2019 年夏季学期　程序设计训练
第三周大作业报告

魏彤

计 86 班　2018011417

2019 年 9 月

摘要

本次大作业基于 Python 和 Django 框架实现了 NBA 新闻的整合和检索系统，用户可以通过浏览器访问服务器。服务器提供的功能包括：键入关键词后对于数据库中的新闻进行内容检索；按照时间顺序查看数据库中的新闻；查看球队热度榜；对球队的主页进行访问；对数据库进行管理和新闻的实时更新。

1　前端界面设计

前端界面设计利用了 HTML 模板，并使用了 Bootstrap，以实现更好的视觉设计和功能。

1.1　搜索主页

搜索主页包括上方的导航栏，标题和搜索框。导航栏的几个入口分别对应着搜索主页、新闻列表、球队热度榜、数据库管理页面和网站介绍。

搜索栏和搜索按钮共同组成一个 HTML 表单，当用户输入搜索内容并点击按钮后，会将表单以 GET 的方式提交给显示搜索结果的网页模板。（参见**图 1**）

1.2　搜索结果页面

通过后端传入参数，搜索页面模板展示了搜索的结果数目、搜索时间和搜索网页结果。每一条结果均包含一个超链接，可以打开新闻内容页。在搜索结果页面中，每一条新闻的标题和包含关键词的正文片段会被展示出来，并且其中包含关键词的部分会被红色高亮。网站支持多关键词搜索，网页按照与关键词的契合程度由高到地排序。

当新闻较多时，网页会进行分页，每页显示 12 条新闻。最下方的分页按钮为上一页和下一页的 URL 连接。（参见**图 2**）

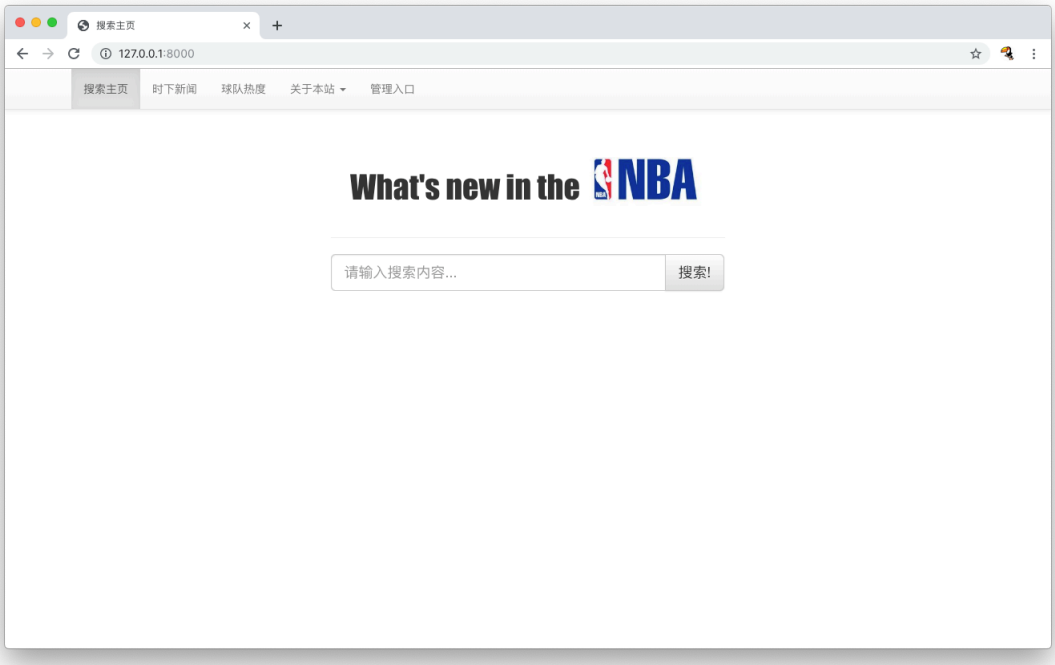


图 1: 搜索主页

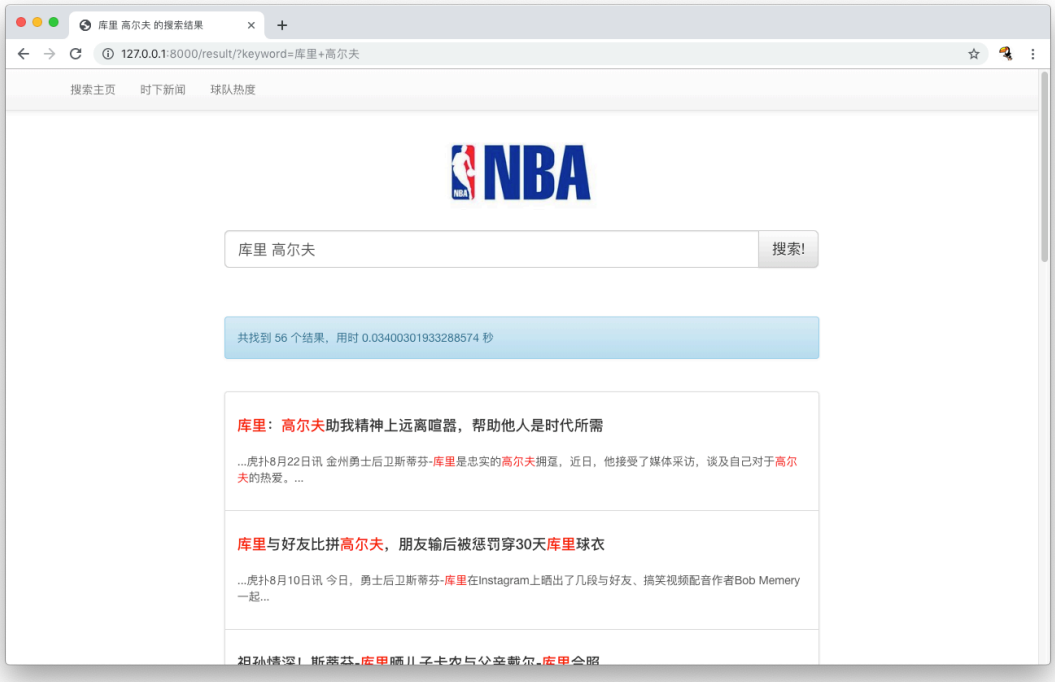


图 2: 搜索结果页面

1.3 新闻内容页面

新闻内容页面包括了每一条新闻的标题、发布时间、新闻来源和新闻正文。其中，新闻正文内所有的球队和球员名称均设置了超链接，可以打开球队/球员所属球队的主页页面。（参见**图 3**）

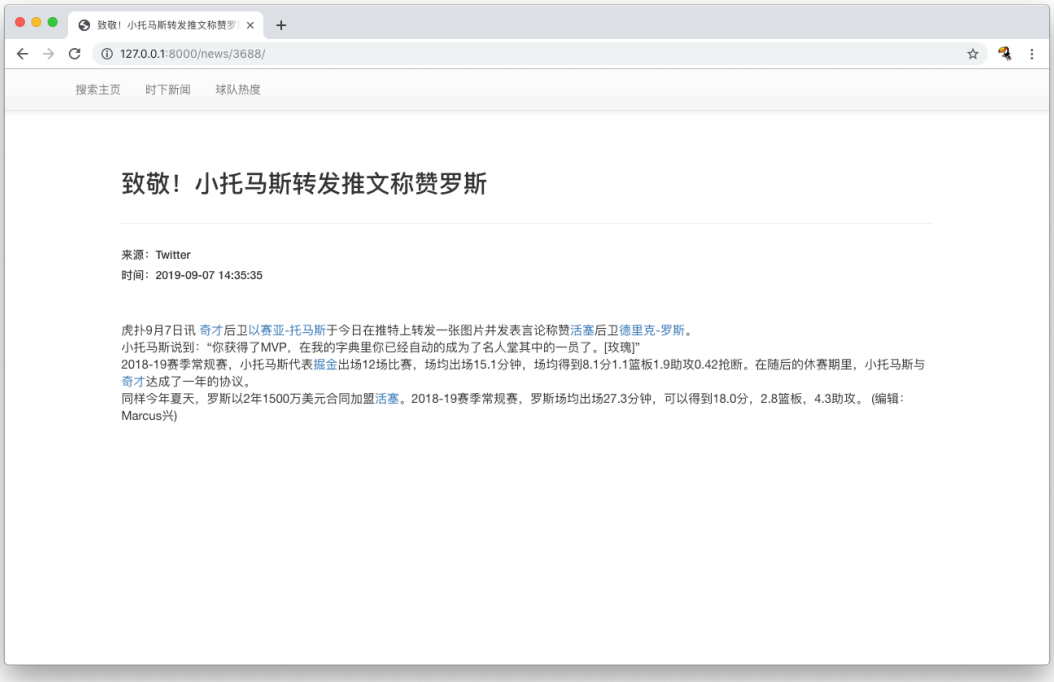


图 3: 新闻内容页面

1.4 新闻列表

新闻列表页面展示了数据库中的全部新闻，按照发布时间的顺序从新到旧进行排列。每一条新闻展示新闻的标题。（参见**图 4**）

1.5 球队热度榜

球队热度榜展示了一个表格，所有的球队按照数据库中含有球队名称的相关新闻的数量，从多到少进行排序。球队的名称制作了超链接，可以打开球队主页页面。（参见**图 5**）

1.6 球队主页

球队主页页面包括了球队的名称、所在城市、加入 NBA 时间的信息和球队现有阵容名单。在下方展示了标题和正文中包含球队名称的相关新闻。同样地，当新闻过多时，会和搜索界面一样进行分页浏览。（参见**图 6**）

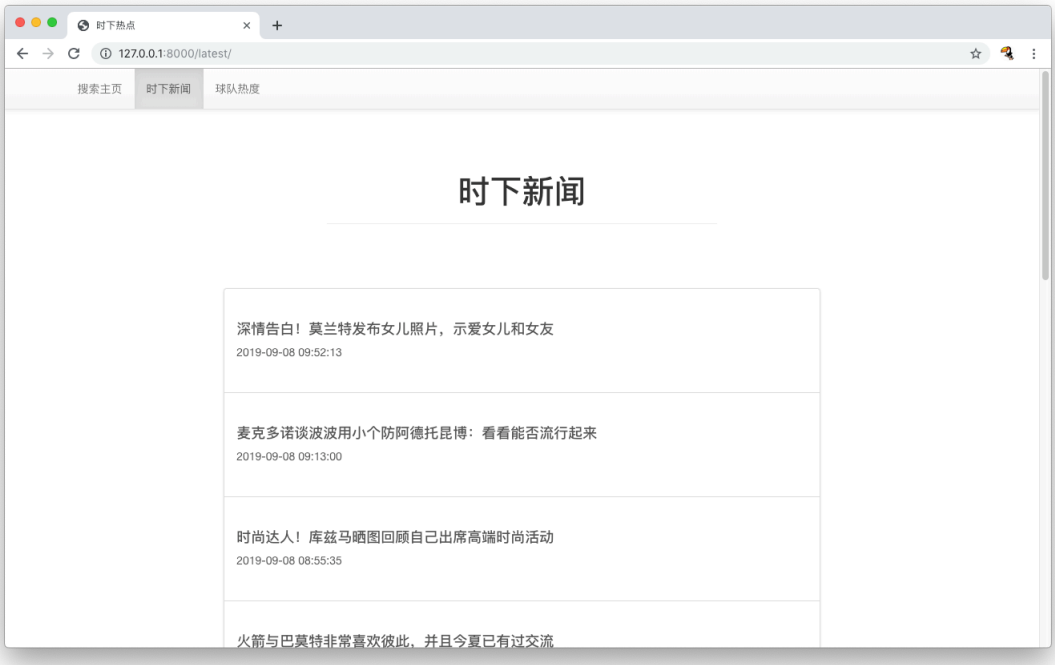


图 4: 新闻列表

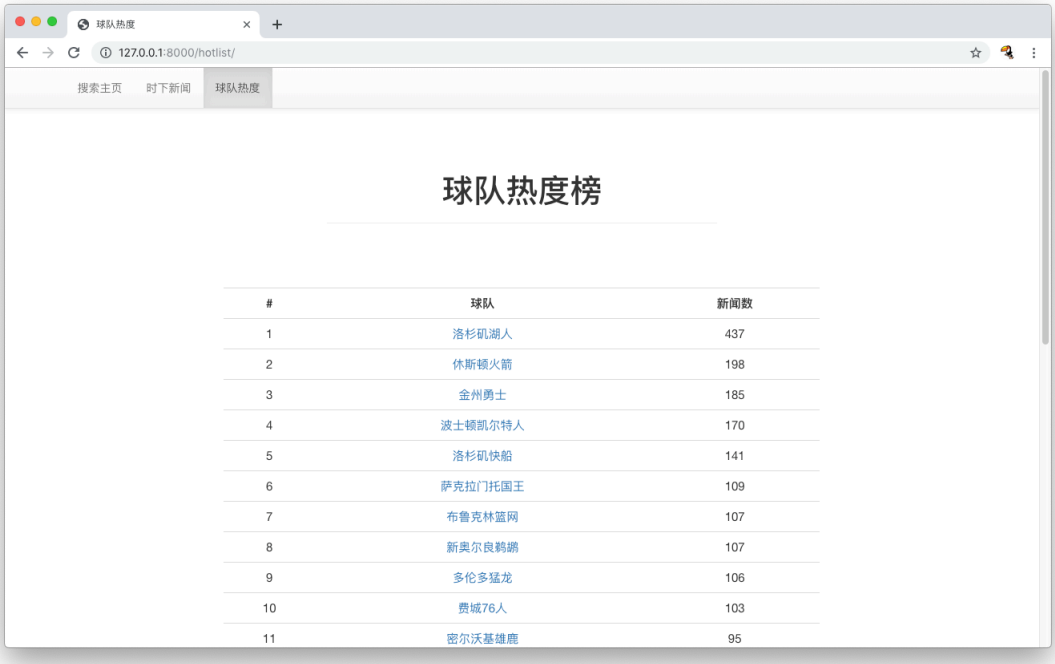
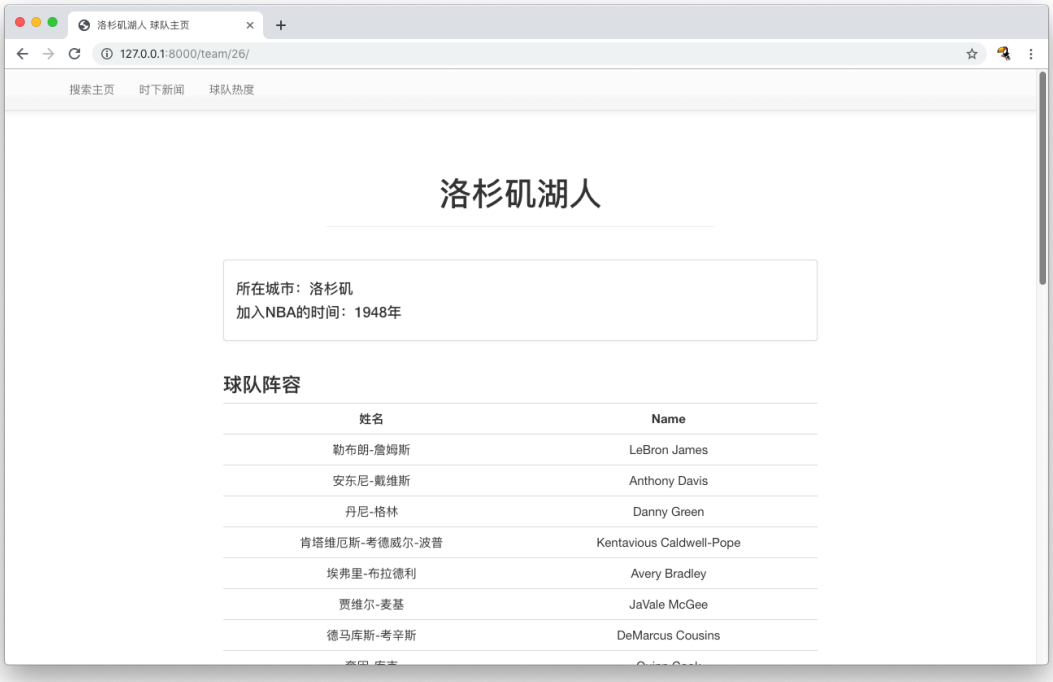
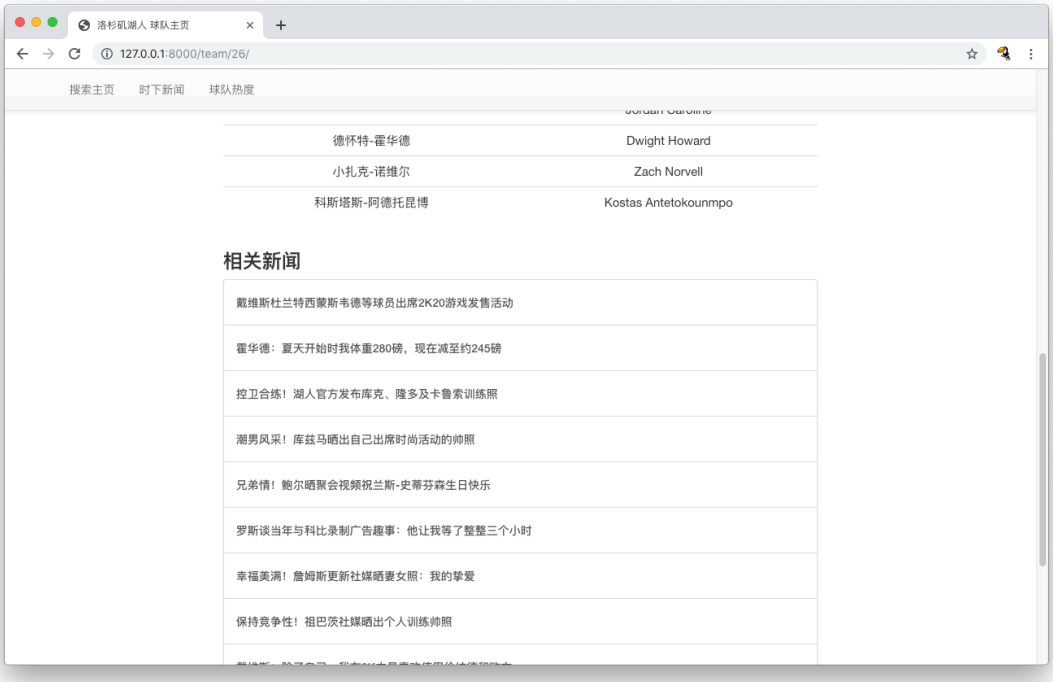


图 5: 球队热度榜



(a) 球队信息和阵容



(b) 球队相关新闻

图 6: 球队主页

1.7 数据库管理和新闻的实时更新

数据库管理页面在 Django 的 admin 页面的基础上进行了修改。用户可以通过命令行创建管理员账号，当使用管理员账号登陆管理页面后，可以对于数据库中的新闻、球员和球队信息进行查看、增删和修改。

此外，在页面的右上角添加了一个按钮，用来进行实时新闻更新的爬虫操作。点击“开始爬虫”按钮后，系统在后台开始爬虫程序，从网站中获取新闻并更新数据库和索引，之后在管理页面中弹出“爬虫完成”对话框。

在系统爬虫的过程中，管理员用户可以点击“停止爬虫”按钮，停止爬虫操作。已经被获取的新闻会被加入数据库和索引，之后在管理页面中弹出“停止爬虫”的对话框。（参见**图 7**、**图 8**）

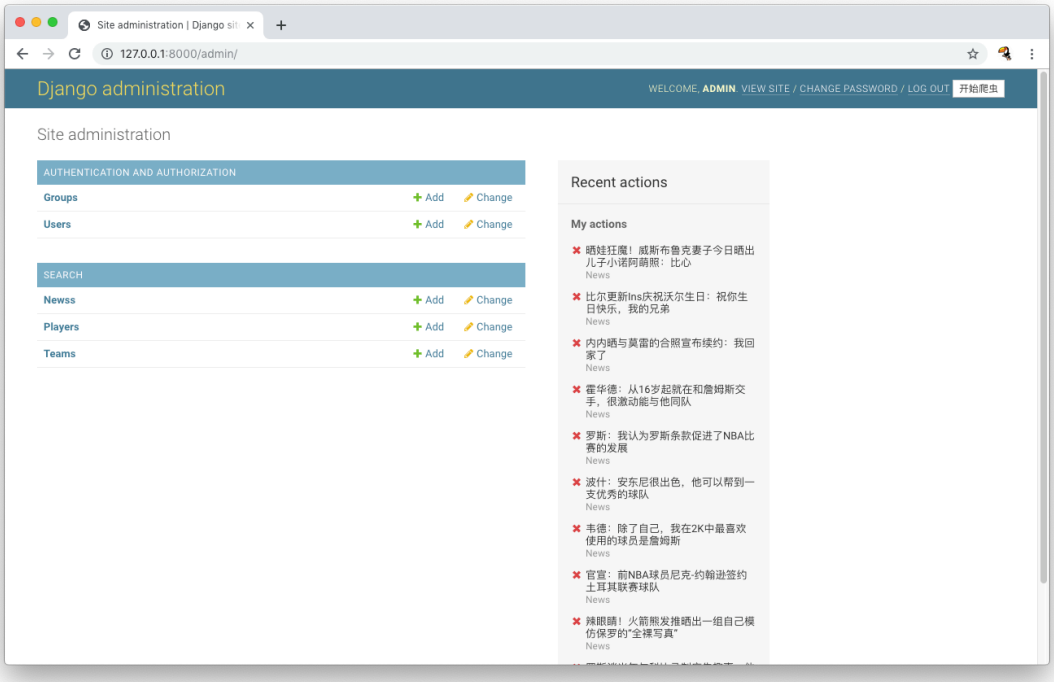


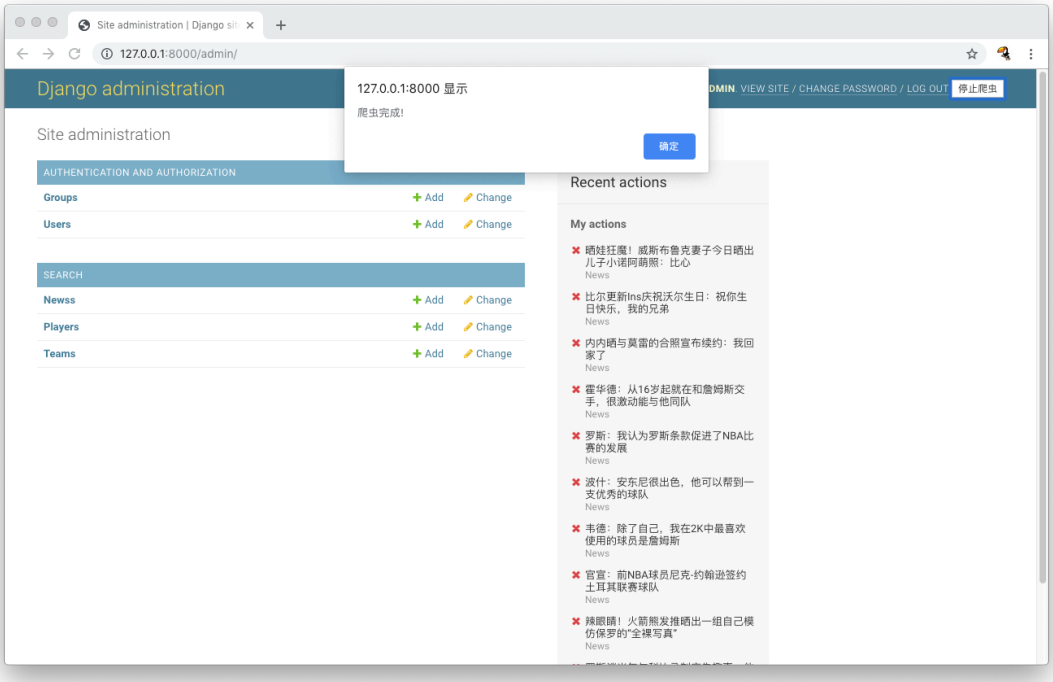
图 7: 管理页面

2 后端架构设计

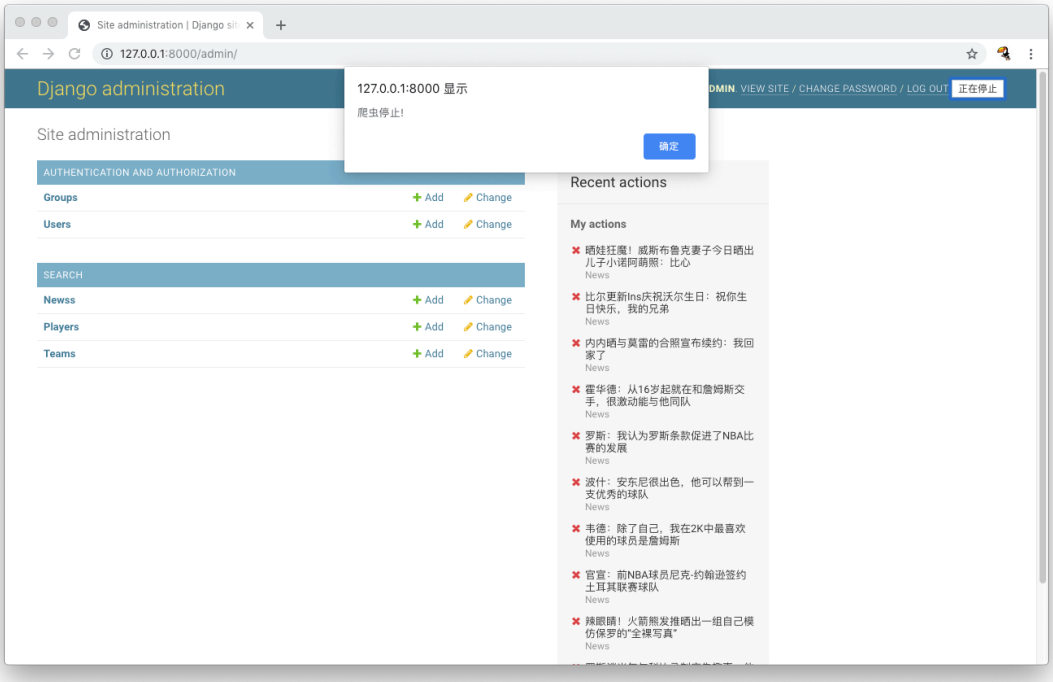
网站的后端主要采用 Python 进行数据的处理和与前端的交互。同时利用了 Django 的数据库和 URL 重定向工具，实现了数据的管理、筛选和传输。

2.1 网页爬虫功能

网页爬虫功能利用了 BeautifulSoup 库，从虎扑 NBA 中获得 NBA 爬取 NBA 新闻和球员信息。程序将每一条新闻的 HTML 页面进行解析，获得编号、标题、时间、来源



(a) 爬虫完成界面



(b) 爬虫停止界面

图 8: 新闻实时更新

和正文信息并储存为本地文件。

对于球队信息，程序将虎扑 NBA 中的球队阵容的 HTML 页面进行解析，将每一个球队的信息和阵容存储为本地文件。

2.2 Django 数据库

对于新闻、球员信息和球队信息的管理利用了 Django 内置的数据库工具。程序建立了相应的三个 Model 并注册在数据库中。在每一次的爬虫中，程序都会调用这些 Model 并向数据库加入相应的条目。在每一次搜索的时候，也是对于数据库中的信息进行查询、复制、处理。

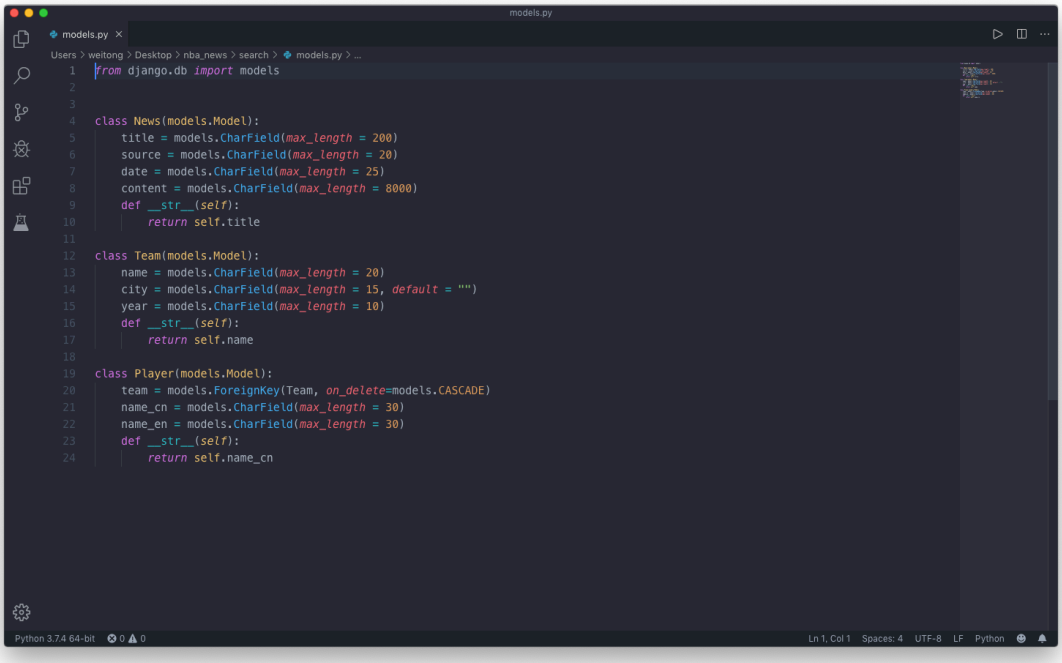


图 9: Django 数据库的模型

2.3 TF-IDF 算法与倒排索引的建立

为了实现更快捷的搜索，程序利用 TF-IDF 算法建立倒排索引。对于每一篇文章，先使用 jieba 库进行分词，之后利用 sklearn 库中的函数获得每个词的 TF-IDF 权值，然后将数据添加进索引中的对应项。

倒排索引存储为字典的字典。存储为“关键词 - 新闻 ID - TF-IDF 权值”。这样的存储模式保证了搜索时的遍历和判断更加便利。

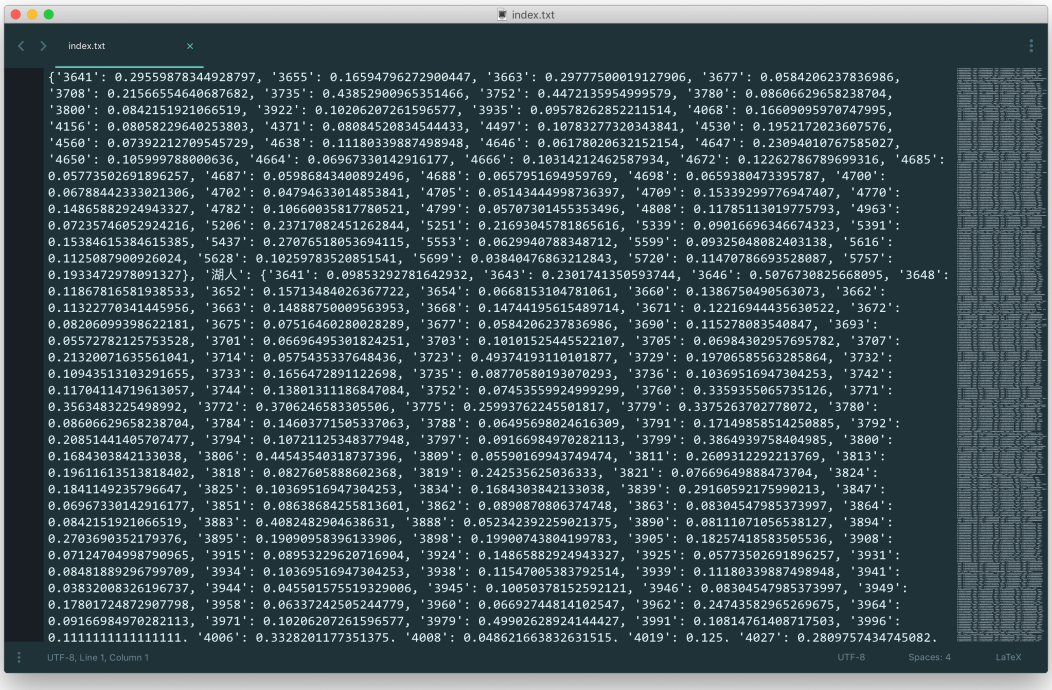


图 10: 倒排索引

2.4 查询过程的实现

当用户键入关键词并点击搜索后，程序首先利用 jieba 将关键词分词并去除其中的标点符号和空格。对于每一个关键词，查找索引以获得每一个文件的关键词权值总和。之后，将全部含有关键词的文件按照权值大小进行排序，得到搜索结果的列表。程序从 HTML 模板中获得当前页数，计算出显示结果列表中的哪些元素。。

对于其中的每个元素，进行正文的截取和关键词的高亮：通过正则匹配，找到正文中第一次出现关键词的位置，向前向后取 40 个字符作为正文片段。在标题和正文片段中对于全部关键词进行正则替换，以获得关键词被标红的文本。

2.5 其他功能的实现

球队热度榜，即在索引中查询球队名称对应的新闻数量，进行比较后传入 HTML 模板中显示。

新闻文本的超链接添加，即在新闻内容页显示前，将新闻正文中全部的球队名称和球员名称进行正则替换以添加超链接。

3 网站相关参数

- **Python 版本**：Python 3.7.4
- **Django 版本**：Django 2.2.5
- **新闻量**：2100
- **搜索时间**：0.01s - 0.03s