基于SSM框架实现学生管理系统

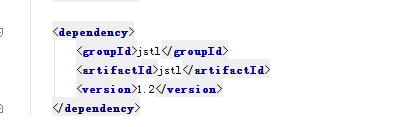
# 1.SSM框架的搭建

## 1.1 创建Maven项目：

创建Maven项目，并创建一个java文件夹作为源代码文件夹。

## 1.2 Pom.xml文件的配置：

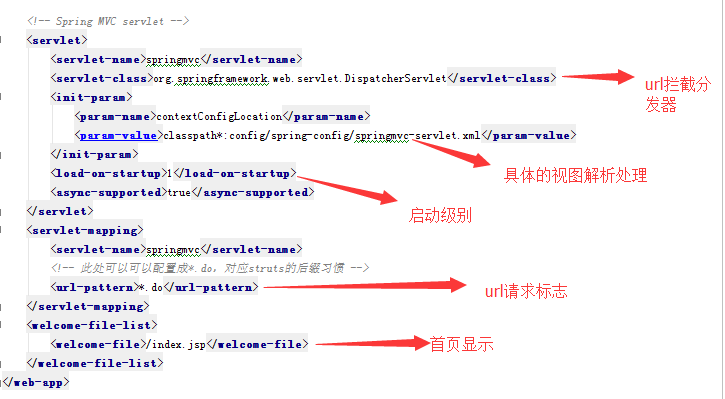
Pom.xml文件具体的作用是实现动态网页下载jar包的文件，只需通过书写相应的jar包版本信息，通过reimport下载所需要的版本。（例如下图的一个jar包例子）



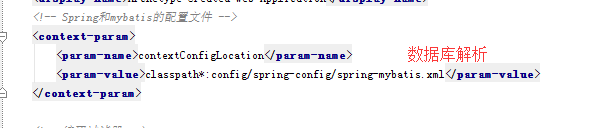
例图：c标签的jar包下载信息

## 1.3 Web.xml文件配置信息：

Web.xml的文件目的：是对用户的url请求地址的处理器；其中，首先需要定一个项目启动时显示的界面，之后配置DispatcherServlet，所有的url的请求的捕捉的一个类，捕捉之后则会跳入对应的springmvc-servlet.xml文件中，（springmvc-servlet.xml具体配置以及作用后面详细解说）。其中，还需要配置数据库连接参数的（spring-batis.xml具体配置以及作用后面详细解说），用于数据库连接。包括字符转码文件、监听等配置。（例图中的两个重要的xml文件配置）。



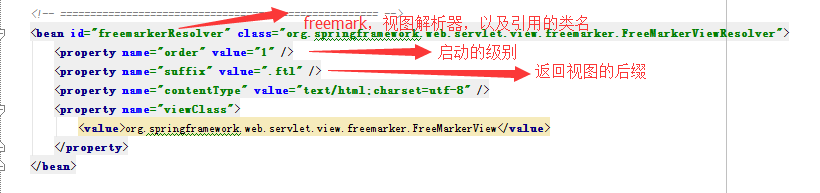
例图：springmvc-servlet.xml解析



例图：spring-batis.xml解析器

## 1.4 springmvc-servlet.xml文件配置加解析：

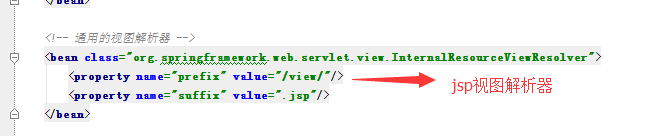
首先，springmvc-servlet.xml的作用是实现用户的请求的url在contrloer层中处理后的视图返回解析器。创建的文件的位置，则位于resources的config.spring-config文件下。其中具体配置，以图文件解析。



例图：ftl格式的视图解析器



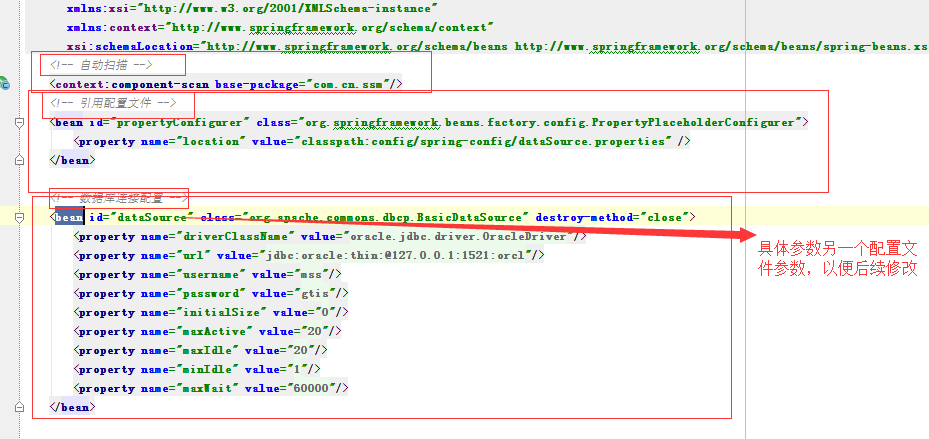
例图：freemark视图解析器的参数配置



例图：jsp界面解析器

## 1.5 spring-batis.xml 文件的作用以及配置文件解析：

Spring-batis.xml文件的作用是实现数据库连接，以及执行sql语句会话工厂的文件配置。具体解析以图文件为准。



例图：数据库连接参数配置解析



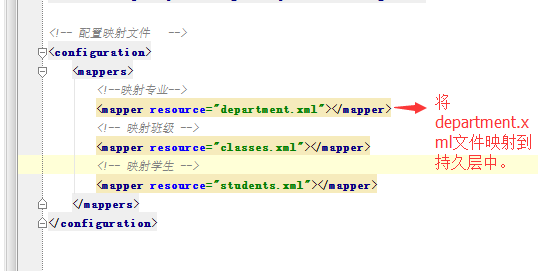
例图：数据库sql语句执行的回话工厂



例图：数据库连接参数的配置datasorce文件

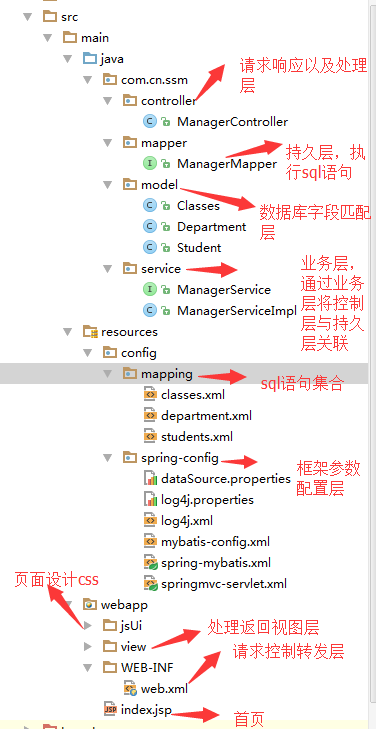
## 1.6 mybatis-config.xml文件作用以及其中的配置解析：

mybatis-config.xml文件的作用是将指定的sql语句集合，映射到持久层中。通过这个映射，持久层中的方法才可对应相关的语句执行。



例图：mybatis-config.xml文件配置解析

## 1.7 完整的一个项目的文件位置以及文件解析（例图）：



例图：完整的springmvc-框架的实现一个项目

# 2.针对该框架下搭建的学生管理系统数据库处理部分。

## 2.1 专业表创建：

create table department (depart\_id varchar(10) PRIMARY KEY, （专业id主键）

depart\_name varchar(50)); （专业名称）

create sequence dpt\_sequence increment by 1 start with 1 nomaxvalue nocycle nocache;（触发器）

create trigger dpt\_trig before （通过触发器实现专业id自增）

insert on DEPARTMENT for each row when (new.depart\_id is null)

begin

select dpt\_sequence.nextval into:new.depart\_id from dual;

end;

## 2.2 班级表的创建：

create table classes (classes\_id varchar(10) PRIMARY KEY, （班级id 主键）

classes\_year integer, （班级入学年份）

classes\_describe varchar2(50), （班级名称）

depart\_id varchar(10) references department(depart\_id) );（关联专业id 作为其外键）

## 2.3 学生表的创建：

create table students (students\_id character(10) PRIMARY KEY, （学生id主键）

students\_name varchar2(10), （学生姓名）

students\_password varchar2(10), （学生密码）

students\_sex character(3), （学生性别）

students\_math integer, （学生数学成绩）

students\_english integer, （学生英语成绩）

depart\_id varchar(10) references department(depart\_id), （关联专业id 作为其外键）

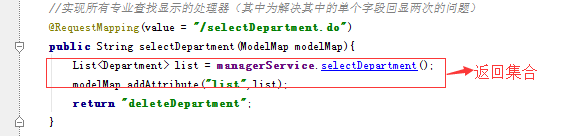
classes\_id varchar(10) references classes(classes\_id) （关联班级id 作为其外键）

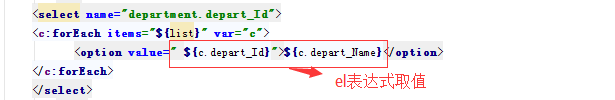
);

# 3. 实现学生管理系统的各个功能模块：

## 注意：

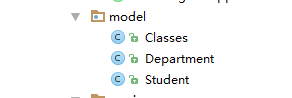
所有的返回集合都通过el表达式接收。





## 3.1 实现数据库字段匹配模块：

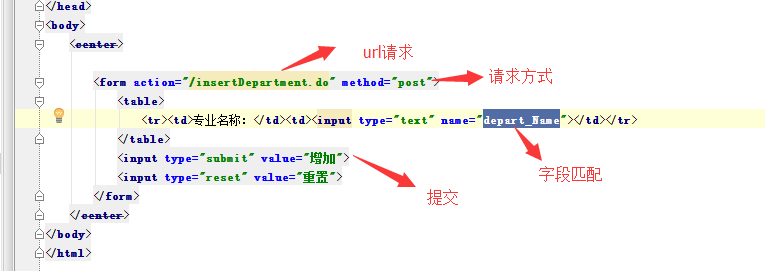
我们需要在model中建立并添加与数据库匹配的字段，生成get and set 方法。所有的类都需要实现Serializable接口。



## 3.2 实现专业功能模块：

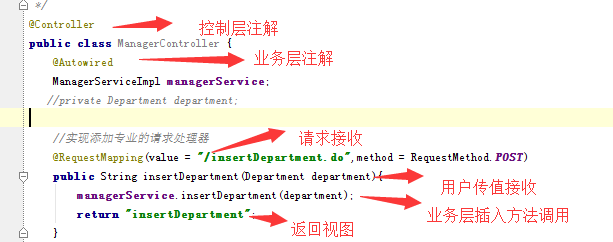
### 1.实现增加专业信息：

实现思路:对于其中的添加页面跳转首次通过静态界面跳转至可供用户输入信息界面，进入用户输入界面时，此时我们需要获取用户数据，则需要通过写一个form表单提交的方式，当用户点击提交时。（如图：）



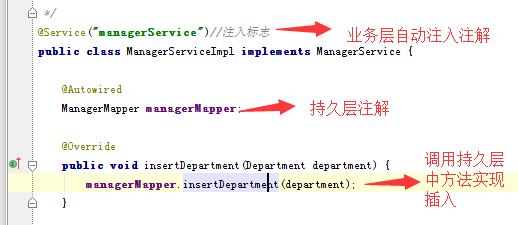
例图：实现添加界面

则将填写的专业名称传入controller层中，此时form表单的提交方法通过“.do”方式，在controoler层中通过@RequestMapping注解方式接收。在该注解下面写一个方法，返回值为String类型的字段（因为返回的视图都是通过视图解析器添加前缀以及后缀）。



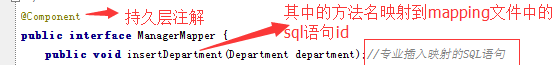
例图：控制层中解析

调用业务层中的方法后，跳入业务层，业务层中通过一个接口写抽象，一个类实现接口，实例方法，在这个类中通过注解的方式，实例持久层对象，并且方法中的执行持久层的方法，跳入持久层。

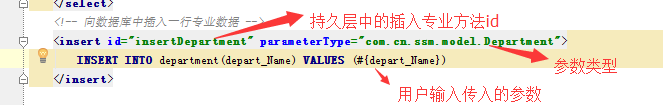


例图：业务层解析

进入业务层，业务层的方法名对应mapping文件夹下的具体的xml文件中的sql语句id来执行sql语句。



例图：持久层解析



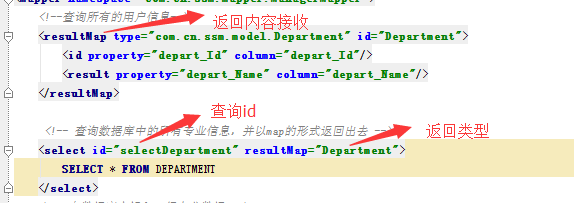
例图：sql语句解析

### 2.实现专业删除：

思路：实现专业删除，则首先要查找到所有的专业信息，再通过用户选中的专业通过id进行删除专业。其中所有的语句执行过程都是和添加专业的过程一样，只是传值的机制不一样，其中查找是通过后台要用户发送数据，而删除是用户向后台发送数据。（具体内容以图解析）



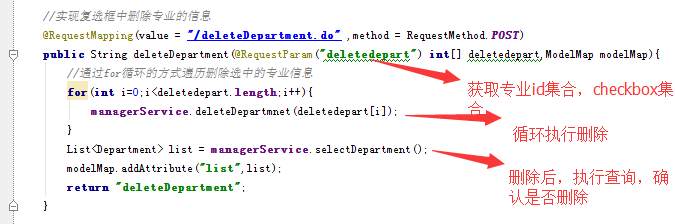
例图：controller中实现专业信息的获取



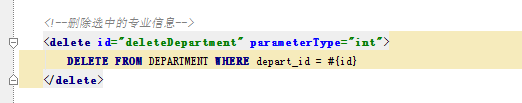
例图：业务层和持久层方法省略，该为查询语句结果获取方式



例图：用户发送删除id信息界面



例图：用户选的删除内容通过循环删除



例图：其中通过业务层持久层省略，该为删除语句

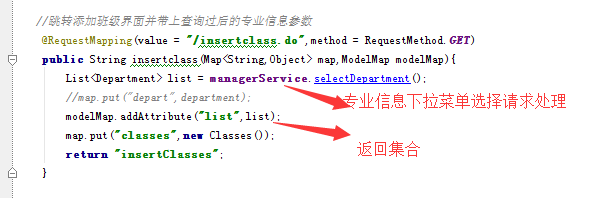
## 3.3 实现班级功能模块：

### 1.实现班级信息添加：

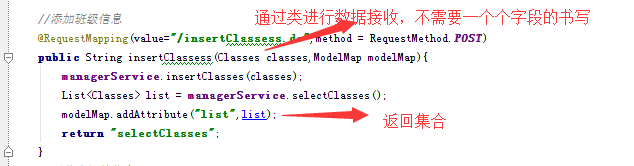
思路：实现班级添加需要首先获得数据库中的专业信息，并提供下拉餐单（对于下拉菜单需要通过首先查询数据库并通过集合返回，并通过c标签实现遍历），用户添加数据时通过form表单实现提交，并通过与添加专业信息一样的流程行走，最后添加完成。



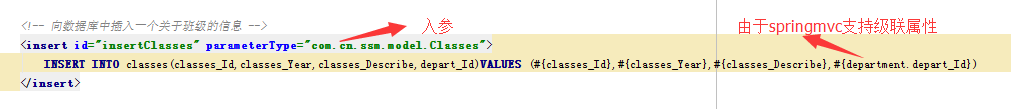
例图：用户添加班级信息界面解析



例图：实现下拉菜单的请求处理（其中的查询语句执行之前有的查询专业信息sql语句）



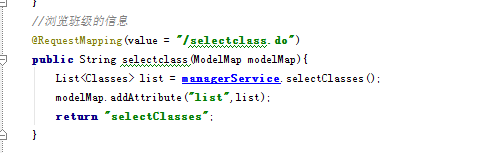
例图：实现班级信息添加的请求处理解析



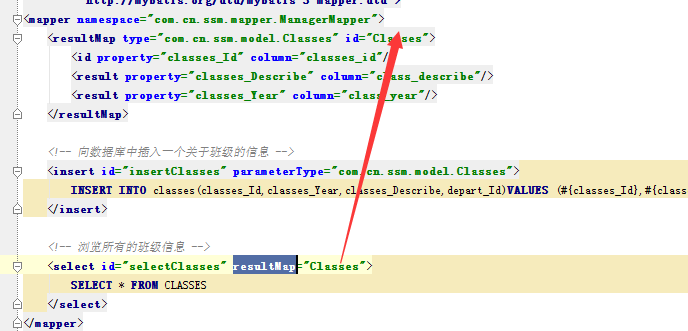
例图：实现班级信息添加sql语句

### 2.实现班级信息查询：

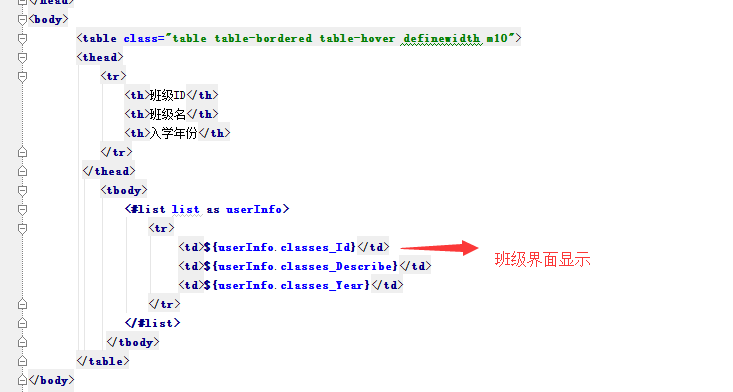
思路：班级信息的查询与专业信息查询方式一样，返回一个集合。界面el表达式显示：



例图：实现查询班级信息



例图：sql语句部分



例图：实现班级查询结构界面

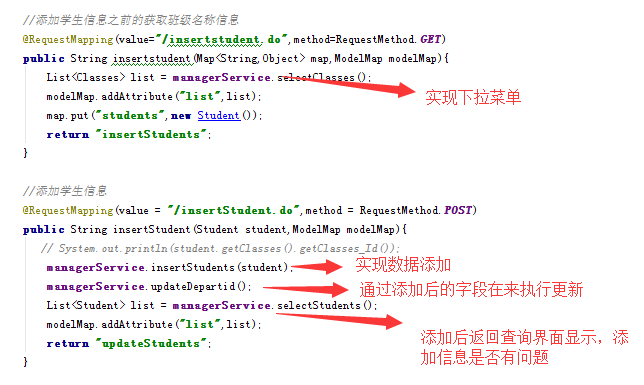
## 3.4 实现学生功能模块：

### 1.实现学生信息的添加：

思路：实现学生添加与班级信息添加方式一样，也用到回传下拉菜单，但是添加的过程中，由于专业id是不能获取的（这里有两种解决方式一是通过添加一个隐藏的下拉级联专业信息之后提交后会随带专业id，二是通过添加信息之后再classes\_id字段在进行update更新字段depart\_id字段），我选择第二种，较为方便。



例图：添加学生信息界面解析



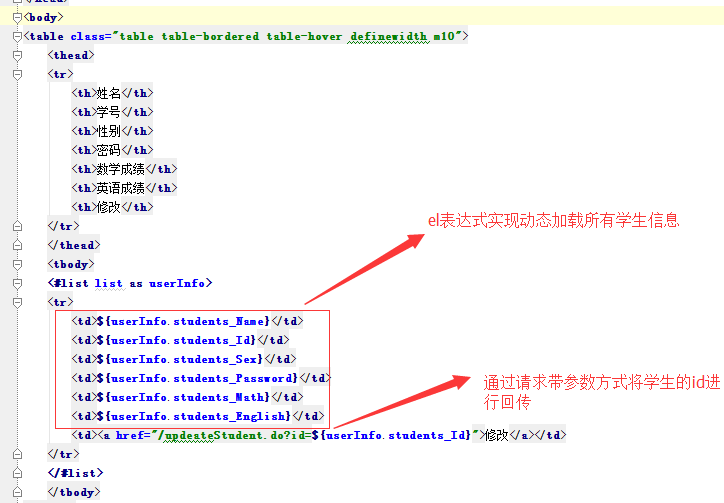
例图：学生添加功能与下拉菜单的请求处理解析



例图： sql语句解析

### 2.实现学生信息的查询并对单个学生进行修改：

思路：对于其中的单个学生的信息更新，首先将所有的信息进行查询出，并通过，添加链接，以及url请求中带上学生的id参数，通过id查找相应的学生信息并返回界面显示，以及修改后，在通过update进行更新。



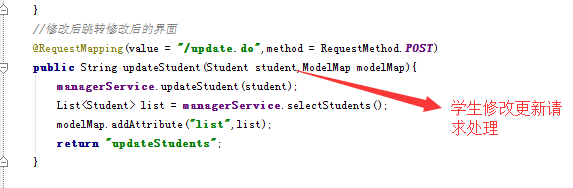
例图： 所有学生信息以及单个学生修改解析



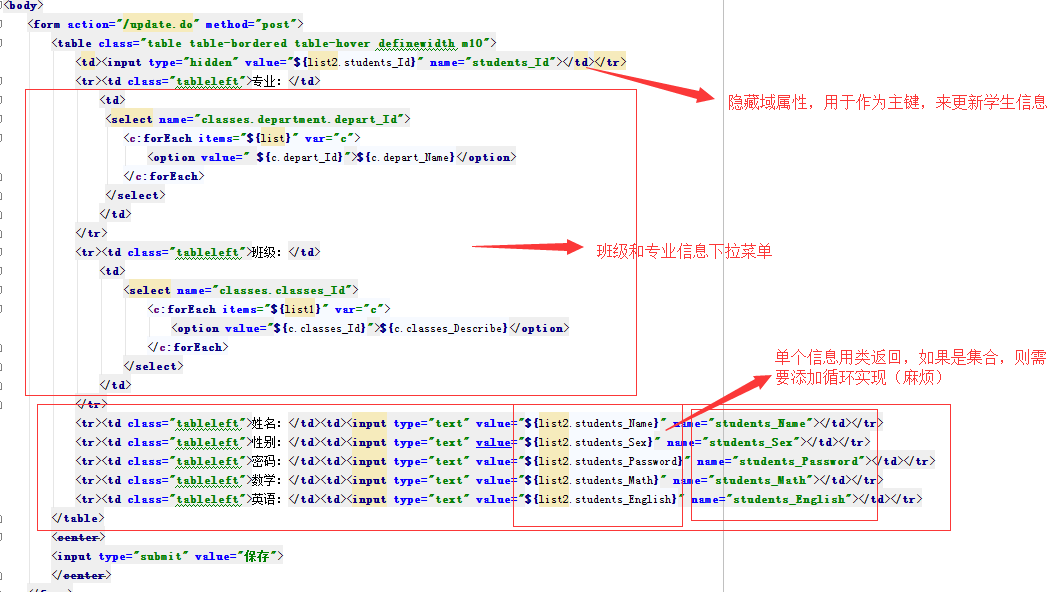
例图：请求处理解析



例图：sql语句解析



例图：学生的信息修改请求解析

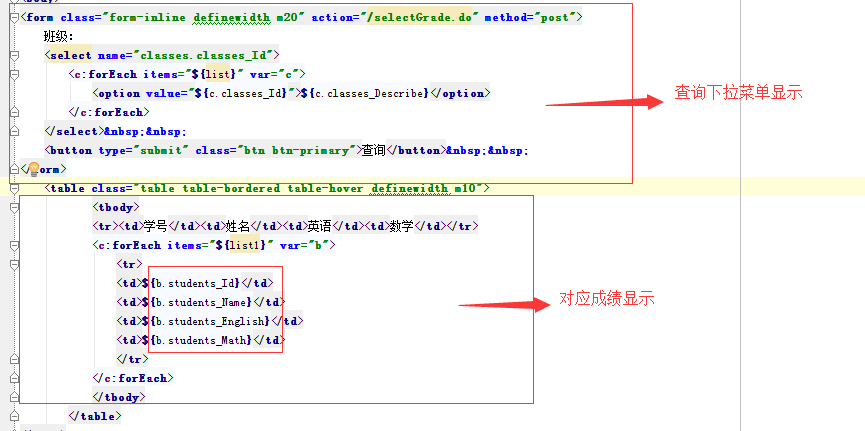


例图：单个学生更新界面解析

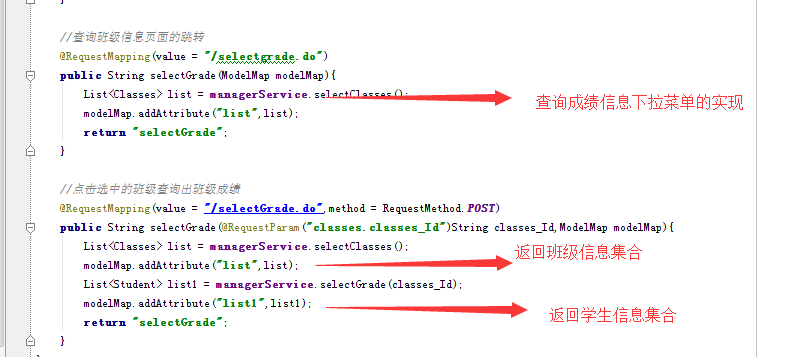
## 3.5 实现查询成绩功能模块：

### 班级信息查询成绩;

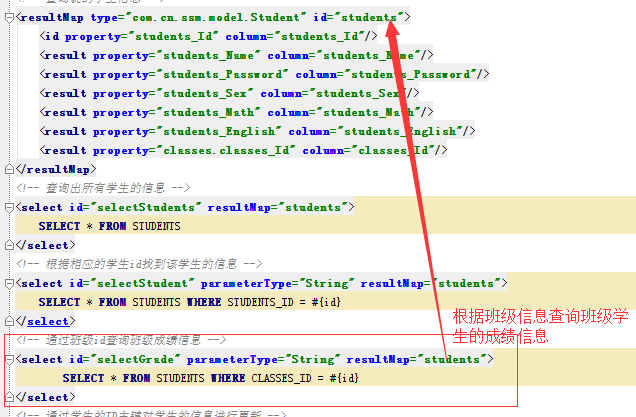
思路：首先通过实现一个班级信息的下拉菜单实现，一个form表单提交后，获取班级id，之后通过查询语句返回集合，班级所有的学生的学号，姓名，以及英语、数学等成绩。



例图：查询成绩界面显示解析



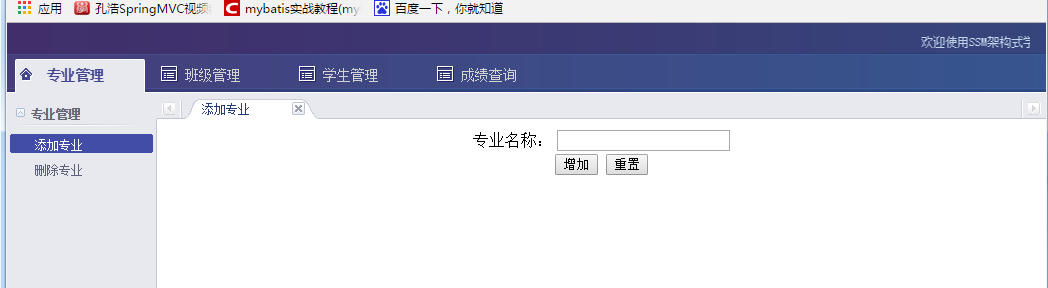
例图：实现成绩查询请求处理解析



例图：sql语句解析

# 4.程序执行的结果界面展示：

## 4.1专业功能模块展示：



例图：添加专业



例图：专业信息删除展示

## 4.2班级功能模块展示：



例图：班级添加展示



例图：浏览班级信息展示

## 4.3学生功能模块展示：



例图：学生信息添加展示



例图：所有学生信息展示



例图：修改学生信息的展示

## 4.4查询成绩界面展示：



例图：学生成绩查询展示