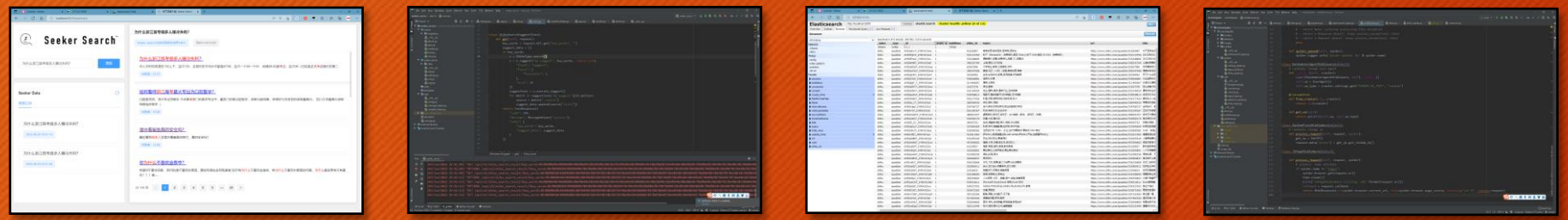
搜索引擎系统最佳实践

Seeker Search

A Best Practice on Search Engine System



大数据应用开发课程设计作品

- 团队：The Seekers（付东源、谭斯谦）
- 汇报人：付东源

By Team The Seekers 2022-06-06

项目概述

项目定义

➢ 一个实现**搜索引擎基本功能**的软件系统

实现目标

➢ **数据爬取**

自动化登录、**自动化**爬虫脚本，目标网站**一键增量爬取**

➢ **索引入库**

将爬取数据按**索引**分类，并进一步**处理，存入数据仓库**

➢ **用户检索**

发送**HTTP请求**至项目服务器，请求**搜索关键词相关数据**

➢ **结果呈现**

开发用户易于接受的**网页应用**，简洁明了**展示搜索结果**

Baidu 百度　yahoo!

Google　bing

搜索引擎

根据**用户需求**与一定**算法**，运用**特定策略**从**互联网**检索出**指定信息**反馈给用户的一门**检索技术**。

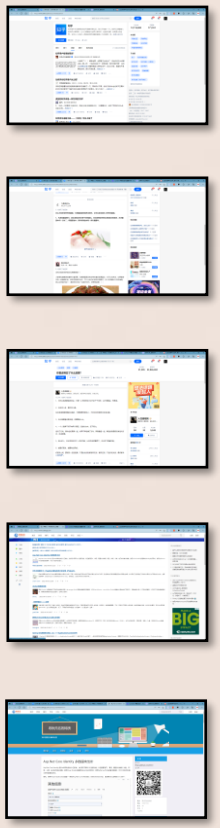
核心模块

爬虫　索引　检索　排序

依托技术

网络爬虫　网页处理　检索排序　NLP　...

项目架构



网站：知乎

网站：博客园

各类门户网站

提供综合性互联网信息资源以及有关信息服务的网站

网络爬虫

通过请求RESTful API或操作Chrome浏览器爬取并解析数据

在各类门户网站定期爬取数据，将爬取结果上报搜索服务器

爬虫服务器

Pipeline

搜索服务器

RESTful API

应用服务器

Pipeline

搜索工具

按预定义方法处理数据
添加索引并存储数据

RESTful API

搜索工具面板

面向搜索工具的WEB应用
供管理员管理内容

RESTful API

关系型数据库

按预定义数据结构
对爬取的数据进行持久化

分析工具

允许以RESTful API形式
接受高级聚合查询指令
返回当前数据的搜索结果

处理并增量存储（持久化）数据，为应用服务器提供检索接口

RESTful API

WEB应用

为用户提供精简易用的
交互界面与
直观的搜索结果展示

应用服务器

与搜索服务器交互
并向WEB应用提供数据接口
实现用户搜索业务

运行WEB应用，提供搜索服务，
打通用户“最后一公里”

浏览器访问

用户

提供关键词，检索数据

Scrapy

Scrapy

ES ES-head Kibana MySQL

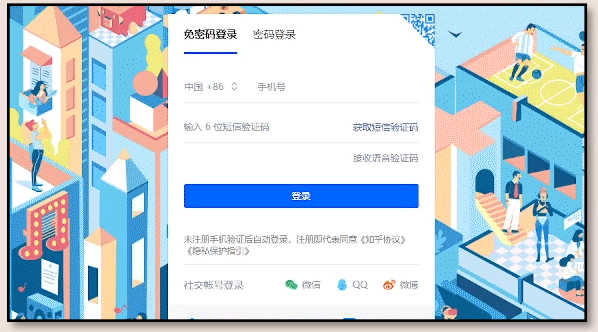
Django Vue Vite

互联网数据 → 爬取 → 处理、索引与持久化 → 检索服务

数据爬取

Scrapy

模拟登录模块



项目**基于cv2库**，实现知乎网站爬虫**拼图验证码识别**与**自动拖动验证**功能，另集成百度智能云在线识别验证码功能。

基于HTML解析的爬取方式

使用ChromeDriver**模拟浏览器操作**，获取HTML文档，再通过**CSS选择器**定位并获取所需内容。所见即所得，**实现简单，但效率较低。**

```
item_loader_question.add_css(
"topics", ".QuestionHeader-
topics .Popover div::text")
```

知乎：基于HTML解析进行爬取
zhihu_spider.py 139:9

{...} 基于AJAX请求的爬取方式

分析网站代码，**提取AJAX请求**，获取网站Cookie与信息id列表后，**直接Request请求**获取数据。**效率较高**，但有时涉及逆向，**难度较大。**

```
Request(url=parse.urljoin(res
ponse.url,
'/NewsAjax/GetAjaxNewsInfo?co
ntentId={}'.format(post_id)))
```

博客园：基于AJAX请求进行爬取
cnblogs_spider.py 94:19

索引入库

Crawled Data Item Object

网站数据模型实例

yield

ElasticSearchPipeline

ElasticSearch数据流处理器

MySQLTwistedPipeline

MySQL数据流处理器

JSONExporterPipeline

数据流JSON文件生成器

ArticleImagePipeline

图片批量下载器

成员变量

Item初始化时，需要提供该数据对象所需的必要字段，**生成数据Model**，以进行后续操作。

```
zhihu_id = scrapy.Field()
topics = scrapy.Field()
url = scrapy.Field()
title = scrapy.Field()
content = scrapy.Field()
create_time = scrapy.Field()
update_time = scrapy.Field()
answers = scrapy.Field()
comments = scrapy.Field()
total_view = scrapy.Field()
clicks = scrapy.Field()
```

Item成员变量 items.py 98:5

基类方法

所有Item都应覆写的方法，包括**存储至ES、生成DML语句等。**

get_insert_sql

方法：生成SQL语句

save_to_es

方法：存储至ElasticSearch

数据入库

```
insert_sql = "insert into zhihu_question(zhihu_id, …)
VALUES (%s, …) ON DUPLICATE KEY UPDATE content =
VALUES(content)…"
```

DML-INSERT模板语句 items.py 111:9

```
params = (zhihu_id, topics, url, title, content,
answers, comments, total_view)
```

参数表 items.py 121:9

提供DML-INSERT模板语句与参数表，通过pymysql数据库驱动，完成**MySQL数据增量入库。**

数据索引

```
class ZhihuType(DocType):
    suggest = Completion(analyzer=ik_analyzer)
    topics = Text(analyzer="ik_max_word")
```

ElasticSearch索引类，使用最大分词analyzer
elasticsearch_types.py 17:1

```
if __name__ ==
"__main__":
    ZhihuType.init()
```

创建ES索引
elasticsearch_types.py 35:1

```
def save_to_es(self):
    zhihu = ZhihuType()
zhihu.save()
```

Item基类方法
items.py 125:5

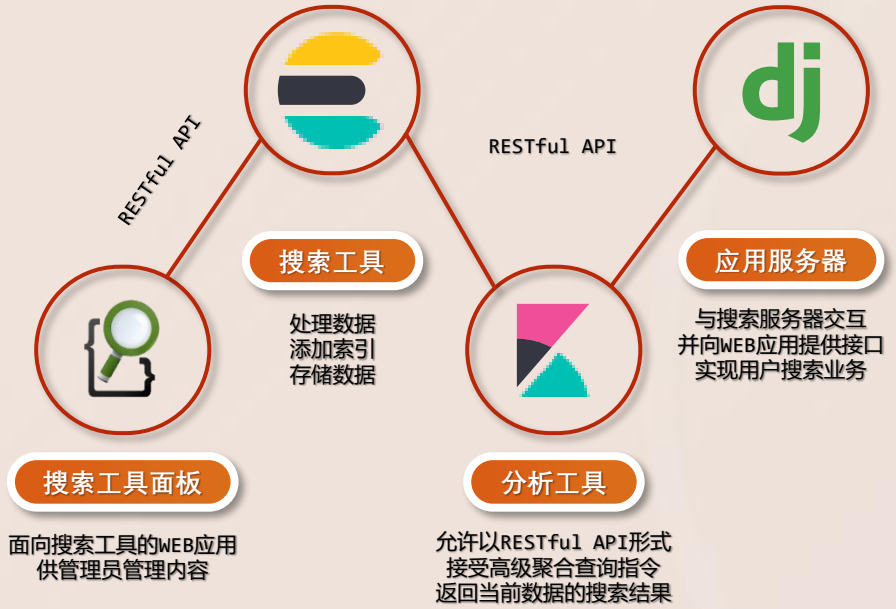
```
class ElasticSearchPipeline(object):
    def process_item(self, item, spider):
        item.save_to_es()
        return item
```

ElasticSearch数据流处理器 pipelines.py 90:1

➢ 通过设置settings.py中的ITEM_PIPELINES属性，**接入指定的pipeline**，灵活切换数据处理方式，**结合yield，实现并行处理**（MySQL+ElasticSearch）

➢ 设计**索引类型，规定索引字段与analyzer**，运行init()方法，在ES中**创建索引**，此后ElasticSearchPipeline**调用Item.save_to_es()，完成该条数据的处理、索引**

用户检索

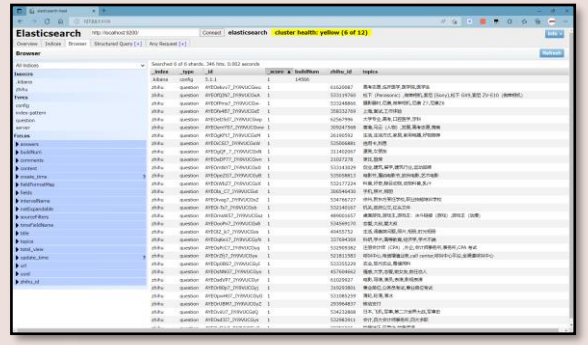


搜索工具：ElasticSearch

```
cd <BASE_DIR>/elastic_search/bin  elasticsearch.bat
```

通过cmd命令启动ElasticSearch

- 以服务器形式运行在9200端口，**为项目搜索核心功能提供支撑**
- 集成ik_max_word、ik_smart等analyzer（分析器），可**对文本进行分词等分析处理**
- 命中搜索结果**局部高亮**（Highlight）



ElasticSearch-Head：ES管理控制台 （Port 9300）

分析工具：Kibana

```
cd <BASE_DIR>/kibana/bin  kibana.bat
```

通过cmd命令启动Kibana

以服务器形式运行在5601端口，提供RESTful API（以请求类型（GET、POST、PUT、DELETE）区分不同功能），**接受高级聚合查询指令，返回查询结果。**

HTTP Request
GET
zhihu/question
/AYEOekvv7_IYi
9VUCGwu?_sourc
e=title

Request

{"_index":"zhihu","_type":"question","_id":"AYEOekvv7_IYi9VUCGwu","_version":1,"found":true,"_source":{"title":"对于医学生来讲是985211的名头重要，还是该校医学本身在全国的排名重要？"}}

Response

应用服务器：Django

- 以服务器形式运行在8000端口，处理**前端搜索业务逻辑**
- 通过urls与view模块，向搜索网页**提供服务接口**
- 根据用户自网页发送的**搜索请求，与Kibana交互**，获取搜索结果，并**响应用户**

用户检索


```
    "_index": "zhihu",
    "_type": "question",
    "_id": "AYEOekvv7_IYi9VUCGwu",
    "_version": 1,
    "_score": 1,
    "_source": {
        "zhihu_id": 61620087,
        "topics": "高考志愿,临床医学,医学院,医学生",
        "url": "https://www.zhihu.com/question/61620087",
        "title": "对于医学生来讲 是985211的名头重要，还是该校医学本身在全国的排名重要？",
        "content": "今年高考，铁了心要学临床医学，以我的分数，可以上的有东大，南方医科大，苏州大学，重庆医科大，大连医科大
这么几个，有的是985211但是医科本身并不厉害… ",
        "answers": 17,
        "comments": 0,
        "total_view": 58648,
        "suggest": [
            {
                "input": [
                    "医学院", "临床医学", "临床", "高考", "医学", "学院", "医学生", "学生", "志愿"
                ],
                "weight": 7
            }
        ]
    }
}
```

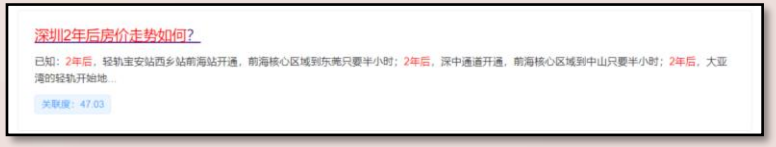

使用Kibana，自项目ElasticSearch引擎获取的一个完整数据条目

结果呈现

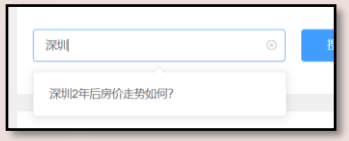
单页WEB应用

- 基于**Vue 3构建的单页WEB应用**，使用Vite2打包构建
- 使用Element UI组件，**美观大方，简洁明了，体验流畅**
- **搜索输入提示、搜索历史、数据概览、分页**等实用功能
- 搜索结果命中部分**高亮，点击跳转**源页面，方便快捷

基于LocalStorage的搜索历史



搜索结果命中部分高亮，关联度显示，点击跳转源页面



搜索输入提示

分页（上）与数据概览（下）

Request

Response

应用服务器

- 基于Django的应用服务器向上述单页WEB应用提供了**两个RESTful API：suggest、search_result**

suggest

类型	入参		出参	
GET	key_words	搜索关键词	key_words	搜索关键词
			suggest_data	搜索建议（数组）

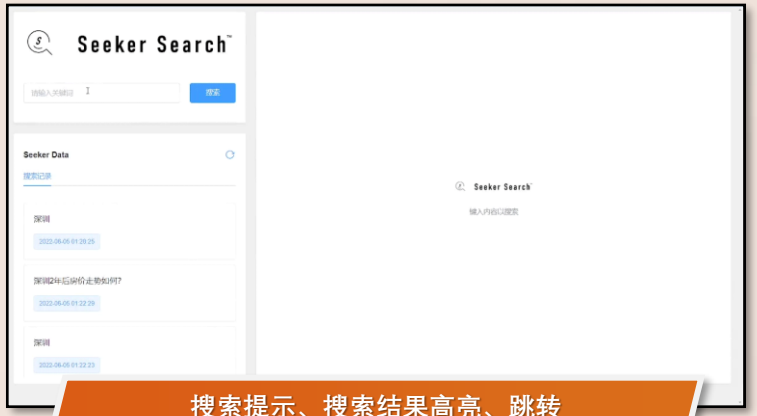
search_result

类型	入参		出参
GET	key_words	搜索关键词	key_words（搜索关键词）、duration（搜索用时（秒））、page_size（一页结果数）、page_index（页码）、total_pages（总页数）、total_result（结果数）、search_data（搜索结果）

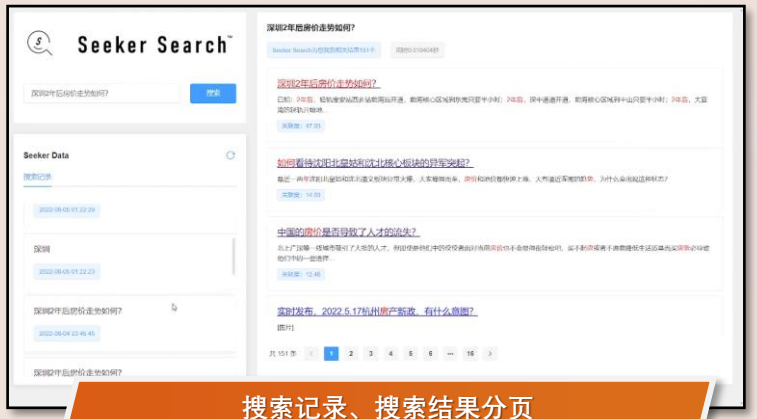
结果呈现

前端WEB应用：单页搜索引擎

结果呈现



搜索提示、搜索结果高亮、跳转



搜索记录、搜索结果分页

suggest

```
GET api/v1/suggest?key_words=深圳

{
    "code": 200,
    "message": "操作成功！",
    "info": {
        "key_words": "深圳",
        "suggest_data": [
            "深圳2年后房价走势如何？"
        ]
    }
}
```

search_result

```
GET api/v1/search_result?key_words=深圳

{
    "code": 200,
    "message": "操作成功！",
    "info": {
        "duration": 0.004272,
        "key_words": "深圳",
        "page_size": 10,
        "page_index": 1,
        "total_result": 1,
        "total_pages": 1,
        "search_data": [
            {
                "title": "<span style='color: red;'>深圳</span>2年后房价走势如何？",
                "content": "已知：2年后，轻轨宝安站西乡站前海站开通，前海核心区域到东莞只要半小时；2年后，深中通道开通，前海核心区域到中山只要半小时；2年后，大亚湾的轻轨开始地… ",
                "url": "https://www.zhihu.com/question/520708726",
                "score": 9.764106
            }
        ]
    }
}
```

应用服务器接口请求URL及返回数据

项目总结

最佳实践

- **技术**

  scrapy + MySQL + ES + ES-Head + Django + Vue，**广泛应用主流技术**，构建**可靠而强大**的搜索引擎

- **功能**

  融通前后端知识，打通**数据爬取、索引入库、用户检索、结果呈现**四大功能，构建**简单而实用**的搜索引擎

- **设计**

  原有网站**增量爬取**，新网站**易于接入**（写爬虫、建表/索引、开始爬取），构建**健壮而易于扩展**的搜索引擎

结语

Seeker Search™

在大数据中

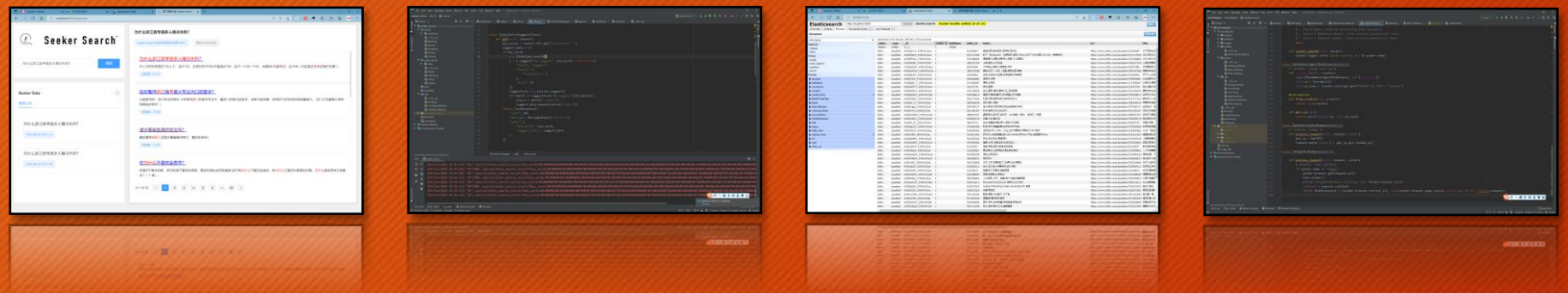
发现更大价值

欢迎扫码访问GitHub仓库

搜索引擎系统最佳实践

Seeker Search

A Best Practice on Search Engine System



感谢聆听！

By Team The Seekers 2022-06-06