# 教研端1.1文档

## 1.改动

在article的views中article\_list方法里面，第15行增加了web\_source，这个是网站来源的搜索，比如ChinaPlus、Chinadaily，它的搜索实现是在第84行到第98行，用web\_source关键词去检索数据库的web\_source字段；

然后是article\_update方法，这个方法里去掉了上一个版本对题目库去重的方法，直接通过id查找到需要编辑的文章，编辑好之后，直接插入到题目库，题目库的id是随机生成的。

然后增加了question这个文件夹，这个是题目库的展示，它的url连接在article-list中右上角有一个题目文章列表，它主要的逻辑实现都在这个文件夹下views里面，question\_list、question\_detail、question\_delete都是和素材库的实现逻辑是一样的，不同的是question\_update,这里面我会对题目文章去重，在第176行，获取到这篇题目文章的id，然后201行插入的时候用的就是这个id，这样就会去重。

## 2.程序说明

### 2.1 素材端

素材端文章的代码在项目的article目录下，其中有两个重要的py文件，一个是前后端连接的urls.py文件，一个是后台逻辑实现的views.py文件。

#### urls.py

它的作用就是连接前后端，前端发送的具体请求，例如删除文章，更新文章，然后先到这个文件，通过这个文件去views.py里面匹配具体功能实现。

#### views.py

这个文件下有四个功能，分别是文章列表页article\_list,文章详情页article\_detail，文章删除article\_delete，文章编辑article\_update。

首先是文章列表页article\_list，这是网站的主页面，会展示文章的目录，然后在这个文章的目录里，可以进行搜索，如代码的11到15行，search的意思是搜索文章的关键词，source是网站文章的来源，列如新华日报，中国日报等，tag是文章的类型，例如社会类，创新类等，web\_source是文章的网站来源，目前来源有三个，ChinaDaily，ChinaPlus，51English。这些搜索词会去数据库里进行匹配，然后返回给前端，并且限定每页10篇文章。

第二是文章的详情页面，article\_detail，当点击某一篇文章的标题，它会跳转到具体文章，后台逻辑是先获取到这个文章的url，然后去数据库查找，并且返回它的标题，内容，tag标签，和超纲词列表，然后返回到前端展示并对超纲词进行标红处理。

第三是文章的删除，article\_delete，文章详情页左上方会有删除的按钮，点击删除时，后台获取文章id，然后去数据库删除。

第四是文章的编辑页面，article\_update，首先获取需要编辑文章的id，然后去数据库超找到这篇文章，当用户修改完点击完成按钮时，后台会获取post内容的标题，文章内容，并把修改之后的内容插入到题目库当中，id是随机生成的，没有经过去重处理，即一篇文章编辑多次会在题目库生成多个题目。

### 题目端

题目端的代码在项目的question目录下，其中有两个重要的py文件，一个是前后端连接的urls.py文件，一个是后台逻辑实现的views.py文件。

#### 1.urls.py

它的作用就是连接前后端，前端发送的具体请求，例如删除文章，更新文章，然后先到这个文件，通过这个文件去views.py里面匹配具体功能实现。

#### 2.views.py

这个文件下有四个功能，分别是题目列表页question\_list, 题目详情页question\_detail，题目删除question\_delete，题目编辑question\_update。

题目的列表页，详情页，删除和上述素材的后台逻辑是一样的， 不同之处在于，第一题目的详情页面没有超纲词检测，直接返回的题目库的文章，第二question\_update方法里面会对题目做去重处理，具体实现是首先获取需要编辑题目的id，然后当用户编辑完点击完成时会那把修改的文章替换成修改前的文章，完成去重。

### 数据库端

ES\_models文件夹下面是数据库建库的代码，es\_material是素材库，es\_subject是题目库。

### 算法端

主要代码在filter里面，其中file.py文件是把爬虫库的文章通过超纲词检测算法，检测到超纲词以后，放入到素材库，目前爬虫库有三个数据库，对应三个网站，素材库是一个库，有一个web\_source字段，表明了来源于爬虫库的哪一个库。

### 程序运行

程序的运行流程，首先需要运行filter文件夹下的file文件，生成带超纲词的素材库，然后运行Django项目，运行python manage.py runserver 8080