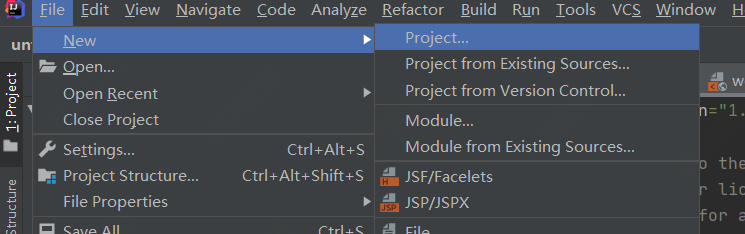
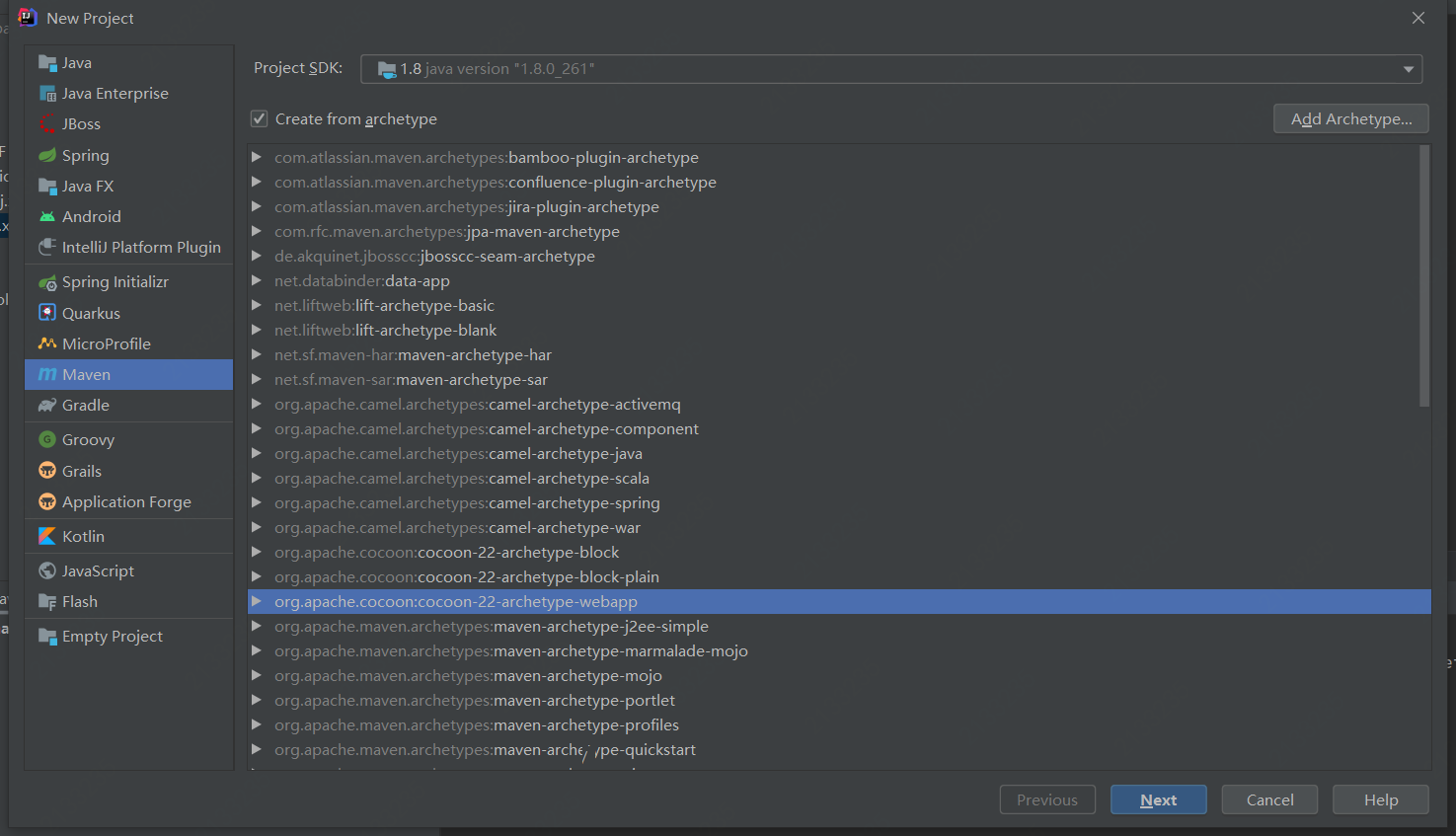
软件：idea2020，mysql5.7，maven，tomcat8.5

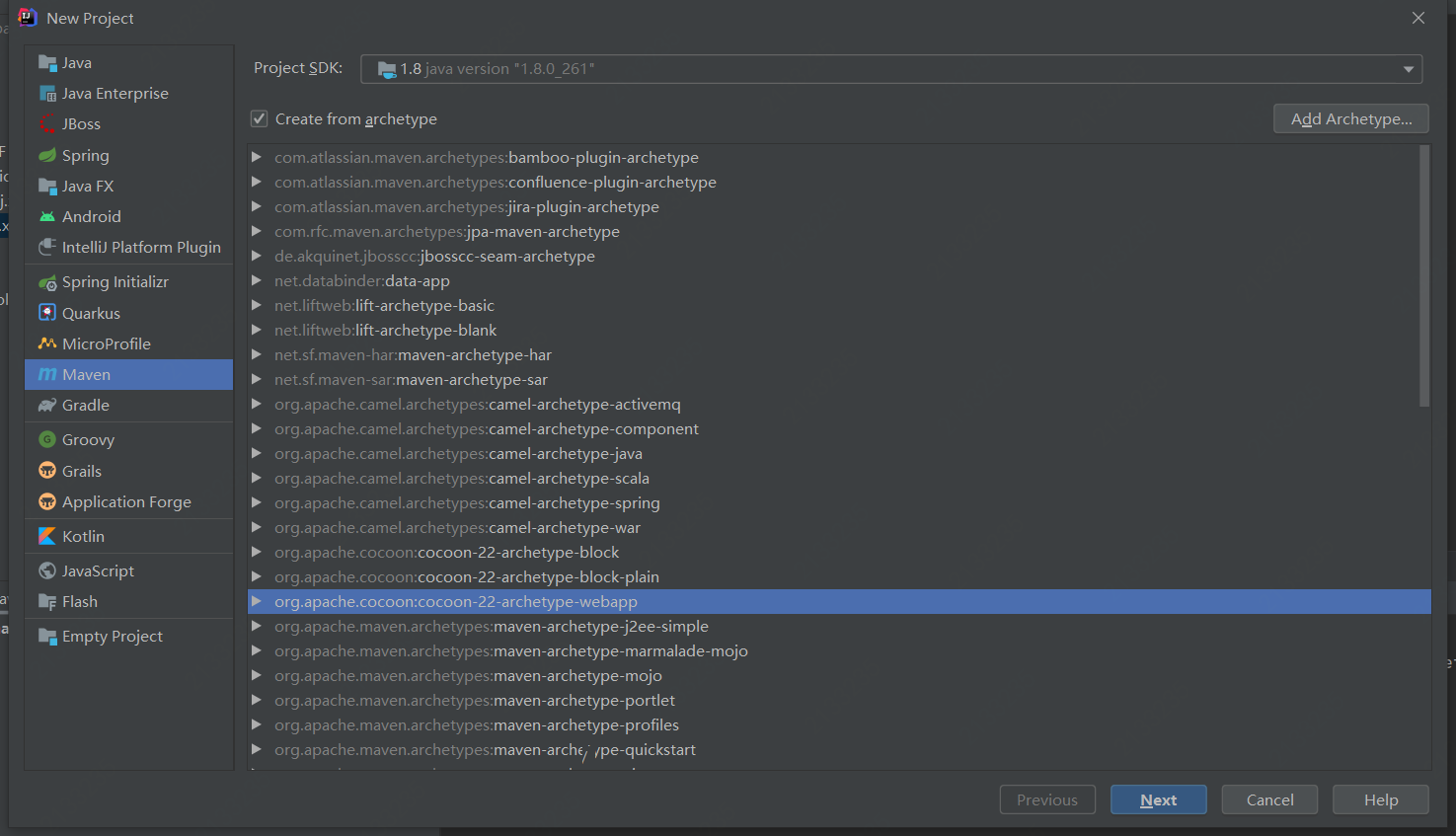
1. 创建maven项目



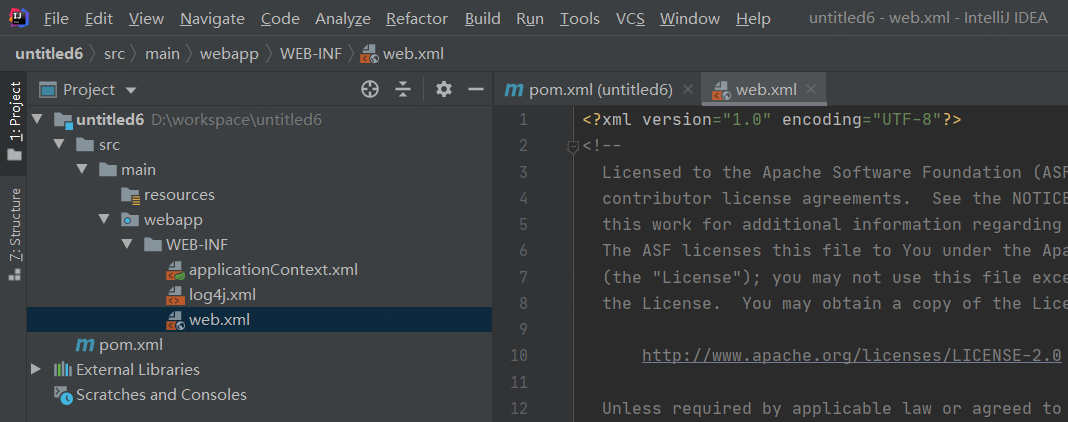
注意打勾



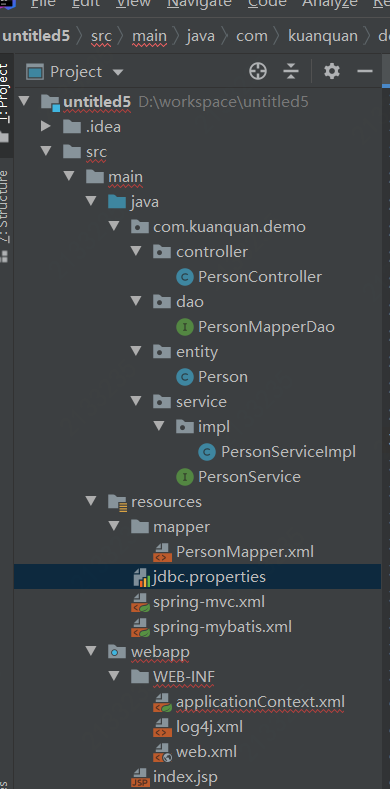
一路next



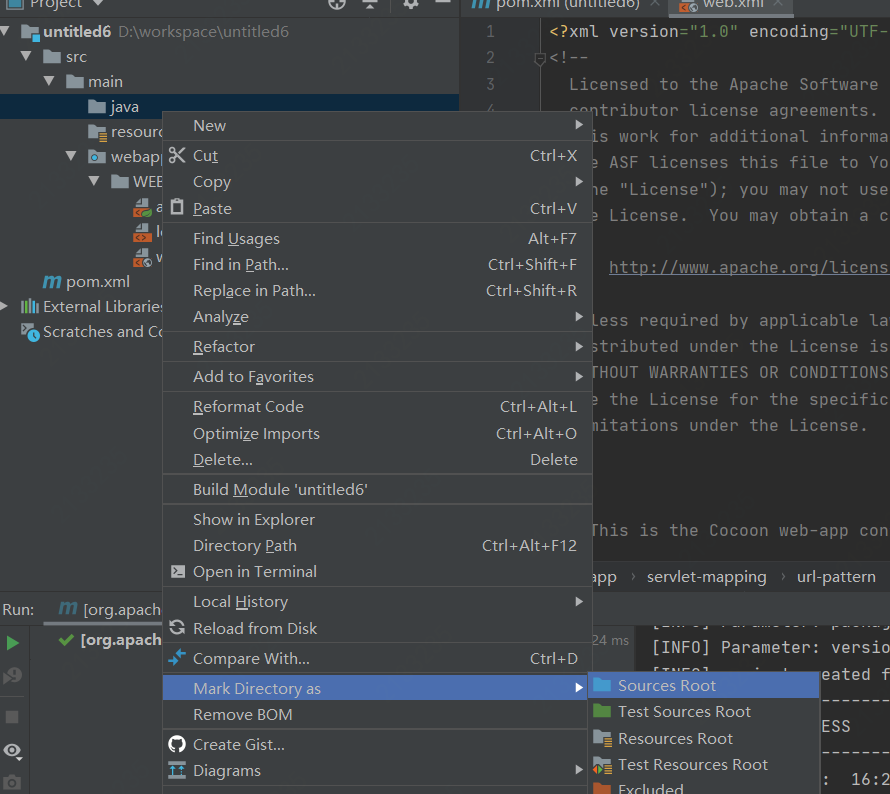
二．建成的maven项目如图



目标架构



新建上述文件，



三．建文件

文件：

PersonController

package com.kuanquan.demo.controller;  
  
import com.alibaba.fastjson.JSON;  
import com.fasterxml.jackson.databind.ObjectMapper;  
import com.kuanquan.demo.entity.Person;  
import com.kuanquan.demo.service.PersonService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
  
  
@Controller  
@RequestMapping("/person")  
public class PersonController {  
  
 @Autowired  
 private PersonService personService;  
 @RequestMapping(value = "/selectPerson",method = RequestMethod.*GET*)  
 public void selectPerson(HttpServletRequest request, HttpServletResponse response) throws IOException {  
  
 request.setCharacterEncoding("utf-8");  
 response.setHeader("content-type", "text/html;charset=UTF-8");  
 long personId = Long.*parseLong*(String.*valueOf*(1));  
 Person person = personService.findPersonById(personId);  
  
 String s = JSON.*toJSONString*(person);  
 System.*out*.println(s);  
 ObjectMapper mapper = new ObjectMapper();  
  
// response.getWriter().write(mapper.writeValueAsString(person));  
 response.getWriter().write(s);  
 response.getWriter().close();  
 }  
  
  
 @RequestMapping(value = "/selectPersonName",method = RequestMethod.*GET*)  
 public void selectPersonName(HttpServletRequest request, HttpServletResponse response) throws IOException {  
  
 request.setCharacterEncoding("utf-8");  
 response.setHeader("content-type", "text/html;charset=UTF-8");  
 long personId = Long.*parseLong*(request.getParameter("id"));  
 String person = personService.findPersonName(personId);  
  
// String s = JSON.toJSONString(person);  
// System.out.println(s);  
// ObjectMapper mapper = new ObjectMapper();  
  
// response.getWriter().write(mapper.writeValueAsString(person));  
 response.getWriter().write(person);  
 response.getWriter().close();  
 }  
}

PersonMapperDao

package com.kuanquan.demo.dao;  
  
import com.kuanquan.demo.entity.Person;  
import org.springframework.stereotype.Repository;  
  
@Repository("personMapperDao")  
public interface PersonMapperDao {  
 Person findPersonById(long id);  
 String findPersonName(long id);  
 void deletePerson(long id);  
 void updataPerson(Person person);  
 void insertPerson(Person person);  
}

Person

package com.kuanquan.demo.entity;  
  
  
*/\*\*  
 \* 数据库表中字段的映射  
 \* 类名一般都是用表明命名的，好区分  
 \*/*public class Person {  
  
 // 下面都是数据库表中每一列字段  
 private int user\_id;  
 private String name;  
 private int age;  
 private int sex;  
 private String create\_time;  
  
  
 public int getUser\_id() {  
 return user\_id;  
 }  
  
 public void setUser\_id(int user\_id) {  
 this.user\_id = user\_id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public int getAge() {  
 return age;  
 }  
  
 public void setAge(int age) {  
 this.age = age;  
 }  
  
 public int getSex() {  
 return sex;  
 }  
  
 public void setSex(int sex) {  
 this.sex = sex;  
 }  
  
 public String getCreate\_time() {  
 return create\_time;  
 }  
  
 public void setCreate\_time(String create\_time) {  
 this.create\_time = create\_time;  
 }  
  
  
  
 @Override  
 public String toString() {  
 return "Person{" +  
 "user\_id=" + user\_id +  
 ", name='" + name + '\'' +  
 ", age=" + age +  
 ", sex=" + sex +  
 ", create\_time='" + create\_time + '\'' +  
  
 '}';  
 }  
}

PersonServiceImpl

package com.kuanquan.demo.service.impl;  
  
import com.kuanquan.demo.dao.PersonMapperDao;  
import com.kuanquan.demo.entity.Person;  
import com.kuanquan.demo.service.PersonService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service("personServiceImpl")  
public class PersonServiceImpl implements PersonService {  
 @Autowired  
 private PersonMapperDao personMapperDao;  
 public Person findPersonById(long id) {  
 return personMapperDao.findPersonById(id);  
 }  
  
 @Override  
 public String findPersonName(long id) {  
 return personMapperDao.findPersonName(id);  
 }  
}

PersonService

package com.kuanquan.demo.service;  
  
import com.kuanquan.demo.entity.Person;  
  
public interface PersonService {  
 Person findPersonById(long id);  
 String findPersonName(long id);  
}

XML文件

PersonMapper.xml

<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
  
<!-- 设置为IUserDao接口方法提供sql语句配置 -->  
<mapper namespace="com.kuanquan.demo.dao.PersonMapperDao">  
  
 <select id="findPersonById" resultType="Person" parameterType="long">  
 SELECT \* FROM user WHERE user\_id = #{id}  
 </select>  
  
 <select id="findPersonName" resultType="String" parameterType="long">  
 SELECT name FROM user WHERE user\_id = #{id}  
  
  
 </select>  
  
</mapper>

Spring-mvc.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"  
 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.xsd  
 http://www.springframework.org/schema/context  
 http://www.springframework.org/schema/context/spring-context.xsd  
 http://www.springframework.org/schema/mvc  
 http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd">  
  
 <!-- 扫描web相关的bean -->  
 <context:component-scan base-package="com.kuanquan.demo"/>  
  
 <!-- 开启SpringMVC注解模式 -->  
 <mvc:annotation-driven/>  
  
 <!-- 静态资源默认servlet配置 -->  
 <mvc:default-servlet-handler/>  
  
 <!-- 配置jsp 显示ViewResolver -->  
 <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">  
 <property name="viewClass" value="org.springframework.web.servlet.view.JstlView"/>  
 <property name="prefix" value="/WEB-INF/views/"/>  
 <property name="suffix" value=""/>  
 </bean>  
</beans>

Spring-mybatis.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"  
 xmlns:context="http://www.springframework.org/schema/context"  
 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/context  
 http://www.springframework.org/schema/context/spring-context.xsd  
 http://www.springframework.org/schema/tx  
 http://www.springframework.org/schema/tx/spring-tx.xsd">  
 <!-- 扫描service包下所有使用注解的类型 -->  
 <context:component-scan base-package="com.kuanquan.demo"/>  
  
 <!-- 配置数据库相关参数properties的属性：${url} -->  
 <context:property-placeholder location="classpath:jdbc.properties"/>  
  
 <!-- 数据库连接池 -->  
 <bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource">  
 <property name="driverClass" value="${jdbc.driver}"/>  
 <property name="jdbcUrl" value="${jdbc.url}"/>  
 <property name="user" value="${jdbc.username}"/>  
 <property name="password" value="${jdbc.password}"/>  
 <property name="maxPoolSize" value="${c3p0.maxPoolSize}"/>  
 <property name="minPoolSize" value="${c3p0.minPoolSize}"/>  
 <property name="autoCommitOnClose" value="${c3p0.autoCommitOnClose}"/>  
 <property name="checkoutTimeout" value="${c3p0.checkoutTimeout}"/>  
 <property name="acquireRetryAttempts" value="${c3p0.acquireRetryAttempts}"/>  
 </bean>  
  
 <!-- 配置SqlSessionFactory对象 -->  
 <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">  
 <!-- 注入数据库连接池 -->  
 <property name="dataSource" ref="dataSource"/>  
 <!-- 扫描model包 使用别名 -->  
 <property name="typeAliasesPackage" value="com.kuanquan.demo.entity"/>  
 <!-- 扫描sql配置文件:mapper需要的xml文件 -->  
 <property name="mapperLocations" value="classpath:mapper/PersonMapper.xml"/>  
 </bean>  
  
 <!-- 配置扫描Dao接口包，动态实现Dao接口，注入到spring容器中 -->  
 <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">  
 <!-- 注入sqlSessionFactory -->  
 <property name="sqlSessionFactoryBeanName" value="sqlSessionFactory"/>  
 <!-- 给出需要扫描Dao接口包 -->  
 <property name="basePackage" value="com.kuanquan.demo.dao"/>  
 </bean>  
  
 <!-- 配置事务管理器 -->  
 <bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">  
 <!-- 注入数据库连接池 -->  
 <property name="dataSource" ref="dataSource"/>  
 </bean>  
  
 <!-- 配置基于注解的声明式事务 -->  
 <tx:annotation-driven transaction-manager="transactionManager"/>  
  
</beans>

Jdbc.properties

jdbc.driver=com.mysql.jdbc.Driver  
#数据库地址 Fly是数据库名称  
jdbc.url=jdbc:mysql://localhost:3306/admin?useUnicode=true&characterEncoding=utf8&serverTimezone=GMT  
#用户名  
jdbc.username=root  
#密码  
jdbc.password=root  
#最大连接数  
c3p0.maxPoolSize=30  
#最小连接数  
c3p0.minPoolSize=10  
#关闭连接后不自动commit  
c3p0.autoCommitOnClose=false  
#获取连接超时时间  
c3p0.checkoutTimeout=10000  
#当获取连接失败重试次数  
c3p0.acquireRetryAttempts=2

web.xml

<?xml version="1.0" encoding="UTF-8"?>  
<!--  
 Licensed to the Apache Software Foundation (ASF) under one or more  
 contributor license agreements. See the NOTICE file distributed with  
 this work for additional information regarding copyright ownership.  
 The ASF licenses this file to You under the Apache License, Version 2.0  
 (the "License"); you may not use this file except in compliance with  
 the License. You may obtain a copy of the License at  
  
 http://www.apache.org/licenses/LICENSE-2.0  
  
 Unless required by applicable law or agreed to in writing, software  
 distributed under the License is distributed on an "AS IS" BASIS,  
 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
 See the License for the specific language governing permissions and  
 limitations under the License.  
-->  
  
<!--  
 - This is the Cocoon web-app configurations file  
 -  
 - $Id$  
 -->  
<web-app version="2.4"  
 xmlns="http://java.sun.com/xml/ns/j2ee"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app\_2\_4.xsd">  
 <context-param>  
 <param-name>javax.faces.CONFIG\_FILES</param-name>  
 <param-value>spring-mvc.xml</param-value>  
 </context-param>  
  
 <!-- Servlet Filters ================================================ -->  
  
 <!--  
 - Declare a filter for multipart MIME handling  
 -->  
 <filter>  
 <filter-name>encodingFilter</filter-name>  
 <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>  
 <init-param>  
 <param-name>encoding</param-name>  
 <param-value>UTF-8</param-value>  
 </init-param>  
 </filter>  
 <filter-mapping>  
 <filter-name>encodingFilter</filter-name>  
 <url-pattern>/\*</url-pattern>  
 </filter-mapping>  
  
 <!-- 配置DispatcherServlet -->  
 <servlet>  
 <servlet-name>SpringMVC</servlet-name>  
 <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  
 <!-- 配置springMVC需要加载的配置文件-->  
 <init-param>  
 <param-name>contextConfigLocation</param-name>  
 <param-value>classpath:/spring-\*.xml</param-value>  
 </init-param>  
 <load-on-startup>1</load-on-startup>  
 <async-supported>true</async-supported>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>SpringMVC</servlet-name>  
 <!-- 匹配所有请求，此处也可以配置成 \*.do 形式 -->  
 <url-pattern>/</url-pattern>  
 </servlet-mapping>  
  
 <welcome-file-list>  
 <welcome-file>index.jsp</welcome-file>  
 </welcome-file-list>  
  
  
 <!--  
 - Servlet that dispatches requests to the Spring managed block servlets  
 -->  
  
 <servlet>  
 <servlet-name>Faces Servlet</servlet-name>  
 <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>  
 <load-on-startup>1</load-on-startup>  
 </servlet>  
  
 <!-- URL space mappings ============================================= -->  
  
 <!--  
 - Cocoon handles all the URL space assigned to the webapp using its sitemap.  
 - It is recommended to leave it unchanged. Under some circumstances though  
 - (like integration with proprietary webapps or servlets) you might have  
 - to change this parameter.  
 -->  
  
 <servlet-mapping>  
 <servlet-name>Faces Servlet</servlet-name>  
 <url-pattern>\*.xhtml</url-pattern>  
 </servlet-mapping>  
  
</web-app>

Index.jsp

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="UTF-8">  
 <title>test</title>  
</head>  
<script>  
 function selectUser() {  
 var xmlhttp = new ***XMLHttpRequest***();  
 xmlhttp.onreadystatechange = function () {  
 if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {  
 ***document***.getElementById("test").innerHTML = xmlhttp.responseText;  
 }  
 }  
 xmlhttp.open("get", "person/selectPerson", true);  
 xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");  
 xmlhttp.send();  
 }  
</script>  
<body>  
<p id="test">Hello World!</p>  
<button type="button" onclick="selectUser()">onclick test</button>  
</body>  
</html>

Pom.xml

<?xml version="1.0" encoding="UTF-8"?>  
<!--  
 Licensed to the Apache Software Foundation (ASF) under one  
 or more contributor license agreements. See the NOTICE file  
 distributed with this work for additional information  
 regarding copyright ownership. The ASF licenses this file  
 to you under the Apache License, Version 2.0 (the  
 "License"); you may not use this file except in compliance  
 with the License. You may obtain a copy of the License at  
  
 http://www.apache.org/licenses/LICENSE-2.0  
  
 Unless required by applicable law or agreed to in writing,  
 software distributed under the License is distributed on an  
 "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY  
 KIND, either express or implied. See the License for the  
 specific language governing permissions and limitations  
 under the License.  
-->  
<!-- $Id: pom.xml 642118 2008-03-28 08:04:16Z reinhard $ -->  
<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>  
 <packaging>war</packaging>  
  
 <name>untitled5</name>  
 <groupId>org.example</groupId>  
 <artifactId>untitled5</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.mortbay.jetty</groupId>  
 <artifactId>maven-jetty-plugin</artifactId>  
 <version>6.1.7</version>  
 <configuration>  
 <connectors>  
 <connector implementation="org.mortbay.jetty.nio.SelectChannelConnector">  
 <port>8888</port>  
 <maxIdleTime>30000</maxIdleTime>  
 </connector>  
 </connectors>  
 <webAppSourceDirectory>${project.build.directory}/${pom.artifactId}-${pom.version}</webAppSourceDirectory>  
 <contextPath>/</contextPath>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
 <properties>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 <maven.compiler.source>1.7</maven.compiler.source>  
 <maven.compiler.target>1.7</maven.compiler.target>  
  
 <!-- spring版本号 -->  
 <spring.version>4.3.8.RELEASE</spring.version>  
  
 <!-- mybatis版本号 -->  
 <mybatis.version>3.2.8</mybatis.version>  
  
 <!-- mysql驱动版本号 -->  
 <mysql-driver.version>8.0.11</mysql-driver.version>  
  
 </properties>  
  
 <dependencies>  
 <!-- 添加junit4依赖 -->  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>4.11</version>  
 <!-- 指定范围，在测试时才会加载 test是包名-->  
 <!--<scope>test</scope>-->  
 </dependency>  
  
 <!-- 添加spring核心依赖 start -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-core</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-web</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-oxm</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-tx</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-jdbc</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context-support</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aop</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-test</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <!-- 添加spring核心依赖 end -->  
  
 <!-- 添加mybatis/spring整合包依赖 -->  
 <dependency>  
 <groupId>org.mybatis</groupId>  
 <artifactId>mybatis-spring</artifactId>  
 <version>1.3.0</version>  
 </dependency>  
 <dependency>  
 <groupId>org.mybatis</groupId>  
 <artifactId>mybatis</artifactId>  
 <version>3.4.1</version>  
 </dependency>  
  
 <!-- 添加mysql驱动依赖 -->  
 <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>${mysql-driver.version}</version>  
 </dependency>  
 <!-- 添加数据库连接池依赖 -->  
 <dependency>  
 <groupId>commons-dbcp</groupId>  
 <artifactId>commons-dbcp</artifactId>  
 <version>1.2.2</version>  
 </dependency>  
  
 <!-- 添加数据库连接池依赖 c3p0 -->  
 <dependency>  
 <groupId>com.mchange</groupId>  
 <artifactId>c3p0</artifactId>  
 <version>0.9.5.2</version>  
 </dependency>  
  
 <!-- 添加fastjson -->  
 <dependency>  
 <groupId>com.alibaba</groupId>  
 <artifactId>fastjson</artifactId>  
 <version>1.1.41</version>  
 </dependency>  
  
 <!--Controller层 HttpServletRequest 和 HttpServletResponse 的依赖-->  
 <dependency>  
 <groupId>org.glassfish</groupId>  
 <artifactId>javax.faces</artifactId>  
 <version>2.2.7</version>  
 </dependency>  
  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>servlet-api</artifactId>  
 <version>2.3</version>  
 <scope>provided</scope>  
 </dependency>  
  
 <!--加入 jackson 依赖 start -->  
 <dependency>  
 <groupId>org.codehaus.jackson</groupId>  
 <artifactId>jackson-mapper-asl</artifactId>  
 <version>1.9.13</version>  
 </dependency>  
 <!--https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-core-->  
 <dependency>  
 <groupId>com.fasterxml.jackson.core</groupId>  
 <artifactId>jackson-core</artifactId>  
 <version>2.8.0</version>  
 </dependency>  
 <!--https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind-->  
 <dependency>  
 <groupId>com.fasterxml.jackson.core</groupId>  
 <artifactId>jackson-databind</artifactId>  
 <version>2.8.0</version>  
 </dependency>  
 <!--加入 jackson 依赖 end -->  
  
  
 </dependencies>  
  
  
</project>

五．数据库构建

mysql> CREATE DATABASE IF NOT EXISTS admin DEFAULT CHARSET utf8 COLLATE utf8\_general\_ci;

mysql> use admin

mysql> create table user(

-> user\_id int(11) primary key auto\_increment,

-> name varchar(64) not null,

-> age int(11) not null,

-> sex int(11) not null,

-> create\_time datetime not null default '2014-09-01 00:00:00')

-> ENGINE=InnoDB DEFAULT CHARSET=utf8;

mysql> insert into user values(1,'张三',20,0,'2014-04-27 18:12:59');

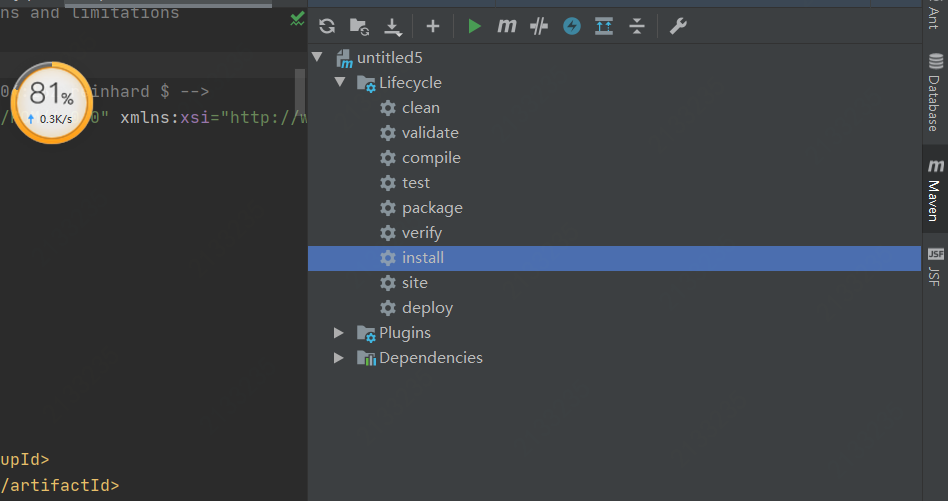
Query OK, 1 row affected (0.31 sec)

mysql> insert into user values(2,'李四',20,0,'2015-04-27 18:12:59');

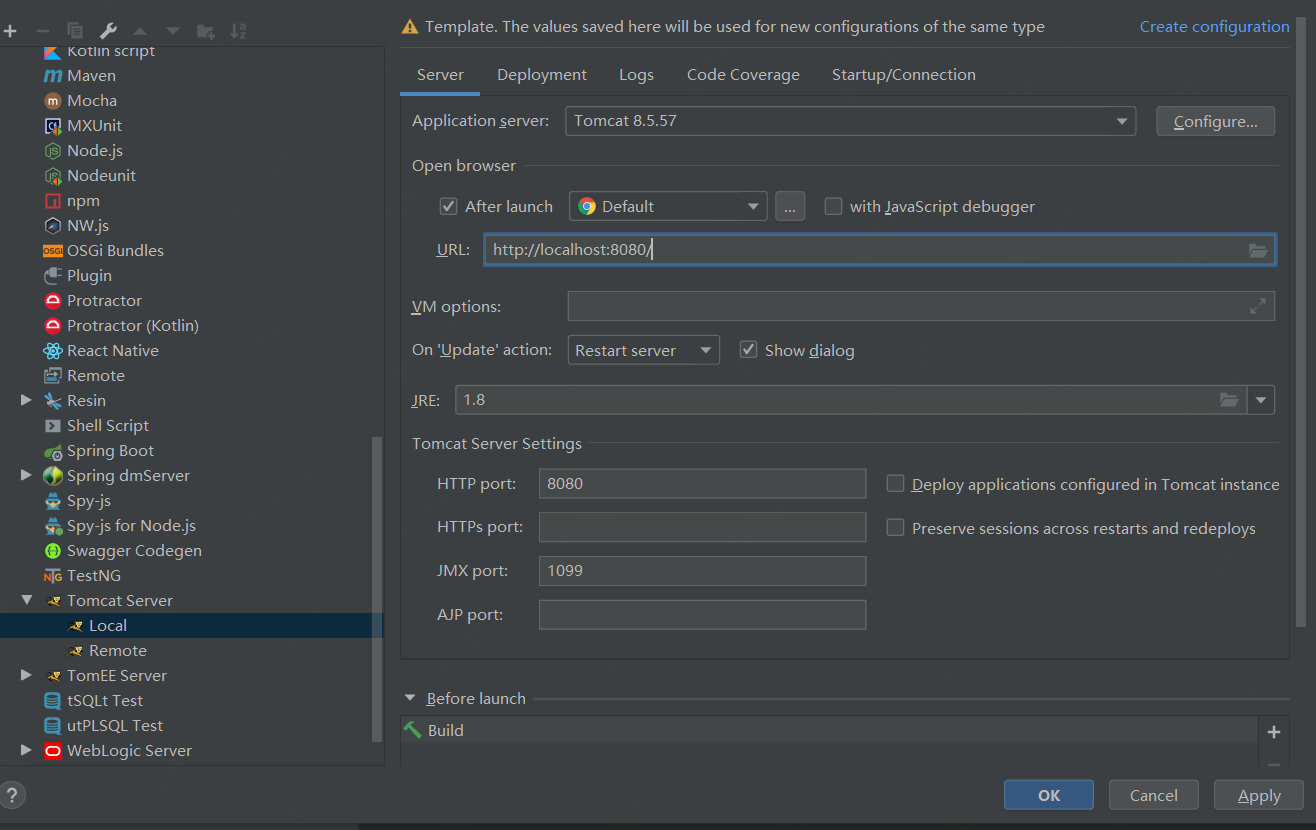
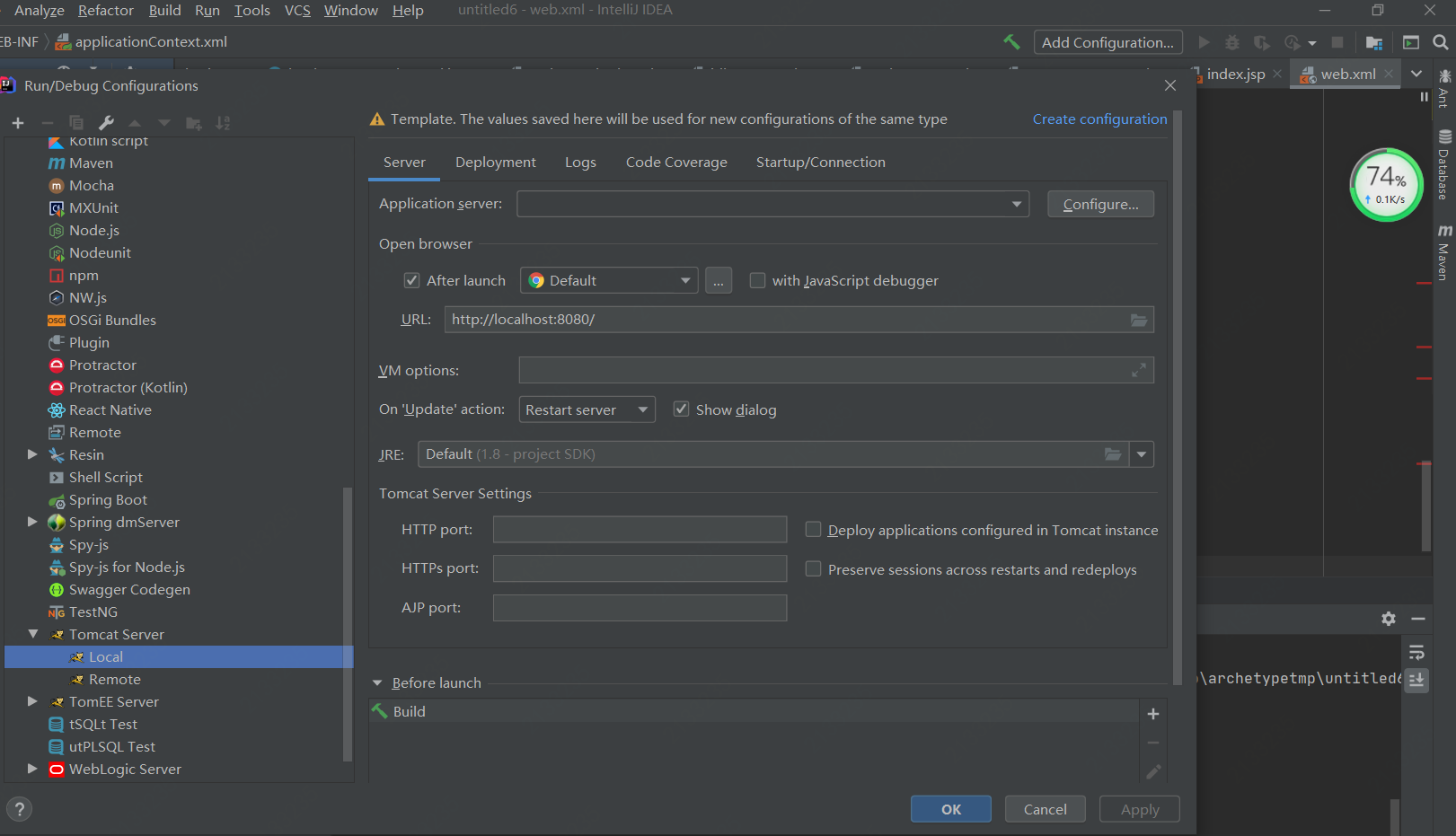
Query OK, 1 row affected (0.06 sec)

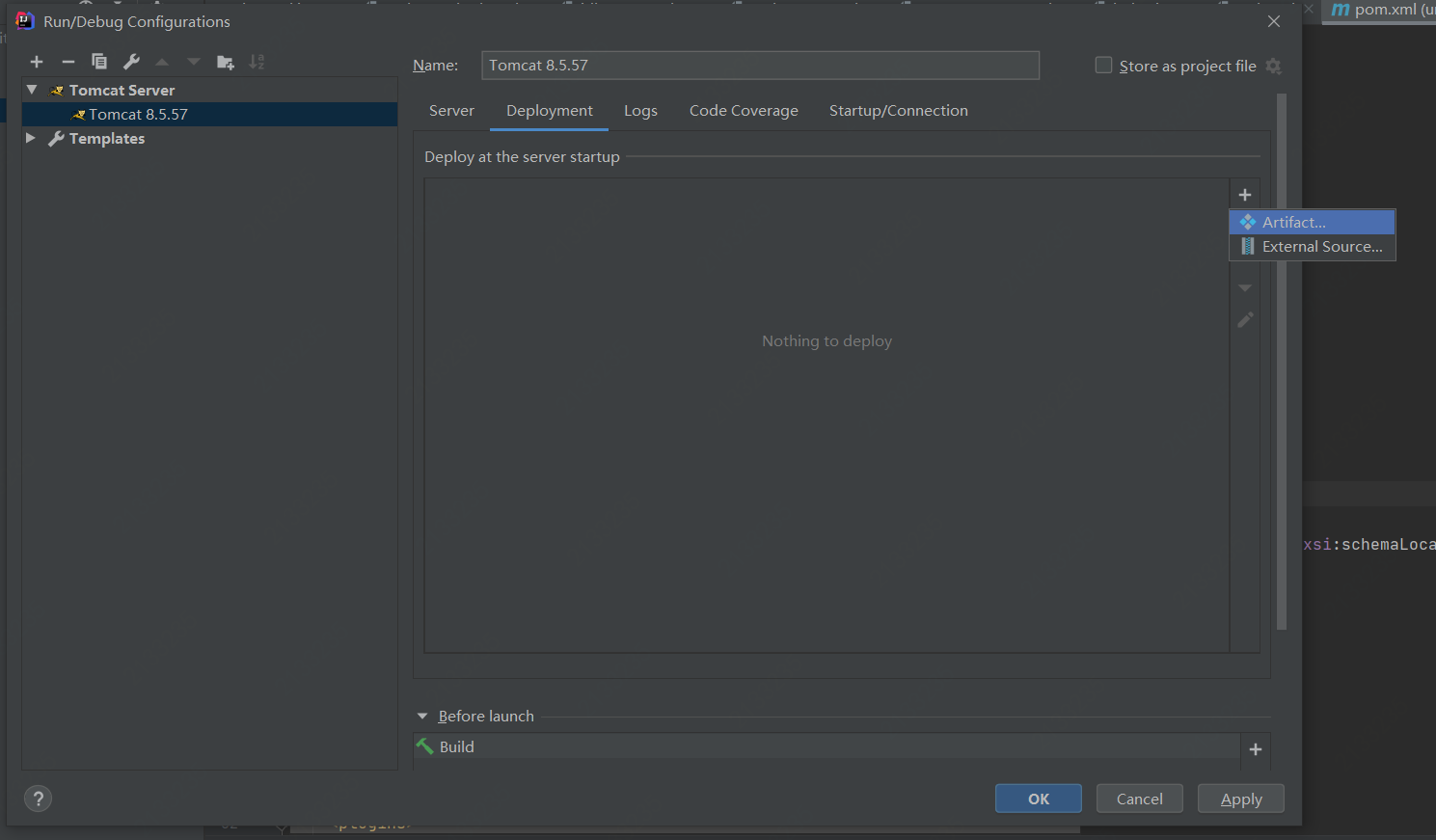
mysql> insert into user values(3,'王五',20,0,'2016-04-27 18:12:59');

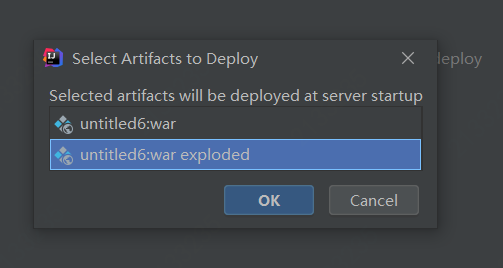
六．打成war



七．Tomcat：运行配置







参考网址；<https://blog.csdn.net/pengyu1801/article/details/81939843>