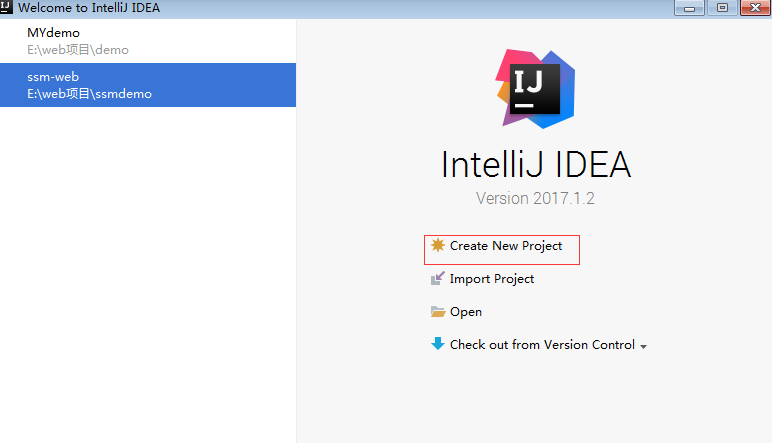
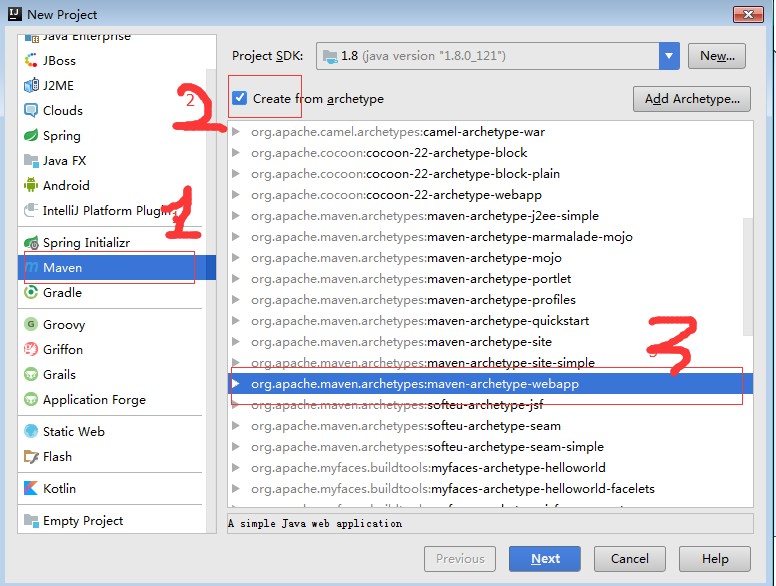
# Idea和Maven 搭建springmvc+myBatis

### idea下创建web项目

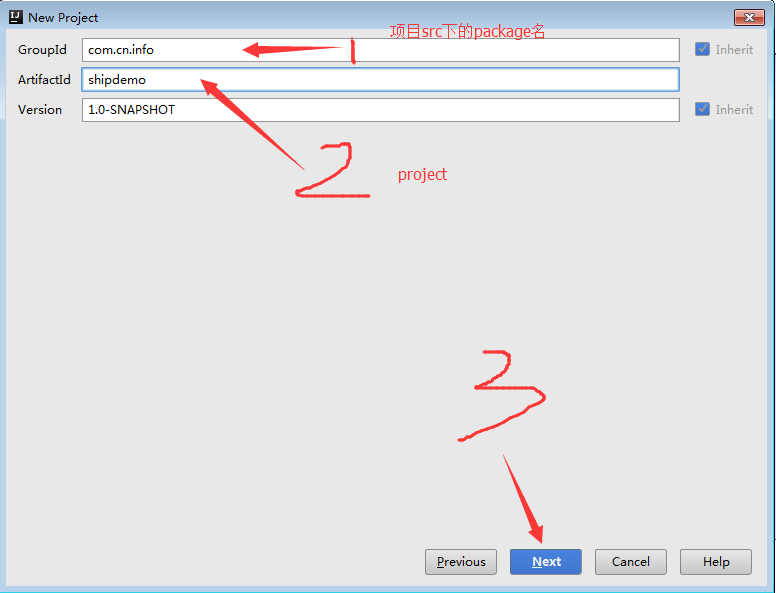
1.1打开idea新建项目



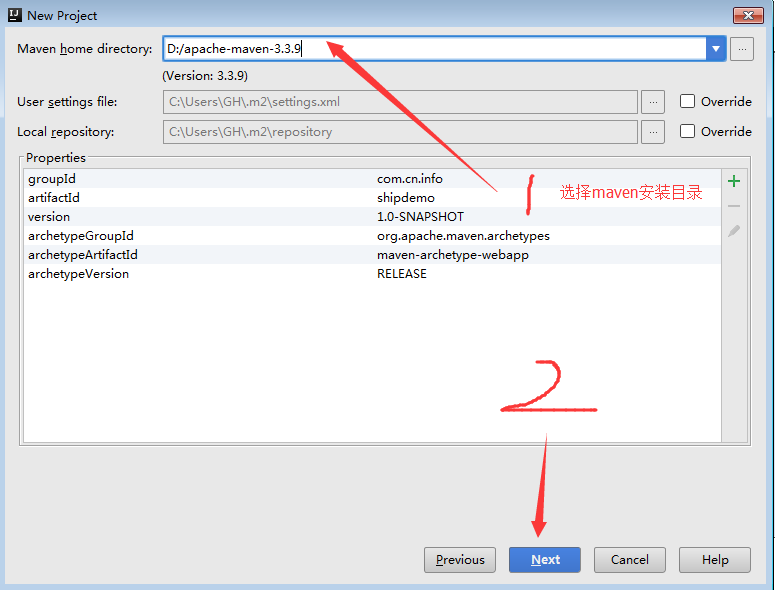
1.2 选择maven，勾上create from archetype，第三步选择webapp，点击next。



1.3 填写src下的package名称和project，点击next。



1.4 选择maven安装目录，点击next。



1.5 选择finish。

### 配置jar依赖

配置pom.xml

需要引入的jar有：

|  |
| --- |
| <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-core</**artifactId**>  <**version**>${spring.version}</**version**> </**dependency**>  <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-beans</**artifactId**>  <**version**>${spring.version}</**version**> </**dependency**>  <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-context</**artifactId**>  <**version**>${spring.version}</**version**> </**dependency**>  <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-tx</**artifactId**>  <**version**>${spring.version}</**version**> </**dependency**>  <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-web</**artifactId**>  <**version**>${spring.version}</**version**> </**dependency**>  <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-webmvc</**artifactId**>  <**version**>${spring.version}</**version**> </**dependency**>  <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-jdbc</**artifactId**>  <**version**>${spring.version}</**version**> </**dependency**>  <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-test</**artifactId**>  <**version**>${spring.version}</**version**>  <**scope**>test</**scope**> </**dependency**>  *<!-- mybatis 包 -->* <**dependency**>  <**groupId**>org.mybatis</**groupId**>  <**artifactId**>mybatis</**artifactId**>  <**version**>3.2.8</**version**> </**dependency**>  *<!--mybatis spring 插件 -->* <**dependency**>  <**groupId**>org.mybatis</**groupId**>  <**artifactId**>mybatis-spring</**artifactId**>  <**version**>1.2.2</**version**> </**dependency**>  *<!-- mysql连接 -->* <**dependency**>  <**groupId**>mysql</**groupId**>  <**artifactId**>mysql-connector-java</**artifactId**>  <**version**>5.1.34</**version**> </**dependency**>  *<!-- 数据源 -->* <**dependency**>  <**groupId**>com.alibaba</**groupId**>  <**artifactId**>druid</**artifactId**>  <**version**>1.0.12</**version**> </**dependency**>  <**dependency**>  <**groupId**>org.aspectj</**groupId**>  <**artifactId**>aspectjweaver</**artifactId**>  <**version**>1.8.4</**version**> </**dependency**> |

### 3. 配置Springmvc + mybatis

配置web.xml

3.1 配置核心调度器和 应用的登录界面

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"   xmlns="http://xmlns.jcp.org/xml/ns/javaee"   xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee   http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"   id="WebApp\_ID" version="3.1"**>    *<!-- 配置spring核心监听器，默认会以 /WEB-INF/applicationContext.xml作为配置文件 -->* <**listener**>  <**listener-class**>org.springframework.web.context.ContextLoaderListener</**listener-class**>  </**listener**>  *<!-- contextConfigLocation参数用来指定Spring的配置文件 -->* <**context-param**>  <**param-name**>contextConfigLocation</**param-name**>  <**param-value**>/WEB-INF/applicationContext\*.xml</**param-value**>  </**context-param**>    *<!-- 定义Spring MVC的前端控制器 -->* <**servlet**>  <**servlet-name**>springmvc</**servlet-name**>  <**servlet-class**>  org.springframework.web.servlet.DispatcherServlet  </**servlet-class**>  <**init-param**>  <**param-name**>contextConfigLocation</**param-name**>  <**param-value**>/WEB-INF/springmvc-config.xml</**param-value**>  </**init-param**>  <**load-on-startup**>1</**load-on-startup**>  </**servlet**>    *<!-- 让Spring MVC的前端控制器拦截所有请求 -->* <**servlet-mapping**>  <**servlet-name**>springmvc</**servlet-name**>  <**url-pattern**>\*.action</**url-pattern**>  </**servlet-mapping**>    *<!-- 编码过滤器 -->* <**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**>  <**welcome-file-list**>  <**welcome-file**>login.html</**welcome-file**>  </**welcome-file-list**>   </**web-app**> |

3.2 配置springmvc-config.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:mvc="http://www.springframework.org/schema/mvc"  xmlns:context="http://www.springframework.org/schema/context"  xsi:schemaLocation="  http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans-4.2.xsd  http://www.springframework.org/schema/mvc  http://www.springframework.org/schema/mvc/spring-mvc-4.2.xsd   http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context-4.2.xsd"**>    *<!-- 自动扫描该包，SpringMVC会将包下用了@controller注解的类注册为Spring的controller -->* <**context:component-scan base-package="com.cn.info.controller"**/>  *<!-- 设置默认配置方案 -->* <**mvc:annotation-driven**/>  *<!-- 使用默认的Servlet来响应静态文件 -->* <**mvc:default-servlet-handler**/>  *<!-- 视图解析器 -->* <**bean id="viewResolver"  class="org.springframework.web.servlet.view.InternalResourceViewResolver"**>  </**bean**>   </**beans**> |

3.3 配置db.properties 数据源文件

|  |
| --- |
| **dataSource.driverClass**=**com.mysql.jdbc.Driver dataSource.jdbcUrl**=**jdbc:mysql://127.0.0.1:3306/ssm dataSource.user**=**root dataSource.password**=**123456 dataSource.maxPoolSize**=**20 dataSource.maxIdleTime** = **1000 dataSource.minPoolSize**=**6 dataSource.initialPoolSize**=**5** |

3.4 配置ApplicationContext.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"   xmlns:mybatis="http://mybatis.org/schema/mybatis-spring"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:p="http://www.springframework.org/schema/p"  xmlns:context="http://www.springframework.org/schema/context"  xmlns:mvc="http://www.springframework.org/schema/mvc"  xmlns:tx="http://www.springframework.org/schema/tx"  xsi:schemaLocation="http://www.springframework.org/schema/beans   http://www.springframework.org/schema/beans/spring-beans-4.2.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context-4.2.xsd  http://www.springframework.org/schema/mvc  http://www.springframework.org/schema/mvc/spring-mvc-4.2.xsd  http://www.springframework.org/schema/tx  http://www.springframework.org/schema/tx/spring-tx-4.1.xsd  http://mybatis.org/schema/mybatis-spring   http://mybatis.org/schema/mybatis-spring.xsd "**>    *<!-- mybatis:scan会将mapper包里的所有接口当作mapper配置，之后可以自动引入mapper类-->* <**mybatis:scan base-package="com/cn.info.dao"**/>    *<!-- 扫描com.cn.info下面的java文件，有Spring的相关注解的类，则把这些类注册为Spring的bean -->* <**context:component-scan base-package="com.cn.info"**/>    *<!-- 使用PropertyOverrideConfigurer后处理器加载数据源参数 -->* <**context:property-override location="classpath:db.properties"**/>   *<!-- 配置c3p0数据源 -->* <**bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"**/>    *<!-- 配置SqlSessionFactory，org.mybatis.spring.SqlSessionFactoryBean是Mybatis社区开发用于整合Spring的bean -->* <**bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean"  p:dataSource-ref="dataSource"**/>    *<!-- JDBC事务管理器 -->* <**bean id="transactionManager"   class="org.springframework.jdbc.datasource.DataSourceTransactionManager"  p:dataSource-ref="dataSource"**/>    *<!-- 启用支持annotation注解方式事务管理 -->* <**tx:annotation-driven transaction-manager="transactionManager"**/>   </**beans**> |

### 4. 测试

编写测试类User

|  |
| --- |
| **public** **class** Login {    **private** String username;    **private** String password;  **public** String getUsername() {  **return** username;  }  **public** **void** setUsername(String username) {  **this**.username = username;  }  **public** String getPassword() {  **return** password;  }  **public** **void** setPassword(String password) {  **this**.password = password;  }  @Override  **public** String toString() {  **return** "User [id=" + id + ", username=" + username + ", password=" + password + "]";  }    } |

Jsp

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <%@ page isELIgnored=*"false"*%>  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  <html>  <head>  <title>login jsp</title>  <script type=*"text/javascript"* src=*"./js/jquery-1.8.3.js"*></script>  </head>  <body>  <h2>login jsp!</h2>  <div  style="width: *400px*; height: *300px*; margin-left: *400px*; margin-top: *130px*; background-color: *#CCCCCC*">  <form action=*"*${pageContext.request.contextPath }*/user/login.do"*>  <table>  <tr style="margin: *10px*;">  <td>user:</td>  <td><input id=*"username"* name=*"username"*></td>  </tr>  <tr style="margin: *10px*;">  <td>password:</td>  <td><input id=*"password"* name=*"password"*></td>  </tr>  <tr style="margin: *10px*;">  <td rowspan=*"2"* align=*"center"*>  <!-- <input type="button"  value="login" onclick="login()"> -->  <input type=*"submit"* >  </td>  </tr>  </table>  </form>  </div>  </body>  <script type=*"text/javascript"*>  **function** login() {  **var** username = $('#username').val();  **var** password = $('#password').val();  $.get("${pageContext.request.contextPath }/user/login.do?username="+username+"&password="+password);  }  </script>  </html> |

Controller

|  |
| --- |
| @Controller **public class** ShipController {   @Autowired  **private** LoginService **loginservice**;  @Autowired  **private** ShipService **shipservice**;  @RequestMapping(value=**"/Login.action"**)  **public** ModelAndView Login(  @RequestParam(**"username"**) String username,  @RequestParam(**"password"**) String password,  HttpSession session)  {  ModelAndView mv=**new** ModelAndView();  Login login=**loginservice**.login(username,password);  **if**(login !=**null**){  session.setAttribute(**"login"**,login);  mv.setViewName(**"/html/main.html"**);  }  **else**{  mv.addObject(**"message"**,**"登录名或密码有误."**);  mv.setViewName(**"/login.html"**);  }   **return** mv;  } |