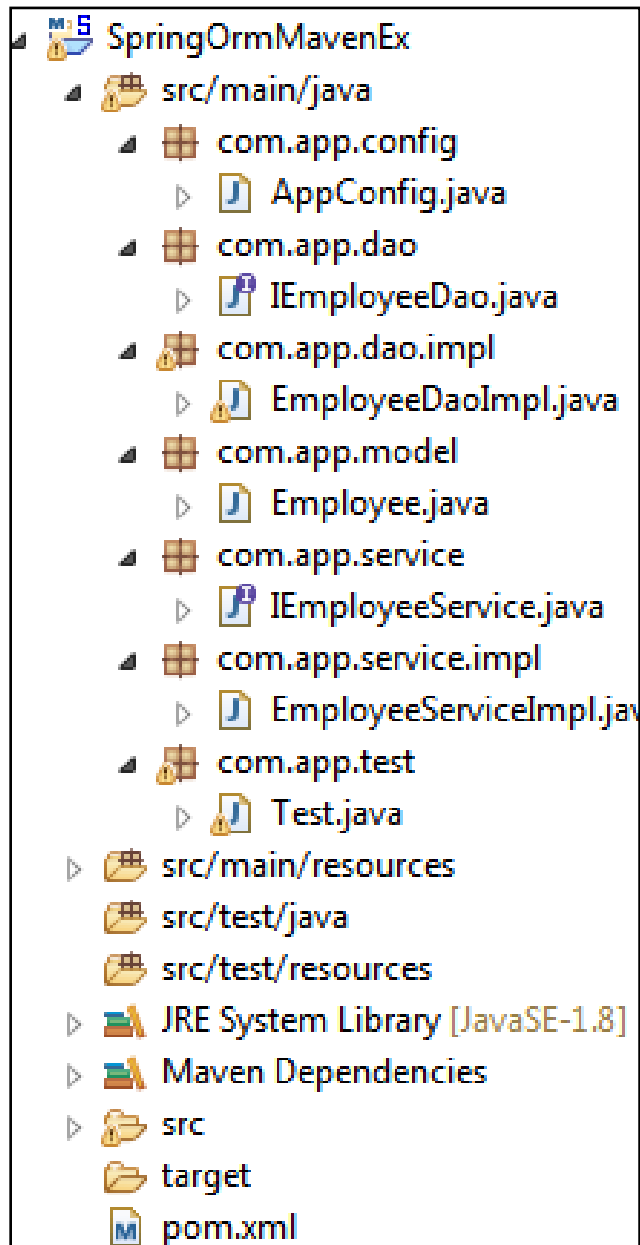


Spring ORM (Hibernate) Using Java and Annotation Config with Maven

[- by RAGHU]

Folder System:



pom.xml

```
<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/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.sathyatech</groupId>
  <artifactId>SpringOrmMavenEx</artifactId>
  <version>1.0</version>

  <dependencies>
    <!-- Spring -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-orm</artifactId>
      <version>5.0.6.RELEASE</version>
    </dependency>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>5.0.6.RELEASE</version>
    </dependency>
    <!-- MySQL -->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>5.1.46</version>
    </dependency>
    <!-- Hibernate -->
    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>5.2.17.Final</version>
    </dependency>
    <!-- Connection Pooling -->
    <dependency>
```

```
        <groupId>org.apache.commons</groupId>
        <artifactId>commons-dbcp2</artifactId>
        <version>2.1.1</version>
    </dependency>
</dependencies>

<!-- JDK 8 compiler and Build plugins -->
<build>
    <plugins>
        <plugin>

            <groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-compiler-plugin</artifactId>
            <version>3.7.0</version>
            <configuration>
                <source>1.8</source>
                <target>1.8</target>
            </configuration>
        </plugin>
    </plugins>
</build>

</project>
```

AppConfig:

```
package com.app.config;

import java.util.Properties;

import org.apache.commons.dbcp2.BasicDataSource;
import
org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
```

```
import
org.springframework.context.annotation.PropertySource;
import org.springframework.core.env.Environment;
import
org.springframework.orm.hibernate5.HibernateTransactionManag
er;
import
org.springframework.orm.hibernate5.LocalSessionFactoryBean;
import
org.springframework.transaction.annotation.EnableTransaction
Management;

import com.app.model.Employee;

@Configuration
@PropertySource("app.properties")
@ComponentScan(basePackages= {"com.app"})
@EnableTransactionManagement
public class AppConfig {

    @Autowired
    private Environment env;

    //1. DataSource Obj
    @Bean
    public BasicDataSource dsObj() {
        BasicDataSource ds=new BasicDataSource();
        ds.setDriverClassName(env.getProperty("dc"));
        ds.setUrl(env.getProperty("url"));
        ds.setUsername(env.getProperty("un"));
        ds.setPassword(env.getProperty("pwd"));

        ds.setInitialSize(1);
        ds.setMaxTotal(10);
        ds.setMaxIdle(5);
        ds.setMinIdle(2);

        return ds;
    }
}
```

```
}  
//2. SessionFactory Bean  
@Bean  
public LocalSessionFactoryBean sfObj() {  
    LocalSessionFactoryBean sf=new  
LocalSessionFactoryBean();  
    sf.setDataSource(dsObj());  
    sf.setHibernateProperties(props());  
    sf.setAnnotatedClasses(Employee.class);  
    return sf;  
}  
  
private Properties props() {  
    Properties p=new Properties();  
    p.put("hibernate.dialect",  
env.getProperty("dialect"));  
    p.put("hibernate.show_sql",  
env.getProperty("showsql"));  
    p.put("hibernate.format_sql",  
env.getProperty("formatsql"));  
  
    p.put("hibernate.hbm2ddl.auto",env.getProperty("ddlauto"  
));  
    return p;  
}  
  
//3. Tx Manager  
@Bean  
public HibernateTransactionManager htx() {  
    HibernateTransactionManager htm=new  
HibernateTransactionManager();  
    htm.setSessionFactory(sfObj().getObject());  
    return htm;  
}  
}
```

Model Class

```
package com.app.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="emptab")
public class Employee {
    @Id
    @Column(name="eid")
    private int empId;
    @Column(name="ename")
    private String empName;
    @Column(name="esal")
    private double empSal;

    public Employee() {
        super();
    }
    public Employee(int empId) {
        super();
        this.empId = empId;
    }
    public int getEmpId() {
        return empId;
    }
    public void setEmpId(int empId) {
        this.empId = empId;
    }
    public String getEmpName() {
        return empName;
    }
}
```

```
}  
public void setEmpName(String empName) {  
    this.empName = empName;  
}  
public double getEmpSal() {  
    return empSal;  
}  
public void setEmpSal(double empSal) {  
    this.empSal = empSal;  
}  
@Override  
public String toString() {  
    return "Employee [empId=" + empId + ", empName=" +  
empName + ", empSal=" + empSal + "];"  
}  
}
```

IDao :

```
package com.app.dao;  
  
import java.util.List;  
  
import com.app.model.Employee;  
  
public interface IEmployeeDao {  
  
    public void saveEmployee(Employee emp);  
    public void updateEmployee(Employee emp);  
    public void deleteEmployee(int empId);  
  
    public Employee getEmployeeById(int empId);  
    public List<Employee> getAllEmployees();  
}
```

DaoImpl:

```
package com.app.dao.impl;

import java.util.List;

import org.hibernate.SessionFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;

import com.app.dao.IEmployeeDao;
import com.app.model.Employee;

@Repository
public class EmployeeDaoImpl implements IEmployeeDao {

    @Autowired
    private SessionFactory sf;

    @Override
    public void saveEmployee(Employee emp) {
        sf.getCurrentSession().save(emp);
    }

    @Override
    public void updateEmployee(Employee emp) {
        sf.getCurrentSession().update(emp);
    }

    @Override
    public void deleteEmployee(int empId) {
        sf.getCurrentSession().delete(new Employee(empId));
    }

    @Override
```



```
    public Employee getEmployeeById(int empId) {  
        return sf.getCurrentSession().get(Employee.class,  
empId);  
    }  
    @Override  
    public List<Employee> getAllEmployees() {  
        return sf.getCurrentSession().createQuery("from  
com.app.model.Employee").list();  
    }  
}
```

IService :

```
package com.app.service;  
  
import java.util.List;  
  
import com.app.model.Employee;  
  
public interface IEmployeeService {  
  
    public void saveEmployee(Employee emp);  
    public void updateEmployee(Employee emp);  
    public void deleteEmployee(int empId);  
  
    public Employee getEmployeeById(int empId);  
    public List<Employee> getAllEmployees();  
  
}
```

ServiceImpl :

```
package com.app.service.impl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import com.app.dao.IEmployeeDao;
import com.app.model.Employee;
import com.app.service.IEmployeeService;

@Service
public class EmployeeServiceImpl implements IEmployeeService
{
    @Autowired
    private IEmployeeDao dao;

    @Transactional
    public void saveEmployee(Employee emp) {
        dao.saveEmployee(emp);
    }
    @Transactional
    public void updateEmployee(Employee emp) {
        dao.updateEmployee(emp);
    }
    @Transactional
    public void deleteEmployee(int empId) {
        dao.deleteEmployee(empId);
    }

    @Transactional(readOnly=true)
    public Employee getEmployeeById(int empId) {
        return dao.getEmployeeById(empId);
    }
}
```

```
@Transactional(readOnly=true)
public List<Employee> getAllEmployees() {
    return dao.getAllEmployees();
}
}
```

Test class :

```
package com.app.test;

import java.util.List;

import org.springframework.context.annotation.AnnotationConfigApplica
tionContext;

import com.app.config.AppConfig;
import com.app.model.Employee;
import com.app.service.IEmployeeService;

public class Test {

    public static void main(String[] args) {
        AnnotationConfigApplicationContext ac=new
AnnotationConfigApplicationContext(AppConfig.class);
        IEmployeeService service=(IEmployeeService)
ac.getBean("employeeServiceImpl");
        /* Ex#1
        Employee emp=new Employee();
        emp.setEmpId(102);
        emp.setEmpName("CC");
        emp.setEmpSal(33.36);

        service.saveEmployee(emp);
        System.out.println("done");*/
    }
}
```

```
/*Ex#2
Employee e=service.getEmployeeById(100);
System.out.println(e);*/
//Ex#3
List<Employee> emps=service.getAllEmployees();
emps.forEach(System.out::println);

}
}
```

FB Group: <https://www.facebook.com/groups/thejavatemple/>

Email : javabyraghu@gmail.com