

Software Development Kit (SDK)

User Manual

Table of Contents

1	General Information.....	3
1.1	Introduction.....	3
1.1.1	Objectives	3
1.1.2	Purpose and Scope	3
1.1.3	Intended Audience	4
2	SDK Capabilities	4
2.1	General Description and High Level User Journey.....	4
2.2	Prerequisites	6
3	Description of the Compliance and Enablement Toolbox SDK	7
3.1	Access and Download the Compliance and Enablement Toolbox SDK	7
3.1.1	Summary.....	7
	Option 1: Access and Download from the ZATCA Website.....	7
3.1.2.1	Detailed Description	7
3.1.3.1	Process Flow.....	7
	Option 2: Access and Download from Developer Portal.....	10
3.1.2.2	Detailed Description	10
3.1.3.2	Process Flow.....	10
3.1.4	Technical Aspects.....	11
3.2	Using the Command Line Interface (CLI) and Validating XMLs and the QR Code. 14	
3.2.1	Summary.....	14
3.2.2	Detailed Description	14
3.2.3	Process Flow.....	15
3.2.4	Technical Aspect	15
4	Additional Information and Considerations	20
5	FAQs	21
5.1	Business FAQs.....	21
5.2	Technical FAQs	22
6	Appendix	23
6.1	Glossary.....	23
6.2	E-Invoice Specifications.....	23

1 General Information

1.1 Introduction

This User Manual contains the essential information required for the user to make full use of the Compliance and Enablement Toolbox SDK, that will be available through the ZATCA website and through ZATCA's Developer Portal at a later date. This manual includes a description of the SDK functions, capabilities and contingencies, as well as a step-by-step process for the SDK download and use.

The User Manual begins by setting out the objectives, purpose, scope and the intended users of the manual, followed by a general description of the SDK and a high-level user journey. A detailed description of the main functionalities of the SDK is provided with the aim of taking users through all steps required for using the SDK, including how to access it, download it and integrate it with the taxpayer / developer system in order to validate a test XML standard or simplified invoice and validate a test QR code. Finally, this User Manual concludes with additional information and a range of FAQs for addressing any potential queries.

1.1.1 Objectives

This User Manual aims to provide the reader with a clear understanding of the SDK and its functionalities from a user perspective. The intended users of this User Manual are the developers of e-invoice generation solutions, developers representing taxpayers or taxpayers; and this manual enables them to learn how to use the SDK, including how to access it, download it and integrate it with the taxpayer / developer system in order to validate a test XML standard or simplified invoice and validate a test QR code.

1.1.2 Purpose and Scope

This User Manual acts as a guide to developers of e-invoice generation solutions, developers representing taxpayers or taxpayers with regards to the Compliance and Enablement Toolbox SDK. It provides an understanding of the SDK, shows readers how to download it and how to integrate within their systems in order to validate test XML invoices and test QR codes. In addition, it provides relevant information regarding the commands that can be run on the SDK (without prior integration). This manual explains in detail the user journey for the SDK including steps, requirements and processes involved in accessing and downloading the Compliance and Enablement Toolbox SDK. Moreover, it provides some technical aspects, such as the minimum requirements that the developer needs to meet in order to be able to use the SDK, as well as how to install and run it.

The scope of this document covers the following functionalities:

- Access and download the Compliance and Enablement Toolbox SDK
- Using the Command Line Interface
 - Validate a test XML simplified and standard e-invoices, credit and debit notes using the SDK
 - Validate a test QR code using the SDK

1.1.3 Intended Audience

This document is intended to be used by:

- Developers of e-invoice generation solutions
- Developers representing taxpayers
- Taxpayers
- Other stakeholders of relevance

2 SDK Capabilities

2.1 General Description and High Level User Journey

The Compliance and Enablement Toolbox SDK is provided by ZATCA to taxpayers and developers of taxpayer e-invoicing solutions and devices in order to allow them to check the compliance of their e-invoice, credit and debit note XML files, as well as the respective QR codes generated, against the e-invoicing standards, specifications and regulations published by ZATCA, prior to the launch of the Generation and Integration phases. The Compliance and Enablement Toolbox SDK enables solution developers to integrate it into their Software Development Life Cycle (SDLC), with the validations performed being in line with the regulations and requirements published by ZATCA, and that will later be used by ZATCA to verify e-invoices, either during a tax audit or following the implementation of the Integration Phase. In order to achieve this, ZATCA will be providing a Compliance and Enablement Toolbox that comprises of a downloadable SDK that can be integrated with taxpayer e-invoicing solutions to test the compliance of the e-invoice, credit and debit note XML files, as well as the respective QR codes.

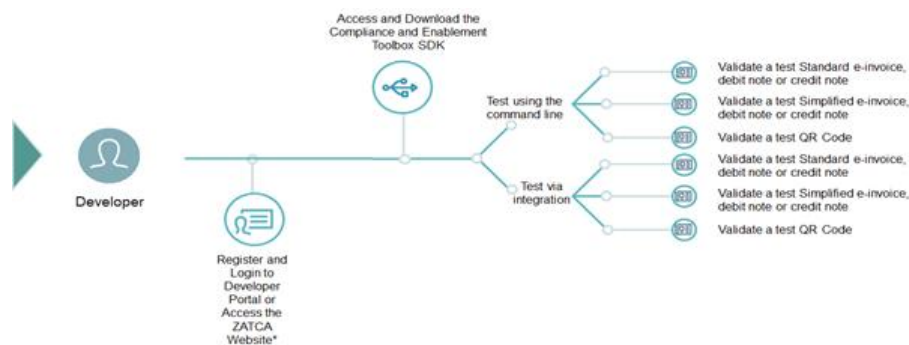
The Compliance and Enablement Toolbox SDK is a standalone application that will run very similar validations as the ZATCA e-invoicing solution, and check the field specifications and business rules expected from a compliant e-invoice, credit or debit note based on ZATCA's published specifications on individual XML files. It also verifies the invoice hash or hashes of a short sequence of such invoices. Since the Compliance and Enablement Toolbox SDK will be standalone, compliance with it does not imply that the XML files have been acknowledged, accepted or approved by ZATCA.

Based on the user journey displayed in the figure below, taxpayers or developers of taxpayer e-invoicing solutions will be able to access the SDK through the ZATCA website or must first register and login to the Developer Portal in order to be able to access and download the SDK and start using it. Once the users have downloaded the Compliance and Enablement Toolbox SDK, they can either test the compliance of their XML files and QR codes via the command line or by integrating the SDK into their solutions.

In case the test is successful, the SDK returns an XML for a compliant e-invoice, credit or debit note along with a QR code string. The taxpayer can request to have the SDK return an XML for a compliant e-invoice, credit or debit note with a dummy cryptographic stamp if they wish (i.e. optional to return the XML signed). In case the test is unsuccessful and errors have been identified, the SDK returns an XML for a non-compliant e-invoice, credit or debit note along with the specific error message(s), where the taxpayer/developer can correct the error(s) identified and re-test the e-invoice, credit or debit note.

A similar process is followed for validating the QR code and the SDK returns successful / error messages depending on the results of the test.

Compliance and Enablement Toolbox SDK



2.1.1 Generation Phase Requirements

The Compliance and Enablement Toolbox SDK for the Generation Phase requirements, allows taxpayers and developers to check the compliance of QR codes generated, against the e-invoicing standards, specifications and regulations published by ZATCA, with regards to the Generation Phase requirements. The users will be able to validate the content of their QR codes generated and in particular compliance with regards to the five fields required for the 4th of December 2021 (Generation Phase) being available and which are:

- Seller's Name
- VAT registration number of the seller
- Timestamp of the electronic invoice or credit/debit note (date and time)
- Electronic invoice or credit/debit note total (with VAT)

- VAT total

The users will be able to check the respective file and its compliance against Generation Phase requirements for the QR code, where the SDK returns successful / error messages depending on the results of the test. The CLI SDK is used to validate a Base 64 encoded string or XML files as described below in section 3.2.4.

2.1.2 Integration Phase Requirements

The Compliance and Enablement Toolbox SDK for the Integration Phase requirements, allows taxpayers and developers of taxpayer e-invoicing solutions and devices to check the compliance of their e-invoice, credit and debit note XML files, as well as the respective QR codes generated, against the e-invoicing standards, specifications and regulations published by ZATCA, with regards to the Integration Phase requirements. The Integration Phase includes in the QR code four additional fields. Utilizing the CLI is the same process as before.

Following the download of the Compliance and Enablement Toolbox SDK, it can then be used to validate the XML files in line with ZATCA's published specifications and standards in the following sequence:

- Validation against the UBL 2.1 XSD schema
- Validation against the EN16931 derived rules (schematron)
- Validation against the KSA specific rules (a separate schematron)
- Validation of cryptographic security features in XML
- Validation of QR code and cryptographic security features in the QR code

It should also be noted that this will allow taxpayers and developers to check the compliance of QR codes generated, against the e-invoicing standards, specifications and regulations published by ZATCA, with regards to the Integration Phase requirements. The users will be able to validate the content of their QR codes generated and in particular compliance with regards to the nine fields required for the Integration Phase, which are:

- Seller's Name
- VAT registration number of the seller
- Timestamp of the electronic invoice or credit/debit note (date and time)
- Electronic invoice or credit/debit note total (with VAT)
- VAT amount
- Hash of XML electronic invoice or credit/debit note
- Cryptographic stamp
- The public key used to generate the Cryptographic stamp
- In the case of simplified tax invoices and their associated notes, the Authority's Portal Cryptographic stamp of the public key of the E-Invoice Solution

2.2 Prerequisites

2.2.1 Targeted users

The Compliance and Enablement Toolbox SDK Page is to be used by:

- Developers of taxpayer e-invoicing solutions and devices
- Taxpayers with e-invoicing solutions and devices
- Taxpayers without their own e-invoicing solutions or devices depending on third party solutions
- Taxpayers with their own e-invoicing solutions and devices but without the function to verify or check the compliance of the provided XML files with the ZATCA standards, specifications and regulations.

2.2.2 Supported platforms:

Both the DLL and JAR are cross-platform and can run on any platform. The Java SDK (JAR) will run on JDK versions ≥ 11 and < 15 , to comply with secp256k1 as per ZATCA security regulations.

3 Description of the Compliance and Enablement Toolbox SDK

3.1 Access and Download the Compliance and Enablement Toolbox SDK

3.1.1 Summary

Taxpayers, developers representing taxpayers or developers representing Solution Providers would wish to obtain the ZATCA E-Invoicing Compliance and Enablement Toolbox SDK so that they can integrate it with their solutions and be able to check the compliance of the XML files for e-invoices, credit or debit notes being generated by their solutions (it should be noted that integration with the testing validation process of the Taxpayer/Developer's own system is optional as the command line could also be used).

Option 1: Access and Download from the ZATCA Website

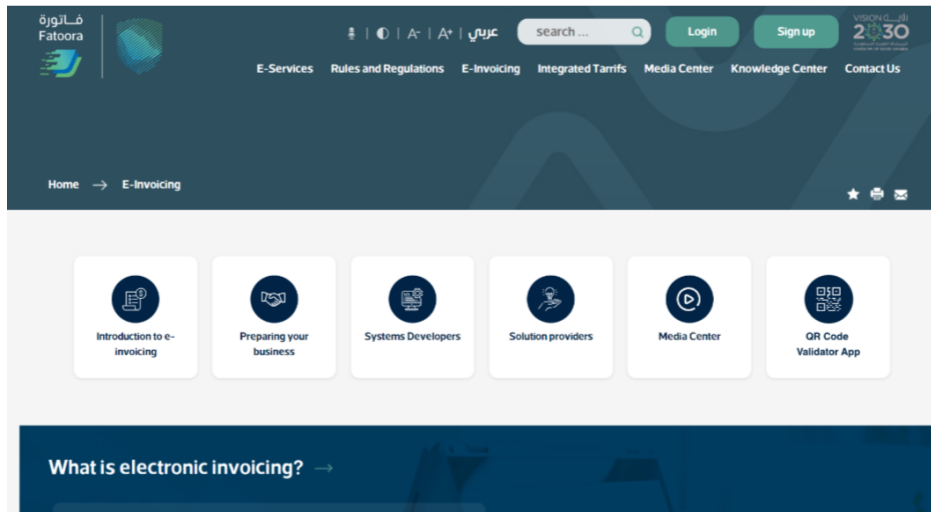
3.1.2.1 Detailed Description

This section details the process of downloading the Compliance and Enablement Toolbox SDK from ZATCA Website to test the compliance of the e-invoice, credit and debit note XML files, as well as the respective QR codes.

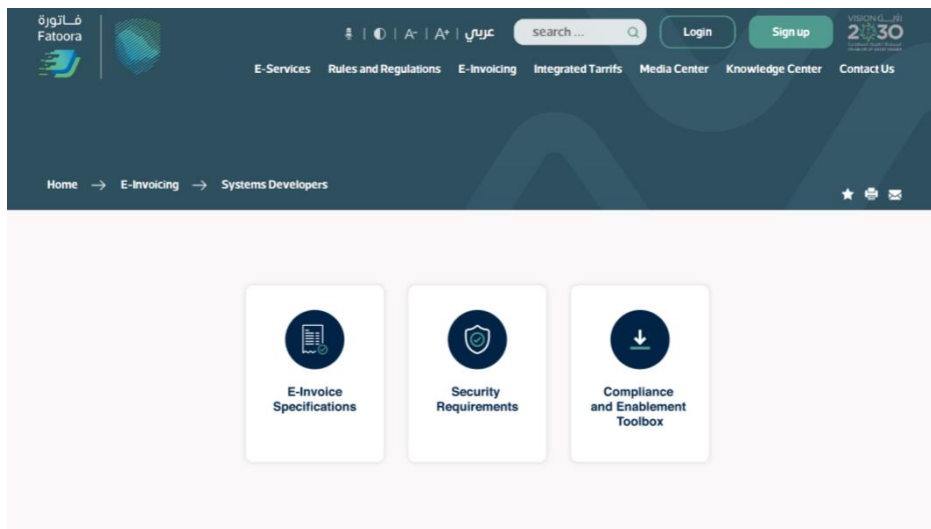
3.1.3.1 Process Flow

The process for accessing and downloading the Compliance and Enablement Toolbox SDK through the ZATCA website is as follows:

- The Taxpayer / Developer can access to ZATCA website
- Taxpayer / Developer has to click on the "Systems Developers" tile



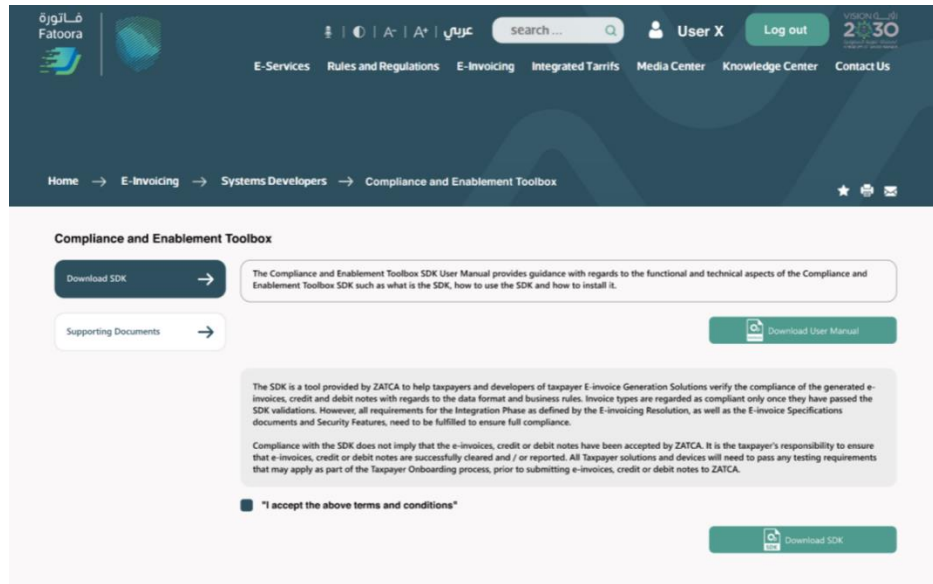
In the Systems Developers page the Taxpayer / Developer can view the following functionalities: E-Invoice Specifications, Security Requirements and Compliance and Enablement Toolbox.



For the "Compliance and Enablement Toolbox":

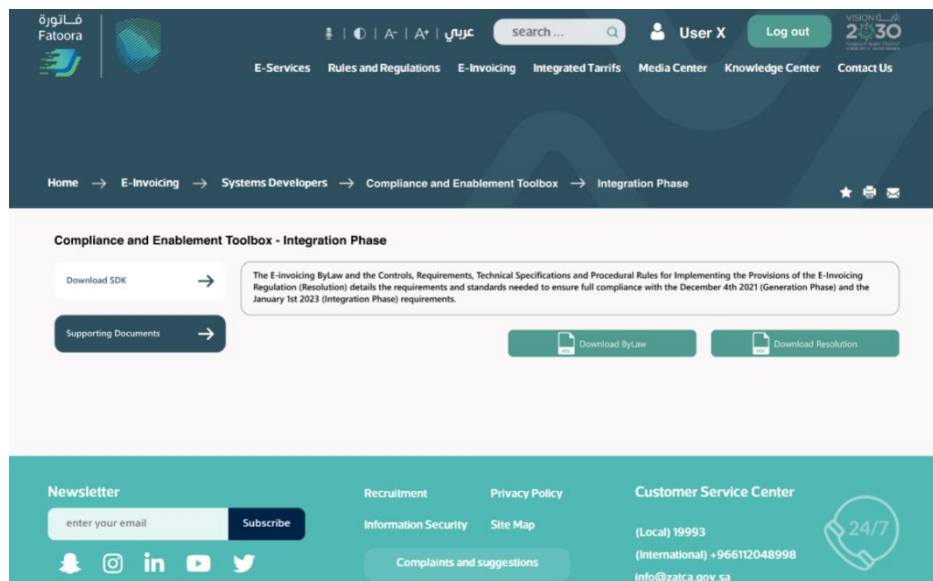
- Taxpayer / Developer needs to click on "I accept the above terms and conditions"

- Afterwards, the "Download SDK" button will be activated and become available for the user to click on
- The Taxpayer / Developer then clicks on "Download SDK" in order to download it
- In addition, the Taxpayer / Developer can click on "Download User Manual" to download and view the User Manual for the SDK



Through the same screen, the Taxpayer / Developer can click on "Supporting Documents" whereby they can click to download the following:

- "Download ByLaw"
- "Download Resolution"



Option 2: Access and Download from Developer Portal

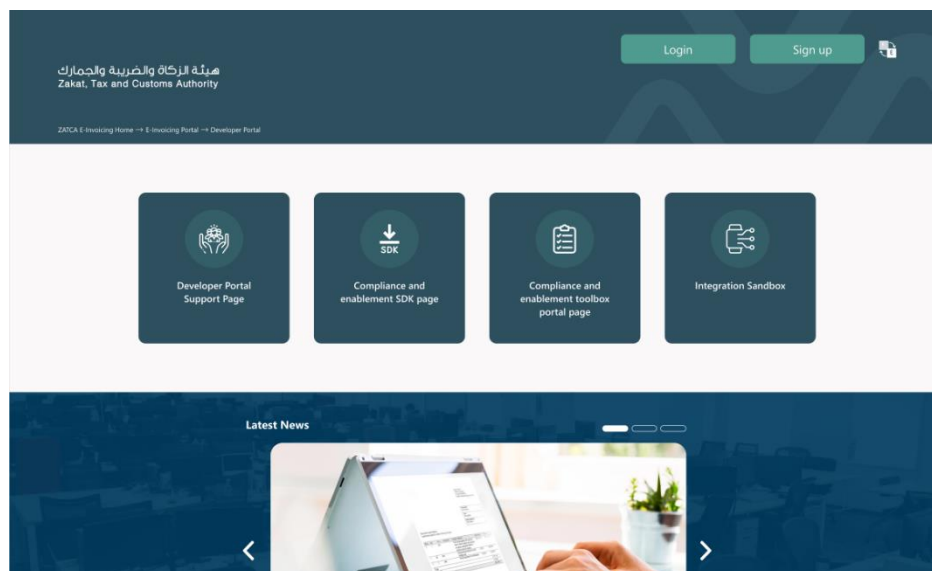
3.1.2.2 Detailed Description

This section details the process of downloading the Compliance and Enablement Toolbox SDK that can be integrated with Taxpayer e-invoicing solutions to test the compliance of the e-invoice, credit and debit note XML files, as well as the respective QR codes. Taxpayers / Developers will be able to access the Compliance and Enablement Toolbox SDK page through the Developer Portal and view the different functionalities being available, such as to download the Compliance and Enablement Toolbox SDK, view the SDK supporting documentations and access the version history of the SDK published periodically (Bitbucket). Taxpayers / Developers will also be able to access and download the Compliance and Enablement Toolbox SDK through the ZATCA website.

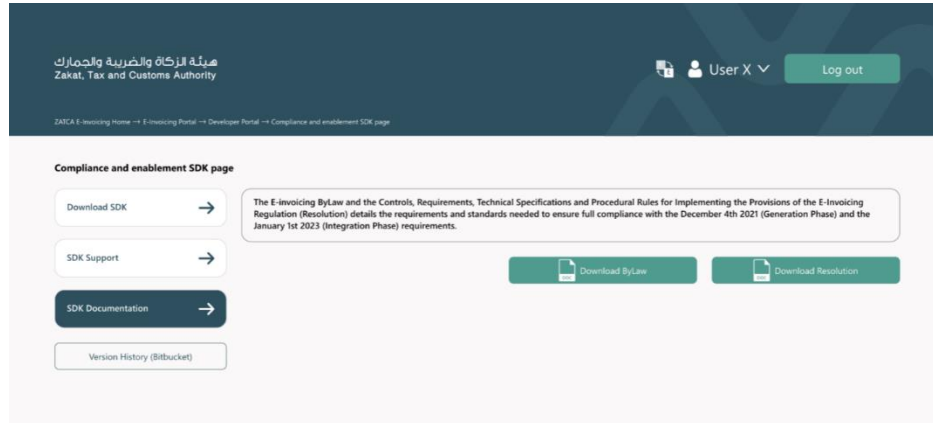
3.1.3.2 Process Flow

The process for accessing and downloading the Compliance and Enablement Toolbox SDK through the Developer Portal is as follows:

- The Taxpayer / Developer should be registered and logged into the Developer Portal successfully
- Taxpayer / Developer should click on "*Compliance and Enablement Toolbox SDK Page*" to view the SDK functionalities.

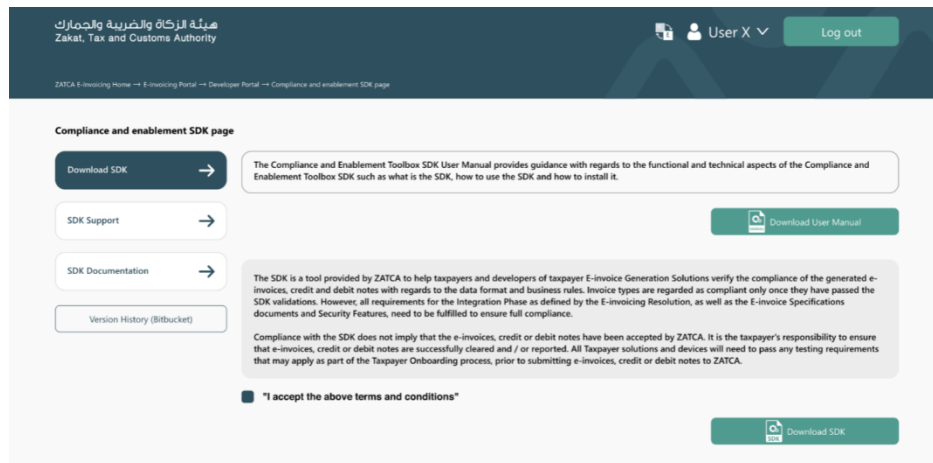


After the user has accessed the *Compliance and Enablement Toolbox SDK Page*, the user can view the SDK documentation, which includes aspects such as how to use the SDK and how it works, as well as the minimum software requirements and the instructions of relevance to the each Operating System / environment. In addition, the user can download the SDK, use the command line interface (CLI) and view the version history via Bitbucket.



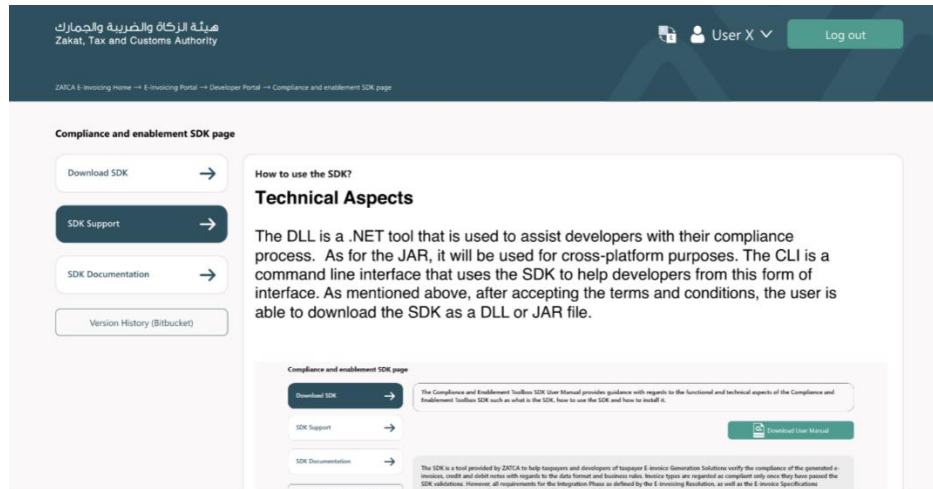
In order to download the SDK to begin testing XML files and the QR code, the process is as follows:

- Taxpayer / Developer clicks on "Download SDK"
- Taxpayer / Developer has to click on "I accept the above terms and conditions"
- Afterwards, the "Download JAR" and "Download DLL" buttons will be activated and become available for the user to click on, depending on whether they would like to download the SDK as JAR or DLL file.



3.1.4 Technical Aspects

The DLL is a .NET tool that is used to assist developers with their compliance process. As for the JAR, it will be used for cross-platform purposes. The CLI is a command line interface that uses the SDK to help developers from this form of interface. As mentioned above, after accepting the terms and conditions, the user is able to download the SDK as a DLL or JAR file.



Minimum requirements for the SDK

The SDK components are lightweight, meaning almost any machine can run them. In essence, if taxpayers / developers machine can run Java Version ≥ 11 and < 15 , then it can run the JAR SDK. The same applies to .NET, if the machine can run the latest .NET Framework requirements then it can run the DLL SDK.

The skills required to run the SDK are two elements. The first is an understanding of the E-Invoicing functionality. From validating an XML in terms of structure to its fields (for example: VAT Number must be 15 characters or Seller Name is required). The other skills required are basic computer skills such as excel, so a relatively competent computer user would be able to run and use the SDK. Previous experience in CLI is helpful, otherwise there is a small learning curve.

However, to use the validation library inside a solution, taxpayers / developers should have intermediate knowledge of Java or .NET programming experience.

How to install/run the SDK

- The prerequisite is using jdk version ≥ 11 and < 15

install for windows

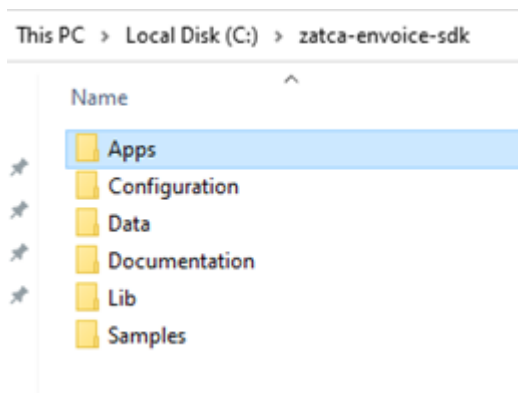
- download the sdk.zip file
- unzip the file
- in the Apps folder run the install.bat
- in the Configuration folder run the install.bat
- then you are ready to run: `fatoora --help` to get a menu of all the commands that can be executed

install for Linux

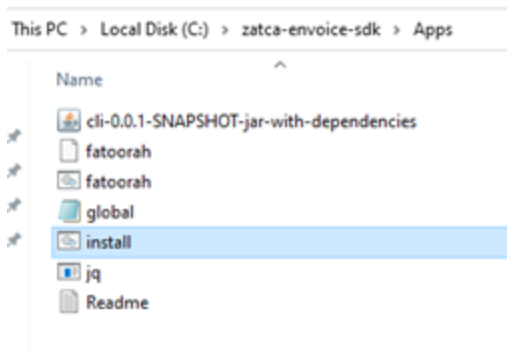
6. install [jq](<https://stedolan.github.io/jq/>)
7. download the sdk.zip file
8. unzip the file
9. in the Apps folder run: source install.sh
10. in the Configuration folder run: source install.sh
11. then you are ready to run: fatoorah --help to get a menu of all the commands that can be executed

Specific instructions for CLI with screenshots

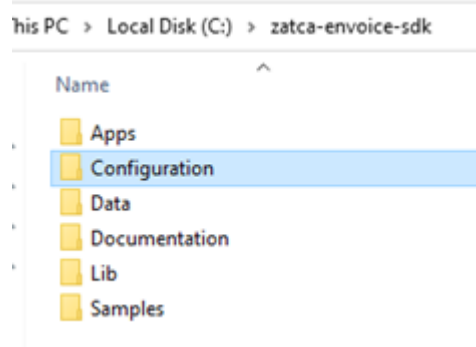
- Unzip the JAR file you have downloaded from the developer portal
- Open the Apps folder



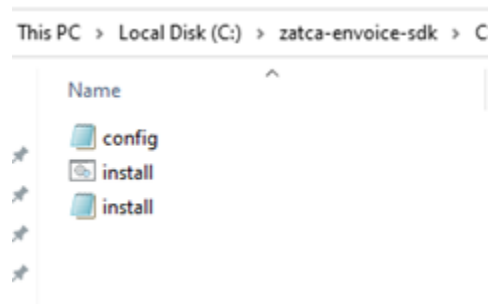
- Right-click Install



- Open the configuration folder



- Right-fatoorah generate -x -f (Invclick Install)



Specific instructions for integrating with Taxpayer systems

- The mentioned installation instruction is mandatory; it also needs to put lib/java/zacta_einvoicing_validation.jar in their system's class path in case of java systems or lib/NET/zacta_einvoicing_validation.dll in their build if using dot Net

3.2 Using the Command Line Interface (CLI) and Validating XMLs and the QR Code

3.2.1 Summary

Taxpayers, developers representing taxpayers or developers representing a Solution Provider can test the compliance of the XML files for standard e-invoices, credit or debit notes generated by their solution in order to check whether they are in line with the ZATCA specifications and regulations, or identify any errors which could be causing non-compliance with the ZATCA specifications and regulations.

3.2.2 Detailed Description

The Command Line Interface (CLI) could be used by taxpayers / developers instead of integrating with the Compliance and Enablement Toolbox SDK. It can allow them to test the compliance of their XML files but does not require integration with the SDK.

3.2.3 Process Flow

The process below indicates how a taxpayer / developer can use the CLI instead of integration with the SDK:

- Taxpayer / developer accesses the "Compliance and Enablement SDK Page" on the Developer Portal
- Taxpayer / developer clicks on "Download SDK"
- Taxpayer / developer accepts the terms and conditions
- Following the above, the taxpayer / developer can download the SDK in JAR or DLL

3.2.4 Technical Aspect

The below are the functionalities of the SDK that are used from the CLI.

- To display (Version)
 - fatoorah -v
- For help
 - fatoorah -h
- CLI Information
 - fatoorah -info
- To validate the XML invoice, credit or debit note for compliance with ZATCA specifications and standards:
 - fatoorah validatexml -f (invoicename.xml)
- Validate Stamp Certificate:
 - fatoorah validatexml -f (Invoicename.xml) -s "new Certificate Path + certificate Name"
- Validate previous hash
 - fatoorah validatexml -f (NewInvoicename.xml) -p (Old Invoicename.xml)
- Validate QR code:
 - fatoorah validateqr -qr
(XXY_QR_CODE_AS_BASE64_ENCODED_STRING_HERE)
- Generate Hash:
 - fatoorah generatehash -f Invoicename.xml
- Generate Signature:
 - fatoorah generate -f (Invoicename.xml) -x
- Generate QR code:
 - fatoorah generate -f (Invoicename.xml) -q

Screenshots

- To display (Version)
 - fatoorah -v

```
C:\Users\MSamy\Desktop\ZATCA\Invoices\Test Data - Invoices\Valid Invoices>fattoorah -v
***** Welcome to ZATCA E-Invoice Java SDK *****
This SDK uses Java to call the SDK (jar) passing it an invoice XML file.
It can take a Standard or Simplified XML, Credit Note, or Debit Note.
It returns if the validation is successful or shows errors where the XML validation fails.
It checks for syntax and content as well.

*****
configuration: Configuration{checkVersion=true, cliVersion='1.0.5', inVerboseMode=false, inHelpMode=false, inInfoMode=false, generateQrCode=false, generateSignature=false, invoicePath='null', stampCertificatePath='null', stampCertificatePassword='123456789', pih='null', qr='null', outputPath='null', commandType=null}
Version : 1.0.5
```

- For help
 - fattoorah -h

```
C:\Users\MSamy\Desktop\ZATCA\Invoices\Test Data - Invoices\Valid Invoices>fattoorah -h
***** Welcome to ZATCA E-Invoice Java SDK *****
This SDK uses Java to call the SDK (jar) passing it an invoice XML file.
It can take a Standard or Simplified XML, Credit Note, or Debit Note.
It returns if the validation is successful or shows errors where the XML validation fails.
It checks for syntax and content as well.

*****
configuration: Configuration{checkVersion=false, cliVersion='1.0.5', inVerboseMode=false, inHelpMode=true, inInfoMode=false, generateQrCode=false, generateSignature=false, invoicePath='null', stampCertificatePath='null', stampCertificatePassword='123456789', pih='null', qr='null', outputPath='null', commandType=null}
Usage : fattoorah [ command ] [ options ]

fattoorah validatexml [arguments/options]
  -f XML invoice file path
  -s stamp certificate path, if this argument not specified it will took the default path in SDK folder under Data\Certificates
  -p Previous invoice hash code to be validated
  >> path for output file, if the not specified XML will be shown on the screen
fattoorah validateqr [arguments/options]
  -qr QR code text based in Base64 encoding
fattoorah generatehash [arguments/options]
  -f Path to XML file of the invoice
fattoorah generate [arguments/options]
  -f XML invoice file path
  -x To generate Signature and inject it inside the passed invoice
  -q To generate qr code and inject it inside the passed invoice

C:\Users\MSamy\Desktop\ZATCA\Invoices\Test Data - Invoices\Valid Invoices>
```

- CLI Information
 - fattoorah -info

```
C:\Users\MSamy\Desktop\ZATCA\Invoices\Test Data - Invoices\Valid Invoices>fattoorah -info
***** Welcome to ZATCA E-Invoice Java SDK *****
This SDK uses Java to call the SDK (jar) passing it an invoice XML file.
It can take a Standard or Simplified XML, Credit Note, or Debit Note.
It returns if the validation is successful or shows errors where the XML validation fails.
It checks for syntax and content as well.

*****
configuration: Configuration{checkVersion=false, cliVersion='1.0.5', inVerboseMode=false, inHelpMode=false, inInfoMode=true, generateQrCode=false, generateSignature=false, invoicePath='null', stampCertificatePath='null', stampCertificatePassword='123456789', pih='null', qr='null', outputPath='null', commandType=null}
This tool allows users to input an XML invoice, credit or debit note and check for compliance with ZATCA specifications and standards
And the current version is: 1.0.5
```

To Validate Standard e-invoice/Debit Note/Credit Note and Simplified e-invoice/Debit Note/Credit Note:

Please note: Validating Simplified/CN/DN and Standard/CN/DN are an identical command. It is the XML file that tells the SDK what kind of file that it is.

There is a section in the XML (a tag) called `invoicetypecode` and it can be set to the following:

- 01 : standard - 02 : simplified
- 383 : debit
- 381 : credit
- 388: tax

fatoorah validatexml -f (invoicename.xml)

```
C:\Users\MSamy\Desktop\New folder (2)>fatoorah validatexml -f "Simplified_Invoice_valid (3).xml"
***** Welcome to ZATCA E-Invoice Java SDK *****
This SDK uses Java to call the SDK (jar) passing it an Invoice XML file.
It can take a Standard or Simplified XML, Credit Note, or Debit Note.
It returns if the validation is successful or shows errors where the XML validation fails.
It checks for syntax and content as well.

*****
configuration: Configuration{checkVersion=false, cliVersion='1.0.5', inVerboseMode=false, inHelpMode=false, inInfoMode=false, generateQrCode=false, generateSignature=false, invocPath='Simplified_Invoice_valid (3).xml', stampCertificatePath='null', stampCertificatePassword='123456789', pin='null', qr='null', outputPath='null', commandType=VALIDATE_XML}
Validate XSD for Invoice : Simplified_Invoice_valid (3).xml
xsd validation done with result : Result{valid=true, error=null, validQrCode=false, validSignature=false}
Validate Schematron using C:\Users\MSamy\Desktop\ZATCA\CLI\Versions\zatca-einvoice-sdk-dec405102021\zatca-einvoice-sdk-dec405102021\zatca-einvoice-sdk-dec405102021\Rules\schematrons\CEN-EN16931-UBL.xsl for Invoice : Simplified_Invoice_valid (3).xml
on validation done with result : Result{valid=true, error=null, validQrCode=false, validSignature=false}
Validate Schematron using C:\Users\MSamy\Desktop\ZATCA\CLI\Versions\zatca-einvoice-sdk-dec405102021\zatca-einvoice-sdk-dec405102021\Rules\schematrons\20210819_ZATCA_E-Invoice_Validation_Rules.xsl for Invoice : Simplified_Invoice_valid (3).xml
xsd validation done with result : Result{valid=true, error=null, validQrCode=false, validSignature=false}
Qr Code validation done with result : Result{valid=true, error=null, validQrCode=true, validSignature=false}
certPassword : 123456789
hashedCert: defec0b9e9d986bb614772e1d5375464ed1a793ded751feb3414980f7c
certPassword : 123456789
certPassword : 123456789
Signature validation done with result : Result{valid=false, error=[invoiceSignedDataDigestValue=wrong invoiceSignedData DigestValue , signatureValue=wrong signature Value ], validQrCode=false, validSignature=false}
```

fatoorah validatexml -f (invoicename.xml)

- Success Example

```
librahing@librahim-Lenovo-Z50-70:~$ fatoorah validatexml -f ~/Downloads/Simplified_Invoice_valid.xml
***** Welcome to ZATCA E-Invoice Java SDK *****
This SDK uses Java to call the SDK (jar) passing it an Invoice XML file.
It can take a Standard or Simplified XML, Credit Note, or Debit Note.
It returns if the validation is successful or shows errors where the XML validation fails.
It checks for syntax and content as well.

*****
configuration: Configuration{checkVersion=false, cliVersion='1.0.5', inVerboseMode=false, inHelpMode=false, inInfoMode=false, generateQrCode=false, generateSignature=false, invocPath='/home/librahim/Downloads/Simplified_Invoice_valid.xml', stampCertificatePath='null', stampCertificatePassword='123456789', pin='null', qr='null', outputPath='null', commandType=VALIDATE_XML}
Validate XSD for Invoice : /home/librahim/Downloads/Simplified_Invoice_valid.xml
xsd validation done with result : Result{valid=true, error=null, validQrCode=false, validSignature=false}
Validate Schematron using /home/librahim/agilez/dtne-se-zatca-einvoice-sdk-resources/Data/Rules/schematrons/CEN-EN16931-UBL.xsl for Invoice : /home/librahim/Downloads/Simplified_Invoice_valid.xml
on validation done with result : Result{valid=true, error=null, validQrCode=false, validSignature=false}
Validate Schematron using /home/librahim/agilez/dtne-se-zatca-einvoice-sdk-resources/Data/Rules/schematrons/20210819_ZATCA_E-Invoice_Validation_Rules.xsl for Invoice : /home/librahim/Downloads/Simplified_Invoice_valid.xml
xsd validation done with result : Result{valid=true, error=null, validQrCode=false, validSignature=false}
Qr Code validation done with result : Result{valid=true, error=null, validQrCode=true, validSignature=false}
certPassword : 123456789
hashedCert: defec0b9e9d986bb614772e1d5375464ed1a793ded751feb3414980f7c
certPassword : 123456789
certPassword : 123456789
Signature validation done with result : Result{valid=true, error=null, validQrCode=false, validSignature=true}
PIN validation done with result : Result{valid=true, error=null, validQrCode=false, validSignature=false}
librahing@librahim-Lenovo-Z50-70:~$
```

- Fail Example 1

```
librahing@librahim-Lenovo-Z50-70:~$
librahing@librahim-Lenovo-Z50-70:~$
librahing@librahim-Lenovo-Z50-70:~$ fatoorah validatexml -f ~/Downloads/Simplified_Invoice_valid.xml
***** Welcome to ZATCA E-Invoice Java SDK *****
This SDK uses Java to call the SDK (jar) passing it an Invoice XML file.
It can take a Standard or Simplified XML, Credit Note, or Debit Note.
It returns if the validation is successful or shows errors where the XML validation fails.
It checks for syntax and content as well.

*****
configuration: Configuration{checkVersion=false, cliVersion='1.0.5', inVerboseMode=false, inHelpMode=false, inInfoMode=false, generateQrCode=false, generateSignature=false, invocPath='/home/librahim/Downloads/Simplified_Invoice_valid.xml', stampCertificatePath='null', stampCertificatePassword='123456789', pin='null', qr='null', outputPath='null', commandType=VALIDATE_XML}
Validate XSD for Invoice : /home/librahim/Downloads/Simplified_Invoice_valid.xml
xsd validation done with result : Result{valid=true, error=null, validQrCode=false, validSignature=false}
Validate Schematron using /home/librahim/agilez/dtne-se-zatca-einvoice-sdk-resources/Data/Rules/schematrons/CEN-EN16931-UBL.xsl for Invoice : /home/librahim/Downloads/Simplified_Invoice_valid.xml
on validation done with result : Result{valid=true, error=null, validQrCode=false, validSignature=false}
Validate Schematron using /home/librahim/agilez/dtne-se-zatca-einvoice-sdk-resources/Data/Rules/schematrons/20210819_ZATCA_E-Invoice_Validation_Rules.xsl for Invoice : /home/librahim/Downloads/Simplified_Invoice_valid.xml
xsd validation done with result : Result{valid=true, error=null, validQrCode=false, validSignature=false}
Qr Code validation done with result : Result{valid=false, error=[sellerName=seller name does not match with qr code seller name, hashedXml=hashedXml does not match with qr code hashedXml], validQrCode=false, validSignature=false}
```

- Fail Example 2

```
librahing@librahim-Lenovo-Z50-70:~$ fatoorah validatexml -f ~/Downloads/KSA-132-amended-GE-1534.xml
***** Welcome to ZATCA E-Invoice Java SDK *****
This SDK uses Java to call the SDK (jar) passing it an Invoice XML file.
It can take a Standard or Simplified XML, Credit Note, or Debit Note.
It returns if the validation is successful or shows errors where the XML validation fails.
It checks for syntax and content as well.

*****
configuration: Configuration{checkVersion=false, cliVersion='1.0.5', inVerboseMode=false, inHelpMode=false, inInfoMode=false, generateQrCode=false, generateSignature=false, invocPath='/home/librahim/Downloads/KSA-132-amended-GE-1534.xml', stampCertificatePath='null', stampCertificatePassword='123456789', pin='null', qr='null', outputPath='null', commandType=VALIDATE_XML}
Validate XSD for Invoice : /home/librahim/Downloads/KSA-132-amended-GE-1534.xml
xsd validation done with result : Result{valid=true, error=null, validQrCode=false, validSignature=false}
Validate Schematron using /home/librahim/agilez/dtne-se-zatca-einvoice-sdk-resources/Data/Rules/schematrons/CEN-EN16931-UBL.xsl for Invoice : /home/librahim/Downloads/KSA-132-amended-GE-1534.xml
on validation done with result : Result{valid=false, error=[BR-CO-16-[BR-CO-16]-Amount due for payment (BT-115) = Invoice total amount with VAT (BT-112) - Paid amount (BT-113) - Rounding amount (BT-114), BR-CO-15-[BR-CO-15]-Invoice total amount with VAT (BT-112) = Invoice total amount without VAT (BT-109) + Invoice total VAT amount (BT-110), BR-13-[BR-13]-An Invoice shall have the Invoice total amount without VAT (BT-109), BR-14-[BR-14]-An Invoice shall have the Invoice total amount with VAT (BT-112), BR-CO-10-[BR-CO-10]-Sum of Invoice line net amount (BT-106) = Invoice line net amount (BT-131), BR-CO-13-[BR-CO-13]-Invoice total amount without VAT (BT-109) = Σ Invoice line net amount (BT-131) - Sum of allowances on document level (BT-107) + Sum of charges on document level (BT-108), BR-CO-11-[BR-CO-11]-Sum of allowances on document level (BT-107) = Σ Document level allowance amount (BT-92), validQrCode=false, validSignature=false}
Validate Schematron using /home/librahim/agilez/dtne-se-zatca-einvoice-sdk-resources/Data/Rules/schematrons/20210819_ZATCA_E-Invoice_Validation_Rules.xsl for Invoice : /home/librahim/Downloads/KSA-132-amended-GE-1534.xml
xsd validation done with result : Result{valid=false, error=[BR-KSA-EN16931-12-[BR-KSA-EN16931-12]-Base quantity (BT-149) must be a positive number above zero, BR-KSA-EN16931-11-[BR-KSA-EN16931-11]-Invoice line net amount (BT-131) must equal (Invoiced quantity (BT-130) * (Item net price (BT-146) / Item price base quantity (BT-149))), Sum of Invoice line net amount (BT-136), BR-KSA-F-04-[BR-KSA-F-04]-All the document amounts and quantities must be positive, BR-KSA-EN16931-03-[BR-KSA-EN16931-03]-Allowance amount (BT-92, BT-130) must equal base amount (BT-93), BT-137 = percentage (BT-94, BT-138) / 100 if base amount and percentage exists, validQrCode=false, validSignature=false}
librahing@librahim-Lenovo-Z50-70:~$
```

Stamp Certificate, Previous Hash

- Validate Stamp Certificate:
 - `fatoorah validatexml -f (Invoicename.xml) -s "new Certificate Path + certificate Name"`

Note:

“The default certificate will be used as an example in folder data/certificates, however if the user requires to sign the invoice with different certificate he/she can pass it threw the command line interface to override the current certificate.”

```
C:\Users\MSam\Desktop\ZATCA\Invoices\Test Data - Invoices\Valid Invoices>fatoorah validatexml -f Debit_Note.xml "C:\Users\ARadwan\Desktop\Create Valid Invoices Workspace - Copy\Create Valid Invoices Workspace\garcertificate.p12"
***** welcome to ZATCA E-Invoice Java SDK *****
This SDK uses Java to call the SDK (jar) passing it an invoice XML file.
It can take a standard or Simplified XML, Credit Note, or Debit Note.
It returns if the validation is successful or shows errors where the XML validation fails.
It checks for syntax and content as well.

*****
configuration: Configuration(checkVersion=false, cliVersion='1.0.5', inVerboseMode=false, inHelpMode=false, inInfoMode=false, generateQrCode=false, generateSignature=false, invoicePath='Debit_Note.xml', stampCertificatePath='null', stampCertificatePassword='123456789', pin='null', qr='null', outputPath='null', commandType=VALIDATE_XML)
Validate XSD for invoice : Debit_Note.xml
xsd validation done with result : Result(valid=true, error=null, validQrCode=false, validSignature=false)
Validate Schematron using C:\Users\MSam\Desktop\ZATCA\CLI\Versions\zatca-environment-sdk-dec405102021\zatca-environment-sdk-dec405102021\zatca-environment-sdk-dec405102021\Rules\schematrons\CEN-EN16931-UBL.xsl for invoice : Debit_Note.xml
en validation done with result : Result(valid=true, error=null, validQrCode=false, validSignature=false)
Validate Schematron using C:\Users\MSam\Desktop\ZATCA\CLI\Versions\zatca-environment-sdk-dec405102021\zatca-environment-sdk-dec405102021\zatca-environment-sdk-dec405102021\Rules\schematrons\20210819_ZATCA_E-invoice_Validation_Rules.xsl for invoice : Debit_Note.xml
ksa validation done with result : Result(valid=true, error=null, validQrCode=false, validSignature=false)
Qr Code validation done with result : Result(valid=true, error={}, validQrCode=true, validSignature=false)
certPassword : 123456789
certPassword : 123456789
SigningTime: 2021-10-20T11:59:53Z
certPassword : 123456789
certPassword : 123456789
buffer length: 141
#### Standard Invoice content ####
```

```
##### end Standard Invoice content #####
Signature validation done with result : Result(valid=true, error=null, validQrCode=false, validSignature=false)
PIN validation done with result : Result(valid=true, error=null, validQrCode=false, validSignature=false)
```

- Validate previous hash
 - `fatoorah validatexml -f (Invoicename.xml) -p 0zaqGXAlpK73XM9FzvQoQbmE6vSroKSeBWSw8guIKIk`

```
C:\Users\MSam\Desktop\ZATCA\Invoices\Test Data - Invoices\Valid Invoices>fatoorah validatexml -f _copy_debit_Note.xml -p 0zaqGXAlpK73XM9FzvQoQbmE6vSroKSeBWSw8guIKIk
***** welcome to ZATCA E-Invoice Java SDK *****
This SDK uses Java to call the SDK (jar) passing it an invoice XML file.
It can take a standard or Simplified XML, Credit Note, or Debit Note.
It returns if the validation is successful or shows errors where the XML validation fails.
It checks for syntax and content as well.

*****
configuration: Configuration(checkVersion=false, cliVersion='1.0.5', inVerboseMode=false, inHelpMode=false, inInfoMode=false, generateQrCode=false, generateSignature=false, invoicePath='_copy_debit_Note.xml', stampCertificatePath='null', stampCertificatePassword='123456789', pin='0zaqGXAlpK73XM9FzvQoQbmE6vSroKSeBWSw8guIKIk', qr='null', outputPath='null', commandType=VALIDATE_XML)
Validate XSD for invoice : _copy_debit_Note.xml
xsd validation done with result : Result(valid=true, error=null, validQrCode=false, validSignature=false)
Validate Schematron using C:\Users\MSam\Desktop\ZATCA\CLI\Versions\zatca-environment-sdk-dec405102021\zatca-environment-sdk-dec405102021\zatca-environment-sdk-dec405102021\Rules\schematrons\CEN-EN16931-UBL.xsl for invoice : _copy_debit_Note.xml
en validation done with result : Result(valid=true, error=null, validQrCode=false, validSignature=false)
Validate Schematron using C:\Users\MSam\Desktop\ZATCA\CLI\Versions\zatca-environment-sdk-dec405102021\zatca-environment-sdk-dec405102021\zatca-environment-sdk-dec405102021\Rules\schematrons\20210819_ZATCA_E-invoice_Validation_Rules.xsl for invoice : _copy_debit_Note.xml
ksa validation done with result : Result(valid=true, error=null, validQrCode=false, validSignature=false)
Qr Code validation done with result : Result(valid=true, error={}, validQrCode=true, validSignature=false)
certPassword : 123456789
certPassword : 123456789
SigningTime: 2021-10-20T12:22:32Z
certPassword : 123456789
certPassword : 123456789
buffer length: 141
#### Standard Invoice content ####
```

```
##### end Standard Invoice content #####
Signature validation done with result : Result(valid=true, error=null, validQrCode=false, validSignature=false)
PIN validation done with result : Result(valid=false, error=(KSA-13-PIN is invalid), validQrCode=false, validSignature=false)
```

In order to validate the QR Code

- `fatoorah validateqr -qr XXY_QR_CODE_AS_BASE64_ENCODED_STRING_HERE`

- You must give the CLI an encoded QR code. This means it is computer language that is not readable by the human eye.
- It decodes the QR code and returns:
 - Seller Name
 - Registration Number
 - Date and time
 - Tax Total
 - VAT total
- It returns an error if the validation fails

```

librahinglibrahin@lenovo-220-701:~$ fatoorah validateqr -qr AQSTYXhacBID3NwaXRhbi1PhZEWtTjYmZkZNTAwMDAzAqYwDILTAhLTA1VDA5OjMyOjQwMgQGRjUwLjNwAAAAA
***** Welcome to ZATCA E-Invoice Java SDK *****
This SDK uses Java to call the SDK (jar) passing it an invoice XML file.
It can take a Standard or Simplified XML, Credit Note, or Debit Note.
It returns if the validation is successful or shows errors where the XML validation fails.
It checks for syntax and content as well.

configuration: Configuration{checkVersion=false, cliVersion='1.0.5', inVerboseMode=false, inHelpMode=false, inInfoMode=false, generateQRCode=false, generateSignature=false, invoicePath='null', stampCertificatePath='null', stampCertificatePassword='123456789', piih='null', qr='AQSTYXhacBID3NwaXRhbi1PhZEWtTjYmZkZNTAwMDAzAqYwDILTAhLTA1VDA5OjMyOjQwMgQGRjUwLjNwAAAAA', outputPath='null', commandType=VALIDATE_QR}
sellerNameFromQRCode: Salah Hospital
vatRegistrationNumberFromQRCode: 31012239350003
lineStampFromQRCode: 2021-01-05T09:32:40Z
taxTotalFromQRCode: 250.00
vatTotalFromQRCode: 0.00
librahinglibrahin@lenovo-220-701:~$
    
```

Generate Hash and Generate Signature

- fatoorah generatehash -f Invoicename.xml
- Hash is the final part of the output from the CLI

```

C:\Users\MEzz\Desktop\zatca\invoice>fatoorah generatehash -f Debit_Note.xml
***** Welcome to ZATCA E-Invoice Java SDK *****
This SDK uses Java to call the SDK (jar) passing it an invoice XML file.
It can take a Standard or Simplified XML, Credit Note, or Debit Note.
It returns if the validation is successful or shows errors where the XML validation fails.
It checks for syntax and content as well.

configuration: Configuration{checkVersion=false, cliVersion='0.0.1-SNAPSHOT', inVerboseMode=false, inHelpMode=false, inInfoMode=false, generateQRCode=false, generateSignature=false, invoicePath='Debit_Note.xml', stampCertificatePath='null', stampCertificatePassword='123456789', piih='null', qr='null', outputPath='null', commandType=GENERATE_HASH}
iWo78u1QqWtkFiaJv+bi2J74NWKQgobrhfzBa+ndRXk=
    
```

- Generate Signature:
 - fatoorah generate -x -f (Invoicename.xml)

```

C:\Users\MSamy\Desktop\ZATCA\invoices\Test Data - Invoices\Valid Invoices>fatoorah generate -x -f _copy_debit_Note.xml
***** Welcome to ZATCA E-Invoice Java SDK *****
This SDK uses Java to call the SDK (jar) passing it an invoice XML file.
It can take a Standard or Simplified XML, Credit Note, or Debit Note.
It returns if the validation is successful or shows errors where the XML validation fails.
It checks for syntax and content as well.

configuration: Configuration{checkVersion=false, cliVersion='1.0.5', inVerboseMode=false, inHelpMode=false, inInfoMode=false, generateQRCode=false, generateSignature=true, invoicePath='_copy_debit_Note.xml', stampCertificatePath='null', stampCertificatePassword='123456789', piih='null', qr='null', outputPath='null', commandType=GENERATE}
certPassword : 123456789
certPassword : 123456789
certPassword : 123456789
SigningTime:2021-10-20T11:44:47Z
certPassword : 123456789
certPassword : 123456789
Generation done successfully.
    
```

Generate QR Code

- fatoorah generate -q -f (Invoicename.xml)

```
C:\Users\MEzz\Desktop\zatca\Invoice>fatoorah generate -q -f Debit_Note.xml
***** Welcome to ZATCA E-Invoice Java SDK *****
This SDK uses Java to call the SDK (jar) passing it an invoice XML file.
It can take a Standard or Simplified XML, Credit Note, or Debit Note.
It returns if the validation is successful or shows errors where the XML validation fails.
It checks for syntax and content as well.

*****
configuration: Configuration{checkVersion=false, cliVersion='0.0.1-SNAPSHOT', inVerboseMode=false, invoicePath='Debit_Note.xml', stampCertificatePath='null', stampCertificatePath='null'}
buffer length: 367
encoding the signature of length: 96
inserting R with tag: 8
inserting R with tag: 9
Generation done successfully.
```

4 Additional Information and Considerations

- The Taxation portal log in credentials cannot be used to access the Developer Portal and download the SDK.
- The Compliance and Enablement Toolbox will allow XML files or PDF/A3 files to be validated using the SDK, where in the case of PDF/A3 files, the embedded XML file will be extracted and validated. PDF/A3 files without an embedded XML, cannot be validated through the SDK.
- The Compliance and Enablement Toolbox SDK will operate on a standalone and offline basis. XML files will be validated and not be transmitted or stored on ZATCA databases.
- Since the Compliance and Enablement Toolbox SDK will operate on a standalone basis it will not simulate the invoice clearance and/or reporting processes, neither will it integrate with any other systems.
- XML files found to be compliant will not be deemed as accepted, approved or acknowledged by ZATCA.
- Non-compliance with the Compliance and Enablement Toolbox SDK would indicate non-compliance during the Integration Sandbox testing therefore, taxpayers and developers will need to ensure adequate testing of the same non-compliant e-invoices takes place across both the Compliance and Enablement Toolbox SDK and the Integration Sandbox, in order to achieve compliance.
- It is not mandatory for Taxpayers to use the Compliance and Enablement Toolbox SDK and therefore not a pre-requisite to access the Integration Sandbox; however, all taxpayer solutions and devices will need to pass the testing requirements as part of the Taxpayer Onboarding process prior to submitting e-invoices, credit or debit notes to ZATCA.

5 FAQs

5.1 Business FAQs

- **Where can I find the Compliance and Enablement Toolbox SDK?**
 - The Compliance and Enablement Toolbox SDK will be accessible through the ZATCA website or the Developer Portal. To download the Compliance and Enablement Toolbox SDK, users should access it through the ZATCA website or should be registered and logged into the Developer Portal.
- **Do I have to use the Compliance and Enablement Toolbox SDK?**
 - It is not mandatory for Taxpayers to use the Compliance and Enablement Toolbox SDK and therefore not a pre-requisite to access the Integration Sandbox; however, all taxpayer solutions and devices will need to pass the testing requirements as part of Taxpayer Onboarding prior to submitting e-invoices, credit or debit notes to ZATCA. It should be noted that ZATCA might require in the future for a proof of a compliant xml file.
- **Will my compliant XML files be accepted by ZATCA?**
 - XML files found to be compliant will not be deemed as accepted, approved or acknowledged by ZATCA. The purpose of the Compliance and Enablement Toolbox SDK is for taxpayers / developers to ensure their XML files are compliant with the ZATCA specifications and any errors identified are corrected before the enforcement of the e-invoicing regulations.
- **What are the QR fields that will be validated in the Generation phase and are required for the 4th of December 2021?**
 - The users will be able to validate the following fields:
 1. Seller's Name
 2. VAT registration number of the seller
 3. Timestamp of the electronic invoice or credit/debit note (date and time)
 4. Electronic invoice or credit/debit note total (with VAT)
 5. VAT total
- **What are the QR fields that will be validated in the Integration phase?**
 - The users will be able to validate the following fields:
 1. Seller's Name
 2. VAT registration number of the seller
 3. Timestamp of the electronic invoice or credit/debit note (date and time)
 4. Electronic invoice or credit/debit note total (with VAT)
 5. VAT amount
 6. Hash of XML electronic invoice or credit/debit note
 7. Cryptographic stamp
 8. The public key used to generate the Cryptographic stamp

9. In the case of simplified tax invoices and their associated notes, the Authority's Portal Cryptographic stamp of the public key of the E-Invoice Solution

5.2 Technical FAQs

- **What is an SDK?**
 - A Software Development Kit (SDK) is a tool used to assist developers to accomplish certain tasks. In our context, and SDK helps developer validate the compliance of their XML (E-Invoice).
- **What is an XML?**
 - An XML file is a good way to organize data with structure. It uses tags and values. In our context, we store information relating to invoices.
 - Sample: `<cbc:IssueDate>2021-04-25</cbc:IssueDate>`
- **What is Command Line Interface (CLI)?**
 - A CLI is a way to access and utilize a software application using commands. It is text-based
 - Sample: `fatoorah validatexml -f (invoicename.xml)`
In this example, we are naming an application called "fatoorah", telling it we want to use `validatexml` feature with a `-f` command. The second part that we add is: `(invoicename.xml)` which is the path and filename of the XML to be validated.
- **What is a Java and JAR?**
 - Java is a programming language that runs on different operating systems (OS), such as Windows and Linux. A JAR is a java-based application. In our previous example (3b), the Jar application (or file) is called "fatoorah". You run "fatoorah" using the CLI as described previously.
- **What is DLL ?**
 - It is the JAR equivalent in .NET; .NET is a Microsoft technology that allows software developers to make rich applications. The DLL can be used inside an environment directly from the code. The only thing that the developer has to do is import the DLL into their solution. Usually developers use Visual Studio, which is an application environment to write the code. In addition, The DLL can also be run from command line (CLI) like its a JAR equivalent.
- **Can I validate the Arabic language fields in a QR within the CLI?**
 - The CLI does not support Arabic characters display, however the user would be able to validate the data by directing the output of the command to a file.
- **What JAVA version should I install before using the SDK?**
 - The prerequisite is using the Java SDK (JAR) versions `>=11` and `<15`.
- **What should the user do when faced with a JAVA error?**
 - When faced with a JAVA error, the user needs to install JAVA (versions `>=11` and `<15`) before running and using the SDK

6 Appendix

6.1 Glossary

ZATCA	ZAKAT, Tax and Customs Authority
XML	Extensible Markup Language
SDK	Software Development Kit
QR Code	Quick Response Code
PKI	Public Key Infrastructure
SDLC	Software Development Life Cycle
CLI	Command Line Interface
JAR	JAVA Archive
DLL	Dynamic-Link Library
Integration Sandbox	The Integration Sandbox should enable solution developers to simulate the integration calls/requests

6.2 E-Invoice Specifications

Please refer to the E-Invoice specifications link below, for accessing the Electronic Invoice data dictionary in addition to the electronic invoice xml implementation standard

[E-Invoice specifications \(zatca.gov.sa\)](https://zatca.gov.sa)