# 3. Security Requirements

ZATCA uses Basic Authentication for its E-invoicing security solution.

| | Basic Authentication |
|---|---|
| Description | **The solution will include a Basic Authentication header with the CSID as the Username and a Secret Value as the Password. Secret value will be issue with the CSID. An additional accept-version: v2 header must be added to V2 API calls.** |
| Onboarding | CSID and Secret are issued for the compliance checks, CSID should be used as the user and the Secret as the password |
| E-invoicing | Production CSID and Secret issued and all eInvoicing calls (reporting and clearance) should include the Basic Authentication header with CSID as the user and Secret as the password. An additional accept-version: v2 header must be added to V2 API calls |

Basic Authentication Format:

- Authorization: Basic {Base64 Encoded String}
- {Base64 Encoded String} = A script containing the CSID, a Colon and the Secret encoded with Base64 (CSID:Secret)

# 4. Frequently Asked Questions (FAQs)

## 4.1 Business FAQs

### 4.1.1 Developer Portal Business FAQs

| # | Question | Answer |
|---|---|---|
| 1 | What is the Developer Portal? | The Developer Portal is a dedicated Portal provided by ZATCA to the e-invoice generation solution developers and the developers community. It provides tailored information in line with e-invoicing requirements and in particular it provides access to a Software Development Kit (SDK) and a Portal-based validator which allows for checking the compliance of specific XML electronic invoices with the  e-invoicing requirements. It also provides access to ZATCA's Integration Sandbox. |
| 2 | What are the main tools or functionalities provided by the Developer Portal? | Through the Developer Portal, users can access:<br>● A Support page, which includes guidance and support information on the Developer Portal functionalities<br>● The SDK page, used for testing the compliance of XML files with the e-invoicing requirements<br>● The Portal-based validator page, which enables non-technical users to check the compliance of XML files by uploading them to the Portal<br>● The Integration Sandbox, which allows developers to test the integration of their systems with a Sandbox environment |

| # | Question | Answer |
|---|---|---|
| 3 | What are the requirements for using the Developer Portal? | The Developer Portal is publicly available to everyone. The users can access the Compliance and Enablement Toolbox (SDK and Web-based Validator) and Support pages without the need for prior registration. However, users who desire to access the SDK and the Integration Sandbox must register by providing the details requested on the page. |
| 4 | Who are the intended users to register for the Developer Portal? | Developers of e-invoice solutions or developers representing taxpayers' in-house teams or non-technical users representing taxpayers (such as tax or accounts teams) who would like to validate the compliance of specific document files (e.g. XML files) with the e-invoicing requirements. |

### 4.1.2 SDK Business FAQs

| # | Question | Answer |
|---|---|---|
| 1 | What is the Compliance and Enablement Toolbox SDK? | The SDK is an offline downloadable tool which can be used to validate the XMLs files in accordance with the E-Invoicing requirements. It also allows validation of the QR codes as per a prescribed structure. |
| 2 | Where can I find the SDK? | The SDK can be found by navigating to the "Systems Developers" page on the ZATCA website, followed by the "Compliance and Enablement Toolbox" page. Through the "Compliance and Enablement Toolbox" page, users can download the SDK after accepting the disclaimer. |
| 3 | Do I have to use the SDK? | It is not mandatory for Taxpayers to use the SDK. However, ZATCA encourages developers to use the SDK to ensure compliance with the E-Invoicing requirements for the QR Code (required from 4 December, 2021) and XML (required starting from 1 January, 2023 onwards).<br>Developers should also use the SDK for offline testing to reduce load on the Integration Sandbox. |
| 4 | Once the XML validation is successful, is it deemed to be accepted by ZATCA? | The purpose of the SDK is to assist the developers to check if the QR Code structure and XML file meets the E-Invoicing requirements and to return specific error messages for correction. Successful validation of XMLs using the SDK should not be deemed as any form of acceptance or approval by ZATCA. |

| | | |
|---|---|---|
| 5 | What are the QR code fields that will be validated in the Generation phase and which are required for the 4th of December 2021? | **The users will be able to validate the following fields:**<br>1. Seller's Name.<br>2. VAT registration number of the seller.<br>3. Timestamp of the electronic invoice or credit/debit note (date and time).<br>4. Electronic invoice or credit/debit note total (with VAT).<br>5. VAT total.<br>Additional fields from the specification and otherwise may be included, but will be disregarded by ZATCA for the 4th of December requirements. |
| 6 | What are the QR code fields that will be validated in the Integration phase starting from 1 January 2023 onwards? | **The users will be able to validate the following fields:**<br>1. Seller's Name.<br>2. VAT registration number of the seller.<br>3. Timestamp of the electronic invoice or credit/debit note (date and time).<br>4. Electronic invoice or credit/debit note total (with VAT).<br>5. VAT amount.<br>6. Hash of XML electronic invoice or credit/debit note.<br>7. Elliptic Curve Digital Signature Algorithm (ECDSA) signature.<br>8. **ECDSA public key:** The public key BLOB format contains only the public portion of an ECDSA key used to generate the Cryptographic Stamp. Length of the public key BLOB for a 256-bit key is 64 bytes (72 bytes including magic number and field length information on some systems). |

| | | |
|---|---|---|
| 6 | Continue | 9. For Simplified Tax Invoices and their associated notes, the ECDSA signature of the cryptographic stamp's public key by ZATCA's technical Certificate Authority (CA) is required.<br>● An ECDSA signature is encoded according to IEEE P1363. This signature format encodes the (r, s) tuple as the concatenation of the big-endian representation of r and the big-endian representation of s.<br>● Each of these values is encoded using the number of bytes required to encode the maximum integer value in the key's mathematical field.<br>● For example, an ECDSA signature from 256-bit elliptic curves (like secp256k1) encodes each of r and s as 32 bytes, and produces a signature output of 64 bytes.<br>**Please find below an example:**<br>public static byte[] extractR(String digitalSignature) throws Exception {<br>MessageDigest digest = MessageDigest.<br>getInstance("SHA-256");<br>byte[] hash =<br>digest.digest(Base64.getDecoder().<br>decode(digitalSignature.getBytes(StandardCharsets.<br>UTF_8)));<br>return Arrays.copyOfRange(hash, 0, 32);<br>}<br>/**<br> * Extract S Component<br> *<br> * @return<br> * @throws Exception<br> */ |

|  |  |  |
|---|---|---|
| 6 | Continue | public static byte[] extractS(String digitalSignature) throws Exception {<br>MessageDigest digest = MessageDigest.getInstance("SHA-256");<br>byte[] hash = digest.digest(Base64.getDecoder().decode(digitalSignature.getBytes(StandardCharsets.UTF_8)));<br>return Arrays.copyOfRange(hash, 32, 64); |

### 4.1.3 Web Based Validator Business FAQs

| # | Question | Answer |
|---|---|---|
| 1 | What is the Web Based Validator for Non-Technical Users? | The Web-based validator can be accessed by anyone from the Developer Portal. It is mainly built to enable non-technical users, (such as some tax and accounting teams,) to test and validate XMLs as per e-invoicing requirements. |
| 2 | Who is eligible to use the Web Based Validator? | The intended users of the Web Based Validator are the non-technical users such as tax teams or accounts teams for taxpayers. Anyone can accesses the Developer Portal (publicly available) to use the Web Based Validator. |
| 3 | What if XML has error(s)? | In case an XML has error(s), specific error messages will be displayed. XMLs can be validated either via the Portal-based validator or the SDK again after the error(s) are fixed. |
| 4 | Is it mandatory to use the Compliance and Enablement Toolbox SDK or Web Based Validator? | It is not mandatory for Taxpayers to use the SDK or the Web Based Validator. However, ZATCA encourages the technical and non-technical users (such as tax teams or accounts teams) to use the SDK and Web Based Validator to ensure compliance with e-invoicing requirements. |

### 4.1.4 Integration Sandbox Business FAQs

| # | Question | Answer |
|---|---|---|
| 1 | What is the Integration Sandbox? | The Integration Sandbox (ISB) is a test platform developed by ZATCA to simulate some of the core e-invoicing platform (FATOORA) functionalities that will be available in the production system. Its primary objective is to allow Solution Developers to build compliant E-invoice Generation Solutions that can submit requests to the ISB and obtain relevant responses to indicate if their integration calls have been successful or if they have any errors. |
| 2 | What is the difference between the Compliance and Enablement Toolbox and the Integration Sandbox? | The Compliance and Enablement Toolbox (CET) comprises of:<br>1. 1. An offline SDK tool to validate QR Code and XMLs; and<br>2. 2. A Portal-based Validator for non-technical users (such as tax or accounts teams) to validate XMLs.<br>The Integration Sandbox allows testing integration of taxpayer's E-invoicing solutions with a sandbox environment using test APIs to send requests and documents in a similar manner to how it would be done on the core e-invoicing platform. This sandbox will perform validations that are part of the SDK and some additional checks that cannot be done offline or are specific to API requests. The SDK requires XML files / QR code strings as inputs while the Integration Sandbox requires an API request as input. |

| | | |
|---|---|---|
| 3 | If my invoices are compliant as per the Compliance and Enablement Toolbox, will they also pass the Integration Sandbox? | XMLs validated by the Toolbox are expected to receive successful responses on the Integration Sandbox also unless there are issues with the API request itself. However, the Sandbox can also run some additional validations. |
| 4 | Will the Integration Sandbox be available to Taxpayers only? | The intended users of the Integration Sandbox are e-invoicing solution developers. Developers can register by providing the requested information and access the API documentation on the Developer Portal. VAT Registration details are not a pre-requisite to register and access the Integration Sandbox. |
| 5 | Does passing the Integration Sandbox mean the E-invoice Generation Solution can be used by a Taxpayer to submit invoices to ZATCA? | No. Taxpayers who are required to integrate with ZATCA will have to undergo an Onboarding and Compliance process to be able to submit electronic documents to ZATCA starting from 1 January 2023 onwards. E-invoice Generation Solutions which undergo adequate testing on the Sandbox will have a higher probability of completing that onboarding and compliance process smoothly. |
| 6 | Can multiple invoices be submitted to the Integration Sandbox? | Yes. However, each invoice, credit or debit note should be part of a separate API call. |
| 7 | Do invoices need to be submitted in sequence to the Integration Sandbox? | The Integration Sandbox does not mandate that the invoices should be submitted in sequence. |
| 8 | Does ZATCA store the invoices submitted to the Integration Sandbox? | No. |

|  |  |  |
|---|---|---|
| 9 | Does the Integration Sandbox require actual taxpayer details on the XML files? | No, dummy information can be provided as long as they meet the syntax and content specifications and the XML implementation standards and validation rules. |
| 10 | If an invoice has been cleared by the Integration Sandbox, can it be issued to a buyer? | No. The Integration Sandbox is not intended to validate actual invoices and is for testing purposes only. The successful validation of an XML using the Integration Sandbox should not be deemed as any kind of acceptance or approval by ZATCA. |
| 11 | Can I use the same username and password that I used to access the Compliance and Enablement Toolbox SDK on the Developer Portal to log into the Integration Sandbox? | Yes, the registration and login process is common for both the Compliance and Enablement Toolbox SDK and the Integration Sandbox. |
| 12 | What is a Cryptographic Stamp Identifier (CSID)? | The CSID is technically a cryptographic certificate, which is a credential that allows for authenticated signing and encryption of communication. The certificate is also known as a public key certificate or an identity certificate. It is an electronic document used as proof of ownership of a public key.<br>The CSID is used to uniquely identify an Invoice Generation Solution Unit in possession of a taxpayer for the purpose of stamping (technically cryptographically signing) Simplified Invoices and for accessing the Reporting and Clearance APIs using TLS authentication. |

|  |  |  |
|---|---|---|
| 13 | What is the difference between a Compliance and Production CSID? | A Compliance CSID is an intermediate CSID provided in response to the CSR submission from an EGS or other solution. In the Core E-invoicing Solution, the Compliance CSID is required to complete some compliance checks before the EGS or other solution is able the Production CSID which is required for authenticating the EGS or other solution to ZATCA. In the Sandbox, the compliance checks are not required, and the purpose is to therefore to test the integration calls of obtaining the Compliance and Production CSIDs. |
| 14 | Can I use the same CSID for any invoice submission? | Yes. As long as the VAT Registration number on the CSID matches the VAT Registration Number on the documents. In other words, for every VAT Registration Number being tested across any API call, a CSID with the same VAT Registration Number is required. Note that the VAT Registration number can be any dummy number that meets the syntax specifications (15 digits, starting with 3 and ending with 3).<br>Only a test Production CSID can be used for submitting invoices, credit or debit notes as well as QR codes. |
| 15 | What is the difference between an error and warning? | Errors are associated with invalid invoices, credit or debit notes causing the rejection of such submissions. Warnings are associated with accepted documents which are still not fully compliant with the specifications and standards. Currently the only warning case is an error with the Seller Address and is meant for EGS units to be able to read warning messages. |
| 16 | Can anyone access sandbox and FATOORA portal? | No, Sandbox can be accessed by anyone, but FATOORA production system can be accessed only by taxpayers using Taxpayer portal credentials (ERAD credentials). |
| 17 | Can FATOORA portal and Sandbox be accessed from anywhere or only from KSA? | Yes, both FATOORA and Sandbox can be accessed from anywhere globally, not only from KSA |

## 4.2 Technical FAQs

### 4.2.1 Developer Portal Technical FAQs

| # | Question | Answer |
|---|---|---|
| 1 | Where can I find more information on the Compliance and Enablement Toolbox SDK, the Portal based validator and the Integration Sandbox? | User manuals contains detailed information on SDK, Portal-based validator and the Integration Sandbox. These can be found in the dedicated pages on the Developer Portal. |

### 4.2.2 SDK Technical FAQs

| # | Question | Answer |
|---|---|---|
| 1 | What is an XML? | An XML is a way to present information in a structured and machine readable format. The ZATCA e-invoice format is based on XML and several other XML-based standards. |
| 2 | What is Command Line Interface (CLI)? | A CLI is a way to access and utilize a software application using commands and it is text-based. CLI tools like the FATOORA tool can be used in scripts to create automations.<br>**Sample:** fatoora validatexml -f (invoicename.xml)<br>In this example, we are naming an application called "fatoora", in which we want to use the validatexml feature with a-f command.<br>**The second part that we add is:** (invoicename.xml) which is the path and filename of the XML to be validated. |

|  |  |  |
|---|---|---|
| 3 | 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 package file format that is generally used to aggregate many Java class files and associated metadata and resources (text, images, etc.) into one file for distribution. |
| 4 | Can I validate the Arabic language fields in a QR code within the CLI? | No, since the CLI does not support Arabic characters display. |
| 5 | What JAVA version should I install before using the SDK? | The prerequisite is using the Java SDK (JAR) versions >=11 and <15. |
| 6 | 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. |
| 7 | Where can I find examples of XML files? | Sample XML files are included in SDK. You can download the latest SDK from Developers portal (Sandbox). |

## 4.2.3 Web Based Validator for Non-Technical Users Technical FAQs

| # | Question | Answer |
|---|---|---|
| 1 | What is a QR code? | A QR code is a coded representation of readable text. In the context of e-invoicing, the QR code should contain specific information in a specific format. |
| 2 | How can the users access information contained in the QR code? | In the context of e-invoicing, users should scan the QR code on e-invoices, debit notes and credit notes by using the ZATCA VAT app. This app is available on the Google Playstore and iOS App Store free of charge. |
| 3 | What can I do if an XML has error(s)? | If an XML has error(s), specific error message(s) will be displayed. Error(s) are likely to occur in cases such as when a mandatory field is missing or a value is in an incorrect format. The user may require the assistance of a technical expert to solve the error(s). |

### 4.2.4 Integration Sandbox Technical FAQs

| # | Question | Answer |
|---|---|---|
| 1 | What is the “Reporting API”? | The "Reporting API" reports a single simplified invoice, credit note, or debit note. Specifically, it accepts a simplified invoice, credit note, or debit note encoded in base64 and validates it to ensure:<br>1. Compliance to the UBL2 XSD.<br>2. EN 16931 Rules subset.<br>3. KSA Specific Rules set. KSA Rules set will override EN 16931 Rules set in case the same rule exists in both sets.<br>4. QR Code validation<br>5. Cryptographic Stamp validation |
| 2 | How can the user access "Reporting API"? | The user will need to do a POST Method on endpoint / invoices/reporting/single and pass it on "authentication-certificate" and accept-language as a parameter in the header. More information can be found on the Integration Sandbox section of the Developer Portal. |
| 3 | What’s the Request Body the user should send while calling "Reporting API"? | The body object should Contain 2 Values: the first one is called "invoiceHash" and the second one is called "invoice".<br>Example:<br>{<br>"invoiceHash": "string",<br>"invoice": "string"<br>}<br>More information can be found on the Integration Sandbox section of the Developer Portal. |

| | | |
|---|---|---|
| 4 | What should the user expect as a response if calling the "Reporting API" was a success? | The response will be 200 HTTP Ok with a Retrieved object containing 4 values : "invoiceHash","Status" ,"Warnings", "errors" .<br>Retrieved object Example:<br>{<br> "invoiceHash": "TODO Add Invoice Hash",<br> "status": "Reported",<br> "warnings": null,<br> "errors": null<br>}<br>More information can be found on the Integration Sandbox section of the Developer Portal. |
| 5 | What is the "Clearance API"? | The "Clearance API" clears a single standard invoice, credit note, or debit note. Specifically, it accepts standard invoice, credit note, or debit note encoded in base64 and validates it to ensure:<br>1. Compliance to the UBL2 XSD.<br>2. EN 16931 Rules subset.<br>3. KSA Specific Rules set. KSA Rules set will override EN 16931 Rules set in case the same rule exists in both sets.<br>On successful validation, the api then applies a cryptographic stamp from ZATCA side and generates a QR Code string. After that the XML is returned back. |
| 6 | How can the user access "Clearance API"? | The user will need to do a POST Method on endpoint /invoices/clearance/single and pass it on "authentication-certificate" and accept-language as a parameter in the header. More information can be found on the Integration Sandbox section of the Developer Portal. |

| | | |
|---|---|---|
| 7 | What's the Request Body the user should send while calling "Clearance API"? | The body object should Contain 2 Values: the first one is called "invoiceHash" and the second one is called "invoice".<br>Example:<br>{<br>"invoiceHash": "string",<br>"invoice": "string"<br>}<br>More information can be found on the Integration Sandbox section of the Developer Portal. |
| 8 | What should the user expect as a response if calling "Clearance API" was a Success? | The response will be 200 HTTP Ok with a Retrieved object containing 4 values : "invoiceHash","Status" ,"Warnings", "errors" .<br>Retrieved object Example:<br>{<br>"invoiceHash": "TODO Add Invoice Hash",<br>"status": "Reported",<br>"warnings": null,<br>"errors": null<br>}<br>More information can be found on the Integration Sandbox section of the Developer Portal. |
| 9 | What are the Response causes (Code & Description) that can appear while calling "Reporting single API"? | Code - Description<br>● 200- HTTP OK<br>● 202- Accepted with Errors, simplified invoice accepted with warning errors<br>● 303- HTTP See Other. Returned when the submitted invoice is a Standard Invoice while clearance is activated<br>● 400- HTTP Bad Request. Returned when the submitted request is invalid<br>● 500- HTTP Internal Server Error. Returned when the service faces internal errors |

| | | |
|---|---|---|
| 10 | What are the Response causes (Code & Description) that can appear while calling "Clearance single API"? | Code - Description<br>● 200- HTTP OK<br>● 202- Accepted with Errors, clearance invoice accepted with warning errors<br>● 303- HTTP See Other. Returned when the submitted invoice is a Standard Invoice while clearance is activated<br>● 400- HTTP Bad Request. Returned when the submitted request is invalid<br>● 500- HTTP Internal Server Error. Returned when the service faces internal errors |
| 11 | What is a CSR ? | A certificate signing request (CSR) is one of the first steps towards getting a Cryptographic Stamp Identifier for a device / solution unit. The CSR contains information (e.g. common name, organization, country) the ZATCA Certificate Authority (CA) will use to create your CSID. It also contains the public key that will be included in your CSID and is signed with the corresponding private key. Please refer to the CSID API Swagger files for more details |

# 5. Appendix

## 5.1 Glossary

| | |
|---|---|
| ZATCA | ZAKAT, Tax and Customs Authority |
| XML | Extensible Markup Language |
| SDK | Software Development Kit |
| QR Code | Quick Response Code |
| SDLC | Software Development Life Cycle |
| CN | Credit Note |
| DN | Debit Note |
| CLI | Command Line Interface |
| Integration Sandbox | The Integration Sandbox should enable solution developers to simulate the integration calls/ requests |
| CET | Compliance and Enablement Toolbox |

| | |
|---|---|
| ISB | Integration Sandbox |
| EGS | E-invoice Generation Solution |
| CRM | Customer Relationship Management |
| PKI | Public Key Infrastructure |
| JAR | JAVA Archive |
| API | Application Programming Interface |
| CSID | Cryptographic Stamp Identifier |
| CSR | Certificate Signing Request |

## 5.2 Developer Portal Security Information

The Developer Portal uses HTTPS, as a secure method of communication between the browser and the server. The user account is protected by a username and password. The session stays alive for 8 hours after which the user will need to sign in again.

## 5.3 Generate CSR

### 5.3.1 Initiate a CSR configuration file (Open SSL Config. File)

As a part of the first-time onboarding and renewal process, the Taxpayer's EGS Unit(s) must submit a CSR to the E-invoicing Platform once an OTP is entered into the EGS unit. The CSR is an encoded text that the EGS Unit(s) submits to the E-invoicing Platform and the ZATCA CA in order to receive a Compliance CSID, which is a self-signed certificate issued by the E-invoicing Platform allowing clients to continue the Onboarding process.The certificate signing request is encoded text that service providers/ own solution will submit it to ZATCA CA. The digital certificate will be stored in the taxpayer device/s and EGS identification data will rely on the data provided by the taxpayer through ZATCA Portal without further validation and therefore, the taxpayer is fully responsible for the accuracy and legitimacy of the data provided. Also, CSR contains the public key that will be included in the certificate, the private key is usually created at the same time that service providers/ own solution create the CSR by their selves.

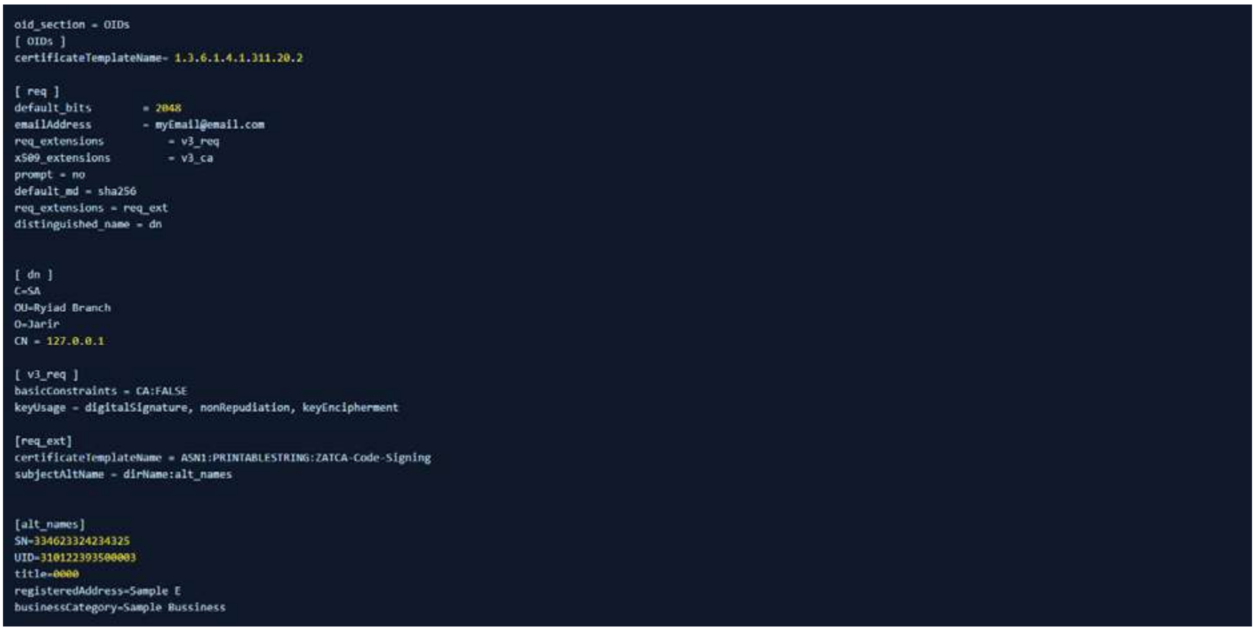
The CSR inputs (Open SSL Config. File) are as follows:

| CSR Inputs | Business Term | Description | Specification |
|---|---|---|---|
| Common Name | Name or Asset Tracking Number for the Solution Unit | Provided by the Taxpayer for each Solution unit: Unique Name or Asset Tracking Number of the Solution Unit | Free text |
| EGS Serial Number | Manufacturer or Solution Provider Name, Model or Version and Serial Number | Automatically filled and not by the taxpayer: Unique identification code for the EGS.<br>Manufacturer serial number for each solution unit including<br>1. Manufacturer or Solution Provider Name \| 2-Model or Version \|3-SerialNumber | Free text<br>Validate the format with a Regular Expression (1-...\| 2-... \|3-....) |

|  |  |  |  |
|---|---|---|---|
| Organization Identifier | VAT or Group VAT Registration Number | VAT Registration Number of the Taxpayer (Taxpayer / Taxpayer device to provide this to allow to check if the OTP is correctly associated with this TIN) | 15 digits, starting and ending with 3 |
| Organization Unit Name | Organization Unit | The branch name for taxpayers. In case of VAT Groups this field should contain the 10-digit TIN number of the individual group member whose EGS Unit is being onboarded | If 11th digit of Organization Identifier is not = 1 then Free text<br>If 11th digit of Organization Identifier = 1 then needs to be a 10 digit number |
| Organization Name | Taxpayer Name | Organization/Taxpayer Name | Free text |
| Country Name | Country Name | Name of the country | 2 letter code |
| Invoice Type (Functionality Map) | Functionality Map | The document type that the Taxpayer's solution unit will be issuing/generating. It can be one or a combination of Standard Tax Invoice (T), Simplified Tax Invoice (S), Buyer QR code (C), Seller's QR code in self-billing (Z). | 4-digit binary number (0s and 1s only, cannot all be 0s) |

| | | | |
|---|---|---|---|
| Invoice Type (Functionality Map) | Functionality Map | The input should be using the digits 0 & 1 and mapping those to "TSCZ" where:<br>0 = False/Not supported<br>1= True/Supported<br>For example: 1000 would mean Solution will be generating Standard Invoices only. 0100 would mean Solution will be generating Simplified invoices only. and 1100 means Solution will be generating both Standard and Simplified invoices | 4-digit binary number (0s and 1s only, cannot all be 0s) |
| Location | Location of Branch or EGS Unit | The address of the Branch or location where the device or solution unit is primarily situated (could be website address for e-commerce) | Free Text |
| Organization Name | Taxpayer Name | Organization/Taxpayer Name | Free text |
| Industry | Industry or Location | Industry or sector for which the device or solution will generate invoices | Free Text |

The screenshot below represents the information the user must use to generate a CSR (using Open SSL Command Tool) and its configuration file as shown below. For further information, please see link: / index.html (openssl.org)



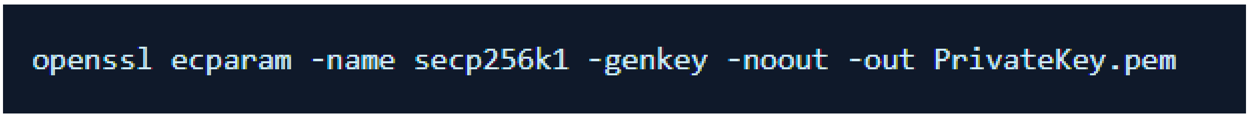
```
oid_section = OIDs
[ OIDs ]
certificateTemplateName= 1.3.6.1.4.1.311.20.2

[ req ]
default_bits            = 2048
emailAddress            = myEmail@email.com
req_extensions              = v3_req
x509_extensions             = v3_ca
prompt = no
default_md = sha256
req_extensions = req_ext
distinguished_name = dn


[ dn ]
C=SA
OU=Ryiad Branch
O=Jarir
CN = 127.0.0.1

[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment

[req_ext]
certificateTemplateName = ASN1:PRINTABLESTRING:ZATCA-Code-Signing
subjectAltName = dirName:alt_names


[alt_names]
SN=334623324234325
UID=310122393500003
title=0000
registeredAddress=Sample E
businessCategory=Sample Bussiness
```

### 5.3.2 Generate public/private key pair

- According to security implementation document the Key pair shall be generated according to FIPS 186. Further, reasonable techniques shall be used to validate the suitability of the generated key pair.
- The suitability of keys shall be done according to either the ECC Full Public Key Validation Routine or the ECC Partial Public Key Validation Routine. [Source: Sections 5.6.2.3.2 and 5.6.2.3.3, respectively, of NIST SP 56A: Revision 2].
- Keys must be marked as non-exportable in order to prohibit key export out of the security module where the key was generated
- A hardware or software based security module can be used to generate and store the key pair as long as the above requirements are met.

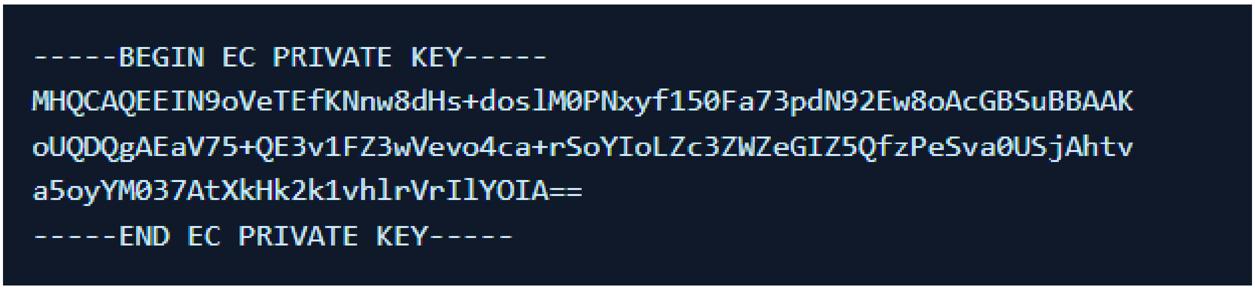
### 5.3.2.1 Generate Private Key

The service providers/ own solution need to keep the private key secret. The created certificate will only work with a particular private key that was generated. So if the private key lost, the certificate will no longer work. we are generating a pair of ECDSA keys with the P-256 (secp256k1) curve, the PrivateKey.pem file will be the generated private key, change the file name to YourPrivateKey.pem. the following command show how to generate a private key using OpenSSL:



```
openssl ecparam -name secp256k1 -genkey -noout -out PrivateKey.pem
```

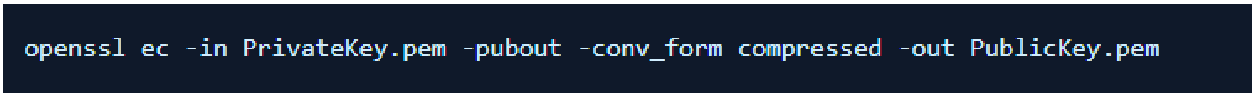
Sample contents of the PrivateKey.pem private key in PEM format:



```
-----BEGIN EC PRIVATE KEY-----
MHQCAQEEIN9oVeTEfKNnw8dHs+doslM0PNxyf150Fa73pdN92Ew8oAcGBSuBBAAK
oUQDQgAEaV75+QE3v1FZ3wVevo4ca+rSoYIoLZc3ZWZeGIZ5QfzPeSva0USjAhtv
a5oyYM037AtXkHk2k1vhlrVrIlYOIA==
-----END EC PRIVATE KEY-----
```

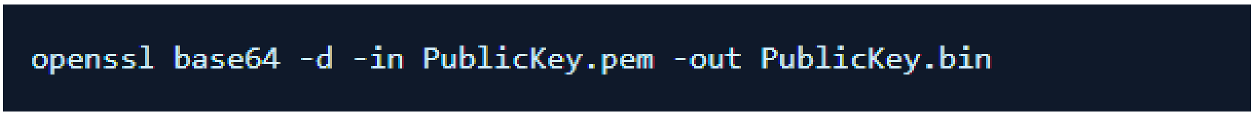
### 5.3.2.2 Generate Public Key

The compressed public key will be created by extracting it from the private key, extracting ECDSA keys with the P-256 (secp256k1) curve, the PublicKey.pem file will be the extracted public key, change the file name to YourPublicKey.pem. The following command show how to extract a public key using OpenSSL:



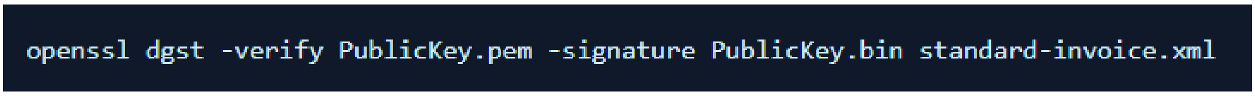
```
openssl ec -in PrivateKey.pem -pubout -conv_form compressed -out PublicKey.pem
```

By using the compressed public key only X values will be used in the elliptic curves:



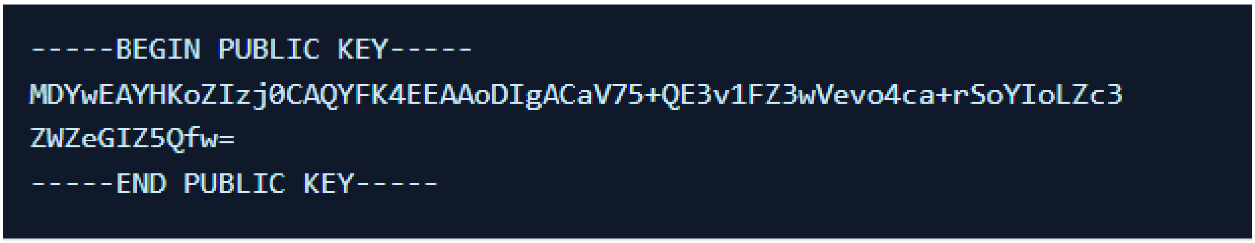
```
openssl base64 -d -in PublicKey.pem -out PublicKey.bin
```

The base64 public key will be use to validate the signature for the standard invoice:



```
openssl dgst -verify PublicKey.pem -signature PublicKey.bin standard-invoice.xml
```

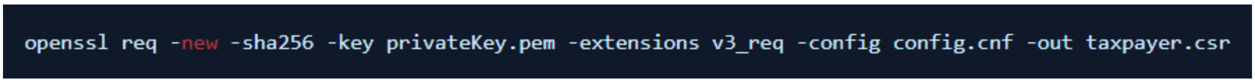
Sample contents of the PublicKey.pem public key in PEM format:



```
-----BEGIN PUBLIC KEY-----
MDYwEAYHKoZIzj0CAQYFK4EEAAoDIgACaV75+QE3v1FZ3wVevo4ca+rSoYIoLZc3
ZWZeGIZ5Qfw=
-----END PUBLIC KEY-----
```

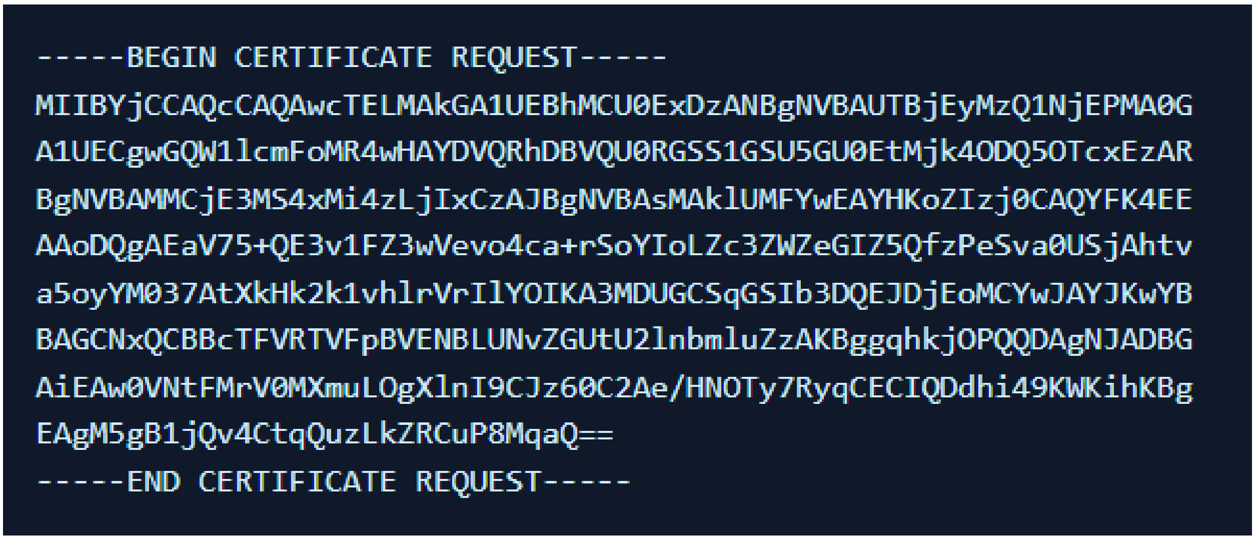
### 5.3.3 Generate a Certificate Signing Request

The service providers / own solution need to run the following command in order to generate the certificate signing request, the command include the request to generate the certificate with -sha256.



```
openssl req -new -sha256 -key privateKey.pem -extensions v3_req -config config.cnf -out taxpayer.csr
```

Sample contents of the CSR:

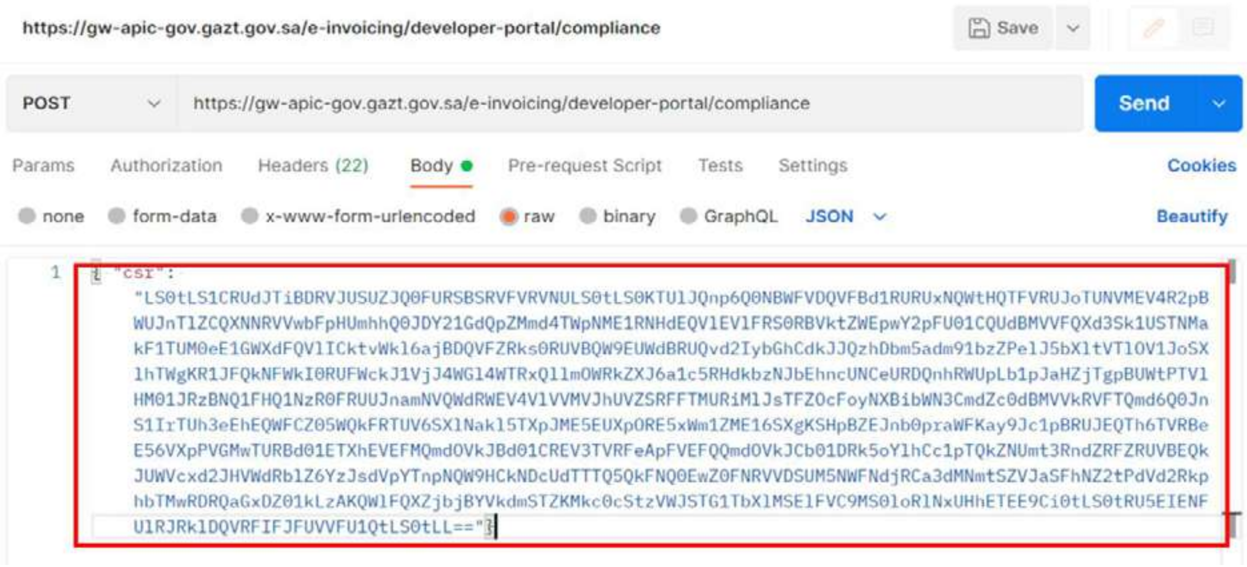


```
-----BEGIN CERTIFICATE REQUEST-----
MIIBYjCCAQcCAQAwcTELMAkGA1UEBhMCU0ExDzANBgNVBAUTBjEyMzQ1NjEPMA0G
A1UECgwGQW1lcmFoMR4wHAYDVQRhDBVQU0RGSS1GSU5GU0EtMjk4ODQ5OTcxEzAR
BgNVBAMMCjE3MS4xMi4zLjIxCzAJBgNVBAsMAklUMFYwEAYHKoZIzj0CAQYFK4EE
AAoDQgAEaV75+QE3v1FZ3wVevo4ca+rSoYIoLZc3ZWZeGIZ5QfzPeSva0USjAhtv
a5oyYM037AtXkHk2k1vhlrVrIlYOIKA3MDUGCSqGSIb3DQEJDjEoMCYwJAYJKwYB
BAGCNxQCBBcTFVRTVFpBVENBLUNvZGUtU2lnbmluZzAKBggqhkjOPQQDAgNJADBG
AiEAw0VNtFMrV0MXmuLOgXlnI9CJz60C2Ae/HNOTy7RyqCECIQDdhi49KWKihKBg
EAgM5gB1jQv4CtqQuzLkZRCuP8MqaQ==
-----END CERTIFICATE REQUEST-----
```

### 5.3.4 Testing the certificate

The service providers/own solution can test the compliance of the generated CSR using the requests that can be found in the below Postman collection:

The service providers/own solution needs to add the generated CSR in the body of the request:



```
{ "csr":
    "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0KTUlJQnp6Q0NBWFVDQVFBd1RURUxNQWtHQTFVRUJoTUNVMEV4R2pB
    WUJnTlZCQXNNRVVwbFpHUmhhQ0JDY21GdQpZMmd4TWpNME1RNHdEQVlEVlFRS0RBVktZWEpwY2pFU01CQUdBMVVFQXd3Sk1USTNMa
    kF1TUM0eE1GWXdFQVlICktvWkl6ajBDQVFZRks0RUVBQW9EUWdBRUQvd2IybGhCdkJJQzhDbm5adm91bzZPelJ5bXltVTlOV1JoSX
    1hTWgKR1JFQkNFWkI0RUFWckJ1VjJ4WGl4WTRxQllmOWRkZXJ6a1c5RHdkbzNJbEhncUNCeURDQnhRWUpLb1pJaHZjTgpBUWtPTVl
    HM01JRzBNQ1FHQ1NzR0FRUUJnamNVQWdRWEV4VlVVMVJhUVZSRFFTMURiM1JsTFZOcFoyNXBibWN3CmdZc0dBMVVkRVFTQmd6Q0Jn
    S1IrTUh3eEhEQWFCZ05WQkFRTUV6SXlNakl5TXpJME5EUXpORE5xWm1ZME16SXgKSHpBZEJnb0praWFKay9Jc1pBRUJEQTh6TVRBe
    E56VXpPVGMwTURBd01ETXhEVEFMQmdOVkJBd01CREV3TVRFeApFVEFQQmdOVkJCb01DRk5oYlhCc1pTQkZNUmt3RndZRFZRUVBEQk
    JUWVcxd2JHVWdRblZ6YzJsdVpYTnpNQW9HCkNDcUdTTTQ5QkFNQ0EwZ0FNRVVDSUM5NWFNdjRCa3dMNmtSZVJaSFhNZ2tPdVd2Rkp
    hbTMwRDRQaGxDZ01kLzAKQWlFQXZjbjBYVkdmSTZKMkc0cStzVWJSTG1TbXlMSElFVC9MS01oRlNxUHhETEE9Ci0tLS0tRU5EIENF
    U1RJRklDQVRFIFJFUVVFU1QtLS0tLL=="}
```

## ! External Document

This guide has been prepared for educational and awareness purposes only, its content may be modified at any time. It is not considered in any way binding to ZATCA and is not considered in any way a legal consultation. It cannot be relied upon as a legal reference in and of itself, It is always necessary to refer to the applicable regulations in this regard. Every person subject to zakat, tax and customs laws must check his duties and obligations, he is solely responsible for compliance with these regulations. ZATCA shall not be responsible in any way for any damage or loss The taxpayer is exposed to that results from non-compliance with the applicable regulations.

scan this code to view the last
version and all published documents
or visit the website zatca.gov.sa