Zatca E-Invoice .Net SDK

Zatca .Net SDK is a combination of many tools to easily integrate E-Invoice into exsisting or new .Net applications.
Here you can find a complete guide on how to make use of the .Net SDK.
- API Documentation
- CLI Documentation

This Document also contains Guide for installing .Net SDKs and Functions Benchmarks for reference.
Target frameworks for all tools are kept as low as possible to support as-many applications as possible.

.Net SDK Consists of: - Contracts DLL
- Concrete Implementation
- Console Application

Contracts:

Contains contracts and models for all available public APIs of the SDK.
All contracts and models are xml commented to provide an easy way to use.
Assembly: Zatca.EInvoice.SDK.Contracts.dll
Namespace: Zatca.EInvoice.SDK.Contracts
Target: .Net Standard 2 and .Net Framework 4

Concrete Implementation:

A concrete implementation of the above contracts following the industry standards for code quality.
Assembly: Zatca.EInvoice.SDK.dll
Namespace: Zatca.EInvoice.SDK
Target: .Net Core 3.1 and .Net Framework 4.

Console Application

This documentation provides detailed explanations and examples for each feature and command available in the console application.
Assembly: Zatca.EInvoice.SDK.CLI [fatooraNet.exe]
Target: .Net Core 3.1

SDK Public API Documentation:

Generate CSR

Description

Generates a Certificate Signing Request (CSR) ### Method Signature

```
CsrResult ICsrGenerator.GenerateCsr(CsrGenerationDto csrGenerationDto, bool
pemFormat, EnvironmentType environment);
```

Inputs

- (csrGenerationDto)[#CsrGenerationDto]
- (pemFormat): whether to generate csr and private key in pem format, if false: the csr file will be generated encoded base64 and private key file will be generated without header or footer
- (environment) : The environment could be Production or Simulation or NonProduction , Each of them has a template name that we use in in the extension in X509ExtensionsGenerator ### Output
- CsrResult: CSR Result

Generate Request

Description

Generates E-Invoice API request from the specified E-Invoice XML document in json format. ### Method Signature

```
public RequestResult GenerateRequest(XmlDocument eInvoice);
```

Inputs

- eInvoice: E-Invoice XML Document ### Output
- (RequestResult)

Generate Hash

Description

Generates E-Invoice Hash ### Method Signature

```
HashResult IEInvoiceHashGenerator.GenerateEInvoiceHashing(XmlDocument
eInvoice);
```

Inputs

- eInvoice: E-Invoice XML Document ### Output
- HashResult: Operation Result

Generate QR Code

Description

Generates E-Invoice QR ### Method Signature

```
QRResult IEInvoiceQRGenerator.GenerateEInvoiceQRCode(XmlDocument eInvoice);
```

Inputs

- eInvoice: E-Invoice XML Document ### Output
- QRResult: Operation Result

Sign E-Invoice

Description

Signing E-Invoice contains the next steps: - Generating Hashing - Generating Signature - Populating Data - Generating and populating QR ### Method Signature

```
SignResult IEInvoiceSigner.SignDocument(XmlDocument eInvoice, string
certificateContent, string privateKeyContent);
```

Inputs

- eInvoice: E-Invoice XML Document
- certificateContent: The content of certificate file as string
- privateKeyContent: The content of private key as string ### Output
- SignResult: Operation Result

Validate E-Invoice

Description

There are two types of E-Invoices “Simplified” & “Standard” - Validation of “Simplified” E-Invoice contains the next steps - Validate XSD - Validate EN Schema Tron - Validate KSA Schema Tron - Validate Signature - Validate QR - Validate PIH - Validation of “Standard” E-Invoice contains the next steps - Validate XSD - Validate EN Schema Tron - Validate KSA Schema Tron - Validate PIH ### Method Signature

```
ValidationResult IEInvoiceValidator.ValidateEInvoice(XmlDocument eInvoice,
string certificateFileContent, string pihFileContent);
```

Inputs

- eInvoice: E-Invoice XML Document
- certificateContent:Certificate file content
- pihFileContent: Current PIH as string ### Output
- ValidationResult: Validation Result

CLI Documentation

Commands

Description

Console Application for Zatca E-Invoice .Net SDK

Usage

fatooraNet [command] options

Options

Option	Description
-help	Show help and usage information

Commands

Command	Description
csr	Generates a Certificate Signing Request (CSR) and Private Key
generateHash	Generates E-Invoice Hash
qr	Generates E-Invoice QR
sign	Signing E-Invoice contains the next steps: - Generating Hashing- Generating Signature- Populating Data- Generating and populating QR
validate	There are two types of E-Invoices “Simplified” & “Standard”Validation of “Simplified” E-Invoice contains the next steps:- Validate XSD- Validate EN Schema Tron- Validate KSA Schema Tron- Validate Signature- Validate QR- Validate PIHValidation of “Standard” E-Invoice contains the next steps:- Validate XSD- Validate EN Schema Tron- Validate KSA Schema Tron- Validate PIH
invoiceRequest	Generates E-Invoice API request

Generate CSR

Description

Generates a Certificate Signing Request (CSR) and Private Key

Usage

fatooraNet csr options

Options

Option	Description
-csrConfig	[REQUIRED] CSR configuration file path
-pem	boolean: Whether to generate csr and private key in pem format, if false: the csr file will be generated encoded base64 and private key file will be

Option	Description
	generated without header or footer [default: False]
-generatedCsr	Generated CSR File Path [default: generated-csr-{DateTime}.csr]
-privateKey	Generated Private Key File Path [default: generated-private-key-{DateTime}.key]
-sim	A flag pointing to use the csr and private key on a simulation server. [default: False]
-nonprod	A flag pointing to use the csr and private key on a non production server. [default:False]
-help	Show help and usage information

Generate Hash

Description

Generates E-Invoice Hash

Usage

fatooraNet generateHash options

Options

Option	Description
-invoice	[REQUIRED] E-Invoice file path
-help	Show help and usage information

Generate QR Code

Description

Generates E-Invoice QR

Usage

fatooraNet qr options

Options

Option	Description
-invoice	[REQUIRED] E-Invoice file path
-help	Show help and usage information

Sign E-Invoice

Description

Signing E-Invoice contains the next steps: - Generating Hashing - Generating Signature - Populating Data - Generating and populating QR

Usage

fatooraNet sign options

Options

Option	Description
-invoice	[REQUIRED] E-Invoice file path
-signedInvoice	Signed E-Invoice file path [default: SignedInvoice-{DateTime}.xml]
-certificate	Certificate File Path [default: ..\..\...pem]
-privateKey	Private Key file path [default: ..\..\..-secp256k1-priv-key.pem]
-help	Show help and usage information

Validate E-Invoice

Description

There are two types of E-Invoices “Simplified” & “Standard” - Validation of “Simplified” E-Invoice contains the next steps - Validate XSD - Validate EN Schema Tron - Validate KSA Schema Tron - Validate Signature - Validate QR - Validate PIH - Validation of “Standard” E-Invoice contains the next steps - Validate XSD - Validate EN Schema Tron - Validate KSA Schema Tron - Validate PIH

Usage

fatooraNet validate options

Options

Option	Description
-invoice	[REQUIRED] E-Invoice file path
-certificate	Certificate File Path [default: ..\..\...pem]
-pih	PIH file path [default: ..\..\...txt]
-help	Show help and usage information

Generate Request

Description

Generates E-Invoice API request.

Usage

fatooraNet invoiceRequest options

Options

Option	Description
-invoice	[REQUIRED] E-Invoice file path
-apiRequest	Generated JSON file path [default: Invoice-{DateTime}.json]
-help	Show help and usage information

Installing the .NET SDKs

To ensure you have the required SDK and runtime installed, follow these steps:

1. **Install .NET Core SDK 3.1:**
   - Download the .NET Core SDK from the official .NET website.
   - Follow the installation instructions for your operating system.
2. **Install .NET Framework 4.0:**
   - Download the .NET Framework from the official .NET website.
   - Follow the installation instructions for your operating system.
3. **Check SDK Installation:**
   - Open a terminal or command prompt.
   - Run the following command to verify the installation:

     ```
     dotnet --version
     ```

   - You should see the installed .NET SDK version displayed.
4. **Verify Target Framework:**
   - Navigate to your project's root directory.
   - Open the `.csproj` files of the projects.
   - Ensure that the `<TargetFramework>` or `<TargetFrameworks>` elements match the versions specified above.

Benchmarks

```
BenchmarkDotNet v0.13.7, Windows 10 (10.0.19045.3208/22H2/2022Update)
Intel Core i5-7200U CPU 2.50GHz (Kaby Lake), 1 CPU, 4 logical and 2 physical
cores
.NET SDK 7.0.306
  [Host]               : .NET Core 3.1.32 (CoreCLR 4.700.22.55902, CoreFX
4.700.22.56512), X64 RyuJIT AVX2
  .NET Core 3.1        : .NET Core 3.1.32 (CoreCLR 4.700.22.55902, CoreFX
```

```
4.700.22.56512), X64 RyuJIT AVX2
  .NET Framework 4.6.2 : .NET Framework 4.8.1 (4.8.9166.0), X64 RyuJIT
VectorSize=256
```

Method	Job	Runtime	Mean	Error	StdDev
Sign_EInvoice	.NET Core 3.1	.NET Core 3.1	28,593.3 μs	156.89 μs	146.76 μs
Validate_EInvoice	.NET Core 3.1	.NET Core 3.1	306,694.4 μs	3,786.93 μs	3,162.26 μs
Generate_Hash	.NET Core 3.1	.NET Core 3.1	4,857.4 μs	57.84 μs	54.10 μs
Generate_QR	.NET Core 3.1	.NET Core 3.1	5,210.3 μs	40.30 μs	35.73 μs
Generate_CSR	.NET Core 3.1	.NET Core 3.1	602.4 μs	3.47 μs	3.07 μs
Sign_EInvoice	.NET Framework 4.6.2	.NET Framework 4.6.2	32,019.8 μs	200.81 μs	178.02 μs
Validate_EInvoice	.NET Framework 4.6.2	.NET Framework 4.6.2	262,071.1 μs	3,140.25 μs	2,783.75 μs
Generate_Hash	.NET Framework 4.6.2	.NET Framework 4.6.2	5,564.2 μs	23.23 μs	19.40 μs
Generate_QR	.NET Framework 4.6.2	.NET Framework 4.6.2	5,989.7 μs	39.59 μs	37.03 μs
Generate_CSR	.NET Framework 4.6.2	.NET Framework 4.6.2	721.4 μs	2.55 μs	2.39 μs