搜索引擎ES SearchUI

开

发

手

册



修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2020-11-17 | 1.0 | 初始版本 | 王吉平 |

# 一：ESSearchUI背景说明

本产品是参考原先公司使用attivio searchui的搜索引擎产品

由于attivio searchui在原来使用过程中，大家普遍发现他的二次改造的复杂度和产品搜索结果准确率和对产品学习成本比较大等问题和不便，所有计划重新设计开发一套基于ElasticSearch的搜索引擎

本工具主要处理的工作内容是：

将医院非结构化文本如电子病历结构化后写入ElasticSearch再提供给客户搜索引擎页面，搜索想要查询的非结构化文档和对应的结构化信息

# 二：需求分析

## BRD（业务需求）

业务目的：替代attivio searchui

业务场景：结构化电子病历非结构化数据同时提供搜索功能

使用人员：运营人员、C端用户

## US（用户故事）

运营人员指定待结构化数据（txt、docx、doc、pdf……）位置（文件夹、数据库）、通过编写结构化规则脚本结构化数据、实时或者增量写入搜索引擎中、终端用户通过搜索关键字搜索得到想要信息并做一些定制化的操作（结果导入spotfire）；运营人员管理索引

## PRD（产品需求）

### 名字定义

1. 用户：使用搜索引擎的人员
2. 运营：编辑结构化脚本，提供索引数据、应用培训的人员
3. 维护：提升性能、扩展服务、部署应用的人员
4. 开发：系统结构设计、结构化工具定制化配置、前端搜索功能设计与实现
5. 运营后台：索引管理、系统配置的后台网页
6. 搜索终端SearchUI：面向用户，提供搜索服务、定制化功能

### ETL模块（logstash、attivio）

类似于attivio提供结构化功能同时将数据写入搜索引擎中

运营人员操作



参考：logstash工具介绍与应用.ppt

### 运营后台（功能较为复杂，我会出一个较为详细的操作手册，kibana暂时满足）

1. 登录功能

输入账号密码登录进入运营后台主页

1. 索引服务管理

删除索引

新增索引

修改索引结构

索引生命周期管理

参考：<https://www.elastic.co/guide/en/elasticsearch/reference/7.9/index-lifecycle-management.html>

索引快照生命周期管理

参考：<https://www.elastic.co/guide/en/elasticsearch/reference/7.9/snapshot-lifecycle-management.html>

3）报警机制（elasticsearch收费部分，要么购买要自己研发一套）

服务器性能告警机制

针对特定索引配置预警实时告警

1. 机器学习（elasticsearch收费部分，要么购买要么自己研发一套）

看看可能配合DS做做机器学习相关任务

1. 系统配置

用户管理

角色管理

1. 安全模块

访问es api时候需验证

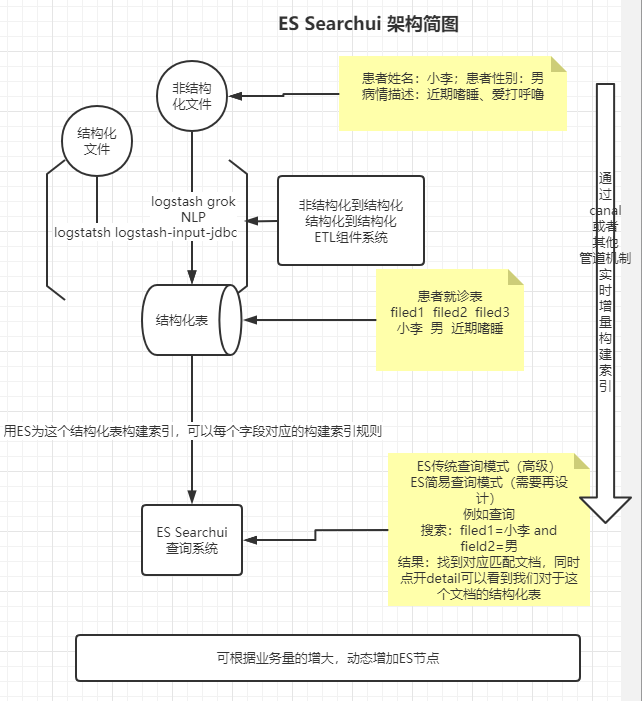
### 搜索终端（这需要自主定制化一下，变为咱们的SearchUI）

1. 简单查询
2. 高级查询
3. 查询结果export
4. 查询结果download
5. 换肤
6. 保存查询语句
7. 打开查询语句
8. 能与spotfire等第三方应用交互

# 三：概要设计

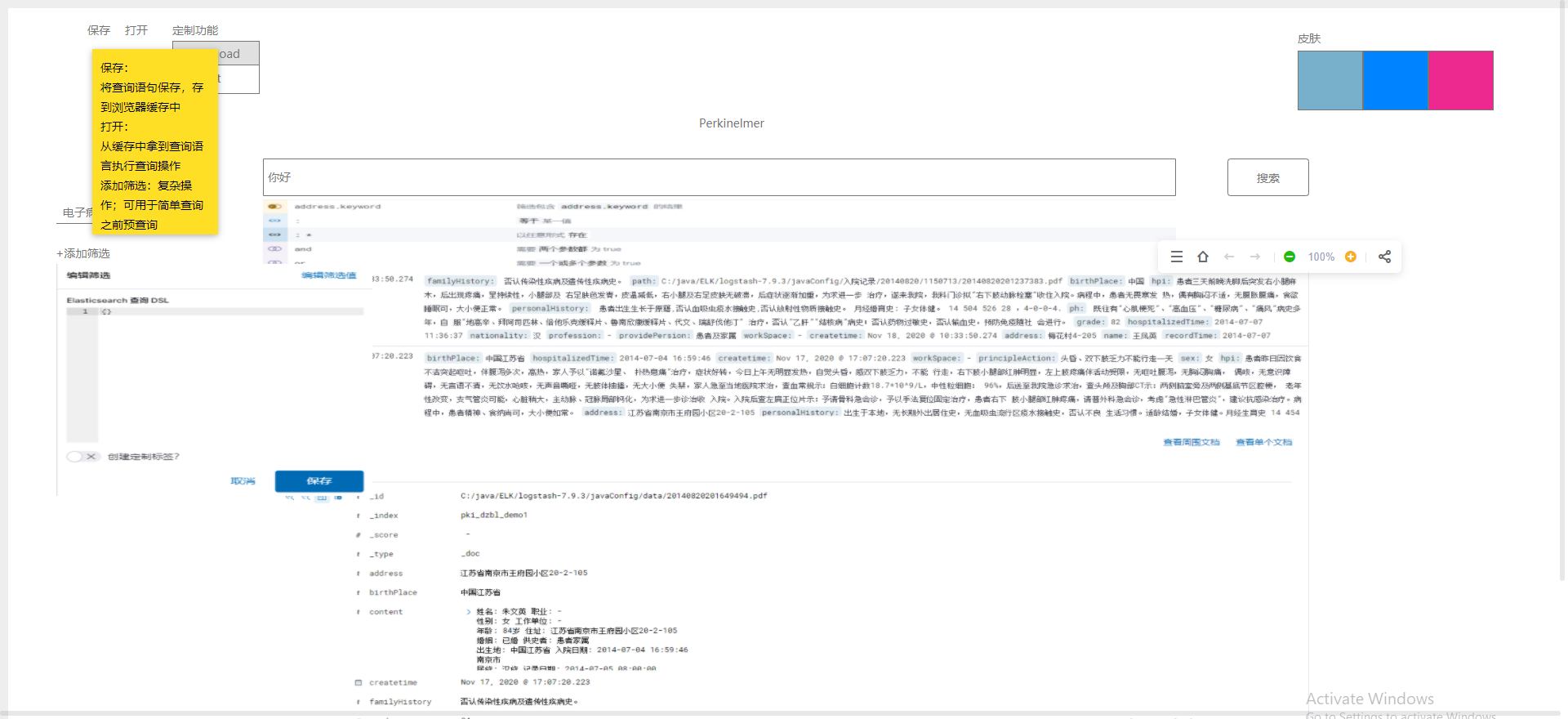
完整架构图位置：[**https://www.processon.com/diagraming/5daefa7ce4b09df550dfd01b**](https://www.processon.com/diagraming/5daefa7ce4b09df550dfd01b)

项目管理：<https://perkinelmer.atlassian.net/jira/software/projects/ES/boards/361>



ES SearchUI架构简图

设计稿参考：<https://app.mockplus.cn/run/rp/J9Y15KkXyGW/bQVyx0dWz9s?ps=0&ha=0&la=0&fc=0&out=1>



SearchUI终端设计稿

# 四：详细设计

## 4.1 ETL模块

暂无补充，参考PRD

## 4.2 运营后台

暂无补充，参考PRD

## 4.3 searchui终端

http://47.103.64.41:5601/app/discover

### 方式一：自定义Kibana中的discover模块

好处：功能强大

难处：kibana项目很复杂，二次开发成本高，需克服

难点1：找到discover模块

难点2：如何单拧出discover模块，还能运行

难点3：如何自定义开发discover模块，可能需要学习一套全新前端技术体系

难点4：如何与kibana、es 交互

### 方式二：完全重新设计开发

好处：自由度大

难点：如何接入KQL

### 方式三：套个外壳方式开发，以iframe角色担任其中

好处：超级简单，设置css即可，很快出效果

坏处：套了壳，感觉好不专业；自定义开发不方便

# 五：ESSearchUI实施流程

“部署安装手册.docx”

# 六：ESSearchUI用户手册

“ES SearchUI用户操作手册.docx”

# 七：ESSearchUI功能介绍

参考用户手册