================ 平台需要回答的问题 ================

定义的关键字：1）文书title、2）案件号、3）法院名、4）文书类型、5）案由、6）程序、7）日期、8）审判长、9）原告、原告律师、原告律师事务所10）被告、被告律师、被告律师事务所

1. 通过title，搜索到对应案件的案件号、原告、原告律师、原告律师事务所、被告、被告律师、被告律师事务所 、处理该案件的法院、该案件的文书类型、案件的案由、现在所处的程序、日期、审判长

2.通过 案件号，搜索到文书title、原告、原告律师、原告律师事务所、被告、被告律师、被告律师事务所 、处理该案件的法院、该案件的文书类型、案件的案由、现在所处的程序、日期、审判长

3.通过 法院名  
3.1搜索到它所处理过的所有案件的案件名  
3.2搜索到它所处理过的所有案件的案件类型  
3.3搜索到它所处理过的所有案件的案由  
3.4搜索到在某一段时间内它所处理过的所有案件

4.搜索到某一文书类型（5种）的所有案件  
4.1搜索到在某一段时间内某种类型的所有案件

5.搜索某一案由下的所有文书  
5.1搜索到在某一段时间内某种案由的所有案件

6.搜索处于某一程序状态下的所有文书

7.搜索某时刻、某段时间里的所有案件：某段时间里的所有民事案件、 某段时间里的所有民事案件……

8.搜索某一审判长处理过的所有案件，包括：他处理过的所有民事案件、他处理过的所有刑事案件……

9.通过 原告  
9.1搜索到所有与之有关的案件、案件号、所有官司类型、所有案由  
9.2搜索到所有处理过他的案件的法院和与法院对应的审判长  
9.3搜索到某段时间内与之相关的所有官司  
9.4所有与之打官司的被告、被告律师、被告律师事务所

10. 通过被告  
10.1搜索到所有与之有关的文书title、所有案件的案件号、所有官司类型、所有案由  
10.2搜索到所有处理过他的案件的法院和与法院对应的审判长  
10.3搜索到某段时间内与之相关的所有官司  
10.4所有与之打官司的原告、原告律师、原告律师事务所

11.通过律师  
11.1搜索到他打过的所有官司  
11.2搜索到他打过的所有民事案件、刑事案件……  
11.3搜索到在某一段时间内打过的所有官司：民事案件、刑事案件……  
11.4搜索到他所在的事务所  
11.5搜索到他给代理过的所有当事人

12.通过律师事务所

12.1搜索到律师事务所打过的所有官司

12.2搜索到律师事务所打过的所有民事案件、刑事案件……  
12.3搜索到律师事务所在某一段时间内打过的所有民事案件、刑事案件……

12.4搜索到律师事务所给代理过的所有当事人

12.5搜索到律师事务所的所有律师

===================== 基本软件的安装 ===================

ElasticSearch && head插件&& ik插件  
1）安装elasticsearch： 到官网下载文件，解压即可

<https://www.elastic.co/guide/en/elasticsearch/reference/current/_installation.html>

2）安装head插件  
> cd elasticsearch-2.3.1/bin   
> ./plugin install mobz/elasticsearch-head

3）安装ik插件

1、解压文件elasticsearch-analysis-ik-master.zip，进入下载目录，执行命令：

> unzip elasticsearch-analysis-ik-master.zip

> cd elasticsearch-analysis-ik-master

2. 因为是源代码，此处需要使用maven打包，执行命令

此处需要安装maven

配置java环境：/etc/profile /.bashrc

export JAVA\_HOME=/usr/lib/jvm/java-7-openjdk-amd64

export CLASSPATH=.:$JAVA\_HOME/lib:$JAVA\_HOME/jre/lib:$CLASSPATH

export PATH=$JAVA\_HOME/bin:$JAVA\_HOME/jre/bin:$PATH

> sudo apt-get install maven

> sudo mvn package

3. elasticsearch-analysis-ik-master目录下会出现target文件夹，将target文件夹下的jar文件elasticsearch-analysis-ik-2.3.2-sources.jar复制到elasticsearch安装目录的lib目录下。

4. 将elasticsearch-analysis-ik-master/config/目录下的ik文件夹复制到elasticsearch目录config文件夹下。

4）运行  
> cd ./elasticsearch-2.3.1/bin  
> ./elasticsearch  
5）建立一个新的index，生成mapping.json文件，内容如下： (TBD: 修改使得title和law\_office可以同时也支持分析) 已完成  
{

"mappings": {

"doc": {

"properties": {

"title" : {"type": "string", "index": "not\_analyzed"},

"title\_search" : {"type": "string", "analyzer": "ik"},

"url": {"type": "string", "index": "not\_analyzed"},

"reason" : {"type": "string", "index": "not\_analyzed"},

"court" : {"type": "string", "index": "not\_analyzed"},

"date" : {"type": "date", "format": "yyyy-MM-dd", "index": "not\_analyzed"},

"doc\_type" : {"type": "string", "index": "not\_analyzed"},

"status" : {"type": "string", "index": "not\_analyzed"},

"content" : {"type": "string", "analyzer": "ik"},

"case\_id" : {"type": "string", "index": "not\_analyzed"},

"lawyers" : {

"properties": {

"name": {"type": "string", "index": "not\_analyzed"},

"law\_office": {"type": "string", "index": "not\_analyzed"},

"law\_office\_search": {"type": "string", "analyzer": "ik"}

}

},

"chief\_judge": {"type": "string", "index": "not\_analyzed"},

"accuser": {"type": "string", "index": "not\_analyzed"},

"accussed": {"type": "string", "index": "not\_analyzed"}

}

}

}

}

7）进入mapping文件所在的目录，导入该mapping.json文件到ELS中  
> cd ./mappings  
> curl -XPUT '[http://localhost:9200/court](http://localhost:9200/court/doc/_mapping" \t "/home/liuxiaoxiao/Documents\x/_blank)' --data-binary @mapping.json

8）使用head查看ELS系统  
> [http://localhost:9200/\_plugin/head/](http://localhost:9200/_plugin/head/" \t "/home/liuxiaoxiao/Documents\x/_blank)  
  
9）安装kibana，下载最新的版本，解压至与elasticsearch目录所在相同父目录下  
> tar -xzvf kibana-4.5.0-linux-x64.tar.gz

10）需要更改kibana的config文件 （使用缺省的即可）  
> vi kibana-4.5.0-linux-x64/config/kibana.yml  
将   
# server.host: "0.0.0.0"  
改为  
server.host: "localhost"  
  
11）启动kibana  
> cd kibana-4.5.0-linux-x64/bin  
> ./kibana

12）使用 [http://localhost:5601](http://localhost:5601/" \t "/home/liuxiaoxiao/Documents\x/_blank) 查看  
（1）settings >> indices 页面下更改index名称为court

========= 一个Court爬虫的部署和运行（前置条件） ========

1) 系统架构环境要求  
Ubuntu 14  
Python 2.7  
MySql >= 5.6  
RabbitMQ >= 3.6  
ElasticSearch >=2.3

2) 安装MySQL，用于保存断点信息： id | court\_id |page\_num | status  
> sudo apt-get install mysql-server mysql-client  
2.1) 创建用于断点的库和表  
CREATE DATABASE IF NOT EXISTS `openlaw` DEFAULT CHARACTER SET utf8 COLLATE utf8\_general\_ci;

USE `openlaw`;

DROP TABLE IF EXISTS `breakpoint`;

CREATE TABLE `breakpoint` (

`id` int NOT NULL AUTO\_INCREMENT ,

`court\_id` varchar(255) NULL ,

`page\_num` int NULL ,

`status` varchar(2) NULL ,

PRIMARY KEY (`id`)

);  
或直接导入

> mysql -uroot -p数据库密码 < openlaw.sql

2.2)将断点信息court\_id里的法院ID和 page\_num导入数据库

进入save\_link\_to\_db.py所在目录

修改密码

> python save\_link\_to\_db.py

3） RabbitMQ 安装与运行（sudo /etc/init.d/rabbitmq-server start），端口？？？15672  
> sudo apt-get install rabbitmq-server  
查看rabbitmq状态  
> sudo /etc/init.d/rabbitmq-server status  
安装 rabbitmq management console (see [https://www.rabbitmq.com/management.html](https://www.rabbitmq.com/management.html" \t "/home/liuxiaoxiao/Documents\x/_blank) )  
> sudo rabbitmq-plugins enable rabbitmq\_management

> sudo service restart rabbitmq  
打开 [http://localhost:15672/](http://localhost:15672/" \t "/home/liuxiaoxiao/Documents\x/_blank) 查看队列状态  
缺省账号和密码 ： guest / guest  
  
Q：如何查看RabbitMQ使用了哪些端口？ （15672是http管理端口）  
  
A:   
  
（1）sudo nmap -p 1-65535 localhost  
4369/tcp  open  epmd  
5672/tcp  open  amqp  
（2）epmd -names  
epmd: up and running on port 4369 with data:  
name rabbit at port 54795  
  
4）安装python所需的API包  
>  sudo apt-get install python-dev build-essential libxml2 libxml2-dev libxslt1-dev libmysqlclient-dev  
>  sudo pip install pyquery selenium pika mysql-python elasticsearch  
用于隐藏firefox窗口的包  
>  sudo apt-get install phantomjs

测试  
进入控制台，输入python回车，进入python交互界面  
依次输入以下命令，查看是否报错，无报错则说明python环境搭建成功  
import pika  
import redis  
import MySQLdb  
from selenium import webdriver  
from pyquery import PyQuery  
from elasticsearch import Elasticsearch

======== 一个Court爬虫的部署和运行（python 程序） =========

文件    配置    描述  
settings.ini     //需修改数据库密码  
配置文件  
manager.py      
TBD  
sender.py      
爬取指定court链接下的所有judgement链接，放入rabbitmq队列中  
worker.py      
爬取rabbitmq中提取judgement链接爬取、解析，并发送到ElasticSearch中  
1） 初始化种子节点（手工完成 - TBD）

将court\_id里的法院ID和 page\_num导入数据库

进入save\_link\_to\_db.py所在目录

> python save\_link\_to\_db.py  
进入数据库查看结果  
2） 运行manager.py  
  可以使用 RabbitMQ的管理端口 [http://localhost:15672/](http://localhost:15672/" \t "/home/liuxiaoxiao/Documents\x/_blank)  查看数据

======== 检索针对问题的结果 （请参见最开始的问题列表）========

使用 [http://localhost:9200/\_plugin/head/](http://localhost:9200/_plugin/head/" \t "/home/liuxiaoxiao/Documents\x/_blank)   
any request >> query 页面中输入检索query，如下：  
  
（1）通过title，搜索到原告、被告、律师、律师事务所

{

"query": {

"match": {

"title\_search": "普实科技"

}

},

"\_source": [

"lawyers.law\_office",

"lawyers.name",

"title"

],

"size": 100

}  
（2）搜索该法院处理过得所有某类型的案件

{

"query": {

"filtered": {

"query": {

"bool": {

"must": [

{

"match": {

"court": "河北省乐亭县人民法院"

}

},

{

"match": {

"doc\_type": "民事裁定书"

}

}

]

}

}

}

},

"\_source": [

"title",

"doc\_type",

"court"

],

"size": 100

}

1. 搜索到在某一段时间内该法院所处理过的所有案件

{

"query": {

"filtered": {

"query": {

"match": {

"court": "河北省乐亭县人民法院"

}

},

"filter": {

"range": {

"date": {

"gte": "2016-02-10",

"lte": "2016-03-01"

}

}

}

}

},

"\_source": [

"date",

"court",

"doc\_type",

"title"

],

"size": 100

}

1. 搜索到在某一段时间内某种类型的所有案件

{

"query": {

"filtered": {

"query": {

"match": {

"doc\_type": "刑事判决书"

}

},

"filter": {

"range": {

"date": {

"gte": "2016-02-10",

"lte": "2016-03-01"

}

}

}

}

},

"fields": [

"date",

"court",

"doc\_type",

"title"

],

"size": 100

}

1. 搜索某段时间、某一审判长处理过的某类型案件

{

"query": {

"filtered": {

"query": {

"bool": {

"must": [

{

"match": {

"chief\_judge": "刘海虹"

}

},

{

"match": {

"doc\_type": "民事裁定书"

}

}

]

}

},

"filter": {

"range": {

"date": {

"gte": "2016-02-10",

"lte": "2016-03-01"

}

}

}

}

},

"fields": [

"date",

"chief\_judge",

"doc\_type",

"title"

],

"size": 100

}

1. 通过 原告搜索所有与之打官司的被告、被告律师、被告律师事务所

{

"query": {

"match": {

"accuser": "北京博华世纪咨询有限公司"

}

},

"\_source": [

"lawyers.law\_office",

"lawyers.name",

"title",

"accuser",

"url",

"accussed"

],

"size": 100

}

1. 查询该律师事务所的律师打过的所有官司

{

"query": {

"match": {

"lawyers.law\_office": "河北章理桥律师事务所"

}

},

"\_source": [

"lawyers.name",

"lawyers.law\_office"

],

"size": 100

}