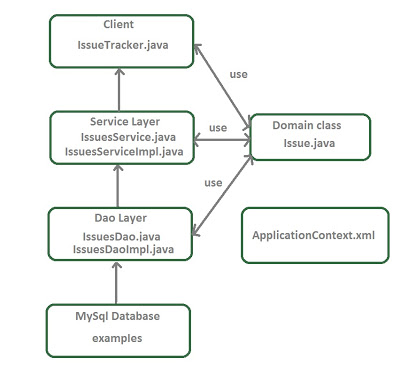
**Spring JDBC Integration Example –using DaoSupport Classes**



***Jars Needed:***

org.springframework.jdbc-3.0.6.RELEASE.jar  
org.springframework.transaction-3.0.6.RELEASE.jar  
org.springframework.core-3.0.6.RELEASE.jar  
org.springframework.context-3.0.6.RELEASE.jar  
org.springframework.beans-3.0.6.RELEASE.jar  
commons-logging-1.1.1.jar  
org.springframework.asm-3.0.6.RELEASE.jar  
org.springframework.expression-3.0.6.RELEASE.jar  
commons-dbcp-1.3.jar  
commons-pool-1.5.7.jar  
mysql-connector-java-5.1.12-bin.jar

**Table Structure**

CREATE TABLE examples.issue(  
  tid INT(11) UNSIGNED NOT NULL AUTO\_INCREMENT,  
  assigned\_to VARCHAR(255) NOT NULL,  
  assigned\_by VARCHAR(255) NOT NULL,  
  status VARCHAR(255) NOT NULL,  
  PRIMARY KEY (tid)  
)  
ENGINE = INNODB  
AUTO\_INCREMENT = 129  
AVG\_ROW\_LENGTH = 268  
CHARACTER SET latin1  
COLLATE latin1\_swedish\_ci;

***Domain***

***Issue.java***

package examples.springjdbc.domain;  
  
public class Issue{  
private int tid;  
private String assigned\_to;  
private String assigned\_by;  
private String status;  
  
public void setTid(int tid){  
    this.tid = tid;  
}  
  
  
public int getTid(){  
    return tid;  
}  
  
public void setAssigned\_to(String assigned\_to){  
    this.assigned\_to = assigned\_to;  
}  
  
public String getAssigned\_to(){  
    return assigned\_to;  
}  
  
public void setAssigned\_by(String assigned\_by){  
    this.assigned\_by = assigned\_by;  
}  
  
public String getAssigned\_by(){  
    return assigned\_by;  
}  
  
public void setStatus(String status){  
    this.status = status;  
}  
  
public String getStatus(){  
    return status;  
}  
}

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Dao Layer**

***IssuesDao.java (an interface)***

package examples.springjdbc.dao;  
  
import examples.springjdbc.domain.Issue;  
import java.util.List;  
  
public interface IssuesDao{  
      
  public void addIssue(Issue issue);  
  public List<Issue> getIssuesByAssignee(String asignee);  
  public Issue getIssuesById(int tid);  
  public int updateIssue(Issue issue);  
  public void deleteIssue(int tid);  
}

***IssuesDaoImpl.java***

package examples.springjdbc.dao;  
  
import examples.springjdbc.domain.Issue;  
  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.List;  
import org.springframework.jdbc.core.RowMapper;  
import org.springframework.jdbc.core.support.JdbcDaoSupport;  
  
public class IssuesDaoImpl extends **JdbcDaoSupport** implements IssuesDao{  
      
  public void addIssue(Issue issue){  
      String sql = "insert into issue(assigned\_to,assigned\_by,status) values(?,?,?)";  
     **getJdbcTemplate().**update(sql,issue.getAssigned\_to(),issue.getAssigned\_by(),issue.getStatus());  
  }  
  public List<Issue> getIssuesByAssignee(String assignee){  
        
     String sql="select \* from issue where assigned\_by=?";  
      
     RowMapper<Issue> myRowMapper = new RowMapper<Issue>(){  
               
       public Issue mapRow(ResultSet rs, int rowNum)throws SQLException{  
             
           Issue issue = new Issue();  
           issue.setTid(rs.getInt("tid"));  
           issue.setAssigned\_to(rs.getString("assigned\_to"));  
           issue.setAssigned\_by(rs.getString("assigned\_by"));  
           issue.setStatus(rs.getString("status"));  
            
           return issue;  
        }  
     };  
      
     List<Issue> issueList = getJdbcTemplate().query(sql,myRowMapper,assignee);  
     return issueList;  
        
  }  
    
  public Issue getIssuesById(int tid){  
      String sql="select \* from issue where tid=?";  
        
      RowMapper<Issue> myRowMapper = new RowMapper<Issue>(){  
               
           public Issue mapRow(ResultSet rs, int rowNum)throws SQLException{  
                 
               Issue issue = new Issue();  
               issue.setTid(rs.getInt("tid"));  
               issue.setAssigned\_to(rs.getString("assigned\_to"));  
               issue.setAssigned\_by(rs.getString("assigned\_by"));  
               issue.setStatus(rs.getString("status"));  
                
               return issue;  
            }  
         };  
           
      Issue issue = getJdbcTemplate().queryForObject(sql,myRowMapper,tid);  
        
      return issue;  
  }  
    
  public int updateIssue(Issue issue){  
      
      String sql = "update issue set assigned\_to=?,assigned\_by=?,status=? where tid=?";  
      int rows=getJdbcTemplate().update(sql,issue.getAssigned\_to(),issue.getAssigned\_by(),issue.getStatus(),issue.getTid());  
      return rows;  
  }  
  public void deleteIssue(int tid){  
      String sql = "delete from issue  where tid=?";  
      getJdbcTemplate().update(sql,tid);}}

## 

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Service Layer**  
***IssuesService.java (an interface)***

package examples.springjdbc.service;  
  
import examples.springjdbc.domain.Issue;  
import java.util.List;  
  
public interface IssuesService{  
      
  public void createIssue(Issue issue);  
  public List<Issue> listByAssignee(String asignee);  
  public Issue getIssuesById(int tid);  
  public int updateIssue(Issue issue);  
  public void deleteIssue(int tid);  
}

***IssuesServiceImpl.java***

package examples.springjdbc.service;  
  
import examples.springjdbc.domain.Issue;  
import examples.springjdbc.dao.IssuesDao;  
import java.util.List;  
  
public class IssuesServiceImpl implements IssuesService{  
  
    private IssuesDao issueDao;  
  
    public void setIssueDao(IssuesDao issueDao){  
        this.issueDao=issueDao;  
    }  
  public void createIssue(Issue issue){  
       
      issueDao.addIssue(issue);  
  }  
  public List<Issue> listByAssignee(String asignee){  
     List<Issue> list= issueDao.getIssuesByAssignee(asignee);  
     return list;  
  }  
    
  public Issue getIssuesById(int tid){  
       
      Issue issue= issueDao.getIssuesById(tid);  
      return issue;  
       
  }  
    
  public int updateIssue(Issue issue){  
      int row=issueDao.updateIssue(issue);  
      return row;  
  }  
  public void deleteIssue(int tid){  
      issueDao.deleteIssue(tid);  
  }  
}

**Client**

***IssueTracker.java***

package examples.springjdbc.client;  
  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStreamReader;  
import java.util.List;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import examples.springjdbc.domain.Issue;  
import examples.springjdbc.service.IssuesService;  
  
public class IssueTracker{  
  
public static void main(String args[]){  
      
  
ApplicationContext appContext = new ClassPathXmlApplicationContext("resources/ApplicationContext.xml");  
  
IssuesService iService = (IssuesService) appContext.getBean("iService");  
  
**/\*Creating new Issues\*/**  
Issue issue = new Issue();  
  
issue.setAssigned\_to("Mr. X");  
issue.setAssigned\_by("Mr. A");  
issue.setStatus("Open");  
  
iService.createIssue(issue);  
  
System.out.println("Issue Created...");  
  
issue = new Issue();  
  
issue.setAssigned\_to("Mr. Y");  
issue.setAssigned\_by("Mr. B");  
issue.setStatus("Open");  
  
iService.createIssue(issue);  
System.out.println("Issue Created...");  
  
issue = new Issue();  
  
issue.setAssigned\_to("Mr. Z");  
issue.setAssigned\_by("Mr. B");  
issue.setStatus("Open");  
  
iService.createIssue(issue);  
System.out.println("Issue Created...");  
  
**/\*Listing Issues\*/**  
List<Issue> issuesList=iService.listByAssignee("Mr. B");  
  
System.out.println("Issues assigned by Mr. B:");  
System.out.println("==========================");  
for(Issue i:issuesList){  
    System.out.print("Id:"+i.getTid());  
    System.out.print("    Assigned By :"+i.getAssigned\_by());  
    System.out.print("    Assigned To:"+i.getAssigned\_to());  
    System.out.println("    Status:"+i.getStatus());  
}  
System.out.println("==========================");  
  
**/\*Updating the issue\*/**  
  
System.out.println("Enter Id of the issue to be updated: ");  
  
int op=0;  
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
try{  
op= Integer.parseInt(br.readLine());  
}catch(IOException e){  
    System.out.println(e);  
      
}  
  
issue=iService.getIssuesById(op);  
  
issue.setStatus("closed");  
  
iService.updateIssue(issue);  
System.out.println("Issue status Updated...");  
  
**/\*deleting the issue\*/**  
System.out.println("Enter Id of the issue to be deleted: ");  
  
try{  
op= Integer.parseInt(br.readLine());  
}catch(IOException e){  
    System.out.println(e);  
      
}  
  
iService.deleteIssue(op);  
System.out.println("Issue Deleted...");  
  
}  
}

**ApplicationContext.xml**

**<?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-3.0.xsd">  
  
<bean id="myDataSource" 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/examples" />  
 <property name="username" value="root" />  
  
 <!-- change 'root' with your password -->  
 <property name="password" value="root" />  
  
</bean>  
  
<!--  No Need To Declare Template  
  
 <bean id="myJdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">  
 <property name="dataSource" ref="myDataSource" />  
</bean>  
-->  
<bean id="myIssueDao" class="examples.springjdbc.dao.IssuesDaoImpl">  
 <!-- <property name="jdbcTemplate" ref="myJdbcTemplate"/>-->  
 <property name="dataSource" ref="myDataSource"/>  
</bean>  
  
<bean id="iService" class="examples.springjdbc.service.IssuesServiceImpl">  
 <property name="issueDao" ref="myIssueDao"/>  
</bean>  
  
</beans>**