



## 金蝶 EAS

### 凭证引入 “webservice 接口” 说明

文档 创立 金向平

日期：确认 2023-11-18

日期：当前 2023-11-18

版本：拷贝 V1.0

数量： 1

维护 吴进

确认日期：2023-9-8

当前版本：V1.1

维护 吴进

确认日期：2023-11-5

维护日期：2023-11-30

当前版本：V5.4.0

维护 焦旺

确认日期：2023-5-24

当前版本：V7.0.1



日期	作者	版本	参考版本	备注
2023-11-18	金向平	V1.0		
2023-9-8	吴进	V1.1	V1.0	
2023-11-5	吴进	V5.4.0	V1.1	
2023-11-3	焦旺	V5.3.0	V1.1	基于万科版本开发的
2023-5-24	焦旺	V7.0.1	V1.1	修改支持的EAS版本

## 名目

<b>1. EAS公布的 WEBSERVICE 接口 .....</b>	<b>6</b>
1.1 功能规划 .....	6
1.2 EASLOGIN 公布 .....	6
1.3 EASLOGIN 代码调用 .....	7
<b>2.</b>	
2.1 IMPORTVOUCHER (凭证引入) .....	11
2.2 凭证引入原版文档如下 .....	13
<b>3.</b>	
<b>4.</b>	
<b>5.</b>	
<b>6.</b>	
<b>7.</b>	
<b>8.</b>	
<b>9.</b>	
<b>10.</b>	
<b>11.</b>	
<b>12.</b>	
<b>13. 组织机构信息的猎取、关心帐类型的猎取、核算工程的猎取、具体核算工程的猎取、会计科目信息的猎取、科目余额的猎取和会计凭证具体信息的猎取接口 .....</b>	<b>36</b>
<b>14.</b>	
<b>15.</b>	

---

1.	“核算工程的猎取”接口说明.....	43
2.	“具体核算工程数据的猎取”接口说明.....	45
3.	“会计科目信息的猎取”接口说明.....	46
4.	“科目余额的猎取”接口说明.....	50
5.	“核算工程余额的猎取”接口说明.....	53
6.	“依据组织机构代码、会计年度和会计期间猎取会计凭证具体信息”接口说明.....	55

## 凭证引入 “webservice 接口” 说明

### 1. EAS 公布的 webservice 接口



### And now... Some Services

- EASLogin ([wsdl](#))
  - login
  - login
- WSWSEmployee ([wsdl](#))
  - getAddressList
  - importPersonCollection
  - importPersonInfo
  - getAdminPositionByPerson
- WSWSPurOrderFacade ([wsdl](#))
  - ImportData
- AdminService ([wsdl](#))
  - AdminService

#### 1.1 功能规划

- 1、 如需调用凭证接口，必需先要登录EAS，先期需要调用 EASLogin 登录 EAS

#### 1.2 EASLogin 公布

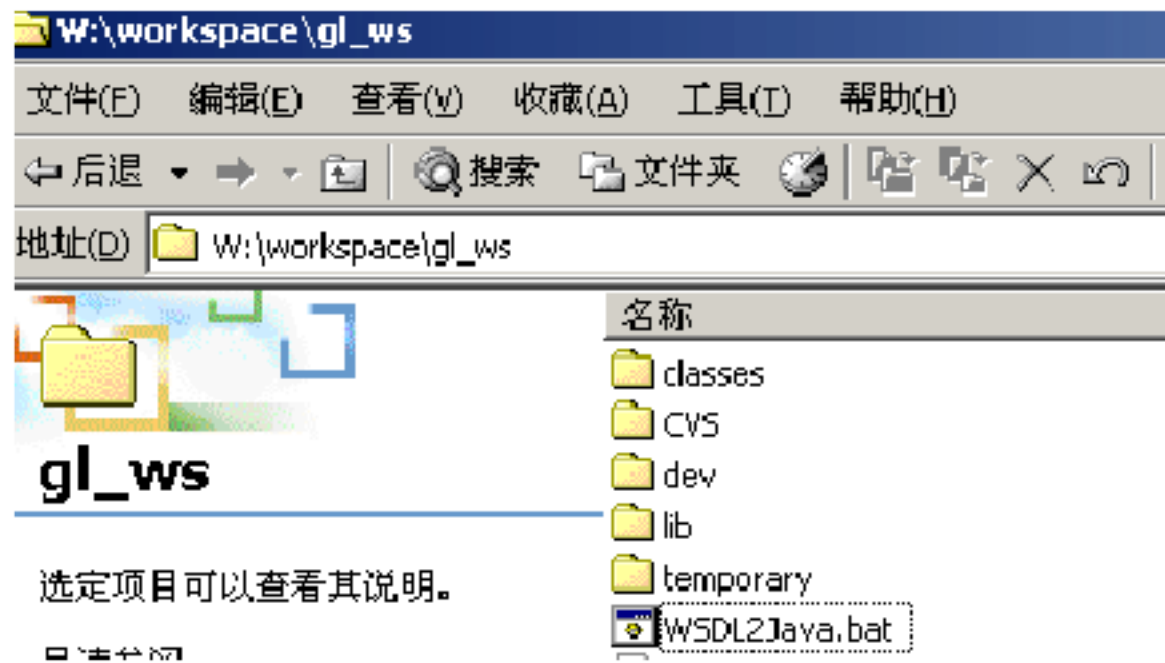
- 1、 公布 EASLogin 客户端
- 2、 点击 EASLogin(wsdl)，猎取其 wsdl 地址



```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions targetNamespace="http://192.168.16.13:6888/ormrpc/services/EASLogin"
  xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="http://192.168.16.13:6888/ormrpc/services/EASLogin"
  xmlns:intf="http://192.168.16.13:6888/ormrpc/services/EASLogin"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns1="urn:client"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:wsdlsoap="http://schemas.xmlsoap.org/soap/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

- 3、制作 bat 文件，利用第三方包，公布登录客户端代码，调用语句如下：

```
java -Djava.ext.dirs=lib org.apache.axis.wsdl.WSDL2Java
: //192.168.16.13:6888/ormrpc/services/EASLogin?wsdl
```



- 4、第三方apache 包

axis.jar	2005-10-19 12:23
axis-ant.jar	2005-10-19 12:23

- 5、代码解释：java Djava.ext.dirs=lib org.apache

lib 是包放置的文件夹名



## 1.1 EASLogin 代码调用

- 1、调用的各所需参数

## 2、所需参数:

- \* @param userName 用户名 String
- \* @param password 密码 String
- \* @param slnName eas String
- \* @param dcName 数据中心 String
- \* @param language 语言 String
- \* @param dbType 数据库类型 int
- \* @param authPattern 验证方式 默认 "BaseDB" ; 其他认证方式 KEY 可从 easAuthPatterns.xml 中猎取 String

## 3、参数说明

- a) 所传参数，如上所列，按挨次传入
- b) 查看 EASLogin(wsdl)，有两个接口，
  - i. 传参 6 个，此方式不提倡使用。为保证原有功能，效劳端最终也会调用传参 7 个的方法，第 7 参数自动调用 "BaseDB "
  - ii. 传参 7 个，如调用此方法，则需要查询 easAuthPatterns.xml 文档
- c) 解决方案: eas
- d) 语言: L2 简体中文 L3 繁体中文
- e) 效劳器地址 + 端口: 组合成 URL <http://192.168.16.13:6888/ormrpc/services/>
- f) 数据库类型: 0 SQL Server 1 Oracle 2 DB2
- g) authPattern 验证方式



- i. 默认 “BaseDB “
- ii. easAuthPatterns.xml 文档路径



- iii. easAuthPatterns.xml 文档说明

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <authPatterns>
3   <authPattern>
4     <name>BaseTrdLtpaToken</name>
5     <displayName>BaseTrdLtpaToken</displayName>
6     <authHandler>com.kingdee.eas.cp.eip.sso.ltpa.LtpaTokenAuthHandler</authHandler>
7     <description>Base third system's Lightweight Third Party Authentication</description>
8   </authPattern>
9   <authPattern>
10    <name>BaseDB</name>
11    <displayName>BaseDB</displayName>
12    <authHandler>com.kingdee.eas.cp.eip.sso.EasDefaultAuthHandler</authHandler>
13    <description> Base Eas user table authentication, is Eas default Authentication</description>
14  </authPattern>
15  <default>BaseDB</default>
16  <scope>session</scope>
17 </authPatterns>
18
```

4、 做一个javaBean 类，把此参数 set 到此javaBean 类中

```
//服务名
ServiceInfo service = new ServiceInfo();

// http://192.168.18.192:6888/ormrpc/services/
service.setServername("http://" + ipAddress + ":" + port + "/ormrpc/services/");

// 解决方案
service.setEas(eas);

// 数据中心
service.setDatacenter(easInstance.getNumber());

// 数据库类型, 0 SQL Server 1 Oracle 2 DB2
service.setType(dbtype.getType());
```

## 5、登录代码

### a) 登录

```
//EAS login
WSContext wsContext = doEASLogin(service);
if(wsContext == null){
    MsgBox.showError(this, "登录失败");
    return ;
}
```

### b) 登录实现

```
protected WSContext doEASLogin(ServiceInfo service) {
    java.net.URL endpoint = null;
    try {
        endpoint = new java.net.URL(service.getServername() + "EASLogin");
    } catch (java.net.MalformedURLException e) {
        e.printStackTrace();
        return null;
    }

    EASLoginProxy proxy = null;
    try {
        proxy = new EASLoginProxyServiceLocator().getEASLogin(endpoint);
    } catch (ServiceException e1) {
        e1.printStackTrace();
        return null;
    }

    try {
        return proxy.login(service.getUser(), service.getPassword(),
            service.getEas(), service.getDatacenter(),
            service.getLang(), service.getType());
    } catch (RemoteException e2) {
        e2.printStackTrace();
    }

    return null;
}
```

## 2. 凭证引入接口

### 2.1 importVoucher (凭证引入)

1、凭证引入接口，两个接口方法都可用，且用法全都

- WSGWebServiceFacade ([wsdl](#))
  - o importAccountBalance
  - o checkVoucher
  - o deleteVoucher
  - o importVoucher
  - o deleteBalance
  - o exploreVoucher
  - o importInitAccountBalance
  - o importAssistBalance
  - o findVoucher
  - o importInitAssistBalance
- WSWSVoucher ([wsdl](#))
  - o importVoucher
  - o importVoucher

2、凭证 webservice 接口生成客户端，可参考 EASLogin 方法

3、凭证引入参数: importVoucher(wsvoucher[], int isVerify, int isCashflow)

- 传参数第1个: WSWSVoucher[] 一般对象数组，见构造EAS凭证字段以及含义说明
- 传参数第2个: isverify
  - 0 不核销，暂存
  - 1 核销，暂存
  - 2 不核销，提交
  - 3 核销，提交
- 传参数第3个: isImpCashflow
  - 是否引入现金流量 0 不引入 1 引入



4、凭证引入代码调用说明

```
private void impVoucher(ActionEvent e) {
    WSWSVoucher[] vouchers = null;
    String server = service.getServername();
    java.net.URL endpoint = null;
    int verify = 0;
    int cashflow = 0;
    // 调用产生凭证的接口协议类
    WSWSVoucherSrvProxyServiceLocator locator = new WSWSVoucherSrvProxyServiceLocator();
    try {
        // 此处构造EAS凭证需要的信息...
        vouchers = getVoucher();
        endpoint = new java.net.URL(server + "WSWSVoucher");
        TypeInfo ve = (TypeInfo) kcbVerify.getSelectedItem();
        TypeInfo ca = (TypeInfo) kcbCashflow.getSelectedItem();
        verify = ve.getType();
        cashflow = ca.getType();

        String[][] ls = locator.getWSWSVoucher(endpoint).importVoucher(vouchers, verify, cashflow);
    }
}
```

## 5、 返回值String[][]

是一个长度为凭证数，宽度是 5 的字符串数据

宽度的定义：[外部系统凭证号](#)，[凭证字](#)，[会计年度](#)，[会计期间](#)，[处理信息](#)，[特别信息](#)，[金蝶凭证号](#)

## 2.1 凭证引入原版文档如下

### 一. “凭证引入”技术方案

其他业务系统数据，传输到EAS系统效劳器，通过凭证引入“webservice接口”，转化为EAS数据。

其他业务系统内的资金流淌信息，实时地、非人工干预的传递到EAS系统，以财务凭证的形式，转化为EAS系统内的财务信息。

### 二. “凭证引入”技术标准

EAS系统供给WebService效劳，以及产生EAS财务凭证的接口，供其他业务系统调用。WebService效劳：金蝶在EAS系统上公布WebService效劳，供给wsdl文件供客户端下载，其他业务系统依据下载的wsdl文件，产生客户端。

EAS财务凭证的接口产生的客户端，分为两局部（java客户端举例）

#### （1）协议类，包名 localhost.bosws.services.WSWSVoucher

协议类用于获得金蝶在EAS系统上公布WebService效劳接口。本用例中为其他业务系统内的资金流淌信息产生EAS财务凭证的接口。

(1) JavaBean类，包名 wsvoucher.client

JavaBean 类供给接口需要的参数。本用例中为EAS 财务凭证需要的信息。

### 三. “凭证导入接口”调用说明

- 接口: importVoucher(wsvoucher[], int isVerify, int isCashflow)
- 传参数第1个: WSWSVoucher[] 一般对象数组，见构造EAS凭证字段以及含义说明
- 传参数第2个: isverify
  - 0 不核销，暂存
  - 1 核销，暂存
  - 2 不核销，提交
  - 3 核销，提交
- 传参数第3个: isImpCashflow
  - 是否引入现金流量 0 不是 1 是

### 四. 构造EAS凭证字段以及含义说明如下:

字段名	类型	中文说明	是否必录	说明	格式
WSWSVoucher 说明：一张凭证占据的行数 = 凭证行*分录行*关心账行					
字体黑色：凭证头字段；深蓝：分录行字段； 橙色：关心账字段；紫罗兰：现金流量					
单元格颜色：灰色 必录项 ； 浅青绿：关心账核算工程（科目多核算工程，最多8个）；浅绿：现金流量					
companyNumber	String	公司编码	是	录入公司编码，必需在 EAS 中有该编码	
bookedDate	String	记账日期	是	凭证的登账日期	2023-11-1
bizDate	String	业务日期	是	业务发生日期	2023-11-1
periodYear	int	会计期间-年	是	必需在EAS 中有	2023
periodNumber	int	会计期间-编码	是	必需在EAS 中有	11
voucherType	String	凭证字（凭证类型）	是	凭证类型中文名称，必需在 EAS 中有	记
attaches	int	附件数量	否		
description	String	参考信息	否		

voucherNumber	String	凭证号	是	凭证唯一区分	1
entrySeq	int	分录行号	是	当前凭证的分录唯一区分	1
voucherAbstract	String	摘要	否	分录摘要	
accountNumber	String	科目	是	分录行科目编码，必需在 EAS 中有	1001.01
currencyNumber	String	币种	是	分录行币别编码，必需在 EAS 中有	001
localRate	double	汇率	否	本位币汇率：没有时默认为 1	1.01
entryDC	int	方向	是	分录行方向：1 借方 -1 贷方	
originalAmount	double	原币金额	是	分录行原币金额	
qty	double	数量	否	分录行数量	
measurement	String	计量单位	否	名称，必需在 EAS 中有	吨
price	double	单价	否	分录行单价	
debitAmount	double	借方金额	是	分录行借方金额：分录方向为借方时必录	
creditAmount	double	贷方金额	是	分录行贷方金额：分录方向为贷方时必录	
creator	String	制单人	是	当前操作人员名称，必需在 EAS 中有	徐晓霞
poster	String	过账人	否	当前操作人员名称，必需在 EAS 中有	
auditor	String	审核人	否	当前操作人员名称，必需在 EAS 中有	
asstSeq	int	关心账行号	否	当前凭证的当前分录的关心账行唯一区分	
bizNumber	String	业务编号	否		

settlementNumber	String	结算方式	否		
settlementType	String	结算号	否		
current	int	核销/挂账	否	默认为 0,	
asstActType1	String	核算工程 1	否	名称, 必需在 EAS 中有。比方: 客户、	客户
asstActNumber1	String	核算对象编码 1	否	编码, 必需在 EAS 中有。比方: 客户 A 的编码	01.01
asstActName1	String	核算对象名称 1	否	名称, 必需在 EAS 中有。比方: 客户 A	客户 A
asstActType2	String	核算工程 2	否	以下: 取决于科目是否核算多个工程, 最多 8 个	
asstActNumber2	String	核算对象编码 2	否		
asstActName2	String	核算对象名称 2	否		
asstActType3	String	核算工程 3	否		
asstActNumber3	String	核算对象编码 3	否		
asstActName3	String	核算对象名称 3	否		
asstActType4	String	核算工程 4	否		
asstActNumber4	String	核算对象编码 4	否		
asstActName4	String	核算对象名称 4	否		
asstActType5	String	核算工程 5	否		
asstActNumber5	String	核算对象编码 5	否		
asstActName5	String	核算对象名称 5	否		
asstActType6	String	核算工程 6	否		
asstActNumber	String	核算对象编码 6	否		



6					
asstActName6	String	核算对象名称 6	否		
asstActType7	String	核算工程 7	否		
asstActNumber7	String	核算对象编码 7	否		
asstActName7	String	核算对象名称 7	否		
asstActType8	String	核算工程 8	否		
asstActNumber8	String	核算对象编码 8	否		
asstActName8	String	核算对象名称 8	否		
itemflag	int	现金流量标记	是	如引入现金流量, 必设置为 1	
oppAccountSeq	int	对方科目分录号	是		
primaryItem	String	主表工程	是	EAS 中必存在的主表工程	
supplyItem	String	附表工程	否		
primaryCoef	int	主表系数	否		
supplyCoef	int	附表系数	否		
cashflowAmountOriginal	double	现金流量原币金额	是		
cashflowAmountLocal	double	现金流量本位币金额	是		
cashflowAmountRpt	double	现金流量报告币金额	是		
cashflowAmountRpt	double	现金流量报告币金额	是		
type	Int	现金流量性质列	否	1、空；2、外部；3、内部	
cashAsstActType1	String	现金流量核算工程 1	否	名称, 必需在 EAS 中有。比方: 客户、	客户
cashAsstActNumber1	String	现金流量核算对象编码 1	否	编码, 必需在 EAS 中有。比方: 客户 A 的编码	01.01

cashAsstActName1	String	现金流量核算对象名称 1	否	名称，必需在 EAS 中有。比方：客户 A	客户 A
cashAsstActType2	String	现金流量核算工程 2	否	以下：取决于科目是否核算多个工程，最多 8 个	
cashAsstActNumber2	String	现金流量核算对象编码 2	否		
cashAsstActName2	String	现金流量核算对象名称 2	否		
cashAsstActType3	String	现金流量核算工程 3	否		
cashAsstActNumber3	String	现金流量核算对象编码 3	否		
cashAsstActName3	String	现金流量核算对象名称 3	否		
cashAsstActType4	String	现金流量核算工程 4	否		
cashAsstActNumber4	String	现金流量核算对象编码 4	否		
cashAsstActName4	String	现金流量核算对象名称 4	否		
cashAsstActType5	String	现金流量核算工程 5	否		
cashAsstActNumber5	String	现金流量核算对象编码 5	否		
cashAsstActName5	String	现金流量核算对象名称 5	否		
cashAsstActType6	String	现金流量核算工程 6	否		
cashAsstActNumber6	String	现金流量核算对象编码 6	否		

cashAsstActName6	String	现金流量核算对象名称 6	否		
cashAsstActType7	String	现金流量核算工程 7	否		
cashAsstActNumber7	String	现金流量核算对象编码 7	否		
cashAsstActName7	String	现金流量核算对象名称 7	否		
cashAsstActType8	String	现金流量核算工程 8	否		
cashAsstActNumber8	String	现金流量核算对象编码 8	否		
cashAsstActName8	String	现金流量核算对象名称 8	否		

## 2. 接口说明:

将正确的符合条件的凭证引入到系统中，作为暂存凭证或者提交状态的凭证，错误的和不符合条件的凭证给与提示，不引入到系统。

### 2. 假设引入核销记录

那么每一张凭证中，需要有往来科目，自动设置核销分录

### 3. 假设引入现金流量

那么一般对象数组中，也存放现金流量数组，数组中的对象也是 WSWSVoucher，与凭证行的区分标记是：itemFlag 字段描述 0 不是现金流量 1 是现金流量

### 4. 返回值 String[][]

是一个长度为凭证数，宽度是 5 的字符串数据

宽度的定义：外部系统凭证号，凭证字，年度，期间，处理信息，特别信息，金蝶凭证号

1. 凭证号：第 3 方系统自身产生的凭证号

2. 凭证字：第 3 方系统自身产生的凭证字

3. 年度：第 3 方系统自身产生的会计年度

4. 期间：第 3 方系统自身产生的会计期间

## 1. 处理信息:

2. 特别信息 isVerify: 应当是 EAS 在保存凭证时产生的特别, 如校验不通过产生的提示信息等。

3. 金蝶返回凭证号: EAS 产生的凭证号, 用于回填 FreightNet 系统的。

4. 处理信息的定义: 0000 成功 其他处理信息见后面

## 5. 处理方法:

将正确的符合条件的凭证、核销记录、现金流量引入, 返回的信息包含全部内容;

有问题的凭证, 只返回凭证号和错误信息定义

## 6. 参数信息描述

WSWSVoucher 一行 是一个凭证对象或者 是一个现金流量对象

凭证对象行包括: 凭证行、分录行、关心账行, itemFlag = 0

现金流量行包括: 凭证号、分录号、现金流量行, itemFlag = 1

在对象传入时, 请挨次传入:

比方共有 5 行记录, 3 行是凭证对象行, 2 行是现金流量行, 那么格式如下

凭证号	分录号	...	关心账	...	现金流量(对方科目分录号)	...	流量标记
1	1						0
1	2						0
1	3						0
1	1				3		1
1	2				3		1

## 7. 其他处理信息:

1000 没有凭证数据传入

1001 依据传入的组织编码找不到组织

- 1002 依据传入的科目编码找不到科目
- 1003 依据传入的币别编码找不到币别
- 1004 依据传入的登账日期找不到适宜的期间
- 1005 依据传入的凭证类型名称找不到凭证类型
- 1006 所引入凭证币别，不属于本科目币别核算值
- 1007 没有找到核算工程
- 1008 制单人不能为空
- 1010 凭证没有找到过账人
- 1111 其他特别：如保存失败，网络特别等

2023 依据编码没有取得现金流量工程

2023 现金流量性质列不正确，当主表工程挂关心账时，性质列必需为外部或内部, 外部凭证号: {0}

2023 没有找到核算工程: {0}

- 3001 没有权限增凭证
- 3003 登账日期和期间不匹配
- 3008 组织单位的报告币折算方式没有选取
- 3100 凭证引入重复
- 3101 参考消息重复
- 4001 本位币借贷不相等。
- 4002 报告币借贷不相等。
- 4003 凭证类型与分录科目不匹配: {0} ({1})
- 4004 第 {0} 条分录的第 {1} 条关心账必需录入结算方式。
- 4005 第 {0} 条分录的第 {1} 条关心账非银行科目不允许录入结算方式。
- 4006 第 {0} 条分录缺少科目。
- 4007 同一凭证不允许同时包含表内科目与表外科目。
- 4008 第 {0} 条分录的科目带自定义属性或核算工程, 但没有关心账。
- 4009 第 {0} 条分录的原币金额不等于关心账原币金额之和。
- 4110 第 {0} 条分录的本位币金额不等于关心账本位币金额之和。
- 4111 第 {0} 条分录的报告币金额不等于关心账报告币金额之和。
- 4112 凭证的会计期间不能早于当前会计期间。
- 4113 分账制，凭证头币种不能为空。

- 4114 第{0}条分录科目“{1}”的核算币别与分录币别“{2}”不全都。
- 4115 会计期间不能为空。
- 4116 凭证类型不能为空。
- 4117 公司不能为空。
- 4118 第{0}条分录缺少币别。
- 4120 第{0}条分录缺少借贷方向。
- 4121 第{0}条分录的第{1}条关心账行缺少核算工程组合。
- 4122 凭证的借方本币与凭证行不符。
- 4123 凭证的贷方本币与凭证行不符。
- 4124 凭证的借方报告币与凭证行不符。
- 4125 凭证的贷方报告币与凭证行不符。
- 4126 凭证头的行数量与凭证行不符。
- 4128 第{0}条分录的第{1}条关心账行缺少必录的自定义属性。
- 4129 凭证应当至少存在两条分录。
- 4130 第{0}条分录的科目“{1}”不是明细科目。
- 4131 第{0}条分录的科目“{1}”已被禁用。
- 4132 该凭证必需录入现金流量。
- 4161 该凭证不存在或已经被删除，不能进展这项操作。
- 4162 凭证的会计期间不能早于当前会计期间。
- 4300 第{0}条分录是核销分录但没有设置核销标志。
- 4301 第{0}条分录不是核销分录但设置了核销标志。
- 4400 第{0}条分录的第{1}条关心账是核销记录但没有设置对应的往来账记录。
- 4401 第{0}条分录的第{1}条关心账不是核销记录但设置了对应的往来账记录。
- 4501 凭证有错误：{0}
- 4800 每张凭证至少应当有一条分录。
- 4302 第{0}条分录缺少摘要。
- 4109 第{0}条分录的币别与凭证头的币别不全都。
- 4666 第{0}条分录的数量、金额不能全都为 0。
- 4667 第{0}条分录第{1}条关心账的数量、金额不能全都为0。
- 4898 第{0}条分录为往来科目，但它的第{1}条关心账缺少到期日。
- 4900 现金流量工程{0}已经被禁用！

4980 第 {0} 条分录科目所属公司与凭证所属公司不全都。

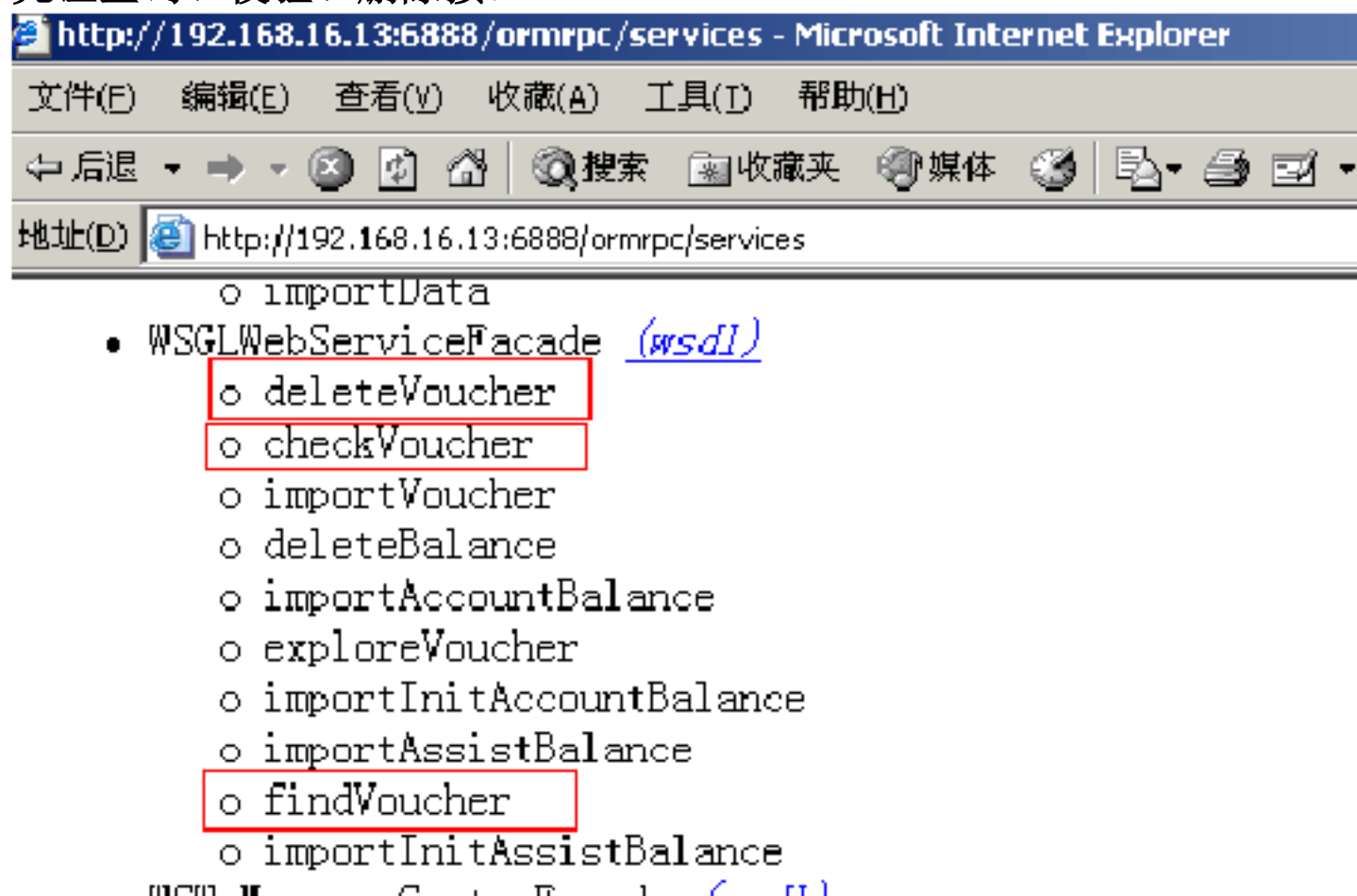
4345 记账日期不能为空。

4346 业务日期不能为空。

4347 凭证类型不能为空。

4355 输入的凭证编码重复

### 3. 凭证查询、校验、删除接口



### 4. “凭证查询”接口说明

findVoucher(String comNumber, int year, int periodNumber)

- 输入：公司代码、期间年（2023）、期间月（7），
- 输出：凭证字、凭证号、参考消息
- 输出类型：String[]，以“，”组合，需要拆分。
- 默认查询条件：所查凭证来源系统必需为“根底系统”（查询的是从前从webservice中引入的凭证，才是根底系统）

- 返回类型: String[]
  - #0001 未找到公司      #0002 未找到期间正
  - 确结果              转账, 0001, FLY20230701001

	公司编码	期间年	期间月
1	wfdcom1		
2	wfdcom2		

```

WSGLWebServiceFacadeSrvProxyServiceLocator locator =
    new WSGLWebServiceFacadeSrvProxyServiceLocator();

for (int i = 0; i < n; i++) {
    if (!StringUtils.isEmpty(companynumber[i]) && !StringUtils.isEmpty(year[i])
        && !StringUtils.isEmpty(month[i]) && companynumber[i] != null
        && year[i] != null && month[i] != null) {
        String[] ls = locator.getWSGLWebServiceFacade(endpoint).findVoucher(companynumber[i],
            Integer.parseInt(year[i]), Integer.parseInt(month[i]));
    }
}

```

#### 4. “凭证校验”接口说明

checkVoucher( String comNumber, int year, int periodNumber, String messageNumber,  
double localAmount)

- 输入: 公司代码、期间年、期间月、参考消息、本位币金额
- 输出: 金蝶凭证号
- 输出类型: String
- 默认查询条件: 所查公司, 必需是财务实体组织
- 返回类型: String

#0001 未找到公司      #0002 未找到期间

#1001 未找到符合条件的凭证

正确结果      凭证号



凭证其它功能		checkVoucher(凭证校验)			
	公司编码	期间年	期间月	参考消息	本位币金额
1	wfdcom1				
2	wfdcom2				

```

WSGLWebServiceFacadeSrvProxyServiceLocator locator
    = new WSGLWebServiceFacadeSrvProxyServiceLocator();

for (int i = 0; i < n; i++) {
    if (!StringUtils.isEmpty(companyNumber[i]) && !StringUtils.isEmpty(year[i])
        && !StringUtils.isEmpty(month[i]) && !StringUtils.isEmpty(fexp[i])
        && !StringUtils.isEmpty(localAmount[i]) && companyNumber[i] != null
        && year[i] != null && month[i] != null && fexp[i] != null
        && localAmount[i] != null) {
        String value = locator.getWSGLWebServiceFacade(endpoint).checkVoucher(companyNumber[i],
            Integer.parseInt(year[i]), Integer.parseInt(month[i]),
            fexp[i], Double.parseDouble(localAmount[i]));
    }
}

```

### 3. “凭证删除”接口说明

deleteVoucher(String companyNumber, String period, String voucherNumber, String fexp )

参数: String companyNumber “HQF001 “ 公司编  
 码String period “2023.09 “ 期间  
 String voucherNumber “20 “ 凭证号  
 String fexp “000000112 “ 参考消

息返回类型: int

0 成功	10000 公司编码不能为空
10001 未找到公司编码	10002 期间输入格式不正确
10003 凭证号不能为空	10004 参考消息不能为空
10005 未找到此凭证或凭证已被删除	
10006 凭证删除失败	


	公司编码	期间	凭证号	参考消息
1	wfdcom1	2006.9		
2	wfdcom2	2006.9		

```
WSGLWebServiceFacadeSrvProxyServiceLocator locator =
    new WSGLWebServiceFacadeSrvProxyServiceLocator();

/*
 * 0 成功; 10000 公司编码不能为空; 10001 未找到公司编码
 * 10002 期间输入格式不正确; 10003 凭证号不能为空
 * 10004 参考消息不能为空; 10005 未找到此凭证或凭证已被删除
 * 10006 凭证删除失败
 */
for (int i = 0; i < n; i++) {
    if (!StringUtils.isEmpty(companyNumber[i]) && !StringUtils.isEmpty(period[i])
        && !StringUtils.isEmpty(voucherNumber[i]) && !StringUtils.isEmpty(fexp[i])
        && companyNumber[i] != null && period[i] != null && voucherNumber[i] != null
        && fexp[i] != null) {
        int iValue = locator.getWSGLWebServiceFacade(endpoint).deleteVoucher(
            companyNumber[i], period[i], voucherNumber[i], fexp[i]);
    }
}
```

### 3. “科目余额引入” 接口说明

说明：两个接口方法都可用，且用法全都。

地址 (Q)  http://localhost:6888/ormrpc/services

- WSWSAccountBalance ([wsdl](#))
  - closePeriod
  - importAccountBalance
  - deleteAccountBalance
  - findObjectByNumber
- WSMoveInWarehsBillFacade ([wsdl](#))
  - importData
- WSOtherIssueBillFacade ([wsdl](#))
  - importData
- WSPurInWarehsBillFacade ([wsdl](#))
  - importData
- WSGLWebServiceFacade ([wsdl](#))
  - getOrg
  - getAccount
  - deleteBalance
  - getAsstActType
  - deleteVoucher
  - importAccountBalance
  - exploreVoucher
  - importInitAccountBalance
  - checkVoucher
  - importAssistBalance
  - findVoucher
  - importInitAssistBalance
  - importVoucher
  - getAcctType
  - getAcctTypeDetail
  - getAccountBalance
  - getAssitBalance
  - getVoucher

importAccountBalance (WSWSAccountBalance[] acctBal)

- 输入: WSWSAccountBalance [] 一般对象数组, 见构造EAS科目余额字段以及含义说明
- 输出: 引入结果
- 输出类型: String
- 返回类型:
 

1000	引入成功
None AccountBlance	输入科目余额对象数组为空
1001	未找到对应编码的公司
1002	未找到对应编码公司的对应编码的科目