云搜

API文档

产品文档

腾讯云

文档目录

API文档

简介

最近更新时间：2017-03-09 04:29:17

欢迎使用腾讯云搜服务。腾讯云搜（Tencent Cloud Search）是腾讯公司基于在搜索领域多年的技术积累，通过对公司内部微信、QQ等各大垂直业务搜索需求进行高度抽象， 把搜索引擎组件化、平台化、服务化，形成的一套成熟的搜索开放能力。为广大移动应用开发者和网站站长推出的一站式搜索服务。

本文档提供的API供您使用请求调用的方式来操作云搜（Tencent Cloud Search）。请确保在使用这些接口前，已充分了解TCS产品、及其使用和收费方式。

在本文档的接口说明部分，凡出现任何参数可选范围等方面与腾讯云官网上给出的数值发生矛盾时，**均以官网上给出的值为准。**

云搜 API 概览

最近更新时间：2018-01-22 10:23:57

接口功能	Action ID
数据操作	DataManipulation
数据检索	DataSearch

调用方式
请求结构

最近更新时间：2017-03-09 05:00:36

1. 服务地址

腾讯云搜服务使用的域名访问地址为：yunsou.api.qcloud.com。

2. 通信协议

腾讯云API的所有接口均通过HTTPS进行通信，提供高安全性的通信通道。

3. 请求方法

同时支持 POST 和 GET 请求，需要注意不能混合使用。即如果使用 GET 方式，则参数均从 Querystring 取得；如果使用 POST 方式，则参数均从 Request Body 中取得，Querystring 中的参数将忽略。两种方式参数格式规则相同，一般使用GET，当参数字符串过长时使用POST，请见各接口详细描述。

4. 字符编码

均使用UTF-8编码。

5. API请求结构

名称	描述	备注
API入口	API调用的 WebService 入口	https://yunsou.api.qcloud.com/v2/index.php
公共参数	每个接口都包含的通用参数	详见 公共参数 页面
指令名称	API要执行的指令的名称，这里使用Action指定，	完整的指令请参见 API概览

	例如Action=DataSearch	
指令参数	每个特定的指令需要的参数	详见每个指令的接口文档

接口鉴权

最近更新时间：2018-05-25 16:56:49

腾讯云 API 会对每个访问的请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证用户身份。签名信息由用户所执有的安全凭证生成，安全凭证包括 SecretId 和 SecretKey，若用户还没有安全凭证，则需要在腾讯云官网上自主申请，否则就无法调用云API接口。

1. 申请安全凭证

在第一次使用云API之前，用户需要在腾讯云CVM控制台上申请安全凭证。安全凭证包括 SecretId 和 SecretKey，其中SecretId 是用于标识 API 调用者身份的，而SecretKey是用于加密签名字符串和服务器端验证签名字符串的密钥。用户应严格保管其SecretKey，避免泄露。

申请安全凭证的具体步骤如下：

1) 登录腾讯云管理中心控制台。

2) 单击【云产品】，选择【监控与管理】栏下的【云API密钥】，进入云API密钥管理页面。

3) 在云API访问密钥管理页面，单击【新建】即可以创建一对SecretId/SecretKey，每个帐号最多可以拥有两对SecretId/SecretKey。

2. 生成签名串

有了安全凭证SecretId 和 SecretKey后，就可以生成签名串了。下面给出了一个生成签名串的详细过程。

假设用户的 SecretId 和 SecretKey 分别是：

> SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA
> SecretKey: Gu5t9xGARNpq86cd98joQYCN3Cozk1qA

注意：这里只是示例，请用户根据自己实际的SecretId和SecretKey进行后续操作！
以查看实例列表(DescribeInstances)请求为例，当用户调用这一接口时，其请求参数可能如下:

参数名称	中文	参数值
Action	方法名	DescribeInstances

参数名称	中文	参数值
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	gz
instanceIds.0	待查询的实例ID	ins-09dx96dg
offset	偏移量	0
limit	最大允许输出	20

由上表可以看出，请求参数中的公共请求参数只有5个：Action、SecretId、Timestamp、Nonce和Region，而不是在“公共请求参数”中所述的6个，事实上，第6个参数Signature（签名串）正是由其他参数（包括指令请求参数）共同生成的，具体步骤如下：

2.1. 对参数排序

首先对所有请求参数按参数名做字典序升序排列，所谓字典序升序排列，直观上就如同在字典中排列单词一样排序，按照字母表或数字表里递增顺序的排列次序，即先考虑第一个“字母”，在相同的情况下考虑第二个“字母”，依此类推。您可以借助编程语言中的相关排序函数来实现这一功能，如php中的ksort函数。上述示例参数的排序结果如下:

```
{
    'Action' : 'DescribeInstances',
    'Nonce' : 11886,
    'Region' : 'gz',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA',
    'Timestamp' : 1465185768,
    'instanceIds.0' : 'ins-09dx96dg',
    'limit' : 20,
    'offset' : 0,
}
```

使用其它程序设计语言开发时, 可对上面示例中的参数进行排序，得到的结果一致即可。

2.2. 拼接请求字符串

此步骤生成请求字符串。

将把上一步排序好的请求参数格式化成“参数名称”=“参数值”的形式，如对Action参数，其参数名称为"Action"，参

数值为"DescribeInstances"，因此格式化后就为Action=DescribeInstances。
注意：1、“参数值”为原始值而非url编码后的值。2、若输入参数中包含下划线，则需要将其转换为“.”。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为:

```
Action=DescribeInstances&Nonce=11886&Region=gz&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3gr
```

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。
签名原文字符串由以下几个参数构成:

1) 请求方法: 支持 POST 和 GET 方式, 这里使用 GET 请求, 注意方法为全大写。
2) 请求主机:查看实例列表(DescribeInstances)的请求域名为：cvm.api.qcloud.com。实际的请求域名根据接口所属模块的不同而不同, 详见各接口说明。
3) 请求路径: 云API的请求路径固定为/v2/index.php。
4) 请求字符串: 即上一步生成的请求字符串。

签名原文串的拼接规则为:

> 请求方法 + 请求主机 +请求路径 + ? + 请求字符串

示例的拼接结果为：

```
GETcvm.api.qcloud.com/v2/index.php?Action=DescribeInstances&Nonce=11886&Region=gz&SecretId=
```

2.4. 生成签名串

此步骤生成签名串。
首先使用 HMAC-SHA1 算法对上一步中获得的**签名原文字符串**进行签，然后将生成的签名串使用 Base64 进行编码，即可获得最终的签名串。

具体代码如下，以 PHP 语言为例:

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3Cozk1qA';
$srcStr = 'GETcvm.api.qcloud.com/v2/index.php?Action=DescribeInstances&Nonce=11886&Region=gz&
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为:

```
NSI3UqqD99b/UJb4tbG/xZpRW64=
```

使用其它程序设计语言开发时, 可用上面示例中的原文进行签名验证, 得到的签名串与例子中的一致即可。

3. 签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

注意：如果用户的请求方法是GET，则对所有请求参数值均需要做URL编码。

如上一步生成的签名串为NSI3UqqD99b/UJb4tbG/xZpRW64=，则其编码后为NSI3UqqD99b/UJb4tbG/xZpRW64=。因此，最终得到的签名串请求参数(Signature)为：NSI3UqqD99b/UJb4tbG/xZpRW64=，它将用于生成最终的请求URL。

公共参数

最近更新时间：2017-03-09 09:01:35

公共参数是用于标识用户和接口鉴权目的的参数, 如非必要, 在每个接口单独的接口文档中不再对这些参数进行说明, 但每次请求均需要携带这些参数, 才能正常发起请求。

名称	类型	描述	必选
Action	String	接口指令的名称，例如: DescribeDomains	是
Region	String	区域参数，用来标识希望操作哪个区域的实例。可选: bj:北京 gz:广州 sh:上海 hk:香港 ca:北美	是
Timestamp	UInt	当前UNIX时间戳	是
Nonce	UInt	随机正整数，与 Timestamp 联合起来, 用于防止重放攻击	是
SecretId	String	由腾讯云平台上申请的标识身份的 SecretId 和 SecretKey, 其中 SecretKey 会用来生成 Signature 具体参考 接口鉴权 页面	是
Signature	String	请求签名，用来验证此次请求的合法性, 具体参考 接口鉴权 页面	是

一个典型的接口请求如下, Action=DescribeInstance表示查询云服务器实例的详情。

```
https://domain/v2/index.php?Action=DescribeInstances
&SecretId=xxxxxxx
&Region=gz
&Timestamp=1402992826
&Nonce=345122
&Signature=mysignature
&instanceId=101
```

其中instanceId为指令参数，其余为通用参数。

返回值
返回值结构

最近更新时间：2017-03-09 08:42:51

如无特别说明, 每次请求的返回值中, 都会包含下面的字段：

名称	类型	描述	必选
code	Int	返回结果的错误码，0表示成功，其它值表示失败。具体错误码的含义可以参考 错误码 页面	
message	String	请求结果	

例如：
使用公共参数部分的示例请求：

```
https://cvm/v2/index.php?Action=DescribeInstances&SecretId=xxxxxxx&Region=gz
&Timestamp=1402992826&Nonce=345122&Signature=mysignature&instanceId=101
```

可能的返回结果如下：

```
{
    "code":0,
    "message": "success",
    "instanceSet":
   [{
        "instanceId":"qcvm1234",
        "cpu":1,
        "mem":2,
        "disk":20,
        "bandwidth":65535,
        "os":"centos_62_64",
        "lanIp":"10.207.248.186",
        "wanIp":null,
        "status":0
   }]
}
```

错误码

最近更新时间：2017-03-09 11:08:29

响应包体内的返回码(code), 反映了腾讯云API调用和执行的概要结果.
当返回码不为 0 时, 表示请求未正常执行, 返回码也称为错误码, 错误描述(message)对该结果进行了细化补充，用户可根据错误码判断API的执行情况。
message在部分终端（例如浏览器），中文会显示成unicode编码，需要解码。

腾讯云 API 可能返回的错误码表如下：

错误代码	错误类型	描述
4000	请求参数非法	缺少必要参数，或者参数值格式不正确，具体错误信息请查看错误描述 message 字段。
4100	鉴权失败	签名鉴权失败，请参考文档中鉴权部分。
4200	请求过期	请求已经过期，请参考文档中请求有效期部分。
4300	拒绝访问	帐号被封禁，或者不在接口针对的用户范围内等。
4400	超过配额	请求的次数超过了配额限制，请参考文档请求配额部分。
4500	重放攻击	请求的 Nonce 和 Timestamp 参数用于确保每次请求只会在服务器端被执行一次,所以本次的 Nonce 和上次的不能重复, Timestamp 与腾讯服务器相差不能超过 2 小时。
4600	协议不支持	协议不支持，请参考文档说明。
5000	资源不存在	资源标识对应的实例不存在，或者实例已经被退还，或者访问了其他用户的资源。
5100	资源操作失败	对资源的操作失败，具体错误信息请查看错误描述 message 字段，稍后重试或者联系客服人员帮忙解决。
5200	资源购买失败	购买资源失败，可能是不支持实例配置，资源不足等等。
5300	资源购买失败	购买资源失败，余额不足。
5400	部分执行成功	批量操作部分执行成功, 详情见方法返回值。
5500	用户资质	购买资源失败，用户资质审核未通过。

	审核未通过	
6000	服务器内部错误	服务器内部出现错误，请稍后重试或者联系客服人员帮忙解决。
6100	版本暂不支持	本版本内不支持此接口或该接口处于维护状态等。注意: 出现这个错误时, 请先确定接口的域名是否正确, 不同的模块, 域名可能不一样。
6200	接口暂时无法访问	当前接口处于停服维护状态，请稍后重试。

异步任务接口返回格式

最近更新时间：2017-03-09 05:46:43

1. 普通异步任务接口返回格式

一次请求只能操作一个资源的异步任务接口，例如创建负载均衡，重置主机操作系统等。

名称	类型	描述	必选
code	Int	返回结果的错误码，0表示成功，其它值表示失败。	Yes
message	String	返回结果的错误信息	No
requestId	String	任务编号	Yes

2. 批量异步任务接口返回格式

一次请求能操作多个资源的异步任务接口，例如修改密码，启动机器，停止机器等。

名称	类型	描述	必选
code	Int	返回结果的错误码，0表示成功，其它值表示失败。	Yes
message	String	返回结果的错误信息	No
detail	Array	以资源ID为key, 返回对该资源操作的code,message,requestId	Yes

例如：

```
{
      "code":0,
      "message": "success",
      "detail":
      {
          "qcvm6a456b0d8f01d4b2b1f5073d3fb8ccc0":
         {
          "code":0,
          "message":"",
          "requestId":"1231231231231":,
         }
```

```
        "qcvm6a456b0d8f01d4b2b1f5073d3fb8ccc0":
      {
        "code":0,
        "message":"",
        "requestId":"1231231231232":,
      }
    }
}
```

> 注意：
> 资源全部操作成功，则最外层code为0
> 资源全部操作失败，则最外层code会返回5100
> 资源部分操作失败，则最外层code会返回5400
> 在第3种情况下，终端可以通过detail得到失败部分的操作信息。

示例代码

最近更新时间：2018-06-11 16:03:00

1. SDK代码下载

开发语言	github地址
PHP	https://github.com/QcloudApi/qcloudapi-sdk-php
Python	https://github.com/QcloudApi/qcloudapi-sdk-python
Java	https://github.com/QcloudApi/qcloudapi-sdk-java
.Net	https://github.com/QcloudApi/qcloudapi-sdk-dotnet
Node.js	https://github.com/CFETeam/qcloudapi-sdk

将示例代码中的 YOUR_SECRET_ID 和 YOUR_SECRET_KEY 替换成实际的 SecretId 和 SecretKey
示例代码仅供参考， 请根据实际情况使用。

2. PHP语言的示例代码

```
<?php


/***************真实调用时，需要根据不同接口修改下面的参数*********************************/
/***************此处以DescribeInstances为例说明 如何获取指定 instanceId 的虚拟机**********/

/*DescribeInstances 接口的 URL地址为 cvm.api.qcloud.com，可从对应的接口说明 “1.接口描述” 章节获取该
$HttpUrl="cvm.api.qcloud.com";

/*除非有特殊说明，如MultipartUploadVodFile，其它接口都支持GET及POST*/
$HttpMethod="GET";

/*是否https协议，大部分接口都必须为https，只有少部分接口除外（如MultipartUploadVodFile）*/
$isHttps =true;

/*需要填写您的密钥，可从 https://console.cloud.tencent.com/capi 获取 SecretId 及 $secretKey*/
$secretKey='XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX';
```

```
/*下面这五个参数为所有接口的 公共参数；对于某些接口没有地域概念，则不用传递Region（如DescribeDeals，
$COMMON_PARAMS = array(
        'Nonce'=> rand(),
        'Timestamp'=>time(NULL),
        'Action'=>'DescribeInstances',
        'SecretId'=> 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX',
        'Region' =>'gz',
);

/*下面这两个参数为 DescribeInstances 接口的私有参数，用于查询特定的虚拟机列表*/
$PRIVATE_PARAMS = array(
        'instanceIds.0'=> 'qcvm00001',
        'instanceIds.1'=> 'qcvm00002',
);


/******************************************************************************/


CreateRequest($HttpUrl,$HttpMethod,$COMMON_PARAMS,$secretKey, $PRIVATE_PARAMS, $isHttps);


function CreateRequest($HttpUrl,$HttpMethod,$COMMON_PARAMS,$secretKey, $PRIVATE_PARAMS, $
{
    $FullHttpUrl = $HttpUrl."/v2/index.php";

    /***************对请求参数 按参数名 做字典序升序排列，注意此排序区分大小写*************/
    $ReqParaArray = array_merge($COMMON_PARAMS, $PRIVATE_PARAMS);
    ksort($ReqParaArray);

    /*********************************生成签名原文*********************************
     * 将 请求方法, URI地址,及排序好的请求参数 按照下面格式 拼接在一起, 生成签名原文，此请求中的原文为
     * GETcvm.api.qcloud.com/v2/index.php?Action=DescribeInstances&Nonce=345122&Region=gz
     * &SecretId=AKIDz8krbsJ5yKBZQ  ·1pn74WFkmLPx3gnPhESA&Timestamp=1408704141
     * &instanceIds.0=qcvm12345&instanceIds.1=qcvm56789
     * ******************************************************************************/
    $SigTxt = $HttpMethod.$FullHttpUrl."?";

    $isFirst = true;
    foreach ($ReqParaArray as $key => $value)
    {
        if (!$isFirst)
        {
            $SigTxt = $SigTxt."&";
```

```
    }
    $isFirst= false;

    /*拼接签名原文时，如果参数名称中携带_，需要替换成.*/
    if(strpos($key, '_'))
    {
        $key = str_replace('_', '.', $key);
    }

    $SigTxt=$SigTxt.$key."=".$value;
}

/*********************根据签名原文字符串 $SigTxt，生成签名 Signature******************/
$Signature = base64_encode(hash_hmac('sha1', $SigTxt, $secretKey, true));


/***************拼接请求串,对于请求参数及签名，需要进行urlencode编码*******************/
$Req = "Signature=".urlencode($Signature);
foreach ($ReqParaArray as $key => $value)
{
    $Req=$Req."&".$key."=".urlencode($value);
}

/*********************************发送请求*********************************/
if($HttpMethod === 'GET')
{
    if($isHttps === true)
    {
        $Req="https://".$FullHttpUrl."?".$Req;
    }
    else
    {
        $Req="http://".$FullHttpUrl."?".$Req;
    }

    $Rsp = file_get_contents($Req);

}
else
{
    if($isHttps === true)
    {
        $Rsp= SendPost("https://".$FullHttpUrl,$Req,$isHttps);
    }
    else
```

```
        {
            $Rsp= SendPost("http://".$FullHttpUrl,$Req,$isHttps);
        }
    }

    var_export(json_decode($Rsp,true));
}

function SendPost($FullHttpUrl,$Req,$isHttps)
{

        $ch = curl_init();
        curl_setopt($ch, CURLOPT_POST, 1);
        curl_setopt($ch, CURLOPT_POSTFIELDS, $Req);

        curl_setopt($ch, CURLOPT_URL, $FullHttpUrl);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
        if ($isHttps === true) {
            curl_setopt($ch, CURLOPT_SSL_VERIFYPEER,  false);
            curl_setopt($ch, CURLOPT_SSL_VERIFYHOST,  false);
        }

        $result = curl_exec($ch);

        return $result;
}
```

TCS云搜相关接口
数据操作

最近更新时间：2018-01-18 18:44:52

接口描述

域名：yunsou.api.qcloud.com
接口名: DataManipulation
功能：对数据进行添加和删除操作

> **注意：**
>
> 上传的数据不能包含特殊字符，只能是支持 GBK 和 UTF8 互转的字符。

输入参数

参数名称	必选	类型	描述
appId	是	Int	云搜的业务 ID，用以表明当前数据请求的业务
op_type	是	String	操作类型，可为 `add` 或者 `del`
contents.n	是	Array	文档内容（UTF-8 编码，长度限制在 63 KB 以下）

输出参数

参数名称	类型	描述
code	Int	错误码， 0：成功，其他值：失败
message	String	错误信息

示例

添加输入

```
https://domain/v2/index.php?Action=DataManipulation
&op_type=add
&appId=1
&contents.0.NA=1000
&contents.0.TA=test1
&contents.0.TB=testtitle1
&contents.0.TC=testcontent1
&contents.1.NA=2000
&contents.1.TA=test2
&contents.1.TB=testtitle2
&contents.1.TC=testcontent2
```

删除输入

```
https://domain/v2/index.php?Action=DataManipulation
&op_type=del
&appId=1
&contents.0.doc_id=1000
&contents.1.doc_id=2000
```

> **注意：**
> 在命令行中输入 `contents.0.doc_id` 可能会出现为空的情况，建议使用数组。

输出


```
{
    retcode: 0,
    errmsg: "succ",
    data: {
        app_id: 14950002,
        result: [
        {
            doc_id: "1000",
            errno: 0,
            result: "succ"
        },
        {
            doc_id: "2000",
            errno: 0,
            result: "succ"
        }
        ],
```

```
        seq: 1427872563,
        total_result: "succ"
    },
    code: 0,
    message: ""
}
```

数据检索

最近更新时间：2018-01-18 18:04:03

1. 接口描述

域名：yunsou.api.qcloud.com
接口名: DataSearch

对数据进行查询操作

2. 输入参数

参数名称	必选	类型	描述
appId	是	Int	云搜的业务ID，用以表明当前数据请求的业务
search_query	是	String	检索串
page_id	是	Int	当前页，从第0页开始计算
num_per_page	是	Int	每页结果数
search_id	否	String	当前检索号，用于定位问题，建议指定并且全局唯一
query_encode	否	Int	请求编码，0表示utf8，1表示gbk，建议指定
rank_type	否	Int	排序类型
num_filter	否	String	数值过滤，结果中按属性过滤（格式参见后面说明）
cl_filter	否	String	分类过滤，导航类检索请求（格式参见后面说明）
extra	否	String	检索用户相关字段（格式参见后面说明））
source_id	否	Int	检索来源
second_search	否	Int	是否进行二次检索，0关闭，1打开,默认是0
max_doc_return	否	Int	指定返回最大篇数，无特殊原因不建议指定，默认300篇
is_smartbox	否	Int	是否smartbox检索，0关闭，1打开,默认是0
enable_abs_highlight	否	Int	是否打开高红标亮，0关闭，1打开,默认是0

qc_bid	否	Int	指定访问QC业务ID

3. 输出参数

参数名称	类型	描述
code	Int	错误码, 0: 成功, 其他值: 失败
message	String	错误信息

4.过滤说明

腾讯云搜主要包括两类过滤方式：数值过滤和分类过滤。
数值过滤
对应检索请求中的num_filter字段，基本表达式可以描述为:

```
[N:meta:start:end]
```

该表达式含义为：属性满足start<=meta值<=end的文档将会被留下，其它的被过滤。其中，meta字段必须为数值字段
另外，数值过滤表达式之间支持"与""或"关系。用&和|符号连接，支持使用括号标识优先级。如，

```
[N:meta1:start:end]&([N:meta2:start:end]|[N:meta3:start:end])
```

分类过滤
对应检索请求中的cl_filter字段，基本表达式可以描述为：

```
[C:meta:value]
```

该表达式含义为：对该meta域使用特殊的倒排索引进行检索，特别适用于导航检索。其中meta字段可为数字或数值。
另外，分类检索表达式支持"与""或"关系。用&和|符号连接，支持使用括号标识优先级。如，

```
[C:meta1:value1]&([C:meta2:value2]|[C:meta3:value3])。
```

5.检索字段附加说明

extra相关性排序字段说明
检索请求中包含extra字段用来标识当次检索结果排序的情况。其基本格式为：

```
META1_TYPE1_META2_TYPE2_rel_TYPE3
```

以上表达式表示的含义为，排序分成三档，仅在前一档值相同情况下，才会进行下一档排序:
第一档按照META1的属性值进行排序，排序类型为TYPE1

第二档按照META2的属性值进行排序，排序类型为TYPE2

第三档按照rel（即文本相关性得分进行排序），排序类型为TYPE3。注意：rel字段的值非业务数据，而是搜索侧的计算结果

表达式取值方法：
1) Meta：meta本身为属性字段名即可（不含下划线，避免冲突），但是需要保证需要进行排序的属性是数值字段

2) Type:

Type=0: 当前属性越小，排序越靠前

Type=1：当前属性越大，排序越靠前

rank_type字段说明
1) rank_type=0：文本相关性打分降序排列

2) rank_type=1：文本相关性打分升序排列

3) rank_type=2：按照用户输入的extra字段，结合用户在高级组件中定制的排序对返回结果的顺序进行排序

4) rank_type=5：不使用额外的排序策略，按搜索结果直接返回

6.示例

输入

```
https://domain/v2/index.php?Action=DataSearch
&appId=1
&search_query=qq
&page_id=0
&num_per_page=10
```

输出

```
{
    "code": 0,
```

```
    "message": ""
    "data": {
        "cost_time": 19,
        "display_num": 2,
        "echo": "",
        "eresult_num": 2,
        "result_list": [
            {
                "doc_id": "200",
                "doc_meta": "{
                            "AA": "41",
                            "NA": "200",
                            "TA": "qq",
                            "TB": "cloudsearchdoc",
                            "TC": "qq"
                            }",
                "l2_score": 0,
                "search_debuginfo": ""
            },
            {
                "doc_id": "11005",
                "doc_meta": "{
                            "AA": "41",
                            "NA": "200",
                            "TA": "qq",
                            "TB": "cloudsearchdoc",
                            "TC": "qq"
                               }",
                "l2_score": 0,
                "search_debuginfo": ""
            }
        ],
        "result_num": 2,
        "seg_list": [
            {
                "seg_str": "qq"
            },
            {
                "seg_str": "腾讯"
            }
        ]
    },
}
```