# 快速上手SSM框架（务必结合代码理解）

[快速上手SSM框架 1](#_Toc2783)

**[ 此次项目的概要目录结构](#_Toc22111)** [2](#_Toc22111)

**[ 项目的详细目录结构以及我们需要添加或修改的文件](#_Toc30600)** [2](#_Toc30600)

[1. entity层 2](#_Toc12545)

[a) po包 2](#_Toc30186)

[i. po类 2](#_Toc15013)

[ii. Mybatis操作类 2](#_Toc15296)

[b) so层 2](#_Toc28027)

[2. provider层 2](#_Toc4181)

[a) dao包 3](#_Toc14869)

[i. dao类 3](#_Toc5791)

[ii. ExtDao类 3](#_Toc400)

[b) Mapper文件夹 3](#_Toc21823)

[i. Mapper配置文件 3](#_Toc6688)

[ii. ExtMapper 3](#_Toc21345)

[3. service层 3](#_Toc30941)

[a) service包 4](#_Toc6620)

[i. service类 4](#_Toc13052)

[ii. serviceimpl类 4](#_Toc8696)

[4. Web层 4](#_Toc23007)

[a) Controller包 5](#_Toc18375)

[i. Controller类 5](#_Toc4260)

[b) Utility包 5](#_Toc17406)

[c) Web文件夹 5](#_Toc28435)

[i. Static 5](#_Toc4506)

[ii. Web-info 5](#_Toc9504)

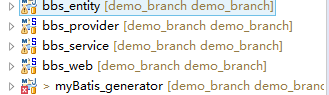
**[ mybitis\_gennerator](#_Toc1517)** [5](#_Toc1517)

[1. 作用 5](#_Toc11099)

[2. 使用方式 5](#_Toc21949)

[3. 注意事项： 6](#_Toc28678)

* **此次项目的概要目录结构**



Entity层：根据需求产生的各种类，包括po，vo，so等。

Provider层：操作数据库。

service层：通过provider层的dao接口操作数据库，实现你业务上需求功能。

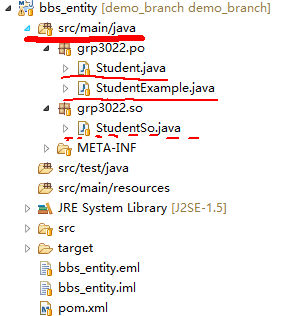
Web层：存放页面和控制器，控制器通过掉用service层的服务实现页面上的功能。

Mybatis\_generator:关于[mybatis\_gennerator](#mybitis_gennerator)。

* **项目的详细目录结构以及我们需要添加或修改的文件**

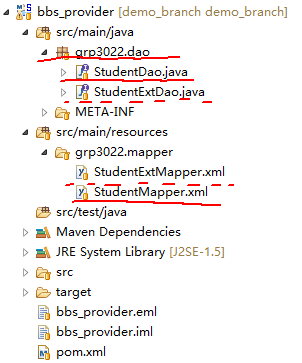
**实线为必须添加，虚线为视情况添加**

1. entity层



* 1. po包：（必须添加，由[mybatis\_generator](#mybitis_gennerator)生成，生成后直接复制粘贴至对应目录下即可）
     1. po类（图中对应student）根据数据库表映射的生成的类，表的每种属性与生成类的属性一一对应
     2. Mybatis操作类（图中对应studentexample），与\*dao.java和\*mapper.xml配套使用，详情请见代码。[使用方法参考](http://blog.csdn.net/luanlouis/article/details/22726635)。
  2. so层：（根据是否有检索的需求自行添加，图中对应StudentSo），so为search object（检索对象），根据po类的一个或多个属性，实现相关的检索功能。

1. provider层

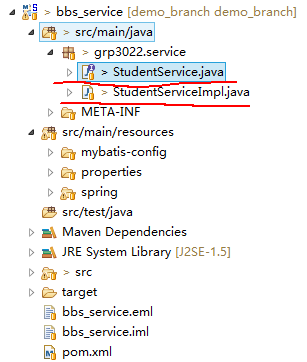


* 1. dao包
     1. dao类（必须添加，由[mybatis\_generator](#mybitis_gennerator)生成，生成后直接复制粘贴至对应目录下即可，图中对应studentDao）
     2. ExtDao类（根据需求自行添加，图中对应StudentExtDao）
  2. Mapper文件夹

作用：通过配置xml文件实现对应dao接口里的方法，里面主要是mybatis标签+sql语句，详情参考[网页](http://www.mybatis.org/mybatis-3/zh/sqlmap-xml.html)及student代码。

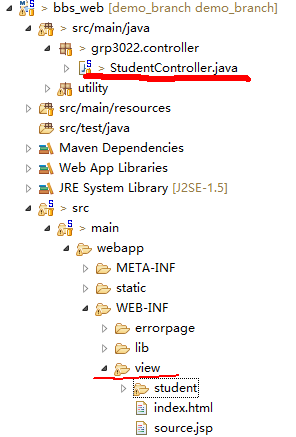
* + 1. Mapper配置文件（必须添加，由[mybatis\_generator](#mybitis_gennerator)生成，生成后直接复制粘贴至对应目录下即可，图中对应studentMapper）
    2. ExtMapper配置文件（根据需求自行添加，图中对应StudentExtMapper）

1. service层

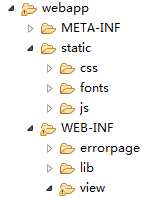


* 1. service包
     1. service类（必须添加），所需服务的功能接口。
     2. serviceimpl类（必须添加），对应接口的功能实现。详情参考student代码。

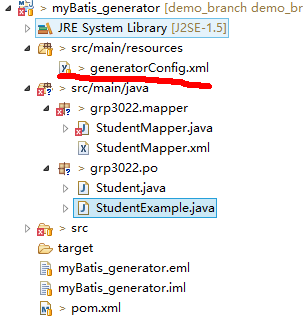
1. Web层



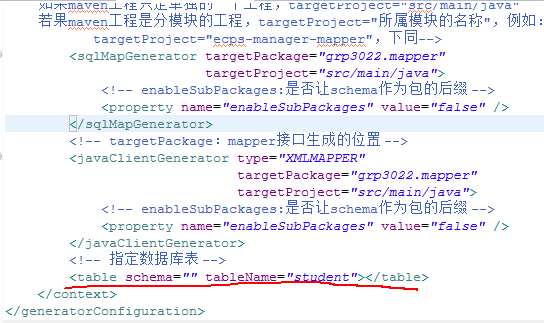
* 1. Controller包（必须添加）
     1. Controller类：根据用户不同的网址请求给用户展现不同的网页。详情参考代码
  2. Utility包：存放一些通用的工具类（类似于处理日期转换，算数处理之类的类）。
  3. Web文件夹：存放页面以及一些前段的使用工具。



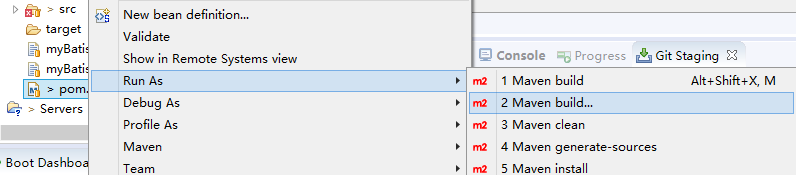
* + 1. Static：存放[css](http://www.w3school.com.cn/css/index.asp)（页面装饰）、[js](http://www.w3school.com.cn/js/index.asp)（页面脚本）、fonts（字体）、imag（图片）等静态文件，供页面调用。
    2. Web-info
       1. Errorpage：错误页面(404、400、500)。
       2. Lib：部署到服务器需要的jar包。
       3. View（必须添加）：页面（[jsp](http://www.runoob.com/jsp/jsp-tutorial.html)、html、php）。
* **mybitis\_gennerator**



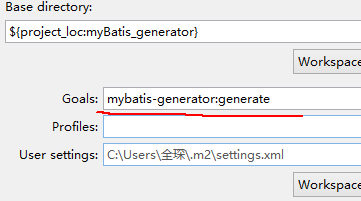
1. 作用：自动生成所需的provider层和entity层代码。
2. 使用方式：修改画线的xml配置文件。



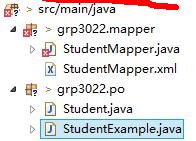
在文件最后的划线处，需要生成哪个表的表名就写该表的名称即可。



写完表名之后右键想要木最下端的pom文件，点击run as ->maven build



在goals栏输入：mybatis-generator:generate，点击run。对应目录下会生成你要的文件。



1. 注意事项：
   1. 生成结束之后文件名称可能与项目中的名称约束不太一样，记得及时修改。
   2. 最好不要一次性生成很多表，在写项目某个模块的时候，生成对应的表即可。
   3. 切忌重不要重复生成，已存在的文件里面的内容会重复添加，即使删除不需要的文件。
   4. 必须要建立相应的目录结构才能生成文件。

* **尝试**

数据库新建两个表，一个specialty表，要求具有就业率属性（其他属性自行设计）；一个professor表，要求具有工龄属性以及specialty外键，两个表都参照student模板具有增删改查功能（so类自行设计）。

specialty页面添加就业率从高到低显示的操作按钮，放置在“添加”按钮旁边，操作栏添加“查看”按钮，能够显示该specialty下的所有professor。

professor页面添加工龄从高到低的按钮显示的操作按钮，放置在添加按钮旁边。