

百度站内搜索 API 文档

(自定义数据版)

百度站内搜索

2016.12

目录

简介	2
使用流程详细说明	3
一、创建搜索引擎	3
二、自定义数据结构	4
三、注册开发者并创建应用	5
四、使用 OPENAPI 接口推送数据	8
五、使用客户端 API/SDK 获取数据	9
六、保证百度品牌露出	9
站内搜索 API 使用方法	10
一、WEB 搜索 API	10
1、代码示例	10
2、代码说明	12
3、方法介绍	13
二、Android SDK	16
1、引入 Android SDK	16
2、Android SDK 使用说明	16
3、代码示例	16
4、方法说明	17
5、Android SDK 语音搜索功能	18
三、IOS SDK	22
1、引入 IOS SDK	22
2、IOS SDK 使用说明	22
3、代码示例	22
4、参数说明	23
联系方式	24

简介

百度搜索 API 是一套由百度搜索（zn.baidu.com）提供的应用程序接口，它能够帮助您低成本地在网站或应用程序中构建起稳定、精准、功能和样式均可自由定制的搜索引擎，进而提升站内或应用内的搜索体验。

站内搜索 API 由“数据提交 API”和“结果获取 API”两部分组成：

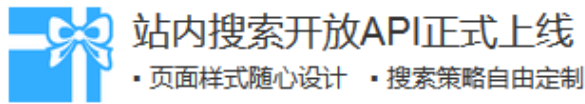
- 通过数据提交 API，您可以：
 - 1) 自由定制用于检索、排序、筛选、展示的数据结构
 - 2) 实现数据的实时推送
- 通过结果获取 API，您可以：
 - 1) 调整搜索算法中各个字段的权重
 - 2) 指定排序和筛选规则
 - 3) 获取搜索结果，按照网站或应用程序的需求对搜索结果进行展示

使用流程详细说明

一、创建搜索引擎

首次登陆，您需要在如下位置点击“创建 API 引擎”

搜索引擎管理 **API引擎管理**



创建API引擎

更多关于API的详细说明，请阅读[百度站内搜索API指导文档](#)

最多可以创建 10 个 API 搜索引擎



搜索引擎管理 **API引擎管理**

API模板管理

+ 新建API引擎

API引擎名称

API引擎信息

er

Web 搜索已有数据格式 [下载指导文档](#)

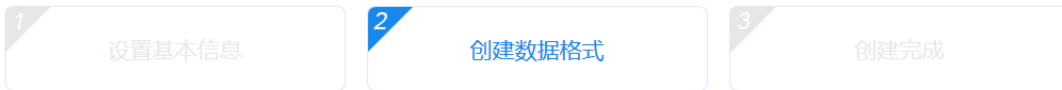
您可以在如下页面按照您的需求选择您需要的 API 客户端类型与数据格式选定您需要使用的 API 客户端类型后，选择“自定义数据格式”后点击下一步即可。

API模板管理

+ 新建API引擎

API引擎名称	API引擎信息	API引擎ID(sid)	操作
3	Web 搜索自定义数据格式 下载指导文档	5518507018280221648	基本信息 查看格式 代码示例 统计报表

二、自定义数据结构



您可以根据网站的搜索需求，定义需要使用的字段，并指定各字段的用途，之后按照模板规定的格式提交数据即可。

	字段名称	类型	可检索	可筛选	可排序	可展示	飘红	示例
1	id	主键	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	sc12345
2	<input type="text" value="请输入字段名称"/>	字符串	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="http://zn.baidu.com"/>
3	<input type="text" value="请输入字段名称"/>	整数	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="123456789"/>
4	<input type="text" value="请输入字段名称"/>	浮点数	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text" value="123.45"/>
5	<input type="text" value="请输入字段名称"/>	文本	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="text" value="样例文本，便于您识别管理"/>

[+ 新增字段](#)

在创建新模板页面，根据您网站的数据结构、搜索需求，按照页面提示进行填写：

- 1) 填写需要的数据字段：填写字段名称（仅由字母、数字、下划线组成，且第一位以字母开头；同一个模板中的字段名称不可重复），选择字段类型，指定该字段的用途。
- 2) 按需求填写示例：示例用于帮助您确认字段的含义，可以不填。

需要注意的是：

- 保存后不能再修改，请务必在点击“保存”前确认字段是否完整和各字段信息是否正确
- 模板可以删除，一旦删除模板，该模板下提交的数据也会被删除
- 如需要使用时间，请使用整数类型，填入 unix 时间戳
- 文本类型最多容纳 100K 内容，字符串类型最多容纳 10K 内容
- 保存后可以查看示例（如下图），json 样例即是提交的数据格式

代码示例

[返回API模板列表](#) > [查看示例](#)

请按照以下json样例，通过站内搜索API接口来推送数据：

```
{  
  "action": "add",  
  "id": "111",  
  "fields": {  
    "title": "样例文本，便于您识别管理",  
    "content": "样例文本，便于您识别管理",  
    "genre": ["样例文本，便于您识别管理"],  
    "url": "http://zn.baidu.com",  
    "date": "123456789"  
  }  
}
```

三、注册开发者并创建应用

如果检测到您并不是百度开发者，您需要点击“注册成为百度开发者”

新建API引擎帮助 退出

1 设置基本信息

2 创建数据格式

3 注册成为百度开发者

sitemap_test5，您还不是百度开发者，点击按钮去百度开放服务平台注册：

[注册成为百度开发者](#) [查看帮助](#)

进入到如下界面：



填写完开发者信息并验证邮箱完毕后即可提交应用，点击首页上方的“开发者服务管理”：



点击创建工程即可新建一个工程。



完成信息的填写之后即可成功的创建一个工程（创建应用）。

< 创建工程

* 应用名称: 0/32

传统接入扩展: 合作网站

解决方案: 使用BAE


创建

创建工程完毕后可以在工程列表中点击工程名称查看该工程的 API Key 与 Secret Key。

基本信息



名称: 站内搜索 

Icon: 

ID: 6059674

API Key: 

Secret Key: 

创建时间: 2015-05-25 11:27:47

更新时间: 2015-06-09 19:08:17

四、使用 OPENAPI 接口推送数据

首先，需要完成 OPENAPI 的 [OAuth 授权](#)，流程选择 [Authorization Code](#)，获取 access_token。

请求说明：<http://developer.baidu.com/wiki/index.php?title=docs/oauth/rest/overview>

接口地址：<https://openapi.baidu.com/rest/2.0/cse/document>

请求方式：POST

请求参数：

sid	搜索引擎 ID
content	结构体，批量每次不超过 100 条
access_token	OAuth 授权获得

content: json 数据（添加单条数据示例）

```
[{
  "action": "add",
  "id": "sc12345",
  "fields": [
    "title": "http://zn.baidu.com",
    "count": "123456789",
    "date": "1432017196",
    "pierce": "123.45",
    "content": "样式文本，便于您识别管理",
    "files": "http://zn.baidu.com"
  ]
}]
```

action=add 数据 id 一样时，字段数据会被更新。

content: json 数据（批量删除数据示例）

```
[
  {
    "action": "delete",
    "id": "sc12345",
  },
  {...}
]
```

action=delete 数据 id 一样时，该条数据会被删除，删除后即无法检索。

请求示例：

```
curl -X POST 'http://openapi.baidu.com/rest/2.0/cse/document' -d
'sid={sid}&access_token={token}&content={json}'
```

其中{sid}、{access_token}、{json}要替换为实际的值

注意：

- 接口一次提交 `content` 内容最多为 100 条的 `json` 对象数组；
- 如果提交的数据 (`action=add`) `id` 已存在，则更新 `doc` 内容；
- 每个 API 模板能容纳 100 万条 `doc` 数据，数据超出一定量后将不能继续使用接口推送 `add` 型 (`action=add`) 数据，`delete` 型 (`action=delete`) 可以正常提交。

调用频率说明：（频率控制策略可能会修改，请注意更新文档）

应用每分钟可访问次数	10000
应用每小时可访问次数	500000
应用每天可访问次数	1000000
用户每小时可访问次数	5000000

五、使用客户端 API/SDK 获取数据

详见本文档[站内搜索 API 使用方法](#)章节

六、保证百度品牌露出

在展示站内搜索结果的页面上保证百度搜索的品牌露出，即用户在您的搜索结果页面上可以见到以下内容：

搜索技术由  百度 提供

其中百度 logo 的下载地址为：<http://t.cn/RALpDXZ>

我们会对您使用 API 服务的页面进行检测，未参照该标准的站点将停止提供 API 服务。

站内搜索 API 使用方法

一、WEB 搜索 API

- 调整搜索算法中各个字段的权重
- 指定排序和筛选规则
- 指定搜索的页面范围
- 获取搜索结果，按照网站或应用程序的需求对搜索结果进行展示

1、代码示例

搜索引擎创建完毕后可以查看示例获取代码。



恭喜您！您的API搜索引擎“df”已成功创建！

您的自定义格式下尚未提交任何数据，请通过站内搜索API接口推送数据

[查看示例](#)

在 API 引擎管理页也可以选择相应引擎来查看代码示例

操作

[基本信息](#) [提交数据](#) [代码示例](#) [统计报表](#)

以下提供的示例仅作为说明，实际代码已后台页面为准。

```
1 <script type="text/javascript">
2
3     var cse;
4
5     //处理结果回调函数示例
6     function display (data) {
7         console.log(data);
8     }
9
10    function init () {
11
12        //参数 1 为模版 id，已自动填写；参数 2 为搜索引擎类型，此处必须为 2。
13        cse = new BCse.Search("2202075344871900467", 2);
14        var queryList = {
15            "title": 3,
16            "content": 3,
17            "url": 1,
18            "author_name": 1
19        }
20
21        //必须，此方法设置您想要检索的字段及各自的权重。
22        cse.setQueryList(queryList);
23        //其他方法类似调用
24
25        //此方法获取搜索结果，参数 1 为搜索词
26        //参数 2 为您获取到结果后想要执行的回调函数。
27        cse.getResult("关键词", callback);
28    }
29    function loadScript () {
30        var script = document.createElement("script");
31        script.type = "text/javascript";
32        script.charset = "utf-8";
33        script.src =
34        "http://zhannei.baidu.com/api/customsearch/apiaccept?sid=2202075344871900467&v=2.0&callback=init";
35        var s = document.getElementsByTagName('script')[0];
36        s.parentNode.insertBefore(script, s);
37    }
38    loadScript();
39
40 </script>
```

2、代码说明

上述示例中：

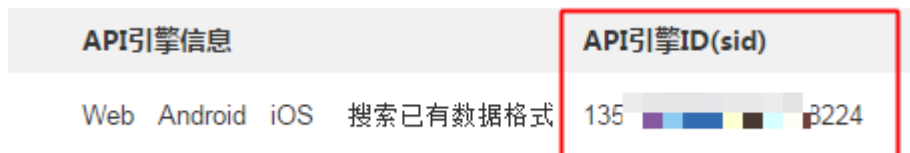
```

33 script.src =
34 "http://zhannei.baidu.com/api/customsearch/apiaccept?sid=2202075344871900467&v=2.0&
35 callback=init";

```

sid	搜索引擎 ID
v	版本号，目前填写 2.0
callback	可选参数，如果您希望异步加载，则可使用此参数定义异步加载时用到的函数名

- **BCse.Search** 为站内搜索 API 核心类，使用 API 首先需要初始化一个此类的对象，此对象变量**建议定义成全局变量**。使用异步加载，主要目的就是在回调函数中初始化此类。初始化此类时，需将 API 引擎 ID，API 引擎 ID 可在后台如下位置获取。



- 调用 **cse.getResult("查询词",callback)** 即执行了一次最简单的搜索。其中第一个参数为待要查询的词语，第二个参数为回调函数。返回的结果数据作为回调函数 **callback** 的参数，返回数据的结构会在下面介绍。

上述示例中：

```

14 var queryList = {
15     "title": 3,
16     "content": 3,
17     "url": 1,
18     "author_name": 1
19 }
20
21 //必须，此方法设置您想要检索的字段及各自的权重。
22 cse.setQueryList(queryList);

```

- 此处 **setQueryList** 方法必须设置，传入参数表示搜索字段权重配比。这里所出现的字段必须为设置了可检索的字段。

3、方法介绍

主要设置方法:

方法名	入参	说明
setQueryList	{ "title": 3, "content": 3, "url": 1, "author_name": 1 }	设置检索字段,以及对应权重表示在 title、content、author 中搜索关键字, 比重 3:2:1 必须, 为 getResult 前置方法, 取值范围 1-100
setDispList	['title','content','author']	设置需要返回的字段,不设置默认返回所有
setFacet	{ fields: ["author"] size: 10, mincount: "" }	fields: 统计字段 size: 限定返回统计结果的条数。默认为 10。 mincount: 只有高于这个值统计信息才会返回
openFieldFilter	{'domain':'baidu.com'}	对于设置了可筛选的字段使用
openFieldOrder(feildObj)	{'time':0}	0: 降序, 1: 升序
setPageNum(num)	10	返回结果数, 有效值为 1 到 10
closeOrder()		恢复默认排序
openGBK()		设置为 gbk 编码, 取决于 query 词编码。
closeFieldFilter()		关闭筛选
openJc()		开启关键词纠错功能。此功能默认开启。开启后, '习 dada' 将被纠正为 '习大大' 进行搜索, 只对结构化数据有效。
closeJc()		关闭关键词纠错功能
openSq()		开启关键词同义转换功能。此功能默认关闭。开启后, 'WOW' 将转换为 '魔兽世界 WOW' 进行搜索, 只对结构化数据有效。
closeSq()		关闭关键词同义转换功能。

获取搜索结果方法:

方法名	方法说明	参数说明
getResult(query,callback,[page])	获取搜索信息	query: String, 必需, 搜索词; callback: String, 必需, 回调函数; page: 可选, 默认 1, 第几页

➤ callback , 第一个参数名自定义, 下面以“resultData”为例说明相关字段;

- 调用 `getResult` 获取结果数据，本方法的第二个参数为回调函数。结果数据直接作为参数传入此回调函数中，您可以根据自己需求展现具体数据；
- 您可以任意定义结果数据的名字，如：`results`，`results` 是个数组，有几条结果则会有几条数据，以第一条数据为例 `resultData = results[0]`，从 `resultData` 中可以获取标题、摘要等具体值；
- 设置了飘红的字段，关键词会被嵌入在 `em` 标签中，如：`关键词`。

搜索信息相关方法:

方法名	方法说明	参数说明	备注
getSearchInfo(query,callback)	获取搜索信息	query: String, 必需, 搜索词; callback: String, 必需, 回调函数;	这两个方法需要在 getResult 方法的回调函数中调用, 否则无法生效。
getError(callback)	获取当前错误信息	callback: String, 必需, 回调函数;	

➤ callback , 第一个参数名自定义, 下面以“data”为例说明相关字段;

➤ 调用 getSearchInfo 方法可获取搜索相关信息, 以下是搜索相关信息的数据结构对应表

结果总条数	data. totalNum
搜索资源类型	data. tplId
当前页码	data. curPage
搜索所用时间	data. searchTime

➤ 调用 getError 可以获取错误信息, 数据结构如下

错误信息状态值	data. status
具体错误信息	data. reason

获取字段统计结果方法:

方法名	方法说明	参数说明
getFacet(query,callback)	获取由 setFacet 方法设置的字段统计结果	query: String, 必需, 搜索词; callback: String, 必需, 回调函数。

➤ 调用 getFacet 方法可获取由 setFacet 方法设置的字段统计结果, 请确保您在调用此方法前已经正确调用 setFacet 方法进行了相关的设置。

➤ getFacet 可以单独调用, 也可以在 getResult 的回调函数中调用, 两个方法不能并行调用。

获取搜索提示词方法:

方法名	方法说明	参数说明
getSug(query,callback,type)	获取搜索提示词	query: String, 必需, 搜索词; callback: String, 必需, 回调函数; type: String, 可选, normal、movie、music、novel、health、game、soft

二、Android SDK

1、引入 Android SDK

Android SDK 以 jar 包形式配置，[点此下载](#) jar 包。

2、Android SDK 使用说明

- 此 jar 包时用 JDK1.5 编译的，不支持 JDK1.5 以下版本开发的 APP。
- jar 包中调用接口获取数据，需网络通信服务，您的 Android 项目中需添加如下权限：

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
```

3、代码示例

查询代码示例：

```
//先实例化一个对象（第一个参数是模板 ID，第二个统一填 this）
CseSearch cseSearch = new CseSearch("模板 id", this);
//设置搜索参数
QueryInfo queryInfo = new QueryInfo();
queryInfo.setResultType(7);
queryInfo.setQuery("xxx");
//发送搜索请求，并获取结果（参数 queryInfo 为封装后的 QueryInfo 类对象）
ReturnInfo returnInfo = cseSearch.getResult(queryInfo);
```

获取返回结果代码示例（**标题和摘要中需要飘红的字段用xxx标签标注了，您需要自行处理**）：

```
//返回的数据封装在 ReturnInfo 类的 openSchemaResults 属性中
//获取 OpenSchema 结果示例，无需引入 json 相关 jar 包
ArrayList<JSONObject> results = returnInfo.getOpenSchemaResults();
//以第一条结果为例，假如您设置的字段中有一个名为 title
String title = results.get(0).getString("title");

//如果设置了按字段统计，即正确调用了 setOpenSchemafa 方法，则可以获取统计结果
JSONObject facetJson = returnInfo.getOpenSchemaFacet();
//假如您设置的统计字段为 title，以下代码可以获取此字段的统计，结果为数组
JSONArray facetArray = facetJson.getJSONArray("title");
//以其中第一条数据为例
```

```
JSONObject facetDetailJson = (JSONObject)facetArray.get(0);
int count = facetArray[0].getInt("count");
String value = facetArray[0].getString("value");
```

4、方法说明

QueryInfo 用来设置搜索的参数，具体方法如下：

必须调用的方法：

方法名	方法说明	参数说明
setResultType(int)	设置搜索类型	此参数必须为 7
setOpenSchemaqf(HashMap<String, String>)	用于设置检索字段，表示搜索词在哪个字段中查询	参数为 HashMap 类型，第一个值为检索字段，第二个值为权重

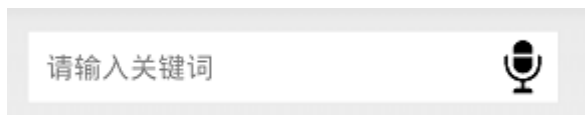
非必须调用的方法：

方法名	方法说明	参数说明
setOpenSchemafl(String[])	表示需要展示的字段，不调用此方法则返回所有字段	参数为数组类型，数组中为需要展示的字段
setOpenSchemafa(HashMap<String, String>)	用于统计相关字段的搜索结果条数	Map 中 key 可以取以下四个值。 fields: 用于指定统计哪个字段，对应的 value 为此字段名，如果多个字段中间用 & 分隔。 size: 限定返回统计结果的条数。默认为 10。 mincount: 只有高于这个统计值才会返回。 示例： <pre>HashMap<String, String> filedfa = new HashMap<String, String>(); filedfa.put("fields", "url&image"); filedfa.put("size", "1"); queryInfo.setOpenSchemafa(filedfa);</pre>
setFieldFilter(HashMap<String, String>)	按某字段筛选	key 为要筛选的字段，value 为此字段对应的值
setFieldOrder(String,int)	按某字段排序	第一个为要排序的字段值，第二个值为 0 表示正排，1 表示倒排
setQuery(String)	设置查询关键词	

5、Android SDK 语音搜索功能

语音搜索功能是百度站内搜索向使用站内搜索 SDK 服务的用户提供的用于 APP 内的语音搜索功能，目前提供 Android 版本的服务。使用此服务可以在您的搜索框内添加一个语音话筒图标，从而进行语音搜索。

点击下图所示搜索框内的话筒图标，您就可以进入搜索界面：



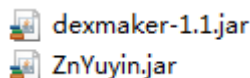
搜索界面如下图所示，按住下方图标说话就能进行语音搜索：



请按照下面的方法使用语音搜索功能。

(1) 引入 Android 语音 jar 包

站内搜索语音功能以 jar 包形式配置，共包含 2 个 jar 包，如下图所示，第一个是依赖包，第二个是语音 jar 包。



[点击下载 jar 包。](#)

下载后解压，添加到 android 工程中即可。

(2) 使用说明

- 此 jar 包是用 JDK1.5 编译的，不支持 JDK1.5 以下版本开发的 APP。
- jar 包中调用接口获取数据，需网络通信服务，录音服务等，您的 Android 项目中需添加如下权限：

```
<uses-permission
android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.READ_SYNC_SETTINGS" />
<uses-permission android:name="android.permission.WRITE_SYNC_SETTINGS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.BROADCAST_STICKY" />
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
<uses-permission android:name="com.tencent.mm.permission.C2D_MESSAGE"/>
<uses-permission android:name="com.android.vending.BILLING"/>
<uses-permission android:name="android.permission.READ_PROFILE" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
<uses-permission
android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
```

- 在解析语音 apk 时，同时需要您在 AndroidManifest.xml 文件中添加：

```
<activity android:name="androidx.pluginmgr.PluginActivity" />
```

(3) 代码示例

运行语音界面时，添加如下代码：

```
ZnYuyin yuyin = ZnYuyin.getInstance(context);
yuyin.setSearchParam("模板 id", QueryInfo);
yuyin.setListener(new SearchListener(){
    @Override
    public void getResult(ReturnInfo query) {
        // 录音后，解析到的数据，并执行后续动作
    }
});
```

- 解释：
- 1、context 参数为 Activity 对象的 this;
 - 2、SearchListener 为监听接口，实现 getResult 方法即可。
 - 3、QueryInfo 为搜索查询类
 - 4、ReturnInfo 为结果类

例如：您想查询一个录完音后的搜索结果，代码如下：

```
QueryInfo queryInfo = new QueryInfo();
queryInfo.setResultType(1);
String[] subDomains = {"zhanzhang.baidu.com"};
queryInfo.setSearchRange(2, subDomains);

ZnYuyin yuyin = ZnYuyin.getInstance(context);
yuyin.setSearchParam("模板 id ", queryInfo);

yuyin.setListener(new SearchListener(){
    @Override
    public void getResult(ReturnInfo query) {
        // 录音后，解析到的数据，并执行后续动作
        List<ResultInfo> returnInfos = query.getResultInfoList();
    }
});
```

(4) 方法和类说明

方法说明：

方法名	方法说明	参数说明	备注
getInstance	获取入口类的一个实例	传入 activity 的 context 参数	无

setListener(SearchListener)	注册监听器方法	将监听器类注册进去，提供实时监听语音返回的结果	无返回值
setSearchParam(sid,QueryInfo)	设置搜索参数 sid 和 query info 信息	模板 id 是你的搜索引擎 sid	无

类说明：

类名	类说明	备注
SearchListener	监听器类，包含一个 getResult 方法，必须实现，返回一个 ReturnInfo 类，为查询后的结果类，具体请看 SDK 类说明	无
ZnYuyin	语音初始化类，激活录音界面，准备录音。	无

(5) 开发案例

[点击此处](#) 下载语音搜索 demo 文件，以便于您参照开发。

三、IOS SDK

1、引入 IOS SDK

IOS SDK 以 framework 形式配置，[点此下载](#)相关压缩包。

2、IOS SDK 使用说明

- 此 SDK 支持 IOS6.0 以上版本；Xcode 版本需要在 6.0 以上
- 解压压缩包后，可以看到 framework 文件夹中共有两个文件夹。
 - release-iphonios: 发版真机模式下的 framework;
 - release-iphonesimulator: 发版模拟器模式下的 framework;
- 开发者可以根据需要使用真机或模拟器的包，如果需同时使用真机和模拟器的包，可以使用 lipo 命令将设备和模拟器 framework 包中的 CseSearchFrameWork 文件合并成一个通用文件。命令如下 lipo -create xxxx1 xxxx2 -output xxxx
- 在 TARGETS->Build Phases-> Link Binary With Libraries,将所需的 CseSearchFrameWork.framework 加入工程。同时在 TARGETS->General->Embedded Binaries 中也要添加。

3、代码示例

```
//需先引入头文件
#import <CseSearchFrameWork/CseSearchManager.h>
#import <CseSearchFrameWork/ CseModel.h>

//先进行部分必填参数的配置
[CseSearchManager defaultManagerWithCseSID:@"13844415381105243735"
                                cseQF: @"title=3"];

//执行搜索。searchWithText 后为要搜索的词，parameters 为搜索中设置的参数，success 为执行成功后的操作，failure 为失败后的操作
[[CseSearchManager defaultManager]
searchWithText:@"搜索词"
parameters:@{@"pn":@(10),@"p":@(0),@"facet":@"size=2&fields[]=title&fields[]=url"}
success:^(id responseObject) {
    // responseObject 即为整个返回结果，是 NSDictionary 类型
    NSArray *blockData = responseObject[@"blockData"];
    NSDictionary *resultData = blockData[0];
    //如果您的数据结构中有 title 字段，可如下获取
```

```

NSString *title = responseData[@"title"];

//如果传参中传了 facet 字段，则在 searchInfo 节点下会有 facet
NSDictionary *searchInfo = responseObject[@"searchInfo"];
NSDictionary * facets = searchInfo[@"facets "];
} failure:^(NSError *error) {
    //handle error message
};
    
```

4、参数说明

执行搜索方法前需配置的几个参数

参数	参数说明	备注
cseSID	API 引擎 ID	无
cseQF	用于设置检索字段，表示搜索词在哪个字段中查询，以及此字段的权重值。	如果想在 title、content 两个字段中查询，且权重分别为 3 和 2，则可配置 cseQF: @"title=3&content=2"

以下是搜索时可配置参数

参数	参数说明	备注
p	表示获取第几页结果，从 0 开始	JS API 和 Android SDK 中是从 1 开始，注意不要混淆
pn	每页显示多少条数据，参数有效值为 1 到 10。	大于 10 的参数会自动置为 10
flt	表示按某个字段筛选	一个示例（表示按 gameCategory 和 genre 这两个字段筛选，=后面表示要筛选的值）： @"flt":@"gameCategory=单机&genre=格斗游戏"
srt	表示按某个字段排序	参数中配置@"flt":@"datePublished"表示按发行时间倒排，配置@"flt":@"-datePublished"表示按发行时间正排
fl	表示要显示那些字段。不传此参数则显示所有字段。	如要显示"title","content","genre"几个字段，则配置@"fl":@"title&content&genre"
facet	表示根据哪些字段统计。fields 标识按哪些字段统计，size 用于限定返回统计的结果条数。	如果是按 url 和 image 两个字段表示统计，限定返回的统计结果条数为 2，则配置@"facet":@"size=2&fields[]=url&fields[]=image"

联系方式

邮件: zn@baidu.com