金蝶云星空WebAPI接口说明书\_V6.0

目 录

[1. 概述 3](#_Toc73621301)

[1.1. 目的 3](#_Toc73621302)

[1.2. 适用对象 3](#_Toc73621303)

[1.3. 参考资料 3](#_Toc73621304)

[2. 问题与解决策略 3](#_Toc73621305)

[3. 目标和约束 4](#_Toc73621306)

[4. WebAPI架构 4](#_Toc73621307)

[4.1. 采用的技术 4](#_Toc73621308)

[4.1.1. Kingdee.BOS.WebApi.FormService.dll 4](#_Toc73621309)

[4.1.2. Kingdee.BOS.WebApi.ServicesStub.dll 4](#_Toc73621310)

[4.1.3. Kingdee.BOS.WebApi.Client.dll 4](#_Toc73621311)

[4.1.4. 开发工具 4](#_Toc73621312)

[5. WebAPI接口详细描述 5](#_Toc73621313)

[5.1.1. 登陆验证接口 5](#_Toc73621314)

[5.1.2. 查看表单数据接口 8](#_Toc73621315)

[5.1.3. 保存表单数据接口 9](#_Toc73621316)

[5.1.4. 批量保存表单数据接口 12](#_Toc73621317)

[5.1.5. 提交表单数据接口 16](#_Toc73621318)

[5.1.6. 审核表单数据接口 17](#_Toc73621319)

[5.1.7. 反审核表单数据接口 18](#_Toc73621320)

[5.1.8. 删除表单数据接口 20](#_Toc73621321)

[5.1.9. 表单数据查询接口 21](#_Toc73621322)

[5.1.10. 自定义WebAPI接口 22](#_Toc73621323)

[5.1.11. 登录验证接口带踢人功能 26](#_Toc73621324)

[5.1.12. 暂存表单数据接口 26](#_Toc73621325)

[5.1.13. 分配表单数据接口 26](#_Toc73621326)

[5.1.14 下推接口 28](#_Toc73621327)

[5.1.15 分组保存接口 29](#_Toc73621328)

[5.1.16 弹性域保存接口 30](#_Toc73621329)

[5.1.17 发送消息接口 31](#_Toc73621330)

[5.1.18 登出接口 32](#_Toc73621331)

[5.1.19 通用操作接口 32](#_Toc73621332)

[5.1.20 切换上下文默认组织接口 33](#_Toc73621333)

[5.1.1. 工作流审批接口 34](#_Toc73621334)

[6. 常见错误代码 36](#_Toc73621335)

# 概述

## 目的

为第三方系统访问金蝶云星空系统数据提供通用的接口。

当企业规模逐渐增大时，作为支撑业务运营的IT建设也变得越来越重要，不过往往企业的IT建设过程中会发现某一家软件供应商基本不能完全覆盖企业所有的业务运营流程，这样的结果就是，企业上的IT系统很多很全，从ERP到HR、CRM、PDM、OA等，貌似所有的业务都覆盖到了，但实际上因为这些系统的不集成，而形成了企业很多新的信息孤岛，非常不利于企业的后续的管理和战略发展。金蝶云星空从现今和往后的发展趋势来看，也不可避免会遇到上述问题，毕竟企业经营的多样化，并不是所有的业务都能在金蝶云星空中完成，所以金蝶云星空产品需要在技术架构上支持更好的与外部系统进行协同。

## 适用对象

本文档适用于：

* 开发工程师：**参考**，对系统集成的实现获取**全局性**的设计**指导**。

## 参考资料

# 问题与解决策略

|  |  |  |
| --- | --- | --- |
| **愿景** | **关注点** | **描述与示例** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 目标和约束

**目标：**

* 提供对金蝶云星空单据和基础信息的查看、保存、提交、审核、反审核和删除等功能；
* 提供对金蝶云星空单据和基础信息的查询功能；

**约束：**

* 数据操作接口仅支持以基础资料编码、单据编号或直接以表单主键去操作数据；
* 支持对某一具体单据的数据查询，但多单关联查询需要二开接口实现。

# WebAPI架构

## 采用的技术

金蝶云星空 WebAPI是一种轻量级的、可维护的、可伸缩的 Web 服务。采用HTTP+JSON，也就是用RESTful的方式来开发。使用.NET Framework 4.0为开发平台，源代码使用C#编写。整个框架由三个组装件组成。

### Kingdee.BOS.WebApi.FormService.dll

此组装件包含WebAPI主要接口的功能实现。部署在应用层服务器。

### Kingdee.BOS.WebApi.ServicesStub.dll

此组装件主要包含WebAPI接口定义，扩展接口定义以及登陆验证接口。部署在应用层服务器。

### Kingdee.BOS.WebApi.Client.dll

此组装件为WebAPI的客户端组件，封装了一些在异构系统客户端访问WebAPI的方法，适用于C#程序调用。由于它应用于异构系统客户端，所以此组装件需要拷贝到异构系统客户端环境中。非C#程序调用可以不用拷贝。

### 开发工具

.Net FrameWork 4.0

Microsoft Visual Studio 2012

# WebAPI接口详细描述

### 登录验证接口

服务地址：

[https:// ServerIp /K3Cloud/Kingdee.BOS.WebApi.ServicesStub.AuthService.LoginBySign.common.kdsvc](https://域名/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.AuthService.LoginBySign.common.kdsvc)

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| acctID | 账套Id，从管理中心数据库查询获得 参考查询：select FDATACENTERID from T\_BAS\_DATACENTER | 必须 |
| username | 用户登录名 | 必须 |
| appId | 应用ID | 必须 |
| timestamp | 时间戳 |  |
| sign | 签名信息 | 签名信息生成方式：  将账套ID、用户名、应用ID、应用秘钥、时间戳 放到数组里面，将数组排序以后 使用SHA256加密以后生成。  注意：PT-146911 8.0.0.202205 之前的版本不支持SHA256加密，需要使用SHA1加密算法 |
| lcid | 语言id,选择哪种语言访问，参考：中文2052，英文1033，繁体3076 |  |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| LoginResultType | //激活  Activation = -7,  //云通行证未绑定Cloud账号  EntryCloudUnBind = -6,   //需要表单处理  DealWithForm = -5,  //登录警告  Wanning = -4,  //密码验证不通过（强制的）  PWInvalid\_Required = -3,  //密码验证不通过（可选的）  PWInvalid\_Optional = -2,  //登录失败  Failure = -1,  //用户或密码错误  PWError = 0,  //登录成功  Success = 1 | 管理员登录可能出现返回-5的情况，这种情况在api验证时也可认为是允许的，具体可以根据实际情况来定。  var result = JObject.Parse(ret)["LoginResultType"].Value<int>();  if (result == 1||result==-5)  {  return true;  } |

调用参考：

1. SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）

说明：下文中出现的client都为此处进行过登录验证的ApiClient实例。

static void Main(string[] args)

        {

            ApiClient client = new ApiClient("http://localhost/k3cloud/");

            //时间戳

            long timestamp = DateTimeFormatUtils.CurrentTimeMillis() / 1000;

            //数据中心ID

            string dbId = "账套ID";

            //用户名称

            string usserName = "用户名";

            //应用Id

            string appId = "应用ID";

            //应用秘钥

            string appSecret = "应用密钥";

            //将账套ID、用户名、应用ID、应用秘钥、时间戳 放到数组里面

            string[] arr = new string[] { dbId, usserName, appId, appSecret, timestamp.ToString() };

            //调用SHA加密算法生成签名信息，加密算法见下面原生开发整改方式 里面的代码示例，注意PT-146911 8.0.0.202205 之前的版本需要使用SHA1加密算法

            string sign = GetSHA256(arr);

            //调用LoginBySign方法

            var loginResult = client.LoginBySign(dbId, usserName, appId, timestamp, sign, 2052);

            //判断登录结果

            var resultType = JObject.Parse(loginResult)["LoginResultType"].Value<int>();

            //登录结果类型等于1，代表登录成功

            if (resultType == 1)

            {

                //调其他接口

            }

        }

private static string GetSHA256(string[] arr)

        {

            if (arr != null && arr.Length > 0)

            {

                Array.Sort(arr, StringComparer.Ordinal);

                byte[] bytes = Encoding.UTF8.GetBytes(string.Join("", arr));

                using (SHA256 sHA = SHA256.Create())

                {

                    byte[] data = sHA.ComputeHash(bytes);

                    var hex = BitConverter.ToString(data).Replace("-", string.Empty);

                    return hex.ToLowerInvariant();

                }

            }

            return string.Empty;

        }

1. 无引用组件示例（不引用金蝶的组件）

using System;

using System.IO;

using System.Net;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

using Newtonsoft.Json;

namespace ConsoleAppFramework

{

    class Program

    {

        static void Main(string[] args)

        {

            // 账套Id，用户名，第三方登录授权应用Id，第三方登录授权应用密钥，语言标识符

            Login("xxxx", "xxxx", "xxxx", "xxxx", 2052);

        }

        private static void Login(string acctId, string userName, string appId, string appSecret, int lcid = 2052)

        {

            var timestamp = CurrentTimeMillis() / 1000;

            var signArray = new string[] { acctId, userName, appId, appSecret, timestamp.ToString() };

            // PT-146911 8.0.0.202205 之前的版本不支持SHA256加密，需要使用 GetSha1(signArray)

            var sign = GetSHA256(signArray);

            string url = "http://172.17.8.158/k3cloud/Kingdee.BOS.WebApi.ServicesStub.AuthService.LoginBySign.common.kdsvc";

            var request = (HttpWebRequest)WebRequest.Create(url);

            request.Method = "POST";

            request.ContentType = "application/json;charset=UTF-8";

            var reqData = new

            {

                acctID = acctId,

                username = userName,

                appId = appId,

                timestamp = timestamp,

                sign = sign,

                lcid = lcid

            };

            var reqJsonData = JsonConvert.SerializeObject(reqData);

            byte[] byteData = Encoding.UTF8.GetBytes(reqJsonData);

            int length = byteData.Length;

            request.ContentLength = length;

            using (Stream writer = request.GetRequestStream())

            {

                writer.Write(byteData, 0, length);

                writer.Close();

            }

            using (var response = (HttpWebResponse)request.GetResponse())

            {

                var responseString = new StreamReader(response.GetResponseStream(), Encoding.UTF8).ReadToEnd();

                Console.WriteLine(responseString);

            }

        }

        private static string GetSHA256(string[] arr)

        {

            if (arr != null && arr.Length > 0)

            {

                Array.Sort(arr, StringComparer.Ordinal);

                byte[] bytes = Encoding.UTF8.GetBytes(string.Join("", arr));

                using (SHA256 sHA = SHA256.Create())

                {

                    byte[] data = sHA.ComputeHash(bytes);

                    var hex = BitConverter.ToString(data).Replace("-", string.Empty);

                    return hex.ToLowerInvariant();

                }

            }

            return string.Empty;

        }

        private static string GetSha1(string[] arr)

        {

            //1. 将数组进行排序

            //2. 将数组拼接成一个字符串进行sha1加密

            Array.Sort(arr, StringComparer.Ordinal);

            var arrString = string.Join("", arr);

            var sha1 = SHA1.Create();

            var sha1Arr = sha1.ComputeHash(Encoding.UTF8.GetBytes(arrString));

            StringBuilder enText = new StringBuilder();

            foreach (var b in sha1Arr)

            {

                enText.AppendFormat("{0:x2}", b);

            }

            return enText.ToString();

        }

        private static DateTime Jan1st1970 = new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);

        private static long CurrentTimeMillis()

        {

            return (long)((DateTime.UtcNow - Jan1st1970).TotalMilliseconds);

        }

    }

}

1. 使用第三方集成密钥文件方式登录

登录代码示例参考: <https://vip.kingdee.com/article/22914>

1. 一次登录多次使用示例

登录一次默认会保持20分钟会话，如果20分钟内调用其它接口，会话时间会重新算；如果后台处理时间超长或后台把会话清理了，接口调用返回的结果会提示会话丢失，只需要重新登录即可，不需要每次调用其它接口之前去调用登录接口。

示例代码参考: <https://vip.kingdee.com/knowledge/specialDetail/650386937144032256?category=650389982225382656&id=650363039845203200&type=Knowledge&productLineId=1&lang=zh-CN>

1. **注意事项**

不要每次调用WebAPI都去先调用登录接口，该方式会导致建立太多连接占用系统资源，也会导致产生过多的上下文缓存不释放问题；

### 查看表单数据接口

服务地址：

<http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.View.common.kdsvc>

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| formid | 表单Id，格式参考："IV\_SALESIC"//表示发票 | 必须 |
| data | 数据包，Number表示单据编号或者基础资料编码，Id为主键，CreateOrgId为创建组织Id。格式参考："{\"CreateOrgId\":0,\"Id\":0,\"Number\":\"SVINV00000003\"}" | CreateOrgId：非必须 Id和Number：任选一个 |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| ResponseStatus | 操作状态 | 操作是否成功，如果失败，具体失败原因 |
| Result | 单据数据包 | 单据完整数据内容 |

调用参考：

1. SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）：

string sJson = "{\"Number\":\"SVINV00000003\"}";

var result = client.Execute<string>("Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.View",

new object[] { "IV\_SALESIC", sJson });

1. 无引用组件示例（不引用金蝶的组件）：

HttpClient httpClient = new HttpClient();

httpClient.Url = "http://192.168.66.60/k3cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.View.common.kdsvc";

List<object> Parameters = new List<object>();

//业务对象Id

String formid = "IV\_SALESIC";//发票

Parameters.Add(formid);

//Json字串

string data = "{\"Number\":\"SVINV00000003\"}";

Parameters.Add(data);

httpClient.Content = JsonConvert.SerializeObject(Parameters);

var result = httpClient.AsyncRequest();

### 保存表单数据接口

服务地址：

[http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.Save.common.kdsvc](http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.View.common.kdsvc)

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| formid | 表单Id，格式参考："BD\_Currency"//表示币别 | 必须 |
| data | 数据包，数据格式参考以币别为例："{\"Creator\":\"\", \"NeedUpDateFields\":[\"\"],  \"IsDeleteEntry\":\"false\",  \"NeedReturnFields\":\" \",  \"IsVerifyBaseDataField\":\"false\",  \"IsEntryBatchFill\":\"false\",  \"InterationFlags\":\"\",  \"IgnoreInterationFlag\":\"\", \"Model\":{\"FNumber\": \"PRENB001\", \"FName\":\"牛币\",\"FCODE\":\"CNYY\", \"FPRICEDIGITS\":6, \"FAMOUNTDIGITS\":2, \"FPRIORITY\":0, \"FIsShowCSymbol\":true}}" | Creator：非必须 NeedUpDateFields：非必须，如果是更新单据而不是新增一个单据，可以设置这个数据集合，内容为字段标识，使用逗号分隔，如果需要更新表体数据，则需要填写表体的标识。一般情况下，如果Model中只填写了需要更新的字段信息，没有其他冗余的字段，那么这个参数不需要设置，否则其他冗余字段会覆盖ERP中的源单数据。 IsDeleteEntry:非必须，在修改单据时是否需要删除已经存在的分录。  NeedReturnFields:非必须，需要返回的所有字段。  IsVerifyBaseDataField:非必须，是否需要验证基础资料。  IsEntryBatchFill:非必须，是否需要批量填充分录。  Model：必须，Model数据格式说明：所有字段和实体都是用元素的标识来识别。单据头字段直接填写字段标识和数据，子单据头字段需要先申明是子单据头信息，然后再填写其字段信息，例如\"SubHeadEntity\":{\"FBARCODE\":\"20305\"}如果是单据体字段则需要申明为集合，例如：\"EntryDetail\":[{第1条明细},{第N条明细}]，需要用中括号括起来。  \"InterationFlags\":交互标志集合，字符串类型，分号分隔，格式：  IgnoreInterationFlag：是否允许忽略交互，布尔类型，默认true（非必录）  "flag1;flag2;..."（非必录） 例如（允许负库存标识：STK\_InvCheckResult） |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| ResponseStatus | 操作状态 {"ResponseStatus":{"ErrorCode":"",  "IsSuccess":false,  "Errors":[{"FieldName":"",  "Message":"","DIndex":0}],  "SuccessEntitys":[{"Id":"","Number":"","DIndex":0}],  "SuccessMessages":[{"FieldName":"","Message":"","DIndex":0}],"MsgCode":0} | IsSuccess:操作是否成功， Errors:如果失败，具体失败原因 DIndex:原始数据行号,  MsgCode:等于1，会话丢失，重新登录；等于0，其他情况 |

调用参考：

1）SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）：

//业务对象Id

string sFormId = "SAL\_OUTSTOCK"; //销售出库单

//Model字串

string sContent = "{\"Creator\":\"\",\"NeedUpDateFields\":[],\"Model\":" + "{\"FID\":\"0\",\"FStockOrgId\":{\"FNumber\":\"210\"},\"FBillTypeID\":{\"FNumber\":\"XSCKD01\_SYS\"},\"FBillNo\":\"CSDGBC21002\",\"FCustomerID\":{\"FNumber\":\"CUST0073\"},\"SubHeadEntity\":{\"FExchangeRate\":6.51},\"FEntity\":[{\"FEntryID\":\"0\",\"FMATERIALID\":{\"FNumber\":\"03.001\"},\"FStockID\":{\"FNumber\":\"CK002\"},\"FRealQty\":324,\"FBaseUnitQty\":324},{\"FEntryID\":\"0\",\"FMATERIALID\":{\"FNumber\":\"03.001\"},\"FStockID\":{\"FNumber\":\"CK004\"},\"FRealQty\":220,\"FBaseUnitQty\":220}]]}}";

object[] saveInfo = new object[]

{

sFormId,

sContent

};

//调用保存接口

var ret = client.Execute<string>("Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.Save", saveInfo);

以下示例说明如何更新单据的表体信息：

更新内码为100017的单据，新增两行表体数据，原有100024和100025行数据保留，如果源单中还有其他分录则删除。

string sFormId = "STK\_MISCELLANEOUS";//其他入库单

string sContent = "{\"Creator\":\"\",\"NeedUpDateFields\":[],\"Model\":{\"FID\":\"100017\",\"FEntity\":[{\"FEntryID\":\"100024\"},{\"FEntryID\":\"100025\"}," + "{\"FMATERIALID\":{\"FNumber\":\"A.0060480933\"},\"FUnitID\":{\"FNumber\":\"03\"},\"FSTOCKID\":{\"FNumber\":\"PRTSH\"},\"FQty\":\"200\",\"FBASEQTY\":\"200\",\"FPlanAmount\":\"132856\",\"FBASEUNITID\":{\"FNumber\":\"03\"},\"FEntryNOTE\":\"2015-7-29\",\"FAmount\":\"132856\",\"FPRICE\":\"664.28\",\"FEntrySelfA9725\":\"2015072802\",\"FEntrySelfA9733\":\"19881\",\"FEntrySelfA9734\":\"2015-6-3\",\"FEntrySelfA9737\":\"200\",\"FEntrySelfA9740\":\"0\",\"FEntrySelfA9738\":\"1\",\"FEntrySelfA9739\":\"2\",\"FEntrySelfA9735\":\"664.28\"}," +

"{\"FMATERIALID\":{\"FNumber\":\"A.0060480933"},\"FUnitID\":{\"FNumber\":\"03\"},\"FSTOCKID\":{\"FNumber\":\"PRTSH\"},\"FQty\":\"200\",\"FBASEQTY\":\"200\",\"FPlanAmount\":\"132856\",\"FBASEUNITID\":{\"FNumber\":\"03\"},\"FEntryNOTE\":\"2015-7-29\",\"FAmount\":\"132856\",\"FPRICE\":\"664.28\",\"FEntrySelfA9725\":\"2015072802\",\"FEntrySelfA9733\":\"19881\",\"FEntrySelfA9734\":\"2015-6-3\",\"FEntrySelfA9737\":\"200\",\"FEntrySelfA9740\":\"0\",\"FEntrySelfA9738\":\"1\",\"FEntrySelfA9739\":\"2\",\"FEntrySelfA9735\":\"664.28\"}]}}";

object[] saveInfo = new object[]

{

sFormId,

sContent

};

//调用保存接口

var ret=

client.Execute<string>("Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.Save", saveInfo);

2）无引用组件示例（不引用金蝶的组件）：

HttpClient httpClient = new HttpClient();

httpClient.Url = "http://192.168.66.60/k3cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.Save.common.kdsvc";

List<object> Parameters = new List<object>();

//业务对象Id

String formid = "SAL\_OUTSTOCK";//销售出库为例

Parameters.Add(formid);

//Json字串

string data = "{\"Creator\":\"\",\"NeedUpDateFields\":[],\"Model\":" + "{\"FID\":\"0\",\"FStockOrgId\":{\"FNumber\":\"210\"},\"FBillTypeID\":{\"FNumber\":\"XSCKD01\_SYS\"},\"FBillNo\":\"CSDGBC21002\",\"FCustomerID\":{\"FNumber\":\"CUST0073\"},\"SubHeadEntity\":{\"FExchangeRate\":6.51},\"FEntity\":[{\"FEntryID\":\"0\",\"FMATERIALID\":{\"FNumber\":\"03.001\"},\"FStockID\":{\"FNumber\":\"CK002\"},\"FRealQty\":324,\"FBaseUnitQty\":324},{\"FEntryID\":\"0\",\"FMATERIALID\":{\"FNumber\":\"03.001\"},\"FStockID\":{\"FNumber\":\"CK004\"},\"FRealQty\":220,\"FBaseUnitQty\":220}]]}}";

Parameters.Add(data);

httpClient.Content = JsonConvert.SerializeObject(Parameters);

var result = httpClient.AsyncRequest();

### 批量保存表单数据接口

服务地址：

[http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.BatchSave.common.kdsvc](http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.View.common.kdsvc)

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| formid | 表单Id，格式参考："SAL\_SaleOrder"//表示销售订单 | 必须 |
| data | 数据包，格式参考以销售订单为例：  "{\"NeedUpDateFields\":[],\"BatchCount\":\"2\",  \"Model\":[{\"FID\":\"0\",\"FBILLTYPEID\":{\"FNumber\":\"XSDD01\_SYS\"},\"FSaleOrgId\":{\"FNumber\":\"NCIC\"},\"FSaleDeptId\":{\"FNumber\":\"01.A13\"},\"FCustId\":{\"FNumber\":\"500\"},\"FDATE\":\"2016/2/23\",\"FSettleCurrId\":{\"FNumber\":\"RMB\"},\"FSalerId\":{\"FNumber\":\"B027\"},\"FBusinessType\":\"NORMAL\",\"FSaleOrderFinance\":{\"FEXCHANGETYPE\":{\"FNumber\":\"NCIC\"},\"FExchangeRate\":\"1\"},\"FSaleOrderEntry\":[{\"FMATERIALID\":{\"FNumber\":\"A.0060480931\"},\"FUNITID\":{\"FNumber\":\"03\"},\"FQty\":\"1\",\"FTaxPrice\":\"100\",\"FEntryTaxRate\":\"17.00\"}]},{\"FID\":\"0\",\"FBILLTYPEID\":{\"FNumber\":\"XSDD01\_SYS\"},\"FSaleOrgId\":{\"FNumber\":\"NCIC\"},\"FSaleDeptId\":{\"FNumber\":\"01.A13\"},\"FCustId\":{\"FNumber\":\"500\"},\"FDATE\":\"2016/2/23\",\"FSettleCurrId\":{\"FNumber\":\"RMB\"},\"FSalerId\":{\"FNumber\":\"B027\"},\"FBusinessType\":\"NORMAL\",\"FSaleOrderFinance\":{\"FEXCHANGETYPE\":{\"FNumber\":\"NCIC\"},\"FExchangeRate\":\"1\"},\"FSaleOrderEntry\":[{\"FMATERIALID\":{\"FNumber\":\"A.0060480931 \"},  \"FUNITID\":{\"FNumber\":\"03\"},\"FQty\":\"1\",\"FTaxPrice\":\"100\",\"FEntryTaxRate\":\"17.00\"}]},{\"FID\":\"0\",\"FBILLTYPEID\":{\"FNumber\":\"XSDD01\_SYS\"},\"FSaleOrgId\":{\"FNumber\":\"NCIC\"},\"FSaleDeptId\":{\"FNumber\":\"01.A13\"},\"FCustId\":{\"FNumber\":\"500\"},\"FDATE\":\"2016/2/23\",\"FSettleCurrId\":{\"FNumber\":\"RMB\"},\"FSalerId\":{\"FNumber\":\"B027\"},\"FBusinessType\":\"NORMAL\",\"FSaleOrderFinance\":{\"FEXCHANGETYPE\":{\"FNumber\":\"NCIC\"},\"FExchangeRate\":\"1\"},\"FSaleOrderEntry\":[{\"FMATERIALID\":{\"FNumber\":\"A.0060480931\"},\"FUNITID\":{\"FNumber\":\"03\"},\"FQty\":\"1\",\"FTaxPrice\":\"100\",\"FEntryTaxRate\":\"17.00\"}]}]}" | Data中的参数跟保存接口是一样的，另外还有BatchCount：非必须。此参数主要用于优化性能，当传入的单据数据量较大时，可以设定此参数的并行分批执行次数。例如传入100张单据数据，此参数设定为10，则表示在k3cloud系统中，以10个单据为一批，分10批，同时并发保存，提升效率。 数据包参数格式和Save接口的类似，主要差别在于批量保存是传入多张单据的数据，Modle数据用[]括起来，而Save仅传入一张单据数据。再一个区别是批量保存多了一个参数BatchCount。 |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| ResponseStatus | 操作状态 {"ResponseStatus":{"ErrorCode":"",  "IsSuccess":false,  "Errors":[{"FieldName":"",  "Message":"","DIndex":0}],  "SuccessEntitys":[{"Id":"","Number":"","DIndex":0}],  "SuccessMessages":[{"FieldName":"","Message":"","DIndex":0}]，"MsgCode":0} | IsSuccess:操作是否成功， Errors:如果失败，具体失败原因 DIndex:原始数据行号，如果接口参数启用了BatchCount，则DIndex返回的顺序不是所有数据的顺序，而是每批次中  MsgCode:等于1，会话丢失，重新登录；等于0，其他情况的顺序。 |

调用参考：

1. SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）：

string sFormId = "SAL\_SaleOrder";//销售订单为例

string sContent = "{\"NeedUpDateFields\":[],\"BatchCount\":\"2\",\"Model\":[{\"FID\":\"0\",\"FBILLTYPEID\":{\"FNumber\":\"XSDD01\_SYS\"},\"FSaleOrgId\":{\"FNumber\":\"NCIC\"},\"FSaleDeptId\":{\"FNumber\":\"01.A13\"},\"FCustId\":{\"FNumber\":\"500\"},\"FDATE\":\"2016/2/23\",\"FSettleCurrId\":{\"FNumber\":\"RMB\"},\"FSalerId\":{\"FNumber\":\"B027\"},\"FBusinessType\":\"NORMAL\",\"FSaleOrderFinance\":{\"FEXCHANGETYPE\":{\"FNumber\":\"NCIC\"},\"FExchangeRate\":\"1\"},\"FSaleOrderEntry\":[{\"FMATERIALID\":{\"FNumber\":\"A.0060480931 \"},\"FUNITID\":{\"FNumber\":\"03\"},\"FQty\":\"1\",\"FTaxPrice\":\"100\",\"FEntryTaxRate\":\"17.00\"}]},{\"FID\":\"0\",\"FBILLTYPEID\":{\"FNumber\":\"XSDD01\_SYS\"},\"FSaleOrgId\":{\"FNumber\":\"NCIC\"},\"FSaleDeptId\":{\"FNumber\":\"01.A13\"},\"FCustId\":{\"FNumber\":\"500\"},\"FDATE\":\"2016/2/23\",\"FSettleCurrId\":{\"FNumber\":\"RMB\"},\"FSalerId\":{\"FNumber\":\"B027\"},\"FBusinessType\":\"NORMAL\",\"FSaleOrderFinance\":{\"FEXCHANGETYPE\":{\"FNumber\":\"NCIC\"},\"FExchangeRate\":\"1\"},\"FSaleOrderEntry\":[{\"FMATERIALID\":{\"FNumber\":\"A.0060480931 \"},\"FUNITID\":{\"FNumber\":\"03\"},\"FQty\":\"1\",\"FTaxPrice\":\"100\",\"FEntryTaxRate\":\"17.00\"}]},{\"FID\":\"0\",\"FBILLTYPEID\":{\"FNumber\":\"XSDD01\_SYS\"},\"FSaleOrgId\":{\"FNumber\":\"NCIC\"},\"FSaleDeptId\":{\"FNumber\":\"01.A13\"},\"FCustId\":{\"FNumber\":\"500\"},\"FDATE\":\"2016/2/23\",\"FSettleCurrId\":{\"FNumber\":\"RMB\"},\"FSalerId\":{\"FNumber\":\"B027\"},\"FBusinessType\":\"NORMAL\",\"FSaleOrderFinance\":{\"FEXCHANGETYPE\":{\"FNumber\":\"NCIC\"},\"FExchangeRate\":\"1\"},\"FSaleOrderEntry\":[{\"FMATERIALID\":{\"FNumber\":\"A.0060480931 \"},\"FUNITID\":{\"FNumber\":\"03\"},\"FQty\":\"1\",\"FTaxPrice\":\"100\",\"FEntryTaxRate\":\"17.00\"}]}]}";

object[] saveInfo = new object[]

{

sFormId,

sContent

};

var ret =

client.Execute<string>("Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.BatchSave", saveInfo);

2）无引用组件示例（不引用金蝶的组件）：

HttpClient httpClient = new HttpClient();

httpClient.Url = "http://192.168.66.60/k3cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.BatchSave.common.kdsvc";

List<object> Parameters = new List<object>();

//业务对象Id

String formid = "SAL\_SaleOrder";//销售订单为例

Parameters.Add(formid);

//Json字串

string data = "{\"NeedUpDateFields\":[],\"BatchCount\":\"2\",\"Model\":[{\"FID\":\"0\",\"FBILLTYPEID\":{\"FNumber\":\"XSDD01\_SYS\"},\"FSaleOrgId\":{\"FNumber\":\"NCIC\"},\"FSaleDeptId\":{\"FNumber\":\"01.A13\"},\"FCustId\":{\"FNumber\":\"500\"},\"FDATE\":\"2016/2/23\",\"FSettleCurrId\":{\"FNumber\":\"RMB\"},\"FSalerId\":{\"FNumber\":\"B027\"},\"FBusinessType\":\"NORMAL\",\"FSaleOrderFinance\":{\"FEXCHANGETYPE\":{\"FNumber\":\"NCIC\"},\"FExchangeRate\":\"1\"},\"FSaleOrderEntry\":[{\"FMATERIALID\":{\"FNumber\":\"A.0060480931 \"},\"FUNITID\":{\"FNumber\":\"03\"},\"FQty\":\"1\",\"FTaxPrice\":\"100\",\"FEntryTaxRate\":\"17.00\"}]},{\"FID\":\"0\",\"FBILLTYPEID\":{\"FNumber\":\"XSDD01\_SYS\"},\"FSaleOrgId\":{\"FNumber\":\"NCIC\"},\"FSaleDeptId\":{\"FNumber\":\"01.A13\"},\"FCustId\":{\"FNumber\":\"500\"},\"FDATE\":\"2016/2/23\",\"FSettleCurrId\":{\"FNumber\":\"RMB\"},\"FSalerId\":{\"FNumber\":\"B027\"},\"FBusinessType\":\"NORMAL\",\"FSaleOrderFinance\":{\"FEXCHANGETYPE\":{\"FNumber\":\"NCIC\"},\"FExchangeRate\":\"1\"},\"FSaleOrderEntry\":[{\"FMATERIALID\":{\"FNumber\":\"A.0060480931 \"},\"FUNITID\":{\"FNumber\":\"03\"},\"FQty\":\"1\",\"FTaxPrice\":\"100\",\"FEntryTaxRate\":\"17.00\"}]},{\"FID\":\"0\",\"FBILLTYPEID\":{\"FNumber\":\"XSDD01\_SYS\"},\"FSaleOrgId\":{\"FNumber\":\"NCIC\"},\"FSaleDeptId\":{\"FNumber\":\"01.A13\"},\"FCustId\":{\"FNumber\":\"500\"},\"FDATE\":\"2016/2/23\",\"FSettleCurrId\":{\"FNumber\":\"RMB\"},\"FSalerId\":{\"FNumber\":\"B027\"},\"FBusinessType\":\"NORMAL\",\"FSaleOrderFinance\":{\"FEXCHANGETYPE\":{\"FNumber\":\"NCIC\"},\"FExchangeRate\":\"1\"},\"FSaleOrderEntry\":[{\"FMATERIALID\":{\"FNumber\":\"A.0060480931 \"},\"FUNITID\":{\"FNumber\":\"03\"},\"FQty\":\"1\",\"FTaxPrice\":\"100\",\"FEntryTaxRate\":\"17.00\"}]}]}";

Parameters.Add(data);

httpClient.Content = JsonConvert.SerializeObject(Parameters);

var result = httpClient.AsyncRequest();

### 提交表单数据接口

服务地址：

[http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.Submit.common.kdsvc](http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.View.common.kdsvc)

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| formid | 表单Id，格式参考："BD\_Currency"//表示币别 | 必须 |
| data | 数据包，Number表示单据编号或者基础资料编码，Id为主键，CreateOrgId为创建组织Id。数据格式参考以币别为例："{\"CreateOrgId\":0,\"UseOrgId\":0,\"Numbers\":[\"PRE002\",\"PRE006\"]}" | CreateOrgId：创建组织Id,非必须  UseOrgId：使用组织Id,非必须 Numbers：必须 |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| ResponseStatus | 操作状态 {"ResponseStatus":{"ErrorCode":"",  "IsSuccess":false,  "Errors":[{"FieldName":"",  "Message":"","DIndex":0}],  "SuccessEntitys":[{"Id":"","Number":"","DIndex":0}],  "SuccessMessages":[{"FieldName":"","Message":"","DIndex":0}],”MsgCode”:0} | IsSuccess:操作是否成功， Errors:如果失败，具体失败原因 DIndex:原始数据行号，  MsgCode:等于1，会话丢失，重新登录；等于0，其他情况 |

调用参考：

1）SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）：

string sJson = "{\"CreateOrgId\":0,\"Numbers\":[\"PRE002\",\"PRE006\"]}";

var result = client.Execute<string>("Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.Submit",

new object[] { "BD\_Currency", sJson });

1. 无引用组件示例（不引用金蝶的组件）：

HttpClient httpClient = new HttpClient();

httpClient.Url = "http://192.168.66.60/k3cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.Submit.common.kdsvc";

List<object> Parameters = new List<object>();

//业务对象Id

String formid = "BD\_Currency";//币别为例

Parameters.Add(formid);

//Json字串

string data = "{\"CreateOrgId\":0,\"Numbers\":[\"PRE002\",\"PRE006\"]}";

Parameters.Add(data);

httpClient.Content = JsonConvert.SerializeObject(Parameters);

var result = httpClient.AsyncRequest();

### 审核表单数据接口

服务地址：

[http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.Audit.common.kdsvc](http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.View.common.kdsvc)

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| formid | 表单Id，格式参考："BD\_Currency"//表示币别 | 必须 |
| data | 数据包，Number表示单据编号或者基础资料编码，Id为主键，CreateOrgId为创建组织Id。数据格式参考以币别为例："{\"CreateOrgId\":0,\"UseOrgId\":0,\"Numbers\":[\"PRE002\",\"PRE006\"]}" | CreateOrgId：创建组织Id,非必须  UseOrgId：使用组织Id,非必须 Numbers：必须 |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| ResponseStatus | 操作状态 {"ResponseStatus":{"ErrorCode":"",  "IsSuccess":false,  "Errors":[{"FieldName":"",  "Message":"","DIndex":0}],  "SuccessEntitys":[{"Id":"","Number":"","DIndex":0}],  "SuccessMessages":[{"FieldName":"","Message":"","DIndex":0}],  “MsgCode”:0} | IsSuccess:操作是否成功， Errors:如果失败，具体失败原因 DIndex:原始数据行号, MsgCode:等于1，会话丢失，重新登录；等于0，其他情况 |

1）SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）：

string sJson = "{\"CreateOrgId\":0,\"Numbers\":[\"PRE002\",\"PRE006\"]}";

var result = client.Execute<string>("Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.Audit",

new object[] { "BD\_Currency", sJson });

1. 无引用组件示例（不引用金蝶的组件）：

HttpClient httpClient = new HttpClient();

httpClient.Url = "http://192.168.66.60/k3cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.Audit.common.kdsvc";

List<object> Parameters = new List<object>();

//业务对象Id

String formid = "BD\_Currency";//币别为例

Parameters.Add(formid);

//Json字串

string data = "{\"CreateOrgId\":0,\"Numbers\":[\"PRE002\",\"PRE006\"]}";

Parameters.Add(data);

httpClient.Content = JsonConvert.SerializeObject(Parameters);

var result = httpClient.AsyncRequest();

### 反审核表单数据接口

服务地址：

[http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.UnAudit.common.kdsvc](http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.View.common.kdsvc)

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| formid | 表单Id，格式参考："BD\_Currency"//表示币别 | 必须 |
| data | 数据包，Number表示单据编号或者基础资料编码，Id为主键，CreateOrgId为创建组织Id。数据格式参考以币别为例："{\"CreateOrgId\":0,\"UseOrgId\":0,\"Numbers\":[\"PRE002\",\"PRE006\"]}" | CreateOrgId：创建组织Id,非必须  UseOrgId：使用组织Id,非必须 Numbers：必须 |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| ResponseStatus | 操作状态 {"ResponseStatus":{"ErrorCode":"",  "IsSuccess":false,  "Errors":[{"FieldName":"",  "Message":"","DIndex":0}],  "SuccessEntitys":[{"Id":"","Number":"","DIndex":0}],  "SuccessMessages":[{"FieldName":"","Message":"","DIndex":0}],  “MsgCode”:0} | IsSuccess:操作是否成功， Errors:如果失败，具体失败原因 DIndex:原始数据行号, MsgCode:等于1，会话丢失，重新登录；等于0，其他情况 |

1）SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）：

string sJson = "{\"CreateOrgId\":0,\"Numbers\":[\"PRE002\",\"PRE006\"]}";

var result = client.Execute<string>("Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.UnAudit",new object[] { "BD\_Currency", sJson });

1. 无引用组件示例（不引用金蝶的组件）：

HttpClient httpClient = new HttpClient();

httpClient.Url = "http://192.168.66.60/k3cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.UnAudit.common.kdsvc";

List<object> Parameters = new List<object>();

//业务对象Id

String formid = "BD\_Currency";//币别为例

Parameters.Add(formid);

//Json字串

string data = "{\"CreateOrgId\":0,\"Numbers\":[\"PRE002\",\"PRE006\"]}";

Parameters.Add(data);

httpClient.Content = JsonConvert.SerializeObject(Parameters);

var result = httpClient.AsyncRequest();

### 删除表单数据接口

服务地址：

[http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.Delete.common.kdsvc](http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.View.common.kdsvc)

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| formid | 表单Id，格式参考："BD\_Currency"//表示币别 | 必须 |
| data | 数据包，Number表示单据编号或者基础资料编码，Id为主键，CreateOrgId为创建组织Id。数据格式参考以币别为例："{\"CreateOrgId\":0,\"UseOrgId\":0,\"Numbers\":[\"PRE002\",\"PRE006\"]}" | CreateOrgId：创建组织Id,非必须  UseOrgId：使用组织Id,非必须 Numbers：必须 |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| ResponseStatus | 操作状态 {"ResponseStatus":{"ErrorCode":"",  "IsSuccess":false,  "Errors":[{"FieldName":"",  "Message":"","DIndex":0}],  "SuccessEntitys":[{"Id":"","Number":"","DIndex":0}],  "SuccessMessages":[{"FieldName":"","Message":"","DIndex":0}],  “MsgCode”:0} | IsSuccess:操作是否成功， Errors:如果失败，具体失败原因 DIndex:原始数据行号,  MsgCode:等于1，会话丢失，重新登录；等于0，其他情况 |

1）SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）：

string sJson = "{\"CreateOrgId\":0,\"Numbers\":[\"PRE002\",\"PRE006\"]}";

var result = client.Execute<string>("Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.Delete",

new object[] { "BD\_Currency", sJson });

1. 无引用组件示例（不引用金蝶的组件）：

HttpClient httpClient = new HttpClient();

httpClient.Url = "http://192.168.66.60/k3cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.Delete.common.kdsvc";

List<object> Parameters = new List<object>();

//业务对象Id

String formid = "BD\_Currency";//币别为例

Parameters.Add(formid);

//Json字串

string data = "{\"CreateOrgId\":0,\"Numbers\":[\"PRE002\",\"PRE006\"]}";

Parameters.Add(data);

httpClient.Content = JsonConvert.SerializeObject(Parameters);

var result = httpClient.AsyncRequest();

### 表单数据查询接口

服务地址：

[http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.ExecuteBillQuery.common.kdsvc](http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.View.common.kdsvc)

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| data | 数据包格式参考：  "{\"FormId\":\"PUR\_PurchaseOrder\",\"TopRowCount\":0,\"Limit\":10,\"StartRow\":0,\"FilterString\":\"FMaterialId.FNumber='HG\_TEST'\",\"OrderString\":\"FID ASC\",\"FieldKeys\":\"FID,FSupplierId,FMaterialId,FMaterialId.FNumber,FMaterialName\"}" | FormId：查询表单元数据唯一标识  TopRowCount： 最多允许查询的数量，0或者不要此属性表示不限制  Limit：分页取数每页允许获取的数据，最大不能超过2000  StartRow：分页取数开始行索引，从0开始，例如每页10行数据，第2页开始是10，第3页开始是20  FilterString：过滤条件  OrderString：排序条件  FieldKeys：待查询表单的字段列表 |

返回参数：

调用参考：

1）SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）：

object[] paramInfo = new object[]

{

"{\"FormId\":\"PUR\_PurchaseOrder\","+// 采购订单formid

"\"TopRowCount\":0,"+// 最多允许查询的数量，0或者不要此属性表示不限制

"\"Limit\":10,"+// 分页取数每页允许获取的数据，最大不能超过2000

"\"StartRow\":0,"+// 分页取数开始行索引，从0开始，例如每页10行数据，第2页开始是10，第3页开始是20

"\"FilterString\":\"FMaterialId.FNumber='HG\_TEST'\","+// 过滤条件

"\"OrderString\":\"FID ASC\","+// 排序条件 "\"FieldKeys\":\"FID,FSupplierId,FMaterialId,FMaterialId.FNumber,FMaterialName\"}"// 获取采购订单数据参数，内码，供应商id，物料id,物料编码，物料名称

};

//调用查询接口

List<List<object>> ret = client.Execute<List<List<object>>>("Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.ExecuteBillQuery", paramInfo);

if (ret != null && ret.Count > 0)

{

// TODO：

}

1. 无引用组件示例（不引用金蝶的组件）：

HttpClient httpClient = new HttpClient();

httpClient.Url = "http://192.168.66.60/k3cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.ExecuteBillQuery.common.kdsvc";

List<object> Parameters = new List<object>();

//Json字串

string data = "{\"FormId\":\"PUR\_PurchaseOrder\",\"TopRowCount\":0,\"Limit\":10,\"StartRow\":0,\"FilterString\":\"FMaterialId.FNumber='HG\_TEST'\",\"OrderString\":\"FID ASC\",\"FieldKeys\":\"FID,FSupplierId,FMaterialId,FMaterialId.FNumber,FMaterialName\"}";

Parameters.Add(data);

httpClient.Content = JsonConvert.SerializeObject(Parameters);

var result = httpClient.AsyncRequest();

1. 注意事项
2. 查询接口一定要有查询条件，并限定取数范围，如果获取数据范围太大，接口可能超时；
3. 默认一次最多查询2000条数据，最大可以调整到10000；
4. 查询的字段数量要进行限制，尤其基础资料字段，字段列表中基础资料字段不要超过30个，包括基础资料多语言字段一起，例如名称。如果超过可能引发性能问题。

### 自定义WebAPI接口

服务地址：

[http://ServerIp/K3Cloud/接口命名空间.接口实现类名.方法,组件名.common.kdsvc](http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.View.common.kdsvc)

例如：

"http://192.168.66.60/k3cloud/ApiServiceTest.AAA.CustomBusinessService.ExecuteService,ApiServiceTest.AAA.common.kdsvc"

需要注意的是，自定义webapi接口服务地址中比其他标准接口地址多了一个组件名，如果需要和其他其他标准接口保持一致，则二开自定义的webapi接口命名空间必须以.ServicesStub结尾，例如以下示例自定义接口则其服务地址可以参考如下格式：

"http://192.168.66.60/k3cloud/Kingdee.K3Erp.WebAPI.ServiceExtend.ServicesStub.CustomBusinessService ExecuteService.common.kdsvc"

/\*

\* 服务需要引用组件如下

\* Kingdee.BOS.dll、

Kingdee.BOS.ServiceFacade.KDServiceFx.dll、

Kingdee.BOS.WebApi.ServicesStub.dll

\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using Kingdee.BOS.ServiceFacade.KDServiceFx;

using Kingdee.BOS.WebApi.ServicesStub;

using Kingdee.BOS.JSON;

namespace Kingdee.K3Erp.WebAPI.ServiceExtend.ServicesStub

{

public class CustomBusinessService : AbstractWebApiBusinessService

{

public CustomBusinessService(KDServiceContext context)

: base(context)

{ }

public JSONArray ExecuteService(string parameter)

{

JSONArray jsonArray = new JSONArray();

jsonArray.Add("Hello World");

return jsonArray;

}

}

}

接口参数：

自定义。

返回参数：

自定义。

调用参考：

1. SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）：

// 此接口为K3Cloud自定义接口，命名控件及类名仅供参考。

using Kingdee.BOS.Core.Metadata;

using Kingdee.BOS.Core.SqlBuilder;

using Kingdee.BOS.JSON;

using Kingdee.BOS.Orm.DataEntity;

using Kingdee.BOS.Orm.Metadata.DataEntity;

using Kingdee.BOS.ServiceHelper;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ApiServiceTest.AAA

{

/// <summary>

///

/// </summary>

public class CustomBusinessService : Kingdee.BOS.WebApi.ServicesStub.AbstractWebApiBusinessService

{

public CustomBusinessService(Kingdee.BOS.ServiceFacade.KDServiceFx.KDServiceContext context)

: base(context)

{ }

/// <summary>

/// 执行服务，根据参数来处理自己的业务并返回结果

/// </summary>

/// <param name="ctx"></param>

/// <param name="parameter"></param>

/// <returns></returns>

public JSONArray ExecuteService(string parameter)

{

// TODO：在此方法中处理业务，可以引用Kingdee现有接口，根据传入的参数来实现自己的业务逻辑，

// 例如查询数据、单据转换、甚至数据同步等都可以

// 以下为查询币别信息的示例代码

// 构造查询参数

QueryBuilderParemeter para = new QueryBuilderParemeter()

{

// 查询表单，测试用例中直接传入了“BD\_Currency”

FormId = parameter,

// 需要查询币别主键、币别名称等信息，此处用字段的标识来构造

SelectItems = SelectorItemInfo.CreateItems("FCURRENCYID,FName"),

};

// 调用查询接口

DynamicObjectCollection list = QueryServiceHelper.GetDynamicObjectCollection(this.KDContext.Session.AppContext, para);

// 返回结果类型可以自定义

// 如果第三方系统不想引用BOS其他组件的，可以使用json字符串、List、Dictionary等类型做为返回结果

// 此处示例代码使用 BOS定义的类型JSONArray

JSONArray jsonArray = new JSONArray();

foreach (DynamicObject dynamicObject in list)

{

jsonArray.Add(ConvertDynamicObject2Json(dynamicObject));

}

return jsonArray;

}

/// <summary>

/// 自定义服务接口

/// </summary>

/// <param name="parameter"></param>

/// <param name="formId"></param>

/// <returns></returns>

public bool UpdateData(string parameter,string formId)

{

return true;

}

private JSONObject ConvertDynamicObject2Json(DynamicObject dynamicObject)

{

DynamicPropertyCollection dynamicPropertyCollection = dynamicObject.DynamicObjectType.Properties;

JSONObject obj = new JSONObject();

foreach (DynamicProperty property in dynamicPropertyCollection)

{

obj.Put(property.Name, dynamicObject[property.Name]);

}

return obj;

}

}

}

var responseOut = client.Execute<JSONArray>("ApiServiceTest.AAA.CustomBusinessService.ExecuteService,ApiServiceTest.AAA", new object[] { "BD\_Currency" });

1. 无引用组件示例（不引用金蝶的组件）：

HttpClient httpClient = new HttpClient();

httpClient.Url = string.Concat("http://192.168.66.60/k3cloud/ApiServiceTest.AAA.CustomBusinessService.ExecuteService,ApiServiceTest.AAA.common.kdsvc");

List<object> Parameters = new List<object>();

Parameters.Add("BD\_Currency");

httpClient.Content = JsonConvert.SerializeObject(Parameters);

var responseOut = httpClient.AsyncRequest();

### 登录验证接口带踢人功能

服务地址：

<http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.AuthService.ValidateUser2.common.kdsvc>

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| acctID | 账套Id，从管理中心数据库查询获得 参考查询：select FDATACENTERID from T\_BAS\_DATACENTER | 必须 |
| username | 用户登陆名 | 必须 |
| password | 密码 | 必须 |
| isKickOff | 是否踢人，true:相同客户端会相互踢人 | 必须 |
| lcid | 语言id,选择哪种语言访问，参考：中文2052，英文1033，繁体3076 | 非必须，引用SDK组件辅助类调用则必须 |

返回参数，示例代码参考5.1.1登录验证接口。

### 暂存表单数据接口

服务地址：

<http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.Draft.common.kdsvc>

接口说明：

请参考保存接口。

### 分配表单数据接口

服务地址：

<http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.Allocate.common.kdsvc>

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| formid | 表单Id，格式参考：" BD\_MATERIAL "//表示物料 | 必录 |
| data | 数据包  "{\"PkIds\":\"id1,id2...\", \"TOrgIds\":\"tId1,tId2…\"  }" | PkIds：必须,被分配的单据主键Id集合. TOrgIds：必须，目标组织Id集合 |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| ResponseStatus | 操作状态 {"ResponseStatus":{"ErrorCode":"",  "IsSuccess":false,  "Errors":[{"FieldName":"",  "Message":"","DIndex":0}],  "SuccessEntitys":[{"Id":"","Number":"","DIndex":0}],  "SuccessMessages":[{"FieldName":"","Message":"","DIndex":0}],”MsgCode”:0} | IsSuccess:操作是否成功， Errors:如果失败，具体失败原因 DIndex:原始数据行号,  MsgCode:等于1，会话丢失，重新登录；等于0，其他情况 |

调用参考：

1）SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）：

//业务对象Id

string sFormId = " BD\_MATERIAL"; //物料

//Data数据

string sData = "{\"PkIds\":\"0\",\"TOrgIds\":\"\"} ";

//调用分配接口

K3CloudApiClient client = new K3CloudApiClient("http://localhost:1200/");

client

var ret = client.Allocate(sFormId, sData);

### 下推接口

服务地址：

<http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.Push.common.kdsvc>

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| formid | 表单Id，格式参考：" BD\_MATERIAL "//表示物料 | 必录 |
| data | 数据包  {\"Ids\":\"\",  \"Numbers\":[],  \"EntryIds\":\"\",  \"RuleId\":\"\",  \"TargetBillTypeId\":\"\",  \"TargetOrgId\":\"0\",  \"TargetFormId\":\"\",  \"IsEnableDefaultRule\":\"false\",  \"CustomParams\":{}} | 1.Ids：单据内码，字符串类型，格式"Id1,Id2,..."（逗号分开，未传Numbers，则必录）  2.Numbers：单据编号集合，数组类型，格式["billNo1","billNo1",...]，数组类型（未传Ids,则必录）  3.EntryIds: 分录内码集合，逗号分隔（分录下推时必录）,注(按分录下推时，单据内码和编码不需要填,否则按整单下推) 3.RuleId：转换规则内码，字符串类型，格式"RuleId"（未启用默认转换规则时，则必录）  4.TargetBillTypeId：目标单据类型内码，字符串类型，格式"Id1"（非必录）  5.TargetOrgId：目标组织内码，整数类型，格式"Id1"（非必录）  6.TargetFormId：目标单据FormId，字符串类型，格式:"FormId"（启用默认转换规则时，则必录）  7.IsEnableDefaultRule：是否启用默认转换规则，布尔类型，默认值False，格式"false"（非必录）  8.CustomParams：自定义参数，字典类型，格式"{key1:value1,key2:value2,...}"（非必录） |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| ResponseStatus | 操作状态 {"ResponseStatus":{"ErrorCode":"",  "IsSuccess":false,  "Errors":[{"FieldName":"",  "Message":"","DIndex":0}],  "SuccessEntitys":[{"Id":"","Number":"","DIndex":0}],  "SuccessMessages":[{"FieldName":"","Message":"","DIndex":0}],  “MsgCode”:0} | 返回保存成功后的下游单据内码和编号；  IsSuccess:操作是否成功， Errors:如果失败，具体失败原因 DIndex:原始数据行号,  MsgCode:等于1，会话丢失，重新登录；等于0，其他情况 |

调用参考：

1）SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）：

//业务对象 FormId

string sFormId = " SAL\_SaleOrder"; //销售订单

//Data数据

string sData = "{\"Ids\":\"\",\"Numbers\":[],\"RuleId\":\"\",\"TargetBillTypeId\":\"\",\"TargetOrgId\":\"0\",\"TargetFormId\":\"\",\"IsEnableDefaultRule\":\"false\",\"CustomParams\":{}} ";

//调用分配接口

K3CloudApiClient client = new K3CloudApiClient("http://localhost:1200/");

client

var ret = client.Push(sFormId, sData);

### 分组保存接口

服务地址：

<http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.GroupSave.common.kdsvc>

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| formid | 表单Id，格式参考：" BD\_MATERIAL "//表示物料 | 必录 |
| data | 数据包  {\" GroupFieldKey\":\"\",  \" FParentId\":,  \" FNumber \":\"\",  \" FName \":\"\",  \" FDescription \":\"\"} | 1. GroupFieldKey：分组字段Key，字符串类型，注(不填时取默认，无默认，取第一个分组)  2. FParentId：父分组内码，整型(非必录)  3. FNumber：分组编码，字符串类型(必录)，注（必须唯一）  4. FName：分组名，字符串类型(必录)  5. FDescription：分组描述，字符串类型(非必录) |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| ResponseStatus | 操作状态 {"ResponseStatus":{"ErrorCode":"",  "IsSuccess":false,  "Errors":[{"FieldName":"",  "Message":"","DIndex":0}],  "SuccessEntitys":[{"Id":"","Number":"","DIndex":0}],  "SuccessMessages":[{"FieldName":"","Message":"","DIndex":0}]”MsgCode”:0} | 返回保存成功后的下游单据内码和编号；  IsSuccess:操作是否成功， Errors:如果失败，具体失败原因 DIndex:原始数据行号,  MsgCode:等于1，会话丢失，重新登录；等于0，其他情况 |

### 弹性域保存接口

服务地址：

<http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.FlexSave.common.kdsvc>

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| formid | 表单Id，格式参考：" BD\_MATERIAL "//表示物料 | 必录 |
| data | 数据包 ,Json格式，例如：  var json = "{"Model":  [  {"FF100001":{"FNumber":"A"},  "FF100002":{"FNumber":"Japen"},"FF100003":22}  ]  }"; | 1. Model：数据包集合，数组类型  2. FF100001：维度，{"FNumber":"A"}维度值 |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| ResponseStatus | 操作状态 {"ResponseStatus":{"ErrorCode":"",  "IsSuccess":false,  "Errors":[{"FieldName":"",  "Message":"","DIndex":0}],  "SuccessEntitys":[{"Id":"","Number":"","DIndex":0}],  "SuccessMessages":[{"FieldName":"","Message":"","DIndex":0}],  “MsgCode”:0} | 返回保存成功后的下游单据内码和编号；  IsSuccess:操作是否成功， Errors:如果失败，具体失败原因 DIndex:原始数据行号, MsgCode:等于1，会话丢失，重新登录；等于0，其他情况 |

### 发送消息接口

服务地址：

<http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.SendMsg.common.kdsvc>

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| data | 数据包 ,Json格式，例如：  {\"Model\":[{\"FTitle\":\"FTitleff\",\"FContent\":\"FConteffb2\",\"FReceivers\":\"110\",\"FType\":\"0\"}," +"{\"FTitle\":\"FTitlffa22\",\"FContent\":\"FContentff33\",\"FReceivers\":\"110\"}]}" | 1. Model：数据包集合，数组类型  2. FTitle：消息标题  3. FContent：消息内容  4. FReceivers：消息接收者  5. FType：消息类型 |

### 登出接口

服务地址：

<http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.AuthService.Logout.common.kdsvc>

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
|  | 无 | 当前请求会话的退出 |

返回结果：true/false 成功or失败

### 通用操作接口

服务地址：

<http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.ExecuteOperation.common.kdsvc>

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| formid | 表单Id，格式参考：" BD\_MATERIAL "//表示物料 | 必录 |
| opNumber | 操作编码，字符串类型 | 必录 |
| data | 1.Parameters：参数集合  2.Numbers：单据编码集合  3.Ids：单据内码集合 4.Model：表单数据包 | 1.Parameters：参数集合，字符串类型（非必录） 注（其值传给操作选项）  2.Numbers：单据编码集合，数组类型，格式：[No1,No2,...]（使用编码时必录）  3.Ids：单据内码集合，字符串类型，格式："Id1,Id2,..."（使用内码时必录）  4.Model：表单数据包，JSON类型（使用数据包时必录） |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| ResponseStatus | 操作状态 {"ResponseStatus":{"ErrorCode":"",  "IsSuccess":false,  "Errors":[{"FieldName":"",  "Message":"","DIndex":0}],  "SuccessEntitys":[{"Id":"","Number":"","DIndex":0}],  "SuccessMessages":[{"FieldName":"","Message":"","DIndex":0}]”MsgCode”:0} | 返回保存成功后的下游单据内码和编号；  IsSuccess:操作是否成功， Errors:如果失败，具体失败原因 DIndex:原始数据行号,  MsgCode:等于1，会话丢失，重新登录；等于0，其他情况 |

### 切换上下文默认组织接口

服务地址：

<http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.SwitchOrg.common.kdsvc>

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| OrgNumber | 组织代码 | 必录 |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| ResponseStatus | 操作状态 {"ResponseStatus": {  "IsSuccess": true,  "Errors": [],  "SuccessEntitys": [  {  "Id": 1,  "Number": "100",  "DIndex": 0  }  ],  "SuccessMessages": [],  "MsgCode": 0  }} | IsSuccess:操作是否成功；  Errors:如果失败，具体失败原因；  Number:组织编号；  MsgCode:参考常见错误代码 ； |

### 工作流审批接口

服务地址：

<http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.WorkflowAudit.common.kdsvc>

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| data | 数据包格式参考：  "{"FormId":\"PUR\_PurchaseOrder\","Ids":[],"Numbers":[\"PRE002\",\"PRE006\"],"UserId":135880,"UserName":"","ApprovalType":1,"ActionResultId":"","PostId":0,"PostNumber":""}" | FormId：查询表单元数据唯一标识  Ids和Numbers： 单据的内码和单据的编码，两者选其一  UserId：审批人用户Id，整型UserName：用户名称FilterString：过滤条件  ApprovalType：审批类型，整型（1：审批通过；2：驳回；3：终止）ActionResultId：审批项Id(跟审批类型两者选其一)  PostId：岗位Id  PostNumber：岗位编码(跟岗位Id两者选其一) |

返回参数：

调用参考：

1）SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）：

object[] paramInfo = new object[]

{

"{\"FormId\":\"PUR\_PurchaseOrder\","+

"\"Ids\":[],\"Numbers\":[\"PRE002\",\"PRE006\"],"+

"\"UserId\":135880,"+

"\"ApprovalType\":1,"+

"\"PostId\":12680"

};

var result = client.Execute<String>("Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.WorkflowAudit", paramInfo);

2）无引用组件示例（不引用金蝶的组件）：

HttpClient httpClient = new HttpClient();

httpClient.Url = "http://192.168.66.60/k3cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.WorkflowAudit.common.kdsvc";

object[] paramInfo = new object[]

{

"{\"FormId\":\"PUR\_PurchaseOrder\","+

"\"Ids\":[],\"Numbers\":[\"PRE002\",\"PRE006\"],"+

"\"UserId\":135880,"+

"\"ApprovalType\":1,"+

"\"PostId\":12680"

};

httpClient.Content = JsonConvert.SerializeObject(paramInfo);

var result = httpClient.AsyncRequest();

### 简单账表查询接口

服务地址：

<http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.GetSysReportData.common.kdsvc>

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| formid | 表单Id，格式参考："BD\_HS\_INOUTSTOCKSUMMARYRPT"//存货收发存汇总表 | 必须 |
| data | 数据包格式参考：  "{\"FieldKeys\":\"FMATERIALBASEID,FMATERIALNAME,FMATERIALGROUP,FSTOCKId,FACCTGRANGENAME,FRECEIVEPrice\",\"SchemeId\":\"\",\"StartRow\":0,\"Limit\":10,\"Model\":{\"FACCTGSYSTEMID\":{\"FNumber\":\"001\"},\"FACCTGORGID\":{\"FNumber\":\"103\"},\"FACCTPOLICYID\":{\"FNumber\":\"002\"},\"FYear\":\"2013\",\"FPeriod\":\"12\",\"FENDYEAR\":\"2024\",\"FEndPeriod\":\"8\",\"FCOMBOTotalType\":\"0\",\"FDimType\":\"1\"}}" | FieldKeys：需查询的字段key集合，字符串类型，格式："key1,key2,..."  SchemeId： 过滤方案内码，字符串类型  StartRow：开始行索引，整型（非必录）  Limit：最大行数，整型，不能超过10000（非必录）  IsVerifyBaseDataField：是否验证所有的基础资料有效性，布尔类，默认true（非必录）  Model：表单数据包(过滤方案中的快捷过滤条件)，JSON类型（必录） |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| Result | {"IsSuccess": true,"RowCount": 0,"Rows": [ ] } | IsSuccess:操作是否成功， RowCount:总行数 Rows:返回的报表数据 |

调用参考：

1）SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）：

object[] paramInfo = new object[]

{

"{\"FormId\":\"PUR\_PurchaseOrder\","+

"\"Ids\":[],\"Numbers\":[\"PRE002\",\"PRE006\"],"+

"\"UserId\":135880,"+

"\"ApprovalType\":1,"+

"\"PostId\":12680"

};

var result = client.Execute<String>("Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.GetSysReportData", paramInfo);

2）无引用组件示例（不引用金蝶的组件）：

HttpClient httpClient = new HttpClient();

httpClient.Url = "http://192.168.66.60/k3cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.GetSysReportData.common.kdsvc";

object[] paramInfo = new object[]

{

"{\"FormId\":\"PUR\_PurchaseOrder\","+

"\"Ids\":[],\"Numbers\":[\"PRE002\",\"PRE006\"],"+

"\"UserId\":135880,"+

"\"ApprovalType\":1,"+

"\"PostId\":12680"

};

httpClient.Content = JsonConvert.SerializeObject(paramInfo);

var result = httpClient.AsyncRequest();

3）注意事项

1. 简单账表可能存在动态字段，FieldKeys指定待查字段时，须确保当前查询条件的结果集中包含待查字段。可以使用相同的查询条件在报表查询界面做下验证，确保查询结果包含待查字段；
2. 默认一次最多查询2000条数据，最大可以调整到10000；
3. 查询的频率必须进行控制，因为账表查询使用临时表，查询频率过高会导致系统频繁创建和删除临时表，可能会引发性能问题。

### 附件上传接口(上传附件并绑定单据)

服务地址：

<http://ServerIP/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.AttachmentUpLoad.common.kdsvc>

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| data | 数据包格式参考：  " {\"FileName\": \"0616.txt\",\"FormId\": \"PUR\_PurchaseOrder\",\"IsLast\": true,\"InterId\": \"100723\",\"BillNO\": \"CGDD000198\",\"AliasFileName\": \"test\",\"SendByte\": \"文件字节数组转base64后的字符串\"}" | FileNamer：文件名，必填  FormId： 表单标识，必填  IsLast：是否最后一次上传，必填  InterId：单据内码，必填  Entrykey：单据体标识，上传单据体附件时候填写所述单据体的标识  EntryinterId：分录内码，如果是单据头附件，要么不填，要么填-1 BillNO：单据编号，必填  AliasFileNamer：附件别名  FileId：文件 id;如果分多次上传，首次上传后必填  SendByte：Base64 后的文件字节流，必填 |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| Result | ResponseStatus":{"IsSuccess":true,"Errors":[],"SuccessEntitys":[{"Id":470257,"Number":null,"DIndex":0}],"SuccessMessages":[],"MsgCode":0},"FileId":"23d8a447a9254a788f2d12a499898755","Message":""} | IsSuccess:操作是否成功， FileId:文件id Message:错误提示信息  SuccessEntitys.Id:附件内码 |

调用参考：

1）SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）：

object[] paramInfo = new object[] {

"{\" FileName\":\"Test.txt\","+

"\"FormId\":\"PUR\_PurchaseOrder\","+

"\"IsLast\":true,"+

"\"InterId\":\"100123\","+

"\"Entrykey\":\"FPOOrderEntry\","+

"\"EntryinterId\":\"101300\","+

"\"BillNO\":\"CGDD10001\","+

"\"AliasFileName\":\"test2\","+

"\"SendByte\":\"77u/MTIzNA==\""

};

var result = client.Execute<String>("Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.AttachmentUpLoad", paramInfo);

2）无引用组件示例（不引用金蝶的组件）：

HttpClient httpClient = new HttpClient();

httpClient.Url = "http://192.168.66.60/k3cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.AttachmentUpLoad.common.kdsvc";

object[] paramInfo = new object[] {

"{\" FileName\":\"Test.txt\","+

"\"FormId\":\"PUR\_PurchaseOrder\","+

"\"IsLast\":true,"+

"\"InterId\":\"100123\","+

"\"Entrykey\":\"FPOOrderEntry\","+

"\"EntryinterId\":\"101300\","+

"\"BillNO\":\"CGDD10001\","+

"\"AliasFileName\":\"test2\","+

"\"SendByte\":\"77u/MTIzNA==\""

};

httpClient.Content = JsonConvert.SerializeObject(paramInfo);

var result = httpClient.AsyncRequest();

### 附件下载接口

服务地址：

<http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.AttachmentDownLoad.common.kdsvc>

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| data | 数据包格式参考：  "{"FileId":\"c671ad15ef884994813e0fc6e0f0f79f\","StartIndex":0 }" | FileId：文件id，必填  StartIndex：下载起始位置，默认为0 |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| Result | {"ResponseStatus":{"IsSuccess":true,"Errors":[],"SuccessEntitys":[],"SuccessMessages":[],"MsgCode":0},"StartIndex":4194304,"IsLast":true,"FileSize":6378,"FileName":"1104.txt","FilePart":"文件字节数组转base64后的字符串","Message":""} | IsSuccess:操作是否成功， StartIndex: 下载起始位置，第一次为 0，IsLast 不为 true 时再次调用下载接口时使用，记录下载文件位置;  IsLast: 为 true 则最后一次，表示下载完成;不为 true时需再次调用下载接;  FileSize:文件大小，  FileName:文件名，  FilePart: 返回 Base64 后的文件字节流， Message:错误提示信息 |

调用参考：

1）SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）：

object[] paramInfo = new object[]

{

"{\" FileID\":\" 54ab21207f3b44d782424cd21902076f \","+

"\" StartIndex\":0"

};

var result = client.Execute<String>("Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.AttachmentDownLoad", paramInfo);

var base64Str = JObject.Parse(result)["Result"]["FilePart"].Value<string>();

var bytes = Convert.FromBase64String(base64Str);

Stream stream = new MemoryStream(bytes);

if (stream.CanRead)

{

using (FileStream fs = new FileStream(@"D:\a99999999.xlsx", FileMode.Create, FileAccess.Write, FileShare.None))

{

const int bufferLength = 4096;

byte[] myBuffer = new byte[bufferLength];

int count;

while ((count = stream.Read(myBuffer, 0, bufferLength)) > 0)

{

fs.Write(myBuffer, 0, count);

}

fs.Close();

stream.Close();

}

}

2）无引用组件示例（不引用金蝶的组件）：

HttpClient httpClient = new HttpClient();

httpClient.Url = "http://192.168.66.60/k3cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService. AttachmentDownLoad.common.kdsvc";

object[] paramInfo = new object[]

{

"{\" FileID\":\" 54ab21207f3b44d782424cd21902076f \","+

"\" StartIndex\":0"

};

httpClient.Content = JsonConvert.SerializeObject(paramInfo);

var result = httpClient.AsyncRequest();

var base64Str = JObject.Parse(result)["Result"]["FilePart"].Value<string>();

var bytes = Convert.FromBase64String(base64Str);

Stream stream = new MemoryStream(bytes);

if (stream.CanRead)

{

using (FileStream fs = new FileStream(@"D:\a99999999.xlsx", FileMode.Create, FileAccess.Write, FileShare.None))

{

const int bufferLength = 4096;

byte[] myBuffer = new byte[bufferLength];

int count;

while ((count = stream.Read(myBuffer, 0, bufferLength)) > 0)

{

fs.Write(myBuffer, 0, count);

}

fs.Close();

stream.Close();

}

}

### 文件上传接口(不绑定单据)

服务地址：

[http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService. UploadFile.common.kdsvc](http://ServerIp/K3Cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.WorkflowAudit.common.kdsvc)

接口参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| data | 数据包格式参考：  " {\"FileName\": \"0616.txt\",\"IsLast\": true,\"SendByte\": \"文件字节数组转base64后的字符串\"}" | FileNamer：文件名，必填  IsLast：是否最后一次上传，必填  FileId：文件 id;如果分多次上传，首次上传后必填  SendByte：Base64 后的文件字节流，必填 |

返回参数：

|  |  |  |
| --- | --- | --- |
| 参数列表 | 参数含义 | 备注 |
| Result | {"ResponseStatus":{"IsSuccess":true,"Errors":[],"SuccessEntitys":[],"SuccessMessages":[],"MsgCode":0},"FileId":"8941112c242942e5b18f7e38839530fa","Message":""} | IsSuccess:操作是否成功， FileId:文件id Message:错误提示信息  SuccessEntitys.Id:文件内码 |

调用参考：

1）SDK辅助类示例（引用Kingdee.BOS.WebAPI.Client.dll）：

object[] paramInfo = new object[] {

"{\" FileName\":\"Test.txt\","+

"\"IsLast\":true,"+

"\"SendByte\":\"77u/MTIzNA==\""

};

var result = client.Execute<String>("Kingdee.BOS.WebApi.ServicesStub.DynamicFormService.UpLoadFile", paramInfo);

2）无引用组件示例（不引用金蝶的组件）：

HttpClient httpClient = new HttpClient();

httpClient.Url = "http://192.168.66.60/k3cloud/Kingdee.BOS.WebApi.ServicesStub.DynamicFormService. UpLoadFile.common.kdsvc";

object[] paramInfo = new object[] {

"{\" FileName\":\"Test.txt\","+

"\"IsLast\":true,"+

"\"SendByte\":\"77u/MTIzNA==\""

};

httpClient.Content = JsonConvert.SerializeObject(paramInfo);

var result = httpClient.AsyncRequest();

# 常见错误代码

|  |  |
| --- | --- |
| 错误代码 | 说明 |
| 0 | 默认 |
| 1 | 上下文丢失 |
| 2 | 没有权限 |
| 3 | 操作标识为空 |
| 4 | 异常 |
| 5 | 单据标识为空 |
| 6 | 数据库操作失败 |
| 7 | 许可错误 |
| 8 | 参数错误 |
| 9 | 指定字段/值不存在 |
| 10 | 未找到对应数据 |
| 11 | 验证失败 |
| 12 | 不可操作 |
| 13 | 网控冲突 |
| 14 | 调用限制 |
| 15 | 禁止管理员登录 |