

开放搜索服务OpenSearch

SDK参考手册



SDK参考手册

JavaSDK手册

版本说明

前言

简介

提供java语言对API接口的封装（应用管理、数据处理、搜索、错误日志等）。

版本说明

当前版本:V2.1.2

maven地址：<http://search.maven.org/#search%7Cga%7C1%7Caliyun-sdk-opensearch>

```
<dependency>
<groupId>com.aliyun.opensearch</groupId>
<artifactId>aliyun-sdk-opensearch</artifactId>
<version>2.1.2</version>
</dependency>
```

历史版本

V2.1.1

新功能：

- 在search、index、doc类中调用client.call方法之后可以调用此接口获得本次操作发送的请求串；
- 在client的构造函数中添加对 请求超时时间timeout和连接超时时间conn_timeout的设置；
- sdk发送的请求支持gzip方式；
- 实例化client时可以设置sdk支持的最大连接数
- 在sdk发送的请求的user-agent中加入sdk的类型和版本信息；
- Search类的config子句支持rerank_size属性；

兼容性：

- 在实例化client时通过设置debug=true输出调试信息的方式已失效，2.1.1版本SDK中默认记录上次请求串，可以在CloudsearchSearch、CloudsearchIndex、CloudsearchDoc类中通过调用getDebugInfo来获取。
- 原config子句中设置startHit、hit和format三个属性的方法setStartHit()、setHits()、setFormat()失效，2.1.1版本中通过设置 configMap来设置config子句的属性。

使用教程

使用教程

准备工作

获取用户AccessKeyId和秘钥(secret)

用户可以使用阿里云的账号登录本系统，在登录完成后，点击“ACCESSKEY管理”可以查看您的Access Key ID (AccessKeyId) 和 Access Key Secret (secret)。也可以在阿里云官网，点击“用户中心>我的服务>安全验证”即可到ACCESSKEY管理中心。

将SDK添加到项目中

使用OpenSearch SDK有两种方式：

- 1. 下载SDK源码包，在下载中心下载最新版的JAVA SDK到本地，并import 到您的工作目录中。
- 2. 引入OpenSearch SDK依赖，通过maven二方库依赖的方式将opensearch的sdk加入到自己的项目中。

```
<dependencies>
<dependency>
  <groupId>com.aliyun.opensearch</groupId>
  <artifactId>aliyun-sdk-opensearch</artifactId>
  <version>2.1.2</version>
</dependency>
</dependencies>
```

创建应用

通过控制台也可以完成创建应用的操作，这里介绍一下如何使用SDK实现。这里使用import SDK的方式，使用上面获取的AccessKey和Secret实例化一个Opensearch Client(下面的操作里将继续使用如下的client),然后通过CloudsearchIndex类来创建一个应用。具体代码如下：

```
import com.opensearch.javasdk.*;

String accesskey= "您的阿里云的Access Key ID";
String secret = "阿里云 Access Key ID 对应的 Access Key Secret";
String appName = "您要创建的应用名称";

Map<String, Object> opts = new HashMap<String, Object>();

// 这里的host需要根据访问应用详情页中提供的API入口来确定
opts.put("host", "http://opensearch-cn-hangzhou.aliyuncs.com");
CloudsearchClient client = new CloudsearchClient(accesskey, secret, opts, KeyTypeEnum.ALIYUN);
CloudsearchIndex app = new CloudsearchIndex(appName, client);

// 您可以根据自己的需求来选择相应的模板，根据模板名称创建应用；创建自定义结构时您需要在网站中先创建好应用结构并保存成模板，然后再通过SDK使用模板名称进行创建。
System.out.println(app.createByTemplateName("template_name"));
```

如果返回结果中status字段为OK表示应用创建成功，否则表示创建失败，同时会提供相应的错误码和错误信息。可以通过下面的方法查看当前应用的状态

```
System.out.println(app.status());
```

上传文档

OpenSearch的文档是一个json类型的字符串，结构如下：

```
[
  {
    "fields" : {...},
    "cmd" : "...",
  }
  ...
]
```

一条文档是由fields字段和cmd字段构成的一个结构体，其中fields字段内包含文档的核心数据，cmd表示针对此条文档所做的操作，对文档的操作包括添加(add),更新(update)和删除 (delete) 三种。一段文档示例如下：

```
[
  {
    "fields" : {
      "id" : "0",
      "summary" : "广大中小企业都有各种结构化的数据需要进行检索，目前一般采用数据库本身提供的搜索功能或者利用open source的搜索软件搭建，这样的做法不但会消耗网站本身的资源，性能也会很容易成为问题，而且相关性通常也不够好。我们的产品的目的是要利用阿里云先进的云计算和搜索技术向广大中小企业提供低成本，高质量，高性能，可定制的数据搜索解决方案。本项目和云搜索的通用解决方案目标略有不同，主要区别为本项目主要针对用户的结构化数据进行搜索，云搜索的通用解决方案则主要是针对网页型数据为处理对象。"
    },
    "cmd" : "ADD"
  },
  {
    "fields" : {
      "id" : "1",
      "summary" : "云搜索( Cloud Search Engine), 是运用云计算( Cloud Computing)技术的搜索引擎，可以绑定多个域名，定义搜索范围和性质，同时，不同域名可以有不同UI和流程，这个UI和流程由运行在云计算服务器上的个性化程序完成。作为新型搜索引擎，与传统搜索引擎需要输入多个关键字不同的是，用户可以告诉搜索引擎每个搜索关键字的比重，每个搜索关键字都被置于“搜索云”中，并用不同大小，粗细的字型区分。"
    },
    "cmd" : "ADD"
  }
]
```

将文档上传到应用的某个表中的代码如下：

```
CloudsearchDoc doc = new CloudsearchDoc(indexName, client);
table_name = "要上传数据的表名";
data = [{"cmd":"add", "fields":{"id":"0","summary":"blabla..."} }];
System.out.println(doc.push(data, table_name));
```

另外还可以通过CloudsearchDoc类的提供的add、update、delete三个接口生成待上传的数据，最后在调用

push方法将数据上传；

开始搜索

OpenSearch通过设置可以实现高度个性化的搜索需求，但通用的基本的搜索功能只需通过非常简单的设置即可实现：

```
CloudsearchSearch search = new CloudsearchSearch(client);
// 添加指定搜索的应用：
search.addIndex("您的应用名称");
// 指定搜索的关键词，这里要指定在哪个索引上搜索，如果不指定的话默认在使用“default”索引（索引字段名称是您在您的数据结构中的“索引到”字段。）
search.setQueryString("词典");
search.setQueryString("default:词典");
search.setQueryString("index_name:词典");
// 指定搜索返回的格式。
search.setFormat("json");
// 设定过滤条件
search.addFilter("price>10");
// 设定排序方式 + 表示正序 - 表示降序
search.addSort("price", "+");
// 返回搜索结果
System.out.println(search.search());
```

调试

通过上面的操作我们已经可以使用基本的搜索功能了，但是优化搜索、提高搜索结果相关性是一个漫长的过程，需要我们不断试错和迭代来一点点改进。在这个过程中如果遇到问题或者发现结果与预期不一致时可以通过下面的接口获得请求的详细信息，您可以通过这些信息排查问题。特别是当遇到您问题，在旺旺群、QQ群中寻求帮助的时候，根据您提供的调试信息我们可以迅速帮您定位到问题所在。

```
//搜索类的调试信息调用接口
System.out.println(search.getDebugInfo());
//文档类的调试信息调用接口
System.out.println(doc.getDebugInfo());
//应用类的调试信息调用接口
System.out.println(index.getDebugInfo());
```

SDK客户端

CloudsearchClient类接口说明

构造函数

接口描述 CloudsearchClient类是线程安全的

参数描述

参数名称	类型	描述
accesskey	String	用户的accesskey，从网站中可以获得此信息。

secret	String	用户的 secret，从网站中可以 获得此信息。
opts	Map<String, Object>	一些可选信息，包含：
keyType	KeyTypeEnum	指定当前的用户类型，取值范围 为 ：KeyTypeEnum.OPENSEARC H，KeyTypeEnum.ALIYUN。 默认值为 KeyTypeEnum.OPENSEARCH， 使用阿里云账号的用户请设置此 参数值为 KeyTypeEnum.ALIYUN

opts: 一些可选信息，包含：

参数名称	参数描述
version	当前使用的API版本，默认值为v2。
host	指定请求的host地址，默认为 ： http://opensearch-cn-hangzhou.aliyuncs.com
timeout	指定请求超时时间，单位为：微秒。用户可以根据 自己的场景来设定此值，例如如果搜索可以设定时 间稍短，如果推送文档，可以设定稍长的时间。单 位为：微妙，默认值为10000
connect_timeout	指定连接超时时间，单位为：微秒，默认值为 5000

接口定义

```
CloudsearchClient(String accesskey, String secret, Map<String, Object> opts, KeyTypeEnum keyType)
```

指定连接池的最大连接数

参数描述

参数名称	类型	描述
maxConnections	int	连接池的最大连接数

接口定义

```
void setMaxConnections(int maxConnections)
```

搜索操作

CloudsearchSearch类接口说明

构造函数

接口描述 CloudsearchSearch类是非线程安全的。建议每次请求完毕后需要调用clear()接口清空上次请求的设置

或者重新实例化一个CloudsearchSearch对象

参数描述

参数名称	类型	描述
client	CloudsearchClient	此对象由CloudsearchClient类实例化。

接口定义

```
CloudsearchSearch(CloudsearchClient client)
```

执行搜索请求(1)

接口描述 参见:API 配置config子句

参数描述

参数名称	类型	描述
opts	Map<String, Object>	此参数如果被赋值，则会把此参数的内容分别赋给相应 的变量。此参数的值 可能有以下内容： ：

opts: 此参数如果被赋值，则会把此参数的内容分别赋给相应 的变量。此参数的值 可能有以下内容：

参数名称	参数描述
query	指定的搜索查询串，可以为query=>'鲜花'，也可以为query=>"索引名:'鲜花'"。
indexes	指定的搜索应用，可以为一个索引，也可以多个索引查询。
fetch_field	设定只返回某些字段的值。
format	指定返回的数据格式，有json,xml和protobuf三种格式可选。
formula_name	指定的精排表达式名称，此名称需在网站中设定。
first_formula_name	指定的粗排表达式名称，此名称需在网站中指定。
summary	指定summary字段一些标红、省略、截断等规则。
start	指定搜索结果集的偏移量。取值范围：[0,5000]，默认值：0
hits	指定返回结果集的数量。取值范围：[0,500]，默认

	值：20
sort	指定排序规则。默认值：“-”（降序）
filter	指定通过某些条件过滤结果集。
aggregate	指定统计类的信息。
distinct	指定distinct排序。
kvpair	指定的kvpair内容。

返回结果

- String 返回搜索结果。

异常描述

IOException

ClientProtocolException

接口定义

```
String search(Map<String, Object> opts)throws ClientProtocolException, IOException, UnknownHostException
```

执行搜索请求(2)

返回结果

- String 返回搜索结果。

异常描述

IOException

ClientProtocolException

接口定义

```
String search() throws ClientProtocolException, IOException
```

添加一个应用列表来进行搜索

参数描述

参数名称	类型	描述
indexes	List<String>	应用名称或应用名称列表。

接口定义

```
void addIndex(List<String> indexes)
```

添加一个应用来进行搜索

参数描述

参数名称	类型	描述
indexName	String	要搜索的应用名称。

接口定义

```
void addIndex(String indexName)
```

在当前搜索中去掉一个应用的搜索结果

参数描述

参数名称	类型	描述
indexName	String	

接口定义

```
void removeIndex(String indexName)
```

获取当前请求中所有的应用名列表

返回结果

- List<String> 返回当前搜索的所有应用列表。

接口定义

```
List<String> getSearchIndexes()
```

设置精排表达式名称

接口描述 此表达式名称和结构需要在网站中已经设定，详情请浏览官网中的应用指定的表达式名称。

参数描述

参数名称	类型	描述
formulaName	String	表达式名称。

接口定义

```
void setFormulaName(String formulaName)
```

获取当前设置的表达式名称

返回结果

- String 返回当前设定的表达式名称。

接口定义

```
String getFormulaName()
```

设置粗排表达式名称

接口描述 此表达式名称和结构需要在网站中已经设定,详情请浏览官网中的应用指定的表达式名称。

参数描述

参数名称	类型	描述
formulaName	String	表达式名称。

接口定义

```
void setFirstFormulaName(String formulaName)
```

获取当前设置的粗排表达式名称

返回结果

- String 返回当前设定的表达式名称。

接口定义

```
String getFirstFormulaName()
```

添加一条动态摘要信息(1)

接口描述 增加了此内容后, fieldName字段可能会被截断、飘红等。

参数描述

参数名称	类型	描述
fieldName	String	指定的生效的字段。此字段必需为可分词的text类型的字段。
len	Integer	指定结果集返回的词字段的字节长度, 一个汉字为2个字节。
element	String	指定命中的query的标红标签, 可以为em等。
ellipsis	String	指定用什么符号来标注未展示完的数据, 例如 "..."。
snippet	Integer	指定query命中几段

		summary内容。
--	--	------------

返回结果

- boolean 返回是否添加成功。

接口定义

```
boolean addSummary(String fieldName, Integer len, String element,String ellipsis, Integer snippet)
```

添加一条动态摘要信息(2)

接口描述 增加了此内容后，fieldName字段可能会被截断、飘红等。

参数描述

参数名称	类型	描述
fieldName	String	指定的生效的字段。此字段必需为可分词的text类型的字段。

返回结果

- boolean 返回是否添加成功。

接口定义

```
boolean addSummary(String fieldName)
```

添加一条动态摘要信息(3)

接口描述 增加了此内容后，fieldName字段可能会被截断、飘红等。

参数描述

参数名称	类型	描述
fieldName	String	指定的生效的字段。此字段必需为可分词的text类型的字段。
len	Integer	指定结果集返回的词字段的字节长度，一个汉字为2个字节。
ellipsis	String	指定用什么符号来标注未展示完的数据，例如“...”。
snippet	Integer	指定query命中几段summary内容。
elementPrefix	String	指定标签前缀。
elementPostfix	String	指定标签后缀。

返回结果

- boolean 返回是否添加成功。

接口定义

```
boolean addSummary(String fieldName, Integer len, String ellipsis,Integer snippet, String elementPrefix, String elementPostfix)
```

获取当前所有设置的摘要信息

返回结果

- Map<String, Map<String, Object>> 返回summary信息

接口定义

```
Map<String, Map<String, Object>> getSummary()
```

获取指定字段的摘要信息

参数描述

参数名称	类型	描述
fieldName	String	指定的字段名称。

返回结果

- Map<String, Object> 返回指定字段的summary信息。

接口定义

```
Map<String, Object> getSummary(String fieldName)
```

把摘要信息生成字符串并返回

返回结果

- String 返回字符串的summary信息。

接口定义

```
String getSummaryString()
```

设置返回的数据格式名称

参数描述

参数名称	类型	描述
format	String	数据格式名称，有xml, json和protobuf 三种类型。默认值为：“xml”

接口定义

```
void setFormat(String format)
```

获取当前的数据格式名称

返回结果

- String 返回当前的数据格式名称。

接口定义

```
String getFormat()
```

设置返回结果的偏移量

参数描述

参数名称	类型	描述
start	int	偏移量。取值范围为 [0,5000], 默认值为：0

接口定义

```
void setStartHit(int start)
```

获取返回结果的偏移量

返回结果

- int 返回当前设定的偏移量。

接口定义

```
int getStartHit()
```

设置当前返回结果集的文档个数

参数描述

参数名称	类型	描述
hits	int	指定的doc个数。默认值为：20,取值范围：[0,500]

接口定义

```
void setHits(int hits)
```

获取当前设定的结果集的文档条数

返回结果

- int 返回当前指定的doc个数。

接口定义

```
int getHits()
```

增加一个排序字段及排序方式

接口描述 详细描述请见API 排序sort子句

参数描述

参数名称	类型	描述
field	String	需要排序的字段名称。
sortChar	String	排序方式，有升序 “+” 和降序 “-” 两种方式。默认值为 “-”

接口定义

```
void addSort(String field, String sortChar)
```

增加一个排序字段

接口描述 详细描述请见API 排序sort子句

参数描述

参数名称	类型	描述
field	String	指定排序的字段名称。

接口定义

```
void addSort(String field)
```

删除指定字段的排序

参数描述

参数名称	类型	描述
field	String	指定的字段名称。

接口定义

```
void removeSort(String field)
```

获取排序信息

返回结果

- 返回当前所有的排序字段及升降序方式。

接口定义

```
Map<String, String> getSort()
```

获取字符串类型的排序信息

接口描述 把排序信息生成字符串并返回。

返回结果

- String 返回字符串类型的排序规则。

接口定义

```
String getSortString()
```

增加一个自定义参数

参数描述

参数名称	类型	描述
paramKey	String	参数名称。
paramValue	String	参数值。

接口定义

```
void addCustomParam(String paramKey, String paramValue)
```

获取自定义参数

返回结果

- 返回自定义参数

接口定义

```
Map<String, String> getCustomParam()
```

增加过滤规则(1)

接口描述 详细请见 API 过滤filter子句

参数描述

参数名称	类型	描述
filter	String	过滤规则，例如fieldName >= 1。

operator	String	操作符，可以为 AND OR。默认为 "AND"
----------	--------	--------------------------

接口定义

```
void addFilter(String filter, String operator)
```

增加过滤规则(2)

接口描述 详细请见 API 过滤filter子句

参数描述

参数名称	类型	描述
filter	String	过滤规则。

接口定义

```
void addFilter(String filter)
```

获取过滤规则

返回结果

- String 返回字符串类型的过滤规则。

接口定义

```
String getFilter()
```

添加统计信息相关参数(1)

接口描述 一个关键词通常能命中数以万计的文档，用户不太可能浏览所有文档来获取信息。而用户感兴趣的可能是些统计类的信息，比如，查询“手机”这个关键词，想知道每个卖家所有商品中的最高价格。

则可以按照卖家的user_id分组，统计每个小组中最大的price值，例如：

```
groupKey:user_id,aggFun:max(price)
```

详细说明请参见：API aggregate子句说明

参数描述

参数名称	类型	描述
groupKey	String	指定需要统计的字段名称。
aggFun	String	指定统计的方法。当前支持：count、max、min、sum等。
range	String	指定统计范围。

maxGroup	String	最大组个数。
aggFilter	String	指定过滤某些统计。
aggSamplerThresHold	String	指定抽样的伐值。
aggSamplerStep	String	指定抽样的步长。

返回结果

- boolean 返回添加成功或失败。

接口定义

```
boolean addAggregate(String groupKey, String aggFun, String range,String maxGroup, String aggFilter, String aggSamplerThresHold,String aggSamplerStep)
```

添加统计信息相关参数(2)

参数描述

参数名称	类型	描述
groupKey	String	指定需要统计的字段名称。
aggFun	String	指定统计的方法。当前支持：count、max、min、sum等。

返回结果

- boolean 返回添加成功或失败。

接口定义

```
boolean addAggregate(String groupKey, String aggFun)
```

获取用户设定的统计相关信息

返回结果

- List<Map<String, Object>> 返回用户设定的统计信息。

接口定义

```
List<Map<String, Object>> getAggregate()
```

返回字符串类型的统计信息

返回结果

- String 返回字符串类型的统计信息。

接口定义

```
String getAggregateString()
```

添加一条排序信息(1)

接口描述 例如：检索关键词“手机”共获得10个结果，分别为：doc1，doc2，doc3，doc4，doc5，doc6，doc7，doc8，doc9，doc10。其中前三个属于用户A，doc4-doc6属于用户B，剩余四个属于用户C。如果前端每页仅展示5个商品，则用户C将没有展示的机会。但是如果按照user_id进行抽取，每轮抽取1个，抽取2次，并保留抽取剩余的结果，则可以获得以下文档排列顺序：doc1、doc4、doc7、doc2、doc5、doc8、doc3、doc6、doc9、doc10。可以看出，通过distinct排序，各个用户的商品都得到了展示机会，结果排序更趋于合理。

详细说明请见：API distinct子句

参数描述

参数名称	类型	描述
key	String	为用户用于做distinct抽取的字段，该字段要求为可过滤字段。
distCount	int	为一次抽取的document数量，默认值为1。
distTimes	int	为抽取的次数，默认值为1。
reserved	String	为是否保留抽取之后剩余的结果，true为保留，false则丢弃，丢弃时totalHits的个数会减去被distinct而丢弃的个数，但这个结果不一定准确，默认为true。
distFilter	String	为过滤条件，被过滤的doc不参与distinct，只在后面的排序中，这些被过滤的doc将和被distinct出来的第一组doc一起参与排序。默认是全部参与distinct。
updateTotalHit	String	当reserved为false时，设置update_total_hit为true，则最终total_hit会减去被distinct丢弃的数目（不一定准确），为false则不减；默认为false。
grade	double	指定档位划分阈值。

返回结果

- 返回是否添加成功。

接口定义

```
boolean addDistinct(String key, int distCount, int distTimes,String reserved, String distFilter, String updateTotalHit, double grade)
```

添加一条排序信息(2)

参数描述

参数名称	类型	描述
key	String	为用户用于做distinct抽取的字段，该字段要求为可过滤字段。

返回结果

- 返回是否添加成功。

接口定义

```
boolean addDistinct(String key)
```

添加一条排序信息(3)

参数描述

参数名称	类型	描述
key	String	为用户用于做distinct抽取的字段，该字段要求为可过滤字段。
distCount	int	为一次抽取的document数量，默认值为1。

返回结果

- 返回是否添加成功。

接口定义

```
boolean addDistinct(String key, int distCount)
```

添加一条排序信息(4)

参数描述

参数名称	类型	描述
key	String	为用户用于做distinct抽取的字段，该字段要求为可过滤字段。
distCount	int	为一次抽取的document数量，默认值为1。
distTimes	int	为抽取的次数，默认值为1。

返回结果

- 返回是否添加成功。

接口定义

```
boolean addDistinct(String key, int distCount, int distTimes)
```

添加一条排序信息(5)

参数描述

参数名称	类型	描述
key	String	为用户用于做distinct抽取的字段，该字段要求为可过滤字段。
distCount	int	为一次抽取的document数量，默认值为1。
distTimes	int	为抽取的次数，默认值为1。
reserved	String	为是否保留抽取之后剩余的结果，true为保留，false则丢弃，丢弃时totalHits的个数会减去被distinct而丢弃的个数，但这个结果不一定准确，默认为true。

返回结果

- 返回是否添加成功。

接口定义

```
boolean addDistinct(String key, int distCount, int distTimes,String reserved)
```

添加一条排序信息(6)

参数描述

参数名称	类型	描述
key	String	为用户用于做distinct抽取的字段，该字段要求为可过滤字段。
distCount	int	为一次抽取的document数量，默认值为1。
distTimes	int	为抽取的次数，默认值为1。
reserved	String	为是否保留抽取之后剩余的结果，true为保留，false则丢弃，丢弃时totalHits的个数会减去被distinct而丢弃的个数，但这个结果不一定准确，默认为true。

distFilter	String	为过滤条件，被过滤的doc不参与distinct，只在后面的排序中，这些被过滤的doc将和被distinct出来的第一组doc一起参与排序。默认是全部参与distinct。
------------	--------	--

返回结果

- 返回是否添加成功。

接口定义

```
boolean addDistinct(String key, int distCount, int distTimes,String reserved, String distFilter)
```

添加一条排序信息(7)

参数描述

参数名称	类型	描述
key	String	为用户用于做distinct抽取的字段，该字段要求为可过滤字段。
distCount	int	为一次抽取的document数量，默认值为1。
distTimes	int	为抽取的次数，默认值为1。
reserved	String	为是否保留抽取之后剩余的结果，true为保留，false则丢弃，丢弃时totalHits的个数会减去被distinct而丢弃的个数，但这个结果不一定准确，默认为true。
distFilter	String	为过滤条件，被过滤的doc不参与distinct，只在后面的排序中，这些被过滤的doc将和被distinct出来的第一组doc一起参与排序。默认是全部参与distinct。
updateTotalHit	String	当reserved为false时，设置update_total_hit为true，则最终total_hit会减去被distinct丢弃的数目（不一定准确），为false则不减；默认为false。

返回结果

- 返回是否添加成功。

接口定义

```
boolean addDistinct(String key, int distCount, int distTimes,String reserved, String distFilter, String updateTotalHit)
```

删除某个字段的所有distinct排序信息

参数描述

参数名称	类型	描述
distinctKey	String	要删除的dist key字段名称。

接口定义

```
void removeDistinct(String distinctKey)
```

获取某所有的distinct排序信息

返回结果

- 返回所有的distinct信息。

接口定义

```
Map<String, Map<String, Object>> getDistinct()
```

获取string类型的所有的distinct信息

返回结果

- 返回字符串类型的distinct信息。

接口定义

```
String getDistinctString()
```

设定指定索引字段范围的搜索关键词

接口描述 此query是查询必需的一部分，可以指定不同的索引名，并同时可指定多个查询及之间的关系（ AND, OR, ANDNOT, RANK ）。

例如查询subject索引字段的query: “手机” ，可以设置为 query=subject:'手机'。

上边例子如果查询price 在1000-2000之间的手机，其查询语句为： query=subject:'手机'

AND price:[1000,2000]

NOTE: text类型索引在建立时做了分词，而string类型的索引则没有分词。

详细说明请见：API query子句

参数描述

参数名称	类型	描述
query	String	设定搜索的查询语法。

接口定义

```
void setQueryString(String query)
```

获取当前指定的查询词内容

返回结果

- 返回当前设定的查询query子句内容。

接口定义

```
String getQuery()
```

设定当前的kvpair

接口描述 详细请见API 自定义kvpair子句

参数描述

参数名称	类型	描述
pair	String	kvpair内容

接口定义

```
void setPair(String pair)
```

获取当前的kvpair

返回结果

- String 返回当前设定的kvpair。

接口定义

```
String getPair()
```

设定rerank_size

参数描述

参数名称	类型	描述
rerank_size	int	精排算分文档个数,默认值200

接口定义

```
void setRerankSize(int rerank_size)
```

获取当前rerank_size

接口描述 rerankSize表示参与精排算分的文档个数，一般不用修改，使用默认值就能满足，不用设置,会自动使用默认值200

返回结果

- int 当前设定的rerank_size

接口定义

```
int getRerankSize()
```

设置搜索返回的索引字段列表

参数描述

参数名称	类型	描述
fields	List<String>	结果集返回的字段。

接口定义

```
void addFetchFields(List<String> fields)
```

添加某个字段到搜索结果字段列表中

参数描述

参数名称	类型	描述
field	String	指定的字段名称。

接口定义

```
void addFetchField(String field)
```

获取搜索结果包含的字段列表

返回结果

- 返回指定返回字段的列表。

接口定义

```
List<String> getFetchFields()
```

清空用户设置的所有搜索条件

接口定义

```
void clear()
```

添加一条自定义配置

参数描述

参数名称	类型	描述
key	String	自定义配置项key
value	Object	自定义配置项value

接口定义

```
void addCustomConfig(String key, Object value)
```

移除自定义配置

参数描述

参数名称	类型	描述
key	String	指定配置项的key

接口定义

```
void removeCustomConfig(String key)
```

获取上次搜索请求的信息

返回结果

- String 上次搜索请求的信息

接口定义

```
String getDebugInfo()
```

搜索示例

```
import com.opensearch.javasdk.*;

String accesskey= "您的阿里云的Access Key ID";
String secret = "阿里云 Access Key ID 对应的 Access Key Secret";

Map<String, Object> opts = new HashMap<String, Object>();

// 这里的host需要根据访问应用基本详情中的API入口来确定
opts.put("host", "http://opensearch-cn-hangzhou.aliyuncs.com");
CloudsearchClient client = new CloudsearchClient(accesskey, secret, opts,
    KeyTypeEnum.ALIYUN);
CloudsearchSearch search = new CloudsearchSearch(client);

// 添加指定搜索的应用：
search.addIndex("my_demo");

// 指定搜索的关键词，如果没有输入索引名称，则使用 "default"
search.setQueryString("词典");
```

```
//和下边等同
search.setQueryString("default:'词典'");

// 或者指定某索引字段进行查找。
// 索引字段名称是您在您的数据结构中的“索引到”字段。
search.setQueryString("index_name:'词典'");

// 指定搜索返回的格式。
search.setFormat("json");

// 设定过滤条件。字段必须设定为可过滤
search.addFilter("price>10");

// 设定排序方式。字段必须设定为可过滤
search.addSort("price", "+");

// 返回搜索结果。
System.out.println(search.search());
```

文档操作

CloudsearchDoc类接口说明

构造函数

接口描述 CloudsearchDoc类是非线程安全的，请每次调用时单独实例化一个实例使用。

参数描述

参数名称	类型	描述
indexName	String	指定操作的索引名称。
client	CloudsearchClient	CloudsearchClient实例。

接口定义

```
CloudsearchDoc(String indexName, CloudsearchClient client)
```

查看文档详情

接口描述 根据doc id获取doc的详细信息。

参数描述

参数名称	类型	描述
docId	String	指定的doc id。

返回结果

- 返回API返回的结果。

异常描述

IOException

ClientProtocolException

接口定义

```
String detail(String docId) throws ClientProtocolException, IOException
```

添加文档

接口描述 设置需要添加的属性名称和属性值，用于生成符合文档格式的数据，所有更新结束之后需要调用push(String tableName)方法

参数描述

参数名称	类型	描述
fields	Map<String, Object>	字段名和字段值的map

异常描述

- JSONException

接口定义

```
void add(Map<String, Object> fields) throws JSONException
```

更新文档

接口描述 设置需要更新的属性名称和属性值，用于生成符合文档格式的数据，所有更新结束之后需要调用push(String tableName)方法

参数描述

参数名称	类型	描述
fields	Map<String, Object>	字段名和字段值的map

异常描述

- JSONException

接口定义

```
void update(Map<String, Object> fields) throws JSONException
```

删除文档

接口描述 设置需要删除的属性名称和属性值，用于生成符合文档格式的数据，所有更新结束之后需要调用push(String tableName)方法

参数描述

参数名称	类型	描述
fields	Map<String, Object>	字段名和字段值的map

异常描述

- JSONException

接口定义

```
void remove(Map<String, Object> fields) throws JSONException
```

执行文档变更操作(1)

接口描述 针对文档的操作add、update和remove会生成符合文档格式的数据，通过调用此接口用户提交的文档变更才会真正生效。

参数描述

参数名称	类型	描述
tableName	String	表名称

返回结果

- 返回的数据

异常描述

IOException

ClientProtocolException

接口定义

```
String push(String tableName) throws ClientProtocolException, IOException
```

执行文档变更操作(2)

接口描述 通过此接口可以直接将符合文档格式的数据直接推送到指定的表中

参数描述

参数名称	类型	描述
docs	String	此docs为用户push的数据，此

		字段为json类型的字符串。
tableName	String	操作的表名。

返回结果

- 请求API并返回相应的结果。

异常描述

IOException

ClientProtocolException

接口定义

String **push**(String docs, String tableName) **throws** ClientProtocolException, IOException

通过文件导入数据(1)

接口描述 导入HA3 doc数据到指定的应用的指定表中

文件编码：UTF-8

支持CMD: add, delete, update。如果给出的字段不是全部，add会在未给出的字段加默认值，覆盖原值；update只会更新给出的字段，未给出的不变。

文件分隔符：

```
<pre>
```

编码-----描述-----显示形态-----

"\x1E\n" 每个doc的分隔符. ^^ (接换行符)

"\x1F\n" 每个字段key和value分隔. ^_ (接换行符)

"\x1D" 多值字段的分隔符. ^]

```
</pre>
```

示例：

```
<pre>
```

CMD=add^_

url=<http://www.opensearch.console.aliyun.com>^_

title=开放搜索^_

body=xxxxxxxxx^

```
multivalue_feild=123^J1234^J12345^
```

```
^^
```

```
CMD=update^_
```

```
...
```

```
</pre>
```

NOTE: 文件结尾的分隔符也必需为"^^\n", 最后一个换行符不能省略。

参数描述

参数名称	类型	描述
filePath	String	指定的文件路径。
tableName	String	指定push数据的表名。

返回结果

- 返回成功或者错误信息。

异常描述

- JSONException

接口定义

```
String pushHADocFile(String filePath, String tableName) throws JSONException
```

通过文件导入数据(2)

接口描述 导入HA3 doc数据到指定的应用的指定表中

参数描述

参数名称	类型	描述
filePath	String	指定的文件路径。
tableName	String	指定push数据的表名。
offset	long	文档数据的偏移量，小于设定的offset行号的文档将被跳过

返回结果

- 返回成功或者错误信息。

异常描述

- JSONException

接口定义

```
String pushHADocFile(String filePath, String tableName, long offset) throws JSONException
```

获取上次请求的信息

返回结果

- String

接口定义

```
String getDebugInfo()
```

文档操作使用示例

应用结构

首先我们回顾一下我们之前创建的应用的结构：

	名称	含义
1	id	标示一个主键id，在应用中唯一的字段，primary_key
2	type_id	类型的id
3	cat_id	分类id，类似于,1标示体育，2、科技，3、军事等等
4	title	资讯的标题
5	body	资讯的内容
6	url	资讯的url连接
7	author	作者
8	thumbnail	缩略图
9	source	来源
10	create_timestamp	资讯的创建时间
11	update_timestamp	资讯的最后修改时间
12	hit_num	点击次数
13	focus_count	关注次数
14	grade	等级
15	comment_count	评论数
16	tag	标签

文档概念

OpenSearch的文档是一个json类型的字符串，结构如下：

```
[
  {
    fields:{...},
    cmd:"..."
  }
  ...
]
```

一条文档是由fields字段和cmd字段构成的一个结构体，其中fields字段内包含文档的核心数据，cmd表示针对此条文档所做的操作，对文档的操作包括添加(add),更新(update)和删除（delete）三种。我们创建的news应用的一条文档示例如下：

```
[
  {
    "fields": {
      "id": "0",
      "type_id": 34,
      "cat_id": 10,
      "title": "阿里云开放搜索助力企业搜索应用",
      "body": "广大中小企业都有各种结构化的数据需要进行检索，目前一般采用数据库本身提供的搜索功能或者利用open source的搜索软件搭建，这样的做法不但会消耗网站本身的资源，性能也会很容易成为问题，而且相关性通常也不好。我们的产品的目的是要利用阿里云先进的云计算和搜索技术向广大中小企业提供低成本，高质量，高性能，可定制的数据搜索解决方案。本项目和云搜索的通用解决方案目标略有不同，主要区别为本项目主要针对用户的结构化数据进行搜索，云搜索的通用解决方案则主要是针对网页型数据为处理对象。",
      "url": "http://opensearch.console.aliyun.com",
      "author": "阿里云",
      "thumbnail": "http://opensearch.console.aliyun.com",
      "source": "新浪科技",
      "create_timestamp": 1426589415,
      "update_timestamp": 1426589415,
      "hit_num": 700,
      "focus_count": 31,
      "grade": 96,
      "comment_count": 68,
      "tag": "搜索"
    },
    "cmd": "ADD"
  },
]
```

文档操作

在建立好应用之后，我们需要把自己应用的数据上传到应用中，并且在运营的过程中对文档的内容进行维护。这里涉及到的文档相关的操作包括以下四种：添加文档、查看文档、更新文档和删除文档。

添加文档 将文档添加到OpenSearch应用的表中对应的操作cmd是ADD，在Java SDK中可以使用 add接口完成文档添加操作：例如上面的fields字段的内容是从数据库中读取的，存放在一个Map<String,Object>类型的数据结构里，可以直接使用add接口生成符合文档格式的json字符串,再调用push接口完成上传；

```
client.add(fields);
client.push("main");//main是上传到的表名
```

如果已经生成了符合文档格式的json_string 可以直接调用push接口实现上传：

```
client.push(json_string, "main");
```

查看文档 将文档上传至应用后可以通过文档id使用detail接口查看文档内容：

```
doc.detail("1");
```

更新文档 更新文档时 文档中cmd字段对应的值为update，并且在fields字段中要提供待更新的文档的id，然后调用push接口使其生效：

```
doc.update(fields);
doc.push("main");//main为待更新的文档所在的表名
```

删除文档 删除文档时文档中的cmd字段对应的值为delete，需要提供待删除的文档的id，然后调用push接口

```
doc.remove("1");
doc.push("main");//main为待删除文档所在的表名
```

使用文件来上传文档 除了上面的方法还可以通过文件量，小于设定的offset行号的doc将被跳过 导入HA3 doc数据到指定的应用的指定表中

文件编码：UTF-8

支持CMD: add, delete, update。如果给出的字段不是全部，add会在未给出的字段加默认值，覆盖原值；update只会更新给出的字段，未给出的不变。

文件分隔符：

```
<pre>
```

编码-----描述-----显示形态-----

"\x1E\n" 每个doc的分隔符. ^^ (接换行符)

"\x1F\n" 每个字段key和value分隔. ^_ (接换行符)

"\x1D" 多值字段的分隔符. ^]

```
</pre>
```

示例：

```
<pre>
```

CMD=add^_

url=<http://www.opensearch.console.aliyun.com>^_

title=开放搜索^_

body=xxxxxxxx^

multivalue_feild=123^J1234^J12345^

^^

CMD=update^_

...

</pre>

注意：文件结尾的分隔符也必需为"^^\n"，最后一个换行符不能省略。

```
doc.pushHADocFile(file_path,table_name);
```

使用HA类型文件推送文档时还可以指定从文件中的某个位置开始推送，对应的接口为：

```
doc.pushHADocFile(file_path,talbe_name,offset);
```

其中：offset 文档数据的偏移量，小于设定的offset行号的doc将被跳过

调试接口 CloudsearchDoc类内置了一个调试接口，通过调用调试接口可以获得操作发出的请求内容，当我们操作文档时发现与预期不符时可以通过此接口查看请求细节。

```
debugInfo = doc.getDebugInfo();
```

应用操作

CloudsearchIndex类接口说明

构造函数

接口描述 CloudsearchIndex类是非线程安全的，请每次使用时单独实例化一个实例使用。

参数描述

参数名称	类型	描述
indexName	String	索引的名称
client	CloudsearchClient	提交请求的client

接口定义

```
CloudsearchIndex(String indexName, CloudsearchClient client)
```

根据模板名称创建应用(1)

参数描述

参数名称	类型	描述
templateName	String	模板名称
opts	Map<String, String>	包含应用的备注信息,具体包括： index_des-应用的描述信息

		,package_id-该应用选择的套餐
--	--	----------------------

返回结果

- 返回api返回的正确或错误的结果

异常描述

IOException

ClientProtocolException

接口定义

String **createByTemplateName**(String templateName,Map<String, String> opts) throws ClientProtocolException,IOException

根据模板名称创建应用(2)

参数描述

参数名称	类型	描述
templateName	String	模板名称

返回结果

- 返回api返回的正确或错误的结果

异常描述

IOException

ClientProtocolException

接口定义

String **createByTemplateName**(String templateName) throws ClientProtocolException, IOException

更新应用名称和备注信息

参数描述

参数名称	类型	描述
toIndexName	String	应用名称
opts	Map<String, String>	相关参数

返回结果

- 返回api返回的正确或错误的结果

异常描述

JSONException

IOException

ClientProtocolException

接口定义

```
String rename(String toIndexName, Map<String, String> opts)throws JSONException, ClientProtocolException,IOException
```

删除当前的应用

返回结果

- 返回API返回的正确或错误的结果

异常描述

IOException

ClientProtocolException

接口定义

```
String delete() throws ClientProtocolException, IOException
```

查看当前应用的状态

返回结果

- 返回API返回的正确或错误的结果

异常描述

IOException

ClientProtocolException

接口定义

```
String status() throws ClientProtocolException, IOException
```

列出所有应用

参数描述

参数名称	类型	描述
page	Integer	开始的页码
pageSize	Integer	获取的记录数

返回结果

- 返回API返回的正确或错误的结果

异常描述

IOException

ClientProtocolException

接口定义

```
String listIndexes(Integer page, Integer pageSize) throws ClientProtocolException, IOException
```

获取当前应用的名称

返回结果

- 应用名称

接口定义

```
String getIndexName()
```

获取错误信息

参数描述

参数名称	类型	描述
page	int	开始页数
pageSize	int	每页的记录数

返回结果

- API返回的错误信息

异常描述

IOException

ClientProtocolException

接口定义

```
String getErrorMessage(int page, int pageSize) throws ClientProtocolException, IOException
```

获取上次请求的信息

返回结果

- String

接口定义

```
String getDebugInfo()
```

应用操作示例

通过OpenSearch控制台也可以完成应用创建、查看、修改等操作，在SDK中也提供了部分操作应用的接口。下面带大家逐一看看各个接口的使用方式。首先我们实例化一个应用类,很简单

```
CloudsearchIndex app = new CloudsearchIndex("sdk_user_demo",client);
```

其中client的实例化可以在CloudsearchClient类的示例中查看,这里不再重复。

使用模板创建应用

OpenSearch内置了多个应用模板(当前包括：资讯类、小说类、应用类、社区类、文档类和电商类)，如果希望使用内置模板，则可以通过下面方式创建一个应用,这里我们使用系统内置的资讯类应用模板(builtin_news)来创建一个应用

```
String ret = app.createByTemplateName("builtin_news");
```

您可以在控制台创建自定义结构的应用，然后保存为应用模板，然后也可以这种方式来创建应用。操作成功时返回

```
{"result":{"index_name":"您要创建的应用名称"},"status":"OK","RequestId":""}
```

创建失败时(可能有多种原因)返回的示例如下：

```
{"status":"FAIL","errors":[{"code":2002,"message":"app already exists"},"RequestId":""}
```

修改应用名称

很简单：

```
String ret = app.rename("要修改成的应用名称");
System.out.println(ret);
```

这个接口还支持修改应用的描述信息，参见接口的参数说明。

删除应用

```
String ret = app.delete();
System.out.println(ret);
```

获取应用名称

```
System.out.println(app.getIndexName());
```

查看应用详情

```
System.out.println(app.status());
```

获取应用列表

获取您的所有的应用列表信息，该接口可以指定页码或者每页的条数。

```
// 获取所有的应用信息，取第一页的数据，每页取10条。
System.out.println(app.listIndexes(1, 10));
```

获取错误信息

错误信息一般情况下在您请求每个接口时实时返回，但是push 数据时如果数据已经到了后端发现错误，则此时这些接口已经拿不到这个数据的错误信息了。这个时候您可以：

- 1、通过网站查看某个应用的push错误信息。
- 2、通过sdk中getErrorMessage() 来获取错误信息的方式来获取某个应用在push的过程中出现了哪些错误信息。

```
// 列出当前应用的所有错误信息。
System.out.println(app.getErrorMessage());
// 或者分页获取，例如列出第一页、每页10条：
System.out.println(app.getErrorMessage(1, 10));
```

获取调试信息

CloudsearchIndex类内置了一个调试接口，通过调用调试接口可以获得操作发出的请求内容，当我们操作文档时发现与预期不符时可以通过此接口查看请求细节。

```
debugInfo = app.getDebugInfo();
```

相关下载

相关下载

最新稳定版

v2.1.2 (2014-11)

- java-sdk-v2.1.2.jar
- [maven地址](#)

```
<dependency>
<groupId>com.aliyun.opensearch</groupId>
<artifactId>aliyun-sdk-opensearch</artifactId>
<version>2.1.2</version>
</dependency>
```

PhpSDK手册

版本说明

前言

简介

提供PHP语言对OpenSearch API接口的封装，实现开放搜索服务相关的应用管理、数据处理、搜索等功能。

版本说明

当前版本为 2.0.4,相对于上一版本(2.0.3),本次升级涉及的功能变更如下：

修改bug

client中取消设置默认时区，使用gmdate代替原有date 获取gmt时间

调试接口

client提供类中getRequest()方法获取上次操作的请求信息。

client参数

添加对 API请求超时时间和client连接超时时间的设置。

User-Agent

在sdk发出的请求的useragent中添加 sdk的类型和版本信息

Search 属性

添加对config子句的rerank_size 属性的支持

历史版本

2.0.3

在2.0.3版本SDK中使用直接在控制台输出调试信息的方式进行调试，2.0.4版本中改用getRequest()接口获取。其他接口对上一版本完全兼容。

使用教程

快速开始

使用openSearch提供搜索功能十分简单，下面给大家迅速上手

准备工作

获取AccessKey和Secret

登录<http://opensearch.console.aliyun.com>，点击 ACCESSKEY管理 查看是否已经创建了阿里云 access key id和access key secret。如果没有创建，则点击创建一套id和secret

下载php SDK并添加到项目中

在左侧栏相关下载中下载php SDK包（Github镜像即将推出）并将SDK添加到项目中

```
<?php
require_once("php_v2.0.4/CloudsearchClient.php");
require_once("php_v2.0.4/CloudsearchIndex.php");
require_once("php_v2.0.4/CloudsearchDoc.php");
require_once("php_v2.0.4/CloudsearchSearch.php");
```

创建应用

我们先实例化一个SDK客户端client，在下面的操作中我们都会使用到它。

```
$access_key = "替换成自己的accesskey";
$secret = "替换成自己的Secret";
//杭州公网API地址：http://opensearch-cn-hangzhou.aliyuncs.com
//北京公网API地址：http://opensearch-cn-beijing.aliyuncs.com
$host = "http://opensearch-cn-hangzhou.aliyuncs.com";//根据自己的应用区域选择API
$key_type = "aliyun"; //固定值，不必修改
$opts = array('host'=>$host);
$client = new CloudsearchClient($access_key,$secret,$opts,$key_type);
```

下面我们通过系统内置的小说类型的模板创建一个应用

```
// 指定即将创建的应用名称
$app_name = "sdk_user_demo";
$index = new CloudsearchIndex($app_name,$client);
$result = $index->createByTemplateName("builtin_novel");
echo $result;
```

通过上面的代码，我们使用系统内置的小说类型的模板完成了一个应用的创建。

上传文档

在上面代码的基础上我们继续使用CloudsearchDoc类向刚创建的应用中上传一些文档，应用名称继续使用上面的\$app_name。OpenSearch应用中的文档是一个json类型的字符串，结构如下：

```
[
  {
    "fields": {},
    "cmd": ""
  }
  ...
]
```

cmd字段可选的值为：ADD、DELETE、UPDATE，分别表示添加、删除以及更新一条文档；fields字段内包含文档本身的字段属性，比如在一个小说应用的结构中可能包含以下字段：title:小说的名字,body：小说主体内容，url:访问小说的地址等。

```
$doc_obj = new CloudsearchDoc($app_name,$client);
$docs_to_upload = array();

for ($i = 0; $i < 10; $i++){
    $item = array();
    //指定文档操作类型为：添加
    $item['cmd'] = 'ADD';
    //添加文档内容
    $item["fields"] = array("id" => $i + 1,
        "title" => "我是一条新文档的标题",
        "body" => "我是一条新文档的body",
        "url" => "http://opensearch-cn-hangzhou.aliyuncs.com",
        "create_timestamp" => time());
    $docs_to_upload[] = $item;
}
//生成json格式字符串
$json = json_encode($docs_to_upload);
// 将文档推送到main表中
echo $doc_obj->add($json,"main");
```

以上我们向sdk_user_demo应用中上传了10条文档

开始搜索

下面我们继续使用CloudsearchSearch类搜索一下刚才上传的数据

```
// 实例化一个搜索类
$search_obj = new CloudsearchSearch($client);
// 指定一个应用用于搜索
$search_obj->addIndex($app_name);
// 指定搜索关键词
$search_obj->setQueryString("default:标题");
// 指定返回的搜索结果的格式为json
$search_obj->setFormat("json");
// 执行搜索，获取搜索结果
```

```
$json = $search_obj->search();
// 将json类型字符串解码
$result = json_decode($json,true);
print_r($result);
?>
```

这样我们就完成了一个简单的搜索功能了！

完整示例代码

大家可以直接复制下面代码，将其中的access key和secret替换成自己的，下载SDK代码，并解压到下面代码文件的同级目录中就可以体验一下OpenSearch了。

```
<?php
require_once("php_v2.0.4/CloudsearchClient.php");
require_once("php_v2.0.4/CloudsearchIndex.php");
require_once("php_v2.0.4/CloudsearchDoc.php");
require_once("php_v2.0.4/CloudsearchSearch.php");

$access_key = "替换成自己的accesskey";
$secret = "替换成自己的Secret";

//杭州公网API地址 : http://opensearch-cn-hangzhou.aliyuncs.com
//北京公网API地址 : http://opensearch-cn-beijing.aliyuncs.com (2015年4月初开放)
$host = "http://opensearch-cn-hangzhou.aliyuncs.com";
$key_type = "aliyun"; //固定值，不必修改
$opts = array('host'=>$host);
// 实例化一个client 使用自己的accesskey和Secret替换相关变量
$client = new CloudsearchClient($access_key,$secret,$opts,$key_type);

$app_name = "sdk_user_demo";
// 实例化一个应用类index_obj
$index_obj = new CloudsearchIndex($app_name,$client);
$result = $index_obj->createTemplateName("builtin_novel");

// 实例化一个文档类 doc_obj 指定应用名称
$doc_obj = new CloudsearchDoc($app_name,$client);
$docs_to_upload = array();

for ($i = 0; $i < 10; $i++){
    $item = array();
    $item['cmd'] = 'ADD';
    $item['fields'] = array("id" => $i + 1,
        "title" => "我是一条新文档的标题",
        "body" => "我是一条新文档的body",
        "url" => "http://opensearch.aliyuncs.com",
        "create_timestamp" => time());
    $docs_to_upload[] = $item;
}

$json = json_encode($docs_to_upload);
// echo $doc_obj->add($json,"main");

// 实例化一个搜索类 search_obj
```

```
$search_obj = new CloudsearchSearch($client);
// 指定一个应用用于搜索
$search_obj->addIndex($app_name);
// 指定搜索关键词
$search_obj->setQueryString("default:标题");
// 指定返回的搜索结果的格式为json
$search_obj->setFormat("json");
// 执行搜索，获取搜索结果
$json = $search_obj->search();
// 将json类型字符串解码
$result = json_decode($json,true);
print_r($result);
```

调试

通过上面的操作我们已经可以使用基本的搜索功能了，但是优化搜索、提高搜索结果相关性是一个漫长的过程，需要我们不断试错和迭代来一点点改进。在这个过程中如果遇到问题或者发现结果与预期不一致时可以通过下面的接口获得请求的详细信息，可以通过这些信息排查问题。特别是当您遇到问题，在旺旺群、QQ群中寻求帮助的时候，根据调试信息我们可以迅速帮您定位到问题所在。

```
$client->getRequest();
```

SDK客户端

CloudsearchClient

构造函数

与服务器交互的客户端，支持单例方式调用

请求参数

参数名称	类型	描述
key	string	用户的key，从阿里云网站中获取的Access Key ID。
secret	string	用户的secret，对应的Access Key Secret。
opts	array	包含下面一些可选信息
key_type	string	key和secret类型，在这里必须设定为'aliyun'，表示这个是aliyun颁发的，默认值opensearch是为了兼容老用户。默认值为:opensearch

opts: 包含下面一些可选信息

参数名称	参数类型	参数描述
version	string	使用的API版本。默认值为:v2

host	string	指定请求的host地址。默认值为: http://opensearch-cn-hangzhou.aliyuncs.com
gzip	string	指定返回的结果用gzip压缩。默认值为:false
debug	string	打印debug信息。默认值为:false
signatureMethod	string	签名方式，目前支持HMAC-SHA1。默认值为:HMAC-SHA1
signatureVersion	string	签名算法版本。默认值为:1.0

函数定义

```
function __construct($key, $secret, $opts = array(), $key_type = 'opensearch')
```

使用示例

```
$access_key = "替换成自己的accesskey";
$secret = "替换成自己的Secret";
//杭州公网API地址 : http://opensearch-cn-hangzhou.aliyuncs.com
//北京公网API地址 : http://opensearch-cn-beijing.aliyuncs.com (2015年4月初开放)
$host = "http://opensearch-cn-hangzhou.aliyuncs.com"; //根据自己的应用区域选择API
$key_type = "aliyun"; //固定值，不必修改
$client = new CloudsearchClient($access_key,$secret,$host,$key_type);
```

调试接口

获取SDK调用的调试信息,需要指定debug=true才能使用

返回结果

类型	描述
array null	调试开关(debug)打开时返回调试信息。

函数定义

```
function getRequest()
```

使用示例

```
$client->getRequest();
```

搜索操作

CloudsearchSearch

构造函数

请求参数

参数名称	类型	描述
client	CloudsearchClient	此对象由CloudsearchClient类实例化。

函数定义

```
function __construct($client)
```

使用示例

```
$search_obj = new CloudsearchSearch($client);
```

执行搜索

执行向API提出搜索请求。 更多说明请参见API接口-搜索相关

请求参数

参数名称	类型	描述
opts	array	此参数如果被赋值，则会把此参数的内容分别赋给相应的变量。此参数的值可能有以下内容：

opts: 此参数如果被复制，则会把此参数的内容分别赋给相应的变量。此参数的值可能有以下内容：

参数名称	参数类型	参数描述
query	string	指定的搜索查询串，可以为 query=>"索引名:鲜花"。
indexes	array	指定的搜索应用，可以为一个应用，也可以多个应用查询。
fetch_field	array	设定返回的字段列表，如果只返回url和title，则为 array('url', 'title')。
format	string	指定返回的数据格式，有 json,xml和protobuf三种格式可选。默认值为：'xml'
formula_name	string	指定的表达式名称，此名称需在网站中设定。
summary	array	指定summary字段一些标红、省略、截断等规则。
start	int	指定搜索结果集的偏移量。默认为0。
hits	int	指定返回结果集的数量。默认为20。

sort	array	指定排序规则。默认值为：'self::SORT_DECREASE' (降序)
filter	string	指定通过某些条件过滤结果集。
aggregate	array	指定统计类的信息。
distinct	array	指定distinct排序。
kvpair	string	指定的kvpair。

返回结果

类型	描述
array	返回搜索结果

函数定义

```
function search($opts = array())
```

使用示例

```
$result = $search_obj->search();
```

增加新的应用来进行检索

请求参数

参数名称	类型	描述
indexName	string array	应用名称或应用名称列表。

函数定义

```
function addIndex($indexName)
```

使用示例

```
$search_obj->addIndex("my_app_1");
$search_obj->addIndex("my_app_2");
$search_obj->addIndex("my_app_3");
```

删除待搜索的应用

在当前检索中删除此应用的检索结果。

请求参数

参数名称	类型	描述
indexName	string	待删除的应用名称

函数定义

```
function removeIndex($indexName)
```

使用示例

```
$search_obj->removeIndex("my_app_3");
```

获得请求应用列表

当前请求中所有的应用名列表。

返回结果

类型	描述
array	返回当前搜索的所有应用列表。

函数定义

```
function getSearchIndexes()
```

使用示例

```
$result = $search_obj->getSearchIndexes();
```

设置表达式名称

此表达式名称和结构需要在网站中已经设定。

请求参数

参数名称	类型	描述
formulaName	string	表达式名称。

函数定义

```
function setFormulaName($formulaName)
```

使用示例

```
// 例如我们在控制台已经设定了一个名称为my_formula_name的排序表达式。  
$search_obj->setFormulaName("my_formula_name");
```

获取表达式名称

获得当前请求中设置的表达式名称。

返回结果

类型	描述
----	----

string

返回当前设定的表达式名称。

函数定义

```
function getFormulaName()
```

使用示例

```
$result = $search_obj->getFormulaName();
```

清空精排表达式名称设置

函数定义

```
function clearFormulaName()
```

使用示例

```
$search_obj->clearFormulaName();
```

设置粗排表达式名称

此表达式名称和结构需要在网站中已经设定。

请求参数

参数名称	类型	描述
FormulaName	string	表达式名称。

函数定义

```
function setFirstFormulaName($formulaName)
```

使用示例

```
// 例如我们在控制台已经设定了一个名称为my_first_formula_name的排序表达式。  
$search_obj->setFirstFormulaName("my_first_formula_name");
```

获取粗排表达式设置

获取当前设置的粗排表达式名称。

返回结果

类型	描述
string	返回当前设定的表达式名称。

函数定义

```
function getFirstFormulaName()
```

使用示例

```
$result = $search_obj->getFirstFormulaName();
```

清空粗排表达式名称设置

函数定义

```
function clearFirstFormulaName()
```

使用示例

```
$search_obj->clearFirstFormulaName();
```

添加一条summary信息

请求参数

参数名称	类型	描述
fieldName	string	指定的生效的字段。此字段必需为可分词的text类型的字段。
len	string	指定结果集返回的词字段的字节长度，一个汉字为2个字节。
element	string	指定命中的query的标红标签，可以为em等。
ellipsis	string	指定用什么符号来标注未展示完的数据，例如“...”。
snipped	string	指定query命中几段summary内容。
elementPrefix	string	如果指定了此参数，则标红的开始标签以此为准。
elementPostfix	string	如果指定了此参数，则标红的结束标签以此为准。

函数定义

```
function addSummary($fieldName, $len = 0, $element = "", $ellipsis = "", $snipped = 0, $elementPrefix = "", $elementPostfix = "")
```

使用示例

```
//title返回的字数长度为50，飘红用"<em></em>"，截取长度后用"..."来结束，返回的结果在一个片段（片段的概念就是返回结果在一段落中还是几段落中）中
$search_obj->addSummary('title',50,'em','...', 1, $elementPrefix = "", $elementPostfix = "");
//title返回的字数长度为250，飘红用"<em></em>"，截取长度后用"..."来结束，返回的结果在2个片段中
$search_obj->addSummary("body", 250, "em", "...", 2);
```

获取当前的summary信息

可以通过指定字段名称返回指定字段的summary信息

请求参数

参数名称	类型	描述
field	string	指定的字段，如果此字段为空，则返回整个summary信息，否则返回指定field的summary信息。

返回结果

类型	描述
array	返回summary信息。

函数定义

```
function getSummary($field = '')
```

使用示例

```
$result = $search_obj->getSummary('title');
```

获取summary字符串

把summary信息生成字符串并返回。

返回结果

类型	描述
string	返回字符串的summary信息。

函数定义

```
function getSummaryString()
```

使用示例

```
$result = $search_obj->getSummaryString();
```

设置返回的数据格式

请求参数

参数名称	类型	描述
format	string	数据格式名称，有xml, json和protobuf 三种类型。

函数定义

```
function setFormat($format)
```

使用示例

```
$search_obj->setFormat("json");
```

获取当前的数据格式名称

返回结果

类型	描述
string	返回当前的数据格式名称。

函数定义

```
function getFormat()
```

使用示例

```
$result = $search_obj->getFormat();
```

设置返回结果的offset偏移量

请求参数

参数名称	类型	描述
start	int	偏移量。

函数定义

```
function setStartHit($start)
```

使用示例

```
$search_obj->setStartHit(100);
```

获取返回结果的offset偏移量

返回结果

类型	描述
int	返回当前设定的偏移量。

函数定义

```
function getStartHit()
```

使用示例

```
$result = $search_obj->getStartHit();
```

设置结果集大小

设置当前返回结果集的doc个数。

请求参数

参数名称	类型	描述
hits	number	指定的doc个数。默认值：20

函数定义

```
function setHits($hits = 20)
```

使用示例

```
//每页获取25条记录
$search_obj->setHits(25);
```

获取结果集大小

获取当前设定的结果集的doc数。

返回结果

类型	描述
number	返回当前指定的doc个数。

函数定义

```
function getHits()
```

使用示例

```
$result = $search_obj->getHits();
```

添加排序设置

增加一个排序字段及排序方式。 更多说明请参见API 排序sort子句

请求参数

参数名称	类型	描述
field	string	字段名称。
sortChar	string	排序方式，有升序+和降序-两种方式。

函数定义

```
function addSort($field, $sortChar = self::SORT_DECREASE)
```

使用示例

```
//按创建时间倒序获取搜索结果
$result = $search_obj->addSort("create_timestamp", '-');
```

删除指定字段的排序

请求参数

参数名称	类型	描述
field	string	指定的字段名称。

函数定义

```
function removeSort($field)
```

使用示例

```
$result = $search_obj->removeSort("create_timestamp");
```

获取排序信息

请求参数

参数名称	类型	描述
sortKey	string	如果此字段为空，则返回所有排序信息，否则只返回指定字段的排序值。

返回结果

类型	描述
string array	返回排序值。

函数定义

```
function getSort($sortKey = '')
```

使用示例

```
$result = $search_obj->getSort();
```

获取排序字符串

把排序信息生成字符串并返回。

返回结果

类型	描述
string	返回字符串类型的排序规则。

函数定义

```
function getSortString()
```

使用示例

```
$result = $search_obj->getSortString();
```

添加过滤规则

针对指定的字段添加过滤规则。 更多说明请参见 [API 过滤filter子句](#)

请求参数

参数名称	类型	描述
filter	string	过滤规则，例如 fieldName>=1。
operator	string	操作符，可以为 AND OR。默认值为：'AND'

函数定义

```
function addFilter($filter, $operator = 'AND')
```

使用示例

```
//cat_id 必需为可过滤或者可聚合属性
$search_obj->addFilter("cat_id=1");
```

获取过滤规则

返回结果

类型	描述
filter	返回字符串类型的过滤规则。

函数定义

```
function getFilter()
```

使用示例

```
$result = $search_obj->getFilter();
```

添加统计信息相关参数

一个关键词通常能命中数以万计的文档，用户不太可能浏览所有文档来获取信息。而用户感兴趣的可能是些统计类的信息，比如，查询“手机”这个关键词，想知道每个卖家所有商品中的最高价格。则可以按照卖家的user_id分组，统计每个小组中最大的price值：groupKey:user_id, aggFun: max(price)

更多说明请参见 API aggregate子句说明

请求参数

参数名称	类型	描述
groupKey	string	指定的group key.
aggFun	string	指定的function。当前支持：count、max、min、sum。
range	string	指定统计范围。
maxGroup	string	最大组个数。
aggFilter	string	表示仅统计满足特定条件的文档。
aggSamplerThresHold	string	抽样统计的阈值。表示该值之前的文档会依次统计，该值之后的文档会进行抽样统计。
aggSamplerStep	string	抽样统计的步长。

函数定义

```
function addAggregate($groupKey, $aggFun, $range = "", $maxGroup = "", $aggFilter = "", $aggSamplerThresHold = '', $aggSamplerStep = '')
```

使用示例

```
$search_obj->addAggregate($groupKey, $aggFun, $range, $maxGroup, $aggFilter, $aggSamplerThresHold, $aggSamplerStep);
```

删除指定的统计数据

请求参数

参数名称	类型	描述
groupKey	string	指定的group key.

函数定义

```
function removeAggregate($groupKey)
```

使用示例

```
$search_obj->removeAggregate("cat_id");
```

获取统计相关信息

请求参数

参数名称	类型	描述
groupKey	string	指定group key获取其相关信息，如果为空，则返回整个信息。

返回结果

类型	描述
array	统计相关信息

函数定义

```
function getAggregate($key = "")
```

使用示例

```
$result = $search_obj->getAggregate();
```

获取字符串类型的统计信息

返回结果

类型	描述
string	获取字符串类型的统计信息

函数定义

```
function getAggregateString()
```

使用示例

```
$result = $search_obj->getAggregateString();
```

添加distinct排序信息

例如：检索关键词“手机”共获得10个结果，分别为：doc1，doc2，doc3，doc4，doc5，doc6，doc7，doc8，doc9，doc10。其中前三个属于用户A，doc4-doc6属于用户B，剩余四个属于用户C。如果前端每页仅展示5个商品，则用户C将没有展示的机会。但是如果按照user_id进行抽取，每轮抽取1个，抽取2次，并保留抽取剩余的结果，则可以获得以下文档排列顺序：doc1、doc4、doc7、doc2、doc5、doc8、doc3、doc6、doc9、doc10。可以看出，通过distinct排序，各个用户的商品都得到了展示机会，结果排序更

趋于合理。

更多说明请参见 API distinct子句

请求参数

参数名称	类型	描述
key	string	为用户用于做distinct抽取的字段，该字段要求建立Attribute索引。
distCount	int	为一次抽取的document数量，默认值为1。
distTimes	int	为抽取的次数，默认值为1。
reserved	string	为是否保留抽取之后剩余的结果，true为保留，false则丢弃，丢弃时totalHits的个数会减去被distinct而丢弃的个数，但这个结果不一定准确，默认为true。
distFilter	string	为过滤条件，被过滤的doc不参与distinct，只在后面的排序中，这些被过滤的doc将和被distinct出来的第一组doc一起参与排序。默认是全部参与distinct。
updateTotalHit	string	当reserved为false时，设置update_total_hit为true，则最终total_hit会减去被distinct丢弃的数目（不一定准确），为false则不减；默认为false。
maxItemCount	int	设置计算distinct时最多保留的doc数目。
grade	number	指定档位划分阈值。

函数定义

```
function addDistinct($key, $distCount = 0, $distTimes = 0, $reserved = "", $distFilter = "", $updateTotalHit = "", $maxItemCount = 0, $grade = "")
```

使用示例

```
//每次抽取3条文档，抽取4次，丢掉抽取之后的结果，并且更新搜索结果中总文档的个数
$search_obj->addDistinct($key, 3, 4, false, "", true);
```

删除某个字段的所有distinct排序信息

请求参数

参数名称	类型	描述
------	----	----

distinctKey	string	指定的字段
-------------	--------	-------

函数定义

```
function removeDistinct($distinctKey)
```

使用示例

```
$search_obj->removeDistinct($distinctKey);
```

获取某字段的distinct排序信息

请求参数

参数名称	类型	描述
key	string	指定的distinct字段，如果字段为空则返回所有distinct信息。

返回结果

类型	描述
array	指定字段的distinct排序信息

函数定义

```
function getDistinct($key = "")
```

使用示例

```
$result = $search_obj->getDistinct();
```

获取字符串类型的所有的distinct信息

返回结果

类型	描述
string	字符串类型的所有的distinct信息

函数定义

```
function getDistinctString()
```

使用示例

```
$result = $search_obj->getDistinctString();
```

设定指定索引字段范围的搜索关键词

[NOTE]:\$query必须指定索引名称，格式类似为 索引名称:'搜索关键词'。此query是查询必需的一部分，可以指定不同的索引名，并同时可指定多个查询及之间的关系

(AND, OR, ANDNOT, RANK)。

例如查询subject索引字段的query: “手机” ，可以设置为

query=subject:'手机'。

NOTE: text类型索引在建立时做了分词，而string类型的索引则没有分词

更多说明请参见 API query子句

请求参数

参数名称	类型	描述
query	string	设定搜索的查询词。
fieldName	string	设定的索引范围。

函数定义

```
function setQueryString($query)
```

使用示例

```
$search_obj->setQueryString('default:阿里云');
```

获取当前指定的查询词内容

返回结果

类型	描述
string	当前指定的查询词内容

函数定义

```
function getQuery()
```

使用示例

```
$result = $search_obj->getQuery();
```

添加指定结果集返回的字段

请求参数

参数名称	类型	描述
field	array string	结果集返回的字段。

函数定义

```
function addFetchFields($field)
```

使用示例

```
//希望搜索结果只返回title,body,url和hit_count四个字段时的设置
$search_obj->addFetchFields("title");
$search_obj->addFetchFields("body");
$search_obj->addFetchFields("url");
$search_obj->addFetchFields("hit_count");
```

删除指定结果集的返回字段

请求参数

参数名称	类型	描述
fieldName	string	指定字段名称。

函数定义

```
function removeFetchField($fieldName)
```

使用示例

```
$search_obj->removeFetchField("url");
```

设置kvpair

更多说明请参见 [API 自定义kvpair子句](#)

请求参数

参数名称	类型	描述
pair	string	指定的pair信息。

函数定义

```
function setPair($pair)
```

使用示例

```
//例如我们在控制台中设计的表达式中需要根据用户的地理位置对结果进行排序
//在搜索时需要传入用户的地理位置信息，比如分别传入用户的经纬度信息
$search_obj->setPair("latitude:67.5367(N),longitude:4.768(E)");
```

获取当前的kvpair

返回结果

类型	描述
----	----

string

返回当前设定的kvpair。

函数定义

```
function getPair()
```

使用示例

```
$result = $search_obj->getPair();
```

增加自定义参数

请求参数

参数名称	类型	描述
paramKey	string	参数名称。
paramValue	string	参数值。

函数定义

```
function addCustomParam($paramKey, $paramValue)
```

使用示例

```
$search_obj->addCustomParam("xxx", "YYY");
```

指定精排算分的文档个数

若不指定则使用默认值200

请求参数

参数名称	类型	描述
rerankSize	int	精排算分文档个数

函数定义

```
function addRerankSize($rerankSize)
```

使用示例

```
$search_obj->addRerankSize(200);
```

获取精排算分文档个数

返回结果

类型	描述
int	精排算分文档个数

函数定义

```
function getRerankSize()
```

使用示例

```
$result = $search_obj->getRerankSize();
```

获取自定义参数

返回结果

类型	描述
string	自定义参数

函数定义

```
function getCustomParam()
```

使用示例

```
$result = $search_obj->getCustomParam();
```

获取指定结果集返回的字段列表

返回结果

类型	描述
array	指定结果集返回的字段列表

函数定义

```
function getFetchFields()
```

使用示例

```
$result = $search_obj->getFetchFields();
```

文档操作

CloudsearchDoc

构造函数

请求参数

参数名称	类型	描述
indexName	string	指定操作的索引名称。

client	CloudsearchClient	cloudsearch客户端
opts	array	可选参数

函数定义

```
function _construct($indexName, $client, $opts = array())
```

使用示例

```
$app_name = "你的应用名称";
$doc_obj = new CloudsearchDoc($app_name, $client);
```

查看文档

根据文档id获取doc的详细信息。

请求参数

参数名称	类型	描述
docId	string	指定的文档id。

返回结果

类型	描述
string	该docId对应的doc详细信息

函数定义

```
function detail($docId)
```

使用示例

```
$result = $doc_obj->detail($docId);
```

更新文档

向指定的表中更新doc。

请求参数

参数名称	类型	描述
docs	array	指定要更新的doc。
tableName	string	指定向哪个表中更新doc。

返回结果

类型	描述
string	返回API返回的结果。

函数定义

```
function update($docs, $tableName)
```

使用示例

```
$doc_obj->update($docs,$table_name);
```

添加文档

向指定的表中增加doc。

请求参数

参数名称	类型	描述
docs	array	指定要添加的doc。
tableName	string	指定向哪个表中增加doc。

返回结果

类型	描述
string	返回API返回的结果。

函数定义

```
function add($docs, $tableName)
```

使用示例

```
$doc_obj->add($docs,$table_name);
```

删除文档

删除指定表中的doc。

请求参数

参数名称	类型	描述
docs	array	指定要删除的doc列表，必须含有主键。
tableName	string	指定要从哪个表删除记录。

返回结果

类型	描述
string	返回API返回的结果。

函数定义

```
function remove($docs, $tableName)
```

使用示例

```
$doc_obj->remove($docs,$table_name);
```

通过csv格式文件上传文档数据

NOTE: 此文件必需为csv格式的文件（“,”分割）；且第一行为数据结构字段名称，例如：id, title, name, date, 1, "我的测试数据\1\测试1", test_name1, "2013-09-21 00:12:22"

...

请求参数

参数名称	类型	描述
fileName	string	本地文件。
primaryKey	string	指定此表的主键。
tableName	string	指定表名。
multiValue	array	指定此表中的多值的字段。默认值为空
offset	int	指定从第offset条记录开始导入。默认值为1
maxSize	number	指定每次push数据的最大值，单位为MB。默认值为4
frequence	int	指定上传数据的频率，默认值为4，单位为次/秒

返回结果

类型	描述
string	返回如果成功上传或上传失败的状态。

函数定义

```
function pushCSVFile($fileName, $primaryKey, $tableName,$multiValue = array(), $offset = 1, $maxSize = self::PUSH_MAX_SIZE,$frequence = self::PUSH_FREQUENCY)
```

使用示例

```
//其他字段使用默认值
$result = $doc_obj->pushCSVFile($fileName,$primaryKey,$tableName);
```

推送HA3格式文档

除了上面的方法还可以通过文件将文档导入到指定的表中 这里的文档需满足一定的格式，我们称之为HA3文档格式。HA3文件的要求如下：

文件编码：UTF-8

支持CMD: add, delete,update。

如果给出的字段不是全部，add会在未给出的字段加默认值，覆盖原值；update只会更新给出的字段，未给出的不变。

文件分隔符：

```
<pre>
```

编码-----描述-----显示形态

"\x1E\n" 每个doc的分隔符. ^^ (接换行符)

"\x1F\n" 每个字段key和value分隔 ^_ (接换行符)

"\x1D" 多值字段的分隔符 ^]

```
</pre>;
```

示例：

```
<pre>;
```

CMD=add^_

url=<http://www.opensearch.console.aliyun.com>^_

title=开放搜索^_

body=xxxxxxxxx^

multivalue_feild=123^]1234^]12345^

^^

CMD=update^_

...

```
</pre>
```

注意：文件结尾的分隔符也必需为"^^\n"，最后一个换行符不能省略。

请求参数

参数名称	类型	描述
fileName	string	指定HA3DOC所有在的路径。
tableName	string	指定要导入的表的名称。
offset	int	指定偏移行数，如果非0，则从当前行一下的数据开始导入。默认值为：1

maxSize	number	指定每次导入到api接口的数据量的大小，单位MB，默认值为：4
frequence	int	指定每秒钟导入的频率，单位次/秒，默认值为：4

返回结果

类型	描述
string	返回导入成功标志。

函数定义

```
function pushHADocFile($fileName, $tableName, $offset = 1, $maxSize = self::PUSH_MAX_SIZE, $frequence = self::PUSH_FREQUENCE)
```

使用示例

```
//其他参数使用默认值
$doc_obj->pushHADocFile($fileName,$tableName);
```

应用操作

CloudsearchIndex

构造函数

请求参数

参数名称	类型	描述
indexName	string	指定操作的应用名称。
client	CloudsearchClient	cloudsearch客户端

函数定义

```
function __construct($indexName, $client)
```

使用示例

```
$index_obj = new CloudsearchIndex($indexName,$client);
```

通过模板名称创建应用

用指定的模板名称创建一个新的应用。

请求参数

参数名称	类型	描述
------	----	----

templateName	string	模板名称(可以使系统内置模板，也可以是自定义模板)
opts	array	包含应用的备注信息。

返回结果

类型	描述
string	返回api返回的结果。

函数定义

```
function createByTemplateName($templateName, $opts = array())
```

使用示例

```
//使用内置的新闻应用模板创建一个应用
$index_obj->createByTemplateName("builtin_news");
```

通过模板创建应用

用指定的模板创建一个新的应用。模版是一个格式化数组,用于描述应用的结构，可以在控制台中通过创建应用->保存模板->导出模板来获得json结构的模板；也可以自己生成，格式见控制台模板管理。

请求参数

参数名称	类型	描述
template	string	使用的模板
opts	array	包含应用的备注信息。

返回结果

类型	描述
string	返回api返回的正确或错误的结果。

函数定义

```
function createByTemplate($template,$opts = array())
```

使用示例

```
$index_obj->createByTemplate($template);
```

修改应用名称和备注

更新当前应用的应用名称和备注信息。

请求参数

参数名称	类型	描述
------	----	----

toIndexName	string	更改后的新名字
opts	array	可选参数,包含： desc 应用备注信息

返回结果

类型	描述
string	API返回的操作结果

函数定义

```
function rename($toIndexName, $opts = array())
```

使用示例

```
$index_obj->rename("new_index_name");
```

删除应用

返回结果

类型	描述
string	API返回的操作结果

函数定义

```
function delete()
```

使用示例

```
$result = $index_obj->delete();
```

查看应用状态

返回结果

类型	描述
string	API返回的操作结果

函数定义

```
function status()
```

使用示例

```
$result = $index_obj->status();
```

列出所有应用

请求参数

参数名称	类型	描述
page	int	页码
pageSize	int	每页的记录条数

函数定义

```
function listIndexes($page = 1, $pageSize = 10)
```

使用示例

```
$result = $index_obj->listIndexes(1,10);
```

获取应用名称

获取当前应用的名称。

返回结果

类型	描述
string	当前应用的名称

函数定义

```
function getIndexName()
```

使用示例

```
$result = $index_obj->getIndexName();
```

获取应用的最近错误列表

请求参数

参数名称	类型	描述
page	int	指定获取第几页的错误信息。默认值：1
pageSize	int	指定每页显示的错误条数。默认值：10

返回结果

类型	描述
array	返回指定页数的错误信息列表。

函数定义

```
function getErrorMessage($page = 1, $pageSize = 10)
```

使用示例

```
$result = $index_obj->getErrorMessage(2,10);
```

相关下载

相关下载

最新稳定版

v2.0.4

- php_v2.0.4.zip
- github镜像即将开放

.NetSDK手册

相关下载

相关下载

最新稳定版

- AliCouldAPI_vs2010_v2.1.1.zip
- AliCouldAPI_vs2012_v2.1.1.zip