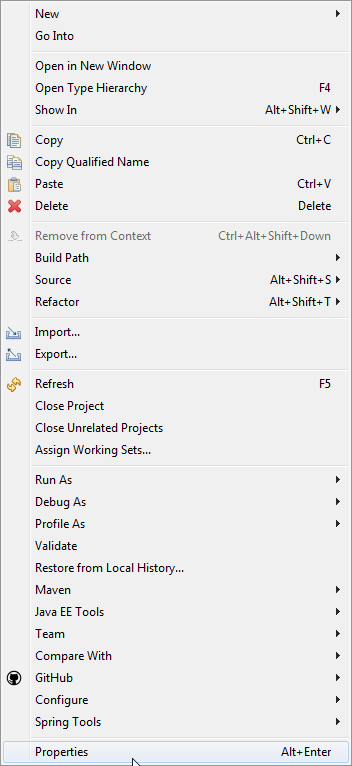
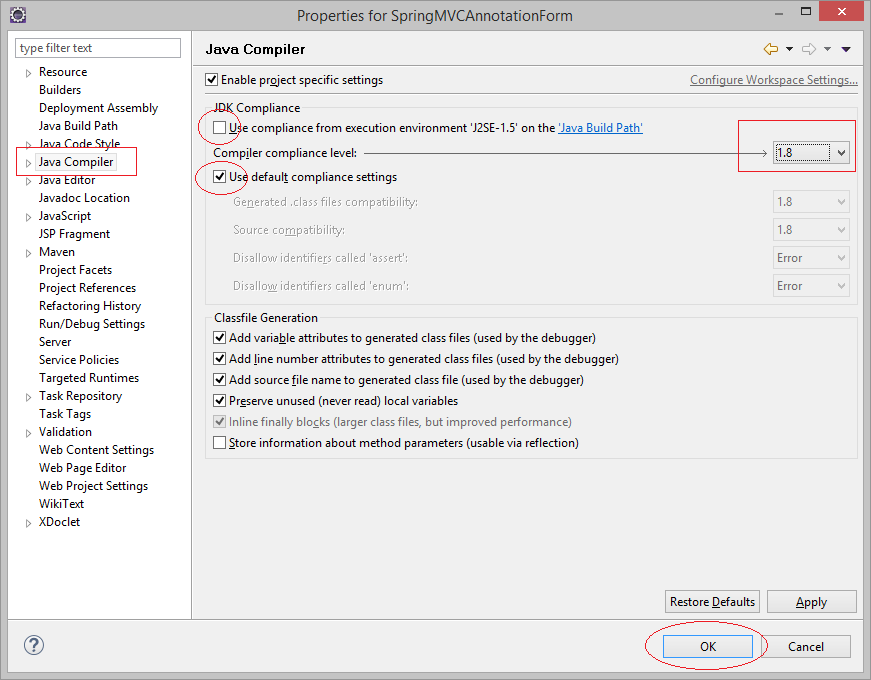
**Spring + Hibernate (CRUD )**

## Loi khong up len dc web 3.0

<http://stackoverflow.com/questions/18122336/cannot-change-version-of-project-facet-dynamic-web-module-to-3-0/18632054#18632054>

## **Chinh Java compiler 1.5 -> 1.8 ( Override problem)**





## Tao database mau

Create a database name : applicants

This is MySQL script

-- Create table

**create** **table** APPLICANTS

**(**

ID **VARCHAR(**50**)** **not** **null,**

GENDER **VARCHAR(**10**)** **not** **null,**

NAME **VARCHAR(**50**)** **not** **null,**

POSITION **VARCHAR(**50**)** **not** **null,**

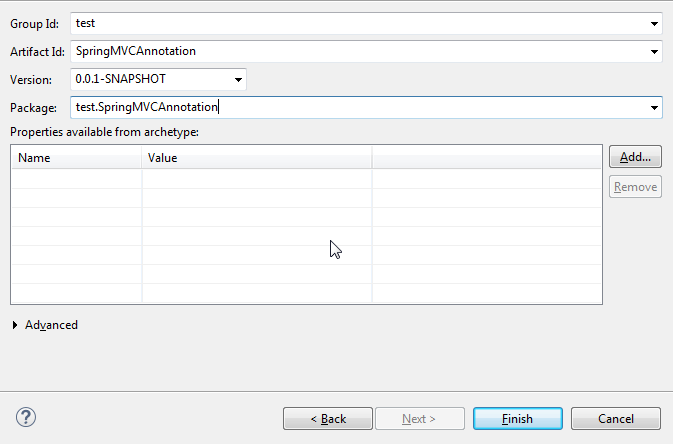
SKILLS **VARCHAR(**255**)** **not** **null,**

EMAIL **VARCHAR(**50**)** **not** **null,**

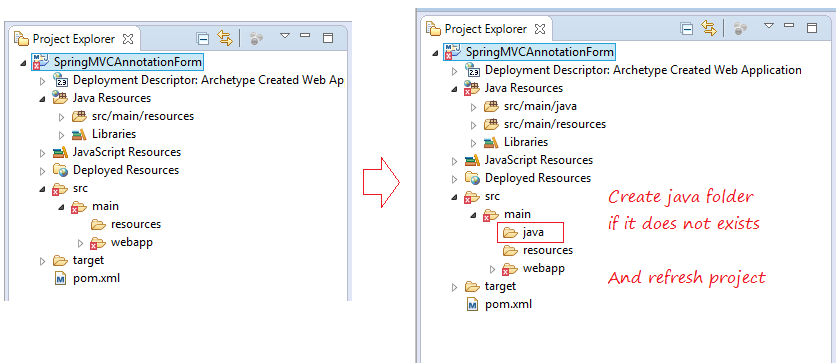
**primary** **key** **(**ID**)**

**)** **;**

Create an maven web-app project



Create java folder and resources folder if it is disappear



## Config Web.XML

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns=*"http://java.sun.com/xml/ns/javaee"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"*

id=*"WebApp\_ID"* version=*"3.0"*>

<display-name>SpringMVCForm</display-name>

</web-app>

## Config pom.xml Maven

<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>test</groupId>

<artifactId>SpringMVCAnnotationForm</artifactId>

<packaging>war</packaging>

<version>1.0.0</version>

<name>SpringMVCAnnotationForm Maven Webapp</name>

<url>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>4.2.5.RELEASE</version>

</dependency>

<!-- http://mvnrepository.com/artifact/org.springframework/spring-web -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>4.2.5.RELEASE</version>

</dependency>

<!-- http://mvnrepository.com/artifact/org.springframework/spring-webmvc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>4.2.5.RELEASE</version>

</dependency>

<!-- http://mvnrepository.com/artifact/org.springframework/spring-orm -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>4.2.5.RELEASE</version>

</dependency>

<!-- Hibernate -->

<!-- http://mvnrepository.com/artifact/org.hibernate/hibernate-core -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.1.0.Final</version>

</dependency>

<!-- http://mvnrepository.com/artifact/org.hibernate/hibernate-entitymanager -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-entitymanager</artifactId>

<version>5.1.0.Final</version>

</dependency>

<!-- http://mvnrepository.com/artifact/org.hibernate/hibernate-c3p0 -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-c3p0</artifactId>

<version>5.1.0.Final</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>

<!-- Email validator,... -->

<!-- http://mvnrepository.com/artifact/commons-validator/commons-validator -->

<dependency>

<groupId>commons-validator</groupId>

<artifactId>commons-validator</artifactId>

<version>1.5.0</version>

</dependency>

</dependencies>

<build>

<finalName>SpringMVCAnnotationForm</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 - /SpringMVCAnnotationForm : 8080) -->

<!--

<configuration>

<path>/</path>

<port>8899</port>

</configuration>

-->

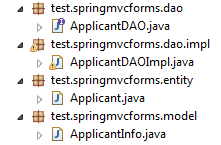
</plugin>

</plugins>

</build>

</project>

## Create Entity , Dao and Implemetation



### Applicant.java

package test**.**springmvcforms**.**entity**;**

**import** java**.**io**.**Serializable**;**

**import** javax**.**persistence**.**Column**;**

**import** javax**.**persistence**.**Entity**;**

**import** javax**.**persistence**.**Id**;**

**import** javax**.**persistence**.**Table**;**

@Entity

@Table**(**name **=** "Applicants"**)**

public class Applicant **implements** Serializable **{**

private static final long serialVersionUID **=** **-**7893237204476214050L**;**

private String id**;**

private String name**;**

private String email**;**

private String position**;**

private String gender**;**

private String skills**;**

@Id

@Column**(**name **=** "ID"**,** length **=** 50**,** nullable **=** **false)**

public String getId**()** **{**

**return** id**;**

**}**

public void setId**(**String id**)** **{**

**this.**id **=** id**;**

**}**

@Column**(**name **=** "Name"**,** length **=** 50**,** nullable **=** **false)**

public String getName**()** **{**

**return** name**;**

**}**

public void setName**(**String name**)** **{**

**this.**name **=** name**;**

**}**

@Column**(**name **=** "Position"**,** length **=** 50**,** nullable **=** **false)**

public String getPosition**()** **{**

**return** position**;**

**}**

public void setPosition**(**String position**)** **{**

**this.**position **=** position**;**

**}**

@Column**(**name **=** "Gender"**,** length **=** 10**,** nullable **=** **false)**

public String getGender**()** **{**

**return** gender**;**

**}**

public void setGender**(**String gender**)** **{**

**this.**gender **=** gender**;**

**}**

@Column**(**name **=** "Email"**,** length **=** 50**,** nullable **=** **false)**

public String getEmail**()** **{**

**return** email**;**

**}**

public void setEmail**(**String email**)** **{**

**this.**email **=** email**;**

**}**

@Column**(**name **=** "Skills"**,** length **=** 255**,** nullable **=** **false)**

public String getSkills**()** **{**

**return** skills**;**

**}**

public void setSkills**(**String skills**)** **{**

**this.**skills **=** skills**;**

**}**

**}**

@Id = primary Key

@Table = knowing this class is a table named Applicants

@Column = setting column and the data types

@Entity = recognize the class as an entity to be used by Hibernate and transactions..

### ApplicantInfo.java

package test**.**springmvcforms**.**model**;**

public class ApplicantInfo **{**

private String id**;**

private String name**;**

private String email**;**

private String position**;**

private String gender**;**

private String skillsString**;**

private String**[]** skills**;**

public ApplicantInfo**()** **{**

**}**

// Không được thay đổi Constructor này,

// Nó được sử dụng trong Hibernate Query.

public ApplicantInfo**(**String id**,** String name**,** String email**,** String gender**,** String position**,** String skillsString**)** **{**

**this.**id **=** id**;**

**this.**name **=** name**;**

**this.**email **=** email**;**

**this.**position **=** position**;**

**this.**gender **=** gender**;**

**this.**skillsString **=** skillsString**;**

**this.**skills **=** toArray**(**skillsString**);**

**}**

private String**[]** toArray**(**String skillStrings**)** **{**

**if** **(**skillStrings **==** **null)** **{**

**return** **new** String**[**0**];**

**}**

String**[]** ret **=** skillStrings**.**split**(**","**);**

**return** ret**;**

**}**

public String getId**()** **{**

**return** id**;**

**}**

public void setId**(**String id**)** **{**

**this.**id **=** id**;**

**}**

public String getName**()** **{**

**return** name**;**

**}**

public void setName**(**String name**)** **{**

**this.**name **=** name**;**

**}**

public String getEmail**()** **{**

**return** email**;**

**}**

public void setEmail**(**String email**)** **{**

**this.**email **=** email**;**

**}**

public String getPosition**()** **{**

**return** position**;**

**}**

public void setPosition**(**String position**)** **{**

**this.**position **=** position**;**

**}**

public String getGender**()** **{**

**return** gender**;**

**}**

public void setGender**(**String gender**)** **{**

**this.**gender **=** gender**;**

**}**

public String**[]** getSkills**()** **{**

**return** skills**;**

**}**

public void setSkills**(**String**[]** skills**)** **{**

**this.**skills **=** skills**;**

**this.**skillsString **=** **this.**joinString**(**skills**);**

**}**

private String joinString**(**String**[]** skills**)** **{**

**if** **(**skills **==** **null)** **{**

**return** ""**;**

**}**

**return** String**.**join**(**","**,** skills**);**

**}**

public String getSkillsString**()** **{**

**return** **this.**skillsString**;**

**}**

**}**package test**.**springmvcforms**.**model**;**

public class ApplicantInfo **{**

private String id**;**

private String name**;**

private String email**;**

private String position**;**

private String gender**;**

private String skillsString**;**

private String**[]** skills**;**

public ApplicantInfo**()** **{**

**}**

// Không được thay đổi Constructor này,

// Nó được sử dụng trong Hibernate Query.

public ApplicantInfo**(**String id**,** String name**,** String email**,** String gender**,** String position**,** String skillsString**)** **{**

**this.**id **=** id**;**

**this.**name **=** name**;**

**this.**email **=** email**;**

**this.**position **=** position**;**

**this.**gender **=** gender**;**

**this.**skillsString **=** skillsString**;**

**this.**skills **=** toArray**(**skillsString**);**

**}**

private String**[]** toArray**(**String skillStrings**)** **{**

**if** **(**skillStrings **==** **null)** **{**

**return** **new** String**[**0**];**

**}**

String**[]** ret **=** skillStrings**.**split**(**","**);**

**return** ret**;**

**}**

public String getId**()** **{**

**return** id**;**

**}**

public void setId**(**String id**)** **{**

**this.**id **=** id**;**

**}**

public String getName**()** **{**

**return** name**;**

**}**

public void setName**(**String name**)** **{**

**this.**name **=** name**;**

**}**

public String getEmail**()** **{**

**return** email**;**

**}**

public void setEmail**(**String email**)** **{**

**this.**email **=** email**;**

**}**

public String getPosition**()** **{**

**return** position**;**

**}**

public void setPosition**(**String position**)** **{**

**this.**position **=** position**;**

**}**

public String getGender**()** **{**

**return** gender**;**

**}**

public void setGender**(**String gender**)** **{**

**this.**gender **=** gender**;**

**}**

public String**[]** getSkills**()** **{**

**return** skills**;**

**}**

public void setSkills**(**String**[]** skills**)** **{**

**this.**skills **=** skills**;**

**this.**skillsString **=** **this.**joinString**(**skills**);**

**}**

private String joinString**(**String**[]** skills**)** **{**

**if** **(**skills **==** **null)** **{**

**return** ""**;**

**}**

**return** String**.**join**(**","**,** skills**);**

**}**

public String getSkillsString**()** **{**

**return** **this.**skillsString**;**

**}**

**}**

### ApplicantDAO.java

package test**.**springmvcforms**.**dao**;**

**import** java**.**util**.**List**;**

**import** test**.**springmvcforms**.**entity**.**Applicant**;**

**import** test**.**springmvcforms**.**model**.**ApplicantInfo**;**

public interface ApplicantDAO **{**

public Applicant findApplicant**(**String id**);**

public List**<**ApplicantInfo**>** listApplicantInfos**();**

public void saveApplicant**(**ApplicantInfo applicantInfo**);**

public ApplicantInfo findApplicantInfo**(**String id**);**

public void deleteApplicant**(**String id**);**

**}**

### ApplicantDAoImpl.java

package test**.**springmvcforms**.**dao**.**impl**;**

**import** java**.**util**.**List**;**

**import** java**.**util**.**UUID**;**

**import** org**.**hibernate**.**Criteria**;**

**import** org**.**hibernate**.**Query**;**

**import** org**.**hibernate**.**Session**;**

**import** org**.**hibernate**.**SessionFactory**;**

**import** org**.**hibernate**.**criterion**.**Restrictions**;**

**import** test**.**springmvcforms**.**dao**.**ApplicantDAO**;**

**import** test**.**springmvcforms**.**entity**.**Applicant**;**

**import** test**.**springmvcforms**.**model**.**ApplicantInfo**;**

**import** org**.**springframework**.**beans**.**factory**.**annotation**.**Autowired**;**

public class ApplicantDAOImpl **implements** ApplicantDAO **{**

@Autowired

private SessionFactory sessionFactory**;**

@Override

public Applicant findApplicant**(**String id**)** **{**

Session session **=** sessionFactory**.**getCurrentSession**();**

Criteria crit **=** session**.**createCriteria**(**Applicant**.**class**);**

crit**.**add**(**Restrictions**.**eq**(**"id"**,** id**));**

**return** **(**Applicant**)** crit**.**uniqueResult**();**

**}**

@Override

public ApplicantInfo findApplicantInfo**(**String id**)** **{**

Applicant applicant **=** **this.**findApplicant**(**id**);**

**if** **(**applicant **==** **null)** **{**

**return** **null;**

**}**

**return** **new** ApplicantInfo**(**applicant**.**getId**(),** applicant**.**getName**(),** //

applicant**.**getEmail**(),** applicant**.**getGender**(),** //

applicant**.**getPosition**(),** applicant**.**getSkills**());**

**}**

@Override

public List**<**ApplicantInfo**>** listApplicantInfos**()** **{**

String sql **=** "Select new " **+** ApplicantInfo**.**class**.**getName**()**//

**+** "(a.id, a.name, a.email, a.gender, a.position, a.skills) "//

**+** " from " **+** Applicant**.**class**.**getName**()** **+** " a "**;**

Session session **=** sessionFactory**.**getCurrentSession**();**

Query query **=** session**.**createQuery**(**sql**);**

**return** query**.**list**();**

**}**

public void saveApplicant**(**ApplicantInfo applicantInfo**)** **{**

String id **=** applicantInfo**.**getId**();**

Applicant applicant **=** **null;**

**if** **(**id **!=** **null)** **{**

applicant **=** **this.**findApplicant**(**id**);**

**}**

boolean isNew **=** **false;**

**if** **(**applicant **==** **null)** **{**

isNew **=** **true;**

applicant **=** **new** Applicant**();**

applicant**.**setId**(**UUID**.**randomUUID**().**toString**());**

**}**

applicant**.**setEmail**(**applicantInfo**.**getEmail**());**

applicant**.**setGender**(**applicantInfo**.**getGender**());**

applicant**.**setName**(**applicantInfo**.**getName**());**

applicant**.**setPosition**(**applicantInfo**.**getPosition**());**

String skillsString **=** applicantInfo**.**getSkillsString**();**

applicant**.**setSkills**(**skillsString**);**

//

**if** **(**isNew**)** **{**

Session session **=** **this.**sessionFactory**.**getCurrentSession**();**

session**.**persist**(**applicant**);**

**}**

**}**

@Override

public void deleteApplicant**(**String id**)** **{**

Applicant applicant **=** **this.**findApplicant**(**id**);**

**if** **(**applicant **!=** **null)** **{**

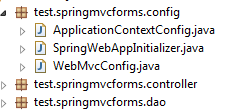
**this.**sessionFactory**.**getCurrentSession**().**delete**(**applicant**);**

**}**

**}**

**}**

## Config Spring MVC



### SpringWebAppInitializer.java

package test**.**springmvcforms**.**config**;**

**import** javax**.**servlet**.**ServletContext**;**

**import** javax**.**servlet**.**ServletException**;**

**import** javax**.**servlet**.**ServletRegistration**;**

**import** org**.**springframework**.**web**.**WebApplicationInitializer**;**

**import** org**.**springframework**.**web**.**context**.**support**.**AnnotationConfigWebApplicationContext**;**

**import** org**.**springframework**.**web**.**servlet**.**DispatcherServlet**;**

public class SpringWebAppInitializer **implements** WebApplicationInitializer **{**

@Override

public void onStartup**(**ServletContext servletContext**)** **throws** ServletException **{**

AnnotationConfigWebApplicationContext appContext **=** **new** AnnotationConfigWebApplicationContext**();**

appContext**.**register**(**ApplicationContextConfig**.**class**);**

ServletRegistration**.**Dynamic dispatcher **=** servletContext**.**addServlet**(**"SpringDispatcher"**,**

**new** DispatcherServlet**(**appContext**));**

dispatcher**.**setLoadOnStartup**(**1**);**

dispatcher**.**addMapping**(**"/"**);**

**}**

**}**

### ApplicationContextConfig.java

package test**.**springmvcforms**.**config**;**

**import** java**.**util**.**Properties**;**

**import** javax**.**sql**.**DataSource**;**

**import** org**.**hibernate**.**SessionFactory**;**

**import** test**.**springmvcforms**.**dao**.**ApplicantDAO**;**

**import** test**.**springmvcforms**.**dao**.**impl**.**ApplicantDAOImpl**;**

**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**.**context**.**support**.**ResourceBundleMessageSource**;**

**import** org**.**springframework**.**core**.**env**.**Environment**;**

**import** org**.**springframework**.**jdbc**.**datasource**.**DriverManagerDataSource**;**

**import** org**.**springframework**.**orm**.**hibernate5**.**HibernateTransactionManager**;**

**import** org**.**springframework**.**orm**.**hibernate5**.**LocalSessionFactoryBean**;**

**import** org**.**springframework**.**transaction**.**annotation**.**EnableTransactionManagement**;**

**import** org**.**springframework**.**web**.**servlet**.**view**.**InternalResourceViewResolver**;**

@Configuration

@ComponentScan**(**"test.springmvcforms.\*"**)**

@EnableTransactionManagement

// Load to Environment.

@PropertySource**(**"classpath:ds-hibernate-cfg.properties"**)**

public class ApplicationContextConfig **{**

// Lưu trữ các giá thuộc tính load bởi @PropertySource.

@Autowired

private Environment env**;**

@Bean

public ResourceBundleMessageSource messageSource**()** **{**

ResourceBundleMessageSource rb **=** **new** ResourceBundleMessageSource**();**

// Load property in message/validator.properties

rb**.**setBasenames**(new** String**[]** **{** "messages/validator"**});**

**return** rb**;**

**}**

@Bean**(**name **=** "viewResolver"**)**

public InternalResourceViewResolver getViewResolver**()** **{**

InternalResourceViewResolver viewResolver **=** **new** InternalResourceViewResolver**();**

viewResolver**.**setPrefix**(**"/WEB-INF/pages/"**);**

viewResolver**.**setSuffix**(**".jsp"**);**

**return** viewResolver**;**

**}**

@Bean**(**name **=** "dataSource"**)**

public DataSource getDataSource**()** **{**

DriverManagerDataSource dataSource **=** **new** DriverManagerDataSource**();**

dataSource**.**setDriverClassName**(**env**.**getProperty**(**"ds.database-driver"**));**

dataSource**.**setUrl**(**env**.**getProperty**(**"ds.url"**));**

dataSource**.**setUsername**(**env**.**getProperty**(**"ds.username"**));**

dataSource**.**setPassword**(**env**.**getProperty**(**"ds.password"**));**

**return** dataSource**;**

**}**

@Autowired

@Bean**(**name **=** "sessionFactory"**)**

public SessionFactory getSessionFactory**(**DataSource dataSource**)** **throws** Exception **{**

Properties properties **=** **new** Properties**();**

// Xem: ds-hibernate-cfg.properties

properties**.**put**(**"hibernate.dialect"**,** env**.**getProperty**(**"hibernate.dialect"**));**

properties**.**put**(**"hibernate.show\_sql"**,** env**.**getProperty**(**"hibernate.show\_sql"**));**

properties**.**put**(**"current\_session\_context\_class"**,** env**.**getProperty**(**"current\_session\_context\_class"**));**

LocalSessionFactoryBean factoryBean **=** **new** LocalSessionFactoryBean**();**

factoryBean**.**setPackagesToScan**(new** String**[]** **{** "test.springmvcforms.entity" **});**

factoryBean**.**setDataSource**(**dataSource**);**

factoryBean**.**setHibernateProperties**(**properties**);**

factoryBean**.**afterPropertiesSet**();**

//

SessionFactory sf **=** factoryBean**.**getObject**();**

**return** sf**;**

**}**

@Autowired

@Bean**(**name **=** "transactionManager"**)**

public HibernateTransactionManager getTransactionManager**(**SessionFactory sessionFactory**)** **{**

HibernateTransactionManager transactionManager **=** **new** HibernateTransactionManager**(**sessionFactory**);**

**return** transactionManager**;**

**}**

@Bean**(**name **=** "applicantDAO"**)**

public ApplicantDAO getApplicantDAO**()** **{**

**return** **new** ApplicantDAOImpl**();**

**}**

**}**

Ds-hibernate-cfg.properties(MySQL)

# DataSource

ds.database-driver=com.mysql.jdbc.Driver

ds.url=jdbc:mysql://localhost:3306/applicants

ds.username=root

ds.password=chorangho

# Hibernate Config

hibernate.dialect=org.hibernate.dialect.MySQLDialect

hibernate.show\_sql=true

current\_session\_context\_class=thread

### WebMvcConfig.Java

package test**.**springmvcforms**.**config**;**

**import** org**.**springframework**.**context**.**annotation**.**Configuration**;**

**import** org**.**springframework**.**web**.**servlet**.**config**.**annotation**.**DefaultServletHandlerConfigurer**;**

**import** org**.**springframework**.**web**.**servlet**.**config**.**annotation**.**EnableWebMvc**;**

**import** org**.**springframework**.**web**.**servlet**.**config**.**annotation**.**ResourceHandlerRegistry**;**

**import** org**.**springframework**.**web**.**servlet**.**config**.**annotation**.**WebMvcConfigurerAdapter**;**

@Configuration

@EnableWebMvc

public class WebMvcConfig **extends** WebMvcConfigurerAdapter **{**

// Cấu hình để sử dụng các file nguồn tĩnh (html, image, ..)

// Tương đương với <mvc:resources/> trong cấu hình sử dụng XML.

@Override

public void addResourceHandlers**(**ResourceHandlerRegistry registry**)** **{**

registry**.**addResourceHandler**(**"/css/\*\*"**).**addResourceLocations**(**"/css/"**).**setCachePeriod**(**31556926**);**

registry**.**addResourceHandler**(**"/img/\*\*"**).**addResourceLocations**(**"/img/"**).**setCachePeriod**(**31556926**);**

registry**.**addResourceHandler**(**"/js/\*\*"**).**addResourceLocations**(**"/js/"**).**setCachePeriod**(**31556926**);**

**}**

// equivalent for <mvc:default-servlet-handler/> tag

@Override

public void configureDefaultServletHandling**(**DefaultServletHandlerConfigurer configurer**)** **{**

configurer**.**enable**();**

**}**

**}**

### 7 MVC Spring Validator

ApplicantValidator.java will be use to validate the data that user inputs to the form. In case of wrong inputs , the class will show errors :



package test**.**springmvcforms**.**validator**;**

**import** org**.**apache**.**commons**.**validator**.**routines**.**EmailValidator**;**

**import** test**.**springmvcforms**.**model**.**ApplicantInfo**;**

**import** org**.**springframework**.**stereotype**.**Component**;**

**import** org**.**springframework**.**validation**.**Errors**;**

**import** org**.**springframework**.**validation**.**ValidationUtils**;**

**import** org**.**springframework**.**validation**.**Validator**;**

@Component

public class ApplicantValidator **implements** Validator **{**

// common-validator library.

private EmailValidator emailValidator **=** EmailValidator**.**getInstance**();**

// Các class được hỗ trợ Validate

@Override

public boolean supports**(**Class**<?>** clazz**)** **{**

**return** clazz **==** ApplicantInfo**.**class**;**

**}**

@Override

public void validate**(**Object target**,** Errors errors**)** **{**

ApplicantInfo applicantInfo **=** **(**ApplicantInfo**)** target**;**

// Kiểm tra các field của ApplicantInfo.

// (Xem thêm file property: messages/validator.property)

ValidationUtils**.**rejectIfEmptyOrWhitespace**(**errors**,** "name"**,** "NotEmpty.applicantForm.name"**);**

ValidationUtils**.**rejectIfEmptyOrWhitespace**(**errors**,** "email"**,** "NotEmpty.applicantForm.email"**);**

ValidationUtils**.**rejectIfEmptyOrWhitespace**(**errors**,** "position"**,** "NotEmpty.applicantForm.position"**);**

ValidationUtils**.**rejectIfEmptyOrWhitespace**(**errors**,** "gender"**,** "NotEmpty.applicantForm.gender"**);**

**if(!**emailValidator**.**isValid**(**applicantInfo**.**getEmail**()))** **{**

// Error in email field.

// Lỗi trường email.

errors**.**rejectValue**(**"email"**,** "Pattern.applicantForm.email"**);**

**}**

**if(**applicantInfo**.**getSkills**()==** **null** **||** applicantInfo**.**getSkills**().**length**==**0 **)** **{**

errors**.**rejectValue**(**"skills"**,** "Select.applicantForm.skills"**);**

**}**

**}**

**}**

Error notifications are edited in :

#### 

#### Messages/validator.properties

NotEmpty.applicantForm.name=Name is required

NotEmpty.applicantForm.email=Email is required

NotEmpty.applicantForm.position=Position is required

NotEmpty.applicantForm.gender=Gender is required

Pattern.applicantForm.email=Email is not valid

Select.applicantForm.skills=Must select at least one skill

The content of validator.properties are loaded by **ApplicationContextConfig** (already inserted in the previous step )

// Load property in message/validator.properties

@Bean

public ResourceBundleMessageSource messageSource**()** **{**

ResourceBundleMessageSource rb **=** **new** ResourceBundleMessageSource**();**

// Load property in message/validator.properties

rb**.**setBasenames**(new** String**[]** **{** "messages/validator"**});**

**return** rb**;**

**}**

## Spring Controller

MyController.java

package test**.**springmvcforms**.**controller**;**

**import** java**.**util**.**ArrayList**;**

**import** java**.**util**.**LinkedHashMap**;**

**import** java**.**util**.**List**;**

**import** java**.**util**.**Map**;**

**import** test**.**springmvcforms**.**dao**.**ApplicantDAO**;**

**import** test**.**springmvcforms**.**model**.**ApplicantInfo**;**

**import** test**.**springmvcforms**.**validator**.**ApplicantValidator**;**

**import** org**.**springframework**.**beans**.**factory**.**annotation**.**Autowired**;**

**import** org**.**springframework**.**stereotype**.**Controller**;**

**import** org**.**springframework**.**transaction**.**annotation**.**Transactional**;**

**import** org**.**springframework**.**ui**.**Model**;**

**import** org**.**springframework**.**validation**.**BindingResult**;**

**import** org**.**springframework**.**validation**.**annotation**.**Validated**;**

**import** org**.**springframework**.**web**.**bind**.**WebDataBinder**;**

**import** org**.**springframework**.**web**.**bind**.**annotation**.**InitBinder**;**

**import** org**.**springframework**.**web**.**bind**.**annotation**.**ModelAttribute**;**

**import** org**.**springframework**.**web**.**bind**.**annotation**.**RequestMapping**;**

**import** org**.**springframework**.**web**.**bind**.**annotation**.**RequestMethod**;**

**import** org**.**springframework**.**web**.**bind**.**annotation**.**RequestParam**;**

**import** org**.**springframework**.**web**.**servlet**.**config**.**annotation**.**EnableWebMvc**;**

**import** org**.**springframework**.**web**.**servlet**.**mvc**.**support**.**RedirectAttributes**;**

@Controller

//Cần thiết cho Hibernate Transaction.

@Transactional

//Cần thiết để sử dụng RedirectAttributes

@EnableWebMvc

public class MyController **{**

@Autowired

private ApplicantDAO applicantDAO**;**

@Autowired

private ApplicantValidator applicantValidator**;**

@RequestMapping**(**"/"**)**

public String homePage**(**Model model**)** **{**

**return** applicantList**(**model**);**

**}**

@RequestMapping**(**"/applicantList"**)**

public String applicantList**(**Model model**)** **{**

List**<**ApplicantInfo**>** list **=** applicantDAO**.**listApplicantInfos**();**

model**.**addAttribute**(**"applicantInfos"**,** list**);**

**return** "applicantList"**;**

**}**

private Map**<**String**,** String**>** dataForPositions**()** **{**

Map**<**String**,** String**>** positionMap **=** **new** LinkedHashMap**<**String**,** String**>();**

positionMap**.**put**(**"Developer"**,** "Developer"**);**

positionMap**.**put**(**"Leader"**,** "Leader"**);**

positionMap**.**put**(**"Tester"**,** "Tester"**);**

**return** positionMap**;**

**}**

private List**<**String**>** dataForSkills**()** **{**

List**<**String**>** list **=** **new** ArrayList**<**String**>();**

list**.**add**(**"Java"**);**

list**.**add**(**"C/C++"**);**

list**.**add**(**"C#"**);**

**return** list**;**

**}**

private String formApplicant**(**Model model**,** ApplicantInfo applicantInfo**)** **{**

model**.**addAttribute**(**"applicantForm"**,** applicantInfo**);**

Map**<**String**,** String**>** positionMap **=** **this.**dataForPositions**();**

model**.**addAttribute**(**"positionMap"**,** positionMap**);**

List**<**String**>** list **=** dataForSkills**();**

model**.**addAttribute**(**"skills"**,** list**);**

**if** **(**applicantInfo**.**getId**()** **==** **null)** **{**

model**.**addAttribute**(**"formTitle"**,** "Create Applicant"**);**

**}** **else** **{**

model**.**addAttribute**(**"formTitle"**,** "Edit Applicant"**);**

**}**

**return** "formApplicant"**;**

**}**

@RequestMapping**(**"/createApplicant"**)**

public String createApplicant**(**Model model**)** **{**

ApplicantInfo applicantInfo **=** **new** ApplicantInfo**();**

**return** **this.**formApplicant**(**model**,** applicantInfo**);**

**}**

@RequestMapping**(**"/editApplicant"**)**

public String editApplicant**(**Model model**,** @RequestParam**(**"id"**)** String id**)** **{**

ApplicantInfo applicantInfo **=** **null;**

**if** **(**id **!=** **null)** **{**

applicantInfo **=** **this.**applicantDAO**.**findApplicantInfo**(**id**);**

**}**

**if** **(**applicantInfo **==** **null)** **{**

**return** "redirect:/applicantList"**;**

**}**

**return** **this.**formApplicant**(**model**,** applicantInfo**);**

**}**

@RequestMapping**(**"/deleteApplicant"**)**

public String deleteApplicant**(**Model model**,** @RequestParam**(**"id"**)** String id**)** **{**

**if** **(**id **!=** **null)** **{**

**this.**applicantDAO**.**deleteApplicant**(**id**);**

**}**

**return** "redirect:/applicantList"**;**

**}**

// Set a form validator

@InitBinder

protected void initBinder**(**WebDataBinder dataBinder**)** **{**

// Form mục tiêu

Object target **=** dataBinder**.**getTarget**();**

**if** **(**target **==** **null)** **{**

**return;**

**}**

System**.**out**.**println**(**"Target=" **+** target**);**

**if** **(**target**.**getClass**()** **==** ApplicantInfo**.**class**)** **{**

dataBinder**.**setValidator**(**applicantValidator**);**

**}**

**}**

// Save or update Applicant

// 1. @ModelAttribute bind form value

// 2. @Validated form validator

// 3. RedirectAttributes for flash value

@RequestMapping**(**value **=** "/saveApplicant"**,** method **=** RequestMethod**.**POST**)**

public String saveApplicant**(**Model model**,** //

@ModelAttribute**(**"applicantForm"**)** @Validated ApplicantInfo applicantInfo**,** //

BindingResult result**,** //

final RedirectAttributes redirectAttributes**)** **{**

// Nếu validate có lỗi.

**if** **(**result**.**hasErrors**())** **{**

**return** **this.**formApplicant**(**model**,** applicantInfo**);**

**}**

**this.**applicantDAO**.**saveApplicant**(**applicantInfo**);**

// Important!!: Need @EnableWebMvc

// Add message to flash scope

redirectAttributes**.**addFlashAttribute**(**"message"**,** "Save Applicant Successful"**);**

**return** "redirect:/applicantList"**;**

**}**

**}**

## JSP files (View )

WEB-INF/pages/applicantList.jsp

**<%@** taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" **%>**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv=**"Content-Type"** content=**"text/html; charset=UTF-8"**>

<title>**Applicant List**</title>

<style>

**table {**

**margin-top: 10px;**

**border: solid black 1px;**

**}**

**table td {**

**padding: 5px;**

**}**

**.message {**

**font-size:90%;**

**color:blue;**

**font-style:italic;**

**margin-top:30px;**

**}**

</style>

</head>

<body>

<a href=**"${pageContext.request.contextPath}/createApplicant"**>**Create Applicant**</a>

<br/>

<table border=**"1"**>

<tr>

<th>**Name**</th>

<th>**Position**</th>

<th>**Gender**</th>

<th>**Email**</th>

<th>**Skills**</th>

<th>**Edit**</th>

<th>**Delete**</th>

</tr>

<c:forEach items="${applicantInfos}" var="info">

<tr>

<td> **${info.name}** </td>

<td> **${info.position}** </td>

<td> **${info.gender}** </td>

<td> **${info.email}** </td>

<td> **${info.skillsString}** </td>

<td><a href=**"deleteApplicant?id=${info.id}"**>**Delete**</a></td>

<td><a href=**"editApplicant?id=${info.id}"**>**Edit**</a></td>

</tr>

</c:forEach>

</table>

<c:if test="${not empty message}">

<div class=**"message"**>**${message}**</div>

</c:if>

</body>

</html>

WEB-INF/pages/formApplicant.jsp

**<%@** taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"**%>**

**<%@** taglib prefix="spring" uri="http://www.springframework.org/tags"**%>**

**<%@** taglib prefix="form" uri="http://www.springframework.org/tags/form"**%>**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv=**"Content-Type"** content=**"text/html; charset=UTF-8"**>

<title>**Create Applicant**</title>

<style>

**.error-message {**

**color: red;**

**font-size:90%;**

**font-style: italic;**

**}**

</style>

</head>

<body>

<h3>**${formTitle}**</h3>

<form:form action="saveApplicant" method="POST"

modelAttribute="applicantForm">

<form:hidden path="id" />

<table>

<tr>

<td>**Name**</td>

<td><form:input path="name" /></td>

<td><form:errors path="name"

class="error-message" /></td>

</tr>

<tr>

<td>**Gender**</td>

<td><form:select path="gender">

<form:option value="" label="- Gender -" />

<form:option value="M" label="Male" />

<form:option value="F" label="Female" />

</form:select></td>

<td><form:errors path="gender" class="error-message" /></td>

</tr>

<tr>

<td>**Email**</td>

<td><form:input path="email" /></td>

<td><form:errors path="email" class="error-message" /></td>

</tr>

<tr>

<td>**Position**</td>

<td><form:select path="position">

<form:options items="${positionMap}" />

</form:select></td>

<td><form:errors path="position" class="error-message" /></td>

</tr>

<tr>

<td>**Skills**</td>

<td><c:forEach items="${skills}" var="skill">

<form:checkbox path="skills" value="${skill}" label="${skill}" />

</c:forEach></td>

<td><form:errors path="skills" class="error-message" /></td>

</tr>

<tr>

<td>&nbsp;</td>

<td><input type=**"submit"** value=**"Submit"** />

<a href=**"${pageContext.request.contextPath}/applicantList"**>**Cancel**</a>

</td>

<td>&nbsp;</td>

</tr>

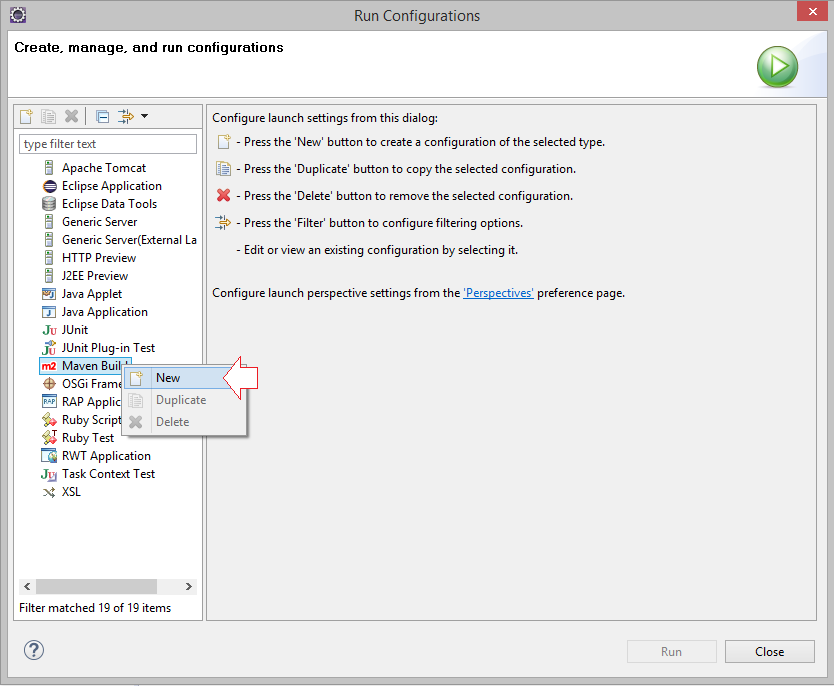
</table>

</form:form>

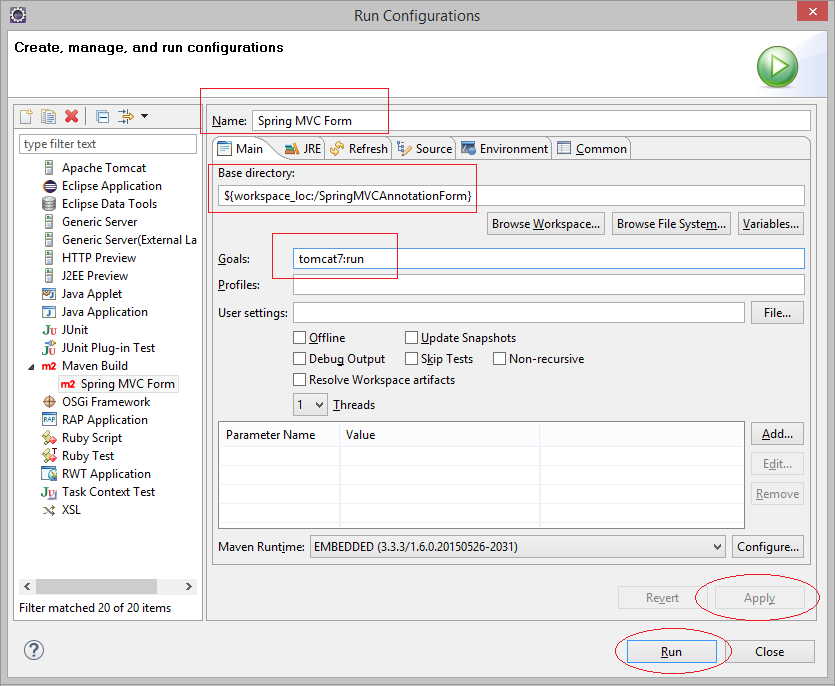
</body>

</html>

### Run project



* **Name:** Spring MVC Form
* **Base directory:** ${workspace\_loc:/SpringMVCAnnotationForm}
* **Goals:** tomcat7:run



<http://localhost:8080/SpringMVCAnnotationForm/applicantList>