

# OpenSearch 典型行业搜索解决方案

## 之 垂直社区

### 目录

OpenSearch 典型行业搜索解决方案 .....	1
之 垂直社区 .....	1
1、概述 .....	2
2、解决方案 .....	2
2.1 创建应用 .....	4
2.1.1、配置数据表 .....	4
2.1.2、配置搜索属性 .....	5
2.2、数据上传 .....	6
2.3、搜索效果配置 .....	8
2.3.1 搜索相关性配置 .....	9
2.3.2 搜索结果摘要 .....	11
2.3.3 查询分析 .....	11
2.3.4 下拉提示 .....	12
2.3.5 热词搜索 .....	13
2.4、搜索 .....	13
3、其他日常功能项 .....	15
3.1 错误日志 .....	15
3.2 索引重建 .....	15
3.3 数据统计 .....	15

# 1、概述

垂直社区大部分都是以内容为主，包括网站本身的资讯、介绍以及服务型论坛，我们以[威锋网](#)为例进行说明。威锋网自建立之日起一直是最具人气的中文 iPhone 社区，给广大 iPhone 爱好者提供了一个自由交流，探讨，学习的平台，目前已经拥有锋科技、测评、新闻、论坛、商城、游戏等多个频道，为 iPhone 在中国的应用及普及发挥了领军作用。

本教程会带领大家一步一步操作看如何实现一个垂直门户类型的搜索。

# 2、解决方案

首先我们来看下目前威锋网搜索页面概览图，从下图可以看出，搜索结果页可以简单分为以下 7 个部分（不算广告部分☺）。接下来我们就来逐步分析下如何使用 **OpenSearch** 实现以上 7 项功能。

1

威锋网 FENG.COM

iphone6s

2

全部 论坛 新闻 锋科技 评测 威智网 安锋网 威锋游戏

3

搜索热词：微信 多开 越狱 4G 锋友堂

4

搜索结果 1102

皮肤癣有治好的案例吗?

牛质皮肤癣治疗医院(上海..

办厂小生意 月赚5~8万..

江城医院

办厂好项目

6

搜索排序

相关度

发帖时间

7

搜索时间

全部时间

一天内

一周内

一个月内

三个月内

半年内

一年内

二年内

三年内

新闻

台积电董事长夫人:三星版iPhone 6s很耗电

一周内 leo\_ng weiphone

威锋网 11月8日消息 前阵子 iPhone 6s 系列所谓的“芯片门”事件到现在已经渐渐平静下来, 不过近日台积电董事长夫人的一番言论恐

锋科技

未来垂直整合哪家强? 苹果微软或谷歌

昨天 一起晒太阳 weiphone

就硬件和软件垂直整合方面来说, 苹果这边已经做得非常纯熟, 比苹果 A9 处理器为例子, iPhone 6S A9 处理器在 Geekbench 3 里的跑分为单

光棍节送 iPhone 6s 《逐鹿天下》极限测试 关注微信抢壕礼

一周内 8295 威游客服 论坛

威锋游戏年度重磅力作《逐鹿天下》, 极限测试11月9日上午11:00正式开启, 关注微信公众帐号, 并扫码转发文章, 即有机会获得iPhone 6s玫瑰金、苹果设计秘籍、创新智能灯泡等给力礼品, ...

新闻

福布斯:4英寸iPhone重出江湖是营销策略

一周内 SvvL 网易科技

如果说电池方面还说的过去, 屏幕的分辨率就说不过去了, iPhone 6s的屏幕分辨率为326 PPI (像素/英寸), 而较大的iPhone 6s为401

锋科技

重回智能手机领域的巅峰? HTC需要做到5件事

5

建议开始前先粗览一遍[帮助文档](#), 本文假设前提: 您已经了解了 OpenSearch 的基本功能; 您的场景适合使用 OpenSearch; 您接受 OpenSearch 的服务条款。关于 OpenSearch 的基本知识, 不在本文讨论范围内, 如有疑问, 可以到[论坛](#)或者[工单](#)上提问。

要实现上图中的搜索功能只需要在 OpenSearch 中进行 4 步：

## 2.1 创建应用

一个应用即为一个独立的搜索服务，简单理解就是一个搜索框。首先进入到 [OpenSearch 控制台](#)，创建应用：



点击下一步。

### 2.1.1、配置数据表

OpenSearch 是结构化的搜索服务，首先我们需要定义表的结构，这里的字段一般是要用来查询、过滤、统计、展示的字段，其他无关字段无需进入 OpenSearch 中。以威锋网为准，我们定义应用结构如下：（本文主要是为了演示，只选取了必要字段，具体需要根据自己的场景来添加、修改或者删除）

字段名称	主键	字段类型	内容转换	选择外键主键	操作
1 id 主键id	*	INT	+	+	删除
2 type_id 频道id, 比如锋科技、新闻、论坛, 可以通过数字字段做频道内的查询			+	+	删除
3 title 标题		TEXT	+	+	删除
4 body 正文		TEXT	HTMLTagRemover   body	+	删除
5 url 点击url		TEXT	去HTML标签, 优化展示效果	+	删除
6 aid 作者id, 用来关联作者表信息		INT	+	author x +	删除
7 create_timestamp 创建时间		INT	+	+	删除
9 tags 网页标签, 支持多个		TEXT_ARRAY	+	+	删除
12 hits 点击数		INT	+	+	删除
13 replies 回复数		INT	+	+	删除
14 source 新闻来源		TEXT	+	+	删除

字段名称	主键	字段类型	内容转换	选择外键主键	操作
1 aid 作者id, 与post中的aid相关联	*	INT	+	+	删除
2 name 作者名字		TEXT	+	+	删除

+ 添加表

保存

注意这里的关联关系，主表字段只能关联附表主键。

这一步非常关键，很多同学并不知道如何配置，是否需要与数据库中字段一致，答案是不需要的，这里的字段均来自搜索需求，比如产品上需要搜索和展示的有**标题**、**正文**、**频道**（新闻、锋科技还是论坛）、**时间**、**作者**、**来源**、**图片 url**；排序和过滤的有**时间**；点击需要有**url**；排序还需要考虑**浏览量**、**点击量**等，所以这样就产出了 OpenSearch 的表结构，再根据字段内容选择合适的类型即可。



这里之所以使用了多表，是因为在实际业务中文章与作者不会放在一张表中做存储，为了方便用户按原业务表推送数据，所以系统也支持了多表功能。当然，您也可以放在一张表，寻找最适合自己的方案即可。

如果配置错误也没有关系，系统提供了动态修改应用结构功能，您可以随时调整，不用担心。

## 2.1.2、配置搜索属性

多表只是为了方便推送数据，进入搜索引擎中的数据都是打平成大宽表的数据，所以搜索的时候不再有表的概念，这一步就来定义各个字段的搜索属性。这里是否要勾选需要根据实际搜索场景产品需求来定，另外细分类型非常重要，尤其是 TEXT 字段，[不同的子类型代表不同的分词效果](#)，需要慎重选择。选择自己想要的，后续经过测试再不断的调整优化。

字段名称	类型	选择细分类型	可搜索	索引到	可过滤	可展示	操作
id	INT32		<input checked="" type="checkbox"/>	id	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
type_id	INT16		<input type="checkbox"/>	type_id	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	删除
title	MWS_TEXT		<input checked="" type="checkbox"/>	title compound +	<input type="checkbox"/>	<input checked="" type="checkbox"/>	删除
body	MWS_TEXT		<input checked="" type="checkbox"/>	compound +	<input type="checkbox"/>	<input checked="" type="checkbox"/>	删除
url	STRING		<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	删除
aid	INT32		<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	删除
create_timestamp	INT64		<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	删除
tags	STRING_ARRAY		<input checked="" type="checkbox"/>	tags +	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	删除
hits	UINT32		<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	删除
replies	UINT32		<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	删除
source	MWS_TEXT		<input checked="" type="checkbox"/>	compound +	<input type="checkbox"/>	<input checked="" type="checkbox"/>	删除
name	MWS_TEXT		<input checked="" type="checkbox"/>	author compound +	<input type="checkbox"/>	<input checked="" type="checkbox"/>	删除

取消上一步下一步

这里再重点说下可搜索与索引到，一个源段可以索引到多个索引字段上（如 title 索引到 title、compound 上），多个原字段也可以索引到同一个索引字段上（如 title、body、source、name 索引到 compound 上），具体如何配置同样是根据搜索需求来。产品上需要综合搜索，比如只要标题、正文、来源、作者中有一项包含都希望搜索出来，那么可以将上述几个字段索引到同一个索引字段上，搜索的时候使用 `query=compound:'iphone'`，那么标题、正文、来源、作者中有一项包含'iphone'都会被召回。另外，产品上还希望提供按照标题搜索，那么可以单独将 title 索引到 title 上，那么 `query=title:'iphone'` 只会召回标题中包含 iphone 的文章，正文、来源、作者中包含则不会被召回。

## 2.2、数据上传

这里有两种选择，一种是后续通过 API/SDK 方式上传（具体参考产品文档的 API/SDK 参考手册），一种是通过数据源的方式导入，目前支持 RDS\ODPS\OSS，这里以 RDS 源为例，其他类似（具体参考产品文档[数据源配置](#)）

### 数据自动同步

配置数据源后数据将自动同步至云端  
支持多种数据源

### 稍后手动上传

稍后配置数据源，或者通过API、  
SDK、上传界面上传文档

post

数据来源: RDS

1 连接数据

\* 实例ID: rdsfarvnrinrvnnn

\* 数据库名: bbs

\* 表名: my\_post

过滤条件:

\* 用户名: opensearch

\* 密码: .....

☒ 数据自动同步

☒ 授权OpenSearch读取RDS数据

连接数据

2 字段映射

2 字段映射

	目标字段名称	内容转换	源字段名称	操作
1	id (主键)	+	<div>pid</div>	删除
2	title	+	<div>title</div>	删除
3	body	+	<div>body</div>	删除
4	aid	+	<div>aid</div>	删除
5	create_timestamp	+	<div>create_timesta</div>	删除
6	hits	+	<div>hits</div>	删除
7	type_id	+	<div>cat_id</div>	删除
8	tags	MultiValueSplitter	<div>cat</div>	删除
+ 添加数据				

将需要同步的字段一一添加上RDS字段映射关系方便做数据同步使用。

保存取消

继续点击下一步，检查无误后即创建成功，按照提示进行应用激活以及索引重建操作（第一次 RDS 需要导入全量，后续走增量即可，无需再索引重建）。

激活应用

返回应用列表

应用名称：weifeng\_example

自定义付费容量

免费容量

实例租用

存储容量: 0.3 12 48 100 1000 2000 0.3 GB

搜索请求

QPS流量: 6 20 50 100 400 1000 1万 6 次/秒

根据实际需要调整，以此来做收费因子（右侧有计费说明），不要调太高，也不能调太低，低于实际值会被限流。

配置详情

实例租用 (元/小时): ¥0.011 ¥0.014

搜索请求 (元/小时): ¥0.001

文档更新 (元/GB): ¥0.75 ¥1

索引重建 (元/GB): ¥0.375 ¥0.5

激活取消

总费用 = [实例租用费用 + 搜索请求费用] + [文档更新费用 + 索引重建费用]

价格总览

## 2.3、搜索效果配置

经过上面的做法，数据已经上传完毕，可以搜索了。在搜索之前我们还需要进行下相关搜索效果的配置，已保证获取更优的展示效果。目前 OpenSearch 主要提供如下功能的支持，让我们来分别看一下：



## 2.3.1 搜索相关性配置

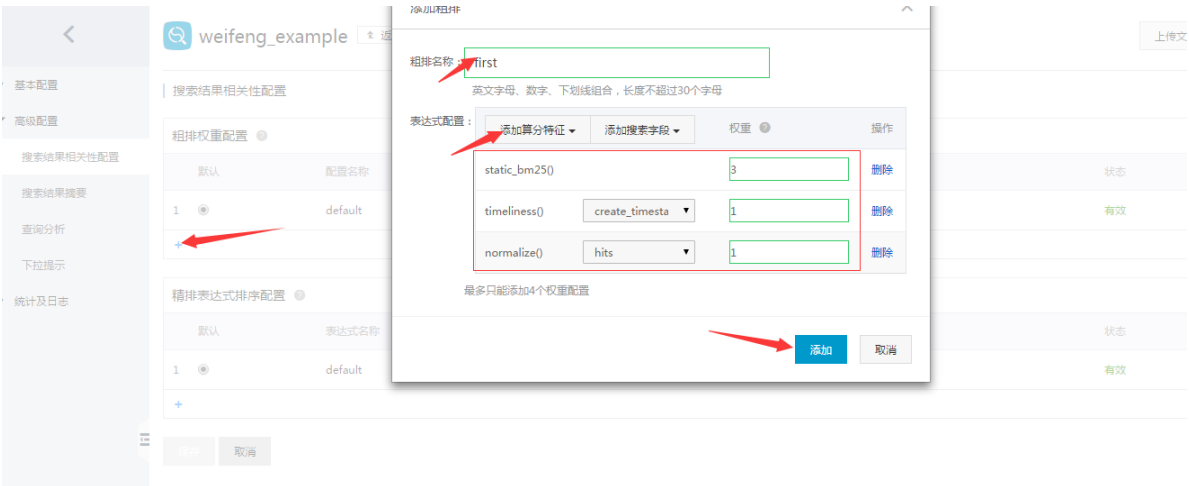
搜索引擎找到查询数据后，会对查询到的文档进行排序，并按照排序后的顺序返回给用户，用户可以采用翻页的方式来进行数据展示。目前 OpenSearch 支持 [sort 多维排序](#)，在本文的案例中，我们需要使用到系统相关性表达式的功能。

目前系统提供的相关性开放了两轮的排序规则：[粗排与精排](#)。粗排即是海选，将符合条件的文档进行粗排表达式计算得分，然后选取 TOP N 再进行精排表达式计算，最终按照精排得分排序后返回结果。从上面的处理逻辑来看，粗排因为涉及文档多，所以需要尽量简单（匹配的文档越多对于粗排表达式要求越高，否则会有好的文档进不了精排，无法展示的情况出现），将质量高的文档挑选出来即可，然后再根据精排来详细算分，做排序。



这里粗排表达式我们选取了三部分，三者之间的权重为 3:1:1（对于关键词查询，最重要的还是文本匹配度）：

- 1, static\_bm25: 简单文本相关性，用来计算查询词与文档内容的匹配程度；
- 2, timeliness(create\_timeliness): 时效分，用来看文档的新旧程度；
- 3, normalize(hits): 点击数，用来看文档的热门程度，normalize 用来做归一化。



精排表达式如下，这里主要有三部分因素考虑：

- 1, 对于一个文本匹配的场景，首先要考虑文本匹配的程度，相关性越高分值越高；这里的 text\_relevance 也包含三方面内容的考虑：bm25、proximity 以及 match\_doc。
- 2, 另外呢，对于威锋网做资讯为主的场景来讲，时效性也很重要，同样搜索 iphone, iphone6

的结果要比 **iphone4** 排在前面；

3，最后文档的点击、评论等因素也标识着一篇文档的质量好坏，所以也可以把这部分内容加入到排序因子中来考虑。

确定主体排序因子后，根据三者之间值域以及重要程度进行合适的权重调整即可，这部分工作需要实际搜索效果中不断调试。目前相关性调试最常见的 [ABTest](#)，寻找 50% 的 TopQuery、20% 长尾 Query（即结果数非常少的查询）、30% 常规 Query（普通查询）进行效果人工排查，发现 badcase 后分析原因，然后调整排序表达式，以此不断优化，达到最佳效果。



设置成功后，别忘了点击保存按钮。一般来讲，一个新的排序表达式设置后可以先通过 API 接口显示指定 `first_formula_name` 以及 `formula_name` 来进行效果测试，并根据测试结果不断调整，效果满意后，可以将其设置为默认即可。一旦设置为默认，则立即对线上查询起作用，无需发布变更，非常方便。

此外，对于较为复杂的搜索场景来讲，可以设置多个排序表达式以满足不同的查询场景。如下例中展示，搜索文章的时候可以使用 `second` 表达式，搜索作者的时候可以使用 `name` 的表达式。

**粗排权重配置**

默认	配置名称	权重配置
1 <input checked="" type="radio"/>	default	static_bm250
2 <input type="radio"/>	first	static_bm250*3+timeliness(create_timestamp)+normalize(hits)
+		

**精排表达式排序配置**

默认	表达式名称	表达式
1 <input type="radio"/>	default	
2 <input checked="" type="radio"/>	second	text_relevance(title)*3+text_relevance(body)*0.9 +if(text_relevance(t...
3 <input type="radio"/>	name	text_relevance(name)
+		

**保存** **取消**

## 2.3.2 搜索结果摘要

摘要顾名思义就是文章的简介，一般来讲，文章内容都会比较长，在搜索结果中不可能全部展示，这时候就需要设置摘要信息，您可以配置改摘要的片段个数、片段长度以及片段连接符，片段选取跟查询词有关，通常选取关键词所在的一个或者多个句子。

同时，OpenSearch 也提供了飘红功能，用来做查询词的高亮显示。

**weifeng\_example** [返回应用列表](#) **上传文件** **搜索测试**

**基本配置**  
**高级配置**  
**搜索结果相关性配置**  
**搜索结果摘要**  
**查询分析**  
**下拉提示**  
**统计及日志**

**搜索结果摘要**

字段	每个摘要片段的长度	飘红标签	多个摘要片段连接符	片段数量
1 title	60	em	—	1
2 body	120	em	—	3
3 name	30	em	—	1
+				

**保存** **取消**

## 2.3.3 查询分析

查询分析主要是为了做查询优化的，他存在的目的就是为了深度解析查询词意图，并智能改写，比如一个查询地址：杭州市西湖区文一西路 969 号，能自动识别并改写成 city:'杭州市' AND district:'西湖区' AND address:'文一西路 969 号'。当然这里只是举个例子，目前还未提供该功能，现在支持 4 部分：停用词（不参与搜索，如特殊符号和一些无意义的副词等）、同义词（如'肯德基'同义'KFC'等）、词权重（对一些不重要的词做 RANK，扩大召回并提升搜索效果）、拼写纠错（如'百毒'纠错成'百度'等）。

您可以视具体的查询情况来配置上述功能，目前提供的还是系统自带词表，后续会开放用

户自定义，一些较特殊的场景（如景点名、影视名等）可以自己配置适合的改写内容。



配置完毕后，可以通过 API 指定 **qp** 参数名字来测试（同上述排序表达式测试流程类似），观察改写效果，测试通过后，添加至上线，即可以对线上查询立即生效，非常方便。



### 2.3.4 下拉提示

下拉提示目前支持多个字段（最多 3 个）的语义单元抽取，目前抽取逻辑非常复杂，如果您发现有符合预期的情况，可以与我们反馈，我们会视情况来不断调整优化。

目前下拉提示对于抽取出来的语义单元进行拼音全拼与前缀等模糊搜索效果，可以很好的达到减少用户输入并规范用户输入提高搜索效果的效果。



配置完毕后生效下拉提示，即可查看效果。

下拉提示查看如何使用

规则名称	来源字段	推荐词条	黑名单	状态	操作
1 suggest	title , body	2	0	未生效	编辑   删除
+					

生效下拉提示

效果测试 ▼ 

查看UI源代码

生效需要一段时间，待状态为“已生效”后，可以再这里查看实际提示效果

效果如下：

效果测试 ▼ suggest

查看UI源代码

推荐词条数据排在最前面

weifeng

效果测试 ▼ suggest

查看UI源代码

拼音简写搜索提示

苹果已在测试ios 10了你是否还有期待

苹果或许每个月能从你的身上赚到153美元

苹果 产品线

苹果 硬件

苹果 硬件业务

## 2.3.5 热词搜索

系统暂未提供，您可以先自行简单处理，将 `accesslog` 中的搜索关键词进行次数统计，取 TOPN 得结果进行展示。

最上面效果图中第 3 部分功能即由此实现。

## 2.4、搜索

系统提供了丰富的搜索语法：[查询](#)、[过滤](#)、[统计](#)、[排序](#)、[打散](#)等。

效果图中第 1 部分功能即为查询语法，功能 2 为统计语法，功能 4、5 为查询结果，功能 6 位排序语法（默认为相关性），功能 7 为过滤语法，接下来我们通过搜索测试页面来形象的展示下。

应用： weifeng\_example

query=compound:'iphone6s' 功能1：查询compound索引中的组合查询

搜索

子

功能7：根据时间做范围过滤

功能2：对查询结果做频道及频道结果数统计

功能5：按频道做打散，避免同一频道结果过于集中

filter= create\_timestamp>1430265607

×

aggregate= group\_key:type\_id,agg\_fun:count()

×

distinct= dist\_key:type\_id,dist\_count:1,dist\_times:100

config= start:0,hit:20

×

sort= -create\_timestamp 功能6：指定排序策略，不填为根据相关性排序

功能5：对搜索结果做翻页控制

[http://opensearch-cn-hangzhou.aliyuncs.com/search?query=query%3Dcompound%3A%27iphone6s%27%26%26filter%3Dcreate\\_timestamp%3E1430265607%26%26aggregate%3Dgroup\\_key%3Atype\\_id%26%26agg\\_fun%3Account%28%29%26%26distinct%3Ddist\\_key%3Atype\\_id%26%26dist\\_count%3A1%26%26dist\\_times%3A100%26%26config%3Dstart%3A0%26%26chit%3A20%26%26sort%3D-create\\_timestamp&index\\_name=weifeng\\_example&client\\_id=6100914425660095&nonce=3158ff449cde7213d2b7875e718bfe7.1448272864&sign=ebc3ec83a859332bee4d2e8c7974c685](http://opensearch-cn-hangzhou.aliyuncs.com/search?query=query%3Dcompound%3A%27iphone6s%27%26%26filter%3Dcreate_timestamp%3E1430265607%26%26aggregate%3Dgroup_key%3Atype_id%26%26agg_fun%3Account%28%29%26%26distinct%3Ddist_key%3Atype_id%26%26dist_count%3A1%26%26dist_times%3A100%26%26config%3Dstart%3A0%26%26chit%3A20%26%26sort%3D-create_timestamp&index_name=weifeng_example&client_id=6100914425660095&nonce=3158ff449cde7213d2b7875e718bfe7.1448272864&sign=ebc3ec83a859332bee4d2e8c7974c685)

找到相关结果 3 条（用时0.012202秒）功能4：查询命中结果数及耗时

id : 3

type\_id : 3

title : 瞧BB 苹果这几年...以及未来几年走势

body : 而那时候 <em>iPhone</em> 神奇的触摸...分2个系列 你可以说苹果还是稳扎稳打，但也说明 苹果也害怕了...不自信了...不敢孤注一掷，因为在 <em>6</em> 与...

url : http://bbs.feng.com/read-htm-tid-10073689.html

aid : 3

create\_timestamp : 1449665607

tags : 讨论 iphone

hits : 24203

replies : 163

source : 威锋网

功能5：搜索结果。可以根据type\_id分别对各频道结果样式进行渲染

API 语法使用范例（根据需要取一项或者多项组合来查询）：

query=compound:'iphone'	找到文章中（标题、正文、作者、来源任一个）中包含 <b>iphone</b> 的文章
query=title:'iphone'	找到标题中包含 <b>iphone</b> 的文章
query=name:'iphone'	找到名字中包含 <b>iphone</b> 的作者
query=title:'iphone' AND type_id:'1'	找到标题中包含 <b>iphone</b> 的并且为新闻频道的文章
filter=create_timestamp>14234873471	找到近 <b>xx</b> 天的文章
filter= create_timestamp>14234873471 AND hits>10000	找到近 <b>xx</b> 天且点击超过 <b>1</b> 万的文章
aggregate=group_key:type_id,agg_fun:count()	统计搜索结果中的频道并计算每个频道出现的文章数
sort=-RANK	按照相关性排序（相关性即为搜索排序表达式）
sort=-create_timestamp	按照时间由新到旧排列
distinct=dist_key:type_id,dist_count:1,dist_time:100	对搜索结果按照频道进行打散，各个频道每次取 <b>1</b> 个结果，一共取 <b>100</b> 次，剩下的文档排在后面
config=start:0,hit:20,format:xml	用来控制翻页，或者返回结果格式等

调试过程中会遇到一些报错请，请根据错误码查找[文档找到错误原因](#)进行修改即可。测试过程需要根据实际情况对排序表达式、应用结构不断进行调整，以便达到满意的效果。

测试完毕后，便可以调用 [API 语法](#)（OpenSearch 提供了 [php SDK](#)、[java SDK](#)）进行实际的搜索结果页开发了。这里就不做深入讨论，论坛上有篇《[OpenSearch phpsdk 从初级到高级的详细教程](#)》各位可以参考，有问题可以联系我们。

## 3、其他日常功能项

### 3.1 错误日志

数据处理过程是异步的，数据成功上传到 OpenSearch 后，会经过若干处理模块最后进入引擎做索引构建（具体查看 [OpenSearch 基础架构](#)），在该处理过程中可能会因为权限、类型、插件等各种原因导致处理失败，这部分失败会在控制台错误日志处有详细展示，您可以根据 [错误码和错误提示](#) 及时修正数据，重新上传。

基本数据	错误日志				
高级配置	选择时间范围：2015-11-23 至 2015-11-23 重置 您当前有1条未读错误信息				
统计及日志					
错误日志	时间区间	表名	错误文档数	错误信息	错误详情
数据统计	2015-11-18 04:10:02 - 2015-11-18 04:12:01	port	2	主键索引不存在 (错误码: 7301)	原始文档: CMD=ADD id= body=946562876528 title=test 原始文档: CMD=ADD id= body=946562... <a href="#">更多</a>

### 3.2 索引重建

用户数据上传后会先保存在离线 HBASE 中，然后经过相关处理后进行索引构建到引擎，供用户查询。如果有应用结构变化，那么需要将数据以新的结构定义重新进行索引构建。这个构建过程是流式的（如控制台索引重建上的进度条所示），新数据会逐渐替换老数据，所以在索引重建过程中会出现新老数据共存的现象，索引重建完毕后新数据完全生效。

在这个过程中，搜索服务不受影响，但是因为新老数据共存，所以需要保证查询的可用性

1，如果是新增字段，那么原字段的逻辑都是兼容的，建议在索引重建完后再访问新字段

2，如果是删除字段，建议在索引重建前先去掉对已删除字段的访问；

3，如果是修改操作（如类型），那么需要保证查询对于修改前后的都是通用的。否则，险起见，建议重新创建新应用，待新应用数据完成后，再将搜索切换到新应用上，删除老应用。

注意：RDS 源用户只要第一次索引重建将全量数据导入即可，后续系统会自动通过 binlog 做增量数据更新，无需定时或者手动重建。

### 3.3 数据统计

应用的访问 PV、文档数的运营数据指标都可以通过数据统计来展示，方便随时掌握应用变化情况。