# Python3.0版SDK说明

# 下载并安装k3cloud\_webapi\_sdk

使用pip安装指令格式：pip install {后缀为.whl的SDK包文件的本地完整目录}

例如：

pip install F:\Python\kingdee.cdp.webapi.sdk-8.0.4-py3-none-any.whl

# 配置文件参数说明

以**conf.ini**命令配置文件名称为例，节点为**config**：（配置参数名称不区分大小写）

[config]  
# 第三方系统登录授权的账套ID（即open.kingdee.com网站的第三方系统登录授权中的数据中心标识）  
X-KDApi-AcctID = 2020\*\*\*\*\*\*  
# 第三方系统登录授权的用户（用于调用接口的用户）  
X-KDApi-UserName = \*\*\*  
# 第三方系统登录授权的应用ID（即金蝶云星空产品中的第三方系统登录授权中的应用ID）  
X-KDApi-AppID = 21\*\*\*\_w53uT\*\*\*  
# 第三方系统登录授权的应用密钥（即金蝶云星空产品中的第三方系统登录授权中的应用密钥）  
X-KDApi-AppSec = \*\*\*\*\*\*  
# 服务Url地址(公有云统一走网关无需配置，需要注释掉，私有云需要配置金蝶云星空产品地址，K3Cloud/结尾)  
# X-KDApi-ServerUrl = https://\*\*\*\*\*\*/k3cloud/  
# 账套语系，默认2052  
# X-KDApi-LCID = 2052  
# 组织编码，启用多组织时配置对应的组织编码才有效  
# X-KDApi-OrgNum = 100  
# 允许的最大连接延时，单位为秒  
# X-KDApi-ConnectTimeout = 120  
# 允许的最大读取延时，单位为秒  
# X-KDApi-RequestTimeout = 120  
# 若使用代理，配置此参数  
# X-KDApi-Proxy = http://localhost:8888/

**说明：2021-08-31已更新，新增支持使用参数初始化方法，可不依赖配置文件。**

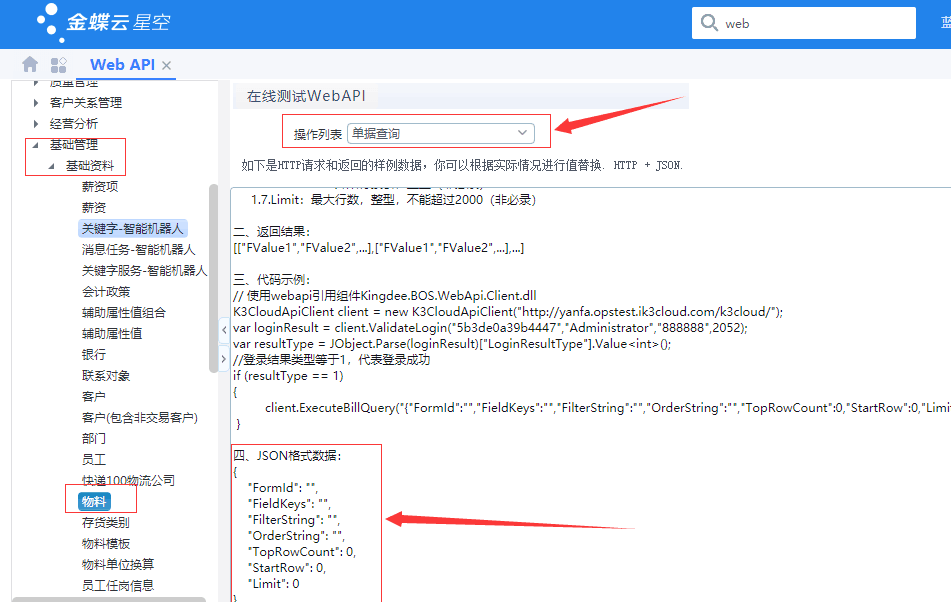
**（见示例工程）**

# 使用k3cloud\_webapi\_sdk的示例

## 接口通用示例参考

所有接口的参数格式，均可参考产品中WebAPI的示例说明，位置如下图所示：





或者登陆OPEN API官方网站查看API文档(<https://openapi.open.kingdee.com/>)。

下面举例部分接口示例：

## 前提：初始化SDK：

（通用部分，初始化一次即可，保存好SDK示例，可反复使用直接调用接口）

#!/usr/bin/python  
# -\*- coding:UTF-8 -\*-  
  
from k3cloud\_webapi\_sdk.main import K3CloudApiSdk  
  
# 首先构造一个SDK实例  
api\_sdk = K3CloudApiSdk()  
  
# 然后初始化SDK，需指定相关参数，否则会导致SDK初始化失败而无法使用：  
  
# 初始化方案一：Init初始化方法，使用conf.ini配置文件  
# config\_path:配置文件的相对或绝对路径，建议使用绝对路径  
# config\_node:配置文件中的节点名称  
# api\_sdk.Init(config\_path='conf.ini', config\_node='config')  
  
# 初始化方案二（新增）：InitConfig初始化方法，直接传参，不使用配置文件  
# acct\_id:第三方系统登录授权的账套ID,user\_name:第三方系统登录授权的用户,app\_id:第三方系统登录授权的应用ID,app\_sec:第三方系统登录授权的应用密钥  
# server\_url:k3cloud环境url(仅私有云环境需要传递),lcid:账套语系(默认2052),org\_num:组织编码(启用多组织时配置对应的组织编码才有效)  
api\_sdk.InitConfig('', '', '', '')

## 保存接口

# 此处仅构造保存接口的部分字段数据示例，使用时请参考WebAPI具体接口的实际参数列表  
current\_time = time.strftime('%Y%m%d%H%M%S', time.localtime())  
save\_data = {"Model": {  
 "FCreateOrgId": {"FNumber": 100},  
 "FUserOrgId": {"FNumber": 100},  
 "FNumber": "Webb" + current\_time + "10001",  
 "FName": "物料名称" + current\_time + "10001"  
}}  
  
# 调用sdk中的保存接口  
print(api\_sdk.Save("BD\_Material", save\_data))

## 批量保存接口

def gen\_seq(loop\_count):  
 # 此方法仅为生成物料编码的演示数据而写，使用时以实际数据为准  
 prefix = time.strftime('%Y%m%d%H%M%S', time.localtime())  
 list\_num = []  
 for index in range(0, loop\_count):  
 list\_num.append(prefix + str(10000 + index + 1))  
 return list\_num  
  
count = 10  
list\_seq = gen\_seq(count)  
# 构造批量保存接口的部分字段数据，使用时请参考WebAPI具体接口的实际参数列表  
list\_data = []  
for i in range(0, count):  
 list\_data.append({  
 "FCreateOrgId": {"FNumber": 100},  
 "FUserOrgId": {"FNumber": 100},  
 "FNumber": "Webb" + list\_seq[i],  
 "FName": "物料名称-" + list\_seq[i]  
 })  
save\_data = {"Model": list\_data}  
  
# 调用sdk中的批量保存接口（同步模式）  
print(api\_sdk.BatchSave("BD\_Material", save\_data))

## 单据查询接口

# 比如查询“物料”(BD\_MATERIAL)单据中的“名称”和“编码”字段

print(api\_sdk.ExecuteBillQuery(  
 {"FormId": "BD\_MATERIAL", "FieldKeys": "FName,FNumber", "FilterString": "FNumber like 'Webb%'", "TopRowCount": 5}))

## 查看接口

# 比如查询“物料”(BD\_MATERIAL)单据中的“名称”和“编码”字段  
print(api\_sdk.View("BD\_MATERIAL", {"Number": "Webb2019101514453410001"}))

## 提交接口

print(api\_sdk.Submit("BD\_MATERIAL", {"Numbers": ["Webb2019101514453410001"]}))

## 审核接口

print(api\_sdk.Audit("BD\_MATERIAL", {"Numbers": ["Webb2019101514453410001"]}))

## 反审核接口

print(api\_sdk.UnAudit("BD\_MATERIAL", {"Numbers": ["Webb2019101514453410001"]}))

## 删除接口

print(api\_sdk.Delete("BD\_MATERIAL", {"Numbers": ["Webb2019101514453410001"]}))

## 操作接口（以禁用、反禁用为例）

# 禁用  
print(api\_sdk.ExcuteOperation("BD\_MATERIAL", "Forbid", {"Numbers": ["Webb2019101514453410002"]}))  
# 反禁用  
# print(api\_sdk.ExcuteOperation("BD\_MATERIAL", "Enable", {"Numbers": ["Webb2019101514453410002"]}))

## 分配接口（需启用多组织）

print(api\_sdk.Allocate("BD\_MATERIAL", {"PkIds": 104378, "TOrgIds": "", "IsAutoSubmitAndAudit": "false"}))

## 更多接口请参考

