For POS Developers

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?

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 testing purposes
4. Consult all the sections in these technical instructions to see understand all the requirements and how they should be implemented
5. Use the testing applications on the Developer Portal to test your POS operation
    - Dev-ESDC for testing POS operation with E-SDC service
    - VSDC Request Submitter for testing POS operation with E-SDC service
6. Compile user documentation for your POS
7. Use the My Accreditations section on Developer Portal to apply for accrediting your POS

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
- Differences Between E-SDC and V-SDC

- Recommendation Examples

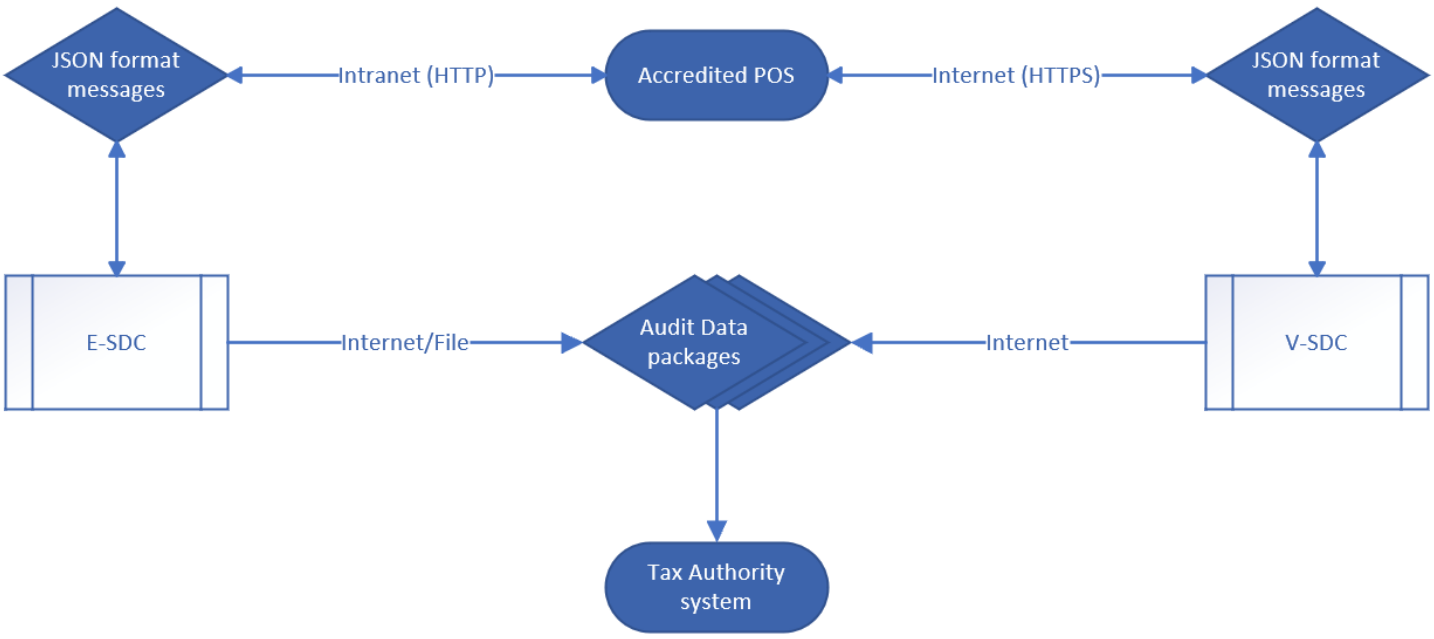


Image of POS connection options with SDC services

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

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

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

4. 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.

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

6. Protocols
   Each POS or Invoicing System must communication with either V-SDC or E-SDC to fiscalize invoice.

7. 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.

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

Related articles

- EFD Vendors General Information

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
HASH property of InvoiceFiscalizationRequest or GetSignedInvoice	NO	YES
Internet connection required to work	YES	NO

*PAC code to be sent in the InvoiceFiscalizationRequest as property, as 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 PIN code to be sent through VerifyPIN method in order for ESDC to be activated

1. Differences-Between-VSDC-and-ESDC
2. Recommendation Examples
   This section gives examples of the most common implementation scenarios, for different end-users.

Differences Between VSDC and ESDC

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	API methods used for E-SDC are: *VerifyPIN*,

GetTaxAuthorityParams.	*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 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

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.

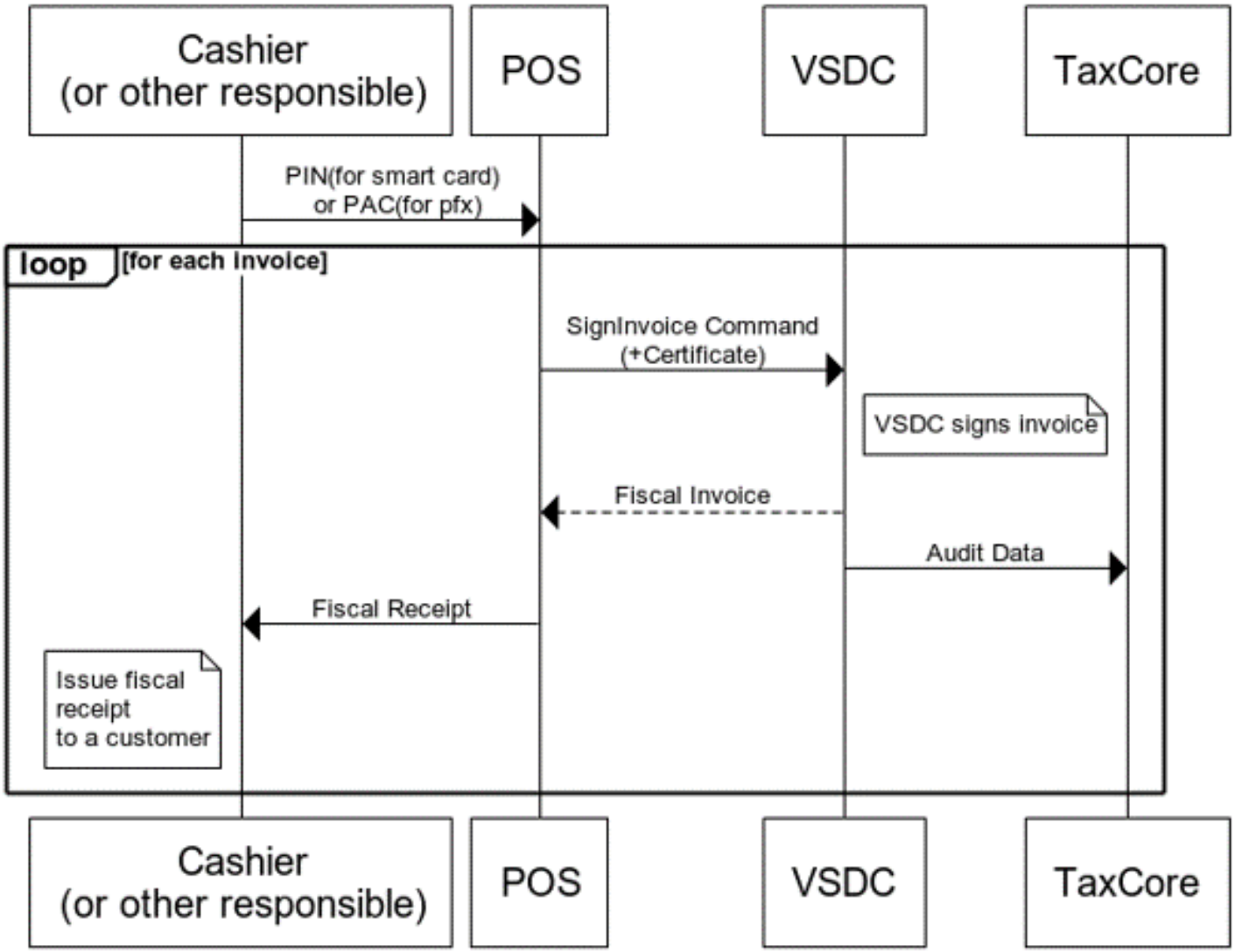
Web Shops

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



Typical Process Flow - Image of a V-SDC process flow

Provide a valid PIN/PAC

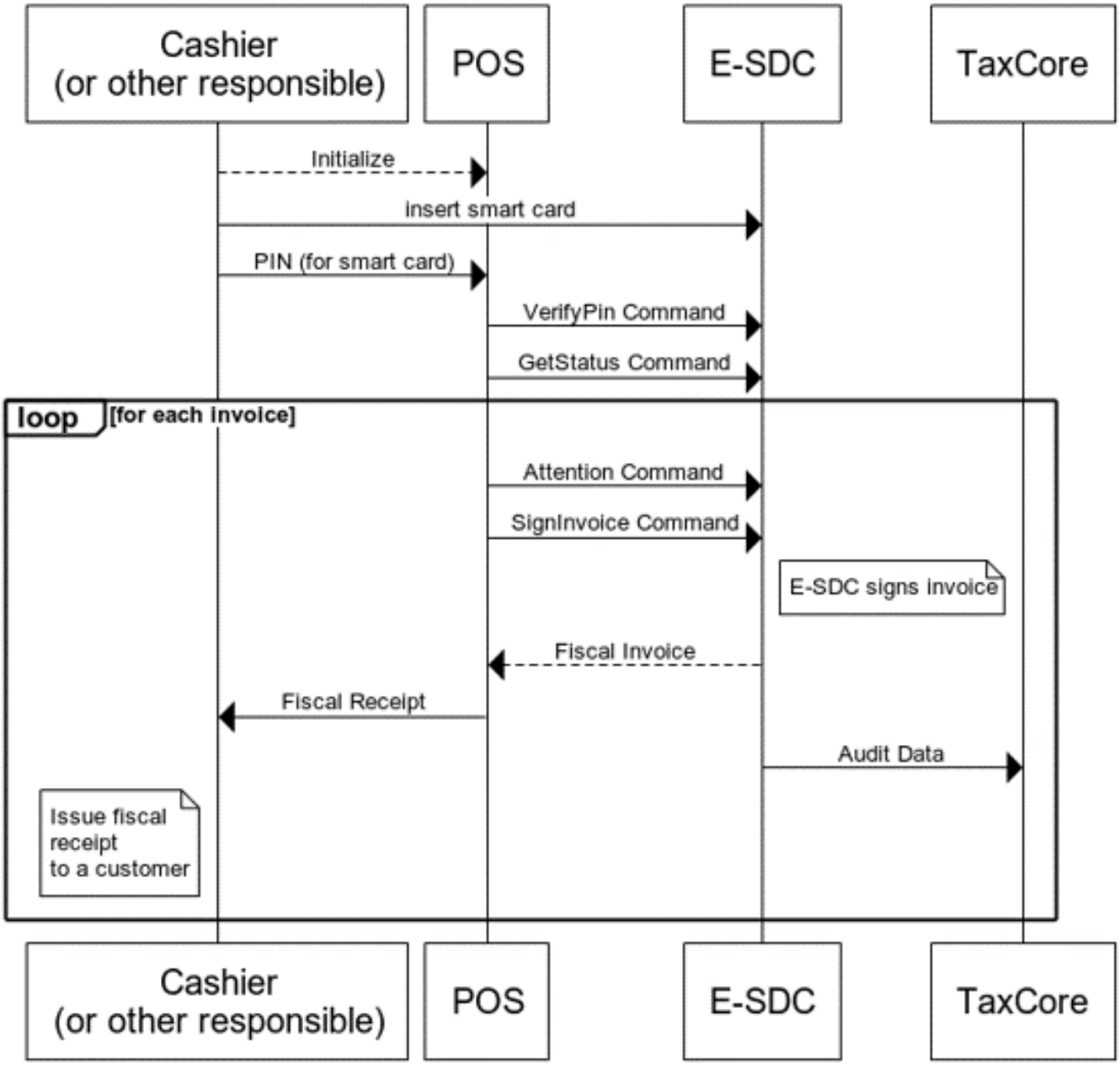
The process begins with a cashier creating an invoice request on an accredited POS. An invoice request transmits the transaction data (POS date and time, list of items, transaction type, payment type, etc.) to an available V-SDC.

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



Typical Process Flow - Image of an 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 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 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 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 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 Service'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 send 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)

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: https://vsdc.sandbox.taxcore.online/.

2. Environments
   In case the certificate does not contain VSDC URL, use one of the following addresses depending on the target environment:

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

4. 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: https://vsdc.sandbox.taxcore.online/.

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.

NOTE:
These will be the OIDs per environment:

Sandbox - 1.3.6.1.4.1.49952.**5.7**.3.7
Fiji Production - 1.3.6.1.4.1.49952.**3.2**.3.7
Samoa Production - 1.3.6.1.4.1.49952.**3.6**.3.7

Accessing V-SDC API - Image of the certificate information

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

NOTE:
These will be the OIDs per environment:

Sandbox - 1.3.6.1.4.1.49952.5.7.**7**
Fiji Production - 1.3.6.1.4.1.49952.3.2.**7**
Samoa Production - 1.3.6.1.4.1.49952.3.6.**7**

7. Read the value of this OID containing the URL of V-SDC.Api

Accessing V-SDC API - Image of an OID containing V-SDC URL

Environments

In case the certificate does not contain VSDC URL, use one of the following addresses depending on the target environment:

Sandbox environment:

https://vsdc.sandbox.taxcore.online/Swagger

API is designed and based on OpenAPI-Specification V2 (https://github.com/OAI/OpenAPI-Specification). You can use OpenAPI-Specification code generators (e.g. https://swagger.io/swagger-codegen/) to quickly build a proxy library for almost any programming language and platform.

Samoa:

https://vsdc.tims.revenue.gov.ws/

Fiji:

https://vsdc.staging.vms.frcs.org.fj/ (for Staging - testing)

https://vsdc.vms.frcs.org.fj/ (for Production)

USA, WA:

https://vsdc.wa.us.taxcore.online/

This page contains *SignInvoice* service operation details, invoice format and some basic examples.

Client Authentication

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

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

Digital Certificates may be distributed in two formats

1. PKCS12 - file (*.pfx or *.p12)
2. On the Smart Cards

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 Service, you will need to provide PAC value in field PAC.

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

static void Main(string[] args)
{

    string invoiceRequest = @"{
"DateAndTimeOfIssue": "2017-06-15T08:56:23.286Z",
"Cashier": "Oliver",
"BD": "8902798054",
"BuyerCostCenterId": "",
"IT": "Normal",
"TT": "Sale",
"PaymentType": "Cash",
"InvoiceNumber": "POS2017/998",
"ReferentDocumentNumber": "ABCD1234-EFGH5678-198",
       "PAC":"",
"Options":{
"OmitQRCodeGen": "1",
"OmitTextualRepresentation": "1"},
       "Items": [{
"Name": "Sport-100 Helmet, Blue",
"Quantity": 2,
"UnitPrice": 34.23,
"Labels": ["A"],
"TotalAmount": 68.46}],
"Hash": "W33lEEgkSRsqTFMO86a8Og=="}";

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

    GetClientAndHandler(out handler, out client);

    var response = client.PostAsync($"api/Sign/SignInvoice", 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();
}

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. Anatomy of a Fiscal Receipt
   A receipt records the sale of goods or the provision of a service. The table below explains the structure of a fiscal receipt. **All elements are mandatory** unless specified otherwise in the detailed explanation below. POS is free to print any content (coupons, logos, etc.) before the beginning and after the ending mark of the fiscal invoice.

3. Tax Amounts
   It is essential to note, that **POS never uses other taxes except the ones received from SDC**. POS

displays the total prices and only the tax values received from and SDC device, in the format described in the previous section.

4. Calculate 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.

5. 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).

Anatomy of a Fiscal Receipt

A receipt records the sale of goods or the provision of a service. The table below explains the structure of a fiscal receipt. **All elements are mandatory** unless specified otherwise in the detailed explanation below. POS is free to print any content (coupons, logos, etc.) before the beginning and after the ending mark of the fiscal invoice.

Elements of a fiscal receipt

Title line – marks the beginning of the fiscal part of a receipt

```
============ FISCAL INVOICE ============
```

Header data is provided by an SDC during fiscalization of the invoice and returned to POS as part of the InvoiceFiscalizationResult object (explained in section Sign Invoice. Values are extracted from the subject field of

the digital certificate stored in the Secure Element Applet.

```
TIN:                          502579006
Company:                         Golf V
Store:                        Sun Store
Address:                  7 Someplace
District:                         Suva
```

Cashier identification is mandatory for every jurisdiction where local regulations mandate POS to send particular data such as Employee ID or some other information that uniquely identifies the POS cashier.

```
Cashier TIN:                1234567890
```

Buyer TIN is mandatory **only** for B2B transactions and in that case, it must be printed on the receipt. When creating a B2B invoice for exporting goods/services, a country code prefix must be added to the Buyer TIN.

B2C transactions are anonymous so they don't require buyer identification. However, the cashier must always have the option to enter a Buyer TIN on customer demand.

Buyer Cost Center is optional and reserved for further use, it must be present **only** for B2B transactions. **POS (Invoice) Number** and **POS (Invoice) time** are optional fields.

```
Buyer TIN:                  5123456789
Buyer Cost Centre:                 123
POS number:                POS2017/998
POS time:          15/6/2017 8:56:23AM
```

Reference (Document) Number is always mandatory for Refund or Copy transactions. **Ref No** is printed on the receipt, containing the **SDC Invoice No** of the document which is being referenced in the format of *RequestedBy-SignedBy-OrdinalNumber*.

Reference combinations are as follows:

		I must/can reference...							
		Normal Sale	Normal Refund	Proforma Sale	Proforma Refund	Copy Sale	Copy Refund	Training Sale	Training Refund
When I issue...	Normal Sale			optional					
	Normal Refund	mandatory							
	Proforma Sale			optional					
	Proforma Refund			mandatory					
	Copy Sale	mandatory		mandatory					
	Copy Refund		mandatory		mandatory				
	Training Sale								
	Training Refund							mandatory	

Anatomy of a Fiscal Receipt - Reference combinations image

Optional reference may be mandatory if requested by a Tax Authority (obligation may apply to certain business activity).

All Reference Numbers are created for connected transactions, and if a **Ref No** is sent to an SDC, it must be displayed on the receipt journal.

If a Copy or Refund is issued for the transaction that was recorded before the introduction of fiscalization, POS should send XXXXXXXX-XXXXXXXX-1 as the value of **Ref No** field.

```
Ref no:          P22VC8VR-JTJC5V65-114906
```

Invoice and **transaction type** description. Normal Sale and Normal Refund are the most common types. Other types of transactions and invoices are defined in Invoice and transaction types.

```
--------------NORMAL SALE---------------
```

List of items with **gross price**, **tax labels**, **unit price** and **quantity**. Tax Labels and their validity dates are published by the Tax Service and they are mandatory for each item, even when the price is 0.00.

When applying discounts, **Unit price** of the line item displayed on the journal (Price column) shows the discounted price, after **all** discounts have been applied.

```
Items
========================================
Name       Price          Qty.      Total
Sport-100 Helmet, Blue (E)
           34.99           10     349.90
Mountain Bike Socks, M (A)
            9.03            4      36.12
HL Road Frame - Red, 58 (F, A)
         1431.50            2    2863.00
Plastic bag (P)
            0.10            5       0.50
```

Total Purchase, **Tax items** and **Total Tax** are calculated by V-SDC or E-SDC during fiscalization of the invoice and are returned to POS as a part of the response.
Payment Method: Cash, Card, Check, Wire Transfer, Voucher, Mobile Money, or Other. Taxpayer's tax liability is based on these tax amounts, calculated by V-SDC or E-SDC. The calculation is explained in the section Tax Amounts. In case of Refund receipts, **Total Refunded** should be printed instead of **Total Purchase**.

```
--------------------------------------
Total Purchase:                3249.52
Payment Method:                   Cash
======================================
Label      Name       Rate          Tax
E           STT      6.00%        19.81
A           VAT      9.00%       219.51
F          ECAL     10.00%       240.59
P            PB      0.10%         0.50
--------------------------------------
Total Tax:                      480.41
```

Fiscal metadata added to the invoice through fiscalization. **SDC Invoice No** - Combination of Requested By (7AF4D923), Signed By (E3B30A31) and Ordinal Invoice Number (234) is a system-wide unique identification of fiscal invoice. It may be used instead of the current receipt/invoice number generated by POS. **SDC Time** is the official date and time relevant to the tax calculation and reporting. **Invoice Counter** is generated by V-SDC or E-SDC and explained in section Sign Invoice, field IC.

```
======================================
SDC Time:            2017-06-15 08:56:25
SDC Invoice No:    7AF4D923-E3B30A31-234
Invoice Counter:               230/234NS
======================================
```

QR Code contains Invoice verification URL. QR Code also contains Internal data and digital signature used for the invoice verification. Every invoice is verifiable by the customer immediately after fiscalization. In case an invoice/receipt is delivered as an electronic document (email), QR Code shall be substituted with Invoice verification URL in (clickable) hyperlink format.

NOTE:
This is just a sample QR code image, not an actual URL.

Anatomy of a Fiscal Receipt - Image of a QR Code

Title line – marks the end of the fiscal part of a receipt

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

Custom message returned from V-SDC or E-SDC

```
 This is a custom message.
```

Example of a fiscal receipt

```
============ FISCAL INVOICE ============
TIN:                                  123
Company:                             Test
Store:                               Test
Address:                            nesto
District:                              Ba
Cashier TIN:                    987654321
---------------NORMAL SALE---------------
                      Items
```

==========

Name	Price	Qty.	Total
pizza (P)			
	3.30	2	6.60
juice (P)			
	2.45	3	7.35
chocolate (E)			
	1.50	4	6.00

Total Purchase: 19.95

Payment Method: Card

==========

Label	Name	Rate	Tax
P	PBL	0.20$	1.00
E	STT	6.00%	0.34

Total Tax: 1.34

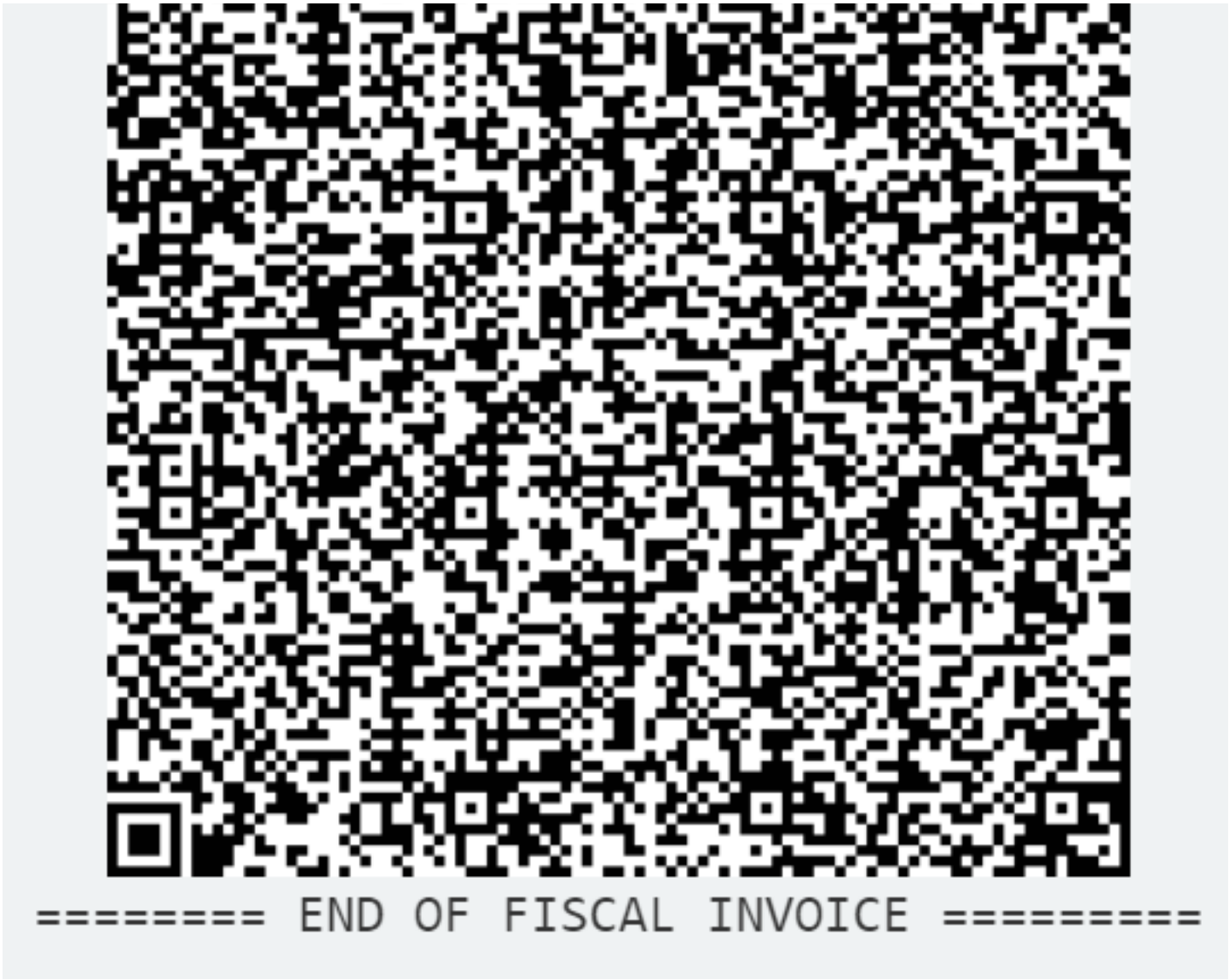
==========

SDC Time: 25/03/2020 09:29:58

SDC Invoice No: XRSYSZWL-373ZQXF4-1

Invoice Counter: 1/1NS

==========



Anatomy of a Fiscal Receipt - Image of a Fiscal invoice

Normal Refund Receipt

Receipt for Normal Refund Invoice must contain visible markings **REFUND**, below the receipt header and above the item description section. Totals on the refund receipt are displayed as negative values, starting with (-), except for Total Refunded. Tax Items are displayed as positive values.

For Refund transaction type, the **Ref no** element (Reference Document Number) is mandatory.

Example:

```
========== FISCAL INVOICE ===========
TIN:                           502579006
Company:                          Golf V
Store:                         Sun Store
Address:                     7 Someplace
District:                           Suva
Cashier TIN:                  123456789
POS number:                89347415-2017
POS time:            2018-03-09 14:57:25
Ref no:            7AF4D923-E3B30A31-234
------------NORMAL REFUND--------------
Items
=========================================
Name    Price        Qty.          Total
Sport-100 Helmet, Blue (E)
        34.99          10        -349.90
Mountain Bike Socks, M (A)
         9.03           4         -36.12
-----------------------------------------
Total Refunded:                   386.02
Payment Method:                     Cash
=========================================
Label        Name     Rate           Tax
E             STT    6.00%         19.81
A             VAT    9.00%          2.98
-----------------------------------------
Total Tax:                         22.79
=========================================
SDC Time:            2018-03-09 14:57:46
SDC Invoice No:    7AF4D923-E3B30A31-235
Invoice Counter:                4/235NR
=========================================
---- QR code omitted for simplicity ----

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

Normal Refund Receipt - Image of a normal refund

Training or Proforma or Copy Receipt

Receipt for Training or Proforma or Copy Invoice must contain visible markings “TRAINING” or “PROFORMA” or “COPY”, below the receipt header and above the item description section.

Receipt must also contain **THIS IS NOT A FISCAL INVOICE** below the total amount payable. Font size is at least twice the size of the text on the receipt that specifies the total amount payable.

Training or Proforma or Copy receipt is produced in the same way as normal, with an exception that totals are not accounted for.

For Copy invoice type **Reference Document Number** element is mandatory.

Example:

```
===== THIS IS NOT A FISCAL RECEIPT =====
TIN:                              502579006
Company:                             Golf V
Store:                            Sun Store
Address:                        7 Someplace
District:                              Suva
Cashier TIN:                      123456789
POS number:                    89347415-2017
POS time:                2018-03-09 14:57:25
-------------TRAINING SALE--------------
Items
==========================================
Name    Price        Qty.           Total
Sport-100 Helmet, Blue (E)
        34.99          10          349.90
Mountain Bike Socks, M (A)
         9.03           4           36.12
------------------------------------------
Total Purchase:                     386.02
Payment Method:                       Cash
==========================================
      THIS IS NOT A FISCAL INVOICE
==========================================
Label        Name     Rate            Tax
E             STT    6.00%          19.81
A             VAT    9.00%           2.98
------------------------------------------
Total Tax:                           22.79
========================================
SDC Time:                2018-03-08 14:57:46
SDC Invoice No:     7AF4D923-E3B30A31-236
Invoice Counter:                   1/236TS
==========================================
---- QR code omitted for simplicity ----
===== THIS IS NOT A FISCAL RECEIPT =====
```

Training or Proforma or Copy Receipt - Image of a training receipt

Tax Amounts

It is essential to note, that **POS never uses other taxes except the ones received from SDC**. POS displays the total prices and only the tax values received from and SDC device, in the format described in the previous section.

A tax label can fall into one of the three tax types: Tax, Tax on Total and Amount per Quantity.

Tax amount for a tax label is calculated per the following formulas:

Tax Type	Formula	Explanation
Tax	**Tax Amount** = ∑(Base price * (**Rate/100**))	For each item with this label.
Tax on Total	**Tax Amount** = ∑(Total price * (**Rate/100**))	For each item with this label (Total price here is item base price with all other

		regular taxes included.
Tax on Quantity	**Tax Amount** = ∑(Fix amount * quantity)	For each item with this label, item quantity is multiplied with fix amount as defined by tax Service. If other tax labels are defined on the item, those taxes are calculated on the remainder.

Calculate 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.

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 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.

In order to calculate a tax, the following algorithm is implemented:

1. Make an array of distinct tax labels associated with the items in the POS request (e.g. A, B, C, F, ...).
2. Calculate the tax amount for each individual label in the array:
    - Iterate through all items in the POS request
    - For each item, calculate tax amounts. One item has one or more tax labels, and each label represents a tax amount. Each tax amount is a part of an item's total price. These tax amounts are calculated as follows:
        - ♠ If an item has a label from the **amount-on-quantity** category applied, subtract the tax rate amount for that label, multiplied with quantity, from the item total price. The resulting amount (the remainder), is used in all further calculation steps instead of item total amount.
        - ♠ If none of the labels' tax category type is ***tax-on-total*** (category 1):
            - ♠ Tax amount for one label is:

$$\frac{item\ total\ amount\ *\ label\ rate}{(100 + \Sigma(all\ tax - on - net\ rates\ on\ item))}$$

Example 1: An item has a total price of 10$ and applied labels: A(5%) and B(6%).

$$A = \frac{10\$ * 5}{(100 + \Sigma(5+6))} \quad B = \frac{10\$ * 6}{(100 + \Sigma(5+6))}$$

Tax amount for label A=0.4505$ and for label B=0.5405$.

- If any of the labels' tax category is ***tax-on-total*** (category 1):
  - Tax amount for every label whose category type is ***tax-on-total*** (category 1) is:

$$\frac{item\ total\ amount}{(1\ +\ \Sigma(all\ tax - on - total\ rates)/100)} * \frac{label\ rate}{100}$$

  - Tax amount for every other label from category 0 is:

$$\frac{item\ total\ amount}{(1\ +\ \Sigma(all\ tax - on - total\ rates)/100)} * \frac{label\ rate}{(100 + \Sigma(all\ tax - on - net\ rates\ on\ item))}$$

  Example 2: Item has a total price 10$ and applied labels: A(5% tax-on-net), B(6% tax-on-net), C(3% tax-on-total) and F(4% tax-on-total).

$$A = \frac{10\$}{(1\ +\ \Sigma(3+4)/100)} * \frac{5}{(100 + \Sigma(5+6))}$$

$$B = \frac{10\$}{(1\ +\ \Sigma(3+4)/100)} * \frac{6}{(100 + \Sigma(5+6))}$$

$$C = \frac{10\$}{(1\ +\ \Sigma(3+4)/100)} * \frac{3}{100}$$

$$F = \frac{10\$}{(1\ +\ \Sigma(3+4)/100)} * \frac{4}{100}$$

Tax amount for label A=0.4210$, for label B=0.5052$, for label C=0.2804$ and for label F=0.3738$.

- Add calculated labels' tax amounts to the label's total amount sum.

  Example 3: the request contains two items from Example 1 and Example 2, the total sum for labels are: A=0.8715$, B=1.0457$, C=0.2804$, F=0.3738$.

```
- Add fixed tax amounts, multiplied with quantity, to the respective labels' to

  Example 4: An item has quantity 2 with total price 10$ and applied labels: A(

  Example 5: An item has quantity 2 with total price 10$ and applied labels: A(
```

3. After all of the items have been processed, calculate the tax amount for all tax categories found in the request. One tax category can consist of one or more tax labels (e.g. A, B...). The tax amount for a tax category is a sum of all label tax amounts related to the category.

   Example 6: The request contains two items from Example 1 and Example 2. Labels A and B are VAT category, C is STT category and F is ET category. Total VAT=1.9172$, STT=0.2804$ and ET=0.3738$.

Once the Tax calculation is completed, assign GroupId of the active tax rate group to the field *TaxGroupRevision* of *InvoiceFiscalizationResult*.

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

Buyer TIN on Export Invoices

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

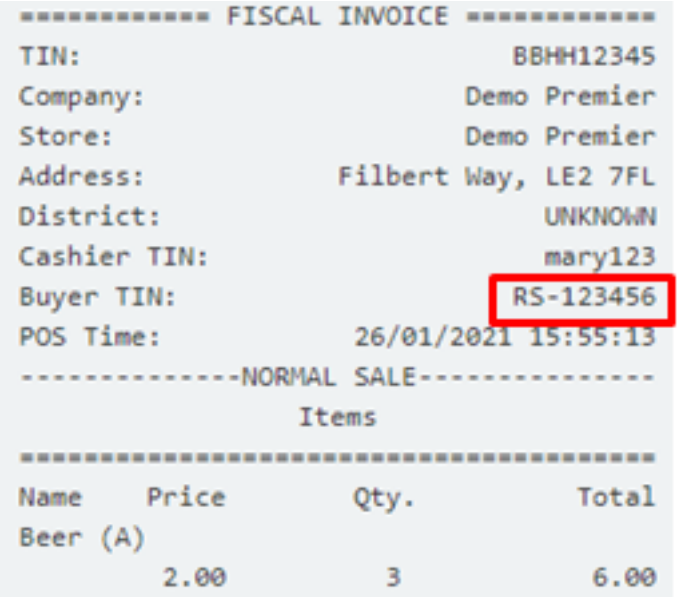
Every time when a taxpayer issues a **B2B fiscal invoice**, it **must contain a valid Buyer TIN** which uniquely identifies the buyer of sold goods/services.

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**.

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

NOTE:
ISO 3166-1 alpha-2 codes are two-letter country codes defined in ISO 3166-1, part of the ISO 3166 standard published by the International Organization for Standardization (ISO).

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



Buyer TIN on Export Invoices - Image of a buyer TIN number on a fiscal invoice

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.

Protocols

Introduction

Each POS or Invoicing System must communication with either V-SDC or E-SDC to fiscalize invoice.

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 (SDC Section) API to all POS and Invoicing systems. Correctness of API implementation is, among other criteria, one of the requirements for accreditation.

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

1. **POS connect to Taxcore.Api** to get environment configuration and tax rates [Optional]
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**. Digital certificate delivered in **PKCS12** format and PAC are used to authenticate to V-SDC.Api
4. **POS Connects to V-SDC**. 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 network level. Fiscalization of 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

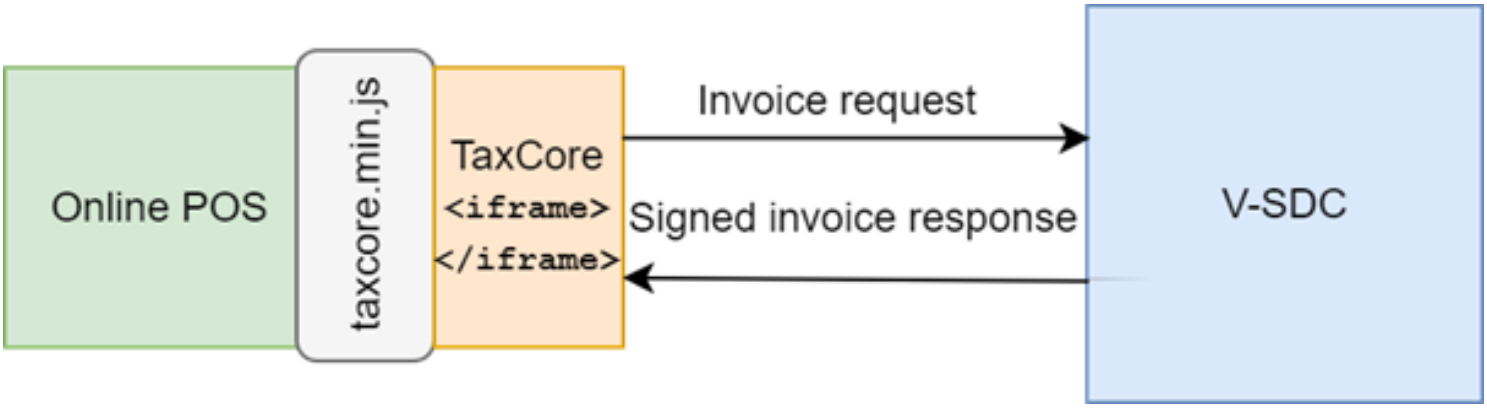
NOTE:
Both components must exist at the integration point for successful integration. JavaScript will first try to find taxcore sign element with id="taxcore_sign_element". If it can't be found, the exception will be thrown with an appropriate message (e.g. "*Could not find taxCoreElement. Check if you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

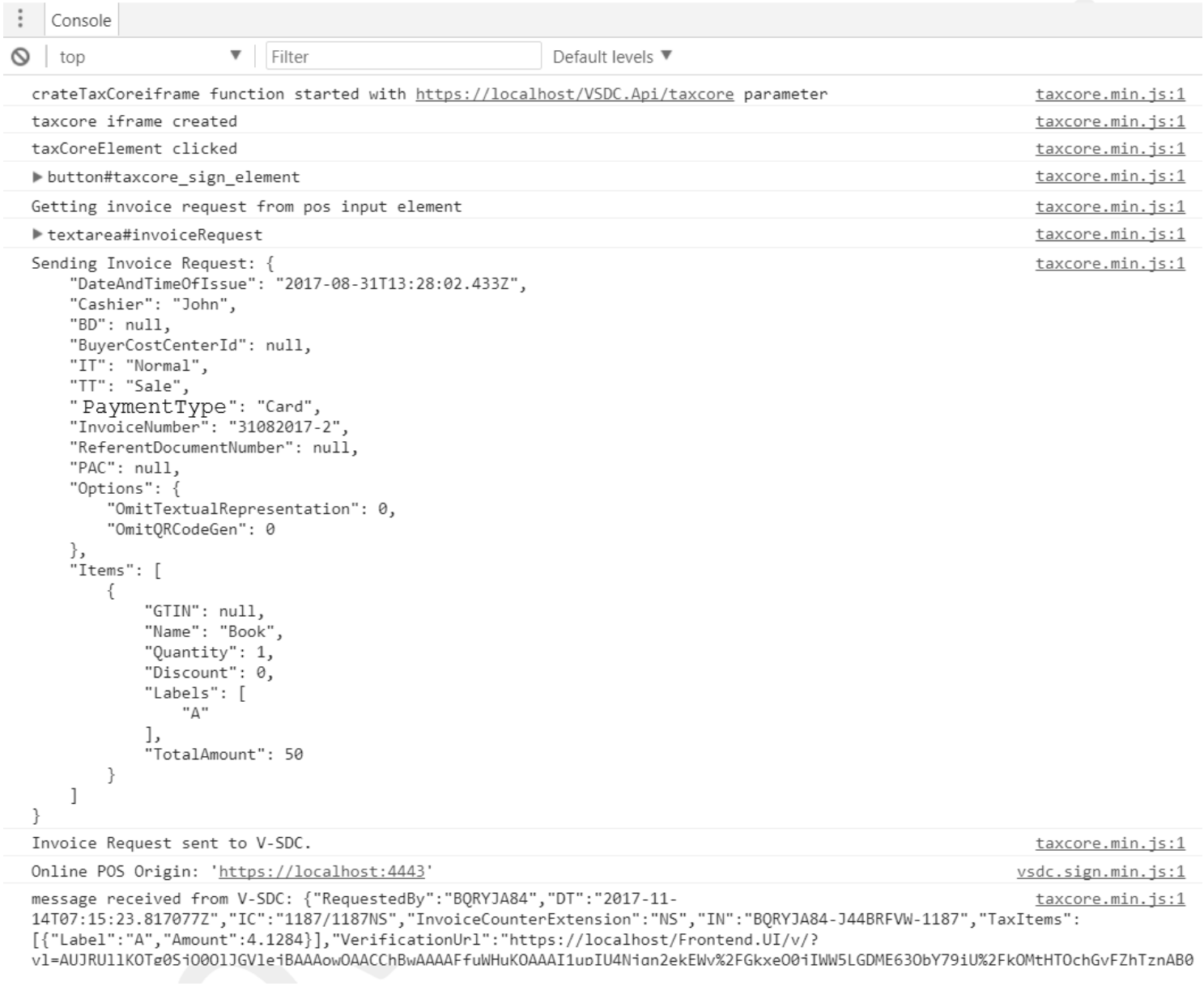
TaxCore Sign element is nothing more than an HTML tag with predefined id, that you'll place at your online POS page. Its task is to collect invoice data and forward them to V-SDC.

> **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."
```

```
    "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": "2017-08-31T13:28:02.433Z",
                "Cashier": "John",
                "BD": null,
                "BuyerCostCenterId": null,
                "IT": "Normal",
                "TT": "Sale",
                "PaymentType": "Card",
                "InvoiceNumber": "31082017-2",
                "ReferentDocumentNumber": null,
                "PAC": null,
                "Options": {
                    "OmitTextualRepresentation": 0,
                    "OmitQRCodeGen": 0
                },
                "Items": [
                    {
                        "GTIN": null,
                        "Name": "Book",
                        "Quantity": 1,
                        "Labels": [
                            "A"
                        ],
                        "TotalAmount": 50
                    }
                ]
            };

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

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 B2B Invoice
   B2B invoices 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 B2B Invoice

B2B invoices 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).