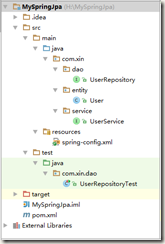
<http://www.cnblogs.com/weilunhui/p/5025383.html>

**1. 在IntelliJ中新建maven项目**

给出一个建好的示例，（本示例中省略了业务逻辑组件UserService）

[](http://images2015.cnblogs.com/blog/563063/201512/563063-20151207102324293-1281274379.png)

**2. 在pom.xml中配置依赖**

[复制代码](javascript:void(0);)

包括：

spring-context

spring-orm

spring-data-jpa

hibernate-core

hibernate-entitymanager

mysql-connector-java

commons-dbcp

junit

spring-test

[复制代码](javascript:void(0);)

代码如下：

|  |  |
| --- | --- |
| 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 | <dependencies>      <!-- Spring & Spring Data JPA -->      <dependency>          <groupId>org.springframework</groupId>          <artifactId>spring-context</artifactId>          <version>4.2.3.RELEASE</version>      </dependency>      <dependency>          <groupId>org.springframework</groupId>          <artifactId>spring-orm</artifactId>          <version>4.2.3.RELEASE</version>      </dependency>      <dependency>          <groupId>org.springframework.data</groupId>          <artifactId>spring-data-jpa</artifactId>          <version>1.9.1.RELEASE</version>      </dependency>        <!-- Hibernate -->      <dependency>          <groupId>org.hibernate</groupId>          <artifactId>hibernate-core</artifactId>          <version>4.3.11.Final</version>      </dependency>      <dependency>          <groupId>org.hibernate</groupId>          <artifactId>hibernate-entitymanager</artifactId>          <version>4.3.11.Final</version>      </dependency>        <!-- MySQL -->      <dependency>          <groupId>mysql</groupId>          <artifactId>mysql-connector-java</artifactId>          <version>5.1.37</version>      </dependency>      <dependency>          <groupId>commons-dbcp</groupId>          <artifactId>commons-dbcp</artifactId>          <version>1.4</version>      </dependency>        <!-- Junit -->      <dependency>          <groupId>junit</groupId>          <artifactId>junit</artifactId>          <version>4.12</version>      </dependency>      <dependency>          <groupId>org.springframework</groupId>          <artifactId>spring-test</artifactId>          <version>4.2.3.RELEASE</version>      </dependency>  </dependencies> |

**3. resources右键new一个Xml Configuration File--Spring Config配置文件：spring-config.xml**（或者applicationContext.xml）

    配置dataSource、entityManagerFactory、transactionManager等。

|  |  |
| --- | --- |
| 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 | <?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:jpa="<http://www.springframework.org/schema/data/jpa>"         xmlns:tx="<http://www.springframework.org/schema/tx>"         xsi:schemaLocation="<http://www.springframework.org/schema/beans>  <http://www.springframework.org/schema/beans/spring-beans.xsd> <http://www.springframework.org/schema/data/jpa>  <http://www.springframework.org/schema/data/jpa/spring-jpa.xsd> <http://www.springframework.org/schema/tx> <http://www.springframework.org/schema/tx/spring-tx.xsd>">        <jpa:repositories base-package="com.xin"/>        <bean id="dataSource" 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/javaee>"/>          <property name="username" value="root"/>          <property name="password" value="root"/>      </bean>        <bean id="entityManagerFactory" class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">          <property name="dataSource" ref="dataSource"/>          <property name="packagesToScan" value="com.xin"/>          <property name="jpaVendorAdapter">              <bean class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter">                  <property name="database" value="MYSQL"/>                  <property name="generateDdl" value="true"/>                  <property name="showSql" value="true"/>                  <property name="databasePlatform" value="org.hibernate.dialect.MySQL5InnoDBDialect"/>              </bean>          </property>          <property name="jpaProperties">              <value>                  hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect                  hibernate.hbm2ddl.auto=update                  hibernate.show\_sql=true    <!--此处的设置可以覆盖jpaVendorAdapter中的相关设置-->                  <!--hibernate.format\_sql=true-->              </value>          </property>      </bean>        <!--配置针对JPA的局部事务管理器-->      <bean id="transactionManager" class="org.springframework.orm.jpa.JpaTransactionManager">          <property name="entityManagerFactory" ref="entityManagerFactory"/>      </bean>      <!--根据事务注解来生成事务代理-->      <tx:annotation-driven transaction-manager="transactionManager"/>  </beans> |

**4. 新建User实体类**，并进行Annotation注解

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | @Entity  public class User {      @Id      @GeneratedValue(strategy = GenerationType.AUTO)      private Long id;      private String firstname, lastname;      private int age;      private String address;  //省略构造方法和setter、getter方法  } |

**5. 新建DAO组件：UserRepository接口**

让此接口继承Repository、CrudRepository、PagingAndSortingRepositories 或者 JpaRepository。JpaRepository接口有一实现类SimpleJpaRepository，上述接口声明的方法均由该类实现。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | public interface UserRepository extends JpaRepository<User, Long> {      //可以根据方法名进行衍生查询，也可以自定义查询。      @Query("select u from User u where u.lastname=?1")      List<User> findByLastname(String lastname);        Long countByLastname(String lastname);  } |

**6.** 接下来便可使用**junit单元测试类**进行测试了

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | @RunWith(SpringJUnit4ClassRunner.class)  @ContextConfiguration(locations = "classpath:spring-config.xml")  public class UserRepositoryTest {      @Autowired      private UserRepository userRepository;        @Test      public void testFindByLastname() throws Exception {          User user = new User("lunhui","wei",28, "nanle");          userRepository.save(user);      }  } |