

# 联网搜索API

## 快速入门



腾讯云

## 【 版权声明 】

©2013–2026 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

## 【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

## 【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

## 【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

# 文档目录

## 快速入门

快速入门概述

腾讯云标准方式

服务 API KEY 方式

# 快速入门

## 快速入门概述

最近更新时间：2026-04-17 11:29:11

为满足不同开发场景与安全要求，腾讯云联网搜索API 提供两种快速接入方式：

- 方式一：[腾讯云标准方式](#)（AK/SK 签名认证）
- 方式二：[服务 API KEY 方式](#)（单 API KEY 认证）

客户可根据自身技术栈、安全策略及迁移成本，灵活选择合适的接入方式。

### 两种接入方式简介

差异项	方式一： <a href="#">腾讯云标准方式</a>	方式二： <a href="#">服务 API KEY 方式</a>
认证方式	通过腾讯云密钥 SecretId/SecretKey 生成签名认证	通过 HTTP Header 携带 Authorization: Bearer YOUR_API_KEY（服务密钥）认证
鉴权体系	腾讯云访问管理 CAM 鉴权体系	联网搜索API 服务自行鉴权
服务域名	wsa.tencentcloudapi.com	api.wsa.cloud.tencent.com
开发门槛	<ul style="list-style-type: none"> <li>• 自主开发门槛相对较高（需实现签名算法）</li> <li>• 支持通过腾讯云 API 3.0 Explorer 在线调试、验证签名、在线生成 SDK 代码示例（支持多种语言，包括 Python、Java、PHP、Go、Node.js、.Net 等）</li> </ul>	<ul style="list-style-type: none"> <li>• 自主开发门槛相对较低（可通过 HTTP 客户端直接调用）</li> <li>• 需参照文档示例自行开发</li> </ul>
使用场景	对安全性/审计要求较高的企业级后端服务生产环境	兼容主流大模型服务商、搜索服务商的 API KEY 认证鉴权机制，适合大模型、Agent 开发项目统一开发集成、复用和迁移。

# 腾讯云标准方式

最近更新时间：2026-04-17 11:29:11

初次使用联网搜索服务时，如果您通过腾讯云标准方式接入服务，可参考本文的指引进行准备工作及入门操作。

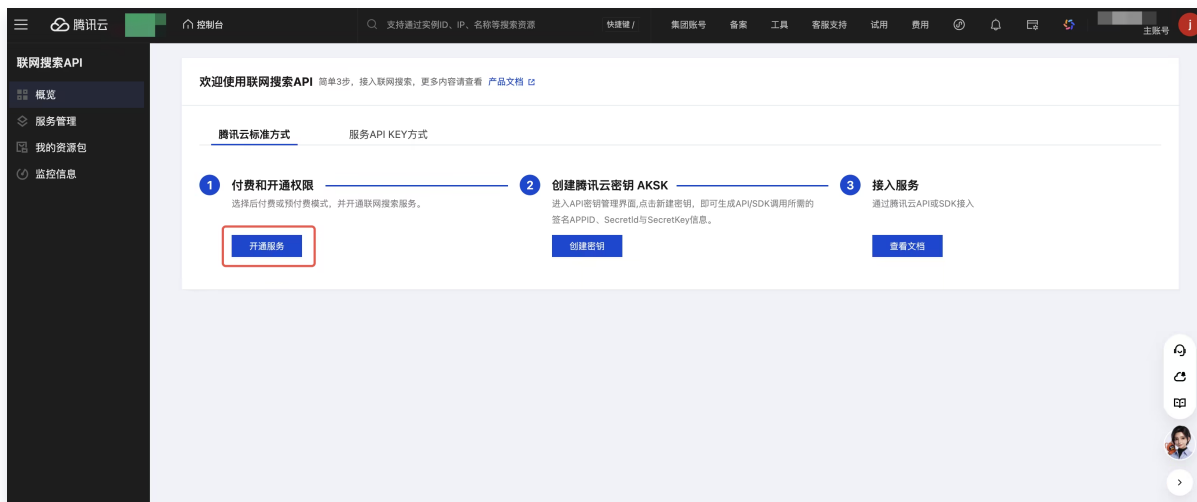
操作步骤	说明
<a href="#">步骤一：登录注册</a>	注册腾讯云账号，完成实名认证并登录
<a href="#">步骤二：开通服务</a>	控制台自助开通联网搜索API 服务
<a href="#">步骤三：获取腾讯云密钥</a>	获取腾讯云密钥 SecretId 和 SecretKey
<a href="#">步骤四：接入 API</a>	通过腾讯云 API 或 SDK 接入

## 步骤一：登录注册

注册并通过个人实名认证或企业认证后，登录 [腾讯云](#)。如果没有账号，请参考 [注册腾讯云](#)。

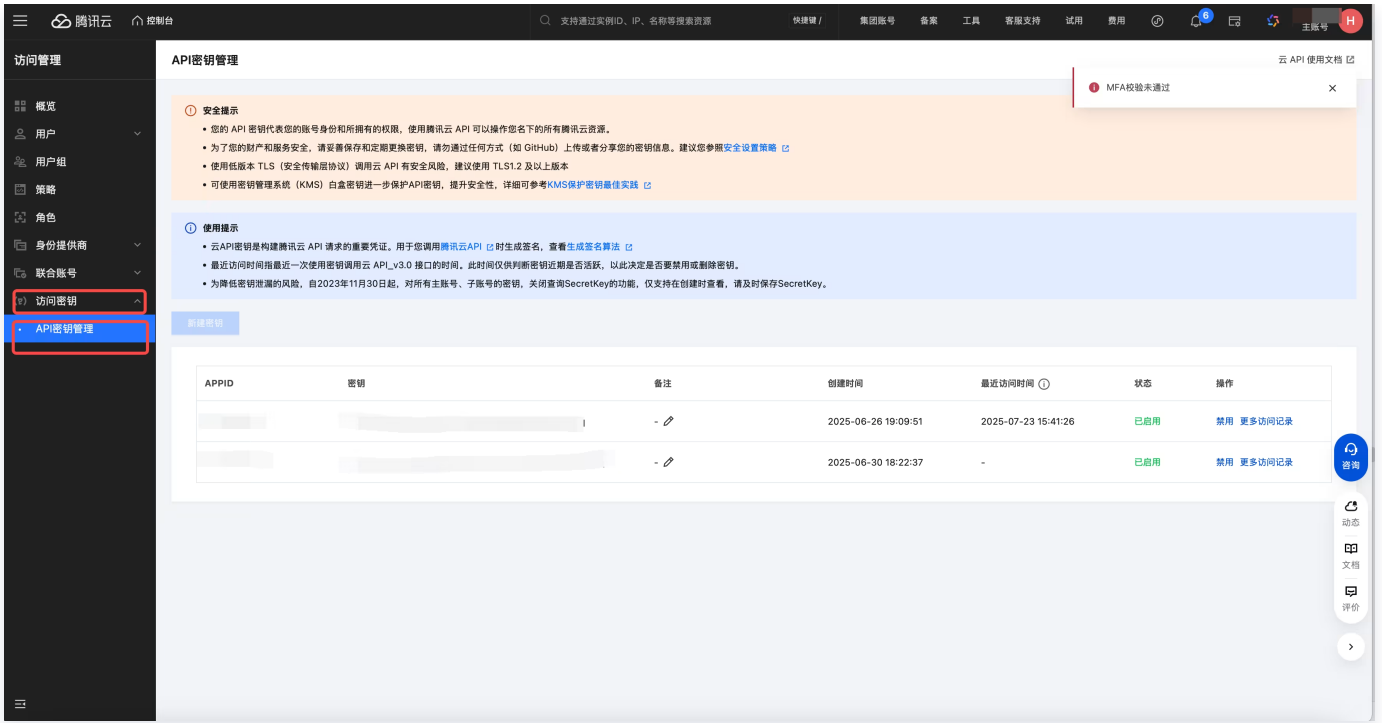
## 步骤二：开通服务

进入 [联网搜索API 控制台](#) 概览页面，单击[开通服务](#)，按照 [购买方式](#) 文档指引按需完成服务开通和购买。



## 步骤三：获取云 API 密钥

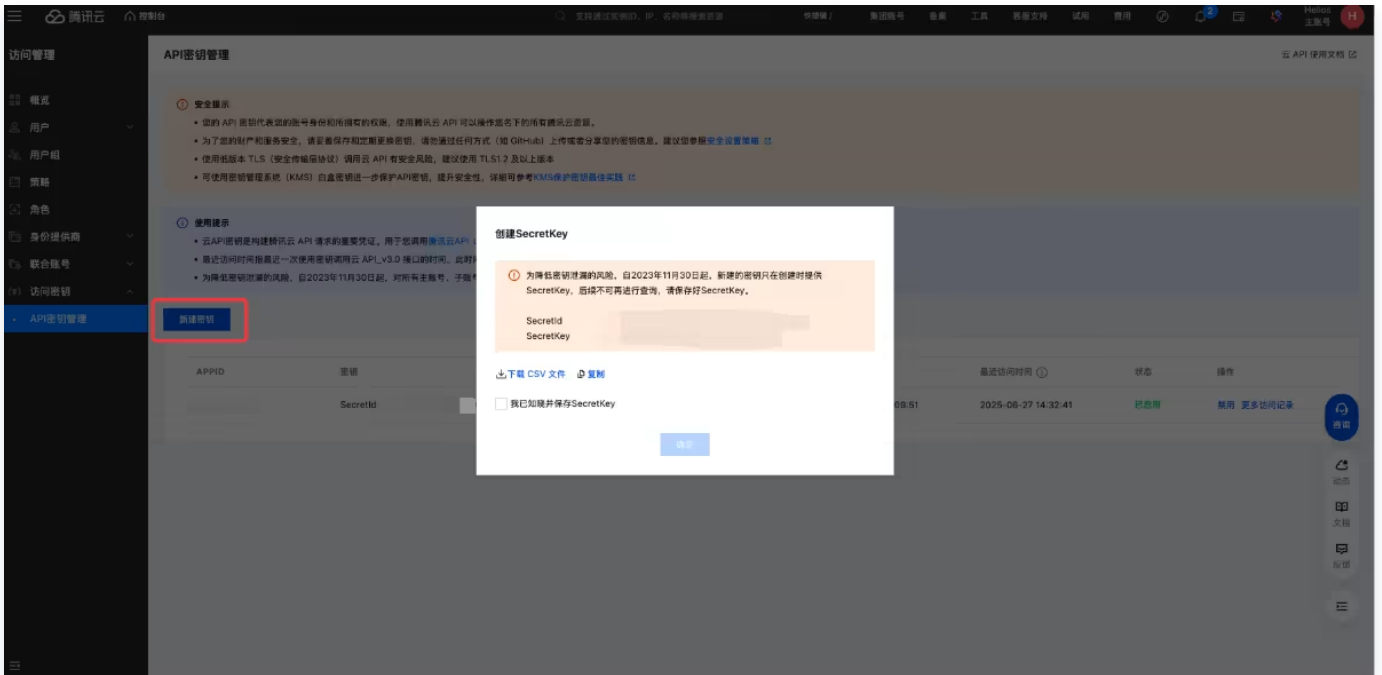
1. 前往腾讯云官网控制台 [API 密钥管理](#) 界面获取密钥。



2. 单击**新建密钥**，弹窗获取 SecretId 和 SecretKey，可单击**下载 CSV 文件**保存至本地。

**注意：**

为降低密钥泄露的风险，自2023年11月30日起，新建的密钥只在创建时提供 SecretKey，后续不可再进行查询，请保存好 SecretKey。



**步骤四：接入 API**

请根据接入指南，参考 [联网搜索接口](#) 示例调用对应的接口，即可开始使用。

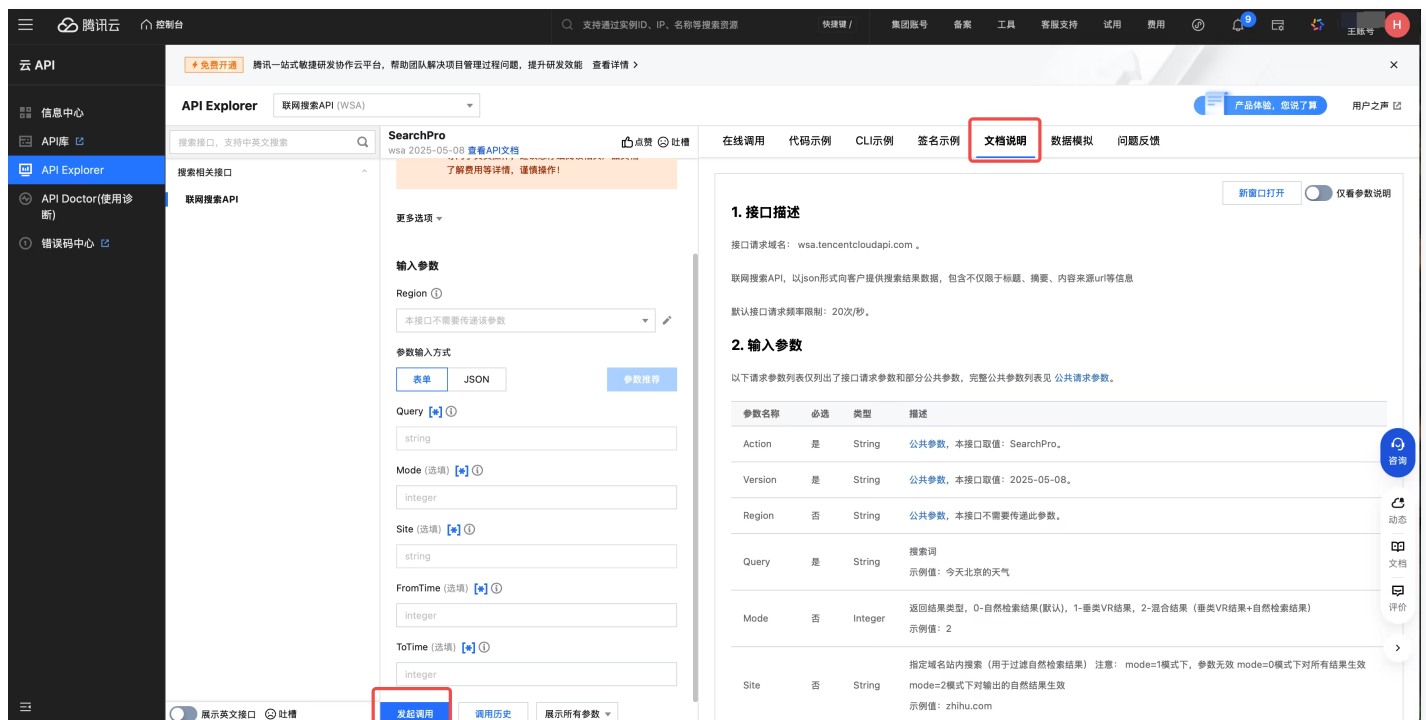
## 通过 API 3.0 Explorer 在线调用

**适用对象：**开发初学者，有代码编写基础人员。

**说明：**此方式能够实现在线调用、签名验证、SDK 代码生成和快速检索接口等能力。

在 [API 概览](#) 选择需要调用的接口，选择 [点击调试](#) 进入调试页面。并填写输入参数。输入参数在 API 3.0 Explorer 界面的“文档说明”选项卡中可以查看对应接口输入参数的具体含义。

**说明：**  
平台将对登录用户提供临时 Access Key，以便进行调试。

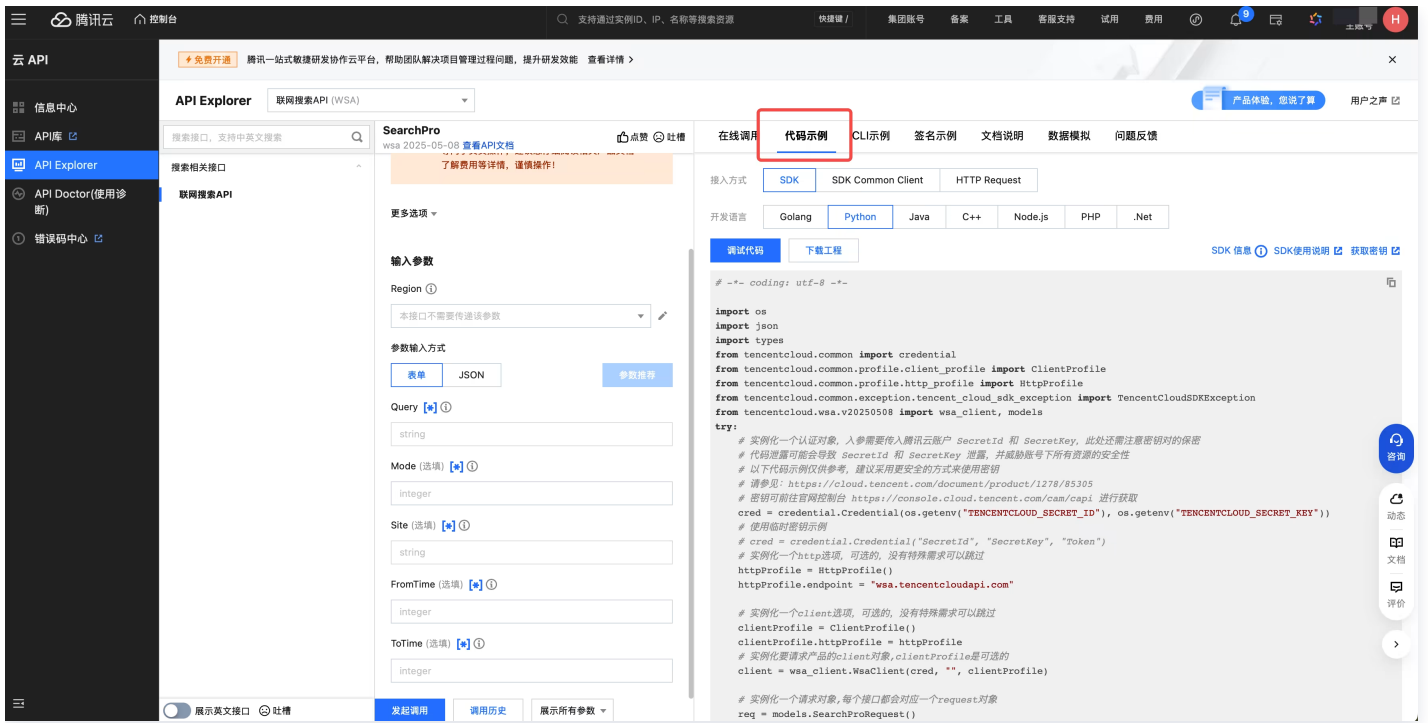


## 通过编写代码，调用 API 进行开发

**适用对象：**开发工程师，熟悉代码编写人员。

**说明：**腾讯云已编写好的开发工具集（SDK），支持通过调用同传服务 API 开发功能。目前 SDK 已支持多种语言，包括 Python、Java、PHP、Go、Node.js、.Net 等，可在每个服务的文档中下载对应的 SDK。

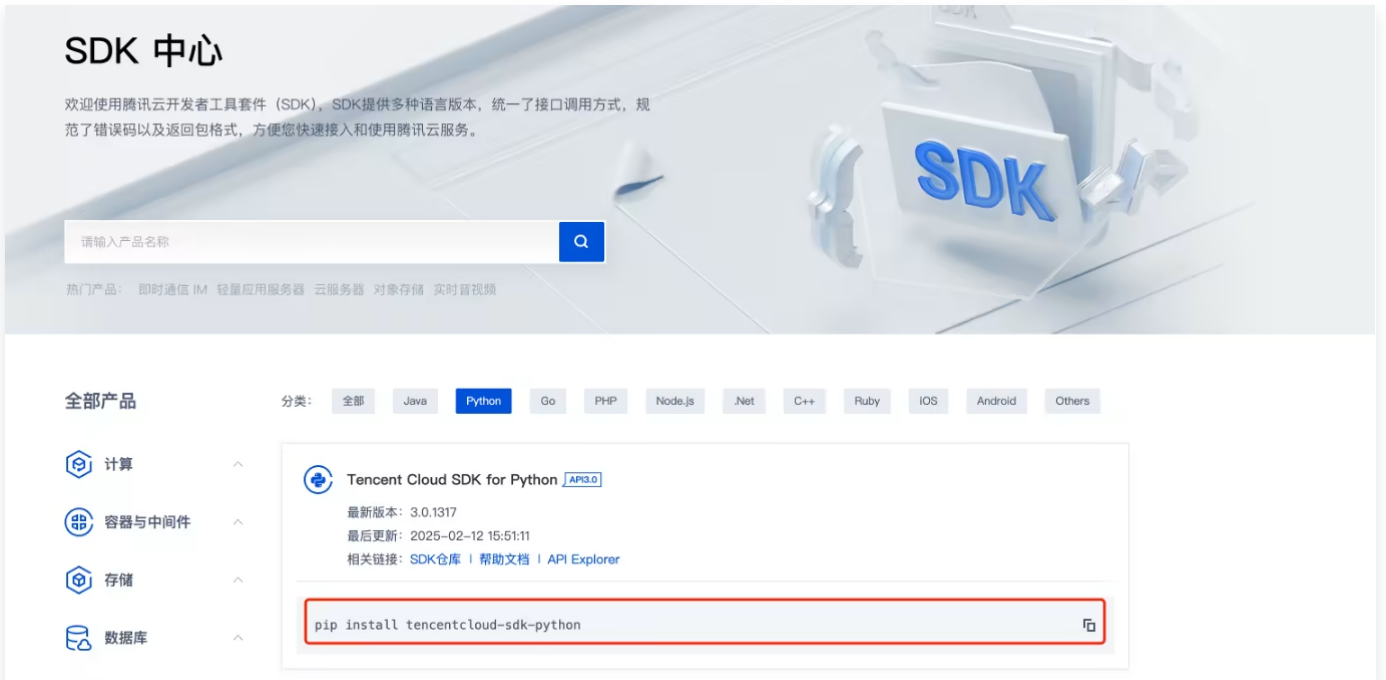
进入 [API Explorer](#) 页面，选择需要调用的接口，在右侧选项卡中选择“[代码示例](#)”，选择接入方式和开发语言，并填写参数，即可得到示例代码。



下面以 Python 代码为例，说明 API 本地调用的方法。

1. 安装 腾讯云 SDK 对应版本代码。

```
pip install tencentcloud-sdk-python
```



2. 按如下示例代码调用接口：

仅需修改如下示例代码中：**TENCENTCLOUD\_SECRET\_ID**、**TENCENTCLOUD\_SECRET\_KEY** 密钥信息，以及对应的接口入参。

```
# -*- coding: utf-8 -*-

import os
import json
import types
from tencentcloud.common import credential
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
from tencentcloud.common.exception.tencent_cloud_sdk_exception import
TencentCloudSDKException
from tencentcloud.wsa.v20250508 import wsa_client, models

try:
    # 实例化一个认证对象，入参需要传入腾讯云账户 SecretId 和 SecretKey，此处还
    # 需注意密钥对的保密
    # 代码泄露可能会导致 SecretId 和 SecretKey 泄露，并威胁账号下所有资源的安
    # 全性
    # 以下代码示例仅供参考，建议采用更安全的方式来使用密钥
    # 请参见：https://cloud.tencent.com/document/product/1278/85305
    # 密钥可前往官网控制台 https://console.cloud.tencent.com/cam/capi 进行
    # 获取
    cred =
    credential.Credential("TENCENTCLOUD_SECRET_ID", "TENCENTCLOUD_SECRET_KE
    Y")
    # 使用临时密钥示例
    # cred = credential.Credential("SecretId", "SecretKey", "Token")
    # 实例化一个http选项，可选的，没有特殊需求可以跳过
    httpProfile = HttpProfile()
    httpProfile.endpoint = "wsa.tencentcloudapi.com"

    # 实例化一个client选项，可选的，没有特殊需求可以跳过
    clientProfile = ClientProfile()
    clientProfile.httpProfile = httpProfile
    # 实例化要请求产品的client对象，clientProfile是可选的
    client = wsa_client.WsaClient(cred, "", clientProfile)

    # 实例化一个请求对象，每个接口都会对应一个request对象
    req = models.SearchProRequest()
    params = {
        "Query": "北京天气"
```

```
}
req.from_json_string(json.dumps(params))

# 返回的resp是一个SearchProResponse的实例，与请求对象对应
resp = client.SearchPro(req)
# 输出json格式的字符串回包
print(resp.to_json_string())

except TencentCloudSDKException as err:
    print(err)
```

# 服务 API KEY 方式

最近更新时间：2026-04-17 11:29:11

初次使用联网搜索服务时，如果您通过服务 API KEY 方式接入服务，可参考本文的指引进行准备工作及入门操作。

操作步骤	说明
<a href="#">步骤一：登录注册</a>	注册腾讯云账号，完成实名认证并登录
<a href="#">步骤二：开通服务</a>	控制台自助开通联网搜索API 服务
<a href="#">步骤三：获取服务密钥 API KEY</a>	获取联网搜索API 的服务密钥 API KEY
<a href="#">步骤四：接入 API</a>	按照 API KEY 兼容模式，通过 API 接入

## 步骤一：登录注册

注册并通过个人实名认证或企业认证后，登录 [腾讯云](#)。如果没有账号，请参考 [注册腾讯云](#)。

## 步骤二：开通服务

进入 [联网搜索API 控制台](#) 概览页面，单击[开通服务](#)，按照 [购买方式](#) 文档指引按需完成服务开通和购买。

腾讯云控制台

支持通过实例ID、IP、名称等搜索资源

快捷键 / 集团账号 备案 工具 客服支持 试用 费用

主账号

### 联网搜索API

- 概览
- 服务管理
- 我的资源包
- 监控信息

腾讯云标准方式 **服务API KEY方式**

- 1 付费和开通权限**  
选择后付费或预付费模式，并开通联网搜索服务。  
[开通服务](#)
- 2 创建服务密钥 API KEY**  
点击下方创建按钮，即可生成联网搜索API服务密钥 API KEY。  
[创建 API KEY](#)
- 3 接入服务**  
按照API KEY兼容模式，通过API接入。  
[查看文档](#)

#### API KEY列表

① 服务密钥API KEY是您请求腾讯云联网搜索API时的安全鉴权凭证。为了您的财产和服务安全，请妥善保管API KEY，请勿通过任何方式上传或公开分享您的密钥信息。每个subuin（包括主账号）最多同时持有3个服务密钥API KEY，请合理创建并分配使用。

API KEY	名称	创建者uin	创建者subuin	创建时间	最近更新时间	操作
暂无数据						

共 0 条

## 步骤三：创建服务密钥 API KEY

🔔 说明：

1. 每个 subuin（包括主账号）最多同时持有**3个**服务密钥 API KEY，请合理创建并分配使用。
2. 使用子账号创建和管理服务密钥 API KEY 时，须主账号事先前往 [访问管理 控制台](#) 为该子用户授予**联网搜索API（WSA）全读写访问权限**。

1. 进入 [联网搜索API 控制台](#)，在**概览 > 服务 API KEY 方式**界面，单击**创建 API KEY**。



2. 输入服务密钥名称，单击**确定**。



3. 弹窗返回您创建的 API KEY，可单击**下载 CSV 文件**保存至本地。或单击**复制**后另行保存。

#### ⚠ 注意：

- 为降低服务密钥泄漏的风险，仅在创建时提供明文 API KEY，后续不可再进行查询，请保存好您的 API KEY。
- 服务密钥 API KEY 是您请求腾讯云联网搜索API 时的安全鉴权凭证。为了您的财产和服务安全，请妥善保存 API KEY，请勿通过任何方式上传或者公开分享您的密钥信息。



## 步骤四：接入 API

请根据下方接入指引，结合实际业务场景完成接入。

### 接口描述

接口名称：SearchPro

服务域名：api.wsa.cloud.tencent.com

请求协议：HTTPS

请求方式：POST

数据格式：Content-Type: application/json; charset=UTF-8

### 输入参数

参数名称	必选	类型	位置	描述
Authorization	是	String	Header	身份验证令牌 示例值：Bearer xxxx
Query	是	String	Body(JSON)	搜索词 示例值：今天北京的天气
Mode	否	Integer	Body(JSON)	返回结果类型，0-自然检索结果(默认)，1-多模态VR结果，2-混合结果(多模态VR结果+自然检索结果) 示例值：2
Site	否	String	Body(JSON)	指定域名站内搜索(用于过滤自然检索结果) 注意：mode=1模式下，参数无效；mode=0模式下，对所有结果生效；mode=2模式下，对输出的自然结果生效 示例值：zhihu.com
FromTime	否	Integer	Body(JSON)	起始时间(用于过滤自然检索结果)，精确到秒时间戳格式

				<p>注意：mode=1模式下，参数无效；mode=0模式下，对所有结果生效；mode=2模式下，对输出的自然结果生效</p> <p>示例值：1745498501</p>
ToTime	否	Integer	Body(JSON)	<p>结束时间（用于过滤自然检索结果），精确到秒时间戳格式</p> <p>注意：mode=1模式下，参数无效；mode=0模式下，对所有结果生效；mode=2模式下，对输出的自然结果生效</p> <p>示例值：1745498501</p>
Cnt	否	Integer	Body(JSON)	<p>cnt=10/20/30/40/50，最多可支持返回50条搜索结果，仅限尊享版使用</p> <p>示例值：10</p>
Industry	否	String	Body(JSON)	<p>Industry=gov/news/acad/finance，对应党政机关、权威媒体、学术（英文）、金融，仅限尊享版使用</p> <p>示例值：news</p>

## 输出参数

输出媒体类型：application/json

参数名称	类型	描述
Query	String	<p>原始查询语</p> <p>示例值：今天北京的天气</p>
Pages	Array of String	<p>搜索结果页面详情，格式为 JSON 字符串。</p> <p>title: 结果标题</p> <p>date: 内容发布时间</p> <p>url: 内容发布源 URL</p> <p>passage: 标准摘要</p> <p>content: 动态摘要（尊享版字段）</p> <p>site: 网站名称，部分不知名站点结果可能为空</p> <p>score: 相关性得分，取值0~1，越靠近1表示越相关</p> <p>images: 图片列表</p> <p>favicon: 网站图标链接，部分不知名站点结果可能为空</p> <p>示例值：["{"passage":"aaa"}", {"passage":"bbb"}"]</p>
Version	String	<p>用户版本：standard/premium/lite</p> <p>示例值：standard</p>
Msg	String	<p>提示信息</p> <p>示例值：hit black query</p>

RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。
-----------	--------	--

## 示例

### 输入示例

```
POST / HTTP/1.1
Host: api.wsa.cloud.tencent.com
Content-Type: application/json
Authorization: Bearer sk-209b****e23d

{
  "Query": "三星堆的由来"
}
```

### 输出示例

```
{
  "Response": {
    "Pages": [
      {
        "passage": "三星堆名字的由来 三星堆遗址位于四川省广汉市西北部，距离成都约40公里，地理位置独特，被三条河流环绕，形成了天然的地理屏障。据《华阳国志》记载，古蜀国“有三江，皆发源于岷山，流至成都平原”。这三条河流，即今之鸭子河、石亭江和江，在三星堆附近交汇，形成了一片肥沃的土地。三星堆之名，最初来源于这一地区的地形特征——三座小山丘呈三角状分布，犹如夜空中的三星，故得名“三星堆”。三星堆名称的演变 三星堆遗址的发掘，始于20世纪20年代末期，但其真正引起广泛关注是在1986年两个大型祭祀坑的发现之后。在此之前，三星堆只是当地的一个地名，直到考古学家们在这里发现了大量的古代遗存，三星堆才逐渐成为了世界瞩目的焦点。随着研究的深入，三星堆之名也从一个简单的地理标识，演变为承载着深厚历史文化的符号。",
        "score": 0.897802,
        "date": "2024-06-07 19:00:51",
        "title": "三星堆名字的由来，你真的了解吗？三星堆的参观攻略，建议收藏青铜文物玉器古蜀国三星堆遗址_网易订阅",
        "url": "https://www.163.com/dy/article/J43Q2STU0553FV3Z.html",
        "site": "网易",
        "images":
      }
    ]
  }
}
```

```
[],\ "favicon\":\ "http://img01.sogoucdn.com/app/a/200913/a51423369a5d5a.png\ "},
  {\ "passage\":\ "一地理环境因素三星堆遗址位于四川省广汉市,距离成都约40公里在四川省广汉市西北部,有三座稍稍隆起的土堆呈直线排列,这种地形布局与夜空中闪烁的三颗星辰相呼应,因此得名“三星堆”这三个土堆历史悠久,据考古学家研究,它们的存在可以追溯到3000多年前这三个土堆不仅形态独特,而且在古代具有一定的宗教或象征意义在三星堆遗址的东北方,有一段残破的城墙,这段城墙呈现出弯月形状,与三个土堆隔河相望这种自然与人工相结合的景观,在当时的社会背景下,很可能被赋予了一定的神话色彩和宗教意义这一景观组合在历史上更是被形象地称为“三星伴月”,进一步增添了三星堆名字的文化韵味其中的“三星”指的就是那三个土堆,而“月”则是指那段弯月般的城墙当地居民根据这一独特的地理特征,赋予了这片土地以“三星伴月”的美称这种说法不仅反映了古人对于自然现象的崇拜和敬畏之情,同时也体现了人们对于美好事物的向往和赞美之心随着时间的推移,这一称谓逐渐深入人心,并最终演变成了“三星堆”这一名称\ "score\":0.8887853,\ "date\":\ "2024-09-03 12:59:00\ ",\ "title\":\ "三星堆名字的由来_腾讯新闻\ ",\ "url\":\ "https://news.qq.com/rain/a/20240903A048N700\ ",\ "site\":\ "腾讯网\ ",\ "images\":
[],\ "favicon\":\ "http://img02.sogoucdn.com/app/a/200913/ab735a258a90e8.png\ "},
  {\ "passage\":\ "最早在1929年,一位当地农民在遗址区域内偶然发现了一个玉器坑,这一发现引起了考古界的关注在四川省广汉市西北部,有<strong>三座稍稍隆起的土堆呈直线排列</strong>,这种地形布局与夜空中闪烁的<strong>三颗星辰</strong>相呼应,因此得名<strong>“三星堆”</strong>\ "score\":0.86471426,\ "date\":\ "2024-09-03 13:06:46\ ",\ "title\":\ "三星堆名字的由来\ ",\ "url\":\ "https://so.html5.qq.com/page/real/search_news?docid=70000021_14266d6996591252\ ",\ "site\":\ "企鹅号\ ",\ "images\":
[],\ "favicon\":\ "http://img02.sogoucdn.com/app/a/200913/ab735a25.png\ "}]
},
  "Query": "三星堆的由来",
  "RequestId": "6f8df221-9a68-4ea2-90d6-a9590cae244c"
}
```

## 错误码

当请求出现错误时,将返回 Error 对象,Error.Code 包含以下错误值:

错误码	描述	说明
InternalServerError	内部错误。	-

InvalidParameter	参数错误。	-
RequestLimitExceeded	请求的次数超过了频率限制。	-
ResourceNotFound	用户资源未开通。	请联系主账号开通服务。
ResourceUnavailable	用户资源不可用。	请检查账号是否欠费。
UnauthorizedOperation	未授权操作。	请检查服务密钥 API KEY 是否正确。

## 完整请求代码示例

### Python 示例

```
# 本代码仅供参考，具体请参照接口文档和实际需求进行开发

import os
import hashlib
import hmac
import json
import sys
import time
from datetime import datetime
if sys.version_info[0] <= 2:
    from httplib import HTTPSConnection
else:
    from http.client import HTTPSConnection

# 密钥信息从环境变量读取，需要提前在环境变量中设置 TENCENTCLOUD_WSA_APIKEY
# 使用环境变量方式可以避免密钥硬编码在代码中，提高安全性
# 生产环境建议使用更安全的密钥管理方案，如密钥管理系统 (KMS)、容器密钥注入等
# 密钥可前往官网控制台进行获取
secret_key = os.getenv("TENCENTCLOUD_WSA_APIKEY")

payload = "{\"Query\":\"三星堆的由来\"}"
params = json.loads(payload)
host = "api.wsa.cloud.tencent.com"
http_request_method = "POST"
```

```
uri = "/SearchPro"

# ***** 构造并发起请求 *****
headers = {
    "Authorization": "Bearer " + secret_key,
    "Content-Type": "application/json; charset=utf-8",
}

try:
    req = HTTPSConnection(host)
    req.request(http_request_method, uri, headers=headers,
body=payload.encode("utf-8"))
    resp = req.getresponse()
    print(resp.read().decode("utf-8"))
except Exception as err:
    print(err)
```

## Java 示例

```
// 本代码仅供参考，具体请参照接口文档和实际需求进行开发

import io.pyroscope.okhttp3.MediaType;
import io.pyroscope.okhttp3.OkHttpClient;
import io.pyroscope.okhttp3.Request;
import io.pyroscope.okhttp3.RequestBody;
import io.pyroscope.okhttp3.Response;
import java.io.IOException;

public class Main {
    public static void main(String[] args) throws IOException {
        // 密钥信息从环境变量读取，需要提前在环境变量中设置
        TENCENTCLOUD_WSA_APIKEY
        // 使用环境变量方式可以避免密钥硬编码在代码中，提高安全性
        // 生产环境建议使用更安全的密钥管理方案，如密钥管理系统 (KMS)、容器密钥注入等
        // 密钥可前往官网控制台进行获取
        String apiKey = System.getenv("TENCENTCLOUD_WSA_APIKEY");
        String body = "{\"Query\":\"三星堆的由来\"}";
        String resp = doRequest(body, apiKey);
    }
}
```

```
        System.out.println(resp);
    }

    // singleton client for connection reuse and better performance
    private static final OkHttpClient client = new OkHttpClient();

    public static String doRequest(String body, String apiKey) throws
    IOException {

        Request request = buildRequest(body, apiKey);

        System.out.println(request.method() + " " + request.url());
        System.out.println(request.headers());
        System.out.println(body);

        Response response = client.newCall(request).execute();
        return response.body().string();
    }

    public static Request buildRequest(String body, String apiKey) {
        String contentType = "application/json; charset=utf-8";
        return new Request.Builder()
            .header("Content-Type", contentType)
            .header("Authorization", "Bearer " + apiKey)
            .url("https://api.wsa.cloud.tencent.com/SearchPro")
            .post(RequestBody.create(MediaType.parse(contentType),
body))
            .build();
    }
}
```