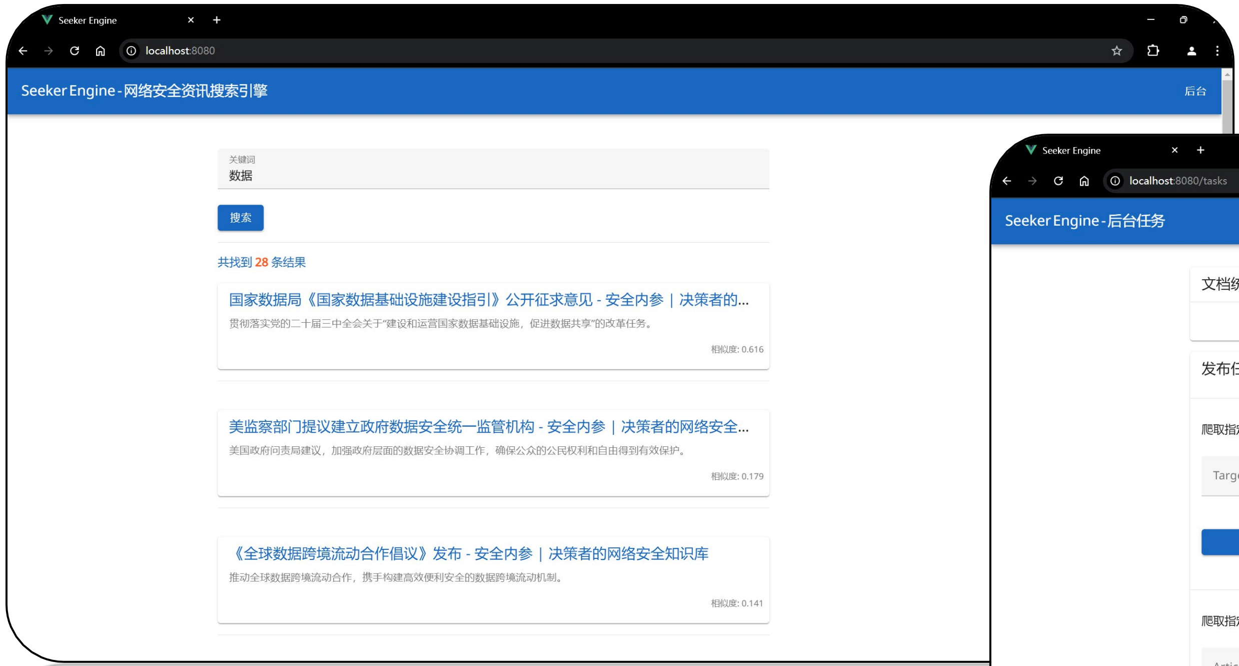
Seeker Engine
网络安全资讯搜索引擎

答辩人：脱敏处理

题目：Project-2 动态网页数据抓取与分页处理

项目介绍

名称：Seeker Engine - 网络安全资讯搜索引擎



Seeker Engine - 网络安全资讯搜索引擎

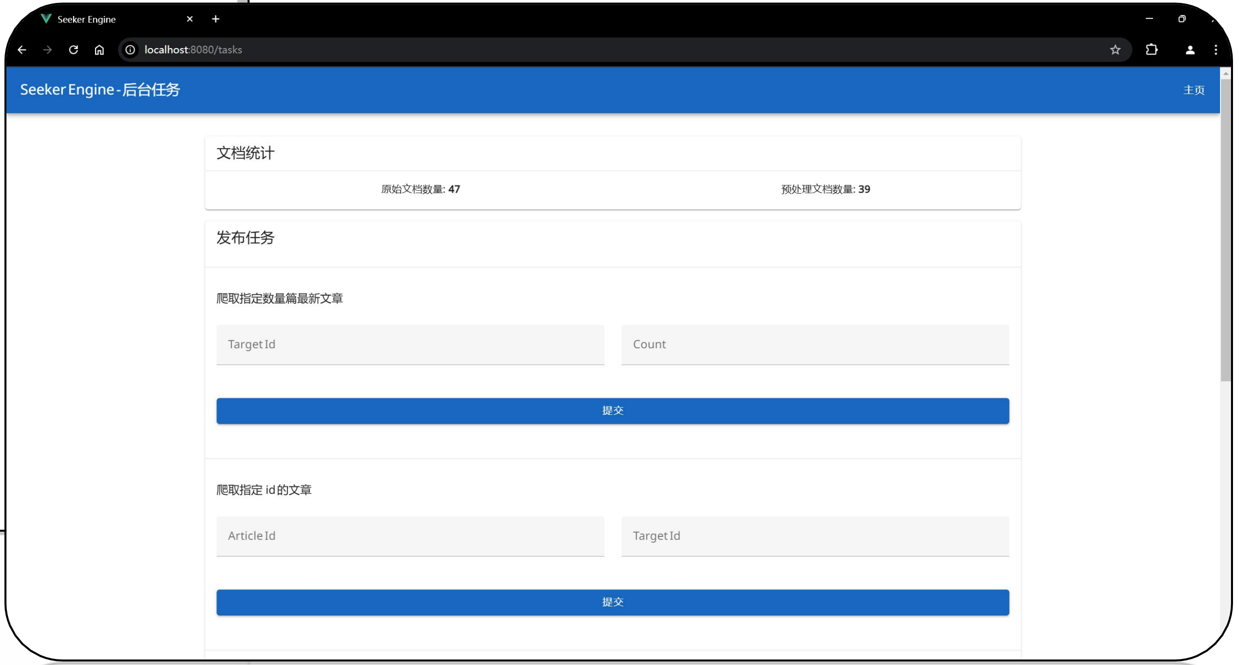
关键词
数据

共找到 28 条结果

国家数据局《国家数据基础设施建设指引》公开征求意见 - 安全内参 | 决策者的...
贯彻落实党的二十届三中全会关于“建设和运营国家数据基础设施，促进数据共享”的改革任务。
相似度: 0.616

美监察部门提议建立政府数据安全统一监管机构 - 安全内参 | 决策者的网络安全...
美国政府问责局建议，加强政府层面的数据安全协调工作，确保公众的公民权利和自由得到有效保护。
相似度: 0.179

《全球数据跨境流动合作倡议》发布 - 安全内参 | 决策者的网络安全知识库
推动全球数据跨境流动合作，携手构建高效便利安全的数据跨境流动机制。
相似度: 0.141



Seeker Engine - 后台任务

文档统计

原始文档数量: **47**　　预处理文档数量: **39**

发布任务

爬取指定数量篇最新文章

Target Id　Count

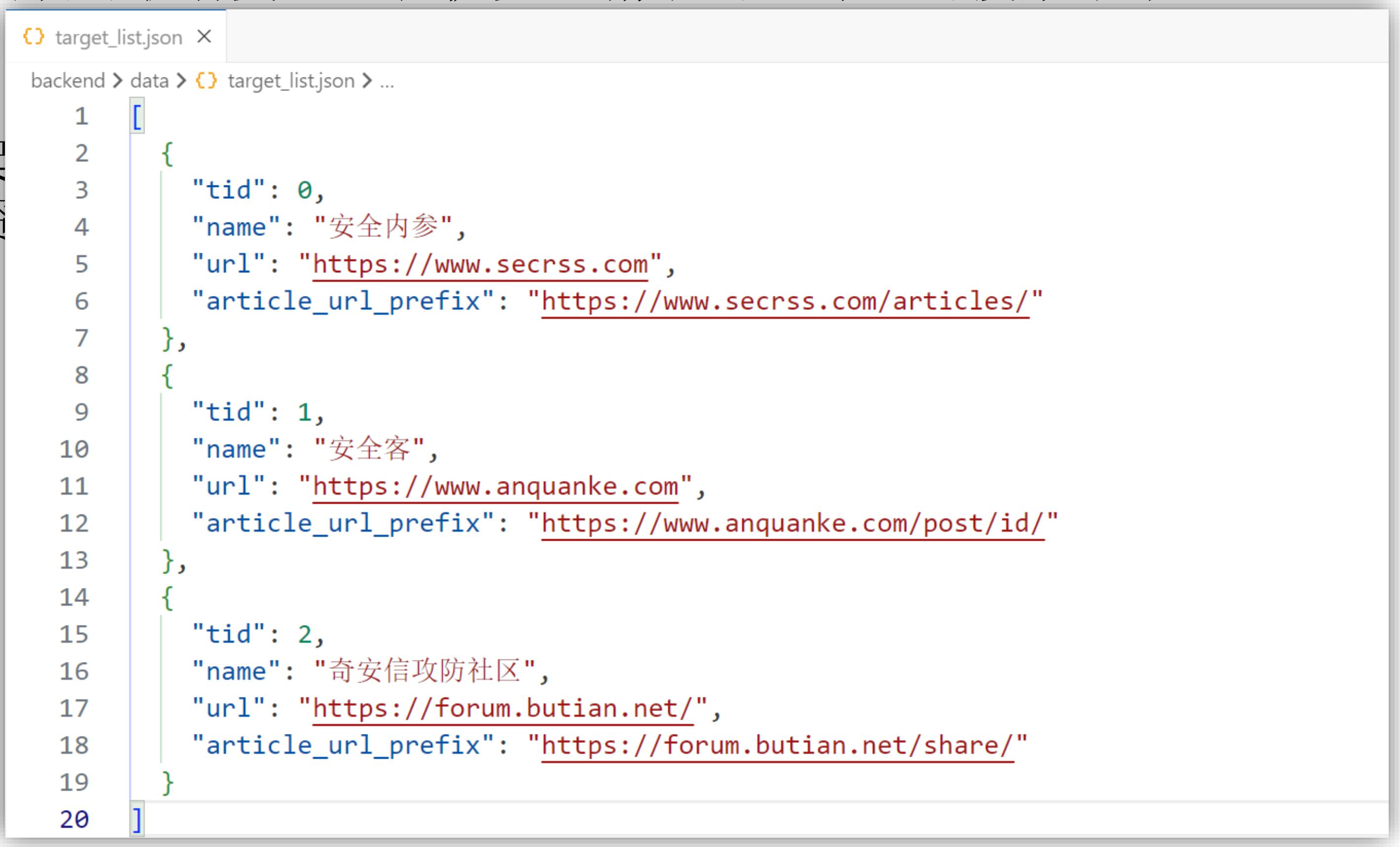
爬取指定 id 的文章

Article Id　Target Id

需求分析

通过搜索引擎精准搜索网络安全领域的最新资讯，提供更加精准的，专业的搜索结果。

基于这样的需求，项目需要定向爬
据处理以构建针对网络安全领域资



target_list.json ×

backend > data > target_list.json > ...

```json
[
  {
    "tid": 0,
    "name": "安全内参",
    "url": "https://www.secrss.com",
    "article_url_prefix": "https://www.secrss.com/articles/"
  },
  {
    "tid": 1,
    "name": "安全客",
    "url": "https://www.anquanke.com",
    "article_url_prefix": "https://www.anquanke.com/post/id/"
  },
  {
    "tid": 2,
    "name": "奇安信攻防社区",
    "url": "https://forum.butian.net/",
    "article_url_prefix": "https://forum.butian.net/share/"
  }
]
```

开发架构

前后端分离开发，其中：

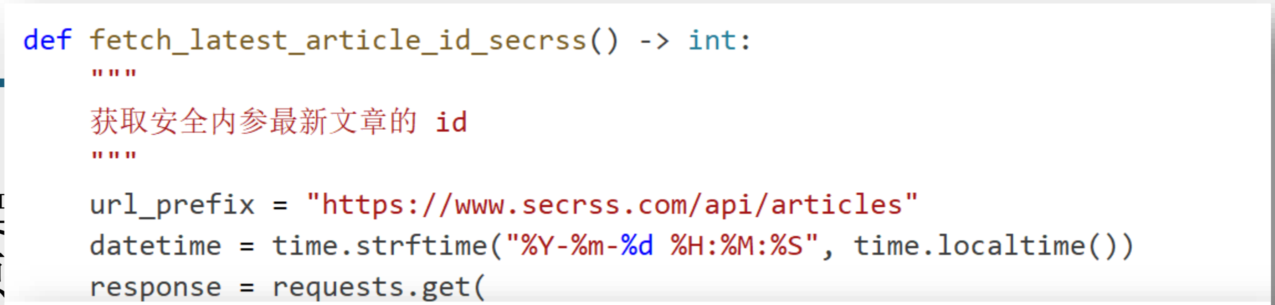
- 前端使用 Vue.js 构建页面
- 后端使用 Python FastAPI 构建 Web API，并集成爬虫、数据预处理、搜索引擎模块。

爬虫模块

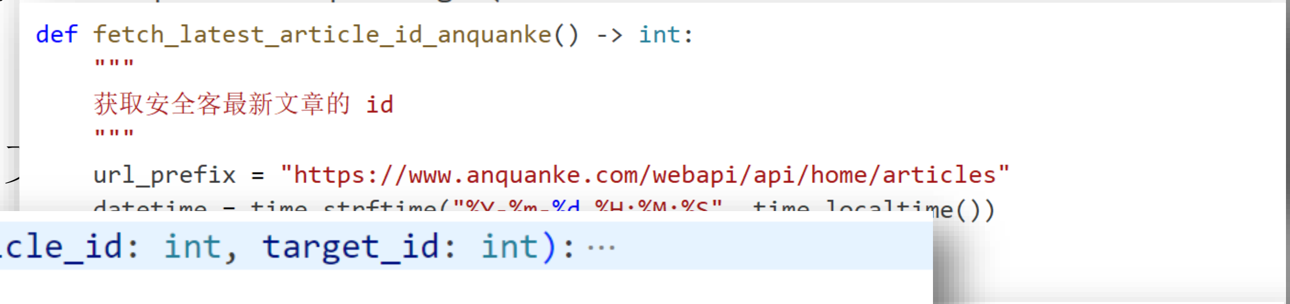
主要使用 **Requests** 和 **BeautifulSoup** 库爬
据库存储 <span>odid, url, title</span> 等元数据, 网页原

其中每个站点，根据分析，获取他们的
则，分别构建

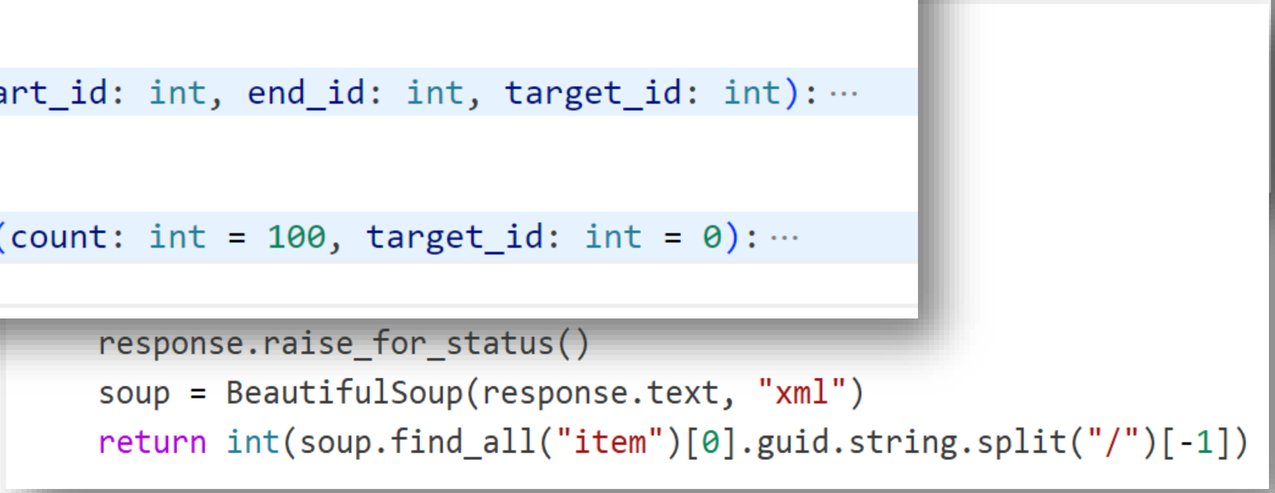
然后利用这几
用于构建后台



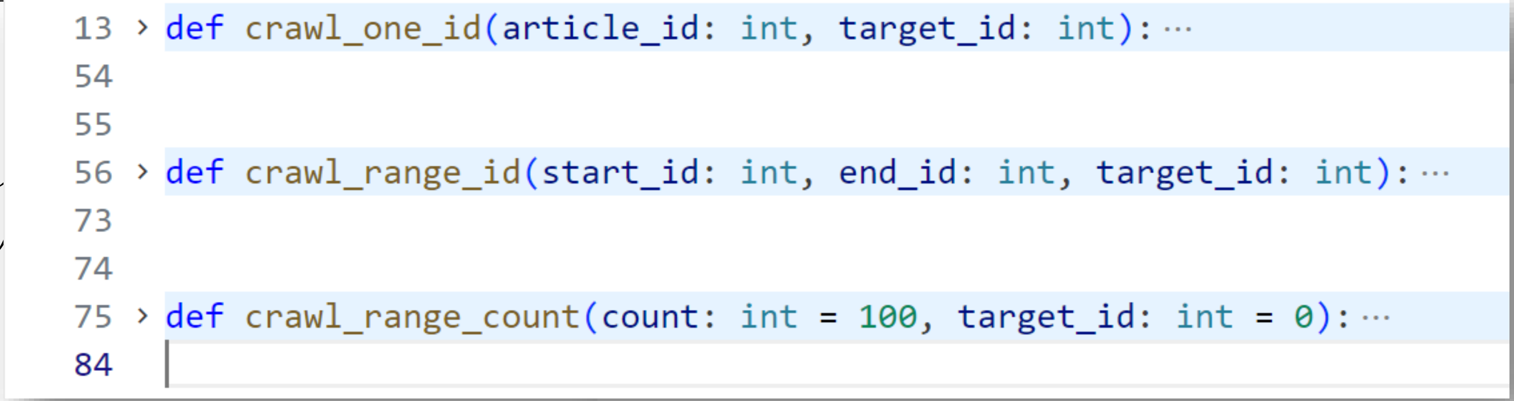
```python
def fetch_latest_article_id_secrss() -> int:
    """
    获取安全内参最新文章的 id
    """
    url_prefix = "https://www.secrss.com/api/articles"
    datetime = time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())
    response = requests.get(
```



```python
def fetch_latest_article_id_anquanke() -> int:
    """
    获取安全客最新文章的 id
    """
    url_prefix = "https://www.anquanke.com/webapi/api/home/articles"
    datetime = time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())
```



```python
    response.raise_for_status()
    soup = BeautifulSoup(response.text, "xml")
    return int(soup.find_all("item")[0].guid.string.split("/")[-1])
```



```python
def crawl_one_id(article_id: int, target_id: int):...


def crawl_range_id(start_id: int, end_id: int, target_id: int):...


def crawl_range_count(count: int = 100, target_id: int = 0):...

```

数据预处理模块 – 原理

倒排索引：即由原始文档集合的关键词构建的倒排索引，基于此可以快速查询某个词语所在的所有文档。

TF-IDF 值：词频（TF，Term Frequency）和逆文档频率（IDF，Inverse Document Frequency）的乘积。

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) * \text{IDF}(t, D)$$

余弦相似度：通过计算两个向量夹角的余弦值来衡量它们的相似度。

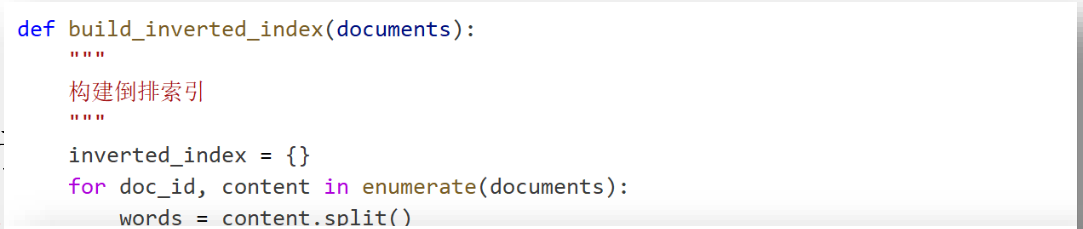
$$\cos(\theta) = (A \cdot B) / (||A|| * ||B||)$$

基于原理分析，项目预处理数据的步骤为：文档分词，构建倒排索引，构建 TF-IDF 矩阵。

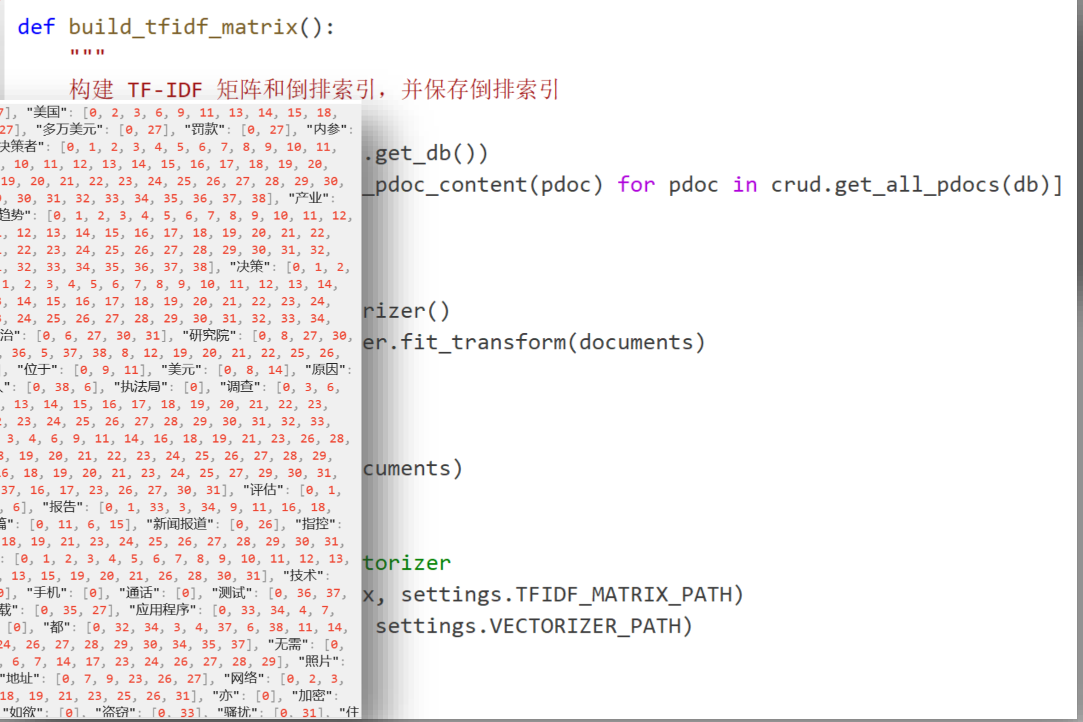
数据预处理模块 – 步骤

文档分词：

中文通过 jieba 库进行分词，英文通过 nltk 库进行
结果文档存储到文件系统预定义目录下的 {pdid}.



```
def build_inverted_index(documents):
    """
    构建倒排索引
    """
    inverted_index = {}
    for doc_id, content in enumerate(documents):
        words = content.split()
```



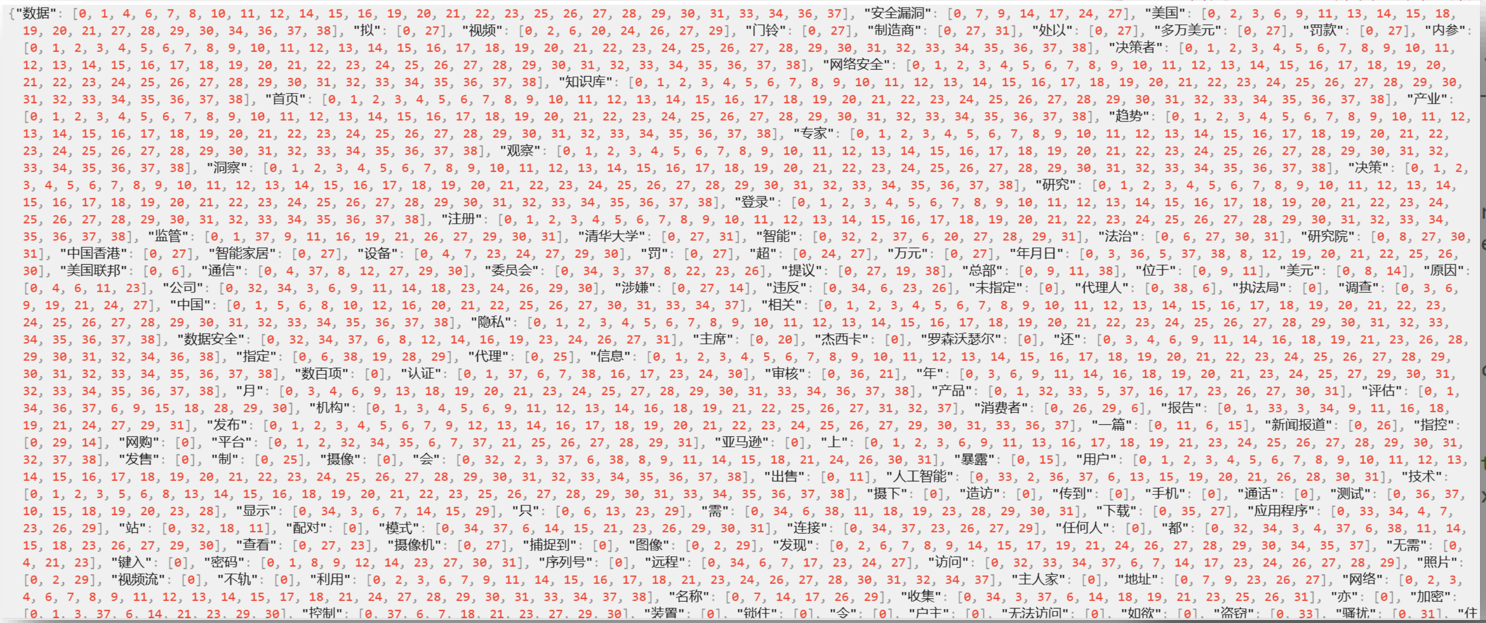
```
def build_tfidf_matrix():
    """
    构建 TF-IDF 矩阵和倒排索引，并保存倒排索引

.get_db())
_pdoc_content(pdoc) for pdoc in crud.get_all_pdocs(db)]


rizer()
er.fit_transform(documents)


cuments)


torizer
x, settings.TFIDF_MATRIX_PATH)
 settings.VECTORIZER_PATH)
```



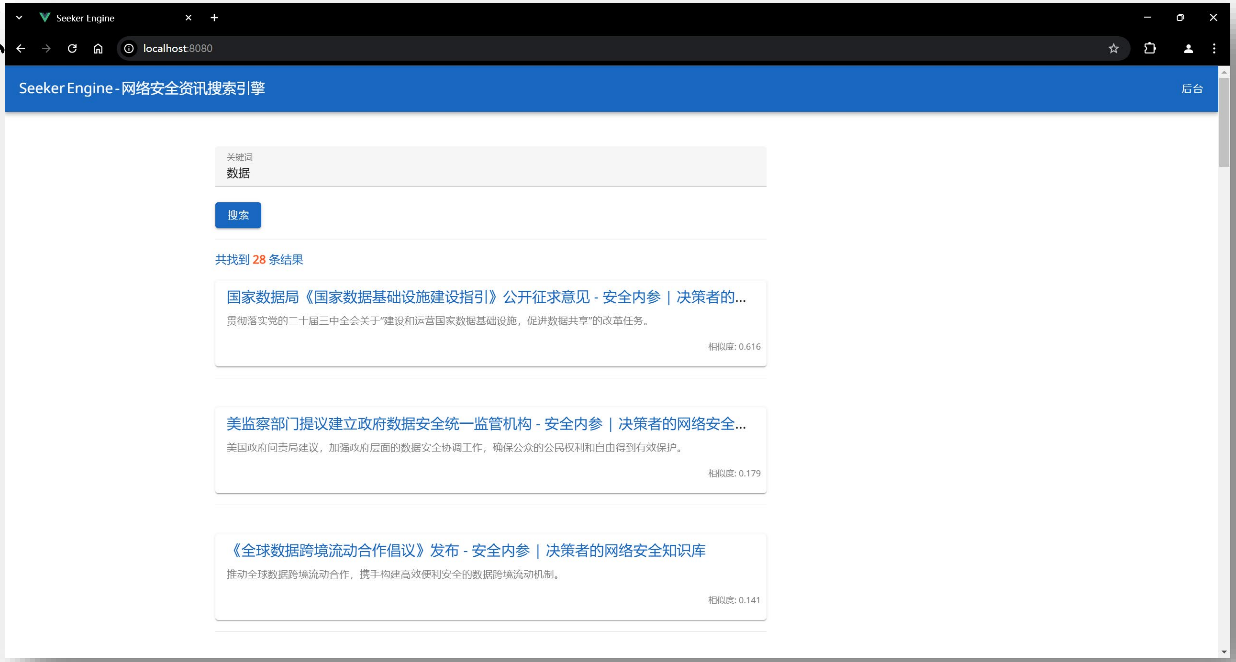
{"数据": [0, 1, 4, 6, 7, 8, 10, 11, 12, 14, 15, 16, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 33, 34, 36, 37], "安全漏洞": [0, 7, 9, 14, 17, 24, 27], "美国": [0, 2, 3, 6, 9, 11, 13, 14, 15, 18,
 19, 20, 21, 27, 28, 29, 30, 34, 36, 37, 38], "拟": [0, 27], "视频": [0, 2, 6, 20, 24, 26, 27, 29], "门铃": [0, 27], "制造商": [0, 27, 31], "处以": [0, 27], "多万美元": [0, 27], "罚款": [0, 27], "内参":
 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38], "决策者": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38], "网络安全": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38], "知识库": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
 31, 32, 33, 34, 35, 36, 37, 38], "首页": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38], "产业":
 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38], "趋势": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38], "专家": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38], "观察": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32,
 33, 34, 35, 36, 37, 38], "洞察": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38], "决策": [0, 1, 2,
 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38], "研究": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38], "登录": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24,
 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38], "注册": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
 35, 36, 37, 38], "监管": [0, 1, 37, 9, 11, 16, 19, 21, 26, 27, 29, 30, 31], "清华大学": [0, 27, 31], "智能": [0, 32, 2, 37, 6, 20, 27, 28, 29, 31], "法治": [0, 6, 27, 30, 31], "研究院": [0, 8, 27, 30,
 31], "中国香港": [0, 27], "智能家居": [0, 27], "设备": [0, 4, 7, 23, 24, 27, 29, 30], "罚": [0, 27], "超": [0, 24, 27], "万元": [0, 27], "年月日": [0, 3, 36, 5, 37, 38, 8, 12, 19, 20, 21, 22, 25, 26,
 30], "美国联邦": [0, 6], "通信": [0, 4, 37, 8, 12, 27, 29, 30], "委员会": [0, 34, 3, 37, 8, 22, 23, 26], "提议": [0, 27, 19, 38], "总部": [0, 9, 11, 38], "位于": [0, 9, 11], "美元": [0, 8, 14], "原因":
 [0, 4, 6, 11, 23], "公司": [0, 32, 34, 3, 6, 9, 11, 14, 18, 23, 24, 26, 29, 30], "涉嫌": [0, 27, 14], "违反": [0, 34, 6, 23, 26], "未指定": [0], "代理人": [0, 38, 6], "执法局": [0], "调查": [0, 3, 6,
 9, 19, 21, 24, 27], "中国": [0, 1, 5, 6, 8, 10, 12, 16, 20, 21, 22, 25, 26, 27, 30, 31, 33, 34, 37], "相关": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38], "隐私": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
 34, 35, 36, 37, 38], "数据安全": [0, 32, 34, 37, 6, 8, 12, 14, 16, 19, 23, 24, 26, 27, 31], "主席": [0, 20], "杰西卡": [0], "罗森沃瑟尔": [0], "还": [0, 3, 4, 6, 9, 11, 14, 16, 18, 19, 21, 23, 26, 28,
 29, 30, 31, 32, 34, 36, 38], "指定": [0, 6, 38, 19, 28, 29], "代理": [0, 25], "信息": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
 30, 31, 32, 33, 34, 35, 36, 37, 38], "数百项": [0], "认证": [0, 1, 37, 6, 7, 38, 16, 17, 23, 24, 30], "审核": [0, 36, 21], "年": [0, 3, 6, 9, 11, 14, 16, 18, 19, 20, 21, 23, 24, 25, 27, 29, 30, 31,
 32, 33, 34, 35, 36, 37, 38], "月": [0, 3, 4, 6, 9, 13, 18, 19, 20, 21, 23, 24, 25, 27, 28, 29, 30, 31, 33, 34, 36, 37, 38], "产品": [0, 1, 32, 33, 5, 37, 16, 17, 23, 26, 27, 30, 31], "评估": [0, 1,
 34, 36, 37, 6, 9, 15, 18, 28, 29, 30], "机构": [0, 1, 3, 4, 5, 6, 9, 11, 12, 13, 14, 16, 18, 19, 21, 22, 25, 26, 27, 31, 32, 37], "消费者": [0, 26, 29, 6], "报告": [0, 1, 33, 3, 34, 9, 11, 16, 18,
 19, 21, 24, 27, 29, 31], "发布": [0, 1, 2, 3, 4, 5, 6, 7, 9, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 29, 30, 31, 33, 36, 37], "一篇": [0, 11, 6, 15], "新闻报道": [0, 26], "指控":
 [0, 29, 14], "网购": [0], "平台": [0, 1, 2, 32, 34, 35, 6, 7, 37, 21, 25, 26, 27, 28, 29, 31], "亚马逊": [0], "上": [0, 1, 2, 3, 6, 9, 11, 13, 16, 17, 18, 19, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31,
 32, 37, 38], "发售": [0], "制": [0, 25], "摄像": [0], "会": [0, 32, 2, 3, 37, 6, 38, 8, 9, 11, 14, 15, 18, 21, 24, 26, 30, 31], "暴露": [0, 15], "用户": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38], "出售": [0, 11], "人工智能": [0, 33, 2, 36, 37, 6, 13, 15, 19, 20, 21, 26, 28, 30, 31], "技术":
 [0, 1, 2, 3, 5, 6, 8, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38], "摄下": [0], "造访": [0], "传到": [0], "手机": [0], "通话": [0], "测试": [0, 36, 37,
 10, 15, 18, 19, 20, 23, 28], "显示": [0, 34, 3, 6, 7, 14, 15, 29], "只": [0, 6, 13, 23, 29], "需": [0, 34, 6, 38, 11, 18, 19, 23, 28, 29, 30, 31], "下载": [0, 35, 27], "应用程序": [0, 33, 34, 4, 7,
 23, 26, 29], "站": [0, 32, 18, 11], "配对": [0], "模式": [0, 34, 37, 6, 14, 15, 21, 23, 26, 29, 30, 31], "连接": [0, 34, 37, 23, 26, 27, 29], "任何人": [0], "都": [0, 32, 34, 3, 4, 37, 6, 38, 11, 14,
 15, 18, 23, 26, 27, 29, 30], "查看": [0, 27, 23], "摄像机": [0, 27], "捕捉到": [0], "图像": [0, 2, 29], "发现": [0, 2, 6, 7, 8, 9, 14, 15, 17, 19, 21, 24, 26, 27, 28, 29, 30, 34, 35, 37], "无需": [0,
 4, 21, 23], "键入": [0], "密码": [0, 1, 8, 9, 12, 14, 23, 27, 30, 31], "序列号": [0], "远程": [0, 34, 6, 7, 17, 23, 24, 27], "访问": [0, 32, 33, 34, 37, 6, 7, 14, 17, 23, 24, 26, 27, 28, 29], "照片":
 [0, 2, 29], "视频流": [0], "不轨": [0], "利用": [0, 2, 3, 6, 7, 9, 11, 14, 15, 16, 17, 18, 21, 23, 24, 26, 27, 28, 30, 31, 32, 34, 37], "主人家": [0], "地址": [0, 7, 9, 23, 26, 27], "网络": [0, 2, 3,
 4, 6, 7, 8, 9, 11, 12, 13, 14, 15, 17, 18, 21, 24, 27, 28, 29, 30, 31, 33, 34, 37, 38], "名称": [0, 7, 14, 17, 26, 29], "收集": [0, 34, 3, 37, 6, 14, 18, 19, 21, 23, 25, 26, 31], "亦": [0], "加密":
 [0, 1, 3, 37, 6, 14, 21, 23, 29, 30], "控制": [0, 37, 6, 7, 18, 21, 23, 27, 29, 30], "装置": [0], "锁件": [0], "令": [0], "户主": [0], "无法访问": [0], "如欲": [0], "盗窃": [0, 33], "骚扰": [0, 31], "仵

搜索模块

1. 前端页面输入搜索词，通过 Web API 调用后端搜索引擎模块。
2. 获取搜索词 -> 分词 -> 根据倒排索引查询相关文档。
3. 查询词通过 TfidfVectorizer 对象 TF-IDF 向量化

➔与文档集合的 TF-IDF 矩阵进行余弦相似

➔返回根据相似度排序的结果。



Seeker Engine - 网络安全资讯搜索引擎

后台

关键词

数据

搜索

共找到 28 条结果

国家数据局《国家数据基础设施建设指引》公开征求意见 - 安全内参 | 决策者的...

贯彻落实党的二十届三中全会关于"建设和运营国家数据基础设施，促进数据共享"的改革任务。

相似度: 0.616

美监察部门提议建立政府数据安全统一监管机构 - 安全内参 | 决策者的网络安全...

美国政府问责局建议，加强政府层面的数据安全协调工作，确保公众的公民权利和自由得到有效保护。

相似度: 0.179

《全球数据跨境流动合作倡议》发布 - 安全内参 | 决策者的网络安全知识库

推动全球数据跨境流动合作，携手构建高效便利安全的数据跨境流动机制。

相似度: 0.141

参考文献

1. Joblib. Running Python functions as pipeline jobs — joblib 1.4.2 documentation[EB/OL]. (2024-11-26)[2024-11-26]. https://joblib.readthedocs.io/en/stable/.
2. fxsjy. jieba/README.md at master · fxsjy/jieba[EB/OL]. (2024-11-26)[2024-11-26]. https://github.com/fxsjy/jieba/blob/master/README.md.
3. NLTK. Natural Language Toolkit[EB/OL]. (2024-11-26)[2024-11-26]. https://www.nltk.org/.
4. 云南旅游攻略搜索引擎设计与实现[EB/OL]. (2024-11-26)[2024-11-26]. https://blog.csdn.net/qq_35993946/article/details/88087827.
5. 知乎用户. 余弦距离与欧式距离[EB/OL]. (2024-11-26)[2024-11-26]. https://zhuanlan.zhihu.com/p/84643138.

谢谢！