# 基本概念

## Spring

Spring是一个轻量级的控制反转（IoC）和面向切面（AOP）的容器框架。

## SpringMVC

Spring MVC属于SpringFrameWork的后续产品，已经融合在Spring Web Flow里面

## MyBatis

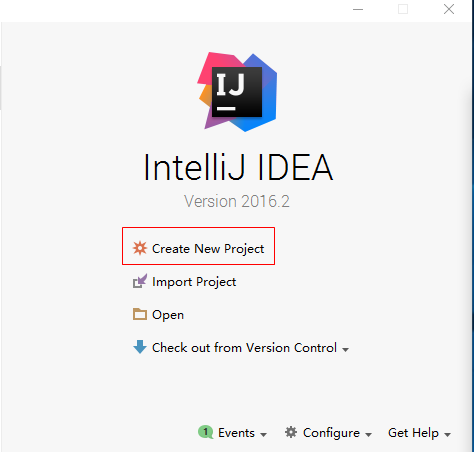
MyBatis 本是apache的一个开源项目iBatis, 2010年这个项目由apache software foundation 迁移到了google code，并且改名为MyBatis 。

MyBatis是一个基于Java的持久层框架。消除了几乎所有的JDBC代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的 XML或注解用于配置和原始映射，将接口和 Java 的POJOs（Plain Old Java Objects，普通的 Java对象）映射成数据库中的记录。

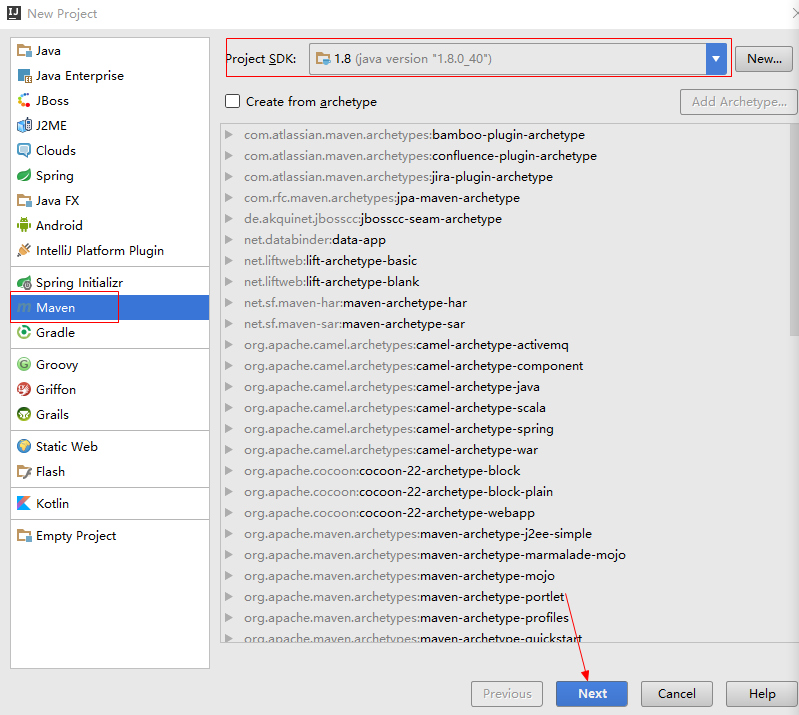
# 创建Maven Web项目

使用IDEA开发工具。

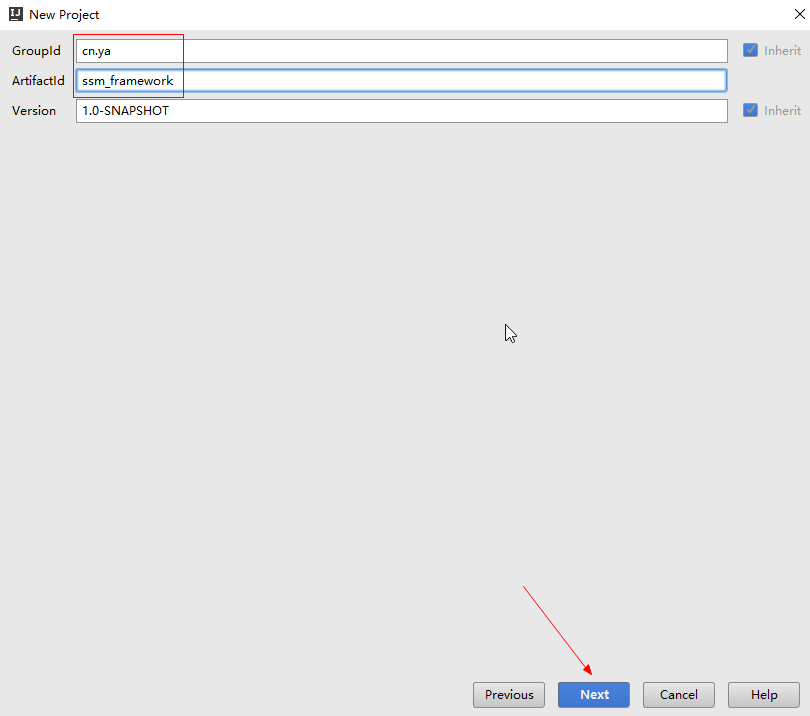
1. 打开IDEA，点击“新建项目”



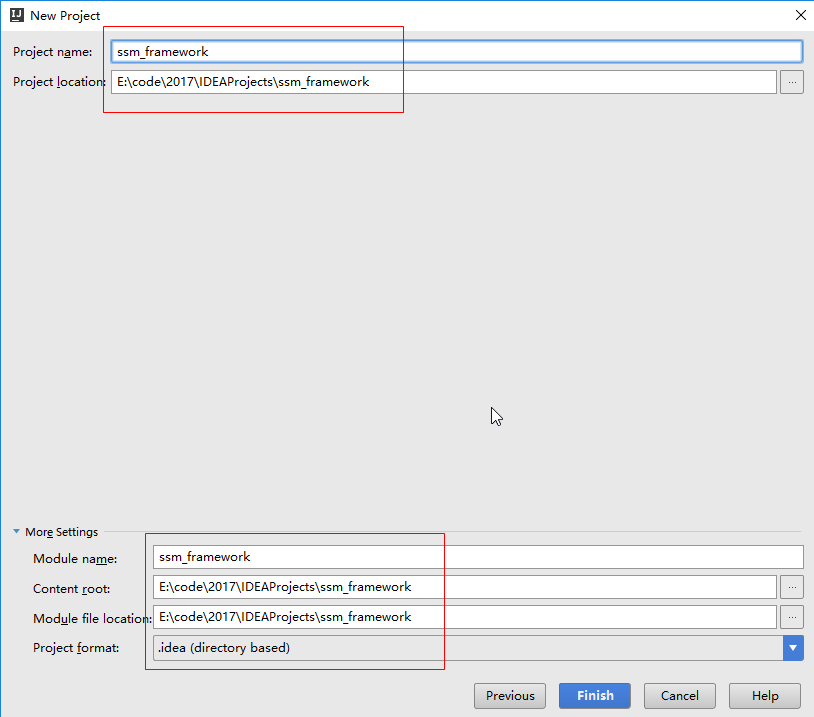
1. 选择“Maven”，注意Project SDK要选择你安装的JDK路径，next



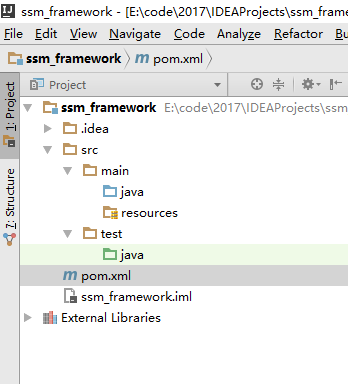
1. 输入项目名，next



1. 然后再次输入上面ArtifactId的名称，作为Project name的名称。输入Project name后，其他的会自动补全，无须另做操作。finish，即完成项目创建。



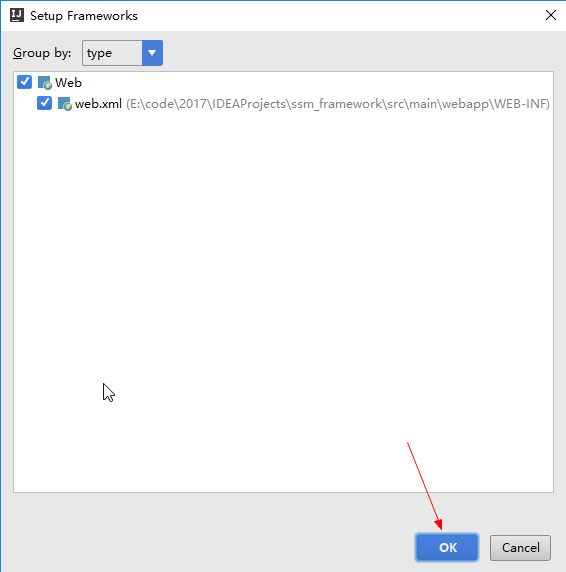
1. 新建后的项目结构如下



1. 将项目改成web项目。三步
   1. 在main目录下，添加webapp目录
   2. 在webapp目录下添加WEB-INF目录
   3. 在WEB-INF目录下添加web.xml文件。
   4. 此时IDEA给出一个提示。



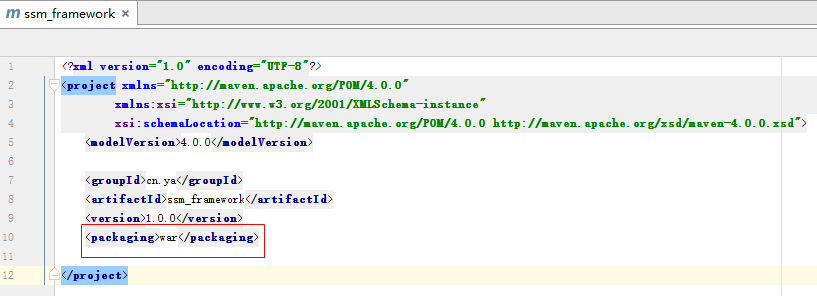
表示IDEA已识别出目前所有了web框架，需要进行一些配置才能使用，点击“Configure”按钮，会看到一个确认框，点击“OK”即可将当前项目变成web项目。



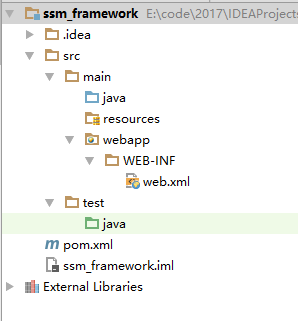
1. 在web.xml中添加如下代码

|  |
| --- |
| <?xml version="1.0" encoding="utf-8" ?>  <web-app  xmlns="http://java.sun.com/xml/ns/javaee"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  version="3.0">    </web-app> |

1. 添加Java web的maven依赖，由于web项目是需要打war包的，因此必须在pom.xml文件里设置packaging为war。

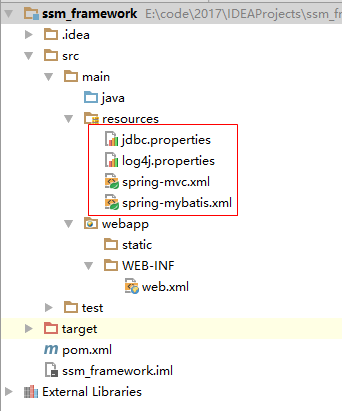


1. 至此，maven web项目即创建成功。maven web项目的目录结构如下



# SSM整合

在main/resources下新建4个配置文件。完整目录结构如下。



## Maven引入需要的JAR包。在pom.xml中引入jar包

*<?***xml version="1.0" encoding="UTF-8"***?>*<**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**>cn.ya</**groupId**>  
 <**artifactId**>ssm\_framework</**artifactId**>  
 <**version**>1.0.0</**version**>  
 <**packaging**>war</**packaging**>  
  
 <**properties**>  
 *<!-- 编码方式-->* <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  
 *<!-- Spring版本号 -->* <**spring.version**>4.1.6.RELEASE</**spring.version**>  
 *<!-- mybatis版本号 -->* <**mybatis.version**>3.3.0</**mybatis.version**>  
 *<!-- log4j日志文件管理包版本 -->* <**log4j.version**>2.5</**log4j.version**>  
 </**properties**>  
  
 <**dependencies**>  
 *<!-- junit单元测试-->* <**dependency**>  
 <**groupId**>junit</**groupId**>  
 <**artifactId**>junit</**artifactId**>  
 <**version**>4.12</**version**>  
 *<!-- 表示开发的时候引入，发布的时候不会加载此包-->* <**scope**>test</**scope**>  
 </**dependency**>  
  
 *<!-- Spring核心包-->* <**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-aop</**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-test</**artifactId**>  
 <**version**>${spring.version}</**version**>  
 </**dependency**>  
  
 *<!-- mybatis核心包-->* <**dependency**>  
 <**groupId**>org.mybatis</**groupId**>  
 <**artifactId**>mybatis</**artifactId**>  
 <**version**>${mybatis.version}</**version**>  
 </**dependency**>  
  
 *<!-- mybatis-Spring包-->* <**dependency**>  
 <**groupId**>org.mybatis</**groupId**>  
 <**artifactId**>mybatis-spring</**artifactId**>  
 <**version**>1.2.2</**version**>  
 </**dependency**>  
  
 *<!-- 导入Java ee包-->* <**dependency**>  
 <**groupId**>javax</**groupId**>  
 <**artifactId**>javaee-api</**artifactId**>  
 <**version**>7.0</**version**>  
 </**dependency**>  
  
 *<!-- 导入MySQL数据库连接包-->* <**dependency**>  
 <**groupId**>mysql</**groupId**>  
 <**artifactId**>mysql-connector-java</**artifactId**>  
 <**version**>5.1.38</**version**>  
 </**dependency**>  
  
 *<!-- 导入c3p0数据源的jar包，用来配置数据库-->* <**dependency**>  
 <**groupId**>c3p0</**groupId**>  
 <**artifactId**>c3p0</**artifactId**>  
 <**version**>0.9.1.2</**version**>  
 </**dependency**>  
  
 *<!-- JSTL标签类-->* <**dependency**>  
 <**groupId**>jstl</**groupId**>  
 <**artifactId**>jstl</**artifactId**>  
 <**version**>1.2</**version**>  
 </**dependency**>  
  
 *<!-- 日志文件管理包-->* <**dependency**>  
 <**groupId**>org.apache.logging.log4j</**groupId**>  
 <**artifactId**>log4j-core</**artifactId**>  
 <**version**>${log4j.version}</**version**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>org.apache.logging.log4j</**groupId**>  
 <**artifactId**>log4j-api</**artifactId**>  
 <**version**>${log4j.version}</**version**>  
 </**dependency**>  
  
 *<!-- 格式化对象， 方便输出日志-->* <**dependency**>  
 <**groupId**>com.alibaba</**groupId**>  
 <**artifactId**>fastjson</**artifactId**>  
 <**version**>1.1.41</**version**>  
 </**dependency**>  
  
 *<!--json-->* <**dependency**>  
 <**groupId**>com.fasterxml.jackson.core</**groupId**>  
 <**artifactId**>jackson-databind</**artifactId**>  
 <**version**>2.4.2</**version**>  
 </**dependency**>  
  
 </**dependencies**>  
  
 <**build**>  
 <**plugins**>  
 *<!-- Compile，统一源代码与编译输出的JDK版本-->* <**plugin**>  
 <**groupId**>org.apache.maven.plugins</**groupId**>  
 <**artifactId**>maven-compiler-plugin</**artifactId**>  
 <**configuration**>  
 <**source**>1.8</**source**>  
 <**target**>1.8</**target**>  
 </**configuration**>  
 </**plugin**>  
 </**plugins**>  
 </**build**>  
</**project**>

## 数据库信息配置文件jdbc.properties

将连接数据库需要的参数放在配置文件中，方便切换数据库时修改参数。

**jdbc.driver**=**com.mysql.jdbc.Driver  
jdbc.url**=**jdbc:mysql://localhost:3306/数据库名  
jdbc.username**=**root  
jdbc.password**=**123***#定义初始连接数***jdbc.initialPoolSize**=**0***#定义最大连接数***jdbc.maxPoolSize**=**20***#定义最大空闲***jdbc.maxIdleTime**=**20**

## log4j日志配置文件

*# Global logging configuration  
#在开发环境下日志级别要设置成DEBUG，生产环境设置成INFO或ERROR***log4j.rootLogger**=**DEBUG, stdout***# Console output...***log4j.appender.stdout**=**org.apache.log4j.ConsoleAppender  
log4j.appender.stdout.layout**=**org.apache.log4j.PatternLayout  
log4j.appender.stdout.layout.ConversionPattern**=**%5p [%t] - %m%n**

## spring和mybatis整合的配置文件spring-mybatis.xml

1. 需要引入数据库信息配置文件
2. 配置数据源
3. 配置sqlSessionFactory(加入数据源)
4. 扫描mapper文件，里面包括mapper基础包和sqlSessionFactory

*<?***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"  
 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-4.1.xsd"**>  
  
 *<!-- mybatis和spring整合的配置文件  
 加载数据库信息配置文件  
 配置数据源  
 配置sqlSessionFactory  
 扫描mapper文件 -->  
  
 <!-- 加载配置文件 -->* <**context:property-placeholder location="classpath:jdbc.properties"**/>  
  
 *<!-- 配置数据源 c3p0 -->* <**bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"**>  
 <**property name="driverClass" value="${jdbc.driver}"**></**property**>  
 <**property name="jdbcUrl" value="${jdbc.url}"**></**property**>  
 <**property name="user" value="${jdbc.username}"**></**property**>  
 <**property name="password" value="${jdbc.password}"**></**property**>  
 *<!-- 初始化连接大小 -->* <**property name="initialPoolSize" value="${jdbc.initialPoolSize}"**></**property**>  
 *<!-- 连接池最大数量 -->* <**property name="maxPoolSize" value="${jdbc.maxPoolSize}"**></**property**>  
 *<!-- 连接池最大空闲 -->* <**property name="maxIdleTime" value="${jdbc.maxIdleTime}"**></**property**>  
 </**bean**>  
  
 *<!-- 配置sqlSessionFactory ，Spring和mybatis整合后，不需要mybatis的配置文件-->* <**bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean"**>  
 <**property name="dataSource" ref="dataSource"**></**property**>  
 </**bean**>  
  
 *<!-- mapper批量扫描 -->  
 <!-- 例：用户模块的mapper -->* <**bean class="org.mybatis.spring.mapper.MapperScannerConfigurer"**>  
 <**property name="basePackage" value="mapper包路径"**></**property**>  
 <**property name="sqlSessionFactoryBeanName" value="sqlSessionFactory"**></**property**>  
 </**bean**>

<**context:component-scan base-package=" mapper包路径"**/>  
  
</**beans**>

## springmvc配置文件spring-mvc.xml

1. 配置适配器和映射器，一键式
2. 配置视图解析器
3. 设置静态资源访问路径
4. 扫描controller包

*<?***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-4.1.xsd  
 http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.1.xsd"**>  
  
 *<!-- 配置适配器和映射器 -->* <**mvc:annotation-driven**></**mvc:annotation-driven**>  
  
 *<!-- 配置视图解析器，可加前后缀 -->* <**bean class="org.springframework.web.servlet.view.InternalResourceViewResolver"**>  
 <**property name="prefix" value="/WEB-INF/"**></**property**>  
 <**property name="suffix" value=".jsp"**></**property**>  
  
 </**bean**>  
  
 *<!-- 静态资源访问 ，webapp目录下建static文件夹，用来存放所有的静态资源-->* <**mvc:resources location="/static/" mapping="/static/\*\*"**></**mvc:resources**>  
  
 *<!-- 扫描controller -->* <**context:component-scan base-package="controller包路径"**></**context:component-scan**>  
  
</**beans**>

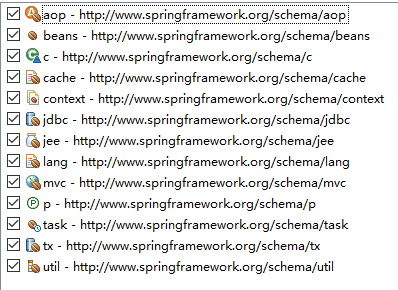
## web.xml配置

1. 配置前端控制器，注意路径是适配器和映射器的配置文件路径spring-mvc.xml
2. 对于spring-mybatis.xml，称之为spring容器，也要进行配置，这种配置需要添加监听器
3. post乱码处理配置

*<?***xml version="1.0" encoding="utf-8"** *?>*<**web-app  
 xmlns="http://java.sun.com/xml/ns/javaee"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  
 http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  
 version="3.0"**>  
  
 *<!-- 加载spring容器，即Spring和mybatis的配置文件 -->* <**context-param**>  
 <**param-name**>contextConfigLocation</**param-name**>  
 <**param-value**>classpath:spring-mybatis.xml</**param-value**>  
 </**context-param**>  
  
 *<!-- post乱码 filter -->* <**filter**>  
 <**filter-name**>CharacterEncodingFilter</**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**>CharacterEncodingFilter</**filter-name**>  
 <**url-pattern**>/\*</**url-pattern**>  
 </**filter-mapping**>  
  
 *<!-- Spring监听器 -->* <**listener**>  
 <**listener-class**>org.springframework.web.context.ContextLoaderListener</**listener-class**>  
 </**listener**>  
  
 *<!-- 配置前端控制器 -->* <**servlet**>  
 <**servlet-name**>springmvc</**servlet-name**>  
 <**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>  
 <**init-param**>  
 <**param-name**>contextConfigLocation</**param-name**>  
 <**param-value**>classpath:spring-mvc.xml</**param-value**>  
 </**init-param**>  
 </**servlet**>  
 <**servlet-mapping**>  
 <**servlet-name**>springmvc</**servlet-name**>  
 <**url-pattern**>/</**url-pattern**>  
 </**servlet-mapping**>  
  
</**web-app**>

## Spring配置文件，命名空间详细

1. 因为IDEA无法想eclipse那样可以直接导入Spring配置文件的前缀命名空间，所以这里利用eclipse将各命名空间的声明先记录下来，方便以后使用到时，直接复制粘贴
2. Spring的全部命名空间



1. 对应的XML源文件

<?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:aop=*"http://www.springframework.org/schema/aop"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:jdbc=*"http://www.springframework.org/schema/jdbc"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"*

xmlns:p=*"http://www.springframework.org/schema/p"*

xmlns:tx=*"http://www.springframework.org/schema/tx"*

xmlns:util=*"http://www.springframework.org/schema/util"*

xmlns:c=*"http://www.springframework.org/schema/c"*

xmlns:cache=*"http://www.springframework.org/schema/cache"*

xmlns:jee=*"http://www.springframework.org/schema/jee"*

xmlns:lang=*"http://www.springframework.org/schema/lang"*

xmlns:task=*"http://www.springframework.org/schema/task"*

xsi:schemaLocation=*"http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.1.xsd*

*http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-4.1.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.1.xsd*

*http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.1.xsd*

*http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-4.1.xsd*

*http://www.springframework.org/schema/jdbc http://www.springframework.org/schema/jdbc/spring-jdbc-4.1.xsd*

*http://www.springframework.org/schema/cache http://www.springframework.org/schema/cache/spring-cache-4.1.xsd*

*http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-4.1.xsd*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/lang http://www.springframework.org/schema/lang/spring-lang-4.1.xsd*

*http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.1.xsd"*>

</beans>

# 简单测试

SSM整合后，写个小demo，做简单测试，看是否整合成功

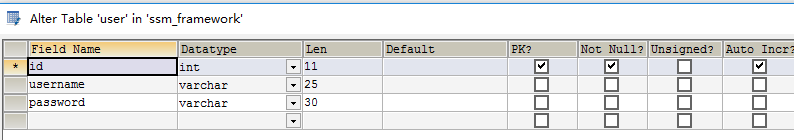
程序开发

mapper：用注解写sql语句

service：里面自动加载mapper

controller：类要加入@controller注解，里面自动加载service，controller里的方法相当于servlet的方法，处理请求和响应，方法头要加入@RequestMapping注解，将请求URL跟实际路径进行映射

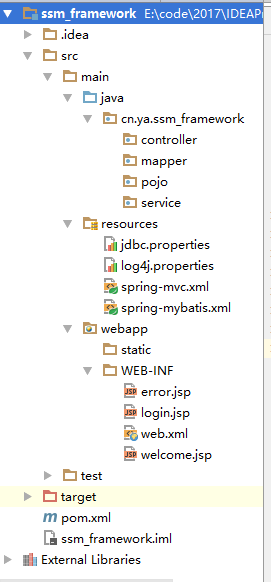
1. 需求：登录验证
2. 数据库设计
   1. 库名ssm\_framework
   2. 表名及表结构user



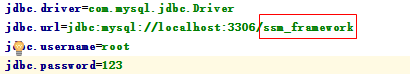
* 1. 初始数据录入



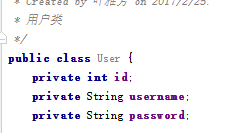
1. 项目结构



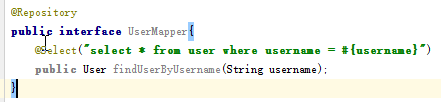
1. 使用ssm\_framework数据库后，需要在jdbc.properties文件中修改数据库名



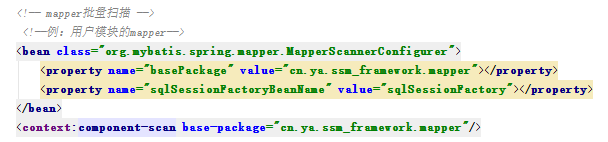
1. 建立pojo



1. 从mapper开始开发



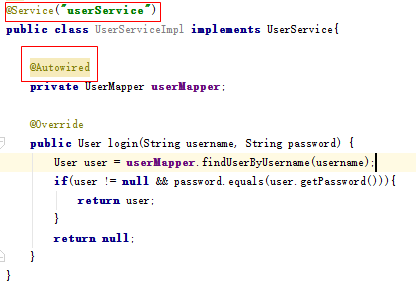
* 1. 在spring-mybatis.xml中扫描mapper



1. 开发service
   1. 先写service接口



* 1. 再写service实现类



* 1. 在spring-mybatis.xml中扫描service



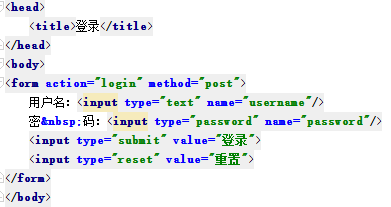
1. 开发controller
   1. controller中有两个方法，一个是项目入口，另一个是登录验证路径
   2. 项目入口
   3. 登录逻辑处理



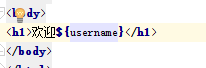
在spring-mvc.xml中扫描control



1. 页面开发
   1. 登录页



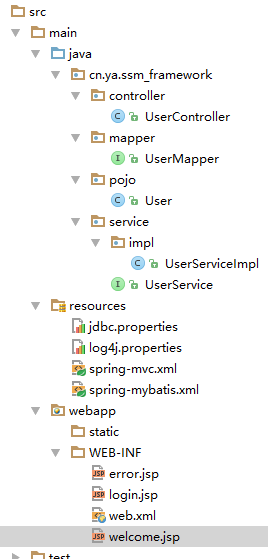
* 1. 欢迎页



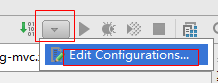
* 1. 错误页



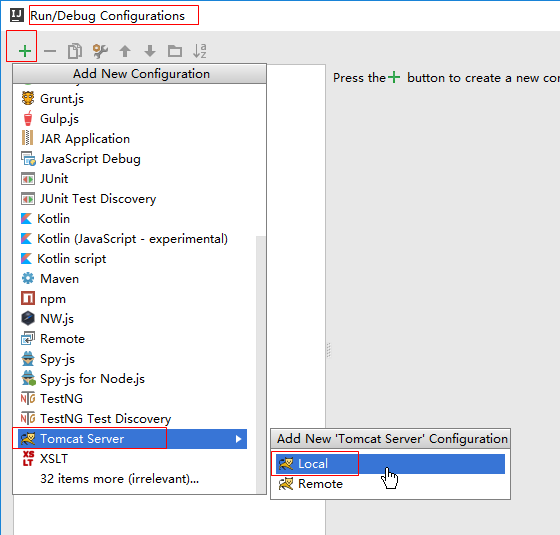
1. 项目完成，最后结构如下



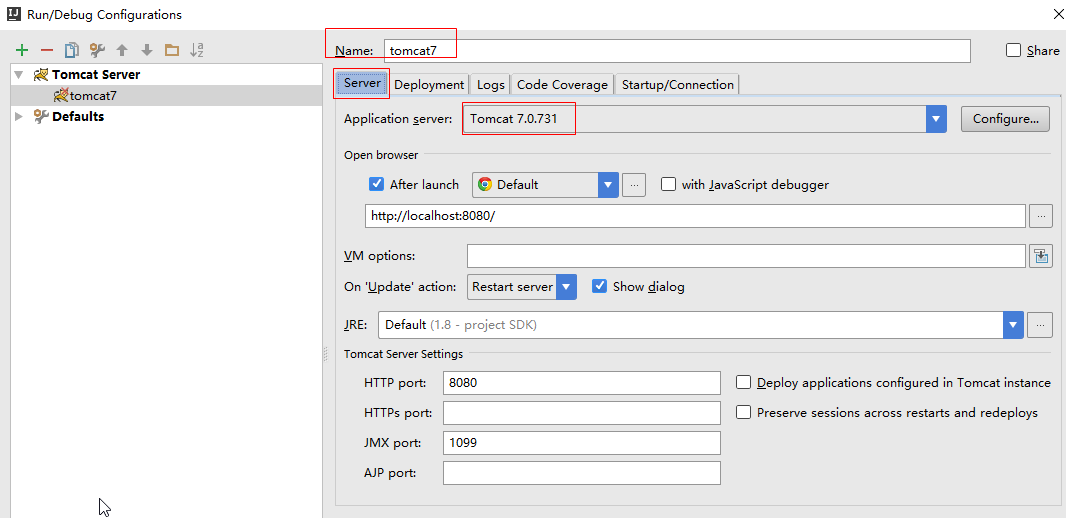
1. 项目部署—配置tomcat
   1. 在IDEA右上角，点击下三角符



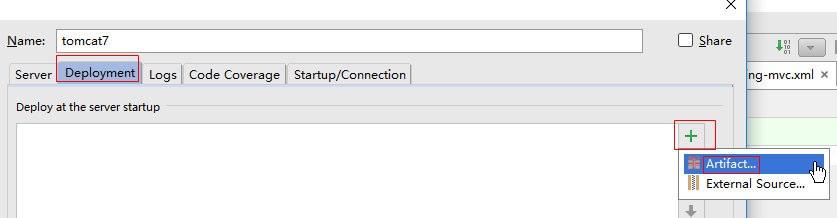
* 1. 在弹出的Run/Debug Configurations中选择左上角的“+”，在弹出的下拉列表中选择Tomcat Service->Local



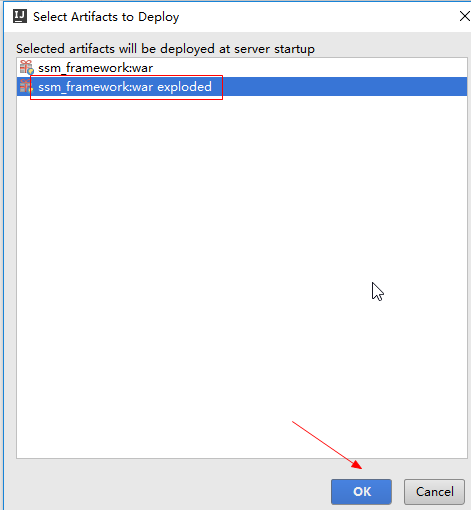
* 1. 给tomcat服务器起个名字，然后在Server选项卡的Application server中选择tomcat的安装路径



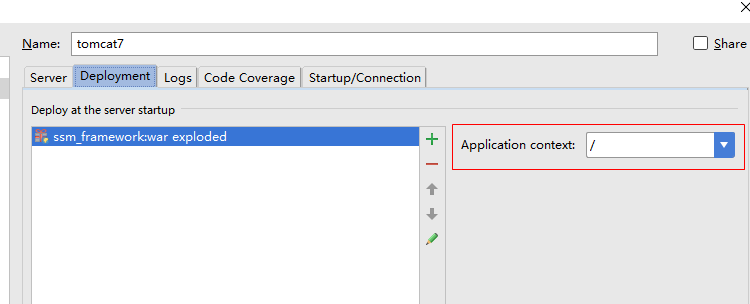
* 1. 打开Deployment选项卡，点击右边的“+”，在弹出的选项中选择“Artifact…”



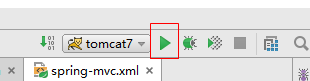
* 1. 选择 war exploded，OK



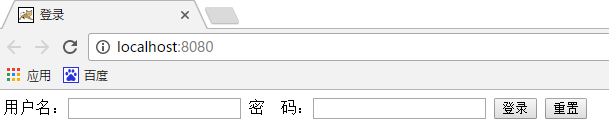
* 1. 在右边的Application context中，可以修改项目访问路径。配置好后点击OK，即完成tomcat的配置



* 1. 启动tomcat



* 1. 弹出如下页面，SSM整合成功



* 1. 可对这个简单的项目做测试，测试成功！

# 将此框架放在GitHub上，以后可直接拿来部署开发

<https://github.com/ya24>