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

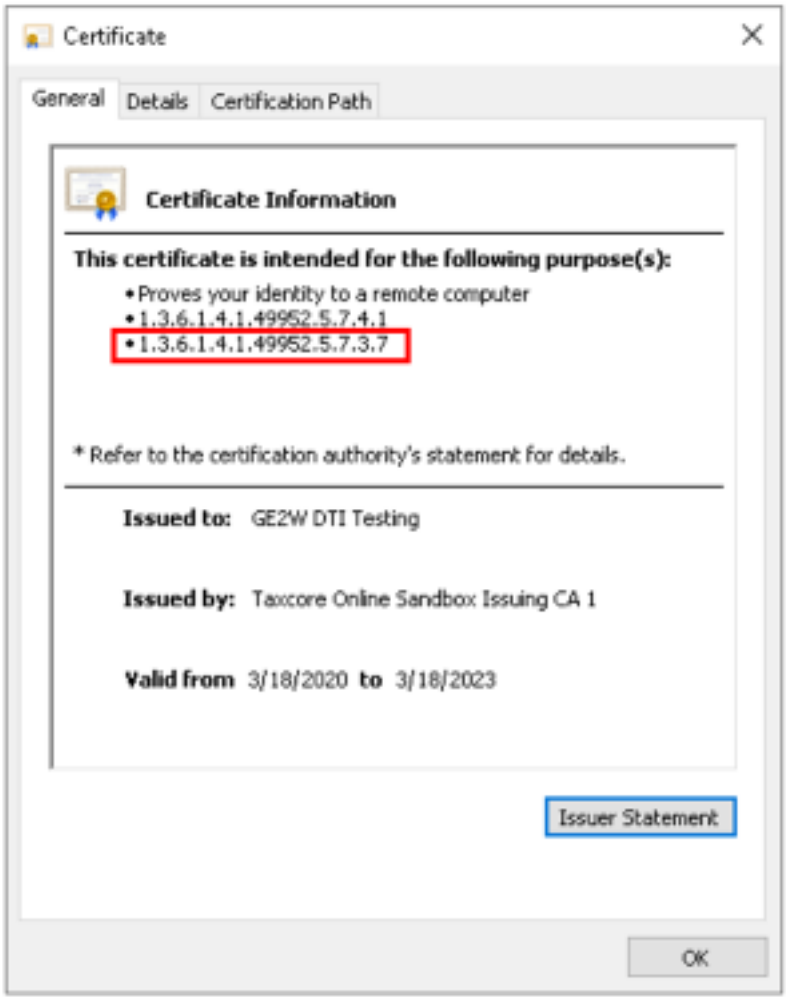
Once valid Test certificate(s) are obtained, you can access the V-SDC.Api description on the following URL: https://vsdc.sandbox.taxcore.online/.

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



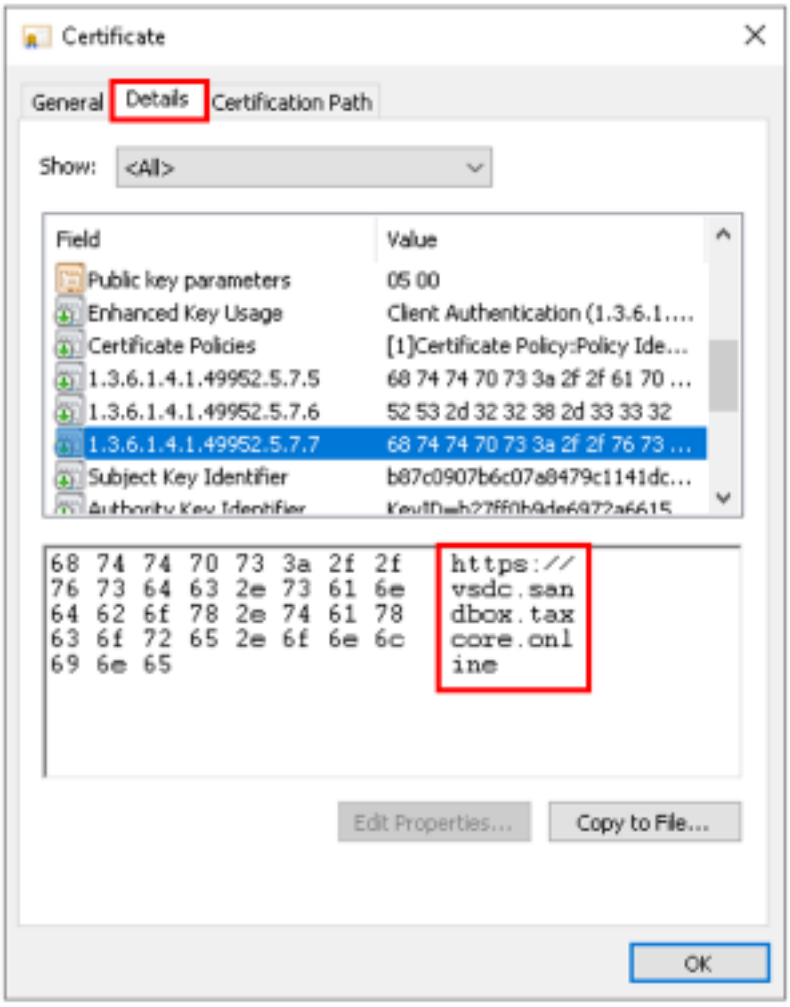
Accessing V-SDC API - Image of the certificate information

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];
}
```