

第一章 基础数据接口

一.1 集成范围

涉及ERP的基本档案包括：组织、部门、人员、用户、客户、供应商、物料分类、计量单位、物料、仓库、税码税率、合同类型、合同分类；

ERP中涉及的操作：新增、修改、删除、启用、停用；

基本档案涉及的控制状态：已经停用的基本档案新增合同时不能继续使用；

一.2 集成方案

一.2.1 请求方式

POST

一.2.2 请求地址

/uapws/rest/databus/queryByCondition

一.2.3 请求参数

参数名	是	必选	类型	说明
	是		Array	参数为数组，支持多个数据类型查询
dataType	是		String	数据类型 由用友提供
ts	是		String	查询时间范围 格式为 yyyy-mm-dd hh24:mi:ss 查询范围为此时间 之后更新/插入的数据

一.2.4 请求示例

```
[  
  
{  
  
  "dataType": "user",  
  
  "ts": "2022-11-08 10:00:00"  
  
},  
  
{  
  
  "dataType": "dept",  
  
  "ts": "2022-11-08 10:00:00"  
  
}  
  
]
```

一.2.5 返回示例

成功

```
{  
  
  "code": 200,  
  
  "message": "query success",  
  
  "success": true,
```

```
"data": {  
  "dept": {  
    "v_dept": []  
  },  
  "user": {  
    "v_user_ext_1": [  
      {  
        "pk_psnjob": "1001",  
        "pk_dept": "1001",  
        "pk_psndoc": "1001",  
        "indutydate": "2022-10-01",  
        "ismainjob": "Y",  
        "pk_org": "0001",  
        "ts": "2022-11-08 11:59:15"  
      },  
      {  
        "pk_psnjob": "1002",  
        "pk_dept": "1002",  
        "pk_psndoc": "1002",  
        "indutydate": "2022-08-01",
```

```
"ismainjob": "Y",

"pk_org": "0002",

"ts": "2022-11-08 13:58:56"

}

],

"v_user": [

{

"code": "001",

"pk_psndoc": "1001",

"name": "张XX",

"pk_org": "0001",

"ts": "2022-11-08 11:59:15"

},

{

"code": "001356",

"pk_psndoc": "1002",

"name": "孙XX",

"pk_org": "0002",

"ts": "2022-11-08 13:58:56"

}
```

```
]
}
}
}
```

失败

```
{
"code": 50003,
"message": "dataType:12 is not support!",
"success": false
}
```

一.2.6 返回参数说明

参数名	必选	类型	说明
code	是	String	操作状态码
message	否	String	返回信息
success	是	Boolean	是否成功
data	是	object	查询结果数据

一.2.7 返回data格式解释

示例：

基础档案名称：□员

档案类型：user

□表：v_user_ext_1：v_档案类型_ext_1

主表：v_user: v_档案类型

一.2.8 数据类型

一.2.9 基本档案字段

参见：[附件1：合同对接内容梳理](#)

第二章 合同接口

二.1 OpenAPI认证

二.1.1 获取token和调用接口

第一步请求获取相应的token，请求的URL路径如下：

```
http://127.0.0.1/nccloud/opm/acc  
cesToken  
?biz_center=1&grant_type=client_credentials&signature=3ca73b3bb506a3  
4059e2bce1ce3bfe128e4e9fb6cefb7325396094f7816a01d9&client_secret=S  
u1s4kk0pQhhgAupsDajkqSFWeWxUEAy78yYh84wTHt1UPyC2ZV3CD7%2BP  
12XB897owyaVFQRJd2g%0D%0AfZcPwkvIxUgq3yrp1PBYxZ1TJ89oLf4Wicn  
%2BsDVAi57pTlsHHZZqQqLow5zdQjNP3Wm04ewszLhu%0D%0AasoViTdsp  
zujiPAmwxY%3D%0D%0A&client_id=wqch
```

其中各个参数的含义如下：

biz_center：访问的nccloud系统的账套code

grant_type : 授权模式 , 此处为client_credentials

client_id : 对应于在第三方应用注册当中的app_id

client_secret : 对应于第三方应用注册当中的app_secret

signature : 请求加签 , 其算法为SHA256Util.getSHA256(client_id + client_secret + pubKey)

其中pubKey为第三方应用注册当中的公钥字段

第二步为最终请求的OpenAPI , 其中URL地址如下所示 :

<http://ip:port/nccloud/api/uapbd/currtype/currtype/queryCurrtypeByCode>

其中在请求头当中包含了相关的参数 , 如下所示 :

access_token : 即上一步当中获取到的token。

ucg_flag : Y

signature : 对请求体进行加签 , 算法如下 : SHA256Util.getSHA256(client_id+requestBody+pubkey)

其中pubKey为第三方应用注册当中的公钥字段。

二.1.2 Java使用示例

二.1.2.1 Java代码

```
package nccloud.api.testcase.base;
```

```
import java.io.BufferedReader;
```

```
import java.io.IOException;
```

```
import java.io.InputStream;
```

```
import java.io.InputStreamReader;

import java.io.OutputStream;

import java.io.UnsupportedEncodingException;

import java.net.HttpURLConnection;

import java.net.URL;

import java.net.URLEncoder;

import java.nio.charset.Charset;

import java.util.HashMap;

import java.util.Map;

import java.util.Properties;


import org.apache.commons.lang3.StringUtils;


import com.google.gson.Gson;


import nccloud.api.test.utils.CompressUtil;

import nccloud.api.test.utils.Decryption;

import nccloud.api.test.utils.Encryption;

import nccloud.api.test.utils.ResultMessageUtil;

import nccloud.api.test.utils.SHA256Util;
```

```
/**
```


- * 1.从resources/config.properties中读取测试api相关的数据
- 2.运行程序，测试查看测试结果

*

*/

```
public class Test {
```

```
private static String client_secret = null;
```

```
private static String pubKey = null;
```

```
private static String client_id = null;
```

```
private static String username = null;
```

```
private static String pwd = null;
```

```
private static String busi_center = null;
```

```
// 获取token方式
```

```
private static String grant_type = null;
```

```
// 服务器ip : port
```

```
private static String baseUrl = null;
```

```
private static String secret_level = null;
```

```
private static String requestBody = null;
```

```
// openapi请求路径
```

```
private static String apiUrl = null;
```

```
public static String token = null;

public static String repeat_check = null;

public static String busi_id = null;


public static void main(String[] args) {

    try {

        // 初始化数据

        init();

        // 请求token

        token = getToken();

        System.out.println("getTokenData:" + token);

        if (token != null) {

            // 测试openapi

            testApi(token);

        }

    } catch (Exception e) {

        e.printStackTrace();

    }

}

/**
```

* 通过refresh_token重新获取token

*

* @param refresh_token

* @return

* @throws UnsupportedOperationException

* @throws Exception

*/

private static String getTokenByRefreshToken(String refresh_token) throws
UnsupportedEncodingException, Exception {

Map<String, String> paramMap = new HashMap<String, String>();

// 密码模式认证

paramMap.put("grant_type", "refresh_token");

// 第三方应用id

paramMap.put("client_id", client_id);

// 第三方应用secret 公钥加密

paramMap.put("client_secret",
URLEncoder.encode(Encryption.pubEncrypt(pubKey, client_secret), "utf-8"));

// 签名

String sign = SHA256Util.getSHA256(client_id + client_secret +
refresh_token + pubKey);

paramMap.put("signature", sign);

```
String url = baseUrl + "/nccloud/opm/accesstoken";

String mediaType = "application/x-www-form-urlencoded";

String token = doPost(url, paramMap, mediaType, null, "");

return token;

}
```

```
private static String getToken() throws Exception {

String token = null;

if ("password".equals(grant_type)) {

// 密码模式

token = getTokenByPWD();

} else if ("client_credentials".equals(grant_type)) {

// 客户端模式

token = getTokenByClient();

} else if ("authorization_code".equals(grant_type)) {

// TODO 页面跳转

// 授权码模式

}

return token;

}
```

```
/**
 * 客户端模式获取token
 *
 * @return
 * @throws Exception
 */
private static String getTokenByClient() throws Exception {
    Map<String, String> paramMap = new HashMap<String, String>();
    // 密码模式认证
    paramMap.put("grant_type", "client_credentials");
    // 第三方应用id
    paramMap.put("client_id", client_id);
    // 第三方应用secret 公钥加密
    paramMap.put("client_secret",
        URLEncoder.encode(Encryption.pubEncrypt(pubKey, client_secret), "utf-8"));
    // 账套编码
    paramMap.put("biz_center", busi_center);
    // // TODO 传递数据源和ncc登录用户
    // paramMap.put("dsname", "TM_0614");
    // paramMap.put("usercode", "1");
}
```

```
// 签名
```

```
String sign = SHA256Util.getSHA256(client_id + client_secret + pubKey);
```

```
paramMap.put("signature", sign);
```

```
String url = baseUrl + "/nccloud/opm/accesstoken";
```

```
String mediaType = "application/x-www-form-urlencoded";
```

```
String token = doPost(url, paramMap, mediaType, null, "");
```

```
return token;
```

```
}
```

```
/**
```

```
 * 密码模式获取token
```

```
 *
```

```
 * @return
```

```
 * @throws Exception
```

```
 */
```

```
@SuppressWarnings("unused")
```

```
private static String getTokenByPWD() throws Exception {
```

```
Map<String, String> paramMap = new HashMap<String, String>();
```

```
// 密码模式认证
```

```
paramMap.put("grant_type", "password");
```

```
// 第三方应用id

paramMap.put("client_id", client_id);

// 第三方应用secret 公钥加密

paramMap.put("client_secret",
URLEncoder.encode(Encryption.pubEncrypt(pubKey, client_secret), "utf-8"));

// ncc用户名

paramMap.put("username", username);

// 密码 公钥加密

paramMap.put("password",
URLEncoder.encode(Encryption.pubEncrypt(pubKey, pwd), "utf-8"));

// 账套编码

paramMap.put("biz_center", busi_center);

// 签名

String sign = SHA256Util.getSHA256(client_id + client_secret + username +
pwd + pubKey);

paramMap.put("signature", sign);

String url = baseUrl + "/nccloud/opm/accesstoken";

String mediaType = "application/x-www-form-urlencoded";

String token = doPost(url, paramMap, mediaType, null, "");

return token;

}
```

