**电子凭证会计数据标准个性化工具包**

**（推广应用版V1.0）**

**使用说明**

用友网络科技股份有限公司

2025年4月

**前 言**

本说明文档的目标读者是电子凭证接收端单位的技术人员，用于了解通过使用电子凭证会计数据标准推广应用工具包来对符合电子凭证会计数据标准的电子凭证进行验签、解析、转换等操作。接收端单位的会计信息系统可通过集成电子凭证会计数据标准推广应用工具包来降低技术难度和实施成本。

如您在使用本工具包过程中有相关疑问，请您联系：010-62277399或400-634-2658，联系时请说明是电子凭证会计数据标准推广应用工具包咨询。

**文档修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **修订号** | **描述** | **编写人** | **审核人** |
| 2025-04 | 1.0 | 建立文档 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. **电子凭证会计数据标准推广应用工具包集成说明**

* **电子凭证会计数据标准推广应用工具包**为业务系统提供验证、解析电子凭证相关功能。
* 调用方集成工具包,需添加DJFileUtil.jar（下简称SDK）为依赖，集成SDK后,业务系统可以通过调用SDK实现在本地服务进行文档的验证、获取xbrl附件、验证xbrl签名、xbrl与json互转功能。
* SDK运行环境应为Linux内核操作系统或Windows操作系统，Linux环境支持64位JDK，Windows环境支持32位JDK与64位JDK。

1. **SDK集成流程**

* **添加libautoseal依赖库**

Linux环境:

将提供的libautoseal.so文件放置到服务运行环境磁盘中，一般为/usr/lib或/usr/lib64下。

Windows环境:

忽略此步骤。

* **添加**SDK**依赖**

1.maven方式(推荐)

SDK的开发过程使用 Maven 项目构建和管理工具，只需将工具包中提供的 jar 包安装到 Maven 仓库即可使用。工具包中包含 1 个 jar 包及对应的安装描述文件：DJFileUtil-1.0.jar和DJFileUtil-1.0.xml。

软件开发人员可使用如下命令将 jar 包安装到本地Maven 仓库。

|  |
| --- |
| mvn install:install-file -Dfile=DJFileUtil-1.0.jar -DpomFile=DJFileUtil-1.0.xml |

上述 jar 包安装完毕后，在工程代码中使用如下依赖方式引入该工具包。

|  |
| --- |
| <dependency>  <groupId>com.dj</groupId>  <artifactId>DJFileUtil</artifactId>  <version>1.0</version>  </dependency> |

1. 包含依赖方式

直接引用DJFileUtil-1.0-jar-with-dependencies.jar

* **SDK调用**

工具包的API入口为com.dianju.cz.api.EVoucherFileUtil类

* **初始化SDK配置**

1. **执行EVoucherFileUtil. setSrvSealUtil (soPath)方法**

Linux环境下soPath参数为libautoseal.so文件的绝对路径

Windows环境下soPath参数为空字符串。

本方法在程序运行期间全局执行一次即可。

**2. 初始化可信证书列表（可选）**

调用方根据场景需要选择是否维护一份可信证书列表。当需要对待验证文件中的电子印章的签章证书进行颁发者校验时，需要执行EVoucherFileUtil. initCertChainCache(String certChainDir)方法，并指定可信颁发者证书存放路径，此方法将加载该路径下的.cer和.p7b文件添加到缓存。在调用文档验证方法并指定了校验签章证书参数为true时，将校验签章证书是否由该路径下的证书颁发。当该路径下证书有变动时，应重新调用此方法初始化缓存。如无变动，则本方法程序运行期间全局执行一次即可。

开启此项验证意味着您需自行维护一份可信证书列表。在验证文档时，SDK将解析出印章的签名证书，并遍历可信证书列表来判断该签名证书是否由可信证书列表内的证书颁发。如果签名证书的颁发者不在可信证书列表内，则验证方法将返回”签名证书不可信”。

如不需要此项验证则忽略本步骤。

1. **SDK接口说明**

* **初始化**

|  |  |  |
| --- | --- | --- |
| **方法** | **参数** | **返回值** |
| EVoucherFileUtil. setSrvSealUtil (soPath) | soPath: Linux:libautoseal.so路径  Windows:空字符串 | 无 |
| EVoucherFileUtil.initCertChainCache(certChainDir) | certChainDir: 可信证书链存放路径 | 无 |

* **EVoucherFileUtil方法**
  + **文档验签**

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数** | **返回值** | **备注** |
| verifyFile(String filePath) | filePath:待验证文档本地绝对路径 | Response {  int code:状态码(1验证通过，其它验证不通过)  String data:验证结果描述  } | 默认不验证签章证书颁发者 |
| verifyFile(String filePath,boolean checkCert) | filePath:待验证文档本地绝对路径  checkCert:是否验证签章证书颁发者,true验证,false不验证 | Response {  int code:状态码(1验证通过，其它验证不通过)  String data:验证结果描述  } | 当checkCert参数为true时，请确保执行了上述“初始化可信证书列表”步骤。 |

* + **Xbrl验签(针对增值税电子发票zip包验签)**

|  |  |  |
| --- | --- | --- |
| **方法** | **参数** | **返回值** |
| verifyXmlAndSign(String signedValue,String xmlData) | signedValue:签名值文本  xmlData:xml文本 | boolean:  true 验证通过  false 验证不通过 |
| verifyXmlAndSign(String zipFilePath) | zipFilePath:  包含了.sign和.xml文件的zip文件本地绝对路径。 | boolean:  true 验证通过  false 验证不通过 |
| verifyXmlAndSign(byte[] zipFileBytes) | zipFilePath:  包含了.sign和.xml文件的zip文件数据 | boolean:  true 验证通过  false 验证不通过 |

* + **获取文档XBRL附件**

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数** | **返回值** | **备注** |
| getXbrIFromFile(  string oriFilePath) | oriFilePath:文档本地绝对路径 | Response {  int code:状态码(1获取成功，其它获取失败)  T data: 当code为1时,data为VoucherFileInfo对象。  当code不为1时，data为String类型的失败信息。  } | VoucherFileInfo{  String voucherType;  String xbrlData;  } |

* + **导出文档XBRL附件**

|  |  |  |
| --- | --- | --- |
| **方法** | **参数** | **返回值** |
| exportXbrlFromFile(String oriFilePath, String targetFilePath) | oriFilePath:文档本地绝对路径  targetFilePath:xbrl附件保存路径 | Response {  int code:状态码(1获取成功，其它获取失败)  String data: 成功时data为voucherType,失败时为错误信息  } |

* + **XBRL转JSON**

|  |  |  |
| --- | --- | --- |
| **方法** | **参数** | **返回值** |
| xbrl2Json(String xbrlXml, String configId) | xbrlXml:xbrlXml数据  configId:xbrl单据类型(见附录) | JSONObject |

* + **JSON转XBRL**

|  |  |  |
| --- | --- | --- |
| **方法** | **参数** | **返回值** |
| json2Xbrl(String jsonValue, String configId) | jsonValue:json字符串  configId:xbrl单据类型(见附录) | String：xbrlXml字符串 |

* + **XML转JSON**

|  |  |  |
| --- | --- | --- |
| **方法** | **参数** | **返回值** |
| xml2Json(String xml) | xml:xml结构化数据 | JSONObject |

1. **附录**

* **工具包中针对不同票据类型提供了对应的配置文件进行适配，在使用 API 时需要传入对应的票据类型编号（configId），票据类型编码清单如下表所示：**

|  |  |  |
| --- | --- | --- |
| **序号** | **票据类型** | **票据类型编号** |
| 1 | 增值税电子普通发票（开具方） | inv\_ord\_issuer |
| 2 | 增值税电子普通发票（接收方） | inv\_ord\_receiver |
| 3 | 增值税电子专用发票（开具方） | inv\_spcl\_issuer |
| 4 | 增值税电子专用发票（接收方） | inv\_spcl\_receiver |
| 5 | 电子非税收入一般缴款书（开具方） | ntrev\_gpm\_issuer |
| 6 | 电子非税收入一般缴款书（接收方） | ntrev\_gpm\_receiver |
| 7 | 电子发票（铁路电子客票）（开具方） | rai\_issuer |
| 8 | 电子发票（铁路电子客票）（接收方） | rai\_receiver |
| 9 | 电子发票（航空运输电子客票行程单）（开具方） | atr\_issuer |
| 10 | 电子发票（航空运输电子客票行程单）（接收方） | atr\_receiver |
| 11 | 银行电子回单（开具方） | bker\_issuer |
| 12 | 银行电子回单（接收方） | bker\_receiver |
| 13 | 银行电子对账单 | bkrs |
| 14 | 财政电子票据 | efi |
| 15 | 全面数字化的电子发票-普通发票-接收方 | einv\_ord\_receiver |
| 16 | 全面数字化的电子发票-增值税专用发票-接收方 | einv\_spcl\_receiver |

* **状态码**

|  |  |
| --- | --- |
| **状态码** | **说明** |
| 1 | 成功 |
| 400 | 参数错误 |
| 500 | 工具包错误 |
| 1000 | 工具包错误 |
| 1002-1003 | 获取xbrl为空 |
| 1011 | 文件读写错误 |
| 1401 | 签章证书不是由可信颁发者列表中内的证书颁发 |