

百度站内搜索 API 文档

(普通版)

百度站内搜索

2016.03

目录

简介	2
使用流程详细说明	3
一、创建搜索引擎	3
二、选择 API 类型与数据提交类型	4
三、使用 API 获取数据	4
四、保证百度品牌露出	5
站内搜索 API 使用方法	6
一、Web 搜索 API	6
1、代码示例	6
2、代码说明	7
3、方法介绍	8
4、结果数据结构说明	11
二、Android SDK	14
1、引入 Android SDK	14
2、Android SDK 使用说明	14
3、代码示例	14
4、方法说明	14
5、结果数据结构说明	19
6、Android SDK 语音搜索功能	21
三、IOS SDK	26
1、引入 IOS SDK	26
2、IOS SDK 使用说明	26
3、代码示例	26
4、参数说明	27
5、结果数据结构说明	30
联系方式	33

简介

百度站内搜索 API 是一套由百度站内搜索 (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。

- 1) 常规搜索功能: API 核心功能, 返回搜索结果。
- 2) 排序功能: 指定用哪些字段进行排序 (如: 发表时间、游戏的下载次数、论坛的回复数等)。
- 3) 筛选功能: 指定用哪些字段进行筛选 (如: 标题、游戏的子分类、小说的更新状态等)。
- 4) 设置结果数据: 指定搜索的资源类型 (如通用、小说、游戏等), 指定查询的是第几页搜索结果, 指定每一页的搜索结果条数。
- 5) 指定搜索范围: 指定搜索结果的页面范围, 支持二级域名和子目录, 默认搜索全站。

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**



站内搜索开放API正式上线

▪ 页面样式随心设计 ▪ 搜索策略自由定制

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

 站内搜索

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、 选择 API 类型与数据提交类型

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，可多选。

1

设置基本信息

2

创建数据格式

3

创建完成

需要使用站内搜索API的客户端类型

- ☐ Web 开发（包括PC站点和WAP站点）
- ☐ Android APP 开发
- ☐ iOS APP 开发

API引擎的名称

为您的API搜索引擎命名，如“手机百度iOS版APP内搜索”

选择符合您搜索需求的数据格式

- ☒ 已有数据格式（当以下6种格式中的1种或多种能满足需求时，选择此项）

您可以通过sitemap提交以下格式的数据，使用Discuz!、织梦CMS、杰哥小说CMS建站的网站还可以通过对应插件提交

网页数据

通用数据

影视数据

小说数据

游戏数据

论坛数据

选定您需要使用的 API 客户端类型后，选择“已有数据格式”中适合您的一种或多种后点击下一步即可完成创建。

1

设置基本信息

2

创建数据格式

3

创建完成



恭喜您！您的API搜索引擎“1”已成功创建！

为了确保搜索效果，建议您尽快提交数据以保证百度对您的收录量

[去提交数据](#)

您还可以

[返回API引擎管理](#)

三、 使用 API 获取数据

不同客户端获取数据的方法具体可见站内搜索 API 使用方法请见[站内搜索 API 使用方法](#)。

四、 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，用户进行搜索时将跳转至站内搜索默认结果页。

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

- 以下是站内搜索 API 异步加载示例：

```
1 <script type="text/javascript">
2     var cse;
3
4     //处理结果回调函数示例
5     function display (data) {
6         console.log(data);
7     }
8
9     //初始化回调函数示例
10    function init () {
11        cse = new BCse.Search("1234567890987654321");    //参数为您的 API
12    引擎 ID，必需。
13        cse.getResult("关键词", display);    //此方法获取搜索结果，参数 1 为
14    搜索词，参数 2 为您获取到结果后想要执行的回调函数。
15    }
16
17    function loadScript () {
18        var script = document.createElement("script");
19        script.type = "text/javascript";
20        script.charset = "utf-8";
21        script.src = "http://zhannei.baidu.com/api/customsearch/apiaccept?sid=您
22    的 API 引擎 ID&v=2.0&callback=init";
23        var s = document.getElementsByTagName('script')[0];
24        s.parentNode.insertBefore(script, s);
25    }
26    loadScript();
27 </script>
```

- 以下是站内搜索 API 同步加载示例：

```
1 <script type="text/javascript" src="http://zhannei.baidu.com/api/customsearch/apiac
2 cept?sid=您的 API 引擎 ID&v=2.0"></script>
3 <script>
4     var cse = new BCse.Search("1234567890987654321");    //参数为您的 API
5     引擎 ID，必需。
6
7     //处理结果回调函数示例
8     function display (data) {
9         console.log(data);
10    }
11
12    cse.getResult("关键词", display);    //此方法获取搜索结果，参数 1 为搜索
13    词，参数 2 为您获取到结果后想要执行的回调函数。
14 </script>
```

2、代码说明

上述示例：

```
21 script.src = "http://zhannei.baidu.com/api/customsearch/apiaccept?sid=您的 API 引擎
22 ID&v=2.0&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 模板 id 作为参数传入。

API 模板 id 可在后台如下位置获取：

API 模板名称	创建时间	模板id
baidu_schema	2015-05-19 09:53:55	 模板id

- 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。
- 同步加载的使用过程中务必注意：必须在您想要放置搜索结果的区域已经加载完毕之后再调用我们的 API 服务，否则会导致错误。
- 因异步加载是在页面其余内容加载完毕之后进行，故适合页面加载速度较快的站点使用，否则会造成搜索结果的显示被您页面的其余内容所阻塞，影响体验。

3、方法介绍

站内搜索 API 提供一系列方法来获取特定条件下的结果数据，如：筛选、排序等。**初始化一个 BCse.Search**，经此类对象定义为 **cse**，则用 **cse.xxxx** 即可调用如下方法。以下会详细介绍各方法使用情况。

- 以下是设置搜索结果相关方法

方法名	方法说明	参数说明	备注
setResultType(num)	设置搜索资源类型	参数 0、1、2、3、5、6 分别代表网页、影视、小说、游戏、论坛、通用	不调用此方法则会默认为网页
setPageNum(num)	指定返回结果中每页条数。	num 的有效值为 1 到 10。	大于 10 的参数会自动置为 10
openGBK()	设置为 gbk 编码		当且仅当您传入的查询词为 GBK 编码时使用
closeGBK()	设置为 utf-8 编码		默认即为 UTF-8 编码，无需刻意调用。
openHiLight()	开启标题关键词飘红		
closeHiLight()	关闭标题关键词飘红		

- 以下是筛选相关方法

方法名	方法说明	参数说明	备注
openTitleFilter()	开启按标题筛选		网页、影视、论坛、通用样式可使用此方法
closeTitleFilter()	关闭按标题筛选		
openTimeFilter(num)	开启按时间筛选	num 参数为分钟值，比如要搜索最近一天的结果，则调用方法 openTimeFilter(1440)	网页、通用、影视、论坛样式可使用此方法
closeTimeFilter	关闭按时间筛选		
openFieldFilter(field Obj)	按结构化因子筛选（如游戏的“分类”，小说的“更新状态”等）	参数为键值对构成的对象，可是多个键值对。 示例： openFieldFilter({'gameCategory':'单机','genre':'格斗游戏'})	通用、小说、游戏论坛类可使用此方法（ 需提交结构化数据 ）。筛选所用到的结构化字段如 详细注释 1 所示。
closeFieldFilter()	关闭结构化因子筛选		

- 以下是排序相关方法

方法名	方法说明	参数说明	备注
openTimeOrder()	开启按时间排序		开启后将按时间倒排。网页、通用、影视、论坛样式可使用此方法
openFieldOrder(fieldObj)	按结构化因子排序（如游戏类的“下载次数”，小说类的“总点击”等）	参数为键值对构成的对象，可是多个键值对。示例： openFieldOrder({'datePublished':1})	通用、小说、游戏、论坛类可使用此方法（ 需提交结构化数据 ）。排序用到的结构化字段如 详细注释 2 所示。
closeOrder()	恢复默认排序		

- 以下是搜索核心方法

方法名	方法说明	参数说明	备注
getResult(query,callback,page)	获取结果数据	query: String, 必需, 搜索词; callback: String, 必需, 回调函数; page: num, 可选, 页码。	Callback 为您自己定义的回调函数名称。num 表示您希望获取第几页的数据，不传此字段则默认为第一页。 注: getResult 为最终获取结果方法，若想使用以上方法限定搜索结果，需在 getResult 前调用

- 以下是搜索信息相关方法

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;	

- 以下是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, 类型。

- **详细注释 1: 按结构化因子筛选**

1). 游戏类筛选用到的结构化字段如下

分类	gameCategory
子分类	gameSubCategory
标签	tag
类型	genre
设备	device

`openFieldFilter({'gameCategory':'单机','genre':'格斗游戏'})`表示按分类是单机、类型是格斗游戏来进行筛选，其余类推。

2). 小说类筛选用到的结构化字段如下

分类	genre
更新状态	updateStatus
阅读费用	trialStatus

`openFieldFilter({'genre':'玄幻小说', trialStatus:'免费'})`表示按分类是玄幻小说、阅读费用是免费来进行筛选，其余类推。

3). 通用类筛选用到的结构化字段如下（以下字段并非一定要是左侧解释的内容，您可以为某个字段自定义含义）

内容标签	tag
层级名称，或频道、版块名称	breadCrumb_title
层级首页地址（对应上面的breadCrumb_title）	breadCrumb_url
博客评论数等	replyCount
如精华、热门、置顶等属性	property
商品价格或折优惠后的现价	price_new
折优惠前的价格	price_old
折扣优惠的折扣值	discount_value
商品、活动、影视著作等的评价	review_rating
评分人数	review_count
地域范围，可以是地区、城市甚至国家	location_area

4). 论坛类筛选可使用此方法筛选出某个版块下的数据

示例（按“站长杂谈”版块筛选）：`openFieldFilter({'cse_block1': forumIn_forumName:站长杂谈'})`

• 详细注释 2：按结构化因子排序

1) 游戏类排序用到的结构化字段如下

发行时间	datePublished
修改时间	dateModified
下载次数	downloadCount

`openFieldOrder({'datePublished':0})`表示按发行时间倒排，`openFieldOrder({'datePublished':1})`表示按发行时间正排。其余类推。

2) 小说类排序用到的结构化字段如下

总点击	totalClick
更新时间	dateModified

方法调用方式与上述游戏类相同。

3) 通用类排序用到的结构化字段如下（以下字段并非一定要是左侧解释的内容，您可

以为某个字段自定义含义)

博客评论数等	replyCount
商品价格或打折优惠后的现价	price_new
打折优惠前的价格	price_old
折扣优惠的折扣值	discount_value
商品、活动、影视著作等的评价	review_rating
评分人数	review_count
开始时间	time_startDate
结束时间	time_endDate

方法调用方式与上述游戏类相同。

4) 论坛类排序用到的结构化字段如下

发表时间	cse_createTime
回复数	cse_category2
浏览数	cse_category3

• **详细注释 3：设置搜索范围 setSearchRange()**

- 1) 指定子目录方法示例：setSearchRange(1,['baidu.com/more'])。第一个参数为 1 代表类型为子目录，第二个参数是数组，里面传的值是指定的子目录名如 baidu.com/more，即为子目录（不支持小说/游戏/论坛）。
- 2) 指定二级域名示例：setSearchRange(2,[map.baidu.com','news.baidu.com'])。第一个参数为 2 代表类型为二级域名，第二个参数可传多个二级域名值。如 news.baidu.com，即为二级域名。
- 3) 指定 inurl: setSearchRange(3,['movie'])。第一个参数为 3 代表类型为 inurl，第二个参数为指定的 inurl 值。
Inurl 是指您希望搜索到的网页 url 都包含的一个字符串。
- 4) 采用以上三种规则中的任意一种都应注意，第二个参数为数组，数组的长度最好控制在 1-3 之间，否则会被截断。

4、结果数据结构说明

- 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。
- 以下是网页、影视、小说、游戏、论坛、通用五种结果类型共有的数据对应表

标题	resultData. title
当前结果链接 URL	resultData. linkUrl
摘要	resultData. abstract
图片路径	resultData. image
URL（供展现使用）	resultData. dispUrl
时间（供展现使用）	resultData. dispTime

- 以下是影视类型专有的数据对应表

主演	resultData. actorName
类型	resultData. genre
地区	resultData. contentLocation
剧情简介	resultData. description
主持人	resultData. host
电视台	resultData. provider

- 以下是小说类型专有的数据对应表

作者	resultData. author
类型	resultData. genre
字数	resultData. wordCount
周点击	resultData. weeklyClick
月点击	resultData. monthlyClick
总点击	resultData. totalClick
更新时间	resultData. dateModified
更新状态	resultData. updateStatus
费用状态	resultData. trialStatus

- 以下为游戏类型专有的数据对应表

子分类	resultData. gameSubCategory
修改时间	resultData. dateModified
下载次数	resultData. downloadCount
标签	resultData. tag
年龄范围	resultData. typicalAgeRange
分类	resultData. gameCategory
发行时间	resultData. datePublished
版本	resultData. version
内容语言	resultData. inLanguage
设备	resultData. device
类型	resultData. genre
下载地址	resultData. downloadUrl

- 以下为论坛类型专有的数据对应表

板块	resultData. forumName
作者	resultData. author
发表时间	resultData. createTime
最新回复时间	resultData. lastReplyTime
回复数	resultData. replyCount
浏览数	resultData. viewCount
帖子类型	resultData. threadClassify

- 以下为通用类型专有的数据对应表（以下字段并非一定要是左侧解释的内容，您可以为某个字段自定义含义）

内容标签，可以是内容主题、领域、分类等的简短说明文字	resultSummary.get("tag")
层级名称，或频道、版块名称	resultSummary.get("breadCrumb_title")
层级首页地址（对应上面的breadCrumb_title）	resultSummary.get("breadCrumb_url")
作者，帖子或博客作者，用户昵称	resultSummary.get("author_nickname")
作者个人首页地址	resultSummary.get("author_url")
回复数，论坛帖子回帖数，或博客评论数	resultSummary.get("replyCount")
内容属性，主要用于帖子，例如精华、热门、置顶等属性	resultSummary.get("property")
商品价格或打折优惠后的现价	resultSummary.get("price_new")
打折优惠前的价格	resultSummary.get("price_old")
折扣优惠的折扣值	resultSummary.get("discount_value")
折扣优惠介绍	resultSummary.get("discount_description")
商品、活动、影视著作等的评价：默认采用 5 分制，即最高分为 5 分、最低分为 1 分	resultSummary.get("review_rating")
评分人数	resultSummary.get("review_count")
商品、活动等所在地	resultSummary.get("location_address")
机构或活动组织者电话	resultSummary.get("location_tel")
地域范围，可以是地区、城市甚至国家	resultSummary.get("location_area")
事件、活动的开始时间	resultSummary.get("time_startDate")
事件、活动的结束时间	resultSummary.get("time_endDate")

- 调用 `getSearchInfo` 方法可获取搜索相关信息，您可以任意定义搜索相关信息的命名，如：`searchInfo`，以下是搜索相关信息的数据结构对应表

结果总条数	searchInfo.totalNum
搜索资源类型	searchInfo.tplId
当前页码（与搜索时传的页码不同，此处 0 表示第一页，1 表示第二页）	searchInfo.curPage
搜索所用时间	searchInfo.searchTime

- 调用 `getError` 可以获取错误信息，您可以任意定义错误信息的命名，如：`erroData`

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

- 调用 `getSug` 方法可获取搜索提示词，您可以任意定义搜索提示词的命名，如：`sugData`，`sugData` 是个数组，`sugData[0].title` 即为第一条提示词。

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(1);
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);
```

4、方法说明

本段将详细介绍搜索类 CseSearch 和查询条件类 QueryInfo 中的各方法。

- 以下是搜索类 CseSearch 中相关方法

方法名	方法说明	参数说明	备注
getResult(QueryInfo)	获取搜索结果	将查询条件类 QueryInfo 封装好各查询条件后，作为参数传给 getResult	此方法返回值为结果信息类 ReturnInfo。
getSug("a","normal")	获取搜索提示词。	第一个参数为查询词，第二个参数为 sug 类型（具体类型见 详细介绍一 ）	此方法返回一个数组，包含最多 10 条搜索提示词。
getHot()	获取热搜词		此方法返回一个数组，包含最多 10 条热搜词。

- 以下是查询条件类 QueryInfo 中组织结果相关方法

方法名	方法说明	参数说明	备注
setQuery(String)	设置搜索词		
setResultType(int)	设置搜索资源类型	参数 0、1、2、3、5、6 分别代表网页、影视、小说、游戏、论坛、通用	不调用此方法则会默认为网页类型。 注：如果选择 1、2、3、5、6 几种类型，均需到 zn.baidu.com 提交对应的数据
setPageNo(int)	指定返回第几页数据	如希望获取第二页，则此参数传 2	不调用此方法则会默认为第一页
setShowCount(int)	指定返回结果中每页条数。	参数有效值为 1 到 10。	不调用此方法则默认为 10。大于 10 的参数会自动置为 10

- 以下是查询条件类 QueryInfo 中组织筛选条件相关方法

方法名	方法说明	参数说明	备注
setTitleFilter(boolean)	设置是否按标题筛选	true 表示按标题搜索，false 表示按全文搜索。默认为 false	网页、通用、影视、论坛样式可使用此方法
setTimeFilter(char, int)	设置是否按时间筛选	第一个参数表示时间类型：‘h’表示小时，‘d’表示天，‘w’表示周，‘m’表示月。第二个参数是具体时间值。 setTimeFilter(‘d’, 1) 表示要查询最近一天的数据。	网页、通用、影视、论坛样式可使用此方法。 第二个参数如果为 0，则表示关闭按时间筛选。
setFieldFilter(HashMap<String, String>)	按结构化因子筛选（如游戏的“分类”，小说的“更新状态”等）	参数为 HashMap，插入多个键值对。键值对中 key 为结构化因子的类型，如：gameCategory；value 为结构化因子的值，如：单机。	小说、游戏、论坛、通用类可使用此方法（ 需提交结构化数据 ）。筛选所用到的结构化字段如 详细注释 2 所示。
openJc()	开启关键词纠错功能。此功能默认开启。	无	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	无	

openSq()	开启关键词同义转换功能。此功能默认关闭。	无	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。	无	

- 以下是查询条件类 QueryInfo 中组织排序条件相关方法

方法名	方法说明	参数说明	备注
setTimeOrder(boolean)	设置是否开启按时间排序	true 为开启。默认不开启。	开启后将按时间倒排。网页、通用、影视、论坛样式可使用此方法
setFieldOrder(String, int)	按结构化因子排序（如游戏类的“下载次数”，小说类的“总点击”等）	第一个参数为排序类型，第二个参数指定正排还是倒排（0 表示倒排，1 表示正排）。	小说、游戏、论坛、通用类可使用此方法（ 需提交结构化数据 ）。排序用到的结构化字段如 详细注释 3 所示。

- 以下是查询条件类 QueryInfo 中设置搜索范围相关方法

方法名	方法说明	参数说明	备注
setSearchRange(int, String[])	设置搜索范围	第一个参数表示搜索范围类型（1 表示子目录，2 表示二级域名，3 表示 inurl），第二个参数为具体搜索范围。	具体见 详细注释 4 。 必须调用此方法 ，设置搜索范围。

- 详细注释 1：搜索提示词（sug）类别

- 搜索提示词各类别对应关系如下

通用	normal
影视	movie
音乐	music
小说	novel
医疗健康	health
游戏	game
软件应用	soft

- 详细注释 2：按结构化因子筛选（如想按某字段筛选，必须提交相关数据）

- 游戏类筛选用到的结构化字段如下

分类	gameCategory
子分类	gameSubCategory
标签	tag
类型	genre
设备	device

以下代码表示按子分类是单机、类型是格斗游戏来进行筛选，其余类推。

```
HashMap<String,String> fieldMap = new HashMap<String,String> ();
fieldMap.put("gameCategory", "单机");
fieldMap.put("genre", "格斗游戏");
setFieldFilter(fieldMap);
```

- 小说类筛选用到的结构化字段如下

分类	genre
更新状态	updateStatus
阅读费用	trialStatus

方法调用方式与游戏相同

- 通用类筛选用到的结构化字段如下（以下字段并非固定为左侧解释的内容，您可以为某个字段自定义含义）

内容标签	tag
层级名称，或频道、版块名称	breadCrumb_title
层级首页地址（对应上面的breadCrumb_title）	breadCrumb_url
博客评论数等	replyCount
如精华、热门、置顶等属性	property
商品价格或打折优惠后的现价	price_new
打折优惠前的价格	price_old
折扣优惠的折扣值	discount_value
商品、活动、影视著作等的评价	review_rating
评分人数	review_count
地域范围，可以是地区、城市甚至国家	location_area

方法调用方式与游戏相同

论坛类筛选可使用此方法筛选出某个版块下的数据

示例（按“站长杂谈”版块筛选）：

```
HashMap<String,String> fieldMap = new HashMap<String,String> ();
fieldMap.put("cse_block1", "forumIn_forumName:站长杂谈");
setFieldFilter(fieldMap);
```

- 详细注释 3：按结构化因子排序
 - 游戏类排序用到的结构化字段如下

发行时间	datePublished
修改时间	dateModified
下载次数	downloadCount

setFieldOrder("datePublished", 0)表示按发行时间倒排。其余类推。

- 小说类排序用到的结构化字段如下

总点击	totalClick
更新时间	dateModified

方法调用方式与上述游戏类相同。

- 通用类排序用到的结构化字段如下（以下字段并非固定为左侧解释的内容，您可以为某个字段自定义含义）

博客评论数等	replyCount
商品价格或打折优惠后的现价	price_new
打折优惠前的价格	price_old
折扣优惠的折扣值	discount_value
商品、活动、影视著作等的评价	review_rating
评分人数	review_count
开始时间	time_startDate
结束时间	time_endDate

方法调用方式与上述游戏类相同。

- 论坛类排序用到的结构化字段如下

发表时间	cse_createTime
回复数	cse_category2
浏览数	cse_category3

方法调用方式与上述游戏类相同。

- 详细注释 4：设置搜索范围 setSearchRange()

- 指定子目录方法示例：

```
String[] subDomains = {"tieba.baidu.com"};
```

```
setSearchRange(1, subDomains);
```

第一个参数为 1 代表类型为子目录，第二个参数是数组，里面传的值是指定的子目录名。（子目录只支持一个）

如 baidu.com/more，即为子目录（不支持小说/游戏/论坛）。

- 指定二级域名示例：

```
String[] subDomains = {"tieba.baidu.com", "baike.baidu.com"};
```

```
setSearchRange(2, subDomains);
```

第一个参数为 2 代表类型为二级域名，第二个参数可传多个二级域数值。

如 news.baidu.com，即为二级域名。（二级域名支持多个）

- 指定 inurl：

```
String[] subDomains = {" movie "};
```

```
setSearchRange(3, subDomains);
```

第一个参数为 3 代表类型为 inurl，第二个参数为指定的 inurl 值。（Inurl 是

指您希望搜索到的网页 url 都包含的一个字符串)

5、结果数据结构说明

- 获取到的数据结果封装在结果信息类 `ReturnInfo` 类中，如：`ReturnInfo returnInfo = cseSearch.getResult(queryInfo);`。

结果信息类 `ReturnInfo` 中有三个变量：`int erroNo`；`SearchInfo serachInfo`；`ArrayList<ResultInfo> resultInfoList`。

`erroNo` 用于判断结果是否正确返回，为 0 时表示正确。`serachInfo` 中封装本次查询信息。`resultInfoList` 即结果列表。

- 以第一条数据为例 `ResultInfo resultInfo = returnInfo.getResultInfoList().get(0)`，从 `resultData` 中可以获取标题、摘要等具体值。
- 以下是网页、影视、小说、游戏、论坛、通用五种结果类型共有的数据对应表（标题和摘要中需要飘红的字段用 `xxx` 标签进行了标注，您需要自行处理）

标题	<code>resultData. title</code>
当前结果链接 URL	<code>resultData. linkUrl</code>
摘要	<code>resultData. abstract</code>
图片路径	<code>resultData. image</code>
URL（供展现使用）	<code>resultData. dispUrl</code>
时间（供展现使用）	<code>resultData. dispTime</code>

- 影视、小说、游戏、论坛均有结构化字段，封装在结果类 `ResultInfo` 的 `summaryWords` 字段中，此字段类型为 `HashMap<String, String>`。

仍以第一条结果数据为例：`HashMap<String, String> resultSummary = resultInfoList [0].getSummaryWords`。

- 以下是影视类型专有的数据对应表

主演	<code>resultSummary.get("actorName")</code>
类型	<code>resultSummary.get("genre")</code>
地区	<code>resultSummary.get("contentLocation")</code>
剧情简介	<code>resultSummary.get("description")</code>
主持人	<code>resultSummary.get("host")</code>
电视台	<code>resultSummary.get("provider")</code>

- 以下是小说类型专有的数据对应表

作者	<code>resultSummary.get("author")</code>
类型	<code>resultSummary.get("genre")</code>
字数	<code>resultSummary.get("wordCount")</code>
周点击	<code>resultSummary.get("weeklyClick")</code>
月点击	<code>resultSummary.get("monthlyClick")</code>
总点击	<code>resultSummary.get("totalClick")</code>
更新时间	<code>resultSummary.get("dateModified")</code>
更新状态	<code>resultSummary.get("updateStatus")</code>
费用状态	<code>resultSummary.get("trialStatus")</code>

- 以下为游戏类型专有的数据对应表

子分类	resultSummary.get("gameSubCategory")
修改时间	resultSummary.get("dateModified")
下载次数	resultSummary.get("downloadCount")
标签	resultSummary.get("tag")
年龄范围	resultSummary.get("typicalAgeRange")
分类	resultSummary.get("gameCategory")
发行时间	resultSummary.get("datePublished")
版本	resultSummary.get("version")
内容语言	resultSummary.get("inLanguage")
设备	resultSummary.get("device")
类型	resultSummary.get("genre")
下载地址	resultSummary.get("downloadUrl")

- 以下为论坛类型专有的数据对应表

板块	resultSummary.get("forumName")
作者	resultSummary.get("author")
发表时间	resultSummary.get("createdTime")
最新回复时间	resultSummary.get("lastReplyTime")
回复数	resultSummary.get("viewCount")
浏览数	resultSummary.get("viewCount")
帖子类型	resultSummary.get("threadClassify")

- 以下为通用类型专有的数据对应表（以下字段并非固定为左侧解释的内容，您可以为某个字段自定义含义）

内容标签，可以是内容主题、领域、分类等的简短说明文字	resultSummary.get("tag")
层级名称，或频道、版块名称	resultSummary.get("breadCrumb_title")
层级首页地址（对应上面的breadCrumb_title）	resultSummary.get("breadCrumb_url")
作者，帖子或博客作者，用户昵称	resultSummary.get("author_nickname")
作者个人首页地址	resultSummary.get("author_url")
回复数，论坛帖子回帖数，或博客评论数	resultSummary.get("replyCount")
内容属性，主要用于帖子，例如精华、热门、置顶等属性	resultSummary.get("property")
商品价格或打折优惠后的现价	resultSummary.get("price_new")
打折优惠前的价格	resultSummary.get("price_old")
折扣优惠的折扣值	resultSummary.get("discount_value")
折扣优惠介绍	resultSummary.get("discount_description")

商品、活动、影视著作等的评价：默认采用 5 分制，即最高分为 5 分、最低分为 1 分	resultSummary.get("review_rating")
评分人数	resultSummary.get("review_count")
商品、活动等所在地	resultSummary.get("location_address")
机构或活动组织者电话	resultSummary.get("location_tel")
地域范围，可以是地区、城市甚至国家	resultSummary.get("location_area")
事件、活动的开始时间	resultSummary.get("time_startDate")
事件、活动的结束时间	resultSummary.get("time_endDate")

- serachInfo 为本次查询的信息

结果总条数	searchInfo.getTotalNum()
搜索资源类型	searchInfo.getTplId()
当前页码（与搜索时传的页码不同，此处 0 表示第一页，1 表示第二页）	searchInfo.getCurPage()
搜索所用时间	searchInfo.getSearchTime()

6、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。

 dexmaker-1.1.jar
 ZnYuyin.jar

[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-iphoneros: 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"
                                cseRT: CseSearchResultTypeNormal];

//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),@"ct":@"2",@"cc":@"baike.baidu.com&tieba.baidu.com"}
success:^(id responseObject) {
    CseSearchResponseObject *obj = (CseSearchResponseObject*)responseObject;
    CseSearchModel *model = obj.blockData.datas[0];
    CseSearchStatusModel *statusModel = obj.errorStatus;
    CseSearchInfoModel *infoModel = obj.searchInfo;
```

```

} failure:^(NSError *error) {
    //handle error message
};

//获取搜索提示词方法示例如下
[[CseSearchManager defaultManager]
searchSuggestionWithText:@"搜索词"
//searchCategory 表示提示词的类型，对应关系见详细注释 2
searchCategory:@"normal"
success:^(id responseObject) {
    CseSearchSugModel *obj = (CseSearchSugModel*)responseObject;
    //CseSearchSugModel.sugs 即为搜索提示词，是个数组。
} failure:^(NSError *error) {
    //handle error message
};

```

4、参数说明

本段将详细介绍搜索类 CseSearchManager 中执行搜索的各参数。

- 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
cseRT	可设置的值为 0、1、2、3、5、6。其中 0 表示网页类型，1 表示影视类型，2 表示小说类型，3 表示游戏类型，5 表示论坛类型，6 表示通用类型。	除了网页类型，其它几种类型均需提交数据才能搜索到结果。

- 以下是搜索时可配置的参数

参数	参数说明	备注
ct	设置搜索范围的类型，1 表示子目录，2 表示二级域名，3 表示 inurl	此参数和下面的 cc 参数为必传参数
cc	具体的搜索范围。示例：ct 为 1 时 1，cc 可传 www.baidu.com/baobaact; ct 为 2 时，cc 可传 "baike.baidu.com&tieba.baidu.com"，可传 1 个，也可传多个，传多个是用&分隔，建议在 3 个以内；ct 为 3 时，cc 可传 "movie"，即传一个希望 url 中包含的字符串即可。	此参数和上面的 ct 参数为必传
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
stp	此参数值为 1 时表示只按标题搜索，不设置此参数表示按全文搜索。	网页、通用、影视、论坛样式支持此参数
sti	表示按某个时间段搜索，将时间段折算成分钟。如按最近一小时搜索，则 sti=60。	网页、通用、影视、论坛样式支持此参数
flt	表示按某个字段筛选	小说、游戏、通用类可使用此参数。筛选所用到的结构化字段如 详细注释 3 所示。
srt	表示按某个字段排序	小说、游戏、论坛、通用类可使用此参数按某字段排序。排序用到的结构化字段如 详细注释 4 所示。如果为网页、影视、论坛可以设置 srt=lds，表示按时间倒排

• 详细注释 1：搜索资源类型

➤ 搜索提示词各类别对应关系如下

网页	CseSearchResultTypeNormal
通用	CseSearchResultTypeNewResource
影视	CseSearchResultTypeMovie
小说	CseSearchResultTypeNovel
游戏	CseSearchResultTypeGame
论坛	CseSearchResultTypeForum

• 详细注释 2：搜索提示词（sug）类别

➤ 搜索提示词各类别对应关系如下

通用	normal
影视	movie
音乐	music
小说	novel
医疗健康	health
游戏	game
软件应用	soft

• 详细注释 3：按结构化因子筛选（如想按某字段筛选，必须提交相关数据）

➤ 游戏类筛选用到的结构化字段如下

分类	gameCategory
子分类	gameSubCategory

标签	tag
类型	genre
设备	device

以下代码表示按子分类是单机、类型是格斗游戏来进行筛选，其余类推。

```
[[CseSearchManager defaultManager]
searchWithText:@"搜索词"
parameters:@{@"pn":@(10),@"p":@(0),@"flt":@"gameCategory= 单机 &
genre=格斗游戏"}]
```

- 小说类筛选用到的结构化字段如下

分类	genre
更新状态	updateStatus
阅读费用	trialStatus

方法调用方式与游戏相同

- 通用类筛选用到的结构化字段如下（以下字段并非固定为左侧解释的内容，您可以为某个字段自定义含义）

内容标签	tag
层级名称，或频道、版块名称	breadCrumb_title
层级首页地址（对应上面的breadCrumb_title）	breadCrumb_url
博客评论数等	replyCount
如精华、热门、置顶等属性	property
商品价格或打折优惠后的现价	price_new
打折优惠前的价格	price_old
折扣优惠的折扣值	discount_value
商品、活动、影视著作等的评价	review_rating
评分人数	review_count
地域范围，可以是地区、城市甚至国家	location_area

方法调用方式与游戏相同

- 详细注释 4：按结构化因子排序

- 游戏类排序用到的结构化字段如下

发行时间	datePublished
修改时间	dateModified
下载次数	downloadCount

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

- 小说类排序用到的结构化字段如下

总点击	totalClick
-----	------------

更新时间	dateModified
------	--------------

方法调用方式与上述游戏类相同。

- 通用类排序用到的结构化字段如下（以下字段并非固定为左侧解释的内容，您可以为某个字段自定义含义）

博客评论数等	replyCount
商品价格或打折优惠后的现价	price_new
打折优惠前的价格	price_old
折扣优惠的折扣值	discount_value
商品、活动、影视著作等的评价	review_rating
评分人数	review_count
开始时间	time_startDate
结束时间	time_endDate

方法调用方式与上述游戏类相同。

- 论坛类排序用到的结构化字段如下

发表时间	cse_createTime
回复数	cse_category2
浏览数	cse_category3

方法调用方式与上述游戏类相同。

5、结果数据结构说明

- 获取到的数据结果封装在 CseSearchResponseObject 类中，如：CseSearchResponseObject *obj = (CseSearchResponseObject*)responseObject;。
- 以第一条数据为例 CseSearchModel *resultData = obj.blockData.datas[0];，从 resultData 中可以获取标题、摘要等具体值。
- 以下是网页、影视、小说、游戏、论坛、通用五种结果类型共有的数据对应表（标题和摘要中需要飘红的字段用xxx标签进行了标注，您需要自行处理）

标题	resultData. title
当前结果链接 URL	resultData. linkUrl
摘要	resultData. abstract
图片路径	resultData. image
URL（供展现使用）	resultData. dispUrl
时间（供展现使用）	resultData. timeshow

- 影视、小说、游戏、论坛均有结构化字段，封装在 resultData 的 summarywords 字段中，如 NSDictionary *resultSummary = resultData.summarywords;
- 以下是影视类型专有的数据对应表

主演	resultSummary[@ "actorName"]
类型	resultSummary[@ "genre"]
地区	resultSummary[@ "contentLocation"]
剧情简介	resultSummary[@ "description"]

主持人	resultSummary["@host"]
电视台	resultSummary["@provider"]

- 以下是小说类型专有的数据对应表

作者	resultSummary["@author"]
类型	resultSummary["@genre"]
字数	resultSummary["@wordCount"]
周点击	resultSummary["@weeklyClick"]
月点击	resultSummary["@monthlyClick"]
总点击	resultSummary["@totalClick"]
更新时间	resultSummary["@dateModified"]
更新状态	resultSummary["@updateStatus"]
费用状态	resultSummary["@trialStatus"]

- 以下为游戏类型专有的数据对应表

子分类	resultSummary["@gameSubCategory"]
修改时间	resultSummary["@dateModified"]
下载次数	resultSummary["@downloadCount"]
标签	resultSummary["@tag"]
年龄范围	resultSummary["@typicalAgeRange"]
分类	resultSummary["@gameCategory"]
发行时间	resultSummary["@datePublished"]
版本	resultSummary["@version"]
内容语言	resultSummary["@inLanguage"]
设备	resultSummary["@device"]
类型	resultSummary["@genre"]
下载地址	resultSummary["@downloadUrl"]

- 以下为论坛类型专有的数据对应表

板块	resultSummary["@forumName"]
作者	resultSummary["@author"]
发表时间	resultSummary["@createdTime"]
最新回复时间	resultSummary["@lastReplyTime"]
回复数	resultSummary["@viewCount"]
浏览数	resultSummary["@viewCount"]
帖子类型	resultSummary["@threadClassify"]

- 以下为通用类型专有的数据对应表（以下字段并非固定为左侧解释的内容，您可以为某个字段自定义含义）

内容标签，可以是内容主题、领域、分类等的简短说明文字	resultSummary["@tag"]
层级名称，或频道、版块名称	resultSummary["@breadCrumb_title"]
层级首页地址（对应上面的	resultSummary["@breadCrumb_url"]

breadCrumb_title)	
作者, 帖子或博客作者, 用户昵称	resultSummary["@author_nickname]
作者个人首页地址	resultSummary["@author_url]
回复数, 论坛帖子回帖数, 或博客评论数	resultSummary["@replyCount]
内容属性, 主要用于帖子, 例如精华、热门、置顶等属性	resultSummary["@property]
商品价格或打折优惠后的现价	resultSummary["@price_new]
打折优惠前的价格	resultSummary["@price_old]
折扣优惠的折扣值	resultSummary["@discount_value]
折扣优惠介绍	resultSummary["@discount_description]
商品、活动、影视著作等的评价: 默认采用 5 分制, 即最高分为 5 分、最低分为 1 分	resultSummary["@review_rating]
评分人数	resultSummary["@review_count]
商品、活动等所在地	resultSummary["@location_address]
机构或活动组织者电话	resultSummary["@location_tel]
地域范围, 可以是地区、城市甚至国家	resultSummary["@location_area]
事件、活动的开始时间	resultSummary["@time_startDate]
事件、活动的结束时间	resultSummary["@time_endDate]

- CseSearchStatusModel 为查询状态信息, 如 CseSearchStatusModel *statusModel = obj.errorStatus;

状态码	statusModel. erroNo
状态值	statusModel. erroStr

- CseSearchInfoModel 为本次查询的信息

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

联系方式

邮件: zn@baidu.com