Демонстрационный пример средства взаимодействия с автоматизированным сервисом Портала АИС "Учет счетов-фактур". Руководство программиста

Версия **1.4**

Дата **22.02.2019**

(С) ЗАО «АВЕСТ»

Количество листов **63**

Минск

2019

# Аннотация

Документ описывает технологии и форматы, используемые в работе автоматизированного сервиса портала электронных счетов-фактур НДС.

Оглавление

[Аннотация 2](#_Toc1744545)

[1. Введение 7](#_Toc1744546)

[2. Автоматизированный сервис портала электронных счетов-фактур НДС 8](#_Toc1744547)

[3. Формат электронного документа версии 1.0 9](#_Toc1744548)

[3.1. Описание элементов документа 10](#_Toc1744549)

[3.2. Квитанция приёма электронного документа 13](#_Toc1744550)

[3.3. Информация о статусе обработки счета-фактуры НДС 14](#_Toc1744551)

[4. Использование компонента ActiveX для взаимодействия с АС портала на стороне отправителя 15](#_Toc1744552)

[4.1. Подпись и направление ЭСЧФ на портал 15](#_Toc1744553)

[4.2. Получение статуса поданного ЭСЧФ 16](#_Toc1744554)

[5. Использование компонента ActiveX для взаимодействия с АС портала на стороне получателя 17](#_Toc1744555)

[5.1. Получение счетов-фактур, поступивших в адрес получателя 17](#_Toc1744556)

[5.2. Добавление подписи и направление ЭСЧФ на портал 19](#_Toc1744557)

[5.3. Получение статуса поданного ЭСЧФ 20](#_Toc1744558)

[6. ActiveX компонент EInvVatService 21](#_Toc1744559)

[6.1. Класс Connector 21](#_Toc1744560)

[Метод Login 21](#_Toc1744561)

[Метод Logout 22](#_Toc1744562)

[Метод Connect 22](#_Toc1744563)

[Метод Disconnect 23](#_Toc1744564)

[Метод CreateEDoc 23](#_Toc1744565)

[Метод SendEDoс 23](#_Toc1744566)

[Метод GetStatus 24](#_Toc1744567)

[Свойство Ticket 24](#_Toc1744568)

[Метод GetList 24](#_Toc1744569)

[Метод GetEDoс 25](#_Toc1744570)

[Метод GetMyCertProperty 25](#_Toc1744571)

[Метод SetProxy 26](#_Toc1744572)

[Метод SetServiceProperty 26](#_Toc1744573)

[Свойство LastError 27](#_Toc1744574)

[Свойство Version 27](#_Toc1744575)

[6.2. Класс EDocument 28](#_Toc1744576)

[Метод LoadFromFile 28](#_Toc1744577)

[Метод SaveToFile 28](#_Toc1744578)

[Метод GetData 29](#_Toc1744579)

[Метод SetData 29](#_Toc1744580)

[Свойство Document 29](#_Toc1744581)

[Метод Document.GetXmlNodeValue 30](#_Toc1744582)

[Метод Document.GetXmlNodeAttribute 30](#_Toc1744583)

[Метод Document.ValidateXml 30](#_Toc1744584)

[Метод Sign 30](#_Toc1744585)

[Метод GetSignCount 31](#_Toc1744586)

[Метод GetSignProperty 31](#_Toc1744587)

[Метод VerifySign 32](#_Toc1744588)

[6.3. Класс Ticket 33](#_Toc1744589)

[Свойство Accepted 33](#_Toc1744590)

[Свойство Message 33](#_Toc1744591)

[6.4. Класс StatusInfo 33](#_Toc1744592)

[Метод Verify 34](#_Toc1744593)

[Свойство Status 34](#_Toc1744594)

[Свойство Message 34](#_Toc1744595)

[Свойство Since 34](#_Toc1744596)

[6.5. Класс EDocList 34](#_Toc1744597)

[Метод Verify 35](#_Toc1744598)

[Свойство ToDate 35](#_Toc1744599)

[Свойство Count 35](#_Toc1744600)

[Метод GetItemAttribute 35](#_Toc1744601)

[7. Использование компонента Java для взаимодействия с АС портала на стороне отправителя 36](#_Toc1744602)

[7.1. Подпись и направление ЭСЧФ на портал 36](#_Toc1744603)

[7.2. Получение статуса поданного ЭСЧФ 37](#_Toc1744604)

[8. Использование компонента Java для взаимодействия с АС портала на стороне получателя 39](#_Toc1744605)

[8.1. Получение счетов-фактур, поступивших в адрес получателя 39](#_Toc1744606)

[8.2. Добавление подписи и направление ЭСЧФ на портал 41](#_Toc1744607)

[8.3. Получение статуса поданного ЭСЧФ 42](#_Toc1744608)

[9. Программный интерфейс EVatService для языка Java 44](#_Toc1744609)

[9.1. Класс EVatService 44](#_Toc1744610)

[Конструктор EVatService 44](#_Toc1744611)

[Метод login 45](#_Toc1744612)

[Метод logout 46](#_Toc1744613)

[Метод connect 46](#_Toc1744614)

[Метод disconnect 46](#_Toc1744615)

[Метод createEDoc 46](#_Toc1744616)

[Метод sendEDoс 47](#_Toc1744617)

[Метод getStatus 47](#_Toc1744618)

[Метод getList 47](#_Toc1744619)

[Метод getEDoс 48](#_Toc1744620)

[Метод getMyCertProperty 48](#_Toc1744621)

[Метод setServiceProperty 49](#_Toc1744622)

[9.2. Класс AvEDoc 50](#_Toc1744623)

[Метод load 50](#_Toc1744624)

[Метод sign 50](#_Toc1744625)

[Метод verify 51](#_Toc1744626)

[Метод getDocument 51](#_Toc1744627)

[Метод getEncoded 51](#_Toc1744628)

[Метод getSignCount 51](#_Toc1744629)

[Метод getSignProperty 52](#_Toc1744630)

[Метод verifySign 52](#_Toc1744631)

[Метод getLastError 52](#_Toc1744632)

[9.3. Класс AvETicket 54](#_Toc1744633)

[Метод accepted 54](#_Toc1744634)

[Метод getMessage 54](#_Toc1744635)

[9.4. Класс AvEStatus 54](#_Toc1744636)

[Метод verify 55](#_Toc1744637)

[Метод getStatus 55](#_Toc1744638)

[Метод getMessage 55](#_Toc1744639)

[Метод getSince 55](#_Toc1744640)

[9.5. Класс AvEList 55](#_Toc1744641)

[Метод verify 56](#_Toc1744642)

[Метод getToDate 56](#_Toc1744643)

[Метод getCount 56](#_Toc1744644)

[Метод getItemAttribute 56](#_Toc1744645)

[9.6. Класс AvDoc 57](#_Toc1744646)

[Метод load 57](#_Toc1744647)

[Метод getXmlNodeValue 57](#_Toc1744648)

[Метод validateXML 58](#_Toc1744649)

[Метод getEncoded 58](#_Toc1744650)

[9.7. Класс AvError 58](#_Toc1744651)

[Метод getError 58](#_Toc1744652)

[Метод getMessage 59](#_Toc1744653)

[Приложение 1. Программное приложение simpleapp 60](#_Toc1744654)

[Приложение 2. Статусы ЭСЧФ НДС 61](#_Toc1744655)

[Термины и сокращения 62](#_Toc1744656)

# 1. Введение

Автоматизированный сервис портала электронных счетов-фактур НДС, далее АС портала ЭСЧФ, предназначен для автоматизации обработки электронных счетов-фактур НДС.

АС портала ЭСЧФ разрабатывается как дополнение к интерактивному порталу электронных счетов-фактур НДС.

Данный сервис позволяет интегрировать сервисы портала с учетными системам предприятий и организаций.

В процессе обработки подготовленных счетов-фактур возникают следующие этапы:

У отправителя счета-фактуры

1. Подпись подготовленного счета-фактуры (XML файла), передача подписанного счета-фактуры на сервис портала.
2. Получение статуса обработки счета-фактуры.

У получателя счет-фактуры

1. Формирование запроса наличия выставленных контрагентом счетов-фактур.
2. Получение в ответ на этот запрос списка выставленных счетов-фактур.
3. Получение счета-фактуры по его номеру.
4. Подписание счетов-фактур, по которым получатель принял решение о приёме к зачету, подписью получателя и направление их на портал через автоматизированный сервис портала.
5. Формирование запроса и получение статуса обработки счета-фактуры, подписанной получателем.

В данном сценарии упомянуты три стороны, участвующие в процессе приёма и обработки электронных счетов-фактур НДС: отправитель счета-фактуры, получатель счета-фактуры, и автоматизированный сервис портала электронных счетов-фактур.

# 2. Автоматизированный сервис портала электронных счетов-фактур НДС

Автоматизированный сервис приёма и обработки электронных счетов-фактур НДС доступен с использованием протоколов TCP/IP, TLS, SOAP и иных, с обязательной аутентификацией клиента на основании сертификата открытого ключа, выданного республиканским удостоверяющим центром государственной системы управления открытыми ключами (ГосСУОК) либо удостоверяющим центром республиканского унитарного предприятия «Информационно-издательский центр по налогам и сборам». Сертификат должен содержать УНП юридического лица, создающего ЭСЧФ или являющегося его получателем.

Все ответы автоматизированного сервиса оформлены в виде электронного документа, содержащего ЭЦП сервиса. Данные ответы рекомендуется сохранять в архив электронных документов или базы данных учетных систем для дальнейшего разбора возможных конфликтных ситуаций.

Для интеграции с учетными системами разработаны два программных интерфейса:

А) программный интерфейс на языке программирования Java;

Б) программный компонент ActiveX для использования в программах ОС Windows.

Программные интерфейсы поставляются с набором примеров, реализующих выполнение типовых операций взаимодействия с автоматизированным сервисом приёма и обработки электронных счетов-фактур НДС. Примеры содержат код на соответствующих языках программирования, файлы, требуемые для сборки(компиляции), командные файлы для запуска примеров и документацию (описание).

Форматы электронных документов, используемых автоматизированным сервисом портала, описаны в разделе 3.

# 3. Формат электронного документа версии 1.0

Электронный документ — это XML документ в кодировке UTF-8 без BOM, содержащий элементы для хранения исходного документа и электронных цифровых подписей.

Формат позволяет сохранить исходный документ, а также набор подписей этого документа в одном файле и обеспечить всю информацию, необходимую для проверки и дальнейшей обработки исходного XML документа.

Формат определяет XML документ e-document следующим образом:

<e-document version="1">

 <document id="id документа" contentType="тип документа" contentEncoding="кодировка документа">

 .... здесь исходный XML документ (счет-фактура, квитанция или статус обработки счета-фактуры)

 </document>

 <signatures>

 <signature>

 <reference URI="#id документа"/>

 <canonicalizationMethod algorithm="…" />

 <value contentType="вид ЭЦП" contentEncoding="кодировка ЭЦП">

  .... здесь данные ЭЦП

 </value>

 </signature>

 <signature>

 .... еще ЭЦП

 </signature>

....

 </signatures>

</e-document>

## 3.1. Описание элементов документа

1. Объявление электронного документа

<e-document version="1">

Объявление электронного документа с указанием версии формата. На данный момент поддерживается только версия 1.

1. Вложенный исходный XML документ

 <document id="id документа" contentType="тип документа" contentEncoding="кодировка документа">

 .... здесь исходный XML документ (счет-фактура, квитанция или статус обработки счета-фактуры)

 </document>

Элемент предназначенный для размещения исходного подписываемого XML документа имеет следующие атрибуты:

* 1. Идентификатор узла документа **id**. Содержит GUID для идентификации подписываемого узла.
	2. Тип документа **contentType**. Поддерживается только один тип документа: **xml**.
	3. Кодировка включения исходного документа **contentEncoding**. Поддерживаются следующие методы кодирования исходного документа:

а) **plain** ­– исходный документ без дополнительного кодирования;

б) **base64**­– исходный документ закодирован Base64;

в) **zip;base64**­– исходный документ упакован в zip архив и закодирован Base64.

Данные исходного документа должны быть помещены в значение элемента document в соответствии с указанными атрибутами типа документа и кодировки документа.

Пример

 <document id="AAD54B32-C76A-476B-BB3B-4CD1A88AD051" contentType="xml" contentEncoding="zip;base64">UEsDBAoAAAAAABh2fUgb+b4dqwoAAKsKAAAIAAAAZG9jdW1lbnQKPGlzc3VhbmNlIHhtbG5z

…

AAAAAAAAAGRvY3VtZW50UEsFBgAAAAABAAEANgAAANEKAAAAAA==

</document>

1. Коллекция элементов ЭЦП

 <signatures>

 <signature>

 ....

 </signature>

 <signature>

 .... еще ЭЦП

 </signature>

....

 </signatures>

Коллекция ЭЦП **signatures** предназначена для хранения одной или нескольких ЭЦП исходного документа.

1. ЭЦП

 <signature>

 <reference URI="#id документа"/>

 <canonicalizationMethod algorithm="…" />

 <value contentType="вид ЭЦП" contentEncoding="кодировка ЭЦП">

 .... здесь данные ЭЦП

 </value>

 </signature>

Элемент signature определяет одну ЭЦП исходного документа.

Элемент имеет следующие вложенные элементы:

* 1. Элемент для описания подписанного документа **reference**. Содержит атрибут URI с указанием ссылки на идентификатор узла подписанного документа.
	2. Элемент описания канонизации исходного документа перед выработкой ЭЦП **canonicalizationMethod**. Элемент содержит один атрибут **algorithm**, указывающий на международный идентификатор используемого алгоритма канонизации XML, рекомендуется использовать метод канонизации [http://www.w3.org/2001/10/xml-exc-c14n#](http://www.w3.org/2001/10/xml-exc-c14n). В случае если используемые программные средства не позволяют использовать канонизацию исходного XML документа, то канонизацию можно не применять, в этом случае элемент **canonicalizationMethod** может быть опущен.
	3. Элемент значения ЭЦП **value**. Значение ЭЦП помещается в значение этого атрибута в соответствие с атрибутами **contentType** и **contentEncoding**. **contentType** должен иметь значение «**pkcs7**», а **contentEncoding** должен быть «**base64**» либо «**zip;base64**». В значение элемента помещается ЭЦП в формате, описанном в СТБ 34.101.23 «Информационные технологии и безопасность. Синтаксис криптографических сообщений» как SignedData, и сертификат открытого ключа подписавшего. При этом поле encapContentInfo должно отсутствовать, поскольку подписываемые данные не содержатся внутри структуры «Криптографическое сообщение» (СТБ 34.101.23, пункты 8.2, 8.3).

## 3.2. Квитанция приёма электронного документа

Автоматизированный сервис портала ЭСЧФ принимает документ и возвращает в ответ электронный документ — квитанцию о приёме электронного документа в формате XML. Квитанция представляет собой электронный документ (раздел 3.1), содержащий в элементе «document» следующую структуру:

 <ticket>

 <issuer URI="https://vat.gov.by/einvoice/"/>

 <reference>

 <digest>

 <method algId="1.2.112.0.2.0.34.101.31.81"/>

 <value>uooqbWYa5VCqcJCbuymBKqm17vY=</value>

 </digest>

 <objectIdentifier path="/issuance/general/number">

 <value>100030893-2015-0000000005</value>

 </objectIdentifier>

 </reference>

 <receipt accepted="no">

 <message>Сертификат абонента "Иванов" отозван</message>

 <date>2016-11-05T13:15:30Z</date>

 </receipt>

 </ticket>

Элемент issuer указывает на сервис, выдавший квитанцию. URI сервиса должен совпадать со значением элемента CN(CommonName) поля subject или subjectAltName технологического сертификата открытого ключа портала ЭСЧФ.

Элемент reference указывает для какого документа выдана квитанция.

Элемент reference/digest описывает хэш для поданного электронного документа (e-document), на который выдана эта квитанция. Элемент reference/digest/algId содержит OID алгоритма хэширования. Элемент reference/digest/value содержит значение функции хэширования в кодировке Base64. Функция хэширования вычисляется от поданного e-document начиная с <e-document> до завершающего </e-document> включительно.

reference/objectIdentifier содержит уникальный номер счета-фактуры, на который выдана квитанция. Атрибут reference/objectIdentifier/path содержит путь в XML документе, содержащий номер счета-фактуры в исходном документе. Элемент reference/objectIdentifier/value – значение уникального номера счета-фактуры.

Элемент receipt определяет результат приема счета-фактуры, включающий значения атрибута accepted и элемент message, содержащий дополнительное текстовое сообщение.

## 3.3. Информация о статусе обработки счета-фактуры НДС

Сервис портала ЭСЧФ при запросе текущего статуса обработки счета-фактуры в ответе возвращает электронный документ — статус обработки электронного документа. Статус представляет собой электронный документ (раздел 3.1), содержащий в элементе «document» следующую структуру:

 <status\_info>

 <issuer URI="https://vat.gov.by/einvoice/"/>

 <reference>

 <objectIdentifier path="/issuance/general/number">

 <value>100030893-2015-0000000005</value>

 </objectIdentifier>

 </reference>

 <document\_state>

 <status>COMPLETED</status>

 <message>ЭСЧФ создан, подписан отправляющей стороной</message>

 <since>2016-11-05T13:15:30Z</since>

 </document\_state>

 </status\_info>

Элемент issuer указывает на сервис, выдавший квитанцию. URI сервиса должен совпадать со значением элемента CN(CommonName) поля subject или subjectAltName технологического сертификата открытого ключа портала ЭСЧФ.

Элемент reference указывает для какого документа выдан статус.

reference/objectIdentifier содержит уникальный номер счета-фактуры, на который выдана квитанция. Атрибут reference/objectIdentifier/path содержит путь, указывающий на номер счета-фактуры в исходном документе. Элемент reference/objectIdentifier/value – значение уникального номера.

Элемент document\_state определяет текущий статус обработки электронного счета-фактуры НДС:

Элемент document\_state**/**status описывает текущий статус обработки счета-фактуры автоматизированным сервисом путем указания одной из констант COMPLETED, COMPLETED\_SIGNED и др.

Элемент document\_state/message содержит дополнительное текстовое сообщение о статусе.

Элемент document\_state/since содержит дату и время присвоения данного статуса ЭСЧФ.

# 4. Использование компонента ActiveX для взаимодействия с АС портала на стороне отправителя

Приведенный в следующих подразделах код показывает какие методы компонента ActiveX EInvVatService и в какой последовательности следует вызывать для взаимодействия с АС портала на стороне отправителя. Данный код приводится на языке Visual Basic, во фрагментах кода не указаны операторы описания переменных, операторы обработки ошибок, другие незначащие элементы. Полный текст примеров содержится в прилагаемых файлах.

## 4.1. Подпись и направление ЭСЧФ на портал

Отправитель:

1. Формирует в учетной системе электронный счет-фактуру НДС и сохраняет в виде XML файла.
2. Создаёт экземпляр класса EInvVatService.Connector для взаимодействия с АС портала и авторизуется с использованием личного сертификата.

 Set EVatService = CreateObject("EInvVatService.Connector")

 EVatService.Login("", 0)

1. Формирует электронный документ и проверяет структуру сформированного XML документа на соответствие XSD схеме.

 set InvVatXml = EVatService.CreateEDoc

 InvVatXml.Document.LoadFromFile(FileName)

 InvVatXml.Document.ValidateXML(xsd, 0)

1. Подписывает счет-фактуру, формируя электронный документ, содержащий счет-фактуру (issuance).

 InvVatXml.Sign(0)

1. Устанавливает соединение с АС портала и направляет подписанный отправителем счет-фактуру на портал.

 EVatService.Connect(objArgs(0))

 EVatService.SendEDoc(InvVatXml)

1. Получает квитанцию, проверяет подпись и проверяет, что документ принят успешно, в противном случае выводит сообщение сервиса.

 set InvVatTicket = EVatService.Ticket

 if InvVatTicket.Accepted <> 0 then

 WScript.Echo "Документ не принят: " + InvVatTicket.Message

 end if

Внимание! В состав дистрибутивного пакета входит приложение simpleapp, в файле eInvVat\simpleapp\vbs\src\signAndUploadDir.vbs продемонстрирована правильная организация процесса подписания и отправки на автоматизированный сервис ЭСЧФ на языке программирования Visual Basic Script с учетом необходимости обработки ошибок.

## 4.2. Получение статуса поданного ЭСЧФ

Отправитель:

1. Создаёт экземпляр класса EInvVatService.Connector для взаимодействия с АС портала и авторизуется с использованием личного сертификата.

 Set EVatService = CreateObject("EInvVatService.Connector")

 EVatService.Login("", 0)

1. Устанавливает соединение с АС портала и запрашивает статус счета-фактуры по его номеру.

 EVatService.Connect(serviceUrl)

 Set InvVatStatus = EVatService.GetStatus(InvVatNumber)

 WScript.Echo "Статус: " + InvVatStatus.status

Описание статусов ЭСЧФ приведено в Приложении 2.

Внимание! В состав дистрибутивного пакета входит приложение simpleapp, в файле eInvVat\simpleapp\vbs\src\updateStatusDir.vbs продемонстрирована правильная организация процесса запроса в автоматизированном сервисе статуса обработки ЭСЧФ на языке программирования Visual Basic Script с учетом необходимости обработки ошибок.

# 5. Использование компонента ActiveX для взаимодействия с АС портала на стороне получателя

## 5.1. Получение счетов-фактур, поступивших в адрес получателя

Получатель:

1. Создаёт экземпляр класса EInvVatService.Connector для взаимодействия с АС портала и авторизуется с использованием личного сертификата.

 Set EVatService = CreateObject("EInvVatService.Connector")

 EVatService.Login("", 0)

1. Устанавливает соединение с АС портала и запрашивает список поступивших счетов-фактур. Параметр fromDate — это дата и время в строковом виде, данный параметр означает, с какой даты требуется запрашивать счета фактуры. Будут запрошены документы от этой даты по текущий момент. Формат строки даты описан в разделе описания метода GetList класса EVatService.

 EVatService.Connect(serviceUrl)

 set InvList = EVatService.GetList(fromDate)

1. Проверяет ЭЦП полученного списка.

 res = InvList.Verify

1. Скачивает счета-фактуры и сохраняет их для просмотра.

for i = 0 to InvList.Count – 1

 InvVatNumber = EVatService.List.GetItemAttribute(i, "document/number")

 // необходимо проверить, что ЭСЧФ с таким номером уже сохранен на диск или в БД

 // или в учетную систему и не скачивать повторно скачанный ранее ЭСЧФ

 If not ужеСохраненаЛокально(invoiceNum) then

 InvVatXml = EVatService.GetEDoc(InvVatNumber)

 res = VerifyDocumentSign(InvVatXml)

 // обязательно сохранить скачанный ЭСЧФ на диск,

 // или в учетную систему пользователя

 res = InvVatXml.SaveToFile( fn )

 end if

next

Метод проверки ЭЦП скачанного ЭСЧФ (в примере выше названный VerifyDocumentSign) должен действовать следующим образом: пройти по списку ЭЦП подписанного документа, если среди подписей есть ЭЦП автоматизированного сервиса портала, то проверить только найденную ЭЦП, если среди подписей нет ЭЦП АС портала, то проверить все подписи в документе. Определить, что подпись сделана АС портала можно прочитав атрибуты сертификата вызовом метода GetSignProperty. Атрибуты сертификата АС портала следующие:

|  |  |
| --- | --- |
| Атрибут  | Значение |
| 2.5.4.3 (commonName) | Автоматизированный сервис портала АИС УСФ |

Скачанные документы необходимо сохранять на компьютере пользователя.

1. Если все документы были успешно получены (за исключением ошибки «вы уже ранее скачали этот документ»), необходимо сохранить в файл, или в базу данных, или в учетную систему пользователя дату и время списка ЭСЧФ для следующего запроса EVatService.GetList.

 toDate = InvList.ToDate

 // сохранение в файл, в БД, или в учетную систему

Тем самым формируется следующая последовательность обращений за получением поступающих ЭСЧФ:

№1. Запрос GetList(дата пустая, поскольку первое обращение к сервису)

№1.1 Скачивание и сохранение поступивших ЭСЧФ для дальнейшей обработки (подписания). Если в процессе скачивания были ошибки (за исключением ошибки «вы уже ранее скачали этот документ»), повторить шаг еще раз после некоторого перерыва, возможен сервис в данный момент перегружен.

№1.2 Сохранение даты InvList.ToDate в файл или в БД.

--- Промежуток времени, например сутки, неделя и т. д.

№2. Запрос GetList(дата прочитанная с диска, полученная на шаге 1.2)

№2.1 Скачивание и сохранение поступивших ЭСЧФ для дальнейшей обработки (подписания). Если в процессе скачивания были ошибки (за исключением ошибки «вы уже ранее скачали этот документ»), повторить шаг еще раз после некоторого перерыва, возможен сервис в данный момент перегружен.

№2.2 Сохранение даты InvList.ToDate в файл или в БД.

--- Промежуток времени, например сутки, неделя и т. д.

№3. Запрос GetList(дата прочитанная с диска, полученная на шаге 2.2)

№3.1 Скачивание и сохранение поступивших ЭСЧФ для дальнейшей обработки (подписания). Если в процессе скачивания были ошибки (за исключением ошибки «вы уже ранее скачали этот документ»), повторить шаг еще раз после некоторого перерыва, возможен сервис в данный момент перегружен.

№3.2 Сохранение даты InvList.ToDate в файл или в БД.

На каждом шаге скачиваются все новые ЭСЧФ, пришедшие в адрес авторизовавшегося УНП.

Внимание! В состав дистрибутивного пакета входит приложение simpleapp, в файле eInvVat\simpleapp\vbs\src\downloadRecvDir.vbs продемонстрирована правильная организация процесса скачивания ЭСЧФ на языке программирования Visual Basic Script с учетом необходимости обработки ошибок.

## 5.2. Добавление подписи и направление ЭСЧФ на портал

Получатель:

1. Создаёт экземпляр класса EInvVatService.Connector для взаимодействия с АС портала и авторизуется с использованием личного сертификата.

 Set EVatService = CreateObject("EInvVatService.Connector")

 EVatService.Login("", 0)

1. Загружает подписанный ЭСЧФ из файла, скачанного ранее.

 set InvVatXml = EVatService.CreateEDoc

 InvVatXml.Document.LoadFromFile(FileName)

1. Подписывает счет-фактуру, добавляя новую ЭЦП.

 InvVatXml.Sign(0)

1. Устанавливает соединение с АС портала и направляет подписанный документ на портал.

 EVatService.Connect(serviceUrl)

 EVatService.SendEDoc(InvVatXml)

1. Получает квитанцию, проверяет подпись и проверяет, что документ принят успешно, в противном случае выводит сообщение сервиса.

 set InvVatTicket = EVatService.Ticket

 if InvVatTicket.Accepted <> 0 then

 WScript.Echo "Документ не принят: " + InvVatTicket.Message

 end if

Внимание! В состав дистрибутивного пакета входит приложение simpleapp, в файле eInvVat\simpleapp\vbs\src\verifySignAndUploadDir.vbs продемонстрирована правильная организация процесса запроса в автоматизированном сервисе статуса обработки ЭСЧФ на языке программирования Visual Basic Script с учетом необходимости обработки ошибок.

## 5.3. Получение статуса поданного ЭСЧФ

Получатель:

1. Создаёт экземпляр класса EInvVatService.Connector для взаимодействия с АС портала и авторизуется с использованием личного сертификата.

 Set EVatService = CreateObject("EInvVatService.Connector")

 EVatService.Login("", 0)

1. Устанавливает соединение с АС портала и запрашивает статус счета-фактуры по его номеру.

 EVatService.Connect(objArgs(0))

 Set InvVatStatus = EVatService.GetStatus(InvVatNumber)

 WScript.Echo "Статус: " + InvVatStatus.status

Описание статусов ЭСЧФ приведено в Приложении 2.

Внимание! В состав дистрибутивного пакета входит приложение simpleapp, в файле eInvVat\simpleapp\vbs\src\updateStatusDir.vbs продемонстрирована правильная организация процесса запроса в автоматизированном сервисе статуса обработки ЭСЧФ на языке программирования Visual Basic Script с учетом необходимости обработки ошибок.

# 6. ActiveX компонент EInvVatService

 Компонент EInvVatService предназначен для генерации, проверки и отправки в службу регистрации электронных документов. Компонент предназначен для использования в операционных системах MS Windows XP — Windows 10. Для работы необходимо установленное ПО: криптопровайдер AvestCSP и компонента ActiveX EInvVatService, в справочниках сертификатов Windows должны быть установлены требуемые сертификаты и СОС, должен быть установлен носитель с личным ключом пользователя. Компонент EInvVatService содержит два основных COM объекта – Connector и EDocument.

## 6.1. Класс Connector

Класс Connector обеспечивает авторизацию с личным сертификатом, создание объектов EDocument, подключение к службе регистрации электронных документов.

Методы и свойства класса:

* Login — авторизация;
* Logout — завершение авторизованной сессии;
* Connect — подключение к службе регистрации электронных документов;
* Disconnect — завершение соединения со службой регистрации электронных документов;
* CreateEDoc — создание объекта EDocument;

* [SendEDo](https://projects.avest.by:4443/prt/prj/einvoice/wiki/PostDoc)с — отправка документа;
* GetStatus — получение статуса обработки документа сервисом;
* Ticket — квитанция о приёме документа;
* GetList — получение списка номеров поступивших документов;
* Get[EDo](https://projects.avest.by:4443/prt/prj/einvoice/wiki/PostDoc)с — получение документа из сервиса;
* SetProxy — установка параметров прокси сервера;
* GetMyCertProperty — чтение атрибута сертификата авторизовавшегося пользователя;
* SetServiceProperty — установка параметров объекта Connector;
* LastError —информация о произошедшей ошибке;
* Version — версия компонента ActiveX.

### Метод Login

Метод предназначен для авторизации с личным ключом и сертификатом.

Входные параметры:

* ConnectStr — строка с параметрами подключения к хранилищу сертификатов, строка может быть пустой.

Параметры передаются в строке в виде ключ=значение;ключ=…

Поддерживаемые ключи:

* + UNP — УНП юридического лица из сертификата открытого ключа;
	+ PUB\_KEY\_ID — идентификатор открытого ключа сертификата для авторизации в шестнадцатеричном виде;
	+ COMMON\_NAME — атрибут Common Name из сертификата;
	+ PASSWORD\_KEY — пароль к личному ключу.

Данные параметры предназначены для выбора личного сертификата для авторизации и ввода пароля без вывода интерактивного окна ввода пароля.

* Flags — набор флагов авторизации:
	+ 0x40 — не выводить окно с выбором сертификата, если найден только один подходящий сертификат.

Результат:

В случае успешного выполнения возвращается 0, иначе возвращается код ошибки. Подробности ошибки можно получить используя свойство LastError объекта Connector.

Пример

EVatService.Login.ConnectStr("PUB\_KEY\_ID=40D0C8DA3AB10ED832810FB4337FFB73A7009E0C;PASSWORD\_KEY=12345678", 0);

В данном примере идентификатор ключа субъекта в сертификате равен 40D0C8DA3AB10ED832810FB4337FFB73A7009E0C, а пароль к контейнеру с личным ключом 12345678.

Вариант с указанием атрибута из COMMON\_NAME сертификата:

EVatService.Login.ConnectStr("COMMON\_NAME=ЗАО "АВЕСТ";PASSWORD\_KEY=12345678", 0);

### Метод Logout

Метод предназначен для завершения авторизованной сессии.

Метод не имеет входных параметров.

Результат:

В случае успешного выполнения возвращается 0, иначе возвращается код ошибки. Подробности ошибки можно получить используя свойство LastError объекта Connector.

### Метод Connect

Подключение к автоматизированному сервису портала ЭСЧФ.

Входные параметры:

* URL — строка с адресом подключения к службе, например “https://vat.gov.by/”;

Результат:

В случае успешного выполнения возвращается 0, иначе возвращается код ошибки. Подробности ошибки можно получить, используя свойство LastError объекта Connector.

### Метод Disconnect

Закрытие подключения к службе приёма электронных документов.

Метод не имеет входных параметров.

Результат:

В случае успешного выполнения возвращается 0, иначе возвращается код ошибки. Подробности ошибки можно получить используя свойство LastError объекта Connector.

### Метод CreateEDoc

Метод предназначен для создания объекта EDocument.

Входные параметры:

* Flags — набор флагов. Зарезервировано. Необходимо передавать 0.

Результат:

В случае успешного выполнения возвращается объект электронный документ, иначе возвращается пустой объект. Подробности ошибки можно получить, используя свойство LastError объекта Connector.

### Метод [SendEDo](https://projects.avest.by:4443/prt/prj/einvoice/wiki/PostDoc)с

Отправка подписанного электронного документа в службу регистрации электронных документов.

Входные параметры:

* Doc — отправляемый электронный документ — подписанный ЭСЧФ.

Результат:

В случае успешного выполнения возвращается 0, иначе возвращается код ошибки. Подробности ошибки можно получить используя свойство LastError объекта Connector.

В случае успешной отправки, в свойстве Ticket объекта Connector будет доступен электронный документ с квитанцией о результате приёма.

### Метод Get[Status](https://projects.avest.by:4443/prt/prj/einvoice/wiki/PostDoc)

Получение текущего статуса обработки электронного счета-фактуры автоматизированным сервисом.

Входные параметры:

* Number — строка с уникальным номером искомого счета-фактуры.

Результат:

В случае успешного выполнения возвращается объект электронный документ, содержащий объект «информация о статусе документа», иначе возвращается пустой объект. Подробности ошибки можно получить, используя свойство LastError объекта Connector.

### Свойство Ticket

Свойство для доступа к квитанции, полученной при регистрации электронного документа. Данное свойство доступно после успешного вызова метода SendEDoc.

Результат:

Значением свойства является объект Ticket, содержащий электронный документ с квитанцией о регистрации.

### Метод GetList

Получение списка поступивших в АС портала в адрес получателя ЭСЧФ. Будут найдены документы, зарегистрированные на портале начиная с переданной в параметре даты и времени по текущее время сервера.

Входные параметры:

* FromDate — дата периода начала поиска документов. Дата должна быть представлена в следующем формате "yyyy-MM-dd'T'HH:mm:ss", например, “2016-08-25T16:40:00”, где символ ‘T’ – это латинская буква с ASCII кодом 0x54. Время должно соответствовать часовому поясу Республики Беларусь.

Результат:

В случае успешного выполнения возвращается объект электронный документ, содержащий объект EDocList со списком реквизитов ЭСЧФ, иначе возвращается пустой объект. Подробности ошибки можно получить, используя свойство LastError объекта Connector.

После получения и обработки списка документов необходимо у списка получить атрибут toDate — формирования списка автоматизированным сервисом и сохранить на диск, или в базу данных, для подачи в метод GetList при следующем обращении.

### Метод Get[EDo](https://projects.avest.by:4443/prt/prj/einvoice/wiki/PostDoc)с

Получение подписанного электронного документа из службы регистрации электронных документов.

Входные параметры:

* Number — строка с уникальным номером искомого счета-фактуры.

Результат:

В случае успешного выполнения возвращается объект электронный документ, содержащий подписанный счет-фактуру НДС, иначе возвращается пустой объект. Подробности ошибки можно получить используя свойство LastError объекта Connector.

### Метод GetMyCertProperty

Метод чтения атрибутов сертификата авторизовавшегося пользователя. Метод позволяет прочитать атрибуты сертификата в виде строки.

Входные параметры:

* OID — строка с объектным идентификатором атрибута.

Возможные значения:

* + 2.5.4.10 — Наименование организации владельца открытого ключа;
	+ 1.2.112.1.2.1.1.1.1.2 — УНП организации;
	+ 1.3.6.1.4.1.12656.106.101 — УНП организации;
	+ 2.5.4.4 — Фамилия;
	+ 2.5.4.41 — Имя, Отчество;
	+ 1.3.6.1.5.5.7.9.1 — Дата рождения;
	+ 2.5.4.6 — Страна;
	+ 2.5.4.8 — Область;
	+ 2.5.4.7 — Населенный пункт;
	+ 2.5.4.11 — Подразделение;
	+ 2.5.4.12 — Должность;
	+ 1.3.6.1.4.1.12656.5.1 — Место работы и должность;
	+ 1.3.6.1.4.1.12656.5.3 — Данные из документа, удостоверяющего личность;
	+ 2.5.4.9 — Адрес;
	+ 2.5.4.3 — Общие данные;
	+ 1.2.840.113549.1.9.1 — Адрес электронной почты.
* Flags — набор флагов. Зарезервировано. Необходимо передавать 0.

УНП организации присваивается один из двух OID в зависимости от того, какой УЦ выдал сертификат.

Результат:

В случае успешного выполнения возвращается значение атрибута в виде строки. В случае ошибки возвращается пустая строка. Подробности ошибки можно получить используя свойство LastError объекта Connector.

### Метод SetProxy

Метод для установки параметров прокси сервера. Метод задает параметры подключения через прокси сервер в случае необходимости использования прокси сервера. Данный метод необходимо вызывать перед методом Connect.

Входные параметры:

* Host — строка с адресом прокси сервера;
* Port — порт прокси сервера, целое число;
* Username — строка с именем пользователя прокси сервера;
* Password — строка с паролем пользователя прокси сервера;
* ProxyType — тип прокси сервера, целое число:
	+ 1 — HTTP прокси;
	+ 4 — Socks прокси версии 4;
	+ 5 — Socks прокси версии 5.

Результат:

В случае успешного выполнения возвращается 0, иначе возвращается код ошибки. Подробности ошибки можно получить используя свойство LastError объекта Connector.

### Метод SetServiceProperty

Метод для установки параметров установка параметров объекта Connector. Метод задает параметры работы компонента ActiveX.

Входные параметры:

* ParamName — строка с наименованием параметра;
* ParamValue — значение параметра в виде строки;
* Flags — набор флагов. Зарезервировано. Необходимо передавать 0.

Поддерживаемые параметры:

* connection.readTimeout — таймаут чтения данных автоматизированного сервиса в секундах, значение по умолчанию «300».

Результат:

В случае успешного выполнения возвращается 0, иначе возвращается код ошибки. Подробности ошибки можно получить используя свойство LastError объекта Connector.

### Свойство LastError

Получение информации о последней произошедшей ошибке.

Значение свойства:

Строка с текстом описания произошедшей ошибки.

### Свойство Version

Получение информации о версии компонента ActiveX EInvVatService.

Значение свойства:

Строка с текстом версии компонента.

## 6.2. Класс EDocument

Класс EDocument предназначен для работы с электронными документами.

Методы и свойства класса:

* LoadFromFile — загрузка документа из файла;
* SaveToFile — сохранение документа в файл;
* GetData — получение документа (сериализация);
* SetData — установка данных документа;
* Document — свойство для доступа к вложенному документу;
* GetXmlNodeValue — получение значения XML узла документа;
* GetXmlNodeAttribute — получение значения атрибута XML узла документа;
* ValidateXml — проверка схемы XML документа по XSD схеме;
* Sign — выработка ЭЦП документа;
* GetSignCount — получение количества ЭЦП в документе;
* GetSignProperty — получение атрибута ЭЦП и сертификата подписавшего;
* VerifySign — проверка ЭЦП документа.

### Метод LoadFromFile

Загрузка документа из файла. Загрузить можно подписанный документ.

Для загрузки XML документа «электронный счет-фактура» необходимо вызвать метод LoadFromFile у объекта доступного через свойство Document.

Входные параметры:

* FileName — строка с именем файла для загрузки.

Результат:

В случае успешного выполнения возвращается 0, иначе возвращается код ошибки. Подробности ошибки можно получить используя свойство LastError объекта Connector.

### Метод SaveToFile

Сохранение электронного документа в файл. Документ сохраняется в кодировке UTF-8 без BOM. При необходимости сохранить вложенный в электронный документ счет-фактуру или квитанцию о приёме, метод необходимо вызывать у объекта, доступного через свойство Document.

Входные параметры:

* FileName — строка с именем файла для сохранения.
* Flags — набор флагов. Зарезервировано. Необходимо передавать 0.

Результат:

В случае успешного выполнения возвращается 0, иначе возвращается код ошибки. Подробности ошибки можно получить используя свойство LastError объекта Connector.

### Метод GetData

Получение документа как строки, представляющей XML код документа в кодировке UTF-8 без BOM.

Входные параметры:

* CodeType — формат данных. Поддерживаемые значения:
	+ 1 – текст в кодировке Base64;
	+ 2 – текст в шестнадцатеричной кодировке Hex.

Результат:

В случае успешного выполнения возвращается строка с данными, иначе возвращается пустой объект. Подробности ошибки можно получить используя свойство LastError объекта Connector.

### Метод SetData

Загрузка документа из закодированной строки.

Входные параметры:

* Blob — входные данные.
* CodeType — формат выходных данных. Поддерживаемые значения:
	+ 1 – текст в кодировке Base64;
	+ 2 – текст в шестнадцатеричной кодировке Hex.

Результат:

В случае успешного выполнения возвращается 0, иначе возвращается код ошибки. Подробности ошибки можно получить, используя свойство LastError объекта Connector.

### Свойство Document

Свойство для доступа к внутреннему документу.

Значение свойства:

Значением свойства является объект EDocument, содержащий внутренний документ. Свойство Document доступно только после проверки ЭЦП документа. До проверки ЭЦП свойство Document возвращает пустой документ.

### Метод Document.GetXmlNodeValue

Получение значения узла XML документа. Метод позволяет получать значения неповторяющихся в XML документе узлов.

Входные параметры:

* Node — строка с путем к узлу, например “issuance/general/number”.

Результат:

В случае успешного выполнения возвращается строка с данными, иначе возвращается пустая строка.

### Метод Document.GetXmlNodeAttribute

Получение значения атрибута узла XML документа. Метод позволяет получать значения неповторяющихся в XML документе атрибутов узлов.

Входные параметры:

* Node — строка с путем к узлу и его атрибуту, например “issuance/sender”.

Результат:

В случае успешного выполнения возвращается строка с данными, иначе возвращается пустая строка.

### Метод Document.ValidateXml

Проверка XML документа на соответствие XML схеме.

Входные параметры:

* XSD — строка с именем файла содержащего XSD схему.
* Flags — набор флагов. Зарезервировано. Необходимо передавать 0.

Результат:

В случае успешного выполнения возвращается 0, иначе возвращается код ошибки.

### Метод Sign

Выработка ЭЦП документа. Если объект содержал XML документ, объект будет содержать электронный документ с ЭЦП. Если объект уже содержал электронный документ с подписью, то в него будет добавлена вторая ЭЦП.

Входные параметры:

* Flags — набор флагов. Зарезервировано. Необходимо передавать 0.

Результат:

В случае успешного выполнения возвращается 0, иначе возвращается код ошибки. Подробности ошибки можно получить, используя свойство LastError объекта Connector.

### Метод GetSignCount

Получение количества ЭЦП в электронном документе.

Входные параметры:

Метод не имеет входных параметров.

Результат:

В случае успешного выполнения возвращается количество ЭЦП в документе.

### Метод GetSignProperty

Метод позволяет получить атрибут подписи, либо сертификата подписавшего.

Входные параметры:

* IDX — целое число, индекс подписи, первая подпись соответствует индексу 0.
* OID — строка с идентификатором требуемого атрибута подписи. Если в параметре передано значение «SIGNDATE», будет возвращена дата подписания документа, иначе в параметре необходимо передать OID атрибута сертификата подписавшего.

Список атрибутов из поля владелец (subject) сертификата:

* + 2.5.4.3 — общее имя (CommonName);
	+ 2.5.4.10 — наименование организации (organizationName);
	+ 2.5.4.11 — подразделение организации (organizationalUnitName);
	+ 2.5.4.6 — код страны (countryName);
	+ 2.5.4.7 — населенный пункт (localityName);
	+ 2.5.4.8 — область(stateOrProvinceName);
	+ 2.5.4.9 — улица (streetAddress);
	+ 2.5.4.4 — фамилия ответственного лица (surname);
	+ 2.5.4.41 — имя ответственного лица (name).

Возможно получение и других атрибутов из сертификата подписавшего.

* Flags — набор флагов. Зарезервировано. Необходимо передавать 0.

Результат:

В случае успешного выполнения возвращается требуемый атрибут в виде строки. Подробности ошибки можно получить, используя свойство LastError объекта Connector.

### Метод VerifySign

Проверка выбранной ЭЦП в документе.

Входные параметры:

* IDX — целое число, индекс подписи, первая подпись соответствует индексу 0.
* Flags — набор флагов. Зарезервировано. Необходимо передавать 0.

Результат:

В случае успешного выполнения возвращается 0, иначе возвращается код ошибки. Подробности ошибки можно получить, используя свойство LastError объекта Connector.

## 6.3. Класс Ticket

Класс Ticket предназначен для работы с квитанцией приёма ЭСЧФ. Квитанция является электронным документом, имеющим методы и свойства, аналогичные классу EDocument.

Методы и свойства класса:

* Accepted — проверка ЭЦП;
* Message — информационное сообщение о статусе документа;

### Свойство Accepted

Свойство для доступа к данным квитанции — был ли документ принят или нет.

Результат:

Значение свойства – целое число, 0 – документ успешно принят, иначе код ошибки. Подробности ошибки можно получить, используя свойство LastError объекта Connector.

### Свойство Message

Свойство для получения текстовой информации из квитанции.

Значение свойства:

Значением свойства является строка с текстовой информацией.

## 6.4. Класс StatusInfo

Класс StatusInfo предназначен для работы с информацией о статусе обработки ЭСЧФ. Статус является электронным документом, имеющим методы и свойства, аналогичные классу EDocument.

Методы и свойства класса:

* Verify — проверка ЭЦП;
* Status — статус документа;
* Message — информационное сообщение о статусе документа;
* Since\* — дата начала действия статуса документа;

\* В настоящее время при запросе статуса ЭСЧФ (посредством метода since класса statusinfo) возвращаются следующие даты:

а) при статусе «Выставлен» или «На согласовании» - дата выставления ЭСЧФ на Портал;

б) при статусе «Выставлен. Подписан получателем» - дата подписания ЭСЧФ получателем;

в) при статусе «Выставлен. Аннулирован поставщиком» - дата выставления ЭСЧФ поставщиком на Портал.

г) при статусе «Аннулирован» - дата аннулирования ЭСЧФ, т. е. дата совершения операции из ЭСЧФ, либо системная дата, которой проводилось аннулирование.

### Метод Verify

Проверка ЭЦП документа «статус обработки ЭСЧФ».

Метод не имеет входных параметров.

Результат:

В случае успешного выполнения возвращается 0, иначе возвращается код ошибки. Подробности ошибки можно получить используя свойство LastError объекта Connector.

### Свойство Status

Свойство для получения статуса обработки ЭСЧФ порталом.

Значение свойства:

Значением свойства является строка со статусом, например COMPLETED.

### Свойство Message

Свойство для получения текстовой информации статуса.

Значение свойства:

Значением свойства является строка с текстовой информацией.

### Свойство Since

Свойство для получения даты и времени начала действия статуса.

Значение свойства:

Значением свойства является строка с описанием даты и времени.

## 6.5. Класс EDocList

Класс EDocList предназначен для работы со списком ЭСЧФ, поступивших в адрес получателя. Список позволяет получить реквизиты ЭСЧФ для скачивания метододом getEDoc. Список является электронным документом, имеющим методы и свойства, аналогичные классу EDocument.

Методы и свойства класса:

* Verify — проверка ЭЦП;
* ToDate — дата выполнения запроса поиска;
* Сount — количество найденных документов;
* GetItemAttribute — получение реквизитов найденных документов;

### Метод Verify

Проверка ЭЦП документа «список поступивших ЭСЧФ».

Метод не имеет входных параметров.

Результат:

В случае успешного выполнения возвращается 0, иначе возвращается код ошибки. Подробности ошибки можно получить используя свойство LastError объекта Connector.

### Свойство ToDate

Свойство для чтения даты и времени выполнения сервисом запроса поиска счетов-фактур. Используется для сохранения и передачи в следующем вызове метода GetList.

Результат:

Дата и время, возвращенное сервисом приёма и обработки электронных счетов-фактур в виде строки.

### Свойство Count

Свойство для доступа к количеству найденных документов.

Результат:

Целое число, большее или равное 0.

### Метод GetItemAttribute

Метод доступа к реквизитам найденных документов.

Входные параметры:

* Index — номер документа из списка. Число от 0 до List.Count – 1.
* Attribute — наименование требуемого реквизита, например “document/number”

Результат:

В случае успешного выполнения возвращается строка со значением требуемого атрибута. В случае ошибки возвращается пустая строка. Подробности ошибки можно получить используя свойство LastError объекта Connector.

# 7. Использование компонента Java для взаимодействия с АС портала на стороне отправителя

Приведенный в следующих подразделах код показывает какие методы компонента Java и в какой последовательности следует вызывать для взаимодействия с АС портала на стороне отправителя. Данный код приводится на языке программирования Java без операторов обработки ошибок и других незначащих элементов. Полный текст примеров содержится в прилагаемых файлах.

## 7.1. Подпись и направление ЭСЧФ на портал

Отправитель:

1. Формирует в учетной системе электронный счет-фактуру НДС и сохраняет в виде XML файла.
2. Создаёт экземпляр класса EVatService для взаимодействия с АС портала и авторизуется с использованием личного сертификата.

 EVatService service = new EVatService(serviceUrl, new KeyInteractiveSelector());

 service.login();

1. Формирует электронный документ и проверяет структуру сформированного XML документа на соответствие XSD схеме.

 AvEDoc eDoc = service.createEDoc();

 eDoc.getDocument().load(readFile(new File(documentFileName)));

 boolean isDocumentValid = eDoc.getDocument().validateXML(xsdSchema);

1. Подписывает счет-фактуру, формируя электронный документ, содержащий счет-фактуру (issuance).

 eDoc.sign();

1. Устанавливает соединение с АС портала и направляет подписанный отправителем счет-фактуру на портал.

 service.connect();

 AvETicket ticket = service.sendEDoc(eDoc);

1. Получает квитанцию, проверяет подпись и проверяет, что счет-фактура принят успешно, в противном случае выводит сообщение сервиса.

if (ticket.accepted()) {

 String resultMessage = ticket.getMessage();

 System.out.println("Счет/фактура НДС принята к обработке: " + resultMessage);

} else {

 AvError err = ticket.getLastError();

 System.err.println("Ошибка: ЭСЧФ не принят. Сообщение: "+ err.getMessage() + ".");

}

Внимание! В состав дистрибутивного пакета входит приложение simpleapp, в файле eInvVat\simpleapp\vbs\src\signAndUploadDir.vbs продемонстрирована правильная организация процесса подписания и отправки на автоматизированный сервис ЭСЧФ на языке программирования Visual Basic Script с учетом необходимости обработки ошибок.

## 7.2. Получение статуса поданного ЭСЧФ

Отправитель:

1. Создаёт экземпляр класса EVatService для взаимодействия с АС портала и авторизуется с использованием личного сертификата.

 EVatService service = new EVatService(serviceUrl, new KeyInteractiveSelector());

 service.login();

1. Устанавливает соединение с АС портала.

 service.connect();

1. Запрашивает статус счета-фактуры по его номеру.

 AvEStatus documentStatus = service.getStatus(invoiceNum);

1. Проверяет ЭЦП и получает статус из документа.

 boolean isValid = documentStatus.verify();

 if (isValid) {

 System.out.println("Статус: " + documentStatus.getStatus());

 System.out.println("Сообщение: " + documentStatus.getMessage());

 System.out.println("Дата изменения статуса: " + documentStatus.getSince());

 } else {

 System.err.println("Ошибка: получен некорректный статус: " +

 documentStatus.getLastError().getMessage());

 }

Описание статусов ЭСЧФ приведено в Приложении 2.

Внимание! В состав дистрибутивного пакета входит приложение simpleapp, в файле eInvVat\simpleapp\vbs\src\updateStatusDir.vbs продемонстрирована правильная организация процесса запроса в автоматизированном сервисе статуса обработки ЭСЧФ на языке программирования Visual Basic Script с учетом необходимости обработки ошибок.

# 8. Использование компонента Java для взаимодействия с АС портала на стороне получателя

Приведенный в следующих подразделах код показывает какие методы компонента Java и в какой последовательности следует вызывать для взаимодействия с АС портала на стороне получателя. Данный код приводится на языке программирования Java без операторов обработки ошибок и других незначащих элементов. Полный текст примеров содержится в прилагаемых файлах.

## 8.1. Получение счетов-фактур, поступивших в адрес получателя

Получатель:

1. Создаёт экземпляр класса EVatService для взаимодействия с АС портала и авторизуется с использованием личного сертификата.

 EVatService service = new EVatService(serviceUrl, new KeyInteractiveSelector());

 service.login();

1. Устанавливает соединение с АС портала и запрашивает список поступивших счетов-фактур за период. Параметр fromDate — это дата и время в строковом виде, данный параметр означает, с какой даты требуется запрашивать счета-фактуры. Будут запрошены документы от этой даты по текущий момент. Формат строки даты описан в разделе описания метода list класса EVatService.

 service.connect();

 AvEList list = service.list(fromDate);

1. Проверяет ЭЦП списка.

 boolean isValid = list.verify();

1. Получает все поступившие ЭСФЧ и сохраняет их.

 while (invoices.hasNext()) {

 String invoiceNum = (String) invoices.next();

 // необходимо проверить, что ЭСЧФ с таким номером уже сохранен на диск или в БД

 // или в учетную систему и не скачивать повторно скачанный ранее ЭСЧФ

 if(!ужеСохраненаЛокально(invoiceNum) {

 AvEDoc eDocXml = service.getEDoc(invoiceNum);

 isValid = verifyDocumentSign(eDocXml);

 if (isValid) {

 // обязательно сохранить скачанный ЭСЧФ на диск,

 // или в учетную систему пользователя

 writeFile(docFile, eDocXml.getEncoded());

 }

 }

 }

Метод проверки ЭЦП скачанного ЭСЧФ (в примере выше названный verifyDocumentSign) должен действовать следующим образом: пройти по списку ЭЦП подписанного документа, если среди подписей есть ЭЦП автоматизированного сервиса портала, то проверить только найденную ЭЦП, если среди подписей нет ЭЦП АС портала, то проверить все подписи в документе. Определить, что подпись сделана АС портала можно прочитав атрибуты сертификата вызовом метода GetSignProperty. Атрибуты сертификата АС портала следующие:

|  |  |
| --- | --- |
| Атрибут  | Значение |
| 2.5.4.3 (commonName) | Автоматизированный сервис портала АИС УСФ |

Скачанные документы необходимо сохранять на компьютере пользователя.

1. Если все документы были успешно получены (за исключением ошибки «вы уже ранее скачали этот документ»), необходимо сохранить в файл, или в базу данных, или в учетную систему пользователя дату и время списка ЭСЧФ для следующего запроса EVatService.list.

 Date toDate = list.getToDate();

 // сохранение в файл, в БД, или в учетную систему

Тем самым формируется следующая последовательность обращений за получением поступающих ЭСЧФ:

№1. Запрос list (дата пустая, поскольку первое обращение к сервису)

№1.1 Скачивание и сохранение поступивших ЭСЧФ для дальнейшей обработки (подписания). Если в процессе скачивания были ошибки (за исключением ошибки «вы уже ранее скачали этот документ»), повторить шаг еще раз после некоторого перерыва, возможен сервис в данный момент перегружен.

№1.2 Сохранение даты InvList. getToDate() в файл или в БД.

--- Промежуток времени, например сутки, неделя и т. д.

№2. Запрос list (дата прочитанная с диска, полученная на шаге 1.2)

№2.1 Скачивание и сохранение поступивших ЭСЧФ для дальнейшей обработки (подписания). Если в процессе скачивания были ошибки (за исключением ошибки «вы уже ранее скачали этот документ»), повторить шаг еще раз после некоторого перерыва, возможен сервис в данный момент перегружен.

№2.2 Сохранение даты InvList. getToDate() в файл или в БД.

--- Промежуток времени, например сутки, неделя и т. д.

№3. Запрос list (дата прочитанная с диска, полученная на шаге 2.2)

№3.1 Скачивание и сохранение поступивших ЭСЧФ для дальнейшей обработки (подписания). Если в процессе скачивания были ошибки (за исключением ошибки «вы уже ранее скачали этот документ»), повторить шаг еще раз после некоторого перерыва, возможен сервис в данный момент перегружен.

№3.2 Сохранение даты InvList. getToDate() в файл или в БД.

На каждом шаге скачиваются все новые ЭСЧФ, пришедшие в адрес авторизовавшегося УНП.

Внимание! В состав дистрибутивного пакета входит приложение simpleapp, в файле eInvVat\simpleapp\vbs\src\downloadRecvDir.vbs продемонстрирована правильная организация процесса скачивания ЭСЧФ на языке программирования Visual Basic Script с учетом необходимости обработки ошибок.

## 8.2. Добавление подписи и направление ЭСЧФ на портал

Получатель:

1. Создаёт экземпляр класса EVatService для взаимодействия с АС портала и авторизуется с использованием личного сертификата.

 EVatService service = new EVatService(serviceUrl, new KeyInteractiveSelector());

 service.login();

1. Загружает подписанную ЭСЧФ из файла, скачанного ранее.

 AvEDoc eDocXml = service.createEDoc();

 eDocXml.load(readFile(new File(documentFileName)));

1. Подписывает счет-фактуру, добавляя новую ЭЦП.

 eDoc.sign();

1. Устанавливает соединение с АС портала и направляет подписанный документ на портал.

 service.connect();

 AvETicket ticket = service.sendEDoc(eDoc);

1. Получает квитанцию, проверяет подпись и проверяет, что документ принят успешно, в противном случае выводит сообщение сервиса.

 if (ticket.accepted()) {

 String resultMessage = ticket.getMessage();

 System.out.println("Счет-фактура НДС принят к обработке: " + resultMessage);

 } else {

 AvError err = ticket.getLastError();

 System.err.println("Ошибка: ЭСЧФ не принят. Сообщение: "+ err.getMessage() + ".");

 }

Внимание! В состав дистрибутивного пакета входит приложение simpleapp, в файле eInvVat\simpleapp\vbs\src\verifySignAndUploadDir.vbs продемонстрирована правильная организация процесса запроса в автоматизированном сервисе статуса обработки ЭСЧФ на языке программирования Visual Basic Script с учетом необходимости обработки ошибок.

## 8.3. Получение статуса поданного ЭСЧФ

Получатель:

1. Создаёт экземпляр класса EVatService для взаимодействия с АС портала и авторизуется с использованием личного сертификата.

 EVatService service = new EVatService(serviceUrl, new KeyInteractiveSelector());

 service.login();

1. Устанавливает соединение с АС портала.

 service.connect();

1. Запрашивает статус счета-фактуры по его номеру.

 AvEStatus documentStatus = service.getStatus(invoiceNum);

1. Проверяет ЭЦП и получает статус из документа.

 boolean isValid = documentStatus.verify();

 if (isValid) {

 System.out.println("Статус: " + documentStatus.getStatus());

 System.out.println("Сообщение: " + documentStatus.getMessage());

 System.out.println("Дата изменения статуса: " + documentStatus.getSince());

 } else {

 System.err.println("Ошибка: получен некорректный статус: " +

 documentStatus.getLastError().getMessage());

 }

Описание статусов ЭСЧФ приведено в Приложении 2.

Внимание! В состав дистрибутивного пакета входит приложение simpleapp, в файле eInvVat\simpleapp\vbs\src\updateStatusDir.vbs продемонстрирована правильная организация процесса запроса в автоматизированном сервисе статуса обработки ЭСЧФ на языке программирования Visual Basic Script с учетом необходимости обработки ошибок.

# 9. Программный интерфейс EVatService для языка Java

Программный интерфейс EVatService предназначен для генерации, проверки и отправки в службу регистрации электронных документов. Компонент предназначен для использования в программах на языке Java в операционных системах MS Windows XP — Windows 10, Linux, совместимых со средой выполнения Java Runtime (JRE) и программным компонентом AvJCEProv. Библиотека содержит набор классов для выполнения задач взаимодействия с автоматизированным сервисом. Должен быть установлен криптопровайдер AvJCEProv, импортированы сертификаты и списки отозванных сертификатов.

Перед обращением к классам интерфейса EVatService необходимо зарегистрировать криптопровайдер:

 ProviderFactory.addAvUniversalProvider();

 Security.addProvider(new AvTLSProvider());

 Security.addProvider(new AvCertStoreProvider());

## 9.1. Класс EVatService

Класс EVatService обеспечивает авторизацию с личным сертификатом, создание объектов AvEDoc, подключение к службе регистрации электронных документов.

Методы класса:

* EVatService — конструктор;
* login — авторизация;
* logout — завершение авторизованной сессии;
* connect — подключение к службе регистрации электронных документов;
* disconnect — завершение соединения со службой регистрации электронных документов;
* createEDoc — создание объекта EDocument;

* [sendEDo](https://projects.avest.by:4443/prt/prj/einvoice/wiki/PostDoc)с — отправка документа;
* getStatus — получение статуса обработки ЭСЧФ;
* getList — получение списка реквизитов поступивших ЭСЧФ;

* [getEDo](https://projects.avest.by:4443/prt/prj/einvoice/wiki/PostDoc)с — скачивание ЭСЧФ с сервиса;
* getMyCertProperty — чтение атрибута сертификата авторизовавшегося пользователя;
* setServiceProperty — установка параметров объекта EVatService.

### Конструктор EVatService

Конструктор предназначен для подготовки к работе библиотеки.

Входные параметры:

* location — URL подключения к автоматизированному сервису портала;
* keyManager — экземпляр класса, управляющего выбором контейнера личного ключа и ввода пароля к личному ключу. Должен реализовывать интерфейс X509KeyManager.

При использовании прокси-сервера для подключения к автоматизированному сервису портала, адрес прокси-сервера должен передаваться через JAVA переменные:

• https.proxyHost -хост прокси-сервера;

• https.proxyPort - порт прокси-сервера;

В случае необходимости аутентификации на прокси-сервере, одним из способов передачи реквизитов пользователя является реализация абстрактного класса java.net.Authenticator. В приведенном примере логин и пароль передаются посредство JAVA-переменных:

 Authenticator.setDefault(new Authenticator() {

 @Override

 public PasswordAuthentication getPasswordAuthentication() {

 return new PasswordAuthentication(System.getProperty("https.proxyUser"),

 System.getProperty("https.proxyPass").toCharArray());

 }

 });

### Метод login

Метод предназначен для авторизации с личным ключом и сертификатом.

Входные параметры:

* ConnectStr — параметры подключения к хранилищу сертификатов;

Параметры передаются в строке в виде ключ=значение;ключ=…

Поддерживаемые ключи:

* + UNP — УНП юридического лица из сертификата открытого ключа;
	+ PUB\_KEY\_ID — идентификатор открытого ключа сертификата для авторизации в шестнадцатеричном виде;
	+ COMMON\_NAME — атрибут Common Name из сертификата;
	+ PASSWORD\_KEY — пароль к личному ключу.

Данные параметры предназначены для выбора личного сертификата для авторизации и ввода пароля без вывода интерактивного окна ввода пароля.

Результат:

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод logout

Метод предназначен для завершения авторизованной сессии.

Метод не имеет входных параметров.

Результат:

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод connect

Подключение к автоматизированному сервису портала ЭСЧФ.

Метод не имеет входных параметров.

Результат:

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод disconnect

Закрытие подключения к службе приёма электронных документов.

Метод не имеет входных параметров.

Результат:

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод createEDoc

Метод предназначен для создания объекта EDocument

Метод не имеет входных параметров.

Результат:

Метод возвращает созданный объект AvEDoc.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод [sendEDo](https://projects.avest.by:4443/prt/prj/einvoice/wiki/PostDoc)с

Отправка подписанного электронного документа в службу регистрации электронных документов.

Входные параметры:

* eDoc — отправляемый электронный документ.

Результат:

Метод возвращает объект AvETicket — полученную от сервиса квитанцию о приёме, подлежащую дальнейшей валидации.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод getStatus

Получение статуса обработки электронного счета-фактуры порталом.

Входные параметры:

* invNum — номер счета-фактуры.

Результат:

Метод возвращает объект AvEStatus — полученный от сервиса объект «статус» , подлежащую дальнейшей валидации.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод getList

Получение списка номеров поступивших ЭСЧФ для дальнейшего скачивания с портала. Будут найдены документы, зарегистрированные на портале начиная с переданной в параметре даты и времени по текущее время сервера.

Входные параметры:

* date — дата периода начала поиска документов. Дата должна быть представлена в следующем формате "yyyy-MM-dd'T'HH:mm:ss", например, “2016-08-25T16:40:00”, где символ ‘T’ – это латинская буква с ASCII кодом 0x54. Время должно соответствовать часовому поясу Республики Беларусь.

Результат:

Метод возвращает объект AvEList — полученный от сервиса объект «статус» , подлежащую дальнейшей валидации.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

После получения и обработки списка документов необходимо у списка получить атрибут toDate — формирования списка автоматизированным сервисом и сохранить на диск, или в базу данных, для подачи в метод GetList при следующем обращении.

### Метод [getEDo](https://projects.avest.by:4443/prt/prj/einvoice/wiki/PostDoc)с

Получение с портала счета-фактуры.

Входные параметры:

* invNum — номер счета-фактуры.

Результат:

Метод возвращает объект AvEDoc — полученный от сервиса объект ЭСЧФ.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод getMyCertProperty

Метод чтения атрибутов сертификата авторизовавшегося пользователя. Метод позволяет прочитать атрибуты сертификата в виде строки.

Входные параметры:

* OID — объектный идентификатор атрибута.

Возможные значения:

* + 2.5.4.10 — Наименование организации владельца открытого ключа;
	+ 1.2.112.1.2.1.1.1.1.2 — УНП организации;
	+ 1.3.6.1.4.1.12656.106.101 — УНП организации;
	+ 2.5.4.4 — Фамилия;
	+ 2.5.4.41 — Имя, Отчество;
	+ 1.3.6.1.5.5.7.9.1 — Дата рождения;
	+ 2.5.4.6 — Страна;
	+ 2.5.4.8 — Область;
	+ 2.5.4.7 — Населенный пункт;
	+ 2.5.4.11 — Подразделение;
	+ 2.5.4.12 — Должность;
	+ 1.3.6.1.4.1.12656.5.1 — Место работы и должность;
	+ 1.3.6.1.4.1.12656.5.3 — Данные из документа, удостоверяющего личность;
	+ 2.5.4.9 — Адрес;
	+ 2.5.4.3 — Общие данные;
	+ 1.2.840.113549.1.9.1 — Адрес электронной почты.

УНП организации присваивается один из двух OID в зависимости от того, какой УЦ выдал сертификат.

Результат:

В случае успешного выполнения возвращается значение атрибута в виде строки. В случае отсутствия в сертификате атрибута с заданным объектным идентификатором возвращается null. В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод setServiceProperty

Метод для установки параметров установка параметров объекта EVatService. Метод задает параметры работы компонента ActiveX.

Входные параметры:

* ParamName — наименование параметра;
* ParamValue — значение параметра в виде строки.

Поддерживаемые параметры:

* connection.readTimeout — таймаут чтения данных автоматизированного сервиса в секундах, значение по умолчанию «300».

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

## 9.2. Класс AvEDoc

Класс AvEDoc предназначен для работы с электронными документами, содержащими одну или несколько ЭЦП.

Методы и свойства класса:

* load — загрузка документа из массива байт;
* sign — подпись документа;
* verify — проверка подписей документа;
* getDocument — получение вложенного документа;
* getEncoded — получение документа (сериализация);
* getSignCount — получение количства ЭЦП в документе;
* getSignProperty — получение атрибута ЭЦП и сертификата подписавшего;
* verifySign — проверка ЭЦП документа.
* getLastError — получение информации о произошедшей ошибке;

### Метод load

Загрузка документа из массива байт. Загрузить можно подписанный документ.

Для загрузки XML документа «электронная счет фактура» без ЭЦП необходимо вызвать метод load у объекта класса AvDoc.

Входные параметры:

* docData — массив байт, содержащий данные электронного документа в кодировке UTF-8 без BOM.

Результат:

Метод не имеет возвращаемого результата.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод sign

Выработка ЭЦП документа. Если объект содержал XML документ, объект будет содержать электронный документ с ЭЦП. Если объект уже содержал электронный документ с подписью, то в него будет добавлена вторая ЭЦП.

Метод не имеет входных параметров.

Результат:

Метод не имеет возвращаемого результата.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод verify

Проверка всех ЭЦП электронного документа.

Метод не имеет входных параметров.

Результат:

Метод возвращает булевое значение.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод getDocument

Метод для доступа к внутреннему документу типа AvDoc.

Метод не имеет входных параметров.

Результат:

Значением метода является объект AvDoc, содержащий внутренний документ.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод getEncoded

Получение электронного документа в кодированном виде как массива байт. Документ сохраняется в кодировке UTF-8 без BOM.

Метод не имеет входных параметров.

Результат:

В случае успешного выполнения возвращается массив байт с данными, иначе возвращается null.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод getSignCount

Получение количества ЭЦП в электронном документе.

Входные параметры:

Метод не имеет входных параметров.

Результат:

В случае успешного выполнения возвращается количество ЭЦП в документе.

### Метод getSignProperty

Метод позволяет получить атрибут подписи либо сертификата подписавшего в документе.

Входные параметры:

* IDX — индекс подписи, первая подпись соответствует индексу 0.
* OID — идентификатор требуемого атрибута подписи. Если в параметре передано значение «SIGNDATE», будет возвращена дата подписания документа, иначе в параметре необходимо передать OID атрибута сертификата подписавшего.

Список атрибутов из поля владелец (subject) сертификата:

* + 2.5.4.3 — общее имя (CommonName);
	+ 2.5.4.10 — наименование организации (organizationName);
	+ 2.5.4.11 — подразделение организации (organizationalUnitName);
	+ 2.5.4.6 — код страны (countryName);
	+ 2.5.4.7 — населенный пункт (localityName);
	+ 2.5.4.8 — область(stateOrProvinceName);
	+ 2.5.4.9 — улица (streetAddress);
	+ 2.5.4.4 — фамилия ответственного лица (surname);
	+ 2.5.4.41 — имя отвественного лица (name).

Возможно получение и других атрибутов из сертификата подписавшего.

Результат:

В случае успешного выполнения возвращается требуемый атрибут в виде строки. Подробности ошибки можно получить используя метод getLastError.

### Метод verifySign

Проверка выбранной ЭЦП в документе.

Входные параметры:

* IDX — индекс подписи, первая подпись соответствует индексу 0.
* Flags — набор флагов. Зарезервировано. Необходимо передавать 0.

Результат:

В случае успешного выполнения возвращается 0, иначе возвращается код ошибки. Подробности ошибки можно получить используя метод getLastError.

### Метод getLastError

Получение информации о произошедшей ошибке.

Метод не имеет входных параметров.

Результат:

Метод возвращает объект класса AvError с описанием произошедшей ошибки либо null.

## 9.3. Класс AvETicket

Класс AvETicket предназначен для работы с квитанцией о приёме ЭСЧФ. Класс расширяет AvEDoc и содержит все методы AvEDoc.

Методы и свойства класса:

* accepted — проверка квитанции;
* getMessage — получение текстового описания квитанции;

### Метод accepted

Результат приёма документа ЭСЧФ автоматизированным сервисом портала.

Метод не имеет входных параметров.

Результат:

Булевое значение true/false.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод getMessage

Получение текстового сообщения квитанции.

Метод не имеет входных параметров.

Результат:

Строка с текстовым сообщением.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

## 9.4. Класс AvEStatus

Класс AvEStatus предназначен для работы с информацией о статусе обработки ЭСЧФ порталом. Класс расширяет AvEDoc и содержит все методы AvEDoc.

Методы и свойства класса:

* verify — проверка ЭЦП;
* getStatus — получение статуса ЭСЧФ;
* getMessage — получение текстового описания статуса;
* getSince — получение даты и времени начал действия статуса.

\* В настоящее время при запросе статуса ЭСЧФ (посредством метода getSince класса AvEStatus) возвращаются следующие даты:

а) при статусе «Выставлен» или «На согласовании» - дата выставления ЭСЧФ на Портал;

б) при статусе «Выставлен. Подписан получателем» - дата подписания ЭСЧФ получателем;

в) при статусе «Выставлен. Аннулирован поставщиком» - дата выставления ЭСЧФ поставщиком на Портал.

г) при статусе «Аннулирован» - дата аннулирования ЭСЧФ, т. е. дата совершения операции из ЭСЧФ, либо системная дата, которой проводилось аннулирование.

### Метод verify

Проверка ЭЦП документа.

Метод не имеет входных параметров.

Результат:

Булевое значение true/false.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод getStatus

Получение статуса обработки документа.

Метод не имеет входных параметров.

Результат:

Строка со значением статуса, например COMPLETED.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод getMessage

Получение текстового сообщения о статусе обработки документа порталом.

Метод не имеет входных параметров.

Результат:

Строка с текстовым сообщением.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод getSince

Получение даты и времени присвоения порталом статуса документу.

Метод не имеет входных параметров.

Результат:

Строка с датой и временем.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

## 9.5. Класс AvEList

Класс AvEList предназначен для работы со списком поступивших счетов-фактур. Класс расширяет AvEDoc и содержит все методы AvEDoc.

Методы и свойства класса:

* verify — проверка ЭЦП;
* getToDate — получение даты формирования списка;
* getCount — получение количества элементов в списке;
* getItemAttribute — получение атрибута элемента списка;

### Метод verify

Проверка ЭЦП документа.

Метод не имеет входных параметров.

Результат:

Булевое значение true/false.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод getToDate

Получение даты формирования списка.

Метод не имеет входных параметров.

Результат:

Строка со значением даты формирования списка. Эта дата используется для подачи в автоматизированный сервис при последующем запросе списка поступивших ЭСЧФ.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод getCount

Получение количества счетов-фактур в списке.

Метод не имеет входных параметров.

Результат:

Целое число – количество ЭСЧФ.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод getItemAttribute

Получение атрибута элемента списка.

Входные параметры:

* index — номер элемента списка;
* attr — наименование атрибута.

Результат:

Строка со значением атрибута.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

## 9.6. Класс AvDoc

Класс AvDoc предназначен для работы с вложенным в электронный документ XML документом.

Методы и свойства класса:

* load — загрузка документа из массива байт;
* getXmlNodeValue — получение элемента XML документа;
* validateXML —проверка документа на соответствие XML схеме;
* getEncoded — получение документа (сериализация);

### Метод load

Загрузка документа из массива байт. Загрузить можно электронный счет-фактуру НДС.

Входные параметры:

* documentEnc — массив байт, содержащий данные XML документа в кодировке UTF-8 без BOM.

Результат:

Метод не имеет возвращаемого результата.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод getXmlNodeValue

Получение значения узла или атрибута узла XML документа.

Входные параметры:

* nodePath — путь к узлу, например “issuance/general/number”, либо путь к атрибуту узла, например “issuance/@sender”.

Результат:

В случае успешного выполнения возвращается строка с данными, иначе возвращается пустая строка.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод validateXML

Проверка XML документа на соответствие XML схеме.

Входные параметры:

* XSD — массив байт, содержащий XSD схему.

Результат:

Возвращается булевое значение true/false.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

### Метод getEncoded

Получение документа в кодированном виде как массива байт. Документ сохраняется в кодировке UTF-8 без BOM.

Метод не имеет входных параметров.

Результат:

В случае успешного выполнения возвращается массив байт с данными, иначе возвращается null.

В случае ошибки генерируется исключение с описанием произошедшей ошибки.

## 9.7. Класс AvError

Класс AvError предназначен для работы с информацией об ошибках.

Методы и свойства класса:

* getError — получение объекта исключения;
* getMessage — получение текстового описания ошибки.

### Метод getError

Получение объекта возникшего исключения.

Метод не имеет входных параметров.

Результат:

Объект исключения.

### Метод getMessage

Получение текстового сообщения о произошедшей ошибке.

Метод не имеет входных параметров.

Результат:

Строка с текстовым сообщением.

# Приложение 1. Программное приложение simpleapp

В состав дистрибутивного пакета eInvVat входит приложение simpleapp в двух вариантах:

1. Приложение на языке программирования Visual Basic Script, использующее ActiveX компонент EInvVatService.
2. Приложение на языке программирования Java, использующее компонент Java avedocclient.

 Оба варианта обеспечивают одинаковую функциональность, их использование описано в документе «Клиентское ПО ЭСЧФ НДС. Руководство пользователя».

Приложение на языке программирования Visual Basic Script поставляется с исходным кодом, данный исходный код рекомендуется использовать как справочный при разработке приложений, осуществляющих взаимодействие с автоматизированным сервисом портала ЭСЧФ НДС. В коде приложения simpleapp использованы практически все возможности компонента ActiveX в том виде, в каком они необходимы для правильной работы с автоматизированным сервисом.

# Приложение 2. Статусы ЭСЧФ НДС

В процессе обработки ЭСЧФ автоматизированным сервисом портала счет-фактура может получать следующие статусы:

|  |  |  |
| --- | --- | --- |
| Номер п.п. | Статус | Описание |
| 1 |  COMPLETED  |  Выставлен |
| 2 |  COMPLETED\_SIGNED  |  Выставлен. Подписан получателем, |
| 3 |  ON\_AGREEMENT  |  На согласовании |
| 4 |  CANCELLED  |  Аннулирован |
| 5 |  ON\_AGREEMENT\_CANCEL  |  Выставлен. Аннулирован поставщиком |
| 6 |  IN\_PROGRESS  |  ЭСЧФ находится в обработке. Запросите статус повторно через 3 часа |
| 7 |  NOT\_FOUND  |  ЭСЧФ нет в базе или нет права для просмотра статуса/выгрузки документа |
| 8 |  ERROR  |  Ошибка при выставлении ЭСЧФ на портал |

# Термины и сокращения

ГосСУОК — Государственная система управления открытыми ключами;

Сертификат открытого ключа — электронный документ, содержащий открытый ключ, информацию о владельце ключа, области применения ключа, подписанный выдавшим его Удостоверяющим центром и подтверждающий принадлежность открытого ключа владельцу.

Автоматизированный сервис портала — в данном документе это сервис портала МНС РБ vat.gov.by по приёму и обработке ЭСЧФ НДС;

Список отозванных сертификатов — электронный документ, содержащий список сертификатов, отозванных данным удостоверяющим центром и заверенный его ЭЦП;

УНП — учётный номер плательщика;

Электронный документ — документ, зафиксированный на машинном носителе и содержащий идентифицированную информацию, подлинность которой удостоверена ЭЦП;

ЭСЧФ НДС — электронный счет-фактура НДС;

ASCII — ASCII (англ. American standard code for information interchange) — название кодировки, в которой некоторым распространённым печатным и непечатным символам английского языка сопоставлены числовые коды;

Base64 — схема кодирования бинарных данных в виде ASCII строки используя перетрансляцию данных в представление с основанием 64;

GUID — (англ. Globally Unique Identifier) — статистически уникальный 128-битный идентификатор;

Java — язык программирования;

JRE — среда выполнения для Java (Java Runtime Environment);

SOAP ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Simple Object Access Protocol — простой протокол доступа к объектам) — [протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) обмена структурированными сообщениями в распределённой вычислительной среде;

TCP/IP — набор сетевых протоколов передачи данных, используемых в сетях, включая сеть Интернет;

TLS — криптографические протоколы, обеспечивающие защищённую передачу данных между узлами в сети Интернет (Transport Layer Security);

URL — единый указатель ресурса (RFC 1738);

UTF-8 (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Unicode Transformation Format, 8-bit — «формат преобразования Юникода, 8-битный») — одна из общепринятых и стандартизированных кодировок текста, которая позволяет хранить символы [Юникода](https://ru.wikipedia.org/wiki/%D0%AE%D0%BD%D0%B8%D0%BA%D0%BE%D0%B4);

UTF-8  без BOM — кодировка UTF-8 без указания Byte Order Mark, то есть без специального символа Unicode, указывающего порядок байтов в документе;

XML — (англ. eXtensible Markup Language) — расширяемый язык разметки.