技术

[1 标题1 错误!未定义书签。](#_Toc449540698)

[1.1 标题2 **错误!未定义书签。**](#_Toc449540699)

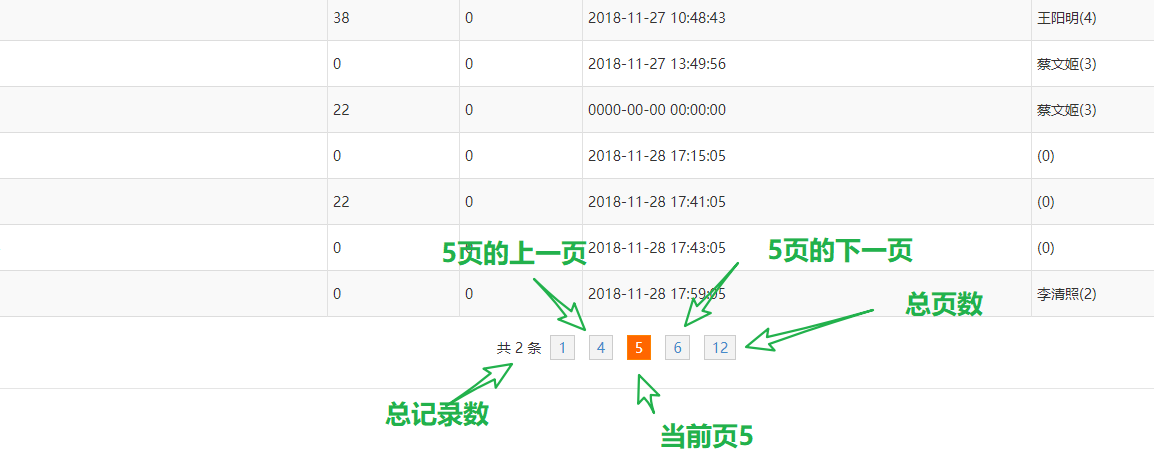
[1.1.1 标题3 **错误!未定义书签。**](#_Toc449540700)

# 列表分页

## 分析

分页，数据量过大，在一个页面不能方便的查看，需要一部分一部分的查看。需要使用使用分页。

效果是：



需要的工作：两项工作

* 依据页面获取当页数据（记录），需要查询中的limit实现
* 提供翻页链接（其他的操作方式也可以，例如输入页码等）。

## 传递页码

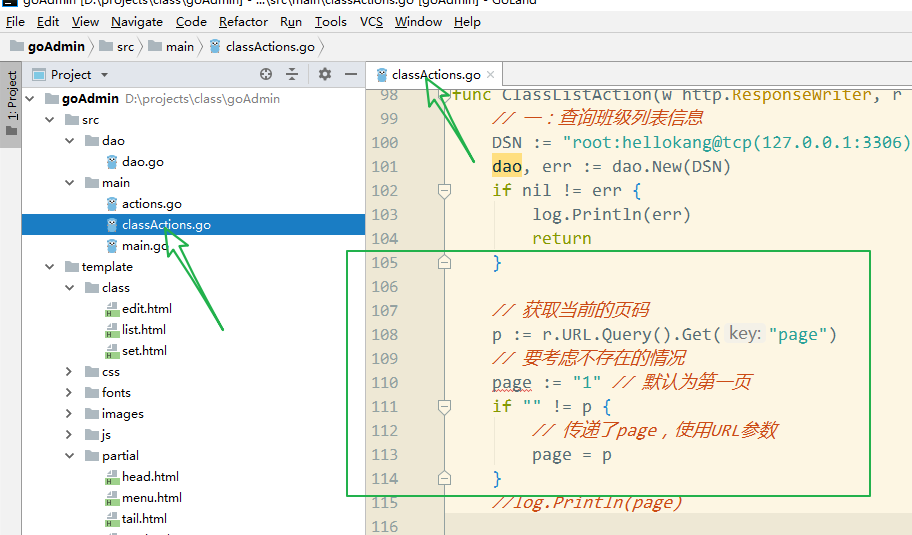
在 ClassListAction 动作中，

在获取列表数据时，要确定，当前的页码。有俩种情况：

1. 用户通过请求URL指定。例如：/class/list/?page=4 以上的URI就表示要获取第4页的数据。
2. 默认为第1页，当没有指定页码时，认为需要显示第一页数据。例如第一次进入列表页时。

在 ClassListAction 中确定当前页码：

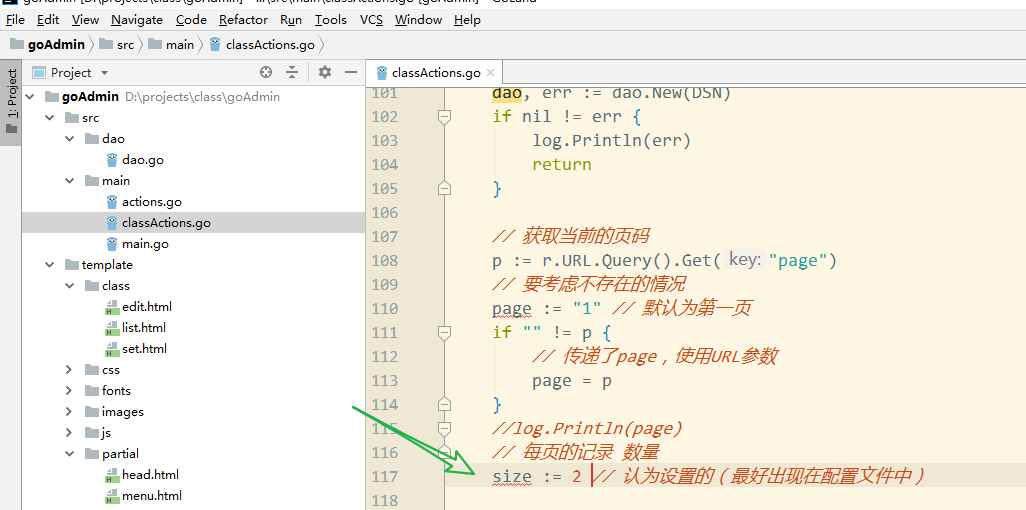
src/main/classActions.go



注意，此时获取的page为字符串类型，后面需要参与计算，需要转换为整型。

## 确定每页展示的记录数量

认为控制的。



## 利用limit限定获取的数据

limit offset 语法：

offset的获取方案：

page 和 size 获取 offset：

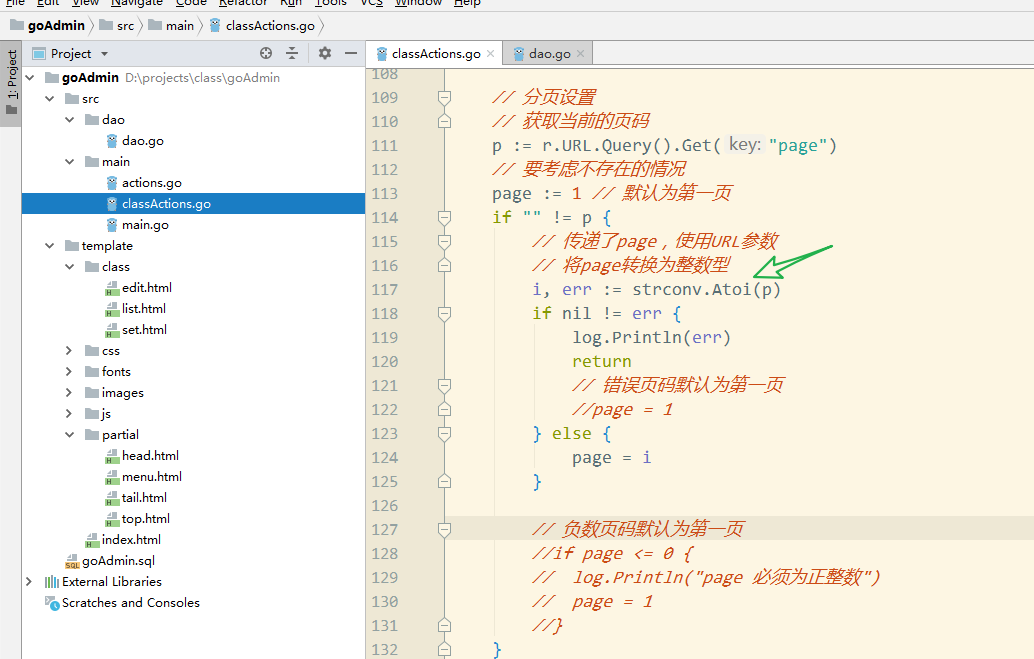
offset = (page-1) \* size

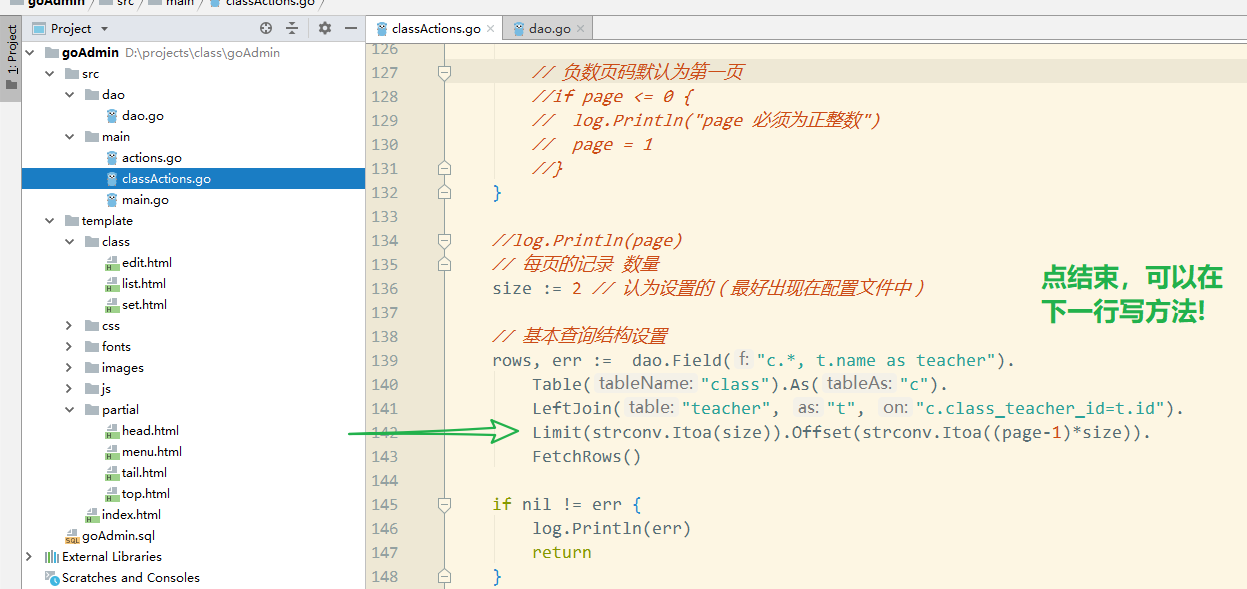
|  |  |  |
| --- | --- | --- |
| page | offset | Limit |
| 1 | 0 | 3 |
| 2 | 3 | 3 |
| 3 | 6 | 3 |

操作时，需要考虑类型计算转换的问题：

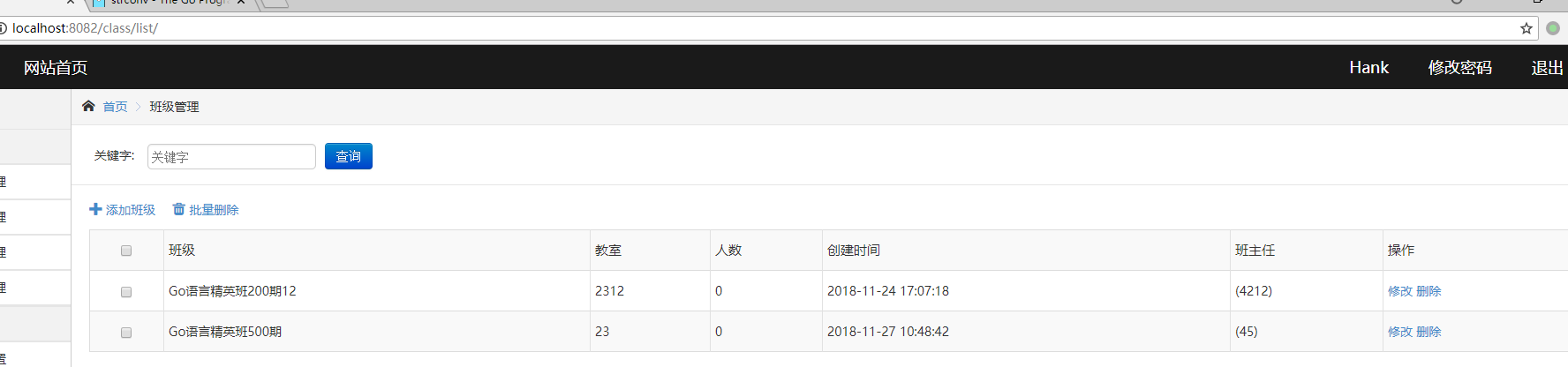
strconv.Itoa(): 整形转字符串

strconv.Atoi()：字符串转整型

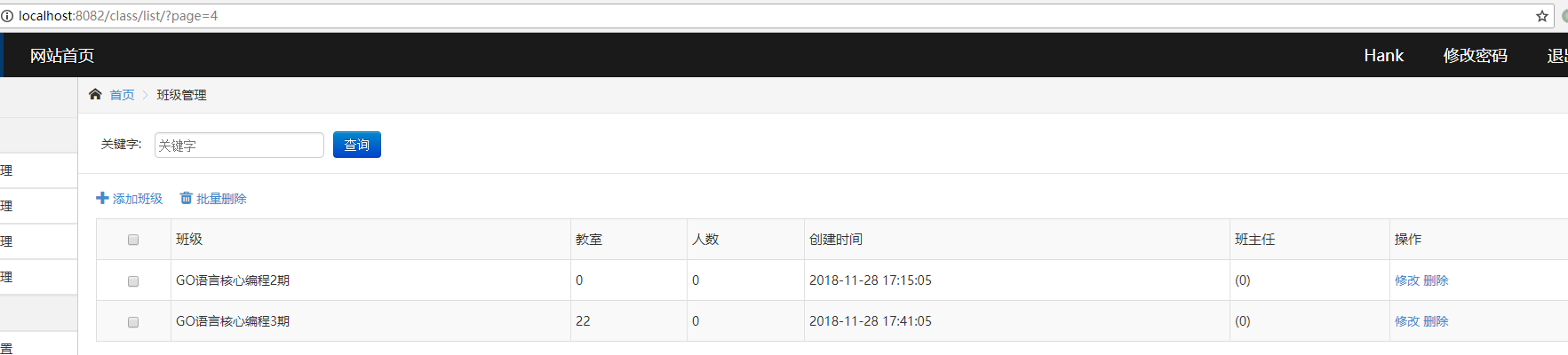




默认页码



URL传递页码：



此时，利用页码获取数据操作完成！

## 翻页分析

形成链接地址URL，要传递page参数进行请求。

需要的数据：

当前是第几页？page

一共有多少记录数？select count(\*) from table

一共有多少页？向上取整(count(\*)/size)

ClassListAction()

## 总记录数，总页数



需要使用独立的查询，获取总记录数。count合计函数。

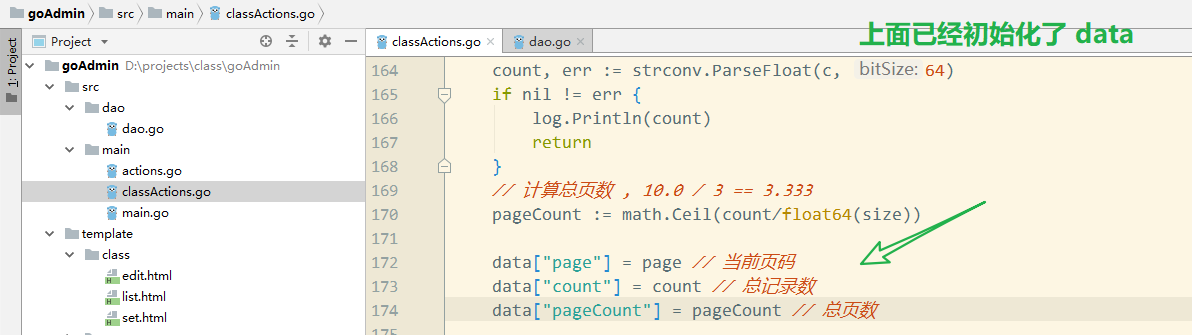
由于计算时，可能有小数部分，使用浮点数进行计算。

strconv.ParseFloat()从字符串中解析浮点数

math.Ceil()将浮点数向上取整（天花板取整）

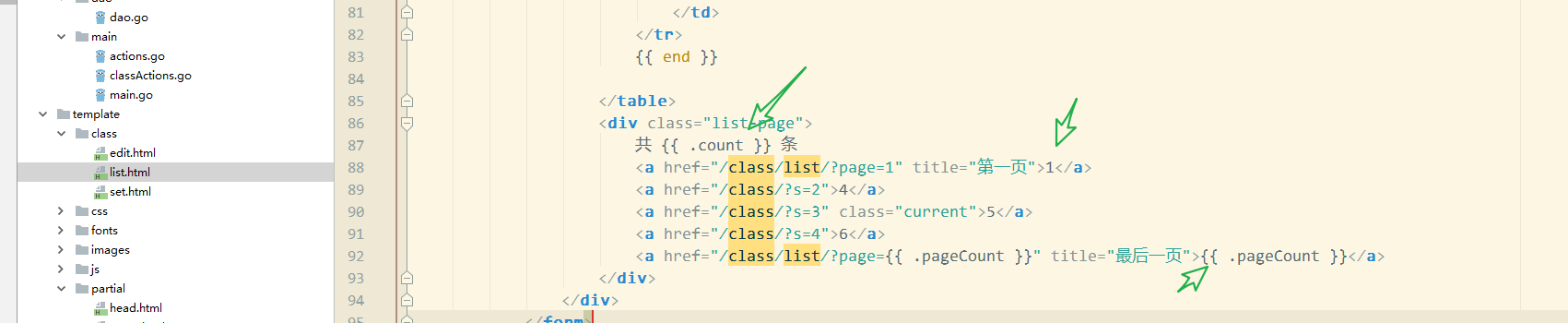
## 展示翻页链接

将 当前页码，总记录数和总页数，分配到模板中使用：



模板中展示：

总记录数，总页数，第一页：

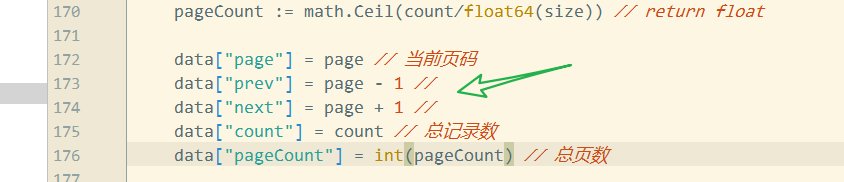


中间的页码：

设计：展示当前页码，以及其前一页和后一页：

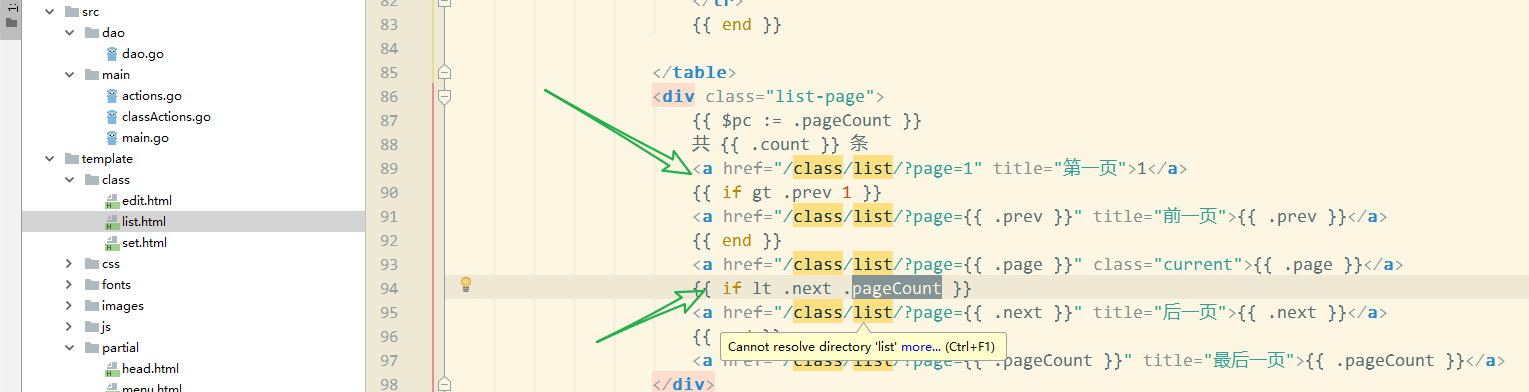
需要将前一页和后一页，计算好后分配到模板中：

ClassListAction()

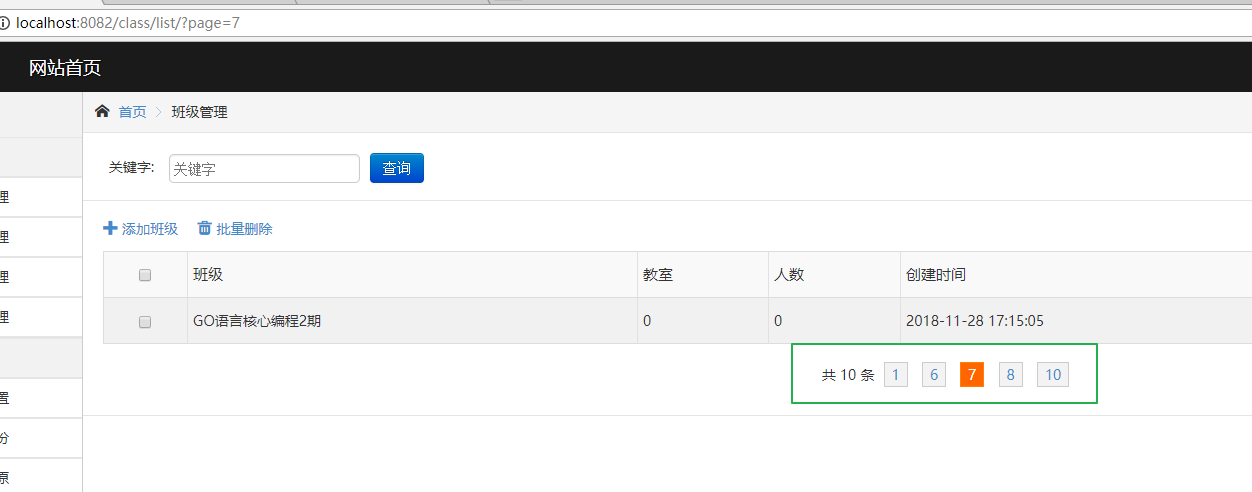


对于前一页和后一页在显示时，越界的情况，加以判断：

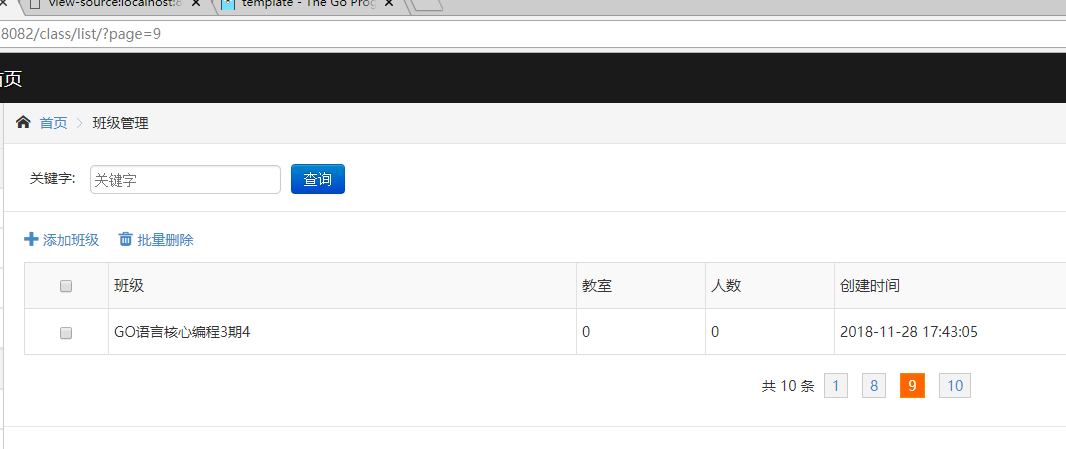
list.html



效果：







## 输入页码翻页

增加一个表单，用于输入页码：

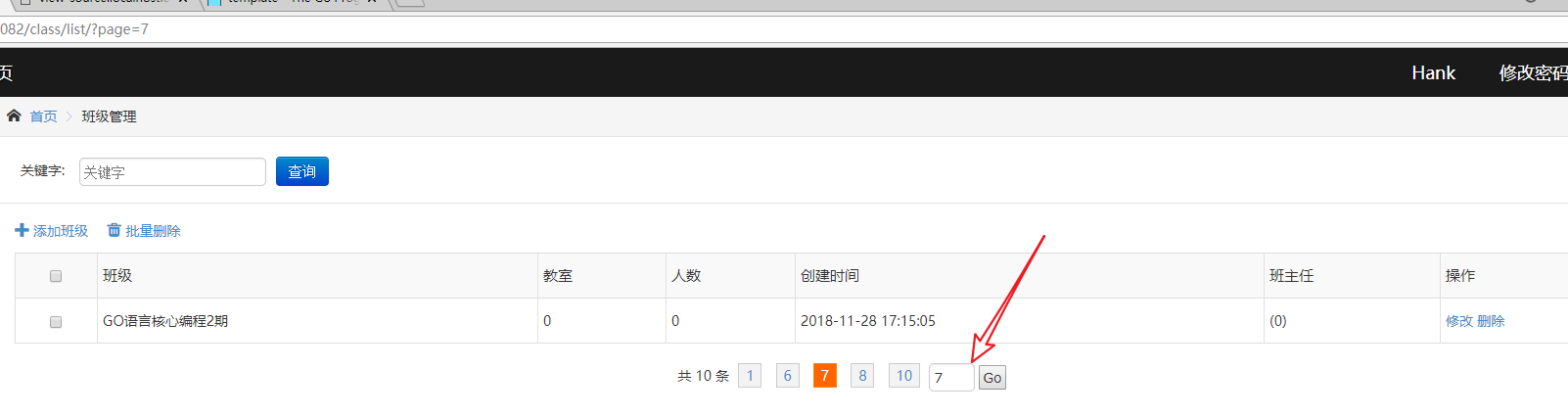
action为 /class/list/

method 为 get，是获取数据：

list.html



效果可以实现：



# 搜索

功能分析：提供输入框，键入搜索关键字后，依据：班级名称，教室，教师 进行搜索。

提供一个搜索表单，get的提交方式。

action地址也是/class/list/

具体实现，利用搜索关键字，在查询时，设置条件。

搜索时使用， like 模式匹配搜索，要求使用前缀搜索， field like ‘keyword%’。

## 搜索表单

list.html



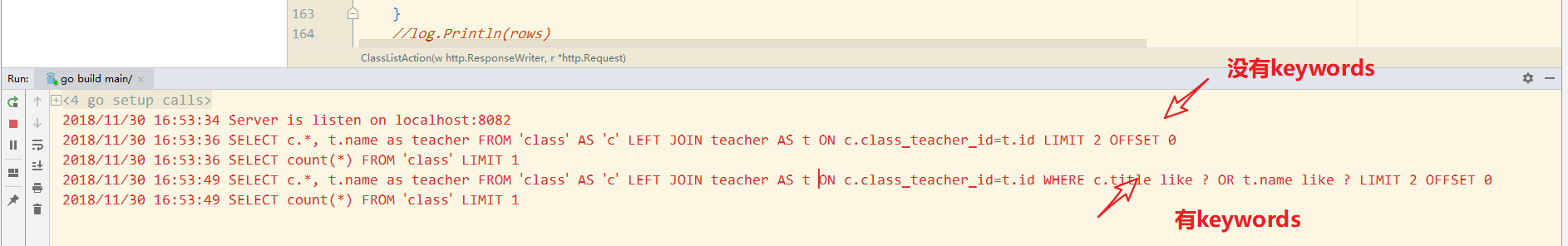
## ClassListAction动作

在拼凑where条件时，需要考虑：

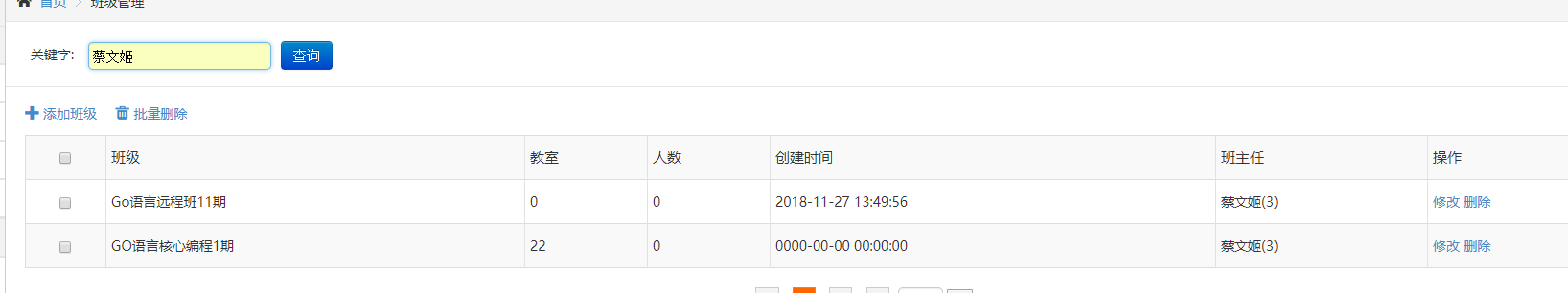
当没有指定查询条件时，不需要where子句。



测试：



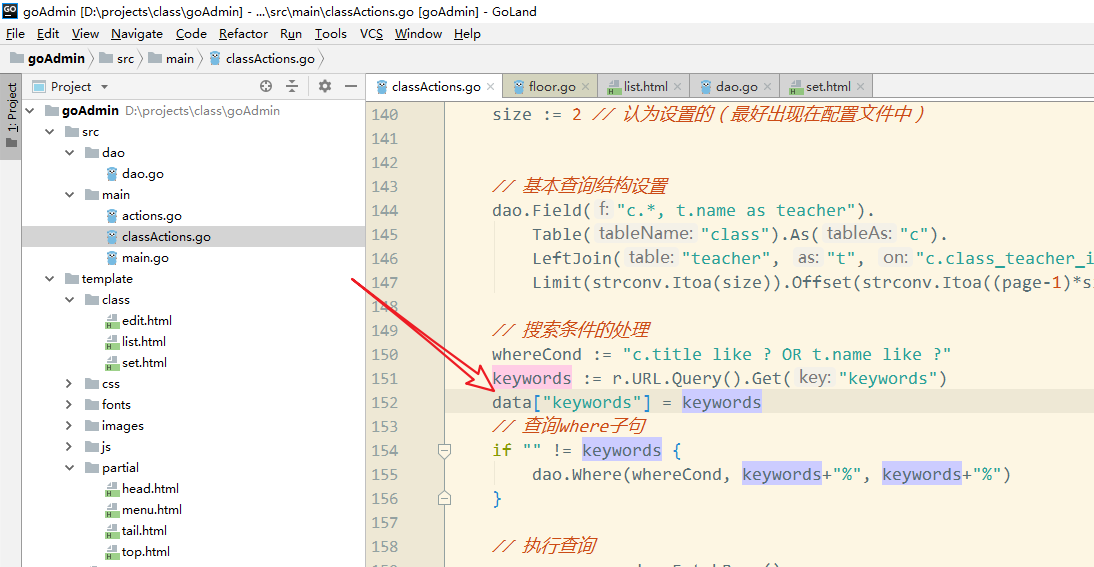
结果：



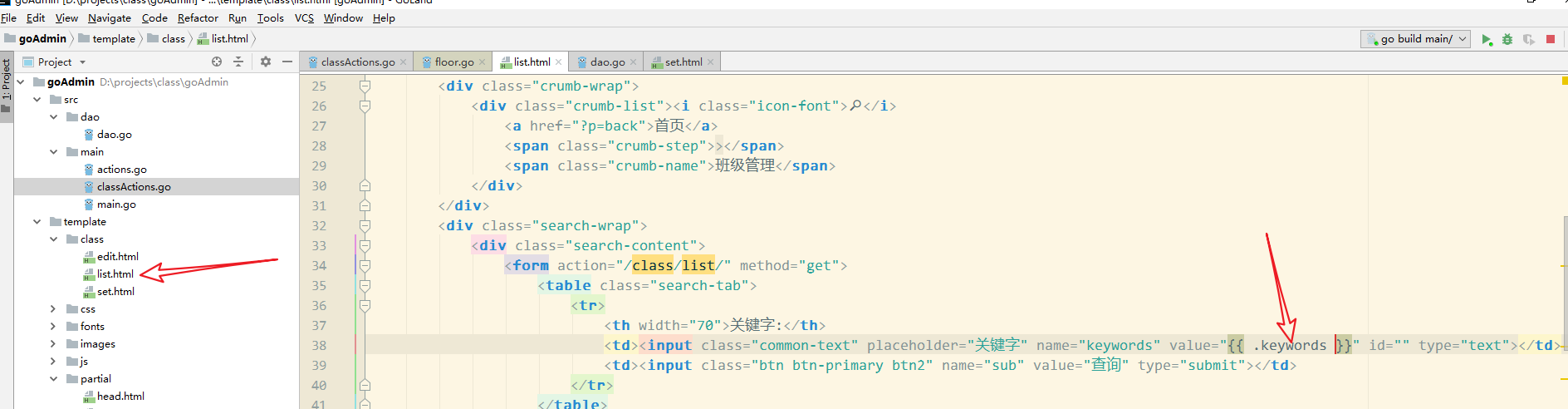
## 记录搜索关键字

在查询出结果时，搜索关键字显示在搜索框内：

在 classListAction中，将搜索keywords分配到模板中：



在 list.html 输出：

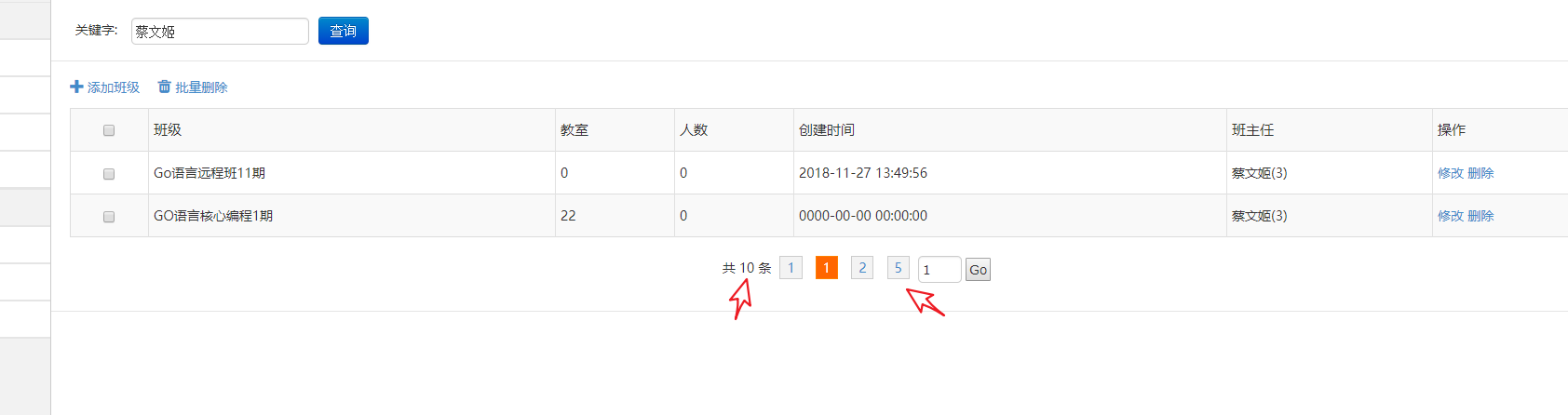


结果：



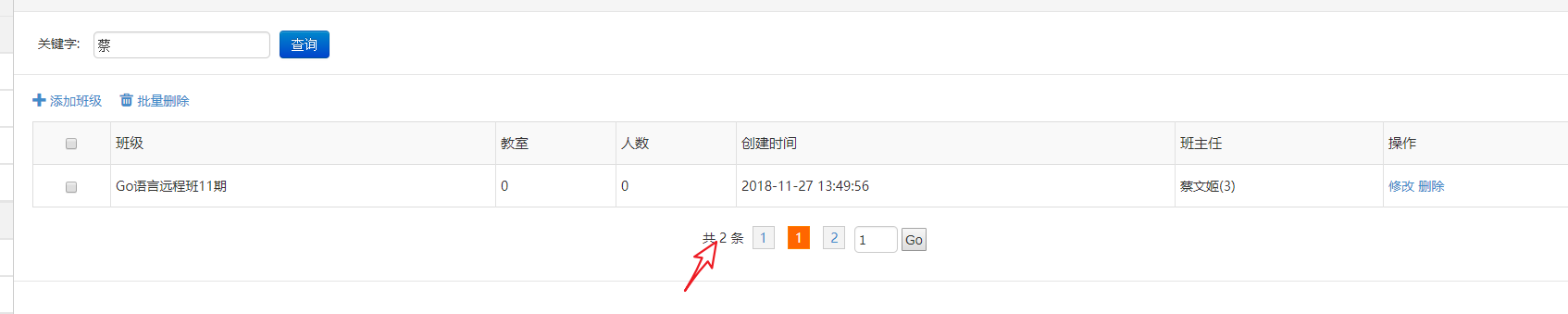
## 与翻页的整合

问题一：搜索条件与总记录数统计条件不匹配，导致结果的记录数统计不对。



解决方案，在查询count()时，指定条件即可：

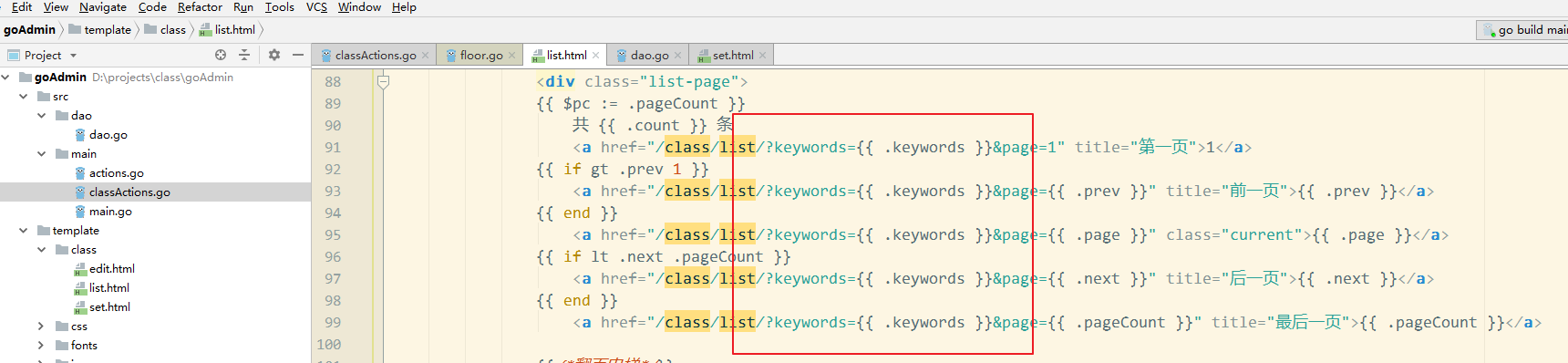




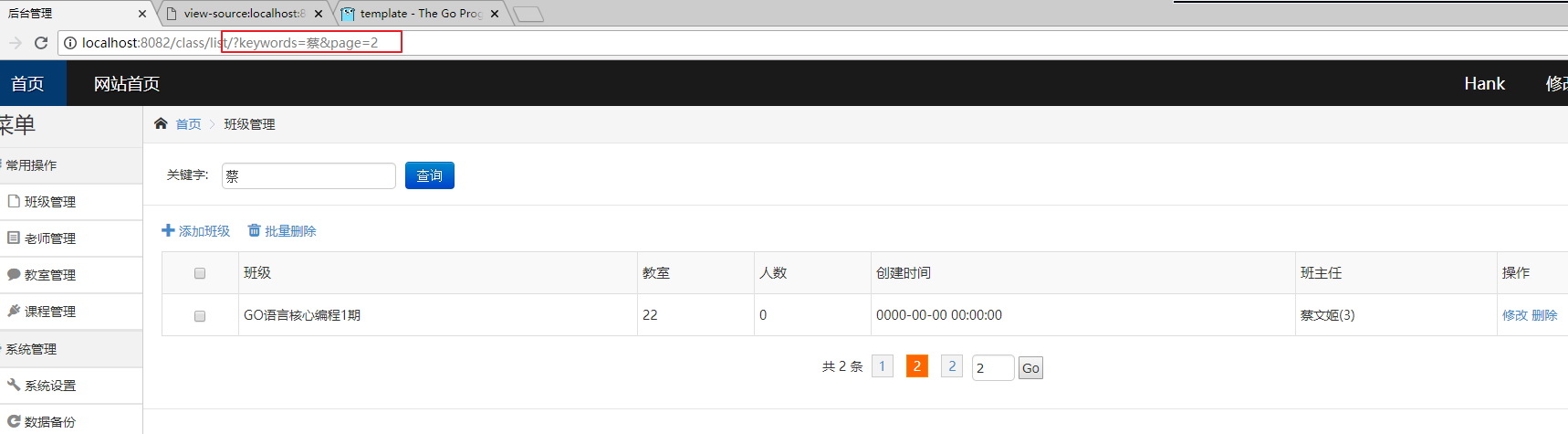
问题二：翻页时需要保留搜索条件，翻页展示的还是搜索的结果。由于翻页请求，没有携带搜索关键字条件。

在形成翻页链接时，将keywords作为参数传递即可

list.html



结果：



# 批量删除（删除）

## 全选，全不选

思路：一个用于全选的checkbox（复选框），他的状态（是否选择），决定每条记录的checkbox的状态。

演示，使用jquery实现：

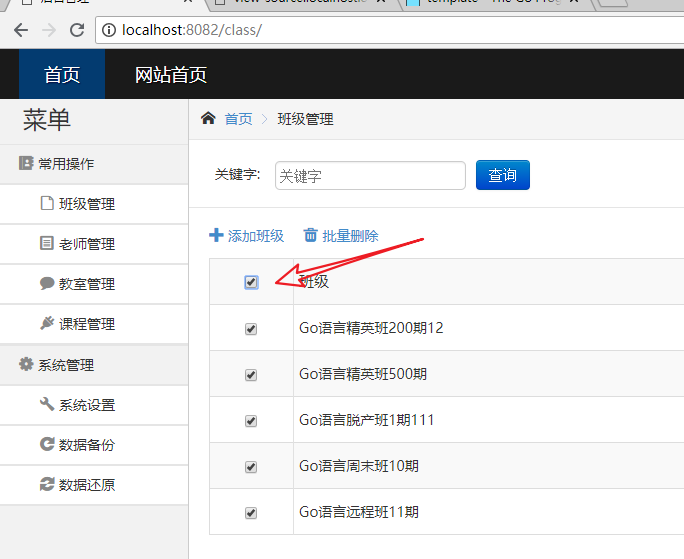




在Go中，不会默认解析[] 集合的提交，因此使用常规name即可，例如 ids：

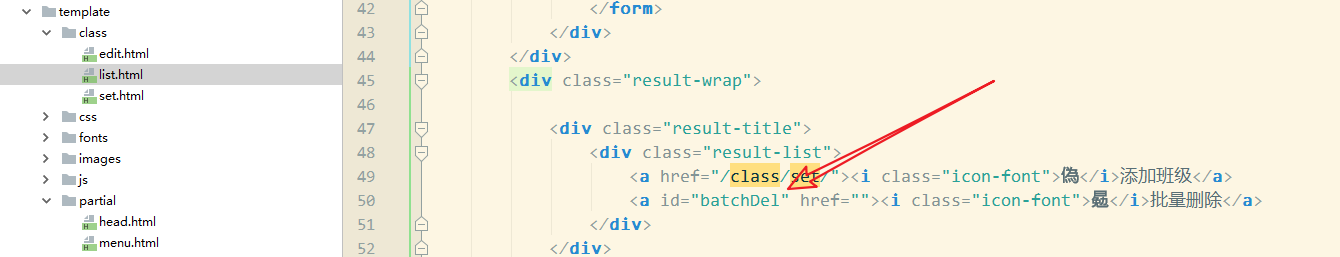


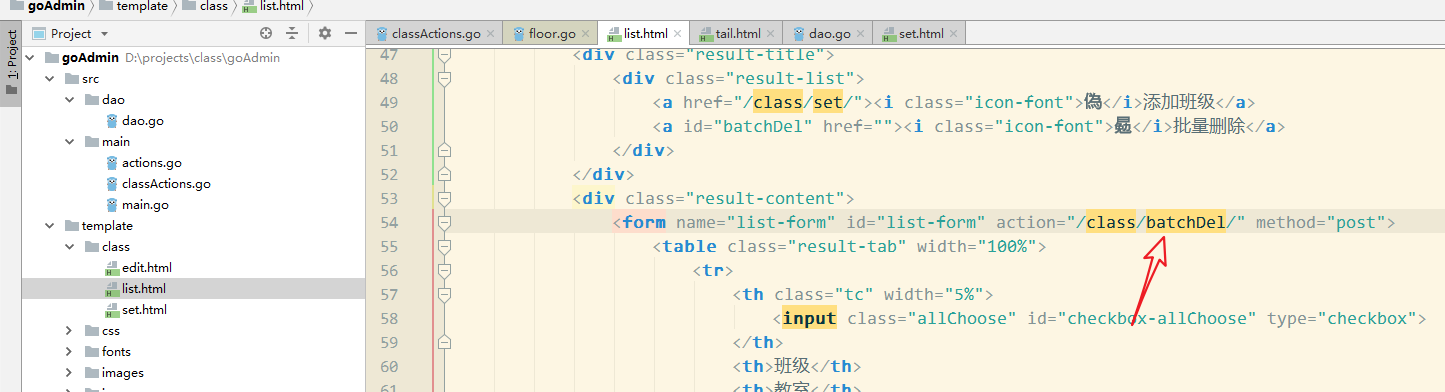




## 表单提交

在链接上绑定事件，将表单提交





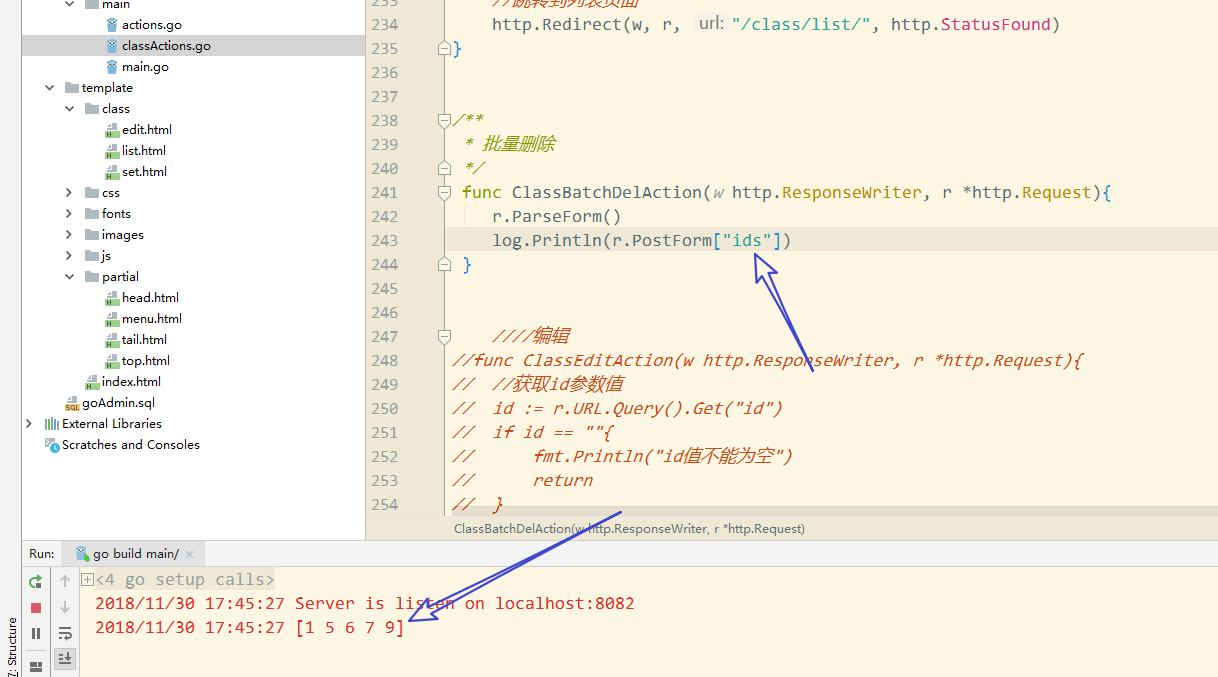


## 在服务器端，增加动作处理批量操作

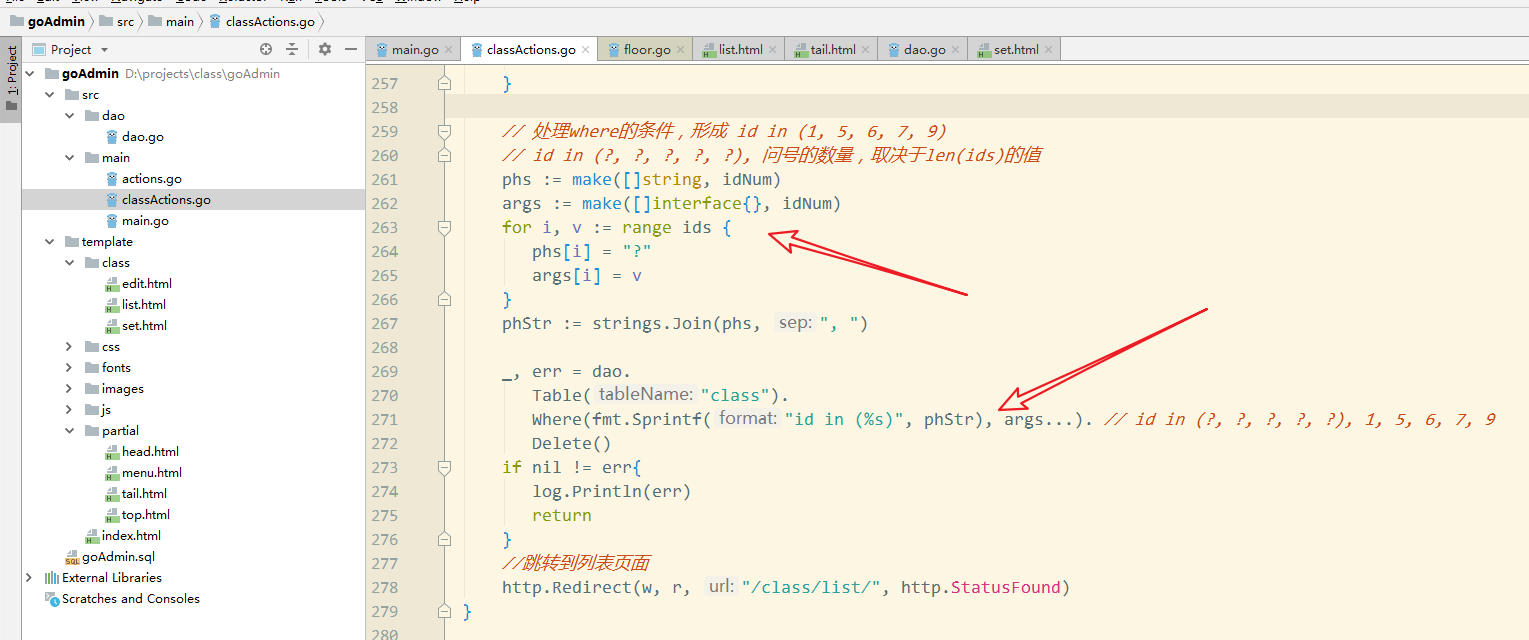
路由：



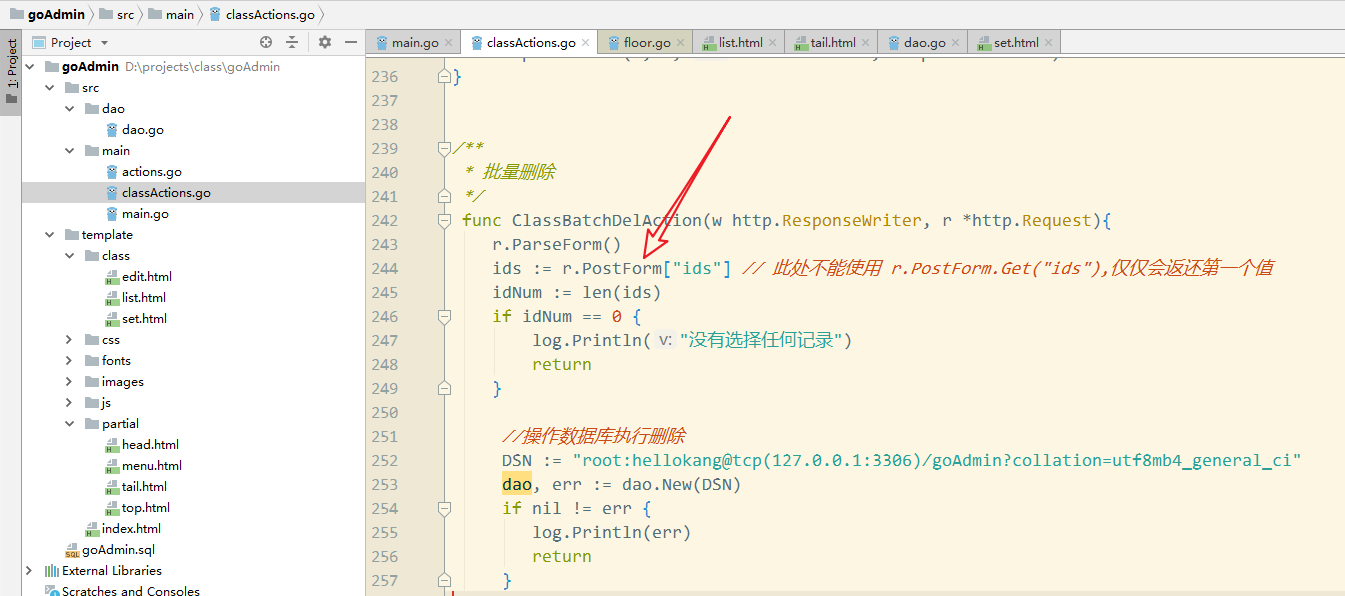
动作：服务器会得到全部提交的id数据，利用where条件操作即可，最典型的使用in作为where的运算符： where id in (1, 5, 6, 7, 9)



利用ids拼凑 where id in () 条件即可：



保证，ids存在用户所选的id，因此增加完整性判断：



# Go对请求参数的解析

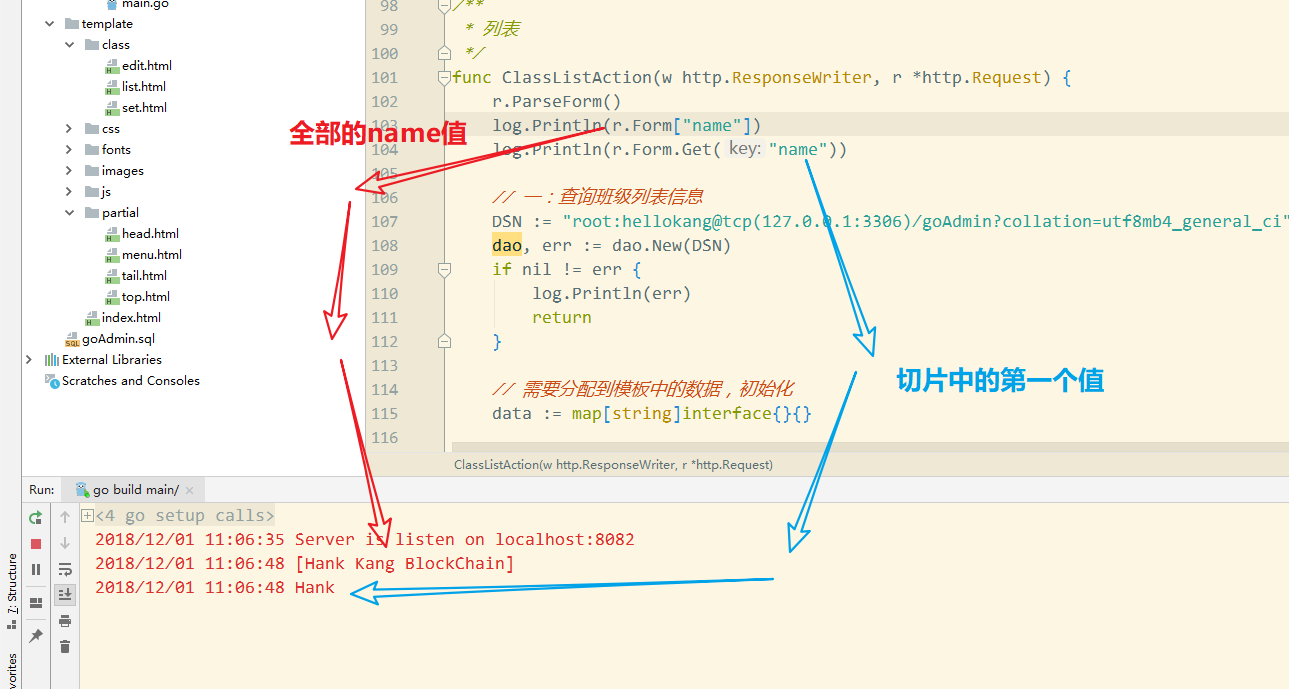
允许（认为），在浏览器请求时，会出现同名的请求数据，包括查询字符串（？GET数据）和post表单数据。



以上的name的解析，在go中，全都有效：解析到一个切片中，存储多个值。

[]中括号下标语法，可以获取该切片

.Get(key)，用于获取切片中的第一个值



功能上，如果传递的数据时多个，使用下标语法，例如批量删除中的id。

如果传递值是一个，使用.Get() 获取一个值。例如 编辑（set）时，使用的是.Get()

在很多额外的程序中，由于不允许出现同名key的情况（通常出现的后果是后边会覆盖前面的），在非Go语言程序中，如果需要传递多个数据类似于ids，需要使用 <input name=”ids[]”>来进行定义。注意name 为 ids[]。

go中不需要。

# 列表查询班级人数



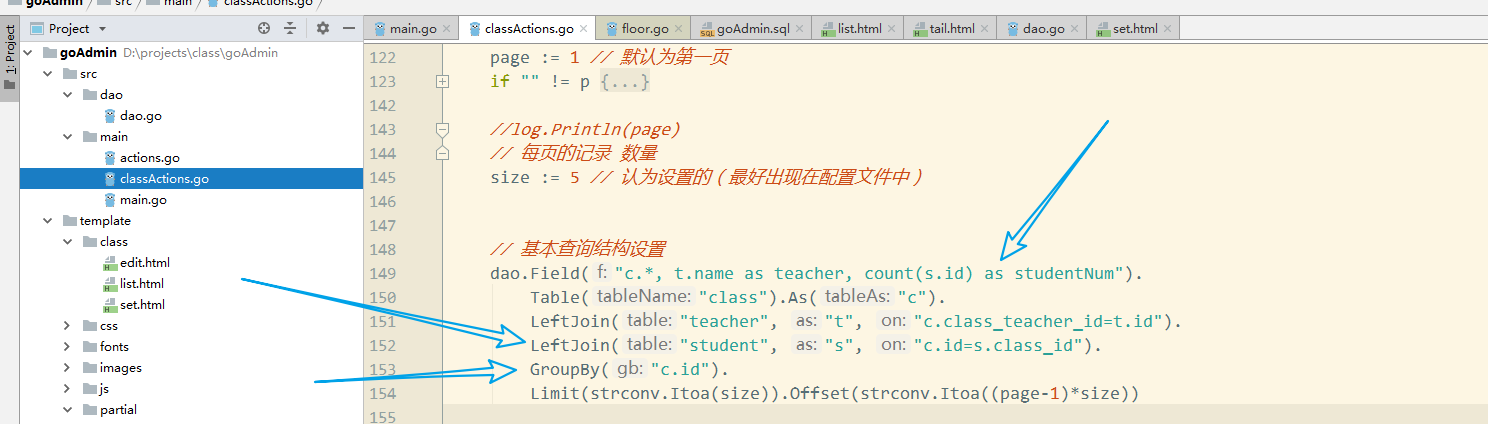
需要有学生表配合。使用测试学生数据，表示学生所属的班级：

班级和学生的关系：一对多。

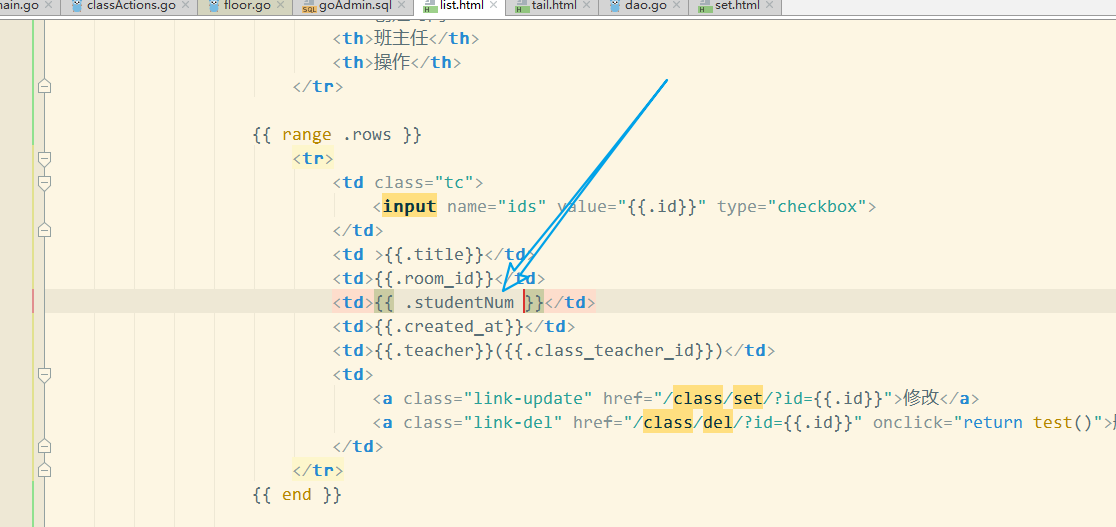


关联查询join

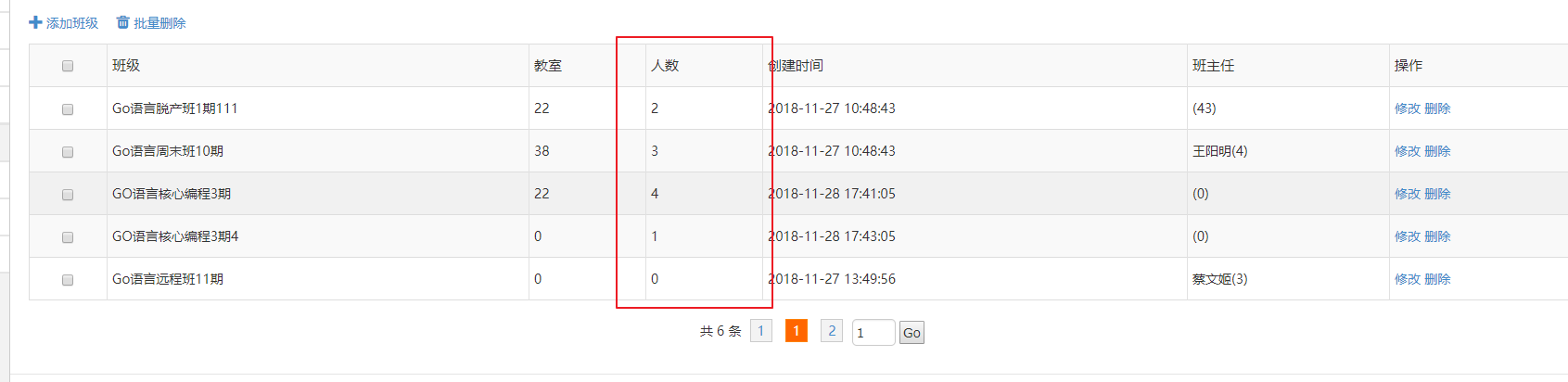
ClassListActiin()动作：



list.html



效果：

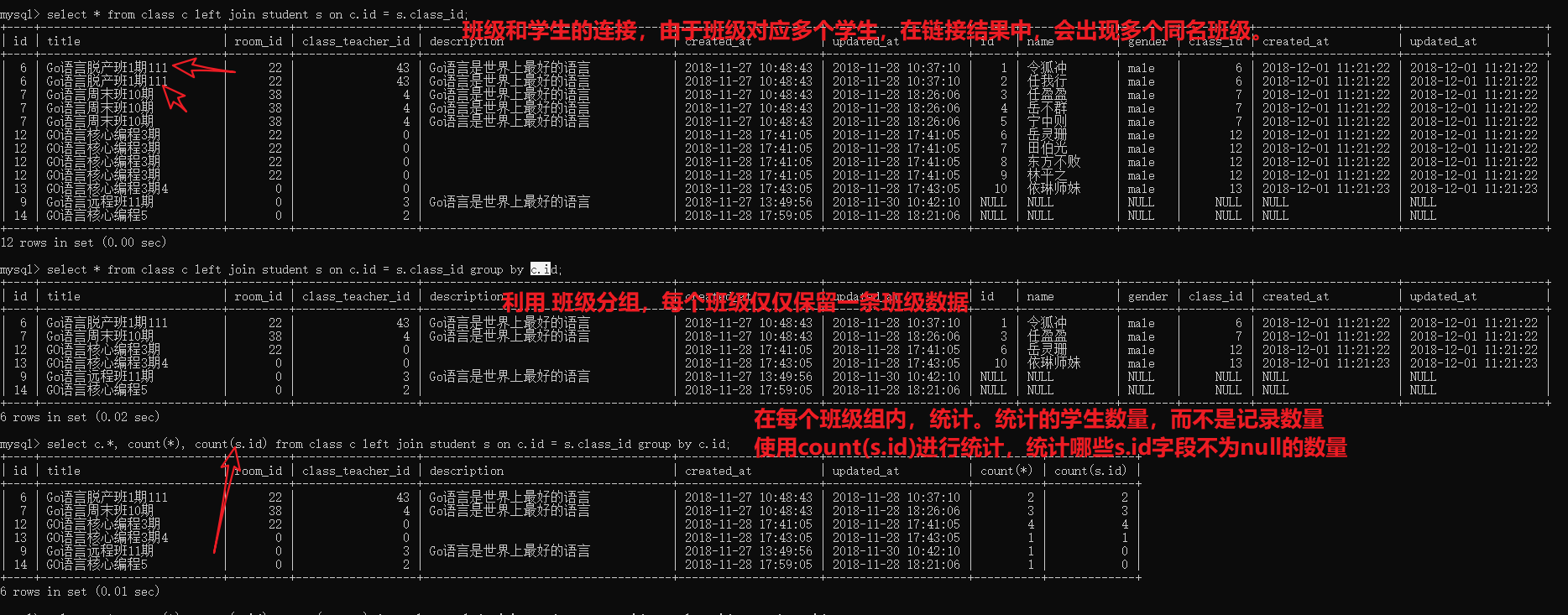


SQL的注意事项：

利用 左外连接（需要保留连接不到学生的班级）

利用 分组

利用 count()合计函数



整理：

班级：班主任，多对一的关系，班级属于某个班主任。班级表增加teacher\_id关联老师。

班级：学生，一对多的关系，班级拥有某些学生。学生表增加class\_id关联班级。

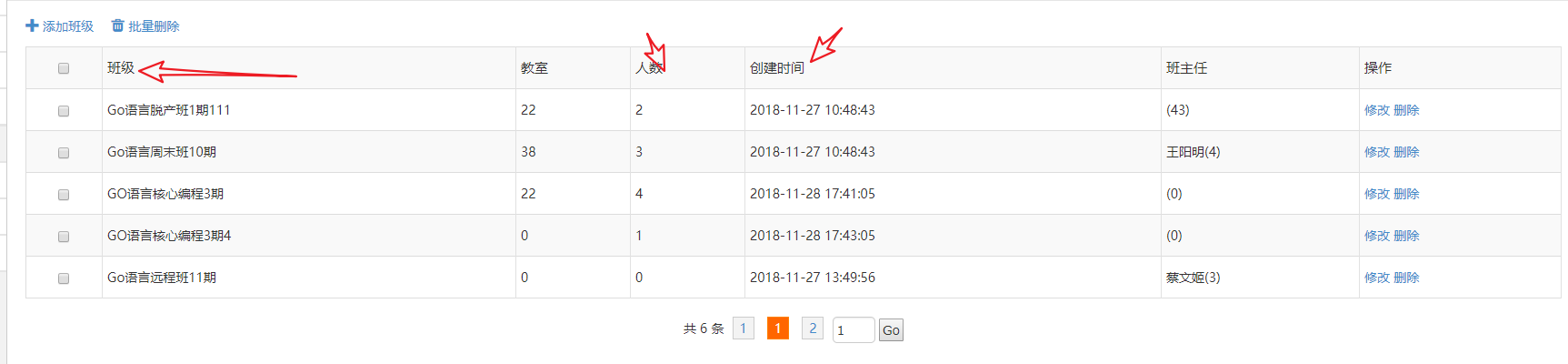
可见：

由多端查找一端（班级找老师），仅仅去连接即可。

由一端找多端（班级找学校），往往需要分组，做分组合计。

# 列表字段排序

实现：点击表头，将列表数据依照当前字段排序。再次点击，逆序排序。



分析：

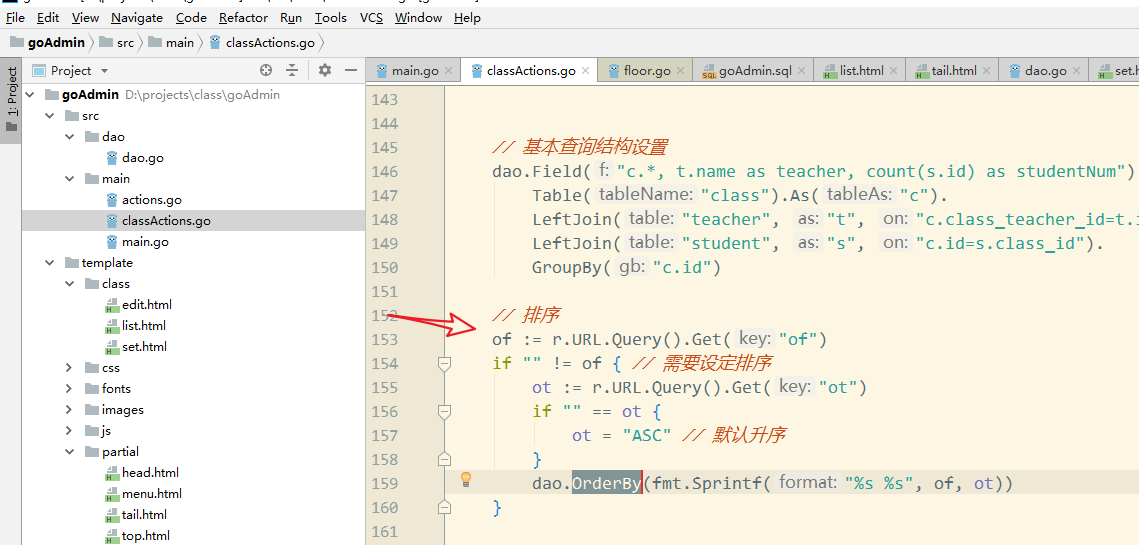
核心，拼凑 order by field ASC|DESC

需要两个参数：字段和排序方式。是由浏览器的请求数据确定的。



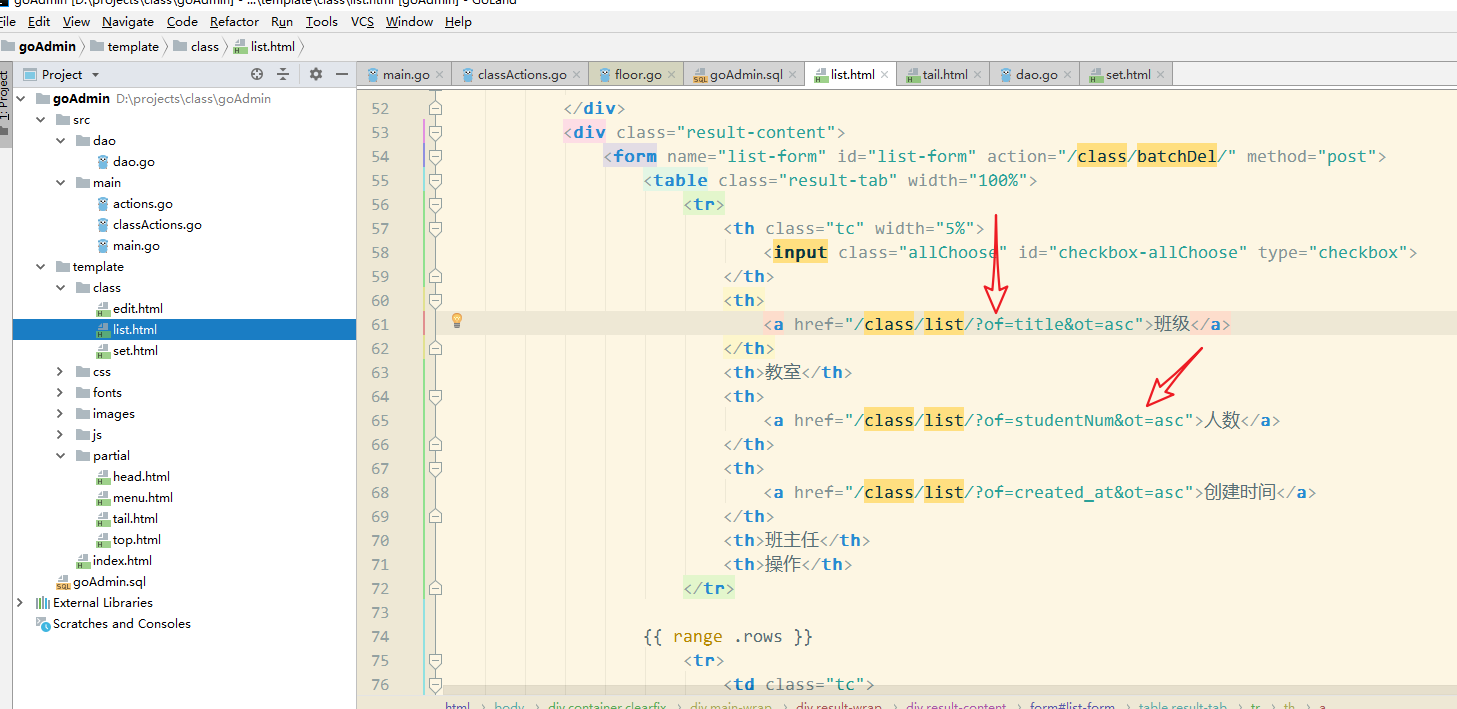
## 更新ClassListAction()

假设可以得到浏览器的请求数据：of, ot,排序字段和排序方式。如果浏览器没有传递该参数，则认为是默认排序。



## 在表头字段上，增加排序参数

list.html

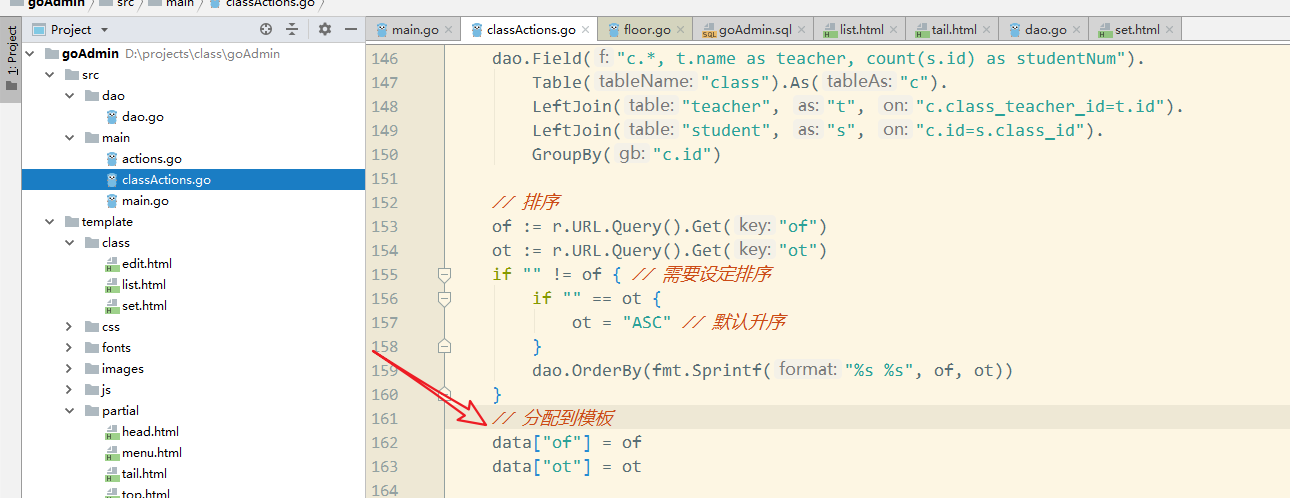


目前，仅仅可以实现依据某个字段升序排序。

需要做到：点击之后，逆序排序呢？

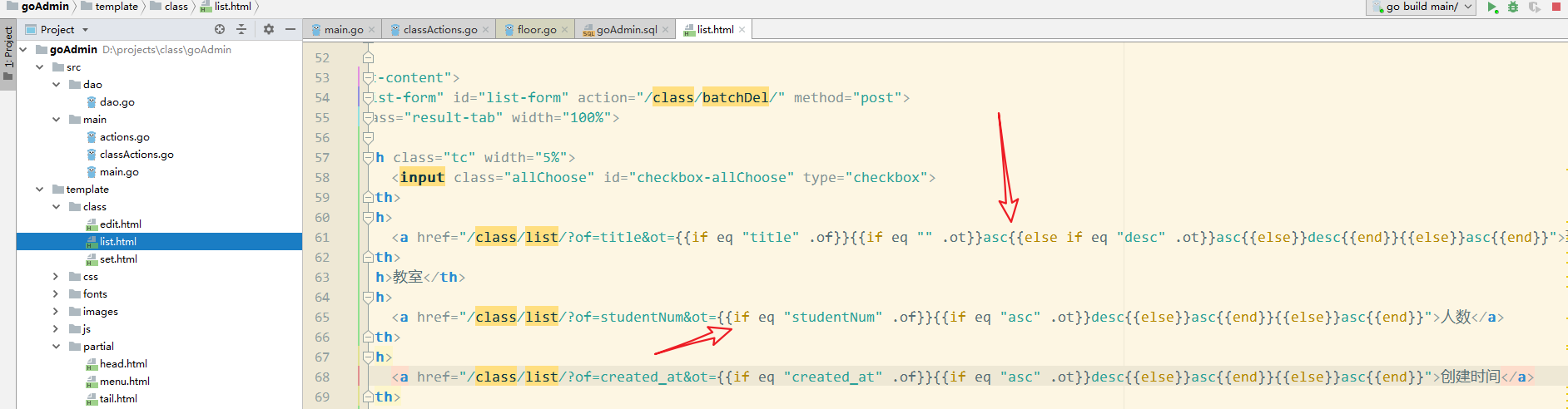
逆转参数ot即可。

将排序信息分配到模板中：



模板中依据当前的排序参数进行判断：

list.html



逻辑：当前排序为该字段的升序时，生成该字段的降序。否则其他情况都生成该字段的升序。

语法：使用 if 结构的嵌套！

伪代码：

if of == “title” && ot==”asc” {

“desc”

} else {

“asc”

}

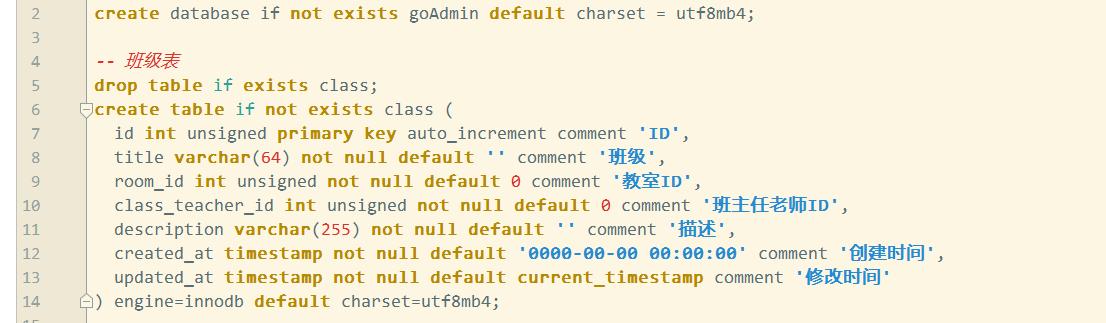
单表的功能完结：



# 班级选课（多对多的操作）

## 设计

班级表



课程表



班级与课程为多对多，如何设计？

增加一张中间表，记录班级和课程的关联关系：

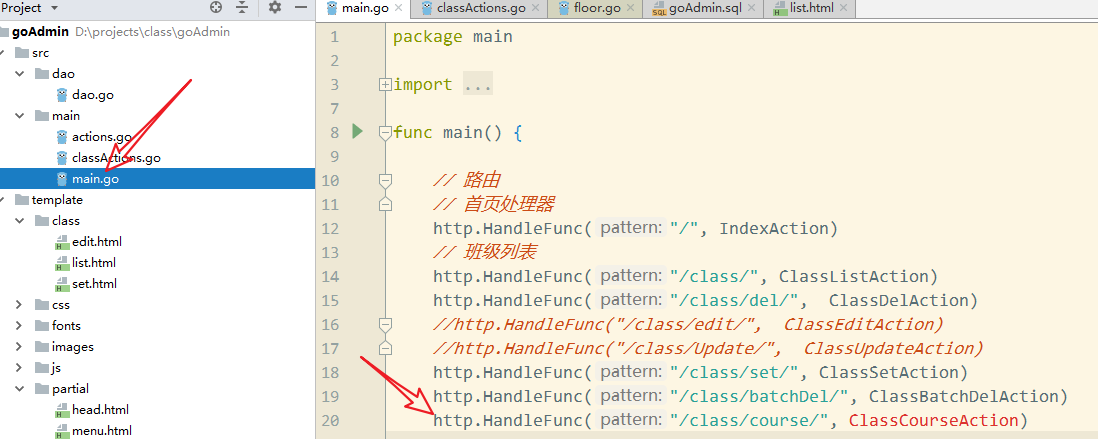


## 班级选课动作

路由：

/class/course/?id=class\_id

id为班级的id



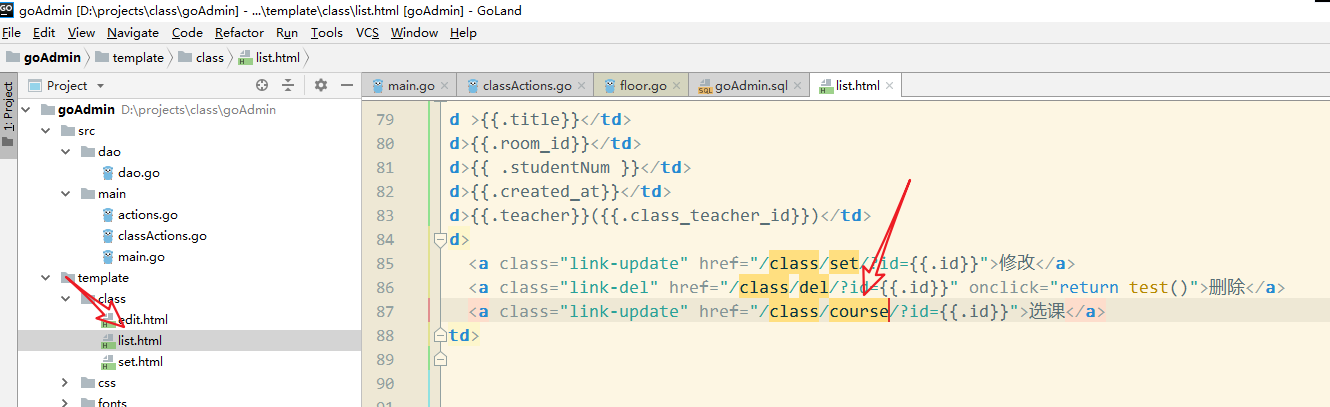
动作：

main/classActions.go



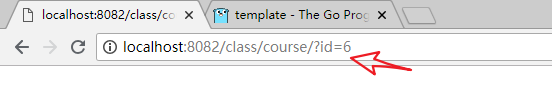
链接：

在班级列表list.html:



效果：





由两个操作构成（类似set）：

1展示全部的课程（将已选的标识出来）

2更新所选的课程

使用 course来实现，通过请求方式 识别区分：

get方式为展示，post方式为更新：

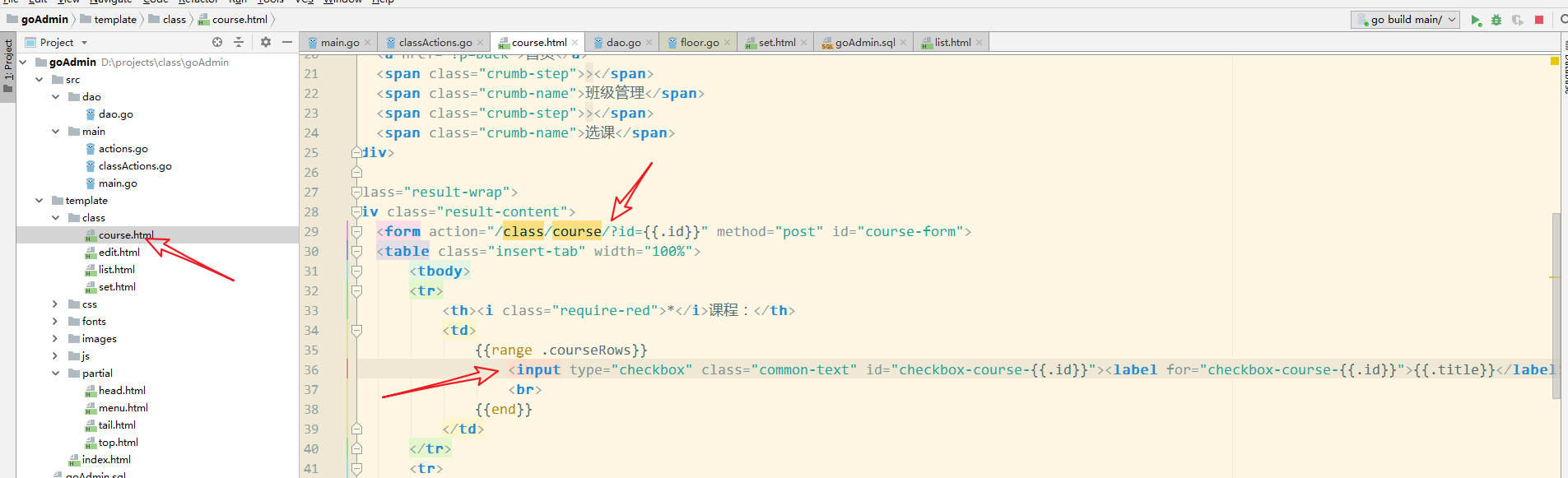
## 展示全部课程（已选课程）

动作：ClassCourseAction





模板：course.html



效果：



## 更新所选课程

在表单页面，选择某些课程。再提交表单，更新class\_course表：

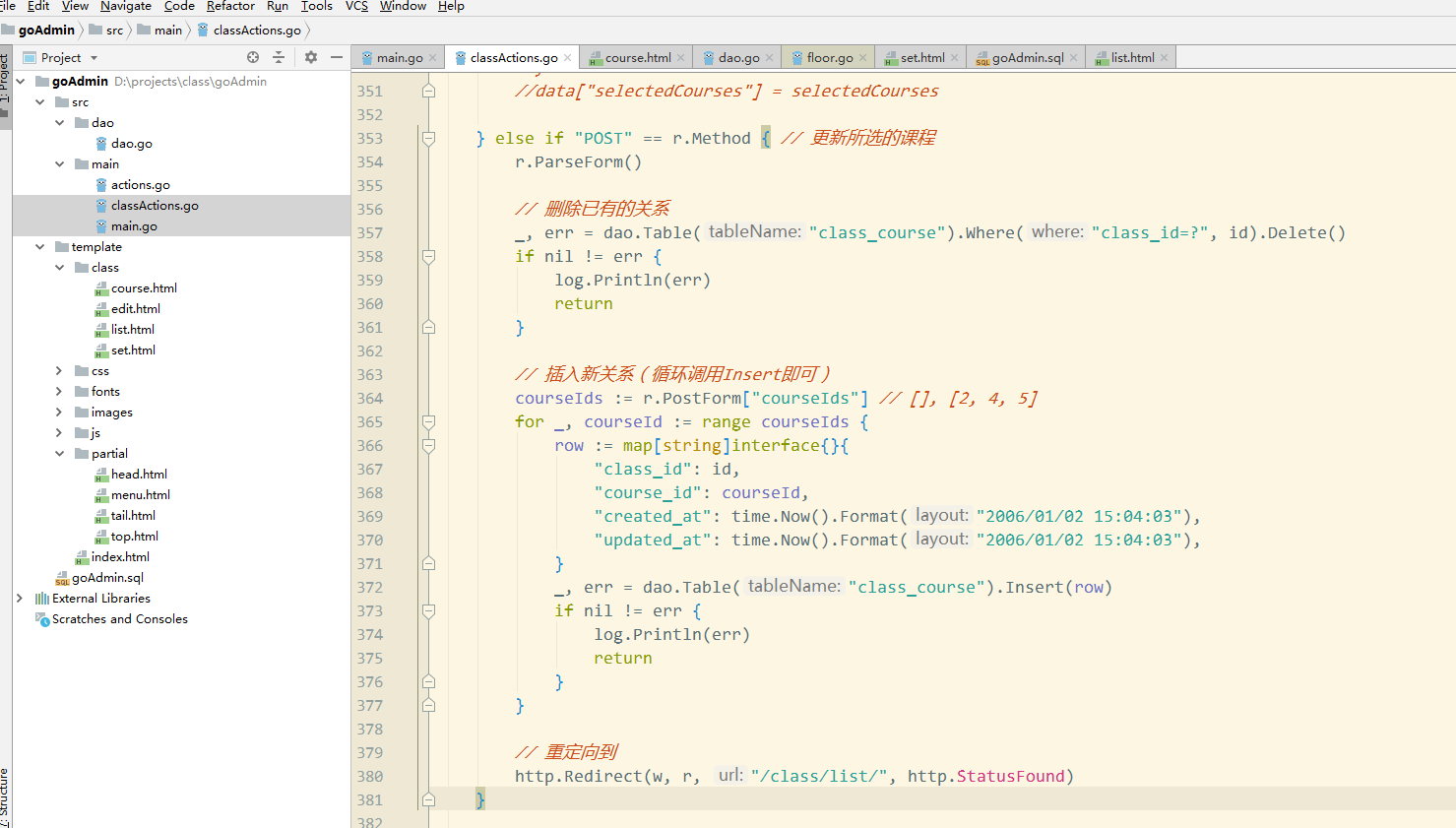
确定checkbox的name和value：course.html



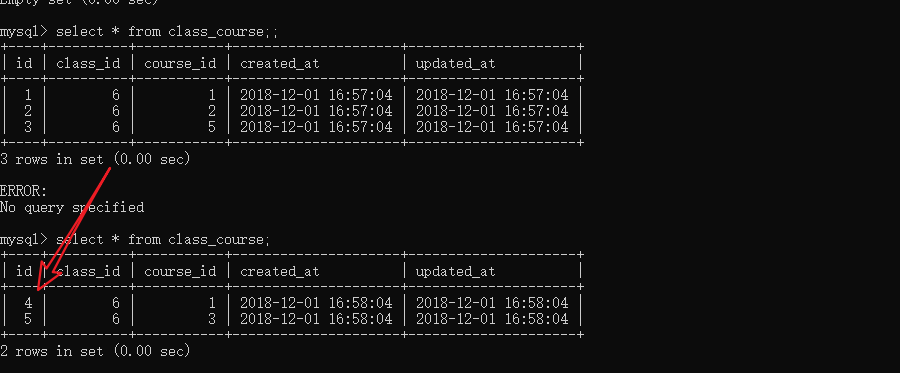
在ClassCourseAction中，完成更新class\_course表：

更新方法：（比较暴力）

删除当前班级与课程已有的关系，再添依据提交数据加新的关系：



由数据测试可以发现成功了：



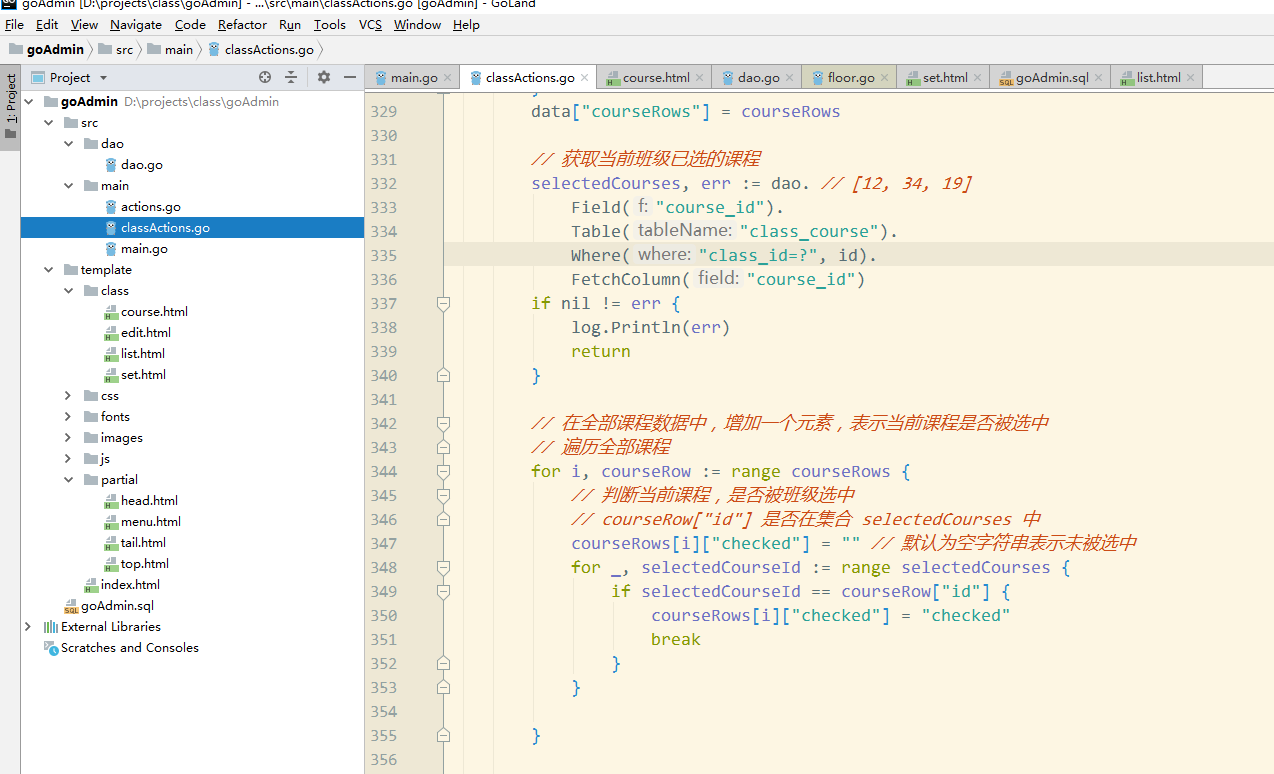
## 已选课程选中

思路：确定哪些课程被选中，在checkbox上使用checked属性即可。

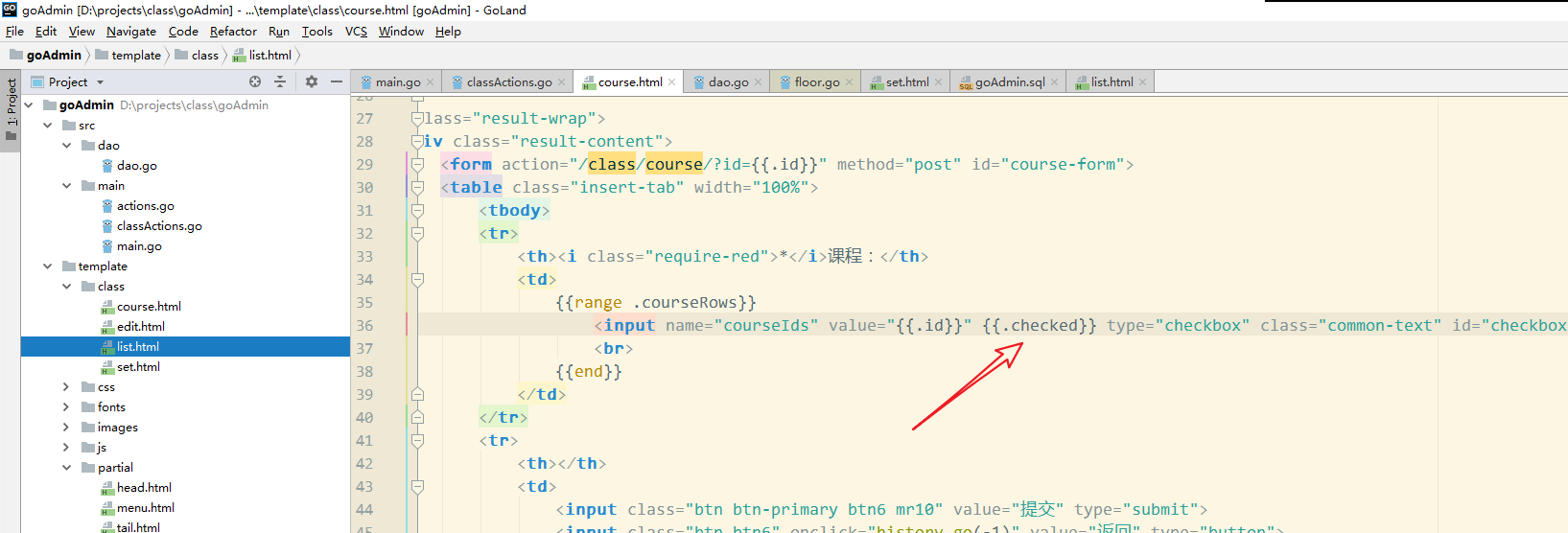
如何确定哪些课程被选中：

获取当前班级所选的全部课程id，判断某个课程是否在这个集合中。

ClassCourseAction()



模板直接输出即可



测试可用。

扩展：班级信息展示：



# 将dao公共初始化

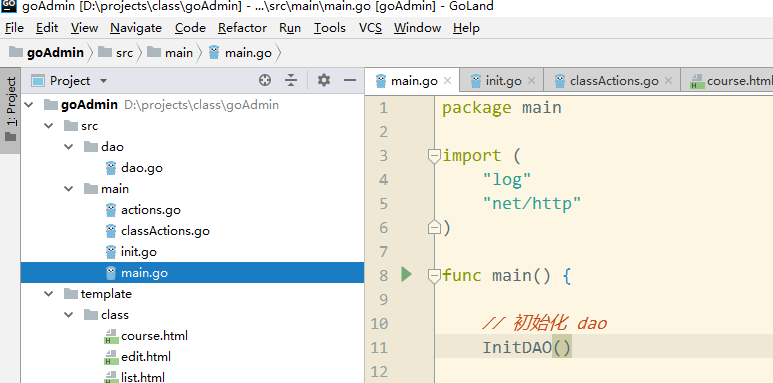
src/main/init.go

InitDAO()



使用全局变量dao存储DAO对象：

main.go中调用



讲Action中的初始化DAO的操作删除。

ClassActions.go



语法中 err变量在后续使用中，需要注意:=的问题。