For POS Developers

Introduction

This article offers useful initial information about technical requirements for accrediting POS solutions.

NOTE:
Before reading the rest of this article, make sure you have read Getting Started With Accreditation
Also, before you start developing your solution, please read General Information for all vendors.

This article offers POS vendors who are interested in accrediting a POS solution the following insight:

- how a POS solution fits in with other EFD components?
- how to navigate the rest of the technical instructions in order to initialize development?

Integration with other EFD components

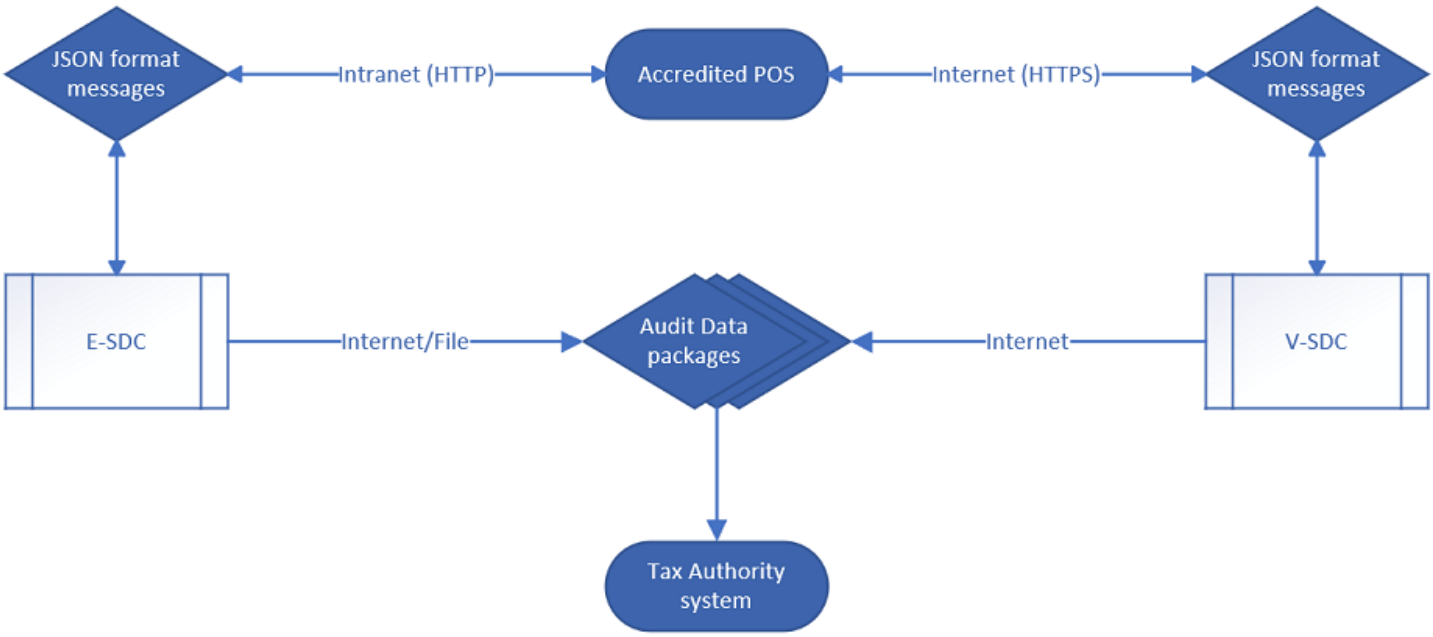
POS is one of three components of any EFD setup.

NOTE:
For an overview of all EFD components and how they communicate with each other, see Electronic Fiscal Device.

POS can work with just one SDC service or with both V-SDC and E-SDC (in which case they are used alternately, depending on the internet availability).

For more details about the differences between the two SDC options, as well as fiscalization recommendations for different business-type scenarios, see:

- Choosing an Appropriate Model
- Recommendation Examples



What can be accredited as a POS solution?

There are many options when it comes to the type of product that can be accredited as a POS solution - the main rule is that it fulfills all technical requirements contained in these instructions. The options include, but are not limited to:

- Standard cash registers
- ERP systems
- Middlewares that serve as a link between an invoicing system and an SDC service
- Mobile applications
- Web applications, etc.

What is the typical process flow of POS operations?

All accredited EFD solutions must follow the basic steps in the process of creating fiscal invoices (although some might have additional, manufacturer-specific, steps).

For a list of these basic steps, see Typical Process Flow.

Connecting with V-SDC service

Issuing fiscal invoices via a V-SDC service requires an internet connection. For more details about the process, see Connected Scenario.

Advantages of V-SDC service

- No specialized hardware
- POS can be implemented as a mobile app
- Compliance of the existing ERP system can be done quickly
-

Cost of taxpayer fiscalization is reduced

Automatic audit

Disadvantages of V-SDC service

- Requires internet connection in order to create fiscal invoices

You also use the V-SDC service to issue fiscal invoices with an online POS solution. For more information, see Online POS and V-SDC Integration.

Communication between a POS solution and V-SDC is established using this POS to SDC protocol.

Connecting with E-SDC service

Issuing fiscal invoices via an E-SDC service can be performed both with or without an internet connection. For more details about the process, see Semi-Connected Scenario.

Advantages of E-SDC service

- Enables issuing fiscal invoices without internet connection

Disadvantages of E-SDC service (if implemented as a hardware/black box solution)

- Increases the cost of taxpayer fiscalization
- If implemented as a hardware/black box solution:
    - Requires a specialized/dedicated hardware
    - Prone to physical damage
    - May require a specialized/dedicated hardware maintenance

Communication between a POS solution and E-SDC is established using the following POS to SDC protocol.

What are the data formats that POS sends and receives?

The formats of all data exchanged between a POS and an SDC service (V-SDC or E-SDC) is described in section Data Formats.

Useful test cases

Please refer to section Test Cases for both standard and special test cases.

All sections of Technical Instructions for POS vendors

Technical instructions specific for POS vendors/developers consist of the following sections:

1. Quick Start

    1. **Register as a vendor** for the Sandbox environment
2. Choosing an Appropriate Model
   The following explanation should help you decide which fiscalization model is the most appropriate for your clients.

3. Typical Process Flow
   This section describes a typical process flow for successful fiscalization scenarios via V-SDC and E-SDC.

4. Connected Scenarios
   In this scenario POS connects to V-SDC and performs instant fiscalization of invoice using web service.

5. Semi Connected Scenarios
   Some jurisdictions may require taxpayers to connect and submit data from their E-SDC on predefined periods of time using any type of audit.

6. Data Formats
   This section describes the main data formats used during fiscalization.

7. Protocols
   Each POS or Invoicing System must communicate with either V-SDC or E-SDC to fiscalize invoices.

8. Online POS and V SDC Integration
   Since Taxpayers are encouraged to use online POS capabilities, TaxCore supports scenarios for browser-based client applications. Accredited online POS creates **Invoice Requests**, and submits them via the HTTPS protocol directly to V-SDC API, using the **digital certificate** issued to Taxpayer. This process completes invoice fiscalization with a signed invoice returned to online POS.

9. How Tos
   This section contains articles, how-tos and cheat sheats that address particular aspects of POS development and operation in context of TaxCore solution

10. Test Cases
    Regardless of the type of invoicing system you are building, the same test cases apply:

Related articles

-

EFD Vendors General Information

Quick Start

Quick step-by-step guide for POS accreditation

1. **Register as a vendor** for the Sandbox environment
2. **Receive a Developer Certificate** and use it to access the Developer Portal
3. **Use the Developer Portal** to request additional certificates for development and testing purposes if required
4. **Consult** all the sections in these **technical instructions** to see understand all the requirements and how they should be implemented
5. Use the applications and services on the Developer Portal to **develop, test and accredit your POS** application
    - Dev-ESDC for testing POS operation with E-SDC service
    - VSDC Request Submitter for testing POS operation with V-SDC service
6. **Compile user documentation** for your POS
7. Use the My Accreditations section on Developer Portal to **apply for accreditation** of your POS.

Choosing an Appropriate Model

The following explanation should help you decide which fiscalization model is the most appropriate for your clients.

	V-SDC	E-SDC
Protocol	HTTPS	HTTP
To establish communication with POS	Certificate required	Certificate Not required

API methods	SignInvoice, GetTaxAuthorityParams	VerifyPIN, SignInvoice, AttentionGetStatus, GetSignedInvoice, GetTaxAuthorityParams
Authentication method	*PAC/PIN	**PIN code
SignInvoice method (Invoice Request)	Same as E-SDC + PAC	Same as V-SDC
Find out what is the latest signed invoice in case of communication failure	YES	YES
Internet connection required to work	YES	NO

*PAC code is sent to the two-factor authentication method (PAC Code and POS PFX Certificate). It is also an option for POS to authenticate to VSDC using the POS secure element (smart card), in which case the PIN is sent by POS during authentication (handshake).

**ESDC requires a PIN code to be sent through the VerifyPIN method in order for ESDC to be activated

Differences Between V-SDC and E-SDC

V-SDC	E-SDC
V-SDC uses the HTTPS protocol.	E-SDC uses the HTTP protocol.
An authentication certificate is required to establish communication between a POS and V-SDC.	No Authentication certificates are required to establish communication between a POS and E-SDC.
API methods used for V-SDC are: *SignInvoice* and *GetTaxAuthorityParams*.	API methods used for E-SDC are: *VerifyPIN*, *SignInvoice*, *Attention*, *GetStatus*, *GetSignedInvoice* and *GetTaxAuthorityParams*.
V-SDC requires sending a PAC code as a property in the *InvoiceFiscalizationRequest*, as a two-factor authentication method (PAC Code and POS PFX Certificate).	E-SDC does not require sending a PAC code as a property in the *InvoiceFiscalizationRequest*. Instead, E-SDC requires sending a PIN code through the VerifyPIN method in order for ESDC to be activated.
SignInvoice methods for both V-SDC and E-SDC are identical, except when a PAC code is required if the authentication between POS and V-SDC is established through the POS PFX certificate.	HASH property of *InvoiceFiscalizationRequest* or *GetSignedInvoice* is designed only for E-SDC, and not for V-SDC.

V-SDC is a cloud-based service, therefore a POS requires an available internet connection in order to sign invoices.	E-SDC is a semi-online-based solution; therefore, it does not require an available internet connection to sign invoices. E-SDC can be connected to the local network (LAN cable or Wi-Fi) or directly to the POS via a LAN cable.

Recommendation Examples

This section gives examples of the most common implementation scenarios, for different end-users.

Small Shops

In small shops, it is possible to use all kinds of devices from tablets to POS applications. The choice of the device generally depends on the number of items, which are on the sale list (PLU) or on the environmental conditions.

Agencies, Individuals and Travelling Salesmen

Agencies are not issuing a large number of receipts and issuing is not time-critical; mobile POS application connection to V-SDC will probably cover their needs.

Supermarkets

Supermarkets are using high volume POS systems with additional different peripherals. Due to the very nature of the supermarket or shop sale process (on the counter) it is required to have offline capabilities (E-SDC) to overcome interruptions of the internet connection.

Restaurants and Hotels

Restaurants have very specific applications. Proforma as transaction type is supported and recorded and can be verified. Offline capabilities (E-SDC) are also important because receipts have to be printed on demand.

Taxi Drivers

Taxi drivers use taximeters that measure parameters from a ride. Old taximeters do this by mechanical methods; there are many challenges in their connection with modern EFD systems. If local regulations allow it, modern taxi terminals or mobile taxi applications are increasingly used worldwide (with GPS locator), which facilitates the connection with the EFD system. Taxi drivers prefer small and robust system. Depending on the chosen taximeter they can use E-SDC or V-SDC.

Remote Sites

POS on the remote or underground sites with an unstable internet connection will have to work with E-SDC devices to provide customers with fiscal invoices. Local audits would be conducted by tax inspectors or taxpayers on a regular basis.

Malls, Shopping Areas

Areas with a high concentration of small shops can contain wireless access point with a dedicated V-SDC for that area.

Enterprises

ERPs and Invoicing systems could utilize both V-SDC and on-site E-SDC devices to fiscalize invoices. It is safe to assume this kind of establishments have permanent (or even redundant) internet connection. Fiscalization using V-SDC service would probably be the most appropriate solution.

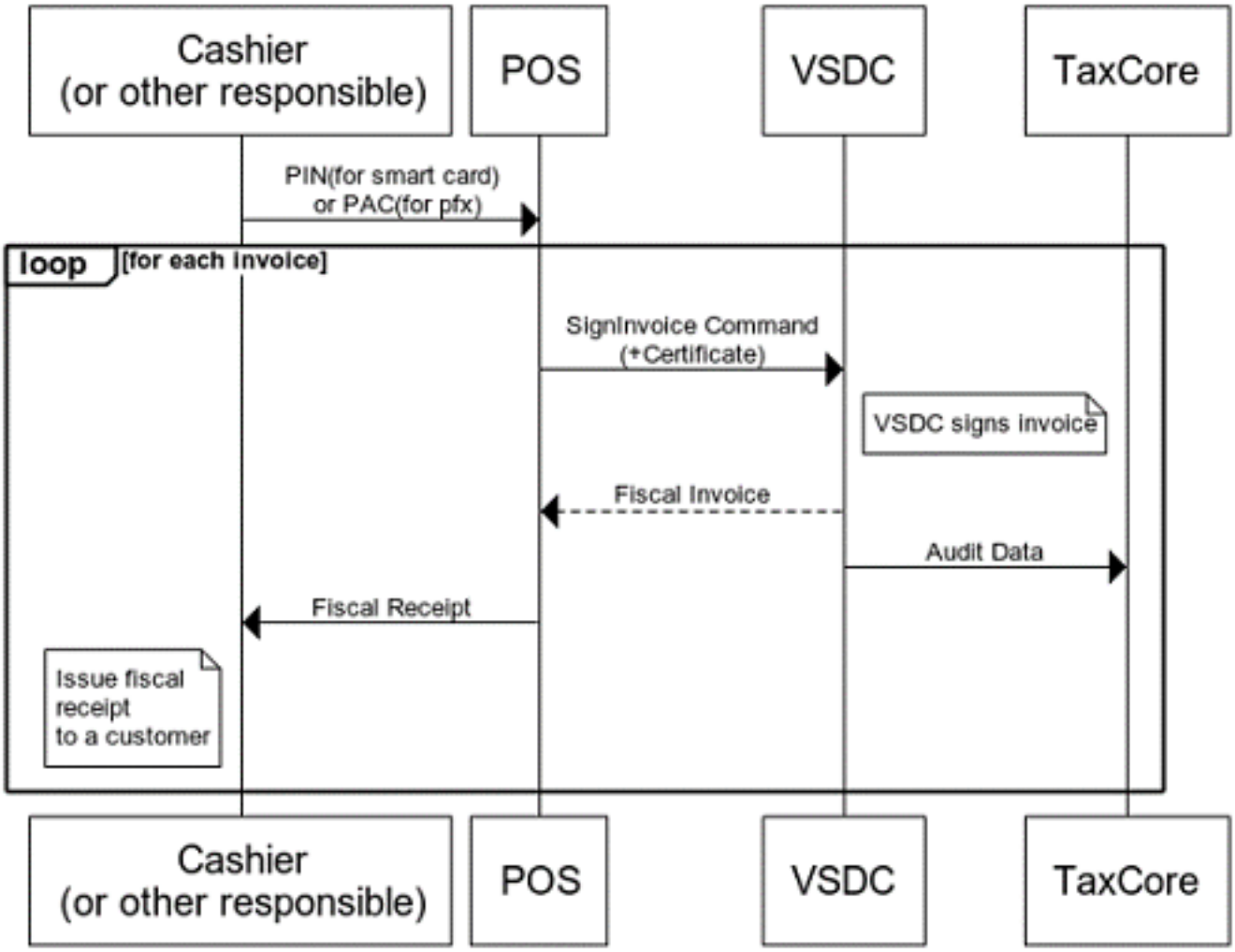
Web Shops

Web Shop applications could connect to V-SDC service using the digital certificate issued to the Taxpayer to fiscalize an invoice at the moment of payment.

Typical Process Flow

This section describes a typical process flow for successful fiscalization scenarios via V-SDC and E-SDC.

V-SDC process flow



Provide a valid PIN/PAC

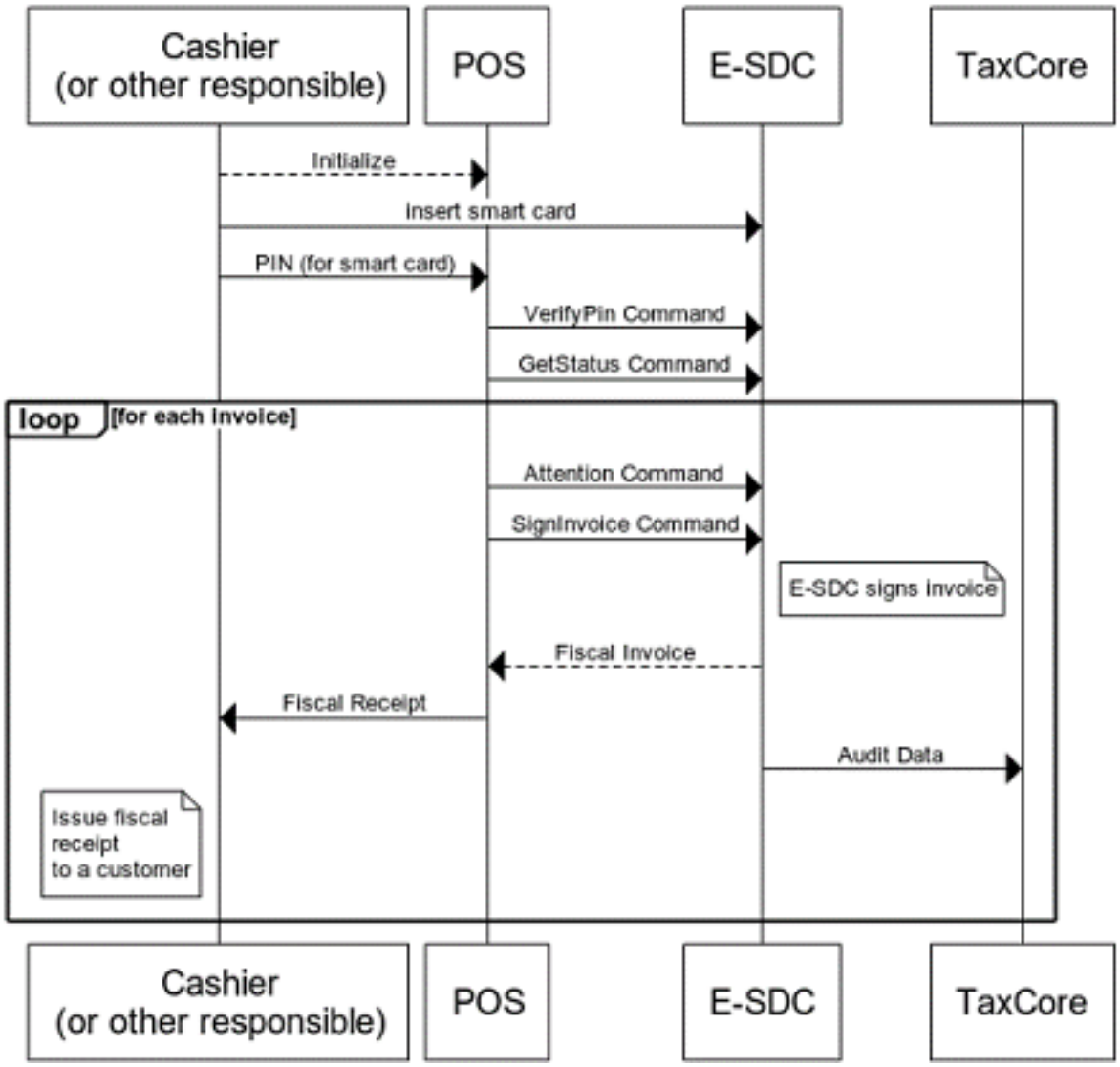
The process begins with a cashier creating an invoice request on an accredited POS. An invoice request transmits the transaction data (POS date and time, list of items, transaction type, payment type, etc.) to an available V-SDC.

When sending the request to a V-SDC, the cashier needs to provide a valid PIN (form smart cards) or PAC (for .pfx certificates) to confirm the authenticity of the certificate.

Fiscalization via V-SDC

In short, V-SDC receives the transaction data from POS, fiscalizes it, returns the fiscalized invoice back to POS and submits the audit data to TaxCore.

E-SDC process flow



Provide a valid PIN

The process begins with a cashier creating an invoice request on an accredited POS. An invoice request transmits the transaction data (POS date and time, list of items, transaction type, payment type, etc.) to the connected E-SDC.

When sending the request to an E-SDC, the cashier needs to provide a valid PIN (form smart cards) to confirm the authenticity of the certificate.

Fisacalization via E-SDC

Just like V-SDC, an E-SDC service also receives the transaction data from POS, fiscalizes it, returns the fiscalized invoice back to POS and submits the audit data to TaxCore - however, it offers an additional option to do all this with an internet connection.

The process involves the following steps:

1. Cashier enters the required information via an accredited POS
2. POS generates a request data and sends it as a request to the E-SDC using JSON via HTTP protocol;
3. E-SDC verifies the format of the invoice;
4. E-SDC calculates taxes based on the current tax rates;
5. E-SDC sends the invoice data to the Secure Element for fiscalization providing the current date and time and PIN code/password if required;

6. Secure element signs the invoice and returns the data to the E-SDC;
7. E-SDC produces a journal file – a textual representation of an invoice;
8. E-SDC generates a verification URL;
9. [optionally] E-SDC creates a QR Code – a graphical representation of a verification URL;
10. E-SDC creates an invoice with all mandatory elements (receipt data, previously generated signature, verification URL and journal), generates a one-time key and encrypts the invoice using a symmetric algorithm. The E-SDC encrypts a one-time symmetric key using the tax authority's system public key and adds it to the package so the TaxCore system shall decrypt the symmetric key and access the package content once it arrives to the TaxCore system.
11. If the internet connection is available, E-SDC sends the Audit Data to TaxCore
12. E-SDC returns the Invoice Fiscalization Result (Journal + QRCode/Verification URL) back to the POS
13. The POS prints the fiscal invoice and the Cashier gives it to the customer (or sends an electronic invoice to the customer)

Transaction Report on Invoicing System

Depending on the tax jurisdiction requirements, an Invoicing System (POS) must be able to generate and print a transaction report of all Normal and Advance invoices. The invoicing system must allow the operator to select the period for displaying the report.

NOTE:
Always check whether the tax jurisdiction where you wish to accredit your product has a requirement for generating transaction reports.

The report should contain the following information:

- Date and time of the report period start and end
- The number of issued invoices per invoice type
    - Normal Sale
    - Normal Refund
    - Advance Sale (if issued by the invoicing system)
    - Advance Refund (if issued by the invoicing system)
- The list of sold items - quantity, unit price, and total price
- Total amount on the Sale (Normal and Advance) invoices
- Total amount on the Refund (Normal and Advance) invoices
- Total amounts per payment type
    - cash
    - card
    - check
    - wire transfer
    - mobile money
    - voucher
    - other

Client Authentication

Introduction

Communication between POS and SDC must be secure and protected from any unauthorized use. Depending on type of Secure Element, specifics of the POS and SDC in use there are different approaches to secure communication between those two parties.

Each POS should support all three scenarios:

1. POS to V-SDC using digital certificate distributed in file format
2. POS to V-SDC using using digital certificate distributed on smart card
3. POS to E-SDC on the local network

POS Connects to V-SDC

POS and V-SDC API communication requires mutual authentication of client (POS) and server (V-SDC). Mutual authentication between parties is conducted using client and server digital certificates.

POS is required to use Client Certificate Authentication with each request targeting V-SDC API.

Secure Elements (and associated digital certificates) may be distributed in two formats

1. Files (PKCS12 *.pfx or *.p12)
2. Smart Cards

Authentication against V-SDC using digital certificate distributed in file format

In this scenario POS is using digital certificate installed in the local certificate store (key chain) for client authentication.

To improve security caller is required to submit PAC associated with used secure element as part of the HTTP request. More info is available in SDC protocol section.

Authentication against V-SDC using digital certificate distributed on smart card

If POS is using secure element from the locally attached smart card for client side authentication PAC is **not** required but user will be automatically prompted to enter PIN code for the selected smart card by Operating system to finalize client-side authentication against V-SDC server.

POS Connects to E-SDC

Communication between POS and E-SDC available on local machine or on the LAN must be safeguarded using network-level security.

E-SDC will require PIN code of the Secure element to enable fiscalization-related operations on the Secure Element Applet used by E-SDC.

Connected Scenarios

Connected Scenario is preferred way of communication between POS and SDC.

In this scenario POS connects to V-SDC and performs instant fiscalization of invoice using web service.

1. Accessing V SDC API
   Once valid Test certificate(s) are obtained, you can access the V-SDC.Api description on the following URL: [[*TaxCore.PublicConfiguration.VSDCApiUrl*]]

2. Example
   This example illustrates how to create and initialize an instance of HttpClient class in C# language. Use it to authenticate against V-SDC and submit an invoice.

Accessing V-SDC API

Once valid Test certificate(s) are obtained, you can access the V-SDC.Api description on the following URL: [[*TaxCore.PublicConfiguration.VSDCApiUrl*]]

NOTE:
The following process applies only to the .pfx digital certificates, and not to smart card certificates.

To extract VSDC.Api URL from a certificate, follow these steps:

1. Open the .pfx certificate installed on your computer
2. In the **General** tab, your certificate should have one of these OIDs **1.3.6.1.4.1.49952.X.Y.3.7**
3. X and Y parameters identify the environment and their values change according to each environment.

4.

Click on the **Details** tab and find the line with an OID in this format - **1.3.6.1.4.1.49952.X.Y.7**

5. Again, X and Y parameters identify the environment, and the values will vary according to different environments, but they will be the same as on the **General** tab
6. Number **7** at the end identifies the V-SDC.Api URL
7. Read the value of this OID containing the URL of V-SDC.Api

Example

This example illustrates how to create and initialize an instance of HttpClient class in C# language. Use it to authenticate against V-SDC and submit an invoice.

When executing this code, you will be asked to provide the PIN for the smart card certificate, which you selected in GetClientCertificate method. In case you selected the installed PFX certificate, which you obtained from the tax authority, you will need to provide PAC value as an HTTP header to each request.

```
using System.Net;
using System.Net.Http;
using System.Security.Cryptography.X509Certificates;
using System.Text;

static void Main(string[] args)
{

    string invoiceRequest = @"{
  "dateAndTimeOfIssue": "2020-12-08T08:55:23.286Z",
  "cashier": "123456789",
  "buyerId": "RS34564565",
  "buyerCostCenterId": "567546",
  "invoiceType": "Normal",
  "transactionType": "Sale",
  "payment": [
    {
      "amount": 70.00,
      "paymentType": "Cash"
    }
  ],
  "invoiceNumber": "POS2017/998",
  "options": {
    "omitQRCodeGen" : "1" ,
    "omitTextualRepresentation" : "0"
  },
  "items": [
    {
      "name": "Sport-100 Helmet, Blue",
      "quantity": 2,
      "unitPrice": 34.23,
      "labels": [
        "A"
      ],
```

```
      "totalAmount": 68.46
    }
  ]
}";

    var httpContent = new StringContent(invoiceRequest,Encoding.UTF8,"application/json");
    HttpClient client;
    WebRequestHandler handler;

    GetClientAndHandler(out handler, out client);

    var response = client.PostAsync($"/api/v3/invoices", httpContent).Result;

    if (response.StatusCode == HttpStatusCode.OK)
    {
        var jsonString = response.Content.ReadAsStringAsync();
        jsonString.Wait();
        var invoiceResponse = jsonString.Result;
        Console.Write(invoiceResponse);
    }
}

static void GetClientAndHandler(out WebRequestHandler handler, out HttpClient client)
{
    handler = CreateWebRequestHandler();
    client = new HttpClient(handler);

    client.BaseAddress = new Uri("https://vsdc.sandbox.taxcore.online/");
    client.DefaultRequestHeaders.Accept.Clear();
    client.DefaultRequestHeaders.Add("PAC", "123456");
}

static WebRequestHandler CreateWebRequestHandler()
{
    var handler = new WebRequestHandler();
    var cert = GetClientCertificate();

    handler.ClientCertificateOptions = ClientCertificateOption.Manual;
    handler.ClientCertificates.Add(cert);
    return handler;
}

static X509Certificate2 GetClientCertificate()
{
    string certName = "9AH3 My Store inc.";
    var store = new X509Store(StoreName.My, StoreLocation.CurrentUser);

    store.Open(OpenFlags.OpenExistingOnly | OpenFlags.ReadOnly);
    return store.Certificates.Find(X509FindType.FindBySubjectName, certName, true)[0];
}
```

Semi-Connected Scenarios

Semi-Connected Scenario enables sale point to stay operational and issue fiscal invoices without permanent or reliable interent connection for prolonged periods of timee.

Some jurisdictions may require taxpayers to connect and submit data from their E-SDC on predefined periods of time using any type of audit.

If Internet connection and TaxCore.API are available to E-SDC at lease for couple of minutes per day it is usually enough time to submit all pending audit packages and comply with maximum audit period requirements.

For more information please refer to Semi-Connected Scenario.

Data Formats

This section describes the main data formats used during fiscalization.

1. Date and Time
   The date and time sent by POS to E-SDC or V-SDC is local time.

2. How SDC Calculates Taxes
   Taxes are calculated by an SDC after a POS has sent a valid request. The tax amount for particular items on an invoice is defined by the tax labels associated with an item.

3. Buyer TIN on Export Invoices
   A Tax Identification Number (TIN) is an identifier used in many countries to uniquely identify each taxpayer.

Date and Time

The date and time sent by POS to E-SDC or V-SDC is local time.

The date and time generated by E-SDC or V-SDC and printed on a receipt is local time.

JSON-based protocols use date and time according to ISO 8601 where applicable (for example: 2017-05-17T10:46:51.910Z).

NOTE:
Date and Time display format on all TaxCore portals is: DateTimeDisplayFormat - "dd/MM/yyyy HH:mm:ss" (e.g. 15/10/2018 14:28:24).

NOTE:
Be mindful of the minimum (1900-01-01T00:00:01Z) and maximum (9999-12-31T23:59:59Z) values.
Also, individual tax jurisdictions might have specific requirements regarding the maximum difference between

the time of invoice creation (SDC time) and the moment when an audit package arrives to VMS. Make sure you are familiar with those requirements.

How SDC Calculates Taxes

Introduction

Taxes are calculated by an SDC after a POS has sent a valid request. The tax amount for particular items on an invoice is defined by the tax labels associated with an item.

How does E-SDC Calculate Taxes?

The process of a tax calculation depends on:

- Invoice and Transaction Type
- the tax rates for each label associated with an item on an invoice
- the Type value of the tax category to which the label belongs

A POS sends an invoice fiscalization request with the line items. Items are sent with the total amounts (taxes included) and zero or more tax labels associated with them, which participated in the total price calculation.

For more information, see article Calculate Taxes in the section of documentation for E-SDC vendors.

Rounding

An SDC (both E-SDC and V-SDC) rounds all amounts to 4 decimal places, using the half-round up method.

The rules for this are simple:

- Decide which is the last digit to keep (in this case, the fourth decimal place)
- Leave it the same if the next digit is less than 5
- But increase it by 1 if the next digit is 5 or more

Examples:

3.44445555666 → 3.4445

3.4440012345 → 3.4440

3.44466012345 → 3.4447

3.444116012345 → 3.4441

Any amount shall be rounded to 2 (two) decimal places, using the half-round up method, **only** on the textual representation of an invoice.

NOTE:
In SDC's internal memory, the amount will always be stored with 4 decimal places (if the amount has 4 or more decimals), regardless of invoice's textual representation with 2 decimal points.

Buyer TIN on Export Invoices

Introduction

A Tax Identification Number (TIN) is an identifier used in many countries to uniquely identify each taxpayer.

Every time when a taxpayer issues a **B2B fiscal invoice**, it **must contain a valid Buyer TIN** which uniquely identifies the buyer of sold goods/services.

However, when issuing a B2B fiscal invoice for exported goods/services (i.e. when goods/services are sold to a buyer from a foreign country), the cashier **must enter a Buyer TIN which starts with an official country code prefix**.

NOTE:
This rule might not apply in every tax jurisdictions. Make sure you become familiarized with each jurisdiction's legal requirements.

How to format TIN

Country code prefix is an **ISO 3166-1 alpha-2 (2 letters) country code**. It identifies the buyer's country of origin.

The complete and up to date list is available on https://www.iso.org/iso-3166-country-codes.html

The rest of the TIN is composed of numeric digits in most countries, but in some countries, it may contain letters.

When issuing a B2B invoice that is not for exporting goods/services, the country code prefix can be included in the Buyer TIN, but it is not mandatory.

Examples - Valid formatting

- RS123456789
- RS-123456789
- RS-12345-6789
- RS 123456789

Example of the printed B2B invoice

```
============ FISCAL INVOICE ============
TIN:                            980361508
Company:          Compete Enterprises Inc
Store:            Compete Enterprises Inc
Address:                    50 Via Del Sol
District:                               Ba
Cashier TIN:
Buyer TIN:                   RS1234567890
Buyers Cost Center:                1256KSR
POS Time:             14/07/2021 08:53:56
--------------NORMAL SALE---------------
Items
========================================
Name     Price         Qty.           Total
Komp 12e (A)
       1200.00           1          1200.00
----------------------------------------
Total Purchase:                    1200.00
Payment Method:                       Cash
========================================
Label         Name      Rate            Tax
A              VAT     9.00%          99.08
----------------------------------------
Total Tax:                           99.08
========================================
SDC Time:             14/07/2021 08:53:58
SDC Invoice No:       HQBK79BT-F6MYL8UM-2
Invoice Counter:                    2/2NS
========================================

[QR Code]

======== END OF FISCAL INVOICE =========
```

Buyer TIN on Export Invoices

Protocols

Introduction

Each POS or Invoicing System must communicate with either V-SDC or E-SDC to fiscalize invoices.

The communication protocol is standardized and public. API Documentation is available as part of this documentation.

Each E-SDC is required to implement and expose documented API to all POS and Invoicing systems. Correctness

of API implementation is, among other criteria, one of the requirements for E-SDC accreditation.

Each POS must connect to SDC to issue a fiscal invoice. Depending on the business and technical requirements POS can connect using one of the following approaches:

1. **POS connects to Taxcore.Api** to get environment configuration and tax rates [Optional, but highly recommended]
2. **POS connects to Development E-SDC**. This is used for development, testing and accreditation purposes only. Development E-SDCs are provided as web service to all POS vendors to enable development without the need to possess real E-SDC applications or devices.
3. **POS connects to V-SDC** using a digital certificate delivered in **PKCS12** format and PAC to authenticate to V-SDC.Api
4. **POS Connects to V-SDC** using digital certificate delivered on **smart card** and PIN Are used to authenticate to V-SDC.Api
5. **POS Connects to E-SDC**. Communication between POS and E-SDC is secured on a network level. Fiscalization of the invoice is performed by E-SDC and Secure Element inserted into E-SDC

Online POS and V-SDC Integration

Since Taxpayers are encouraged to use online POS capabilities, TaxCore supports scenarios for browser-based client applications. Accredited online POS creates **Invoice Requests**, and submits them via the HTTPS protocol directly to V-SDC API, using the **digital certificate** issued to Taxpayer. This process completes invoice fiscalization with a signed invoice returned to online POS.

Online POS **submits requests to V-SDC API directly**, in order to create an HTTPS request with the client certificate. For client-side, JavaScript-based applications, the most common solution for achieving the best user experience is using AJAX. For security reasons, browsers restrict cross-origin HTTPS requests initiated from within the scripts. TaxCore offers a solution for online POS, which will overcome issues with CORS and digital certificates sent from the client.

We recommend the usage of a secure network communication with SSL protocol for online POS, although the V-SDC API will accept requests from an unsecured online POS.

1. Quick Start
   Our pre-built TaxCore element is used to collect data from the online POS page. Online POS prepares Invoice Request data for this page thus TaxCore element can collect and send this data to V-SDC. Quick integration can be achieved as follows:

2. Detailed Specs
   There are two important components that enable the integration of online POS with V-SDC.

3. [Supported Browsers]
   Online POS and V-SDC are tested with the following browsers:

4. [Example of Integration Using a Simple HTML Page]
   The following code is an example of simple integration. The HTML serves as an integration point for V-SDC. Instead of this HTML, you should use page of your online POS application.

Quick Start

Our pre-built TaxCore element is used to collect data from the online POS page. Online POS prepares Invoice Request data for this page thus TaxCore element can collect and send this data to V-SDC. Quick integration can be achieved as follows:

1. Online POS needs to provide a page for collecting and sending **Invoices**. This is the **integration point** on the POS side.
2. Add the following script tag to integration point, with src attribute reffering to taxcore.min.js file*:

```
<script src="[vsdcurl]/onlinepos/v1/taxcore.min.js"></script>
```

3. Add TaxCore Sign Element to your integration point with required id and data-* attributes.

```
<!--TaxCore html element-->
<button id="taxcore_sign_element"
        data-taxcore-vsdc-url="[vsdcurl]"
        data-taxcore-input-id="[inputDataContainerId]"
        data-taxcore-output-id="[outputDataContainerId]">Sign Invoice</button>
```

- Id attribute is required and it must be equal to string "taxcore_sign_element".
- vsdcurl – provide V-SDC API url
- inputDataContainerId – provide id of HTML tag element which contains invoice request json, usually input tag
- outputDataContainerId – provide id of HTML tag element which will be populated by response from V-SDC API

Detailed Specs

There are two important components that enable the integration of online POS with V-SDC.

1. taxcore.min.js file
2. taxcore sign element

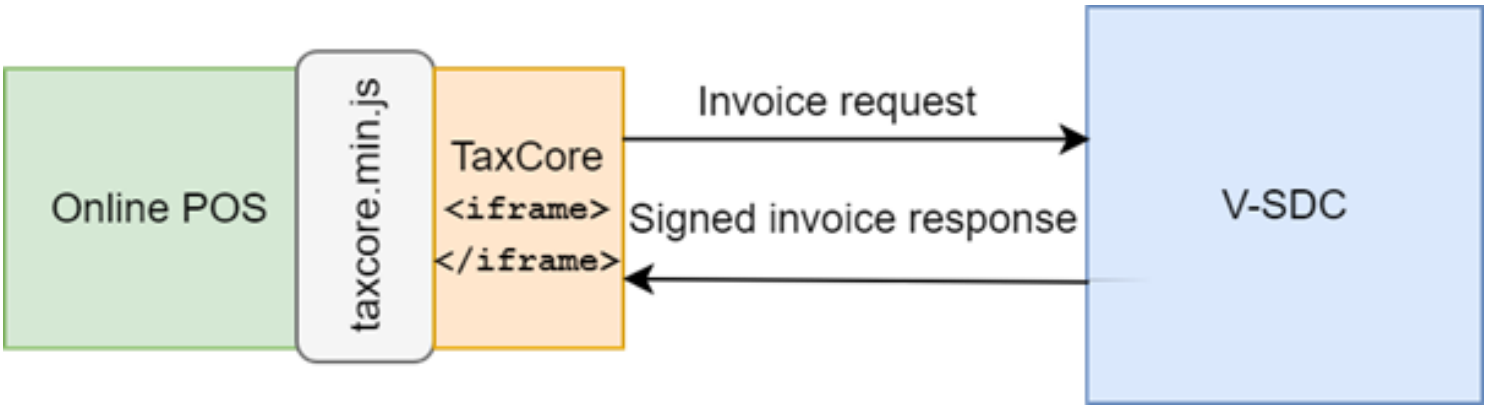
NOTE:
Both components must exist at the integration point for successful integration. JavaScript will first try to find taxcore sign element with id="taxcore_sign_element". If it can't be found, the exception will be thrown with an appropriate message (e.g. "*Could not find taxCoreElement. Check if you have valid HTML tag with expected id taxcore_sign_element*").

TaxCore JavaScript file - taxcore.min.js

The first step in enabling your online POS application page to work with V-SDC is to include script reference to **taxcore.min.js** file, provided by TaxCore. This JavaScript will prepare your page to handle HTTP invoice requests to V-SDC API.

Online POS must directly submit data to V-SDC API to transport the digital certificate over the network from the client side (in this case from Taxpayer's web browser). In order to achieve this requirement, taxcore.min.js script will create **iframe element** within online POS, effectively embedding the V-SDC HTML page into the current POS page. POS (Parent page) sends messages to iframe (Child page), and then iframe performs post to V-SDC. By doing so, we can overcome CORS issues with digital certificates, since client certificate in CORS preflight OPTIONS requests is omitted (https://www.w3.org/TR/cors/#cross-origin-request-with-preflight-0).

The other features, provided by this js file are related to sending and receiving data to and from V-SDC. Your online POS needs to prepare data for input and to handle returned data. All other job is delegated to taxcore.min.js.



Detailed Specs - Image of taxcore.min.js as the communication interface between POS and V-SDC

After successfully creating the iframe element, we can send messages between the Online POS parent page and the TaxCore child page. The responsibility of initiating the message is part of the second component TaxCore Sign Element described in the next section.

TaxCore Sign Element

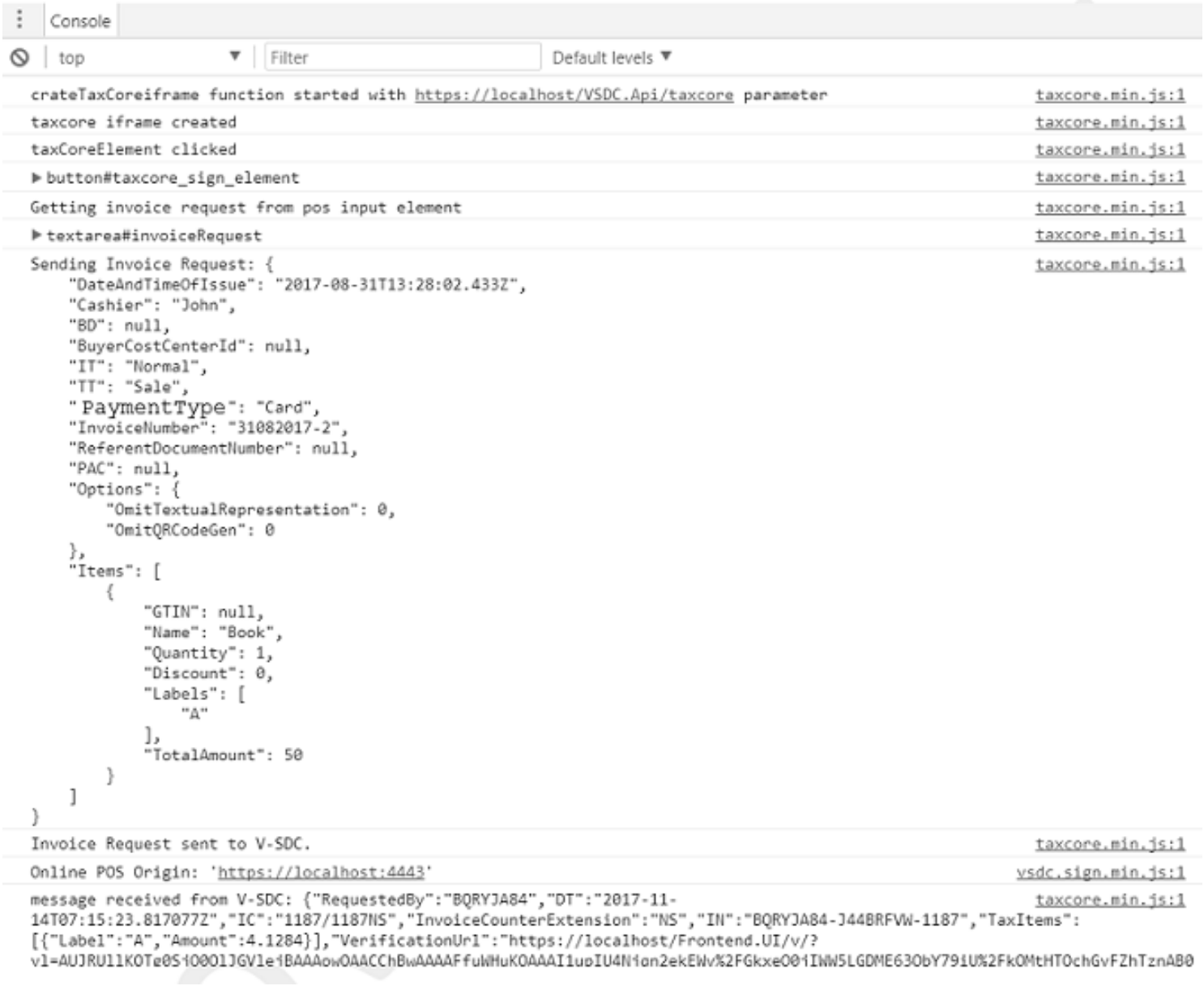
TaxCore Sign element is nothing more than an HTML tag with predefined id, that you'll place at your online POS page. Its task is to collect invoice data and forward them to V-SDC.

> **Note:** The id of TaxCore Sign Element **must** be equal to the string "**taxcore_sign_element**", so that code in taxcore.min.js could be able to find it. Otherwise, the data collection would not be possible.

The best practice for this element is to be clickable HTML element, e.g. a button, but it's not restricted to it. TaxCore Sign Element is also used to configure its behavior using HTML data attributes, as follows:

```
<!--TaxCore html element-->
<button id="taxcore_sign_element"
    data-taxcore-vsdc-url=" https://vsdc.sandbox.taxcore.online/"
    data-taxcore-input-id="invoiceRequest"
    data-taxcore-output-id="invoiceResponse"
    data-taxcore-invoice-request=""
    data-taxcore-debug="true"
    data-taxcore-signed-invoice-response="">
Sign Invoice
</button>
```

Data attribute	Restriction	Description
data-taxcore-vsdc-url	Required	V-SDC URL
data-taxcore-input-id	required if data-taxcore-invoice-request attribute is not used	Id of the HTML element on POS page, which contains prepared invoice request as required JSON scheme.
data-taxcore-output-id	It is optional since V-SDC will always store signed invoice response at data-taxcore-signed-invoice-response attribute	Id of the HTML element on POS page used to store signed invoice response JSON from V-SDC.
data-taxcore-invoice-request	required if data-taxcore-input-id attribute is not used	If you do not want to use separate HTML element for invoice request JSON, you can store it at this data attribute, and it will be collected when TaxCore Sign Element is clicked.
data-taxcore-signed-invoice-response		This is the default storage location for the signed invoice response JSON from V-SDC. It will be always populated.
data-taxcore-debug	Optional	Used to log relevant information during invoice fiscalization. The log is written to the browser console.



Detailed Specs - Image of the sign-in element

Logs written to browser console when data-taxcore-debug is set to true

Subscribe to TaxCore messages

If you need to perform some tasks, after the message from V-SDC is received, you can listen on the following event and do the necessary handling. For example, the following code will be executed after the signed invoice response JSON from V-SDC is written to the appropriate storage location.

```
 // Listen to message from taxcore
window.onmessage = function (e) {
    console.log(e.data);
}
```

Message received from V-SDC is well-formatted JSON message in the following format:

```
{
 "status_code": 400,
 "response": {
   "Message": "The request is invalid.",
```

```
   "ModelState": {
     "invoice.PaymentType": [
       "2805"
     ],
     "invoice.Items[0].Labels[0]": [
       "2310"
     ]
   }
 }
}
```

If the status code is 200, the request is successfully signed with VSDC and the response is a signed invoice in JSON format.

If the status code is not 200, the response filed will contain an error message and error description received from VSDC. You can use the following code to determine the reason.

```
window.onmessage = function (e) {
            var response = JSON.parse(e.data);
            if (response.status_code != '200') {
                var error_reason = response.response;
            }
       }
```

Supported Browsers

Online POS and V-SDC are tested with the following browsers:

- Google Chrome 61.0.3163.100+
- Microsoft Internet Explorer 11 (not supported by Microsoft, use it on your own responsibility)
- Microsoft Edge 40.15063.674.0+
- Firefox 56.0.2+
- Opera 48+

Example of Integration Using a Simple HTML Page

The following code is an example of simple integration. The HTML serves as an integration point for V-SDC. Instead of this HTML, you should use page of your online POS application.

```
<!DOCTYPE html>
<html>
<head>
    <title>Online POS</title>
    <meta charset="utf-8">
</head>
<body>
    <h1>Online POS</h1>
    <label for="invoiceRequest">Invoice request json</label>
    <textarea id="invoiceRequest" cols="100" rows="30" style="display:block"></textarea>
    <label for="taxcore_sign_element">Send Invoice Request:</label>
```

```
    <!--TaxCore HTML element-->
    <button id="taxcore_sign_element"
                data-taxcore-vsdc-url=" https://vsdc.sandbox.taxcore.online/"
                data-taxcore-input-id="invoiceRequest"
                data-taxcore-output-id="results"
                data-taxcore-invoice-request=""
                data-taxcore-debug="true"
                data-taxcore-signed-invoice-response="">Sign Invoice</button>
    <label for="results">Received Signed Invoice:</label>
    <textarea readonly id="results" cols="100" rows="30"></textarea>
    <!-- TAXCORE.JS -->
    <script src=" https://vsdc.sandbox.taxcore.online/onlinepos/v1/taxcore.min.js"></script>
    <!-- Custom script at Online POS -->
    <script>
        document.getElementById("invoiceRequest").innerHTML =
        JSON.stringify(CreateExampleInvoiceRequest(), undefined, 4);

document.getElementById("taxcore_sign_element").dataset.taxcoreInvoiceRequest =
        JSON.stringify(CreateExampleInvoiceRequest());

// Listen to messages from TaxCore
        window.onmessage = function (e) {
            console.log(e.data);
        }

        function CreateExampleInvoiceRequest() {
            var invoiceRequest = {
  "dateAndTimeOfIssue": "2020-12-08T08:55:23.286Z",
  "cashier": "123456789",
  "buyerId": "RS34564565",
  "buyerCostCenterId": "567546",
  "invoiceType": "Normal",
  "transactionType": "Sale",
  "payment": [
    {
      "amount": 70.00,
      "paymentType": "Cash"
    }
  ],
  "invoiceNumber": "POS2017/998",
  "options": {
    "omitQRCodeGen" : "1" ,
    "omitTextualRepresentation" : "0"
  },
  "items": [
    {
      "name": "Sport-100 Helmet, Blue",
      "quantity": 2,
      "unitPrice": 34.23,
      "labels": [
        "A"
      ],
      "totalAmount": 68.46
    }
  ]
};

            return invoiceRequest;
        }
    </script>
</body>
</html>
```

How Tos

This section contains articles, how-tos and cheat sheats that address particular aspects of POS development and operation in context of TaxCore solution

1. Issuing Copy Invoice In Case of Power Outage or Printer Spooler Failure
   Issuing an invoice copy to a customer is obligatory, this guarantees that the consumer always knows what they've purchased, and it is in line with the rules of the local tax authority.

2. How to Determine the Tax Amount Before Issuing a Sale Invoice?
   In an Invoice Request, a POS sends the **'TotalAmount'** which represents the gross amount of the products/services sold. This reflects the amount that a customer needs to pay before sending the API call to an SDC.

3. Handling Payments on Automatic Payment Stations
   Very often, businesses employ automatic payment stations to make it easier and faster for the customers to pay whatever they might be buying. This is most frequently the case with gas stations where customers don't necessarily need to physically enter in order to pay (unless they wish to buy food or other goods).

4. How to Print a QR Code Using a Dot Matrix Printer?
   Digitalizing the tax collection system comes with a lot of benefits for both consumers and businesses. One of the best solutions TaxCore came up with is allowing a customer to easily and quickly verify their receipt.

5. Printing Receipts Is Not the Only Option
   TaxCore solution doesn't require taxpayers to necessarily print receipts. It offers one more options better suited for the digital age.

Issuing Copy Invoice In Case of Power Outage or Printer Spooler Failure

Issuing an invoice copy to a customer is obligatory, this guarantees that the consumer always knows what they've purchased, and it is in line with the rules of the local tax authority.

Unfortunately, a business can suffer a sudden power outage, communication between POS and SDC may fail for variety of reasons and printer paper may jam in the middle of operation. In this case, the situation doesn't permit issuing an invoice copy to a customer. The solution is simple – it is enough to have the reprint option, which can be manual or automatic.

Most importantly, the copy needs to be processed by either E-SDC or V-SDC if one is available. Please note that

TaxCore as a system allows that invoices do not require a physical copy in order to be verified.

In case a business has a loyalty program, it can send invoices to customers personal e-mail addresses. The digital copies of invoices can also be sent via different messaging apps, such as WhatsApp, Viber, and WeChat.

In case the problem is not the power outage but a printer spooler failure, there is a solution for this too. A customer can be offered a QR code on the customer-facing display which they can scan using their smart-phones. This will provide them with the URL with an exact digital copy of their receipt.

TaxCore allows many features and solutions to businesses and taxpayers, and these should be taken advantage of whenever different solutions are necessary.

How to Determine the Tax Amount Before Issuing a Sale Invoice?

In an Invoice Request, a POS sends the **'TotalAmount'** which represents the gross amount of the products/services sold. This reflects the amount that a customer needs to pay before sending the API call to an SDC.

In Invoice Response, the POS will receive a breakdown of tax amounts as per labels used in the Invoice Request.

POS can learn in advance what is the valid tax rate, exposed by the Tax Authority by using the Get Environment Parameters method.

If anyone wishes to know in advance what will be the tax portion after fiscalization, the instructions on how an SDC (External or Virtual) calculates taxes can be found here - Calculate Taxes.

For any other reason, a POS can always print a Proforma Sale (PS) invoice to obtain a so-called 'quotation'. This will tell the cashier exactly what the tax portion will be prior to concluding the sale. Just remember that every PS should be referenced in the Normal Sale (NS) invoice which makes the transaction conclusive. For more information on document referencing see Anatomy of a Fiscal Receipt.

Handling Payments on Automatic Payment Stations

Very often, businesses employ automatic payment stations to make it easier and faster for the customers to pay whatever they might be buying. This is most frequently the case with gas stations where customers don't necessarily need to physically enter in order to pay (unless they wish to buy food or other goods).

If a customer doesn't physically enter the shop to pay for the gas, how will they receive a fiscal receipt? In this case, an OPT (Outdoor Payment Terminal) is installed where customers can pay using both cash and debit/credit cards.

This still means that all the transactions done at the station must be sent to VMS, regardless of whether POS and

cashier were directly involved or not (in case of an OPT). It is mandatory by law that each customer receives a verifiable invoice (receipt) of the goods or services they've purchased.

OTPs are usually equipped with a printer, which means that customers can easily receive a printed version of an invoice in case they paid for fuel, for example. In case a business can't equip an OTP with a printer, it should find a method to deliver the invoice to a customer's e-mail.

The platform offers many commodities to businesses, and the goal is to help everything run smoother and easier. Issuing a physical copy of an invoice is no longer mandatory and it can be done by simply sending it digitally, which results in happy customers.

How to Print a QR Code Using a Dot-Matrix Printer?

Digitalizing the tax collection system comes with a lot of benefits for both consumers and businesses. One of the best solutions TaxCore came up with is allowing a customer to easily and quickly verify their receipt.

The best way for a customer to see their receipt is to provide them with a URL that would show them a digital version of everything they purchased, may it be goods or services. TaxCore considered a couple of options that could help customers verify their purchase; however, none proved to be as simple and efficient as a QR code.

There Shouldn't Be a Mismatch

POS (Point of Sale) must assure that every fiscal invoice is in line with accreditation requirements before separating the invoice body and the QR code that would be printed on the dot-matrix copy. However, it is extremely important that the QR code applied to the dot-matrix copy is the correct one. Any possible mismatch between the QR code and the content will be subject to penalty as this means the law has been broken.

If there are any minor discrepancies, the consumer has the right to report the invoice as this is an option TaxCore provides them with so to protect their rights.

Why a QR Code and Not a URL?

In one way or another, a URL must be shown to a consumer - however, it is the form in which they see it that matters. Using a dot-matrix printer to print an entire URL on paper would be a huge waste of material and ink. Mostly because printing an entire URL would take up a whole A4 size page. Providing a consumer with this would be unacceptable and unsustainable. Moreover, no one would know how to verify a URL in its usual, long format.

Of course, there is the option to print a short URL version. And although this is somewhat more convenient, it is still not a good option because Tax Authority couldn't guarantee such an invoice. Moreover, this type of URL would be only a temporary one, which is not a long-term solution. This would automatically create a set of problems, and POS not being able to pass accreditation is probably the biggest one of them.

For reasons like these, QR code is the best possible option as it doesn't take up a lot of space, and it can be easily scanned. Probably the best solution to include it is to print the QR code on a separate slip printer and attach it to a dot-matrix invoice. As long as the customer preserves their right to verify a receipt, a dot-matrix printer is allowed to remain in service. Even though it may seem expensive, it is probably the best and easiest solution.

Printing Receipts Is Not the Only Option

TaxCore solution doesn't require taxpayers to necessarily print receipts. It offers one more options better suited for the digital age.

Vendors can easily send a verification hyperlink to a consumer. In case a receipt is issued electronically, it doesn't need to contain a QR code. It comes with a hyperlink "click here to verify invoice."

By clicking this message, the customer opens a hyperlink which verifies the receipt, and that receipt's internal data are automatically sent to the tax authority. This means that for a receipt to be verified and legitimate, it doesn't need to be printed – a digital copy is more than enough.

Different applications of electronic receipts

In retail

A digital invoice can be sent to someone who is buying through a webshop. The webshop can show the digital receipt to a customer. This receipt doesn't need to contain a QR code, only the "click here to verify invoice" message. Clicking this message opens a hyperlink that serves as a means to verify the receipt.

This option can prove to be very useful when it comes to loyal customers as well. For example, your store offers a loyalty program that customers can benefit from using a card you issued. Assuming you have this customer's e-mail, by simply scanning their loyalty card, the receipt will be automatically forwarded to them. It will contain all the products the customer bought, and the option to verify the receipt as well.

Using different messaging apps

In case you are using mobile POS to issue receipts, you can easily implement the share option. This option lets you share receipts via chat applications such as WhatsApp, Viber or WeChat. It is another option that you can use with loyal customers that helps you save a lot of time and speed up the process.

Showing the QR code to customers on the display

Another way to stop wasting both money and paper on printing receipts is to show the QR code on the customer-facing display. The consumer can scan this QR code using any QR code scanner and automatically have the digital version of the receipt on their phone.

Business to Business

Digital invoice has been used in business to business purchases for a while now. However, TaxCore makes the entire process easier and faster; and this is something businesses should take advantage of. Digitial invoices of all kind can be easily forwarded back and forth digitally, which is a great step-up from the way things were done before. Sending invoices digitally consumes much less time and allows for more efficiency.

TaxCore as a system allows a lot of functions to taxpayers, and it is up to them to decide which one of these suit their needs the most. Additionally, POS vendors are always welcome to initialize changes and come up with many new functions that make doing business easier than ever.

Fiscal Invoice - Printing and sharing option image

Test Cases

Regardless of the type of invoicing system you are building, the same test cases apply:

1. Every Normal Sale invoice (NS) is assigned with a unique SDC Invoice No consisted of RequestedBy UID – SignedBy UID – OrdinalNumber
2. Refund is made as a result of the previous Sale, so you'll use the SDC Invoice No of the Sale invoice in the Ref No field
3. Copy (CS or CR) is made based on Normal receipt, so you'll use NS or NR SDC invoice No in the Ref No field

Different invoice use-cases and their appropriate labels are presented in the following table.

INVOICE TYPE	TRANSACTION TYPE	Invoice label
Normal	Sales	NS
Normal	Refund	NR
Copy	Sales	CS
Copy	Refund	CR
Training	Sales	TS
Training	Refund	TR
Proforma	Sales	PS
Proforma	Refund	PR

1. Issue Invoice
   A receipt must contain visible markings Receipt Type “NORMAL”, “COPY”, “TRAINING” or “PROFORMA”.

2. Issue Normal Sale or Refund Invoice With Buyer Identification
   Invoices with buyer identification can be issued in all invoice types and transaction types. At a beginning of the invoice creation cashier must ask the buyer for the Buyer's TIN, and optionally a Buyer Cost Center.

3. Special Cases
   This section covers special cases when it comes to issuing fiscal invoices. Its goal is to help you provide clear guidelines for your customers on how to handle these situations.

Issue Invoice

A receipt must contain visible markings Receipt Type “NORMAL”, “COPY”, “TRAINING” or “PROFORMA”.

Steps

1.

Cashier on Accredited POS selects an invoice type and then registers the transaction by:

- typing items,
- selecting items from the previously made list,
- scanning bar code with a bar code reader.

2. The Cashier chooses the payment method and finishes the invoice.
3. Next, an accredited POS sends a message to V-SDC/E-SDC. After a successful invoice data verification, the invoice is signed, counters and totals are updated and internal data is completed.
4. The V-SDC/E-SDS sends back an Invoice response to Accredited POS.
5. The receipt is delivered to the customer.

Expected Result

A fiscal receipt is the final result of this procedure. The receipt can be printed or shared via an email or other chat application message if a customer requires it.

Every receipt is digitally signed. Internal data is stored in the TaxCore database.

A valid QR code is at the end of the receipt (or the verification URL is displayed in a clickable hyperlink form). The receipt counter is in the form a/b IL (a-total number of signed invoices per type/ b-total number of signed invoices, IL – Invoice label).

Issue Normal Sale or Refund Invoice With Buyer Identification

Invoices with buyer identification can be issued in all invoice types and transaction types. At a beginning of the invoice creation cashier must ask the buyer for the Buyer's TIN, and optionally a Buyer Cost Center.

Special Cases

This section covers special cases when it comes to issuing fiscal invoices. Its goal is to help you provide clear guidelines for your customers on how to handle these situations.

NOTE:
This article has an advisory status and simply offers helpful examples. DO NOT COPY-PASTE THESE GUIDELINES TO YOUR USER MANUAL without checking the legal requirements regarding fiscal law that apply in your country or jurisdiction. It is the responsibility of each POS developer to create a POS solution that will enable their

customers to comply with the requirements.

General rules

All payments must conform to the following rules:

- If the payment is not considered a taxpayer income, then no fiscal receipt shall be issued for that payment (as it is not an actual transaction).
- If the payment is considered as advance payment, then the fiscal receipt shall be issued as a NORMAL-SALE transaction.

A fiscal receipt is required when tax liability occurs which is in most cases when:

- payment is taken by a taxpayer
- goods/services are delivered to a customer

This is constituted by the "time of supply" rule and affects different business activities such as hospitality, construction, medical services, legal cases and others, whereby advance payment can be canceled or reduced due to the contract amendments. This provokes a refund or a credit note.

NOTE:
All the variations on the subject could be differently regulated differently, as jurisdiction-specific. Note that TaxCore is designed for monitoring, so the Tax Authority doesn't expect settlement of the amount registered by an EFD (POS+SDC) at the moment fiscal receipt is issued.

The following examples illustrate applications of these rules:

Deposit

Depending on the purpose of a taken deposit, this case can be resolved in two ways:

1. If the Deposit is subject of some internal agreement/document (it is not the taxpayer's income), then no fiscal receipt is issued for the Deposit (for example, cash necessary to start a business day in retail).
2. If the Deposit is considered an advance payment, then the fiscal receipt is issued as a NORMAL-SALE transaction. It is advisable the item name is descriptive enough to clearly state that this payment is partial ("Project X - advance payment 30%")

When Items are ready for Delivery and final SALE, the POS should provide one or both of the following options:

1. Create another transaction to complete the deliverable - new NORMAL-SALE. It is advisable the item name is descriptive enough to clearly state that this payment is partial and completes the final price ("Project X - final payment 70%"), or
2.

Create two new transactions: 1) NORMAL-REFUND to cancel the previously recorded amount, followed by 2) NORMAL-SALE with the full price (“Project X”).

Quotes / Pre-sale

A sales quote or pre-sale gives an estimated price of a product or service and allows a prospective buyer to see the costs that will be involved for desired work.

Assuming no advance payment takes part in the transaction, this is a PROFORMA-SALE transaction.

Installments/Layby

This is treated as a series of separate, successive supplies of services corresponding to the successive parts of the period of the lease or agreement, or as determined by the law. Each successive supply is treated as occurring on the earlier of the date on which the payment for that successive supply is due or received.

This means that each installment is treated as a separate payment and should be covered with a NORMAL-SALE invoice. It is advised to name each installment with a descriptive name, similar to “10% of the (Item Name)” or "(Service Name) - first installment".

When the last installment payment is made, POS must provide one or both of the following options:

1. Create another transaction for the last installment - “15% of the (Item Name)” or "(Service Name) - fifth/last installment", or
2. For each previously paid installment, create a NORMAL-REFUND invoice, followed with one NORMAL-SALE invoice (containing the item or service name and the full price).