# Day10

今日内容

1、删除操作

2、全选删除

3、分页

## 删除操作

思路：最终删除数据，肯定执行的是delete from table where id=?,我们可以根据这个进行反向推导，最终知道页面参数是什么。很显然再点击删除链接的时候把主键传递到后台。

Servlet中，获得参数，调用业务层方法。根据返回的结果弹出提示框，由于提示框是客户端执行的，所以，我们最后到列表页，也可以继续通过js过去。

步骤：

1、修改list.jsp的链接



2、编写一个Servlet来处理请求

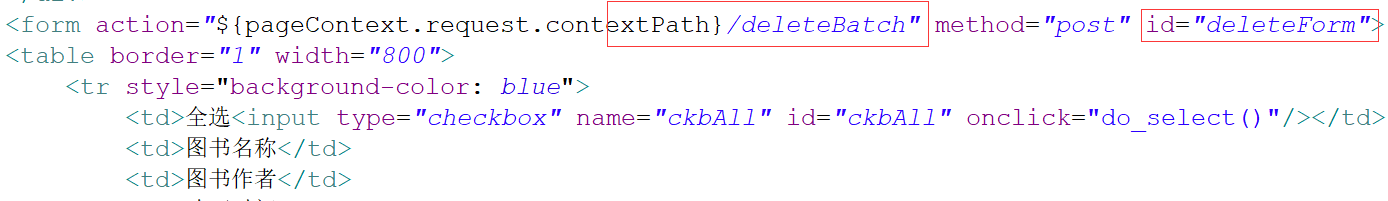


## 批量删除

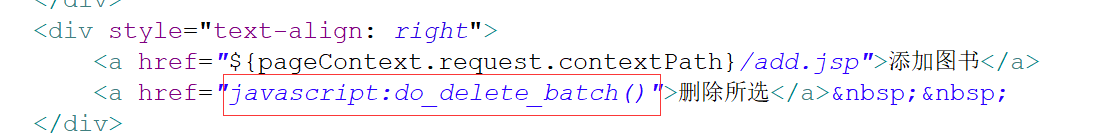
思路：编写js，去处理全选和全不选，以及反选；如果在table的外面包裹一个表单，让表单提交到批量删除的后台Servlet。现在的问题是，该table中没有提交按钮，现在业务需要通过链接去提交表单。我们的思路写一个js，在js中提交表单。当点击批量删除的链接的时候调用该js.

步骤：

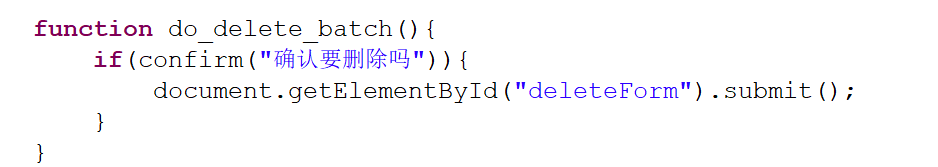
1）前台的页面



2）提交的链接关联js



3）通过js可以将表单提交到后台

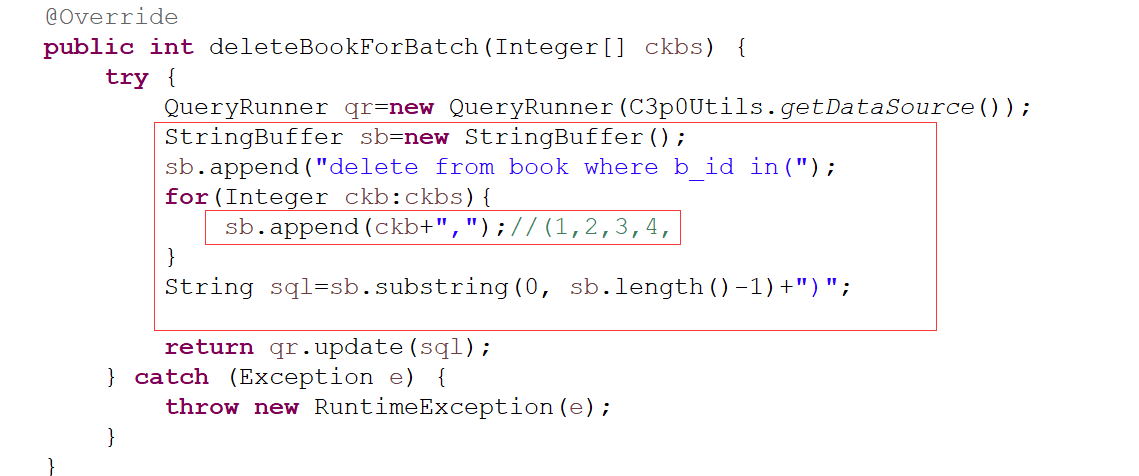


4）后台的Servlet



说明：通过循环构建一个Integer数组，作为参数传递到业务层。

5）数据操作层



## 分页操作

1、为什么需要分页？

数据太多，一页放不下，所以分页。需要什么数据，就查询什么数据，而不是全部查询。

2、分页的设计

问题：如果有102条记录，每页显示10条，一共有多少页？

总条数：totalCount=102

每页显示的条数：pageSize=10

总页数：totalPage=102%10==0?102/10:102/10+1;

totalCount%pageSize==0?totalCount/pageSize:totalCount/pageSize+1;

当前页:pageNo

尾页：就是总页数totalPage

首页:1

上一页：pageNo-1

下一页：pageNo+1

通过上面的分析：

1）计算总条数，肯定是在DAO,因为只有SQL语句才能算出

2）这里的计算关系：pageSize----totalCount-----totalPage

3）显示的页面list.jsp需要显示列表数据list、还需要当前页PageNo以及totalPage，那么反向推导，必须在Servlet中存储这几个数据到域对象。

4）问题：Servlet如何获得以上要存储到域对象的数据？

* pageSize:用户自己定义
* pageNo:第1次是用户赋值，其他时候：pageNo=request.getParameter(“pageNo”);如何判断是否是第1次查询，就看pageNo==null,如果是,赋值为1,否则就是提交的值；
* totalPage:通过公式去计算。也要调用业务层方法获得总条数totalCount
* list:调用业务层方法,传递pageNo和pageSize

5）问题：

当我们已经到了尾页，继续点击下一页，最好是停留在尾页。这意味着，我们必须对pageNo限定查询上限。如果pageNo>totalPage,那么就让pageNo=totalPage;

当我们已经到了第1页，或者是首页，继续点击上一页，最好是停留在首页。这也意味着我们对pageNo要限定下限。如果pageNo<1；那么pageNo=1;

通过以上的分析，很明显在查询list之前，你必须控制pageNo的上下限。意味着必须先计算出总页数totalPage.

所以综合分析：

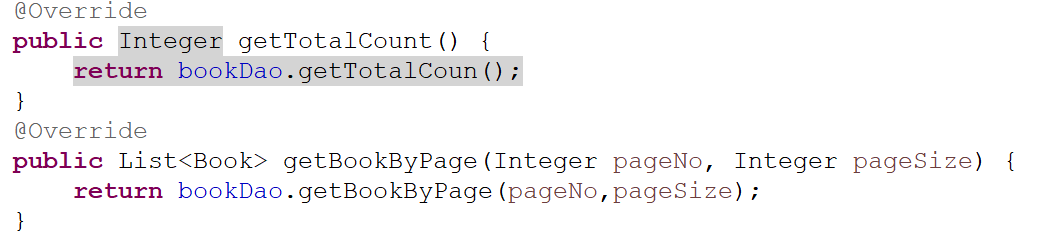
我们知道，有这么一个隐含的线索。pageSize-----totalCount--------totalPage-------pageNo----list.这也是我们的编码的顺序。

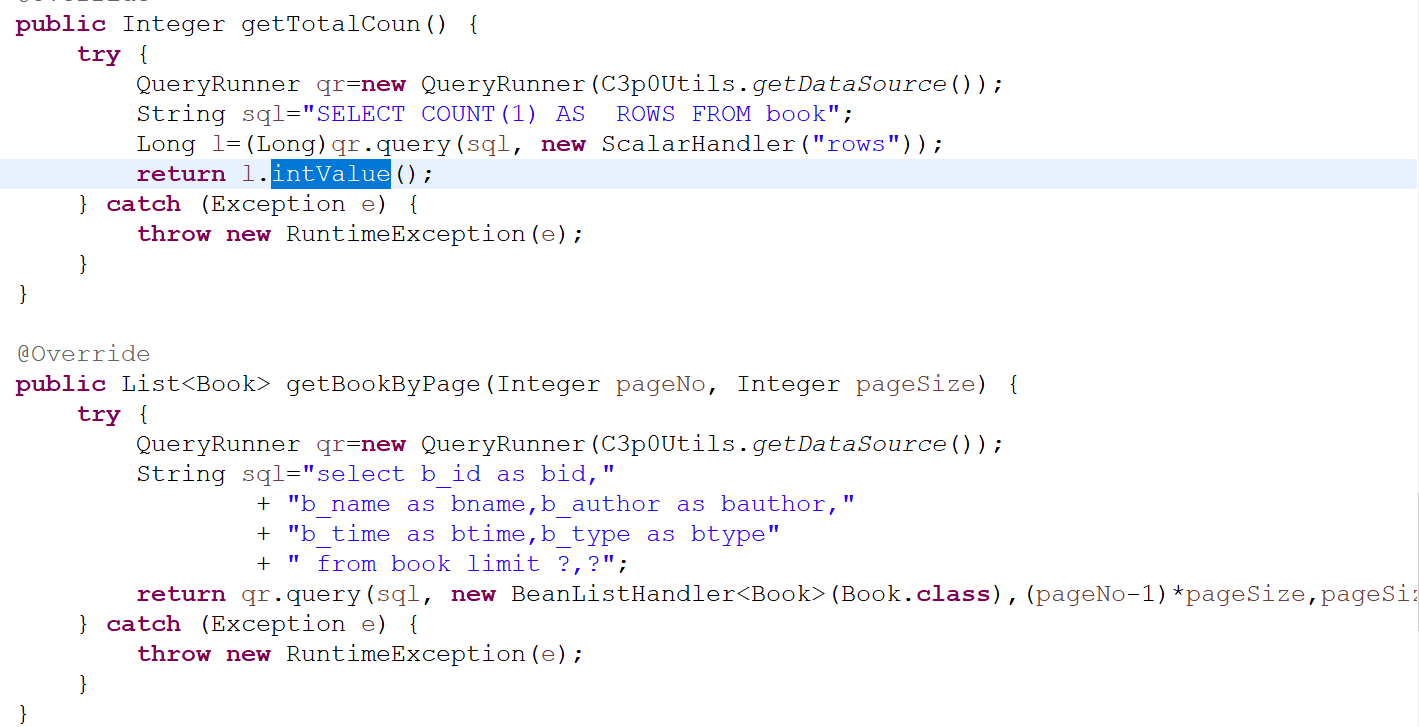
步骤：

1）优化分页的Servlet



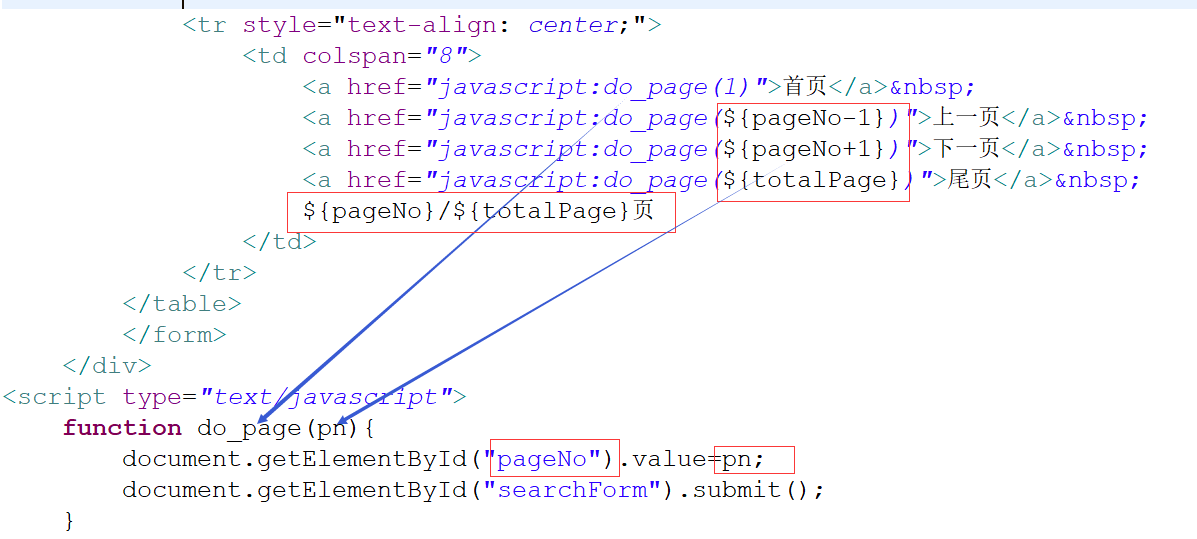
2）优化Service和DAO





3）前台页面绑定数据

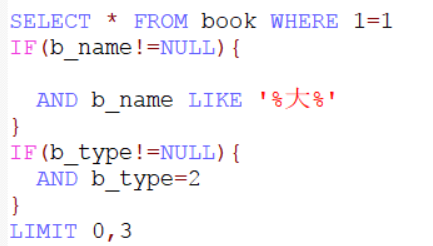




## 分页升级

业务：需要带多个条件的查询分页

分析：可以在DAO层查询的时候通过拼凑SQl语句完成，以下是伪代码



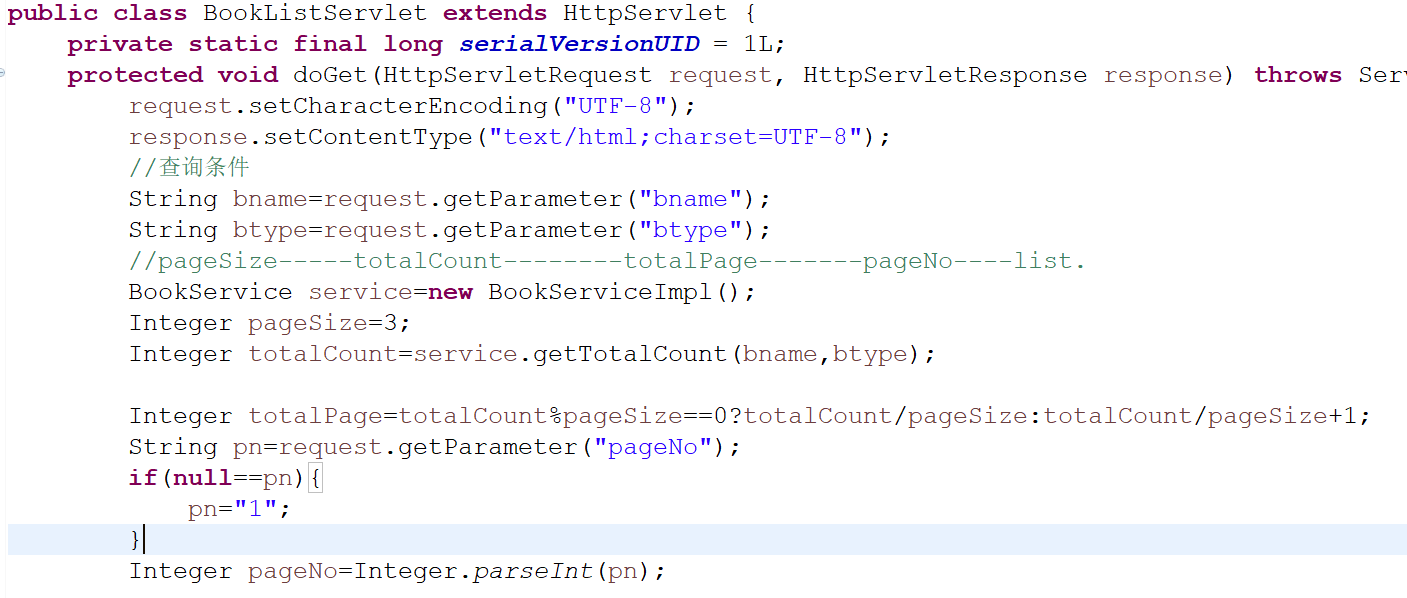
思路：由于带条件，我们可以通过一种技巧，就是拼凑SQL语句，在原来查询语句的基础上先写条件 where 0=0.然后根据输入的值是否为空，去拼凑。

这意味着：查询list的方法和总条数的方法需要把条件带入。这样才可以带入到业务层，然后通过业务层带入DAO层。这样我们就可以用于拼凑语句了。

由于在查询页经过Servlet后，又重新到了查询页，一般来说，用户希望上次填写的条件仍然保存。这就是所谓的数据回显。根据我们之前讲解的理论，最好把回显条件新存储在域对象。然后再重新到页面，后面只是一个绑定的问题。

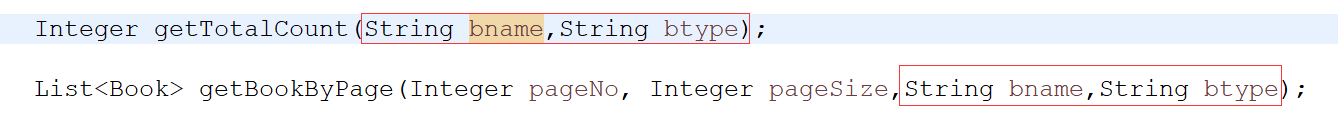
步骤：(反向推导的)

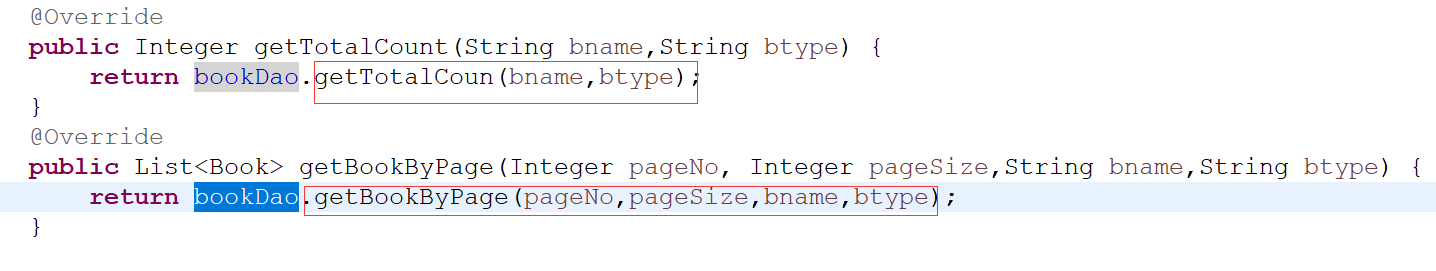
1）优化后台的Servlet



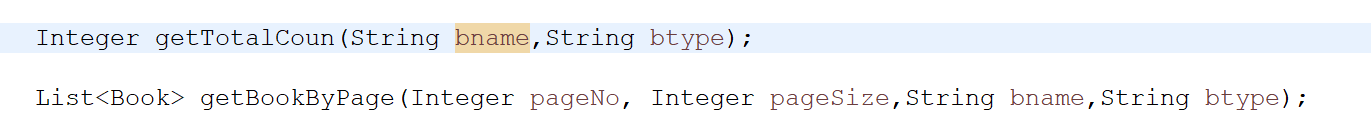


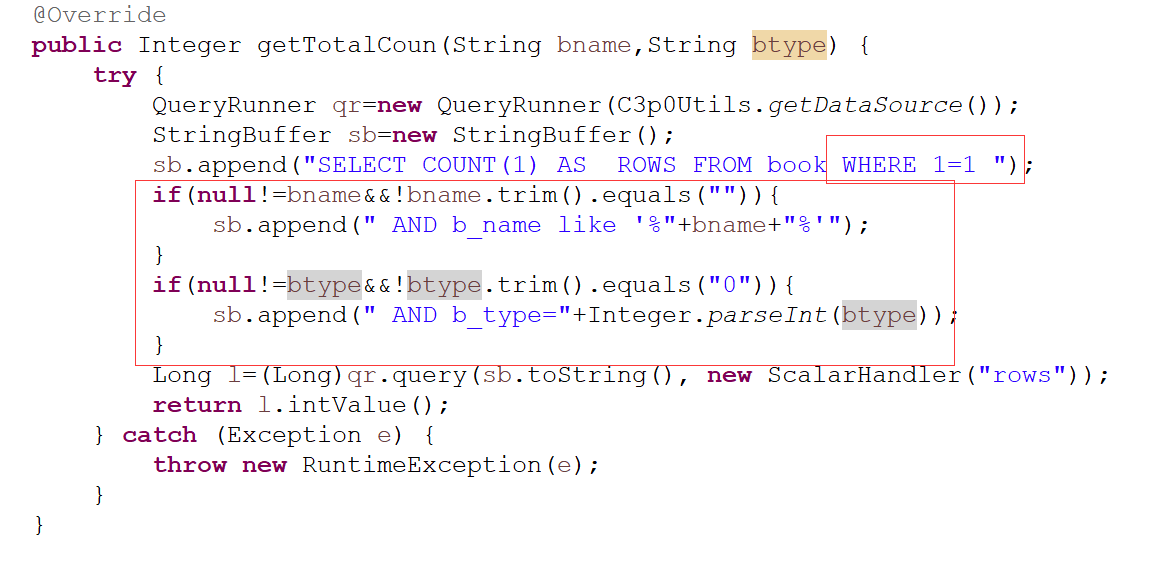
2）优化Servlet的接口和实现类

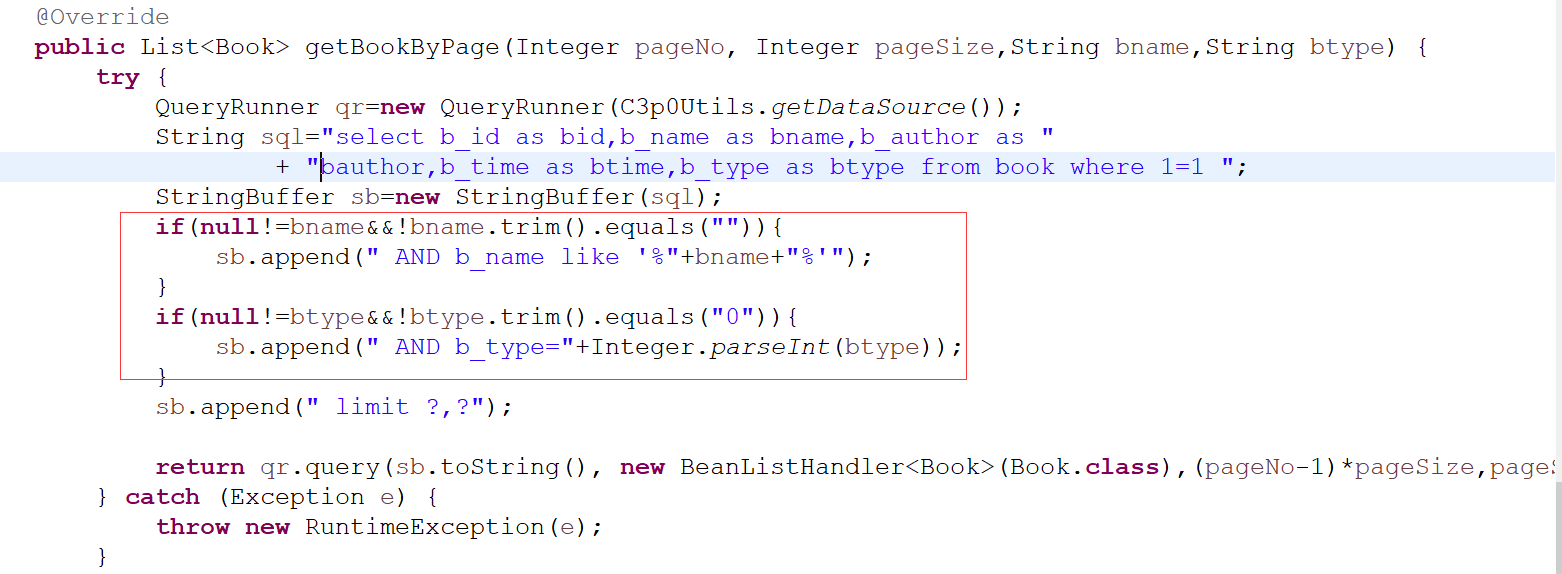




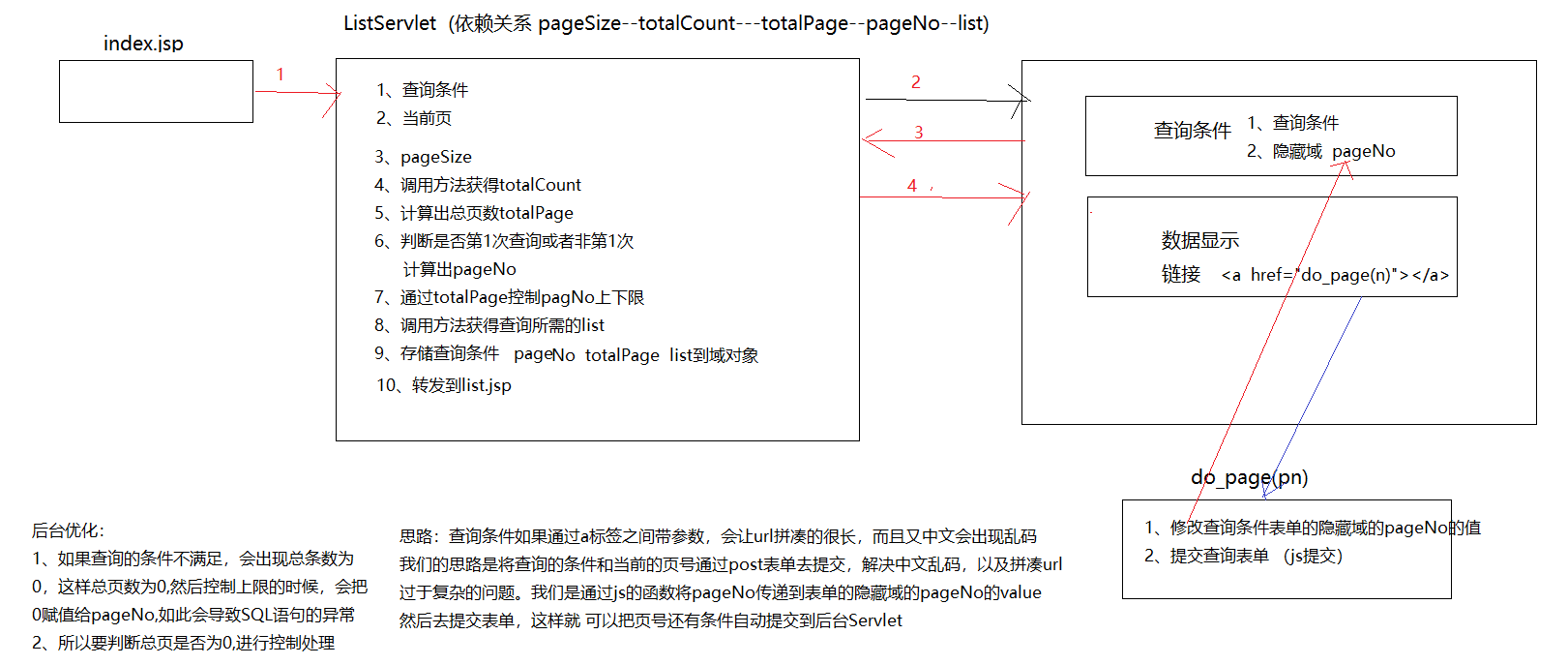
3）优化DAO的接口和实现类





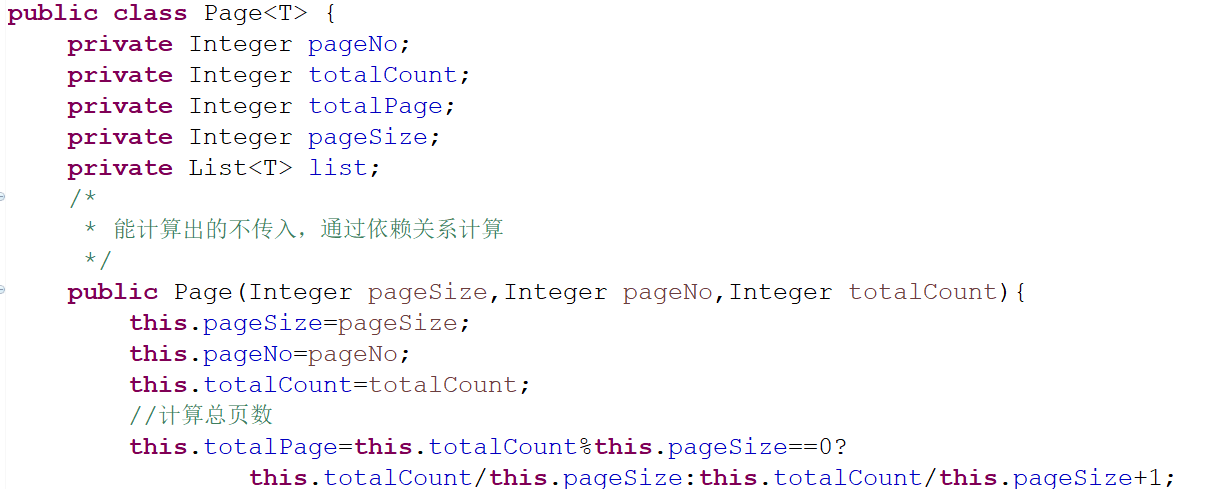


## 分页的思路总结

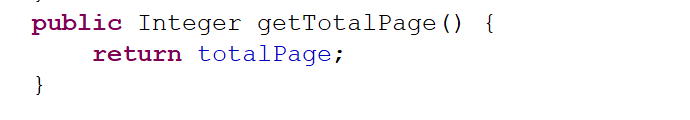


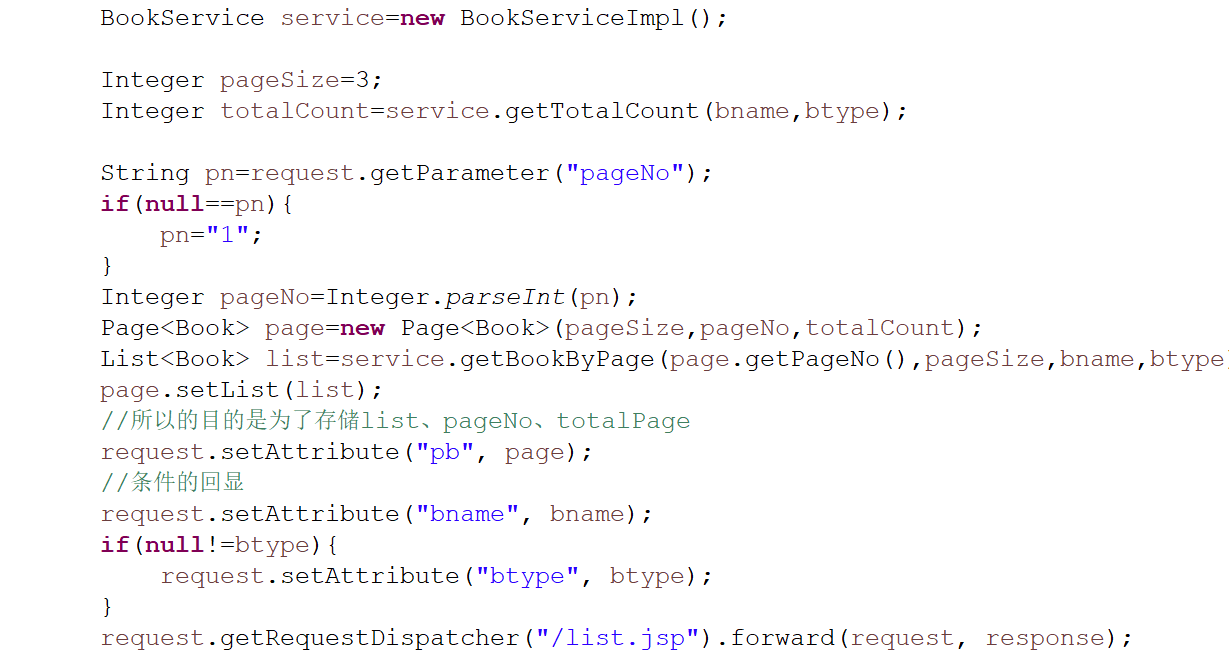
## 分页优化

编写工具类Page，来优化原来的代码

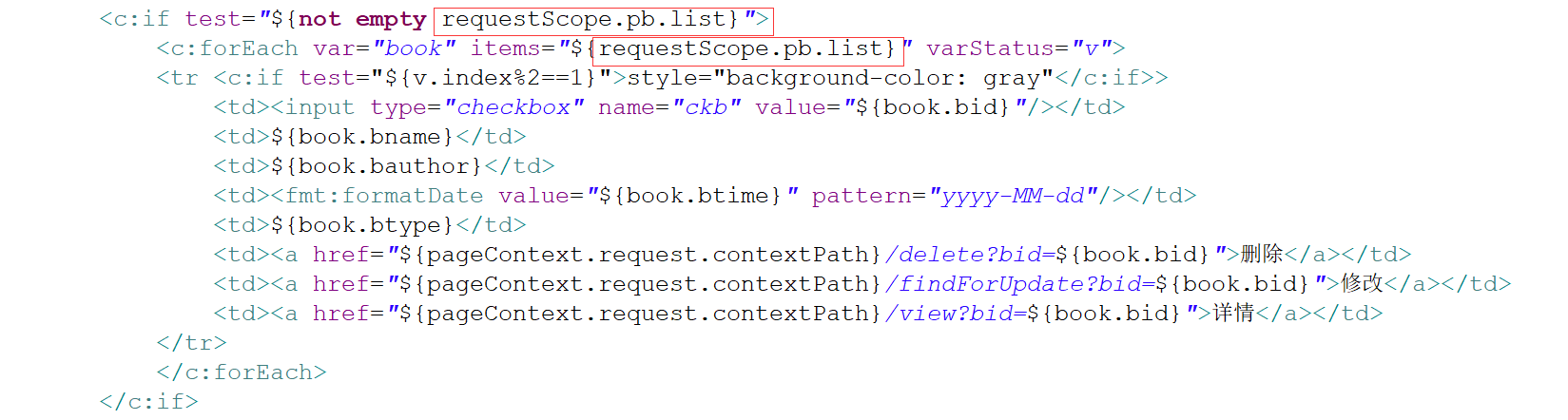


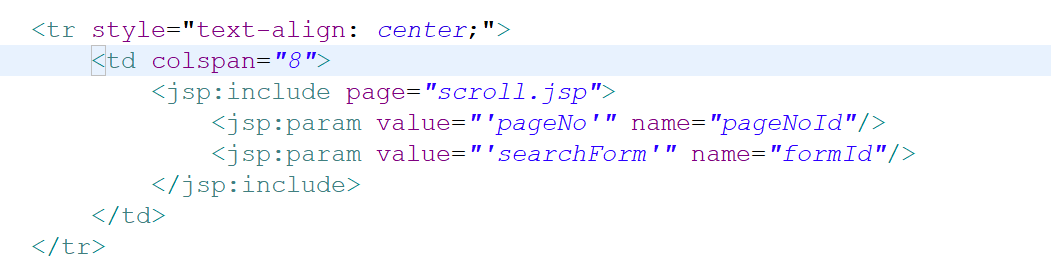




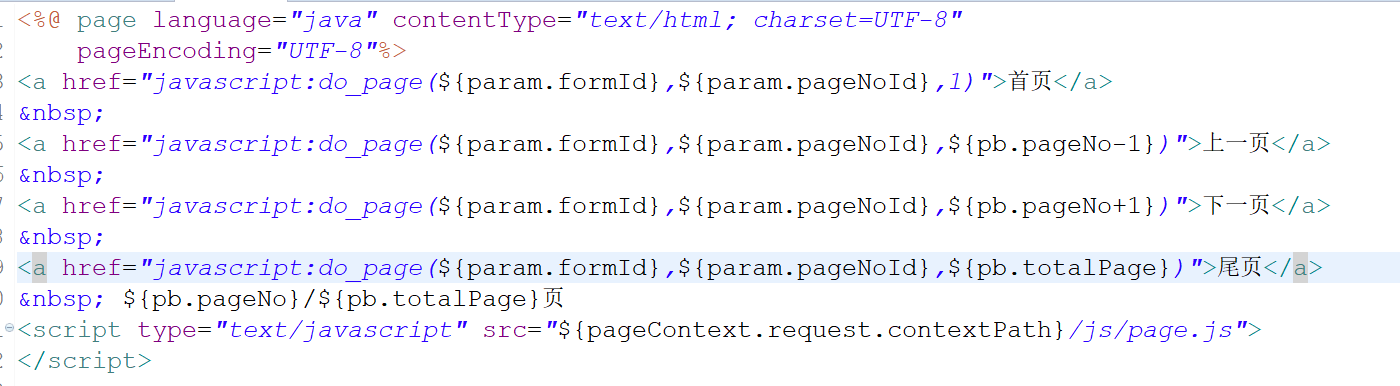


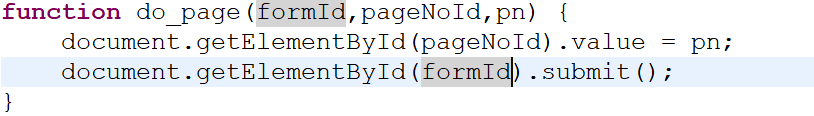
页面部分：

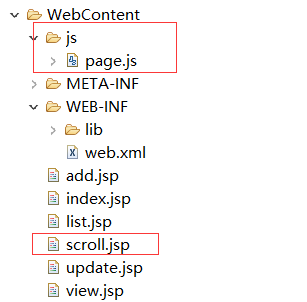




Scroll.jsp







总结：

作业：

1、批量的删除操作（必做）

2、带条件的分页（必做）