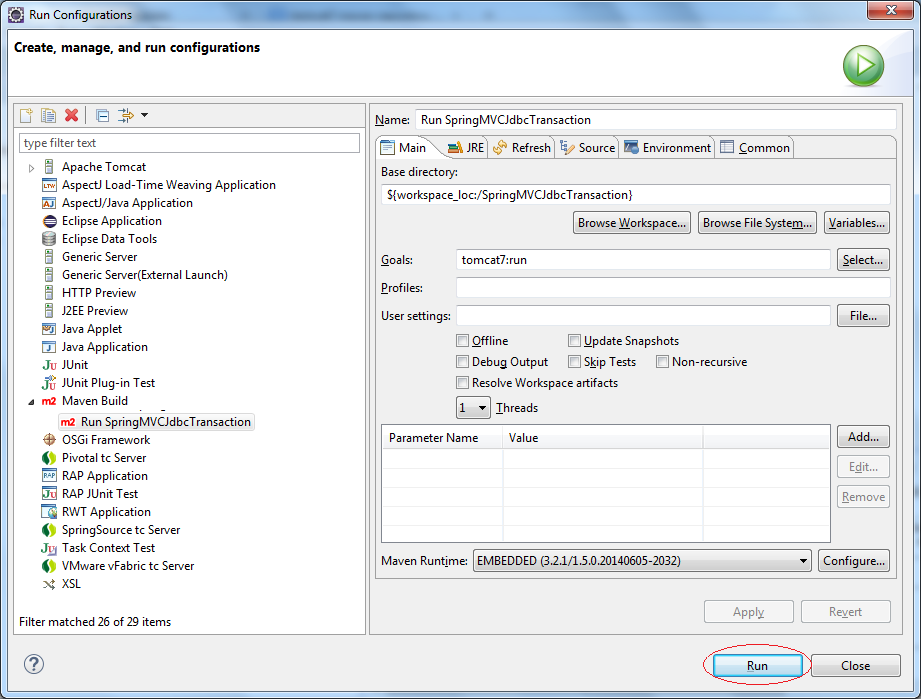


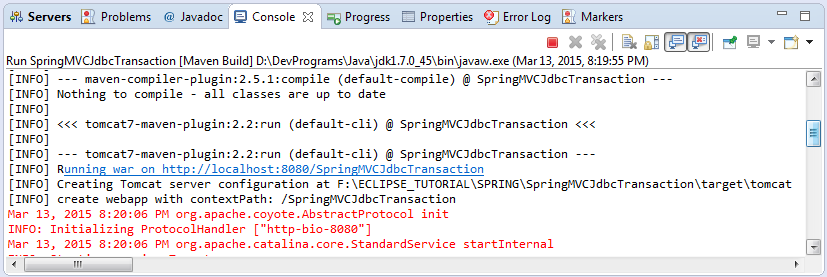
Enter:

* **Name:** Run SpringMVCJdbcTransaction
* **Base directory:** ${workspace\_loc:/SpringMVCJdbcTransaction}
* **Goals:** tomcat7:run



Click Run:





* <http://localhost:8080/SpringMVCJdbcTransaction/>

