



2. Developer Portal Overview

2.1 Pre-requisites

The Developer Portal itself is a web-based application and can be run from any modern browser such as Google Chrome, Microsoft Edge or Apple Safari.

The SDK is a Java based JAR file that can run on all leading platforms including Windows, Linux and Mac. The Java SDK (JAR) will run on JDK versions ≥ 11 and < 15 , to comply with secp256k1 as per ZATCA security regulations.

The Integration Sandbox APIs can be accessed from all leading platforms as those mentioned above. REST APIs can be accessed from any Rest Client tools (Postman) for testing or using any coding languages (java, .Net, PHP, Nodejs, etc.) to call the rest services using HTTPs Protocol.





2.2 Structure / Sitemap

The Developer Portal is comprised of the following:

Developer Portal			
Login		Access Portal Based Validator	Access Developer Portal Support Page
Access SDK Page	Access Integration Sandbox Page		
<ul style="list-style-type: none">● Download SDK● SDK Support● SDK Documentation● SDK Version History	<ul style="list-style-type: none">● Access API Documentation (Swagger Files)● Test APIs for Onboarding, Renewal, Reporting and Clearance	Validate XMLs	Access FAQs
Outside Developer Portal			
Using the SDK	Using the Integration Sandbox (APIs)		
<ul style="list-style-type: none">● Test compliance of XML● Test compliance of QR Code (Generation Phase)● Test Compliance of QR Code (Integration Phase)	<ul style="list-style-type: none">● Test APIs to obtain Compliance CSID and Production CSID (as part of Onboarding process)● Test APIs to obtain new Compliance CSID and Production CSID (Test the Renewal process)● Test API to submit documents for Reporting● Test API to submit documents for Clearance		





2.3 User Journeys

The recommended steps for Solution Developers are:

1. Read the XML Implementation Standards, Security Features Implementation Standards and Data Dictionary
2. Access the Developer Portal
3. Create a Developer Portal Account
4. Login to the Developer Portal as a Registered User
5. Access the SDK Page
6. Read the SDK Support and Documentation
7. Download the SDK
8. Test XML compliance using the SDK via CLI / local integration
9. Access the Integration Sandbox Page
10. Go through the API Documentation on Swagger
11. Test the APIs through Swagger
12. Test the APIs via integration
13. Leveraging the Developer Portal Support page FAQs for troubleshooting

The recommended steps for Non-technical users are:

1. Access the Developer Portal
2. Accessing the Compliance and Enablement Toolbox Portal Page
3. Test XML compliance
4. Provide the error messages / responses (if any) to Solution Developers
5. Leveraging the Developer Portal Support page FAQs for troubleshooting



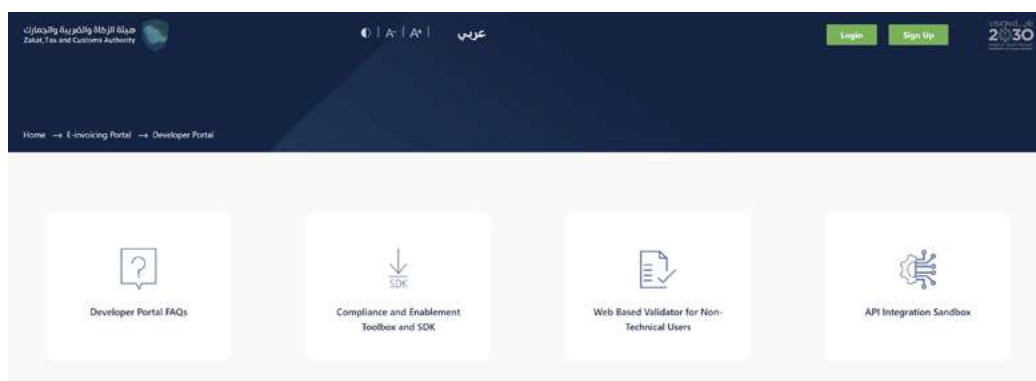


2.3.1 Accessing the Developer Portal

The process for accessing the Developer Portal is as follows:

1. Access the Developer Portal through the following weblink (<https://sandbox.zatca.gov.sa/>).
2. The user is directed to the Developer Portal main dashboard / landing page
 1. In this page the user can access the below sections without registration or login:
 1. Developer Portal Support Page which includes the FAQs.
 2. Web Based Validator for Non-Technical Users.
 2. The following sections would require the user to create a Developer Portal account:
 1. Compliance and Enablement Toolbox SDK Page.
 2. Integration Sandbox Page.

Note: The User can chose to toggle the language between English and Arabic by using the icon on the top right-hand side of the page.



Developer Portal main landing page





2.3.2 Creating a Developer Portal Account

As mentioned above, a Developer Portal account is required for accessing the Compliance and Enablement Toolbox SDK page and the Integration Sandbox page. You can ignore this step if you only wish to access the Web Based Validator or the Developer Portal Support page.

Once the user is on the main dashboard of the Developer Portal, they can click on the "Sign up" button at the top right-hand side as seen in the Figure below.

- Email ID
- First Name
- Last Name
- Company Name (optional field)
- Password
- Confirm Password

In the Sign Up page (as seen in the Figure below), the user will be prompted to create a new account by providing the following details:

The email must be a valid email and the password must be at least 8 characters comprising of at least one number, one letter each in lower and upper case, and one symbol.

After completing all the necessary fields, the user should click on the CAPTCHA verification followed by the Sign up button.





Login Page

After the user has signed up and created their account credentials, they can proceed to the Login page where they will be prompted to:

- Fill in the User Name and Password (as created by the user).
- Click the CAPTCHA.
- In addition, the user can click "Forgot Password"
- In the case where the user does not have an account set up and requires one, the user can click on the Sign Up option, in order to create a new account and proceed to the process described in this Section 2.3.2 of the User Manual for registration.
- After filling in all the information, the user should click on the Login button in order to proceed to the main dashboard again where the user will now also be able to access the Compliance and Enablement Toolbox SDK page and the Integration Sandbox page.
- A logged in user can logout at any time by clicking on the logout option on the header. The user can also change the password at any time by clicking on the arrow next to the user profile icon in header.





Home -> E-invoicing Portal -> Developer Portal -> Sign Up

Developer Portal Signup

Email ID *

First Name *

Last Name *

Company Name

Password *

Confirm Password *

☐ I'm not a robot

reCAPTCHA

Sign Up

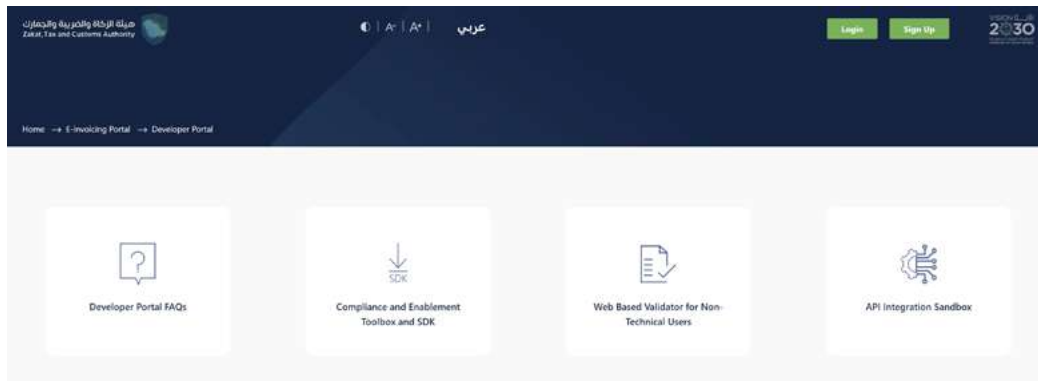
2.3.3 Accessing the Compliance and Enablement Toolbox SDK Page

The Compliance and Enablement Toolbox (SDK) which is an offline downloadable tool that can be used to validate an XML based e-invoice, credit or debit note files in accordance with the ZATCA published XML Implementation Standards. It also allows validation of the QR codes as per the prescribed structure. Developers can integrate their EGS units with the SDK locally (offline) or also test using a CLI.

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

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



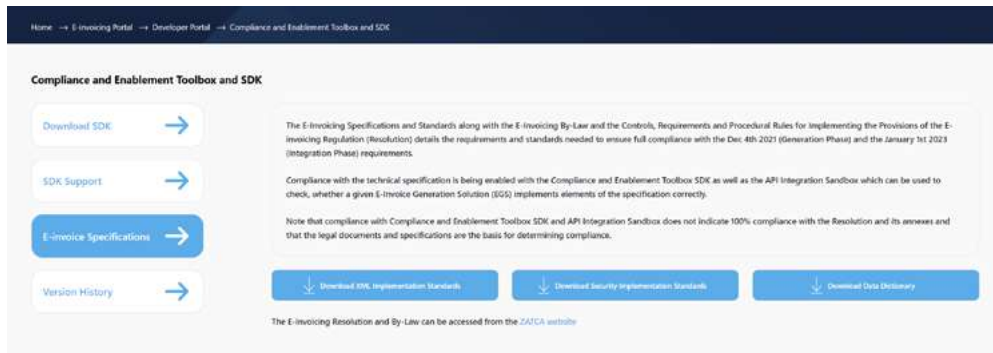


Accessing the Compliance and Enablement Toolbox (SDK)

After the user has accessed the Compliance and Enablement Toolbox SDK Page, the user can:

- Access the SDK support, 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 each Operating System/ environment.
- Access documentation such as the XML Implementation Standards (E-Invoice XML Implementation Standard), Security Features and Implementation Standards (E-Invoice Security Features and Implementation Standards) & Data Dictionary (E-Invoice Data Dictionary)
- Download the SDK after accepting the terms and conditions.
- View the version history which contains earlier releases of the SDK.



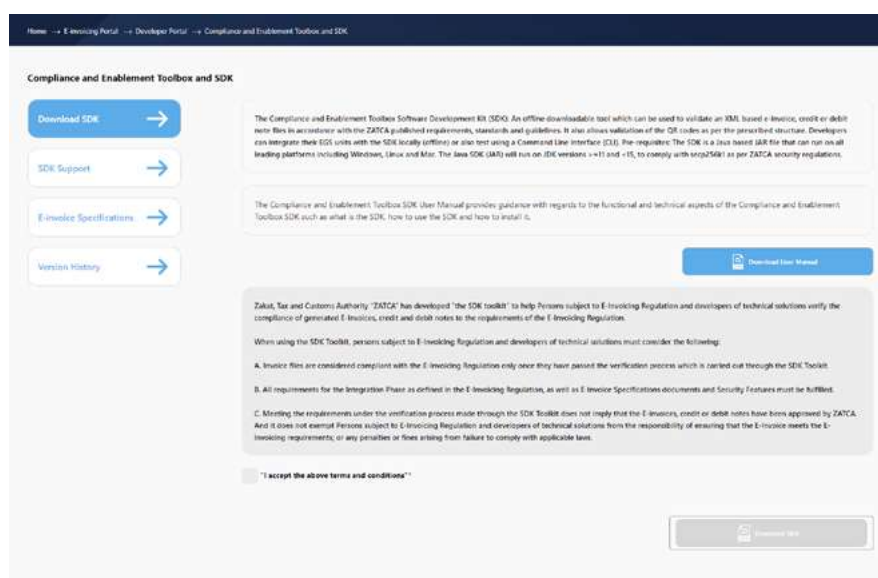


Accessing the E-Invoicing specification documents

2.3.4 Downloading the SDK

In order to download the SDK, the process is as follows:

- The user clicks on "Download SDK"
- The user has to click on "I accept the above terms and conditions"
- As the above is clicked, the "Download SDK" button will be activated and become available for the user to click on



downloading the SDK





2.3.5 Using the SDK (outside of the Developer Portal)

Please refer to the ZATCA E-Invoice Java SDK (CLI) Manual on the below link by downloading the SDK and then navigate to readme folder.

<https://zatca.gov.sa/ar/E-Invoicing/SystemsDevelopers/ComplianceEnablementToolbox/Pages/DownloadSDK.aspx>.

2.3.6 Accessing the Web Based Validator for Non-Technical Users

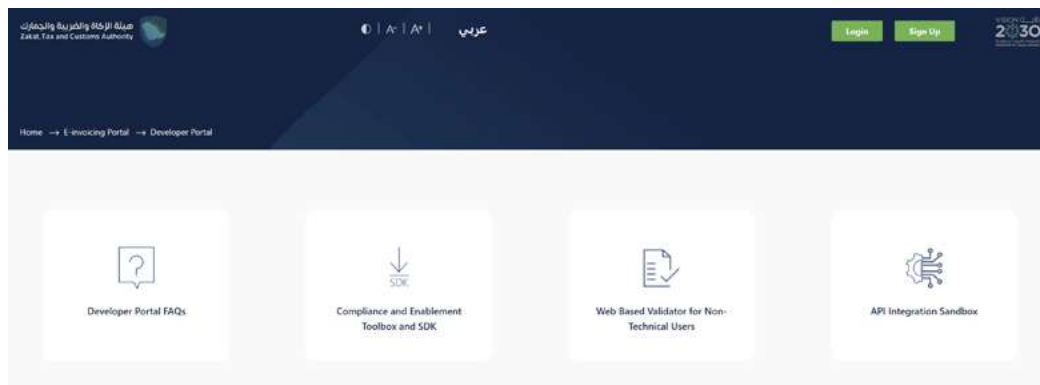
The user can test - using a web portal - the compliance of the XMLs of standard e-invoices, credit or debit notes generated so that they can know if they are in line with the ZATCA e-invoicing specifications and regulations or so that they can be alerted to any errors which are causing non-compliance with the ZATCA specifications and regulations. It is aimed at intermediate or non-technical users to validate XML based e-invoices, credit or debit note files from the portal directly, i.e. without the need to download the SDK or possess the technical know-how to run it.

This section details the process of accessing the "Web Based Validator for Non-Technical Users" in order to test the compliance of the e-invoice, credit and debit note XMLs. Users can access the "Web Based Validator for Non-Technical Users" Page through the Developer Portal (no prior registration or login is required). On this page, users can view information related to what the Web Based Validator aims to achieve and the user can access this and begin uploading the XMLs that they would want to test and validate.

The process for accessing the "Web Based Validator for Non-Technical Users" page is as follows:

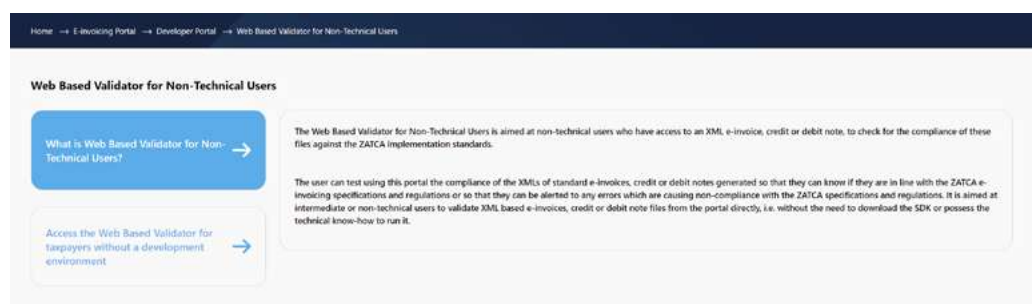
- The User accesses the "Web Based Validator for Non-Technical Users" on the Developer Portal (no prior registration or log in required).





Accessing the web based validator

- On the "Web Based Validator for Non-Technical Users", users can view information related to an explanation of the Web Based Validator and what it aims to do
- In addition, users can click on "Access the Web Based Validator for taxpayers without a development environment" in order to begin testing and validating their XMLs.



Web based validator Page

- Once users have chosen to "Access the Web Based Validator for taxpayers without a development environment", a disclaimer is shown detailing that:





The portal validation page is a standalone application and compliance does not necessarily imply the e-invoices, credit or debit notes have been accepted by ZATCA. All Taxpayer E-invoicing solution unit will need to pass the testing requirements as part of Registration/Taxpayer Onboarding prior to submitting e-invoices, credit or debit notes to ZATCA.

- The User has to acknowledge the disclaimer in order to proceed to test their XML files.

Home → E-invoicing Portal → Developer Portal → Web Based Validator for Non-Technical Users

Web Based Validator for Non-Technical Users

What is Web Based Validator for Non-Technical Users? →

The portal validation page is a standalone and offline application and compliance does not necessarily imply the e-invoices, credit or debit notes have been accepted by ZATCA. All Taxpayer solutions and devices will need to pass the testing requirements as part of Registration/Taxpayer Onboarding prior to submitting e-invoices, credit or debit notes to ZATCA.

Access the Web Based Validator for taxpayers without a development environment →

☐ "I accept the above terms and conditions"

Test XML file

Web based validator Disclaimer

2.3.7 Using the Web Based Validator for Non-Technical Users

An XML file can be validated according to its structure (schema), fields, or ZATCA requirements (i.e. The VAT registration number must be 15 numeric digits). The way this works is that the user submits an XML and the portal will read it, analyze it, and return the status of the validation.

Note that the Web Based Validator can be used to validate up to 5 XMLs and if more than 1 XML is provided, the validator also checks for the sequence in terms of Previous Invoice (Document) Hash. Note that for a single XML the Previous Document Hash check is always considered as valid or True.





The process for validating XMLs from the Web Based Validator for Non-Technical Users page is as follows:

Click on "Upload XML file" and choose a file, then click "Validate."

The screenshot shows a web interface with a dark blue header containing a breadcrumb trail: Home → E-invoicing Portal → Developer Portal → Web Based Validator for Non-Technical Users → Test XML file. Below the header, the page title 'Test XML file' is displayed. There is a text input field labeled 'Upload XML File' with a small note below it stating 'Upload at maximum 5 files only'. To the right of the input field is a grey button labeled 'Validate'.

Uploading an XML file on the web based validator

If the XML is compliant, you will receive a "Valid": true message.

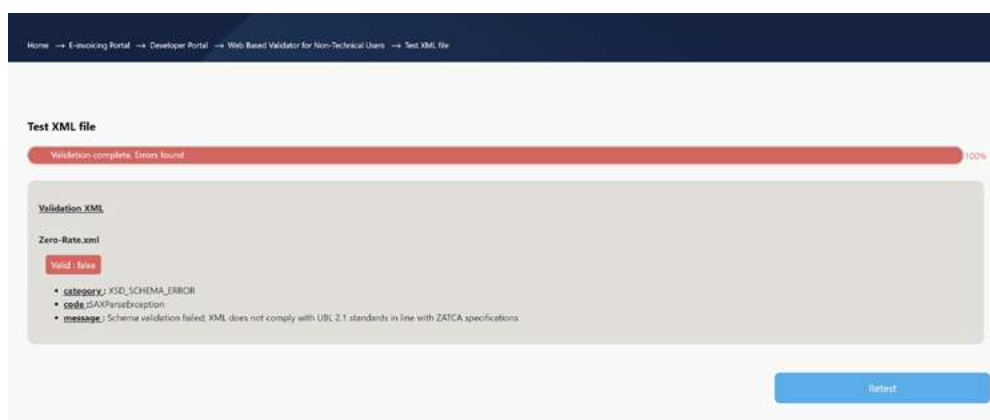
The screenshot shows the same web interface as before, but now with a green progress bar at the top indicating 'Validation complete. No Errors found' with a 100% completion status. Below the progress bar, the text 'Validation XML' is shown, followed by the filename 'Standard_Debit_Note_signed.xml'. A green button labeled 'Valid : true' is displayed. At the bottom right, there is a blue button labeled 'Retest'.

Web based validator - XML validation complete and no errors found





If not compliant, the following message is shown.



Web based validator - XML complete and errors found

The non-technical user is expected to share the validation outcomes with the Solution Developer to take necessary action.

2.3.8 Accessing the Integration Sandbox Page

The Integration Sandbox as covered in this user manual comprises of two components - the Sandbox specific front-end web pages (which is part of the Developer Portal and access to which requires a Developer Portal registered user account) and an API based Sandbox backend to integrate with.

A registered and logged in user can access the Integration Sandbox page from the main dashboard while a non-registered and non-logged in user is taken to the login screen. Once on the Integration Sandbox page the user is given a high level summary of the current version release of the Sandbox as well as links to any previous releases.

The ZATCA e-invoicing integration Sandbox is meant to be used for testing purposes only. Any inputs submitted on the Sandbox are not considered as acknowledged, approved or accepted by ZATCA. Taxpayers will be required to login using SSO credentials for the Taxation portal (ERAD) prior to officially be able to submit official documents. Test CSIDs provided by the Sandbox cannot be used in the Core E-invoicing Solution.





Developers must also take into account that documents or requests submitted on the Core E-invoicing Solution will be subjected to additional validations such as security features, prohibited functionalities, additional business rule validations and/or referential checks based such as validating Seller/Buyer information entered in the documents, validations based on previously submitted documents.

Home -> E-invoicing Portal -> Developer Portal -> API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

API Integration Sandbox

- Sandbox Release 2.1.1 (Latest version)
- Sandbox Release 2
- Sandbox Release 1.5
- Sandbox Release 1

The ZATCA e-invoicing API Integration Sandbox is meant to be used for testing purposes only. Any e-invoices or their associated notes submitted on the Sandbox are not considered as acknowledged, approved or accepted by ZATCA. Taxpayers will be required to register for e-invoicing on the ZATCA e-invoicing production system (FATOORA) in order to officially be able to submit their e-invoices to ZATCA.

Developers must also take into account that e-invoices, credit and debit notes submitted on the production system will be subjected to additional validations such as security features, prohibited functionalities and additional business rule validations as part of the Clearance process.

The following table provides a summary description of the APIs including the key outputs and inputs/pre-requisites for each API. Additional details can be obtained from the User Manual provided at the end.

#	API Name	Description	Output	Pre-requisites
---	----------	-------------	--------	----------------

API Integration sandbox landing page

On the left navigation bar of the page the user is able to access the links to the API documentation which are maintained as Swagger files (each API call is described in section 2.3.10 below along with the possible outcomes).





2.3.9 Accessing the API and associated Documentation (Swagger Files)

Access to the Swagger files is provided from the Integration Sandbox page. API documentation is provided covering all the API calls that can be tested on the Sandbox such as:

- Test request for Compliance CSID as part of a new onboarding (requires a signed test CSR to be submitted - details provided in the Swagger files)
- Test request for Production CSID as part of a new onboarding (requires a test Compliance CSID to be submitted)

Note: The Core E-invoicing Solution will require specific compliance checks to be completed in between the Compliance CSID and Production CSID requests and the latter will return an invalid response until these compliance checks are completed. This invalid response can be tested in the Sandbox by providing a specific input which is covered in the Swagger files below.

- Test request for a new Production CSID as part of renewal (requires a test Compliance CSID to be submitted)
- Test submission of documents for Clearance (requires a test Production CSID)
- Test submission of documents for Reporting (requires a test Production CSID)

Although the Sandbox uses test CSIDs, it is important to note that the VAT Registration number used to obtain the test CSID must match with the VAT Registration number in the Renewal CSR and/or e-invoices, credit notes, debit notes and QR codes submitted in all subsequent calls made using that specific test CSID. In other words for every VAT Registration Number that is used in the Sandbox integration, a separate CSID will have to be requested. Of course the VAT Registration Numbers can be dummy inputs.

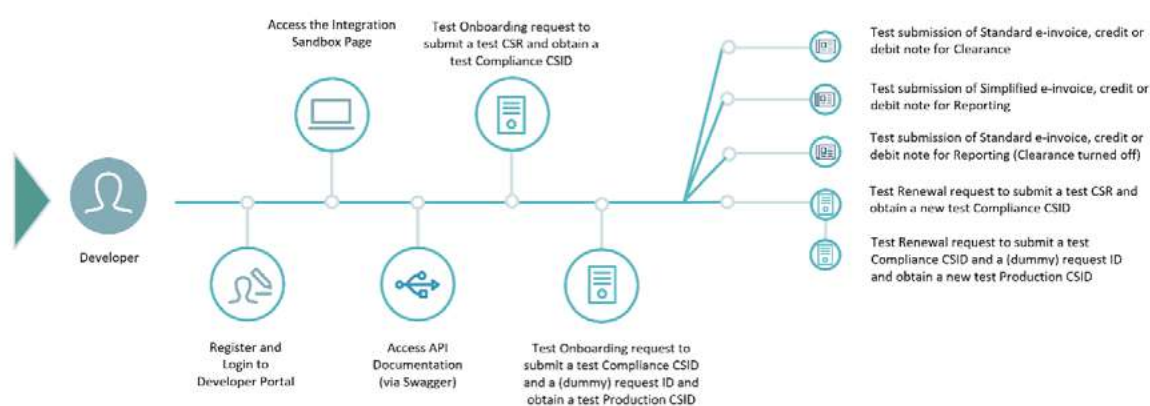
- Please refer to the API Documentation through the following [LINK](#).

Note: Please make sure to log-in in order to view the API documentation





2.3.10 Step by step guide to make a successful call to APIs



1. For Reporting and Clearance (testing the submission of E-invoices, credit and debit notes)

- The users' E-Invoice Generation Solution (EGS) needs to generate compliant XML documents. For more details on generating compliant XML documents please refer to the XML Implementation Standards and the Data Dictionary ([E-Invoice specifications \(zatca.gov.sa\)](https://zatca.gov.sa)). It is also recommend to test the compliance using the Compliance and Enablement Toolbox SDK ([Download SDK \(zatca.gov.sa\)](https://zatca.gov.sa)) or Portal based validator for non-technical users ([Compliance and Enablement Toolbox portal](https://zatca.gov.sa)).
- For Simplified documents (and optionally for Standard documents), the EGS also needs to generate compliant QR codes. For more details on generating compliant QR codes please refer to the Security Features and Implementation Standards ([E-Invoice Security Features and Implementation Standards](https://zatca.gov.sa)).





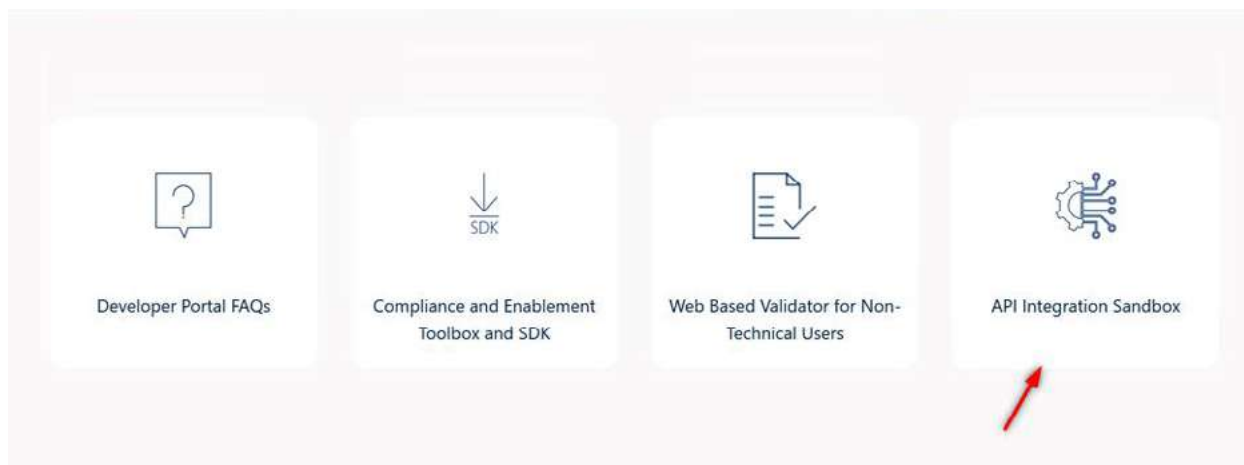
- Note that EGS must obtain a test Cryptographic Stamp Identifier (CSID) first, by using the test integration calls for Onboarding or Renewal.

2. For Cryptographic Stamp Identifier (testing the Onboarding and Renewal processes).

- The users' EGS needs to generate a compliant CSR to obtain a test CSID. For more details on generating a compliant CSR and CSID specifications please refer to ([E-Invoice Security Features and Implementation Standards](#)).
- Note that EGS must obtain a test Cryptographic Stamp Identifier (CSID) first, by using the test integration calls for Onboarding, in order to test the integration call for Renewal which requires a test CSID to be included in the request.

2.3.10.1 Compliance CSID

- Step 1: Navigate to Developer Portal link
- Step 2: Login with correct credentials
- Step 3: Navigate to API Integration Sandbox





- Step 4: Click on API documentation guides

Home → E-invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

Overview →

API Documentation guide

Reporting
Reporting API →

Clearance
Clearance API →

Compliance CSID
Compliance CSID API →

Compliance Checks
Compliance Invoice API →

Production CSID
Production CSID (Onboarding) API →
Production CSID (Renewal) API →

API Integration Sandbox

Sandbox Release 2.1.1 (Latest version) ✓
Sandbox Release 2 ✓
Sandbox Release 1.5 ✓
Sandbox Release 1 ✓

The ZATCA e-invoicing API Integration Sandbox is meant to be used for testing purposes only. Any e-invoices or their associated notes submitted on the Sandbox are not considered as acknowledged, approved or accepted by ZATCA. Taxpayers will be required to register for e-invoicing on the ZATCA e-invoicing production system (FATOORA) in order to officially be able to submit their e-invoices to ZATCA.

Developers must also take into account that e-invoices, credit and debit notes submitted on the production system will be subjected to additional validations such as security features, prohibited functionalities and additional business rule validations as part of the Clearance process.

The following table provides a summary description of the APIs including the key outputs and inputs/pre-requisites for each API. Additional details can be obtained from the User Manual provided at the end.

#	API Name	Description	Output	Pre-requisites
---	----------	-------------	--------	----------------

- Step 5: Click on Compliance CSID API drop-down

Home → E-invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

Overview →

API Documentation guide

Reporting
Reporting API →

Clearance
Clearance API →

Compliance CSID
Compliance CSID API →

Compliance Checks
Compliance Invoice API →

Production CSID
Production CSID (Onboarding) API →
Production CSID (Renewal) API →

e-Invoicing Sandbox Release (2.1.0) 1.0.0 GA

ZATCA wants to provide Taxpayers and Developers of Taxpayer e-invoicing solutions and devices the opportunity to test the integration of the systems with a ZATCA Sandbox environment prior to the launch of the production system. The Integration Sandbox (ISB) should enable solution developers to simulate the integration calls/requests that will be required later as part of the registration process and the submission of e-invoices, credit and debit notes to the production system. The Sandbox backend will accordingly simulate the validations and responses as part of the Cryptographic Stamp Identifiers issuance, renewal and revocation as well as the Reporting and Clearance function.

Although the ISB will give ZATCA an indication of the adoption rate for e-invoicing solutions in the market, it will not be mandatory to complete Sandbox testing as a pre-requisite for Registration/Taxpayer onboarding or accessing the production system. Similar to the Compliance and Enablement Toolbox (CET), the ISB is also aimed at Developers to build/update their solutions which are in line with ZATCA specifications and standards and are able to integrate with a ZATCA backend. Accordingly access to the ISB test/mock APIs will not be limited to Taxpayers and any user can register for a Developer account to access the ISB test/mock APIs and associated documentation. This registration will enable ZATCA to monitor the solution providers who intend to develop/update their solutions to integrate with ZATCA.

It should be noted that although the ISB will simulate most of the core functionalities of the production system, any validations that require integrations/access with external systems and/or storage as well as scenarios involving any backend exceptional handling (for example overriding the clearance process) will not be part of the ISB and will be covered by the core solution. Accordingly the ISB should not be considered as representative of all integrations and/or APIs that will be part of the production system.

This swagger documents the set of apis for the Sandbox (ISB) solution.

Developers can also refer to section 2.3.10 of the Developer Portal User Manual for additional guidance and steps.

Servers
https://gw-epic-gov.gat.gov.sa/e-invoicing/developer-portal

Compliance CSID (Certificate)

POST /compliance Issues an XSD Compliance Cryptographic Stamp Identifier (CCSID/Certificate) (CSID) based on submitted CSR.





- Step 4: Click on API documentation guide

Home → E-Invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

Overview →

API Documentation guide

Reporting

Reporting API →

Clearance

Clearance API →

Compliance CSID

Compliance CSID API →

Compliance Checks

Compliance Invoice API →

Production CSID

Production CSID (Onboarding) API →

Production CSID (Renewal) API →

API Integration Sandbox

Sandbox Release 2.1.1 (Latest version) ✓

Sandbox Release 2 ✓

Sandbox Release 1.5 ✓

Sandbox Release 1 ✓

The ZATCA e-invoicing API Integration Sandbox is meant to be used for testing purposes only. Any e-invoices or their associated notes submitted on the Sandbox are not considered as acknowledged, approved or accepted by ZATCA. Taxpayers will be required to register for e-invoicing on the ZATCA e-invoicing production system (FATOORA) in order to officially be able to submit their e-invoices to ZATCA.

Developers must also take into account that e-invoices, credit and debit notes submitted on the production system will be subjected to additional validations such as security features, prohibited functionalities and additional business rule validations as part of the Clearance process.

The following table provides a summary description of the APIs including the key outputs and inputs/pre-requisites for each API. Additional details can be obtained from the User Manual provided at the end.

#	API Name	Description	Output	Pre-requisites
---	----------	-------------	--------	----------------

- Step 5: Click on Authorize Compliance Invoice API

Home → E-Invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

Overview →

API Documentation guide

Reporting

Reporting API →

Clearance

Clearance API →

Compliance CSID

Compliance CSID API →

Compliance Checks

Compliance Invoice API →

Production CSID

Production CSID (Onboarding) API →

Production CSID (Renewal) API →

e-Invoicing Sandbox Release (2.1.0) ^{1.0.0} ^{CRS0}

ZATCA wants to provide Taxpayers and Developers of Taxpayer e-invoicing solutions and devices the opportunity to test the integration of the systems with a ZATCA Sandbox environment prior to the launch of the production system. The Integration Sandbox (ISB) should enable solution developers to simulate the integration calls/requests that will be required later as part of the registration process and the submission of e-invoices, credit and debit notes to the production system. The Sandbox backend will accordingly simulate the validations and responses as part of the Cryptographic Stamp Identifiers issuance, renewal and revocation as well as the Reporting and Clearance function.

Although the ISB will give ZATCA an indication of the adoption rate for e-invoicing solutions in the market, it will not be mandatory to complete Sandbox testing as a pre-requisite for Registration/Taxpayer onboarding or accessing the production system. Similar to the Compliance and Enablement Toolbox (CET), the ISB is also aimed at Developers to build/update their solutions which are in line with ZATCA specifications and standards and are able to integrate with a ZATCA backend. Accordingly access to the ISB testmock APIs will not be limited to Taxpayers and any user can register for a Developer account to access the ISB testmock APIs and associated documentation. This registration will enable ZATCA to monitor the solution providers who intend to develop/update their solutions to integrate with ZATCA.

It should be noted that although the ISB will simulate most of the core functionalities of the production system, any validations that require integrations/access with external systems and/or storage as well as scenarios involving any backend exceptional handling (for example overriding the clearance process) will not be part of the ISB and will be covered by the core solution. Accordingly the ISB should not be considered as representative of all integrations and/or APIs that will be part of the production system.

This swagger documents the set of apis for the Sandbox (ISB) solution.

Developers can also refer to section 2.3.10 of the Developer Portal User Manual for additional guidance and steps.

Servers: Authorize

Compliance Invoice

POST /compliance/Invoices It performs compliance checks on invoice documents

Note: This step is to be repeated on the number of invoices to be sent as part of the compliance checks.





- Step 6:

For the Sandbox, use the sample dummy Username and Password provided to you on the Authorization screen.

For Production, run the Compliance CSID API to obtain the “binarySecurityToken” to be used as the Username and “secret” as the Password.

Please refer to Chapter 3 of this document for further details.

Compliance Invoice API version 1

Compliance Invoice API version 2

Available authorizations

hmiREj0TWpNM0ZETXRNV5TVI4d0hRWUIdWkItaVpQeUxHUUJBUXdQTXpBd01EYzFOVQ
c0TnpBd01EQXpNUTB3Q3dZRFZRUI1EQVF4TVRBd01SRXdEd1IEVFRYURBaGFZWfJqWV
NBeE1qRVINQIHQTFVRUR3d1BSbTQWkNCQ2RYTnphVzVsYzNNek1CMEdBMMVvKRGdRV0J
CU2dtSvdeNnuJQZnUjS2tVHdPSUjYdkiSDIakFmQmdOVkhTTUvHREFXZ0JSMILjejdCcUInzV
jFjMWSyK2FyS2NybvRXMUx6Qk9CZ05WSFI4RVJ6QkZNRU9hUWFB12hgmW0kSFJ3T2k4dm
RITjBZM0pzTG5waGRhTmhmMbWR2ZGk1eITOURaWEowUlc1eWlyeHhNMVJUV2tWSIRsWIB
TVU5GTFZOMVlrTkjMVEV1WTNk01JR3RCZ2dyQmdFRk1RY0JBUNNCb0RDQm5UQnVCZ2
dyQmdFRk1RY3d3BvWppYUhsMGNEb3ZMM1J6ZE0eWJDNtZZWFJqWVMttmlzWkVjMkV2
UTJWeWRFVvVjbtzYkM5VVUxcEZhVzUyYjJsaipWTkRRVEV1WthoMFoyRjZkQzVuYjNzdWJH
OWpZV3hmVhZ0YVJvBE9Waz9KUTBVdFUzVmIRMEV0TVNneEITNWpjbF3S3dZSUk3WUJCU
VWtUFRGgyaDBkSEE2THk5MGmZUmpjbXd1ZW1GMFkyRXVamJkyTG5OaEwyOWpM0F3R
GdZRFZSMFBBUUGvQkFRREFnZUFNQ8BHQTFVZEPRUvVdNQIFHQ0NzR0FRVUZZC01DQmd
ndUlnRUZCUWNEQXpBbkjha3JCZ0VFQVUM0ZRB0VhKfZTUFvR0NDc0dBVVQndNQ01
Bb0dDQ3NHQVFRk1JTURNQW9HQ0NwR1NNNDICQU1DQTBrcQU1FWUNJUUUNWd0RNY3E
2UE8rTWnrc0JYVXovdFhZGhHcDdyCVNhmKf4VE1TdjgzOEBSWhBT0JOREJ0CSszRFNsaW
pwVmZ4enJkRGg1MjhxQzZM3c21FZG9HV1ZyU3BHMQ==

Password: QydVsQAqTeflBK1Ujw7wBhzcofz1HfmuZ0kWB3B8=

Username:

Password:

Authorize Close

Compliance Invoice





- Step 7: Click on Compliance Invoice API drop-down

Home → E-invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

Overview →

API Documentation guide

- Reporting
 - Reporting API →
- Clearance
 - Clearance API →
- Compliance CSID
 - Compliance CSID API →
- Compliance Checks
 - Compliance Invoice API →
- Production CSID
 - Production CSID (Onboarding) API →
 - Production CSID (Renewal) API →

e-Invoicing Sandbox Release (2.1.0) 1.0.0 Changelog

ZATCA wants to provide Taxpayers and Developers of Taxpayer e-invoicing solutions and devices the opportunity to test the integration of the systems with a ZATCA Sandbox environment prior to the launch of the production system. The Integration Sandbox (ISB) should enable solution developers to simulate the integration calls/requests that will be required later as part of the registration process and the submission of e-invoices, credit and debit notes to the production system. The Sandbox backend will accordingly simulate the validations and responses as part of the Cryptographic Stamp Identifiers issuance, renewal and revocation as well as the Reporting and Clearance function.

Although the ISB will give ZATCA an indication of the adoption rate for e-invoicing solutions in the market, it will not be mandatory to complete Sandbox testing as a pre-requisite for Registration/Taxpayer onboarding or accessing the production system. Similar to the Compliance and Enablement Toolbox (CET), the ISB is also aimed at Developers to build/update their solutions which are in line with ZATCA specifications and standards and are able to integrate with a ZATCA backend. Accordingly access to the ISB test/mock APIs will not be limited to Taxpayers and any user can register for a Developer account to access the ISB test/mock APIs and associated documentation. This registration will enable ZATCA to monitor the solution providers who intend to develop/update their solutions to integrate with ZATCA.

It should be noted that although the ISB will simulate most of the core functionalities of the production system, any validations that require integrations/access with external systems and/or storage as well as scenarios involving any backend exceptional handling (for example overriding the clearance process) will not be part of the ISB and will be covered by the core solution. Accordingly the ISB should not be considered as representative of all integrations and/or APIs that will be part of the production system.

This swagger documents the set of apis for the Sandbox (ISB) solution.

Developers can also refer to section 2.3.10 of the Developer Portal User Manual for additional guidance and steps.

Servers
 Authorize

Compliance Invoice

POST /compliance/invoices If performs compliance checks on invoice documents





- Step 8: Click on Try it out button

Parameters

Try it out

Name	Description
Authorization string (header)	<p>Username (Compliance Certificate) and Password (Secret) should be taken from the Compliance certificate response and put in the Authorize popup :</p> <ul style="list-style-type: none">• Username: 'binarySecurityToken' which represnts the Compliance Certificate• Password: 'secret' <div>Authorization</div>
Accept-Language string (header)	<p>Specifies the languague in which the response will be returned. Currently supported languages are English (en) and Arabic (ar) and it defaults to English.</p> <p>Examples:</p> <div>English</div> <div>en</div>
Accept-Version * required string (header)	<p>Example : V2</p> <div>V2</div>

Note: V2 refers to the Version of the APIs used and should be mentioned in the API calls (V2 is currently the only valid version).





2.3.10.3 Production CSID (Onboarding) API

- Step 1: Navigate to Developer Portal link
- Step 2: Login with correct credentials
- Step 3: Navigate to API Integration Sandbox



- Step 4: Click on API documentation guide

Home → E-Invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

Overview →

API Documentation guide

Reporting

Reporting API →

Clearance

Clearance API →

Compliance CSID

Compliance CSID API →

Compliance Checks

Compliance Invoice API →

Production CSID

Production CSID (Onboarding) API →

Production CSID (Renewal) API →

API Integration Sandbox

Sandbox Release 2.1.1 (Latest version) ✓

Sandbox Release 2 ✓

Sandbox Release 1.5 ✓

Sandbox Release 1 ✓

The ZATCA e-invoicing API Integration Sandbox is meant to be used for testing purposes only. Any e-invoices or their associated notes submitted on the Sandbox are not considered as acknowledged, approved or accepted by ZATCA. Taxpayers will be required to register for e-invoicing on the ZATCA e-invoicing production system (FATOORA) in order to officially be able to submit their e-invoices to ZATCA.

Developers must also take into account that e-invoices, credit and debit notes submitted on the production system will be subjected to additional validations such as security features, prohibited functionalities and additional business rule validations as part of the Clearance process.

The following table provides a summary description of the APIs including the key outputs and inputs/pre-requisites for each API. Additional details can be obtained from the User Manual provided at the end.

#	API Name	Description	Output	Pre-requisites
---	----------	-------------	--------	----------------





- Step 5: Click on Authorize Production CSID (Onboarding) API

Home → E-Invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

Overview →

API Documentation guide

- Reporting
 - Reporting API →
- Clearance
 - Clearance API →
- Compliance CSID
 - Compliance CSID API →
- Compliance Checks
 - Compliance Invoice API →
- Production CSID
 - Production CSID (Onboarding) API →
 - Production CSID (Renewal) API →

e-Invoicing Sandbox Release (2.1.0) 4.0.0 CSMS

ZATCA wants to provide Taxpayers and Developers of Taxpayer e-invoicing solutions and devices the opportunity to test the integration of the systems with a ZATCA Sandbox environment prior to the launch of the production system. The Integration Sandbox (ISB) should enable solution developers to simulate the integration calls/requests that will be required later as part of the registration process and the submission of e-invoices, credit and debit notes to the production system. The Sandbox backend will accordingly simulate the validations and responses as part of the Cryptographic Stamp Identifiers issuance, renewal and revocation as well as the Reporting and Clearance function.

Although the ISB will give ZATCA an indication of the adoption rate for e-invoicing solutions in the market, it will not be mandatory to complete Sandbox testing as a pre-requisite for Registration/Taxpayer onboarding or accessing the production system. Similar to the Compliance and Enablement Toolbox (CET), the ISB is also aimed at Developers to build/update their solutions which are in line with ZATCA specifications and standards and are able to integrate with a ZATCA backend. Accordingly access to the ISB test/mock APIs will not be limited to Taxpayers and any user can register for a Developer account to access the ISB test/mock APIs and associated documentation. This registration will enable ZATCA to monitor the solution providers who intend to develop/update their solutions to integrate with ZATCA.

It should be noted that although the ISB will simulate most of the core functionalities of the production system, any validations that require integrations/access with external systems and/or storage as well as scenarios involving any backend exceptional handling (for example overriding the clearance process) will not be part of the ISB and will be covered by the core solution. Accordingly the ISB should not be considered as representative of all integrations and/or APIs that will be part of the production system.

This swagger documents the set of apis for the Sandbox (ISB) solution.

Developers can also refer to section 2.3.10 of the Developer Portal User Manual for additional guidance and steps.

Servers:
 Authorize

Cryptographic Stamp Identifier (Certificate) Endpoint(s)

POST /production/csids Issues an XSD Production Cryptographic Stamp Identifier (PCSID/Certificate) (CSID) based on submitted CSR.





- Step 6:

For the Sandbox, use the sample dummy Username and Password provided to you on the Authorization screen.

For Production, run the Compliance CSID API to obtain the “binarySecurityToken” to be used as the Username and “secret” as the Password.

Please refer to Chapter 3 of this document for further details.

Available authorizations

BasicAuth (http, Basic)

Username: TUUQhEQ0N6WmFrQXduQkFmSudBWUZ1TzBaRk1Bb0dDQ3FHU000OUJBtUNNQV4RXpB UkJnTIZCQU1NQ21WsmJuWnZhv05wYm1jd0hcY05Nak3TnpFMk1qQXpOak8V2hT1qY3dO akUyTURBd01EQXdXakJQTVFzd0NRWURWUUVFHRXdKVFfURVhNQVhQTfVRUN3d09ZvZf 0WVc0Z1FuSmhibU5vWTJneEV6QVJCZ05WQkFvTUNtaGhV0VnZVdGbklETXhFakFRQmdOV kjBTU1DVEV5Tnk0d0xqQXVNVJEJXTUJBR0J5cUdTTTQ6QWdFR0JtdUJJCQUFLQTBjQUJJCd U10aWYySy84NndRvdVdys4VnhIRWpIZTIZVFdMQzJLVTFpNVhNzNU2NNQ3IGdms0V3d0Z08 aRdRRC59FdXpqNno2ddEeShwU0NtaVR4ZWpWm93Z1pjd0RBWURWUjBUQVFIL0JB5Xd8 RENCAGdZRFZSMFJCSDh3ZmFmSN011a3hHekFaQmdOVkUjBUU1FakV0YUdGNVYd3IMVEBT kh3ekxUTTF0REVMtUWwR0NnbVNKb21UOGI4a0FRRU1Eek14TURFM05UTTV0eIF3TURBd0 16RU5NQXNHQTfVRURBd0VNVEV3TURFUU1BNEdBMVVF2d3SFdRjBZMkVnTXpFWU1C WUdBMVVFRRhd3UfJhOXZaQ0JDZFRhOmFXNWxjM016TUFRNDcUdTTTQ5QkFNQ0EwZ0F NRVVDSUVheUg5UGRlVUWU3B2dFhnb0JSMdZPSjph5aU1BWVVFQnFMNmZyM01LNHZAu VBL2JBaENIK2NXV21uchHUYay6vNkY3bXk2ZDhXWENib3g5TjRlbj5VEXMbz0=

Password: LdustC+/JHbOBZno8HDeOzBk8ON4wmixWFBNGKQNI8=

Username:

Password:

Authorize **Close**

Cryptographic Stamp Identifier (Certificate) Endpoint(s)

POST /production/csid/ Access to X509 Production Cryptographic Stamp Identifier (PCSID) Certificates (CSID) issued on embedded





- Step 7: Click on Production CSID (Onboarding) API drop-down

Home → E-invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

Overview →

e-Invoicing Sandbox Release (2.1.0) 1.0.0 OKSD

ZATCA wants to provide Taxpayers and Developers of Taxpayer e-invoicing solutions and devices the opportunity to test the integration of the systems with a ZATCA Sandbox environment prior to the launch of the production system. The Integration Sandbox (ISB) should enable solution developers to simulate the integration calls/requests that will be required later as part of the registration process and the submission of e-invoices, credit and debit notes to the production system. The Sandbox backend will accordingly simulate the validations and responses as part of the Cryptographic Stamp Identifiers issuance, renewal and revocation as well as the Reporting and Clearance function.

Although the ISB will give ZATCA an indication of the adoption rate for e-invoicing solutions in the market, it will not be mandatory to complete Sandbox testing as a pre-requisite for Registration/Taxpayer onboarding or accessing the production system. Similar to the Compliance and Enablement Toolbox (CET), the ISB is also aimed at Developers to build/update their solutions which are in line with ZATCA specifications and standards and are able to integrate with a ZATCA backend. Accordingly access to the ISB test/mock APIs will not be limited to Taxpayers and any user can register for a Developer account to access the ISB test/mock APIs and associated documentation. This registration will enable ZATCA to monitor the solution providers who intend to develop/update their solutions to integrate with ZATCA.

It should be noted that although the ISB will simulate most of the core functionalities of the production system, any validations that require integrations/access with external systems and/or storage as well as scenarios involving any backend exceptional handling (for example overriding the clearance process) will not be part of the ISB and will be covered by the core solution. Accordingly the ISB should not be considered as representative of all integrations and/or APIs that will be part of the production system.

This swagger documents the set of APIs for the Sandbox (ISB) solution.

Developers can also refer to section 2.3.10 of the Developer Portal User Manual for additional guidance and steps.

Servers
 Authorize

Cryptographic Stamp Identifier (Certificate) Endpoint(s)

POST /production/csids Issues an X509 Production Cryptographic Stamp Identifier (PCSID/Certificate) (CSID) based on submitted CSR.

- Step 8: Click on Try it now button

Compliance Checks
Compliance Invoice API →

Production CSID
Production CSID (Onboarding) API →
Production CSID (Renewal) API →

Servers
 Authorize

Cryptographic Stamp Identifier (Certificate) Endpoint(s)

POST /production/csids Issues an X509 Production Cryptographic Stamp Identifier (PCSID/Certificate) (CSID) based on submitted CSR.

This Production CSID is a simulation of ZATCA rootCA moreover it is used to sign invoice documents and authenticate invoicing api calls. Specifically, it is sent via the authentication header for those api calls. This Production CSID is a simulation of ZATCA rootCA moreover it is used to sign invoice documents and authenticate invoicing api calls. Specifically, it is sent via the authentication header for those api calls.

Parameters Try it out

Name	Description
Authorization string (header)	Username (Compliance Certificate) and Password (Secret) should be taken from the Compliance certificate response and put in the Authorize popup: <ul style="list-style-type: none">Username: 'binarySecurityToken' which represents the Compliance CertificatePassword: 'secret'
Accept-Version string (header)	V2

Note: V2 refers to the Version of the APIs used and should be mentioned in the API calls (V2 is currently the only valid version).





- Step 9: Fill the request body (compliance request ID)
- Step 10: Click on Execute button

Request body

application/json

```
{  
  "compliance_request_id": "1234567890123"  
}
```

Execute





- Result (200)

Code

Details

200

Response body

```
{
  "requestID": "30368",
  "tokenType": "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3",
  "dispositionMessage": "ISSUED",
  "binarySecurityToken": "TU13RD36QBNBNENQXDJQKfNSV81d0F8ZHFEBUlocXNqC001Q3d8QkFRQj7vREFLQmdncmhrak9QVFEQpCak1SVXdfdl1lQ1p3bWl0aUhlM2I2FCR1J2RmJHdWpZV3d4RXP8
UK3nb0pamFKay91c1p8RUp6Z05uYjNZeE76QVZCZ3d0ka1lHsmevSXNaQVNaRmdkb0VlUmsZdM4wTV33d0dMUBMUVFERXDOVUxcEITVTXVQOBaRFJTWmKvQpEUNWweE1CNFhEVE1STURNu9ERTFORF16TV
xvNERUSXlN8E16TURFmP5N9rGmb093VFRFTE1ba0d8MVVFQbNqLUwaXhaaFQgWd0V3Bb1RCVXBvY21eU5b1d8QVlEVlFRTEV4RktaVlJmWd0Z1F0Seh1bUSVTVR7ekSESVNMqkFhQTVVRUF4TupNYek2
T0pBdU1DNHNR113RUFZStvMl1aJ00QVFRKs0RUVBQW9EUMd8RUQvdZTyb0hcDk3Jqzh0bsada91bzZPe1J5bXltVlOV13oSKlhtWmHUKVCQ0vaQJRFQVzyqVWmMhYaXhZNHFCMYSZGRlcnprV21Ed2
RvM0laSgdXT0NBaW03Z2d3bU1JReXCZ05MSF3FRmdZTXd0mUNRZmpCOE1Sd3dH21lEVlFRURCTXlNak15T0pNeU5EUTBNe1fEYUclabUSETXlMujh35FFZS0NaSWlplB5TEdRqkFRd1BnekY3TVRjMU16azN0
REF3TURBek1RmHd0d1lEVlFRURBUXNREV4TV3F3d0R3MURMUVTRERFoVfL0Kkd1R1Vnu1RfWk1CYed0MwVFRhd3UVUyRnkjR3haSUVK0W0qTnb1bV26Y3p0ZC7nTlZ1UTFRmRdRvW0Xy3N1YkpeakQ1KdPa3
dCSuXDK3d0VazLW0dZld1EVlWakJCZ3d0b0FV2G1DTSt3YndYR2RYTlozUG1xENSUNisxdFM4d1RmMURWUjBmQVjd1JUQkRvRUdnU0RZOWFIUjBj6G92T0NSemBhtn1lQ1u2WVhsall1TWMSlM11vzJfD1Ey
Vn1kRVZ1Y205c23DOVVVVK0GU1U1V1Qb0ERSUZFUFZdKRFfTmHhMBU5SYKRODQJRMulldl1lCQlRVSEFRARVnVUF3Z1owd2JmKUl1dl1lCQlRVSE1BR0dZbWguzEh8Nkxs0T8Jm1qY213dWvtrJ0ZMkV1MjISMk
xuTmHMMESY25SRmJUsnZ1R3d2V6Z0YVJ3bHvkbTlWMTJWVFw0XhMBVY0ZEkaGVuUxVamJkyT014d1kyRnMYM3UJv2tHs1Rsk18TVU5GTFZQWFIrTk3MVEVTVNRdvkz5JBNQ3NHQ0NZR0FRVUZCEkFCaG5
bZRIundPaThZ2EH0PfkZ5NMB0NB0ZE0aE0xtZkZk1V6WYH0d1kZ1ndNQTRHQfFVZER3RU1vd1FFQxd3S0dQW0CZ05W5FVVRUZQVVCZ2dyQmdFRk3RYERBZ11353dzQk3RVU0Bd013SndZskT3WUJCQud0Tn
hV501Cb3dHREFLQmdncK3nRUZCUMNEQq0807nZ3JCZ0V0QlFjREF6QUtCZ2dxaGtqT1BRU0R8Z05KQUR0pRUF5TahSV1EzYk5sTEZkT184cVlUml3WUVRXZ25Lmud0MESIZGNTHRQZkMwQ0lRQ1NBd0hY
dnY3dGV0VUw2QVdQcDhCeG5MTE13ZX34MhCbmV3by9nRjNfSkE9PQ==",
  "secret": "f9YRhopH/G7x0TECOY6nK5CHLNY1b5r1AHSFPIC04qn="
}
```

Response headers

```
cache-control: no-store,max-age=0
content-type: application/json
expires: 0
pragma: no-cache
x-ratelimit-limit: name=default,100;
x-ratelimit-remaining: name=default,84;
```





2.3.10.4 Production CSID (Renewal) API

- Step 1: Navigate to Developer Portal link
- Step 2: Login with correct credentials
- Step 3: Navigate to API Integration Sandbox



- Step 4: Click on API documentation guide

Home → E-Invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

Overview →

API Documentation guide →

Reporting

Reporting API →

Clearance

Clearance API →

Compliance CSID

Compliance CSID API →

Compliance Checks

Compliance Invoice API →

Production CSID

Production CSID (Onboarding) API →

Production CSID (Renewal) API →

API Integration Sandbox

Sandbox Release 2.1.1 (Latest version) ✓

Sandbox Release 2 ✓

Sandbox Release 1.5 ✓

Sandbox Release 1 ✓

The ZATCA e-invoicing API Integration Sandbox is meant to be used for testing purposes only. Any e-invoices or their associated notes submitted on the Sandbox are not considered as acknowledged, approved or accepted by ZATCA. Taxpayers will be required to register for e-invoicing on the ZATCA e-invoicing production system (FATOORA) in order to officially be able to submit their e-invoices to ZATCA.

Developers must also take into account that e-invoices, credit and debit notes submitted on the production system will be subjected to additional validations such as security features, prohibited functionalities and additional business rule validations as part of the Clearance process.

The following table provides a summary description of the APIs including the key outputs and inputs/pre-requisites for each API. Additional details can be obtained from the User Manual provided at the end.

#	API Name	Description	Output	Pre-requisites
---	----------	-------------	--------	----------------





- Step 5: Click on Authorize Production CSID (Renewal) API

Home → E-Invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

Overview →

API Documentation guide

- Reporting
 - Reporting API →
- Clearance
 - Clearance API →
- Compliance CSID
 - Compliance CSID API →
- Compliance Checks
 - Compliance Invoice API →
- Production CSID
 - Production CSID (Onboarding) API →
 - Production CSID (Renewal) API →

e-Invoicing Sandbox Release (2.1.0) 1.0.0 OAS3

ZATCA wants to provide Taxpayers and Developers of Taxpayer e-invoicing solutions and devices the opportunity to test the integration of the systems with a ZATCA Sandbox environment prior to the launch of the production system. The Integration Sandbox (ISB) should enable solution developers to simulate the integration calls/requests that will be required later as part of the registration process and the submission of e-invoices, credit and debit notes to the production system. The Sandbox backend will accordingly simulate the validations and responses as part of the Cryptographic Stamp Identifiers issuance, renewal and revocation as well as the Reporting and Clearance function.

Although the ISB will give ZATCA an indication of the adoption rate for e-invoicing solutions in the market, it will not be mandatory to complete Sandbox testing as a pre-requisite for Registration/Taxpayer onboarding or accessing the production system. Similar to the Compliance and Enablement Toolbox (CET), the ISB is also aimed at Developers to build/update their solutions which are in line with ZATCA specifications and standards and are able to integrate with a ZATCA backend. Accordingly access to the ISB test/mock APIs will not be limited to Taxpayers and any user can register for a Developer account to access the ISB test/mock APIs and associated documentation. This registration will enable ZATCA to monitor the solution providers who intend to develop/update their solutions to integrate with ZATCA.

It should be noted that although the ISB will simulate most of the core functionalities of the production system, any validations that require integrations/access with external systems and/or storage as well as scenarios involving any backend exceptional handling (for example overriding the clearance process) will not be part of the ISB and will be covered by the core solution. Accordingly the ISB should not be considered as representative of all integrations and/or APIs that will be part of the production system.

This swagger documents the set of apis for the Sandbox (ISB) solution.

Developers can also refer to section 2.3.10 of the Developer Portal User Manual for additional guidance and steps.

Servers
 Authorize

Cryptographic Stamp Identifier (Certificate) Endpoint(s)

PATCH	/production/csids	Renews an X509 Certificate (CSID) based on submitted CSPL.
-------	-------------------	--





- Step 6:

For the Sandbox, use the sample dummy Username and Password provided to you on the Authorization screen.

For Production, run the Compliance CSID API to obtain the “binarySecurityToken” to be used as the Username and “secret” as the Password.

Please refer to Chapter 3 of this document for further details.

Production CSID (Renewal) API version 1

Production CSID (Renewal) API version 2

Available authorizations

nmRE0TWpNMGZETXRNVESVTVi4d0hRVVUOWMtaVpQeUxHUUUBUXdQTXpB901EYzFOVG
c0TnpB001EQXpNUTB3Q3dZRFZRUU1EQVF4TVRBd01SRXdEd1EVFRYURBaGFZWFJqWV
N8eE1qRVINQIHQTfVRUR3d1BSbT2WkNCQ2RYTnphVzVsYzNNk1CMEdBMVVKRGdRV0J
CU2dSvdENmJQZm3S2tVhdPSUJYdksQ0IakFmQmdOVkhTTUvHREFXZ0JSMUjejdCcU/NzV
fJlMWSjK2FyS2Ny6VRXOMUx6Qk9CZ05WSFI4RVj6QkZNRU9mUWFBZ2hglW9k3fJ3T2k4dm
RtTjBZM0pzTQ5waGRHTmMhWWR2ZGk1eITOUraWEowUlcfeWyeHNMMVJUV2fWSIRaWB
TVUSGTFZOMVITkUMVEV1WTNKc01JR3RCZ2dyQmdFRkIRY0UBUUVNCb0RDQm5UQnVCZ2
dyQmdFRkIRY3dBWVppYUhsMGNeb3ZMM1J6ZEdeWJDNZ2ZWfJqWVM1bmizV0KVjMkV2
UTJWwWRFVnVj5TlzYkM5VVUxcEZnVzUyYjJsaipWTKRREVEV1WthoMfOyRjZkQZVUYjNZdWUjH
OWpZV3hmVXZ0YVJv6E9WazKUTBVdFlizVmIRMEV0TVNneEITNWppcIF3S3dZSU3WUJCU
VWITUFHR0gyaDBkSEEE2Thk5MGMZumpgXd1ZW1GMFkyRXVIMkyTGS0aEwyOWpM0F3R
GdZRFZSMFBBUUgvcQFRREFEZUFNQjBHQTfVZEgRUVdNQIFHQ0NzR0FRVUZCd01DQmd
ndkLnRUZCUWNEQXpB8kUna3JCZ0VFQVUM0ZRB0VhskFZTUfVr0NDc0dBuVVVGQndNQ01
Bb0dDQ3NHQVFVRkU3TURNQW9HQ0NkR1NNNDICQU1DQTBkQU1FWUNJUUUNWdORNY3E
3UE8rTWntc0JYVXovqFHZGhHcDdyCVNhmKf4VEITdgzOEIBSWNBtGUOREJ00SszRfNsaw
pvVmZ4enUR9g1MjRkXQZM3c21FZG5Hv1ZyU3BHMQ==

Password: Xij15LyMCgSC66QbnEOiqVPnSbs3kDTjWnGneYhtSs=

Username:

Password:

[Authorize](#) [Close](#)

Cryptographic Stamp Identifier (Certificate) Endpoint(s)





- Step 7: Click on Production CSID (renewal) API drop-down

Home → E-invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

Overview →

e-Invoicing Sandbox Release (2.1.0) 1.0.0 OAS3

ZATCA wants to provide Taxpayers and Developers of Taxpayer e-invoicing solutions and devices the opportunity to test the integration of the systems with a ZATCA Sandbox environment prior to the launch of the production system. The Integration Sandbox (ISB) should enable solution developers to simulate the integration calls/requests that will be required later as part of the registration process and the submission of e-invoices, credit and debit notes to the production system. The Sandbox backend will accordingly simulate the validations and responses as part of the Cryptographic Stamp Identifiers issuance, renewal and revocation as well as the Reporting and Clearance function.

Although the ISB will give ZATCA an indication of the adoption rate for e-invoicing solutions in the market, it will not be mandatory to complete Sandbox testing as a pre-requisite for Registration/Taxpayer onboarding or accessing the production system. Similar to the Compliance and Enablement Toolbox (CET), the ISB is also aimed at Developers to build/update their solutions which are in line with ZATCA specifications and standards and are able to integrate with a ZATCA backend. Accordingly access to the ISB test/mock APIs will not be limited to Taxpayers and any user can register for a Developer account to access the ISB test/mock APIs and associated documentation. This registration will enable ZATCA to monitor the solution providers who intend to develop/update their solutions to integrate with ZATCA.

It should be noted that although the ISB will simulate most of the core functionalities of the production system, any validations that require integrations/access with external systems and/or storage as well as scenarios involving any backend exceptional handling (for example overriding the clearance process) will not be part of the ISB and will be covered by the core solution. Accordingly the ISB should not be considered as representative of all integrations and/or APIs that will be part of the production system.

This swagger documents the set of apis for the Sandbox (ISB) solution.

Developers can also refer to section 2.3.10 of the Developer Portal User Manual for additional guidance and steps.

Servers
 Authorize

Cryptographic Stamp Identifier (Certificate) Endpoint(s)

PATCH /production/csids Renews an X509 Certificate (CSID) based on submitted CSR.

- Step 8: Click on Try it now

Compliance CSID API →

Compliance Checks
Compliance Invoice API →

Production CSID
Production CSID (Onboarding) API →
Production CSID (Renewal) API →

Developers can also refer to section 2.3.10 of the Developer Portal User Manual for additional guidance and steps.

Servers

Cryptographic Stamp Identifier (Certificate) Endpoint(s)

PATCH /production/csids Renews an X509 Certificate (CSID) based on submitted CSR.

Renews an X509 Certificate (CSID) based on submitted CSR

Parameters Try it out

Name	Description
OTP <small>required</small> string (body)	One time password generated from Fatoora portal Examples: <div>Invalid OTP</div> <div>111111</div>
accept-language string (header)	Specifies the language in which the response will be returned. Currently supported languages are English (en) and Arabic (ar) and it defaults to English. Examples: <div>English</div> <div>ar</div>





- Step 9: Insert valid OTP

Compliance Checks

- Compliance Invoice API →

Production CSID

- Production CSID (Onboarding) API →
- Production CSID (Renewal) API →

Servers

https://gw-epic-gov.gat.gov.sole-invoicing/developer-portal Authorize

Cryptographic Stamp Identifier (Certificate) Endpoint(s)

PATCH /production/csids Renews an X509 Certificate (CSID) based on submitted CSR

Renews an X509 Certificate (CSID) based on submitted CSR

Parameters Try it out

Name	Description
OTP * <small>required</small>	One time password generated from Fatoora portal
string	
(header)	
Examples:	
Invalid OTP	
111111	
accept-language	Specifies the language in which the response will be returned. Currently supported languages are English (en) and Arabic (ar) and it defaults to English
string	
(header)	
Examples:	
English	
en	
Accept-Version * <small>required</small>	V2
string	
(header)	

Note: V2 refers to the Version of the APIs used and should be mentioned in the API calls (V2 is currently the only valid version).





- Step 10: Fill the request body (CSR)
- Step 11: Click on Execute button

Request body required

application/json

CSR Request in body as Base64.

```
{
  "csr": "LS0tLS1CRUdJTS1BRDVJUSUZ3QeFURS5SRVFRVNL5etLS9Nck13SUNEakNDQ0VQeFRQXdkkVHTUFR8EeXVUVcaE1DVTBFeEh6QURCZ05wQkFzTUZrRmp1V1VnVj7sa1oyVjANCnc2TEhNtutay3lCTV7FUXhIekFkQmdov
k3Bb01Ga0ZqY1dV11yYb0taH1Yudz7Nq2d51pJeuJNVKVR8EppqusNckJnT1L2CQ1NSF25VFZDMDRPRFkuTXpFeE5EIOXRnekF3TV9jeE5UMTVPFVF3TURBek1GUXdFQV1IS29a5XpqlEMNCKFRUJZLNEVFQURFRFFnQUV1d0N
NchBaN19ta0dnazMONULRRwt3amSVr05UauqzuitJphkhebjqxRfJd0ZeeKENCkpFa0FIdj8OTHY1QnBRUJZ1Z10V2NueGZUw3Q1Tm90Z2pnN2g2Q0IzakNCndZsktVnk1odmNoQvfrT01Z5E4Nck1J5etNqevHQ1NzR0FRU
UJ7amVQadRVUcKSmFvU1JEU0uR81Y0mVnK5wGj1IcUJtY3d0VYFHQTFVZEVU0ZWCmsEQ0Z1YVUCb0pQ0et6RfDnR0tHQTFVURUJbd11N0zFvVTF50E1pHVVUVMV4TXXb0pES1LaakZrT0h0ESERKkNckXU0XhV0t0T1d
JMU90WntFV0U0WmpFefUUT8Cv1L5h0pBZ1n0b0pRakfKay9JclpBRU1EQThTVR0B0UUXcnckXU0XpRREF3TURNeERUQUXcZ05wQkFzTUZrRmp1V1VnVj7sa1oyVjANCnc2TEhNtutay3lCTV7FUXhIekFkQmdov
0ZzZE6nZ1EYRn1aVEFLQmdncHrak9QVUFQ0u0SEFEQkVBAUJ0uUE13c07Bc1dubJ3lZ1I2R2tvW0NCK1N0HEStEUAJZE0zYqk0h05UVBw0d0Z1R0K0cyRjYvhtzV4cnJMR005b0dEbnA2b0501dyN0N0b2UwX10R11UNcn3
5TTONC10tLS0tRUSEIENFU1RJRk1DQVRFF1F3UVVUFU1QTL50tLQ0k"
}
```

Execute

- Result (200)

Code

Details

200

Response body

```
{
  "requestID": "30360",
  "tokenType": "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0WS09v3",
  "dispositionMessage": "ISSUED",
  "binarySecurityToken": "TU1JR0Z6Q0MBENENQXdkJkFnsV1d0F8ZHF8bU0cXNqC601Q3d0QfBQj3vREF1QmdncHrak9QUVEQpCak15VxdFdl1lQip7b01aU1W1FCR13ZRMJH0wPZV3d4RXPB
Uk7n0bpramfKay9JclpBRU1EQThTVR0B0UUXcnckXU0XpRREF3TURNeERUQUXcZ05wQkFzTUZrRmp1V1VnVj7sa1oyVjANCnc2TEhNtutay3lCTV7FUXhIekFkQmdov
k3Bb01Ga0ZqY1dV11yYb0taH1Yudz7Nq2d51pJeuJNVKVR8EppqusNckJnT1L2CQ1NSF25VFZDMDRPRFkuTXpFeE5EIOXRnekF3TV9jeE5UMTVPFVF3TURBek1GUXdFQV1IS29a5XpqlEMNCKFRUJZLNEVFQURFRFFnQUV1d0N
NchBaN19ta0dnazMONULRRwt3amSVr05UauqzuitJphkhebjqxRfJd0ZeeKENCkpFa0FIdj8OTHY1QnBRUJZ1Z10V2NueGZUw3Q1Tm90Z2pnN2g2Q0IzakNCndZsktVnk1odmNoQvfrT01Z5E4Nck1J5etNqevHQ1NzR0FRU
UJ7amVQadRVUcKSmFvU1JEU0uR81Y0mVnK5wGj1IcUJtY3d0VYFHQTFVZEVU0ZWCmsEQ0Z1YVUCb0pQ0et6RfDnR0tHQTFVURUJbd11N0zFvVTF50E1pHVVUVMV4TXXb0pES1LaakZrT0h0ESERKkNckXU0XhV0t0T1d
JMU90WntFV0U0WmpFefUUT8Cv1L5h0pBZ1n0b0pRakfKay9JclpBRU1EQThTVR0B0UUXcnckXU0XpRREF3TURNeERUQUXcZ05wQkFzTUZrRmp1V1VnVj7sa1oyVjANCnc2TEhNtutay3lCTV7FUXhIekFkQmdov
0ZzZE6nZ1EYRn1aVEFLQmdncHrak9QVUFQ0u0SEFEQkVBAUJ0uUE13c07Bc1dubJ3lZ1I2R2tvW0NCK1N0HEStEUAJZE0zYqk0h05UVBw0d0Z1R0K0cyRjYvhtzV4cnJMR005b0dEbnA2b0501dyN0N0b2UwX10R11UNcn3
5TTONC10tLS0tRUSEIENFU1RJRk1DQVRFF1F3UVVUFU1QTL50tLQ0k"
  "secret": "f9YRhopN/G7x0TECOY6nKSchLW1b5r1AHSFPICo4qe="
}
```

Response headers

```
cache-control: no-store,max-age=0
content-type: application/json
expires: 0
pragma: no-cache
x-ratelimit-limit: name=default,100;
x-ratelimit-remaining: name=default,80;
```





2.3.10.5 REPORTING

- Step 1: Open CMD and Generate simplified invoice
 - Step 2: On SDK, sign the XML invoice and get the hash value using the following:
 - `fatoora -sign -qr -invoice invoiceName.xml -signedinvoice signedinvoiceName.xml` (hash is returned and signed invoice is generated)
 - `fatoora -generateHash -invoice invoiceName.xml` (optional)
- Afterwards, validate the XML invoice.

```
Microsoft Windows [Version 10.0.19041.450]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\HYaghmour\Downloads\SDK_204_Samples (1)\Samples\Simplified\Invoices>fatoora -sign -invoice simplified_invoice.xml
***** Welcome to ZATCA E-Invoice Java SDK 3.0.9 *****
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.

*****
2022-07-05 15:19:36,924 [INFO] InvoiceSigningService - invoice has been signed successfully
2022-07-05 15:19:36,926 [INFO] InvoiceSigningService - *** INVOICE HASH = QnVEexW4nWv4CaE39a/66Jp/OX0/evHQ8pD1G7weq/4=

C:\Users\HYaghmour\Downloads\SDK_204_Samples (1)\Samples\Simplified\Invoices>
```

```
2022-07-05 15:29:05,324 [INFO] InvoiceRequestGenerationService - invoice request has been generated successfully

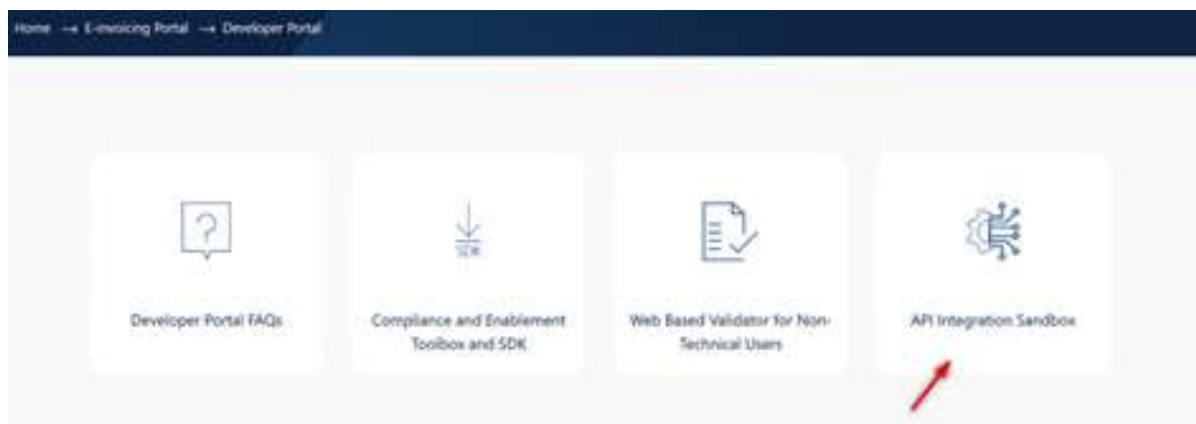
C:\Users\HYaghmour\Downloads\SDK_204_Samples (1)\Samples\Simplified\Invoices>fatoora -qr -invoice simplified_invoice.xml
***** Welcome to ZATCA E-Invoice Java SDK 3.0.9 *****
This SDK uses Java to call the SDK (jar) passing it an invoice XML file.
It can take a Standard or Simplified XML, Credit Note, 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.

*****
2022-07-05 15:36:30,698 [INFO] QrGenerationService - Qr has been generated successfully
2022-07-05 15:36:30,701 [INFO] QrGenerationService - *** QR code = ARd8aG1lZCBnb2hhbWVkeTEFhIEFobWfkeQIPHzAwMTIxOTcxNTAwMDAaZAAxQyQDIyLTZlZD00jQwOjQwMgQ*MTExOC45MAUHFHTQ
-jkGLFFuVkvV1eFrc0bld2NEMhRTMSYS82NkpLb9YtY9ldkHRCBEBEc3d2VxLzQ9B2BNRWDSVFDL3dyCHfBcG1URH3Q3FLN2ZLHm8zhF1EL1jyeHNDejBKZF1qd1dpV1V5Z0lnZFV2d8FqVkrPaHgvUxvTlwtDaGV1UDJ5RDN
jGt4cWRHTG9xZVpXbGpVND0INDBiMBAGByqGSH49AgEGBSuBBAKA8IABGGDDKDMHWAITDv7LXqLX2cmr6+qddUkpcLCvWs5rC2029M/hS4aJAK4Qdnahym6MaijX75Cg3j4ao0yYXJCEJRzBFa1EA4CHMet8Z44shs8h50Bj
veJhVqshieEgjlUCzxZ5NMUCITRo/gbV3KHu8KNVMyDr+sP600l8gVDe5v4n5b1T08p
```





- Step 3: Encode the Signed XML invoice using Base 64
- Step 4: Open Developer Portal and choose integration sandbox





- Step 5: Click on API documentation guides

Home → E-Invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

API Integration Sandbox

Sandbox Release 2.1.1 (Latest version)

Sandbox Release 2

Sandbox Release 1.5

Sandbox Release 1

The ZATCA e-invoicing API Integration Sandbox is meant to be used for testing purposes only. Any e-invoices or their associated notes submitted on the Sandbox are not considered as acknowledged, approved or accepted by ZATCA. Taxpayers will be required to register for e-invoicing on the ZATCA e-invoicing production system (FATOORA) in order to officially be able to submit their e-invoices to ZATCA.

Developers must also take into account that e-invoices, credit and debit notes submitted on the production system will be subjected to additional validations such as security features, prohibited functionalities and additional business rule validations as part of the Clearance process.

The following table provides a summary description of the APIs including the key outputs and inputs/pre-requisites for each API. Additional details can be obtained from the User Manual provided at the end.

API Name	Description	Output	Pre-requisites
----------	-------------	--------	----------------

- Step 6: Click on Authorize Reporting API

Home → E-Invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

Overview

API Documentation guide

Reporting

Reporting API

Clearance

Clearance API

Compliance CSID

Compliance CSID API

Compliance Checks

Compliance Invoice API

Production CSID

Production CSID (Onboarding) API

Production CSID (Renewal) API

e-Invoicing Sandbox Release (2.1.0)

ZATCA wants to provide Taxpayers and Developers of Taxpayer e-invoicing solutions and devices the opportunity to test the integration of the systems with a ZATCA Sandbox environment prior to the launch of the production system. The Integration Sandbox (ISB) should enable solution developers to simulate the integration calls/requests that will be required later as part of the registration process and the submission of e-invoices, credit and debit notes to the production system. The Sandbox backend will accordingly simulate the validations and responses as part of the Cryptographic Stamp Identifiers issuance, renewal and revocation as well as the Reporting and Clearance function.

Although the ISB will give ZATCA an indication of the adoption rate for e-invoicing solutions in the market, it will not be mandatory to complete Sandbox testing as a pre-requisite for Registration/Taxpayer onboarding or accessing the production system. Similar to the Compliance and Enablement Toolbox (CET), the ISB is also aimed at Developers to build/update their solutions which are in line with ZATCA specifications and standards and are able to integrate with a ZATCA backend. Accordingly access to the ISB test/mock APIs will not be limited to Taxpayers and any user can register for a Developer account to access the ISB test/mock APIs and associated documentation. This registration will enable ZATCA to monitor the solution providers who intend to develop/update their solutions to integrate with ZATCA.

It should be noted that although the ISB will simulate most of the core functionalities of the production system, any validations that require integrations/access with external systems and/or storage as well as scenarios involving any backend exceptional handling (for example overriding the clearance process) will not be part of the ISB and will be covered by the core solution. Accordingly the ISB should not be considered as representative of all integrations and/or APIs that will be part of the production system.

Kindly note that validations which can result in an UBL XSD error also apply to optional fields if the tag is present and data input is not compliant. This includes leaving such fields blank. However if the tag itself is absent then the validations will not be performed.

This swagger documents the set of apis for the Sandbox (ISB) solution.

Developers can also refer to section 2.3.10 of the Developer Portal User Manual for additional guidance and steps.

Servers

<https://gw-apis-gov.gazt.gov.sa/e-invoicing/developer-portal>

Authorize

Reporting Model Endpoint(s)

POST `/invoices/reporting/single` Reports a single Invoice.



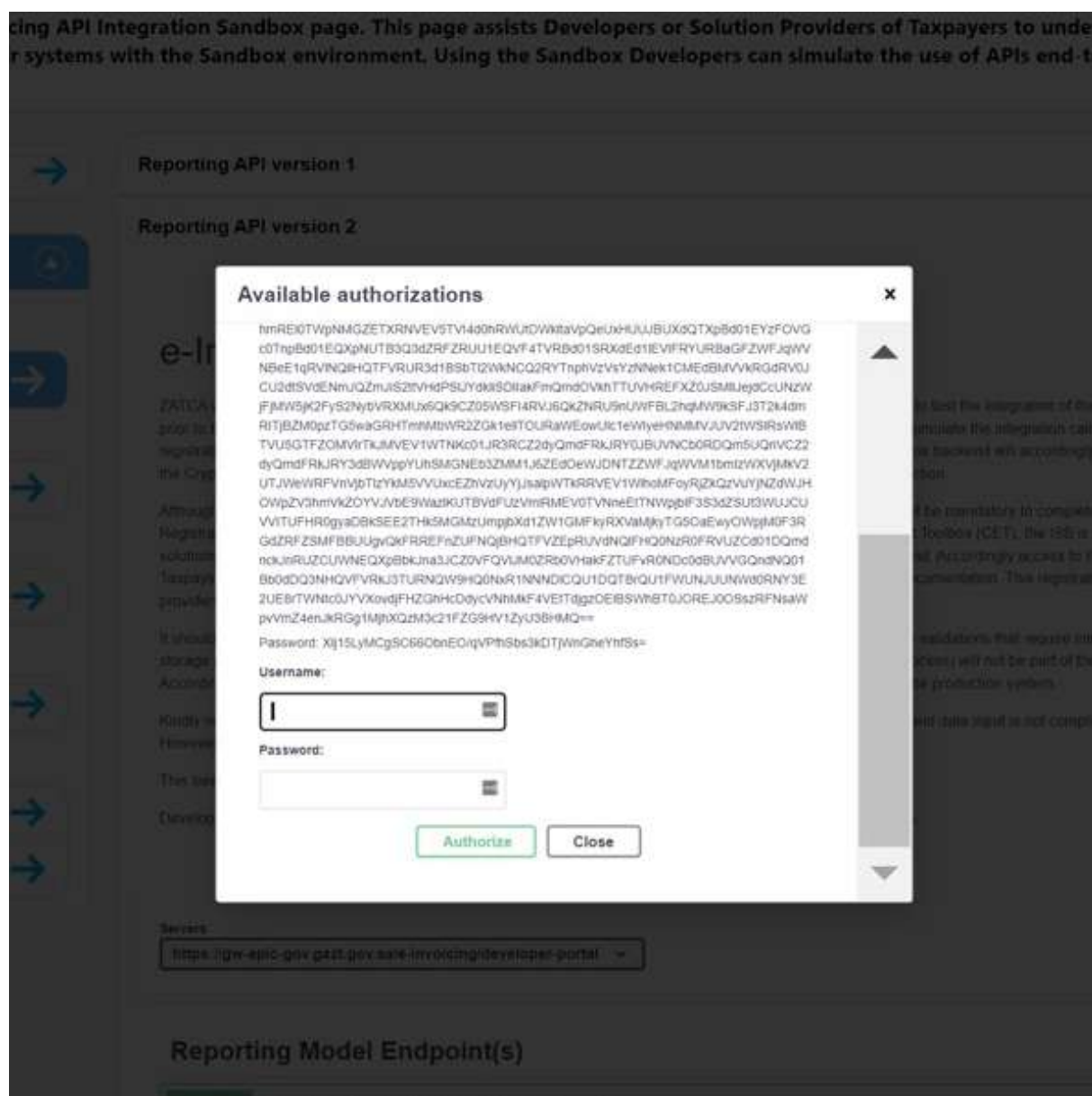


- Step 7:

For the Sandbox, use the sample dummy Username and Password provided to you on the Authorization screen.

For Production, run the Compliance CSID API to obtain the “binarySecurityToken” to be used as the Username and “secret” as the Password.

Please refer to Chapter 3 of this document for further details.





- Step 8: Click on Reporting API drop-down

Home → E-Invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

Overview →

e-Invoicing Sandbox Release (2.1.0) NEW

ZATCA wants to provide Taxpayers and Developers of Taxpayer e-invoicing solutions and devices the opportunity to test the integration of the systems with a ZATCA Sandbox environment prior to the launch of the production system. The Integration Sandbox (ISB) should enable solution developers to simulate the integration calls/requests that will be required later as part of the registration process and the submission of e-invoices, credit and debit notes to the production system. The Sandbox backend will accordingly simulate the validations and responses as part of the Cryptographic Stamp Identifiers issuance, renewal and revocation as well as the Reporting and Clearance function.

Although the ISB will give ZATCA an indication of the adoption rate for e-invoicing solutions in the market, it will not be mandatory to complete Sandbox testing as a pre-requisite for Registration/Taxpayer onboarding or accessing the production system. Similar to the Compliance and Enablement Toolbox (CET), the ISB is also aimed at Developers to build/update their solutions which are in line with ZATCA specifications and standards and are able to integrate with a ZATCA backend. Accordingly access to the ISB test/mock APIs will not be limited to Taxpayers and any user can register for a Developer account to access the ISB test/mock APIs and associated documentation. This registration will enable ZATCA to monitor the solution providers who intend to develop/update their solutions to integrate with ZATCA.

It should be noted that although the ISB will simulate most of the core functionalities of the production system, any validations that require integrations/access with external systems and/or storage as well as scenarios involving any backend exceptional handling (for example overriding the clearance process) will not be part of the ISB and will be covered by the core solution. Accordingly the ISB should not be considered as representative of all integrations and/or APIs that will be part of the production system.

Kindly note that validations which can result in an URL, XSD error also apply to optional fields if the tag is present and data input is not compliant. This includes leaving such fields blank. However if the tag itself is absent then the validations will not be performed.

This swagger documents the set of APIs for the Sandbox (ISB) solution.

Developers can also refer to section 2.3.10 of the Developer Portal User Manual for additional guidance and steps.

Servers
 Authorize

Reporting Model Endpoint(s)

POST `/Invoices/reporting/single` Reports a single Invoice.

- Step 9: Click on try it out

Paramotors Try it out

Name	Description
Authorization string (header)	Username (Production Certificate) and Password (Secret) should be taken from the Production certificate response and put in the Authorize popup : <ul style="list-style-type: none">• Username: 'binarySecurityToken' which represents the Production Certificate• Password: 'secret' <input type="text" value="Authorization"/>
accept-language string (header)	Specifies the language in which the response will be returned. Currently supported languages are English (en) and Arabic (ar) and it defaults to English. Examples: <input type="text" value="English"/> <input type="text" value="en"/>
Clearance-Status <small>* required</small> string (header)	Specifies the clearance status, while "0" when clearance is disabled and "1" when clearance is enabled Examples: <input type="text" value="Disabled"/> <input type="text" value="0"/>
Accept-Version <small>* required</small> string (header)	<input type="text" value="V2"/>

Note: V2 refers to the Version of the APIs used and should be mentioned in the API calls (V2 is currently the only valid version).



52



- Result (200)

Code

Details

200

Response body

```
{
  "validationResults": {
    "infoMessages": [
      {
        "type": "INFO",
        "code": "XSD_ZATCA_VALID",
        "category": "XSD validation",
        "message": "Complied with UBL 2.1 standards in line with ZATCA specifications",
        "status": "PASS"
      }
    ],
    "warningMessages": [],
    "errorMessages": [],
    "status": "PASS"
  },
  "reportingStatus": "REPORTED"
}
```

Response headers

```
cache-control: no-store,max-age=0
content-type: application/json
x-ratelimit-limit: name=default,100;
x-ratelimit-remaining: name=default,59;
```

Download

- Result (400)

400

HTTP Bad Request. Returned when the submitted request is invalid.

Media type

application/json

Examples

Invalid Invoice Hash

Example Value

Schema

```
{
  "validationResults": {
    "infoMessages": [
      {
        "type": "INFO",
        "code": "XSD_ZATCA_VALID",
        "category": "XSD validation",
        "message": "Complied with UBL 2.1 standards in line with ZATCA specifications",
        "status": "PASS"
      }
    ],
    "warningMessages": [],
    "errorMessages": [
      {
        "type": "ERROR",
        "code": "InvoiceHash_QRCODE_INVALID",
        "category": "QRCODE_VALIDATION",
        "message": "Invoice xml hash does not match with qr code invoice xml hash",
        "status": "ERROR"
      }
    ],
    "status": "ERROR"
  }
}
```





2.3.10.6 CLEARANCE

- Step 1: Open CMD and Generate standard invoice
- Step 2: On SDK, get the hash value using the following:
 - fatoorah -generateHash -invoice invoiceName.xml

Afterwards, validate the XML invoice.

```
Microsoft Windows [Version 10.0.19041.450]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\HYaghmour\Downloads\SDK_204_Samples (1)\Samples\Standard\Invoices>fatoorah -validate -invoice standred_invoice.xml
***** Welcome to ZATCA E-Invoice Java SDK 3.0.9 *****
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.

*****
2022-07-05 14:37:35,081 [INFO] XsdValidator - Validate XSD for invoice : standred_invoice.xml
2022-07-05 14:37:35,960 [INFO] ValidationProcessorImpl - [XSD] validation result : PASSED
2022-07-05 14:37:35,963 [INFO] SchematronValidator - Validate Schematron using C:\zatca-envoice-sdk-204\zatca-envoice-sdk-
invoice : standred_invoice.xml
2022-07-05 14:37:37,834 [INFO] ValidationProcessorImpl - [EN] validation result : PASSED
2022-07-05 14:37:37,834 [INFO] SchematronValidator - Validate Schematron using C:\zatca-envoice-sdk-204\zatca-envoice-sdk-
e_validation_rules.xml for invoice : standred_invoice.xml
2022-07-05 14:37:38,377 [INFO] ValidationProcessorImpl - [KSA] validation result : PASSED
2022-07-05 14:37:38,507 [INFO] ValidationProcessorImpl - [PIH] validation result : PASSED
2022-07-05 14:37:38,507 [INFO] InvoiceValidationService - *** GLOBAL VALIDATION RESULT = PASSED

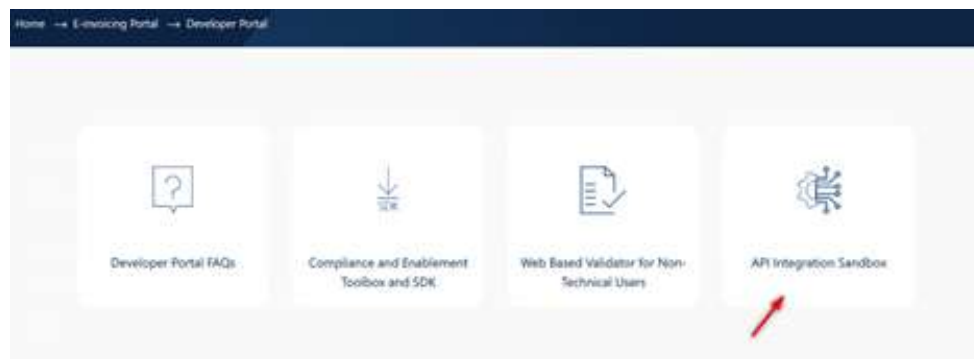
C:\Users\HYaghmour\Downloads\SDK_204_Samples (1)\Samples\Standard\Invoices>
```

- Step 3: Encode the XML invoice using Base 64





- Step 4: Open Developer Portal and choose integration sandbox





- Step 5: Click on API documentation guides

Home → E-Invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

Overview →

API Documentation guide

Reporting

Exporting API →

Clearance

Clearance API →

Compliance CSID

Compliance CSID API →

Compliance Checks

Compliance Invoice API →

Production CSID

Production CSID (Onboarding) API →

Production CSID (Renewal) API →

API Integration Sandbox

Sandbox Release 2.1.1 (Latest version) ▾

Sandbox Release 2 ▾

Sandbox Release 1.5 ▾

Sandbox Release 1 ▾

The ZATCA e-invoicing API Integration Sandbox is meant to be used for testing purposes only. Any e-invoices or their associated notes submitted on the Sandbox are not considered as acknowledged, approved or accepted by ZATCA. Taxpayers will be required to register for e-invoicing on the ZATCA e-invoicing production system (FATOORA) in order to officially be able to submit their e-invoices to ZATCA.

Developers must also take into account that e-invoices, credit and debit notes submitted on the production system will be subjected to additional validations such as security features, prohibited functionalities and additional business rule validations as part of the Clearance process.

The following table provides a summary description of the APIs including the key outputs and inputs/pre-requisites for each API. Additional details can be obtained from the User Manual provided at the end.

API Name	Description	Output	Pre-requisites
----------	-------------	--------	----------------

- Step 6: Click on Authorize Clearance API

Home → E-Invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

Overview →

API Documentation guide

Reporting

Reporting API →

Clearance

Clearance API →

Compliance CSID

Compliance CSID API →

Compliance Checks

Compliance Invoice API →

Production CSID

Production CSID (Onboarding) API →

Production CSID (Renewal) API →

e-Invoicing Sandbox Release (2.1.0) CSID CRSD

ZATCA wants to provide Taxpayers and Developers of Taxpayer e-invoicing solutions and devices the opportunity to test the integration of the systems with a ZATCA Sandbox environment prior to the launch of the production system. The Integration Sandbox (ISB) should enable solution developers to simulate the integration calls/requests that will be required later as part of the registration process and the submission of e-invoices, credit and debit notes to the production system. The Sandbox backend will accordingly simulate the validations and responses as part of the Cryptographic Stamp Identifier Issuance, renewal and revocation as well as the Reporting and Clearance function.

Although the ISB will give ZATCA an indication of the adoption rate for e-invoicing solutions in the market, it will not be mandatory to complete Sandbox testing as a pre-requisite for Registration/Taxpayer onboarding or accessing the production system. Similar to the Compliance and Enablement Toolkit (CET), the ISB is also aimed at Developers to build/update their solutions which are in line with ZATCA specifications and standards and are able to integrate with a ZATCA backend. Accordingly access to the ISB test/mock APIs will not be limited to Taxpayers and any user can register for a Developer account to access the ISB test/mock APIs and associated documentation. This registration will enable ZATCA to monitor the solution providers who intend to develop/update their solutions to integrate with ZATCA.

It should be noted that although the ISB will simulate most of the core functionalities of the production system, any validations that require integrations/clearance with external systems and/or storage as well as scenarios involving any backend exceptional handling (for example overriding the clearance process) will not be part of the ISB and will be covered by the core solution. Accordingly the ISB should not be considered as representative of all integrations and/or APIs that will be part of the production system.

Kindly note that validations which can result in an ISB VSD error also apply to optional fields if the tag is present and data input is not compliant. This includes leaving such fields blank. However if the tag itself is absent then the validations will not be performed.

This sandbox documents the set of APIs for the Sandbox (ISB) solution.

Developers can also refer to section 2.3.10 of the Developer Portal User Manual for additional guidance and steps.

Servers
 [Authorize](#)

Clearance Model Endpoint(s)

POST /invoices/clearance/single clear a single invoice



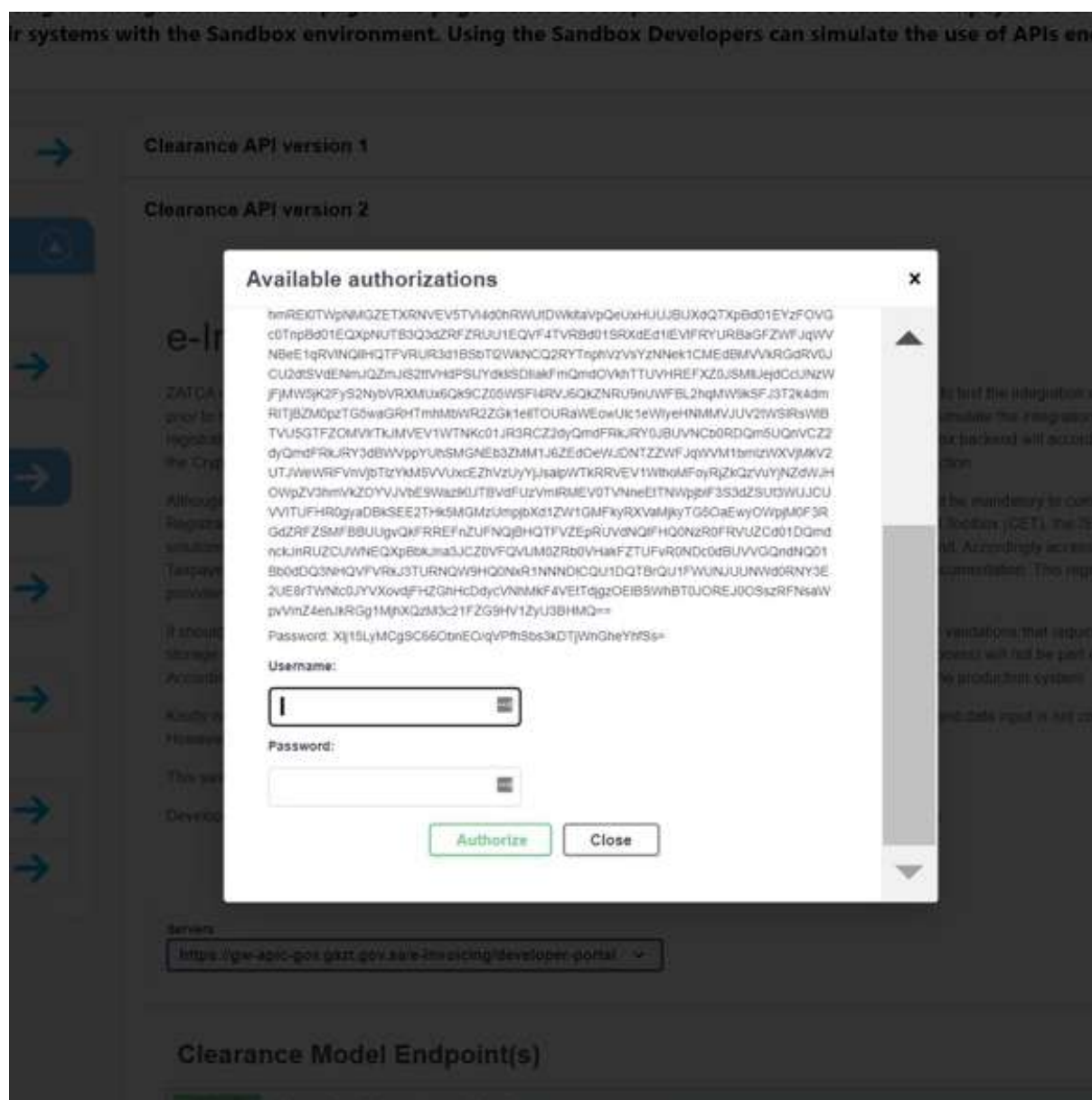


- Step 7:

For the Sandbox, use the sample dummy Username and Password provided to you on the Authorization screen.

For Production, run the Compliance CSID API to obtain the “binarySecurityToken” to be used as the Username and “secret” as the Password.

Please refer to Chapter 3 of this document for further details.





- Step 8: Click on Clearance API drop-down

Home → E-Invoicing Portal → Developer Portal → API Integration Sandbox

API Integration Sandbox

Welcome to the ZATCA e-invoicing API Integration Sandbox page. This page assists Developers or Solution Providers of Taxpayers to understand and test the integration of their systems with the Sandbox environment. Using the Sandbox Developers can simulate the use of APIs end-to-end.

Overview →

e-Invoicing Sandbox Release (2.1.0) 0.0.0 GAES

ZATCA wants to provide Taxpayers and Developers of Taxpayer e-invoicing solutions and devices the opportunity to test the integration of the systems with a ZATCA Sandbox environment prior to the launch of the production system. The Integration Sandbox (ISB) should enable solution developers to simulate the integration calls/requests that will be required later as part of the registration process and the submission of e-invoices, credit and debit notes to the production system. The Sandbox backend will accordingly simulate the validations and responses as part of the Cryptographic Stamp Identifiers issuance, renewal and revocation as well as the Reporting and Clearance function.

Although the ISB will give ZATCA an indication of the adoption rate for e-invoicing solutions in the market, it will not be mandatory to complete Sandbox testing as a pre-requisite for Registration/Taxpayer onboarding or accessing the production system. Similar to the Compliance and Enablement Toolbox (CET), the ISB is also aimed at Developers to build/update their solutions which are in line with ZATCA specifications and standards and are able to integrate with a ZATCA backend. Accordingly access to the ISB test/mock APIs will not be limited to Taxpayers and any user can register for a Developer account to access the ISB test/mock APIs and associated documentation. This registration will enable ZATCA to monitor the solution providers who intend to develop/update their solutions to integrate with ZATCA.

It should be noted that although the ISB will simulate most of the core functionalities of the production system, any validations that require integrations/access with external systems and/or storage as well as scenarios involving any backend exceptional handling (for example overriding the clearance process) will not be part of the ISB and will be covered by the core solution. Accordingly the ISB should not be considered as representative of all integrations and/or APIs that will be part of the production system.

Kindly note that validations which can result in an UBL XSD error also apply to optional fields if the tag is present and data input is not compliant. This includes leaving such fields blank. However if the tag itself is absent then the validations will not be performed.

This swagger documents the set of apis for the Sandbox (ISB) solution.

Developers can also refer to section 2.3.10 of the Developer Portal User Manual for additional guidance and steps.

Servers
https://gw-api-gov.gst.gov.sau/e-invoicing/developer-portal

Authorize

Clearance Model Endpoint(s)

POST /invoices/clearance/single clear a single invoice.





- Step 9: Click on try it out

Parameters Try it out

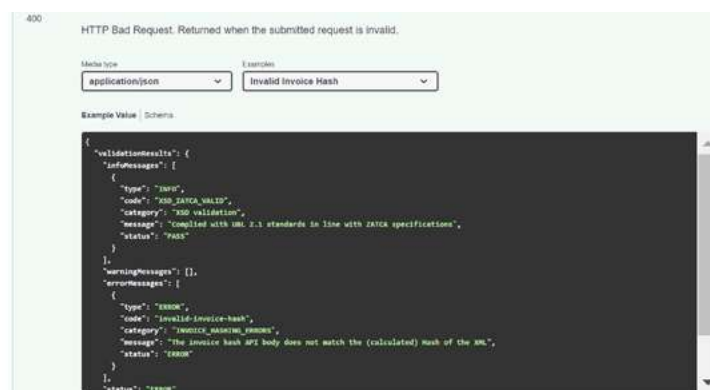
Name	Description
Authorization string (header)	Username (Production Certificate) and Password (Secret) should be taken from the Productin certificate response and put in the Authorize popup : <ul style="list-style-type: none">• Username: 'binarySecurityToken' which represents the Production Certificate• Password: 'secret' <input type="text" value="Authorization"/>
accept-language string (header)	Specifies the language in which the response will be returned. Currently supported languages are English (en) and Arabic (ar) and it defaults to English. Examples: <input type="text" value="English"/> <input type="text" value="en"/>
Clearance-Status * required string (header)	Specifies the clearance status, while "0" when clearance is disabled and "1" when clearance is enabled Examples: <input type="text" value="Disabled"/> <input type="text" value="0"/>
Accept-Version * required string (header)	<input type="text" value="V2"/>

Note: V2 refers to the Version of the APIs used and should be mentioned in the API calls (V2 is currently the only valid version).





- Result (400)



2.3.11 API Summary

Table 1

The following table gives a more detailed summary of the differences between the Integration Sandbox releases in terms of the APIs as well associated components. The current release also indicates the new additions/changes made in comparison to the previous release.

Functionality	Description	Release 1 (November 2021)	Release 1.5 (February 2022)	Release 2 (Current - April 2022)
APIs	The list of APIs that are covered in each release including references to the functionalities they are part of	Invoices APIs: <ul style="list-style-type: none">Reporting APIClearance API Onboarding APIs: <ul style="list-style-type: none">CSID API (for Onboarding)CSID API (for Renewal)	Invoices APIs: <ul style="list-style-type: none">Reporting APIClearance API Onboarding APIs: <ul style="list-style-type: none">Compliance CSID APIProduction CSID API (for Onboarding)Production CSID API (for Renewal)Compliance Checks APIs (for Onboarding / Renewal)	Invoices APIs: <ul style="list-style-type: none">Reporting APIClearance API Onboarding APIs: <ul style="list-style-type: none">Compliance CSID APIProduction CSID API (for Onboarding)Production CSID API (for Renewal)Compliance Checks APIs (for Onboarding / Renewal)





Validation Engine (For Invoices)	The treatment of validations and exceptions as part of the Reporting and Clearance process. Exceptions here refer to warnings which are similar to errors but do not cause the submitted invoices/documents to be rejected but are still indicated in the response so that they can be corrected in future submissions.	<ul style="list-style-type: none"> As per original (published) data dictionary, XML Implementation Standards and Security Features and Implementation Standards No exceptions (Invoices are either accepted or rejected) Not possible to test for Sandbox behavior When Clearance is disabled 	<ul style="list-style-type: none"> As per updated data dictionary, XML Implementation Standards and Security Features and Implementation Standards (including updates to CSR and CSID standards) Seller Address field will be accepted with warning for Taxpayer devices / solution units to differentiate between a warning and an error response Not possible to test for Sandbox behavior when Clearance is disabled 	<ul style="list-style-type: none"> As per updated data dictionary, XML Implementation Standards and Security Features and Implementation Standards (including updates to CSR and CSID standards) Seller Address field will be accepted with warning for Taxpayer devices / solution units to differentiate Between a warning and an error response Two variants of the Reporting and Clearance APIs which are configured with Clearance disabled is being provided - NEW Note: In the Core Invoicing Solution there will only be one API each for Reporting and Clearance which at any point of time will either be configured to Clearance being enabled or disabled
CSR and CSID (For Onboarding)	The formats and fields for the Certificate Signing Request (CSR) and the resultant Cryptographic Stamp Identifier (CSID) that is used as part of the Onboarding process.	As per the original (published) Security Features and Implementation Standards	As per the updated Security Features and Implementation Standards	As per the updated Security Features and Implementation Standards
Swagger Files (API Specifications)	The API documentation associated with the Swagger files.	<ul style="list-style-type: none"> Covers the APIs mentioned above No provisions for Exceptions or turning off Clearance 	<ul style="list-style-type: none"> Covers the APIs mentioned Provision for 1 Exception (Non-compliance in the Seller's Address field is accepted as a warning) No provisions for turning off Clearance Covers the two separate APIs for Compliance and Production CSIDs 	<ul style="list-style-type: none"> Covers the Reporting and Clearance APIs above with provision for 1 Exception Covers the Reporting and Clearance APIs above with provision for turning off Clearance (through two additional variants of the Reporting and Clearance APIs) - NEW Covers the two separate APIs for Compliance and Production CSIDs





Table 2

The following table provides a summary description of the APIs including the key outputs and inputs/pre-requisites for each API.

API Name	Description	Output	Pre-requisites
Reporting API	<p>This API should be used to test submitting Simplified e-invoices, credit or debit note to the ZATCA backend system as part of the Reporting process</p> <p>When Clearance is disabled, this API can also be used to test submitting Standard e-invoices, credit or debit notes for Reporting</p> <p>Note: In the Integration Sandbox there will be two variants of the Reporting API, one which is configured to Clearance being enabled (i.e. it will not accept Standard documents) and one which is configured to Clearance being disabled (i.e. it will also accept Standard documents to be submitted for Reporting)</p>	<ul style="list-style-type: none"> ● If no errors or warnings: Accepted ● If error in Seller Address: Accepted with warning message ● If errors other than Seller Address: Rejected with error messages 	<ul style="list-style-type: none"> ● A test Production CSID obtained from API #5 or #6 below ● Simplified invoice, credit or debit note in XML format ● Standard invoice, credit or debit note in XML format when Clearance is disabled
Clearance API	<p>This API should be used to test submitting test Standard e-invoices, credit or debit note to the ZATCA backend system as part of the Clearance process</p> <p>When Clearance is disabled, this API will return a 303 Response indicating that the Reporting API be used to submit Standard documents as well</p> <p>Note: In the Integration Sandbox there will be two variants of the Clearance API, one which is configured to Clearance being enabled (i.e. it will validate and clear Standard documents) and one which is configured to Clearance being disabled (i.e. it will return response 303 stating that Clearance is currently disabled and the Reporting API must be used to submit Standard documents as well)</p>	<ul style="list-style-type: none"> ● If no errors or warnings: Accepted and document is returned with test ZATCA stamp and QR code ● If error in Seller Address: Accepted with warning message and document is returned with test ZATCA stamp and QR code ● If errors other than Seller Address: Rejected with error messages ● Response 303 when Clearance is disabled asking the Reporting API to be used to submit Standard documents 	<ul style="list-style-type: none"> ● A test Production CSID obtained from API #5 or #6 below ● Standard invoice, credit or debit note in XML format





Compliance CSID API	This API should be used to test submitting test CSRs (Certificate Signing Requests) to the ZATCA backend system as part of the Onboarding and renewal process	<ul style="list-style-type: none"> Valid request: Test Compliance CSID and a test Request ID are obtained Invalid request: Error message(s) 	<ul style="list-style-type: none"> Public Private Key pair Signed CSR
Production CSID API (for Onboarding)	This API will be used to submit a test Request ID to a test ZATCA backend system as part of the Onboarding process	<ul style="list-style-type: none"> Valid request: Test Production CSID is obtained Invalid request: Error message(s) 	<ul style="list-style-type: none"> A test Compliance CSID obtained from APIs #3 above A test (dummy) request ID
Production CSID API (for Renewal)	This API will be used to submit a test Request ID to a test ZATCA backend system as part of the Onboarding process	<ul style="list-style-type: none"> Valid request: Test Production CSID is obtained Invalid request: Error message(s) 	<ul style="list-style-type: none"> A test Compliance CSID obtained from APIs #3 above A test (dummy) request ID
Compliance Checks APIs (for Onboarding / Renewal)	<p>These APIs should be used to test the compliance check for the device / solution unit (EGS) as part of the Onboarding and/or Renewal processes</p> <p>The compliance checks include checking compliance of Standard and/or Simplified documents when Clearance is enabled (Compliance Invoice API) or when Clearance is disabled (Compliance Invoice Clearance Disabled API);</p>	<ul style="list-style-type: none"> All Compliance checks passed One or more compliance checks failed with error messages 	<ul style="list-style-type: none"> A test Compliance CSID obtained from APIs #3 above Standard and/or Simplified invoices, credit or debit notes in XML format





2.3.12 Accessing the Developer Portal Support Page

The Developer Portal Support Page can be accessed from the main dashboard of the Developer Portal and does not require any prior registration / log in. Through this page, the user can view the different types of support available which includes the Toolbox and Sandbox documentation. In addition, the user can view the FAQ section to find readily available answers to common inquiries they may have on the Developer Portal tools and functionalities as well as more specific questions on testing the compliance of their XMLs. Users can also find the support contact information that they can access should they require any support. This includes phone number / hotline, international phone number and the email address. Users could also provide any suggestions or complaints they may have.

The user can access the Developer Portal Support Page from the main Developer Portal page.

The following categories are available to users:

- General support
- SDK support
- Integration Sandbox support
- Compliance and Enablement Toolbox support





A search bar is also readily available for users to search and obtain the relevant information easily. The user can view common enquiries in the FAQ page. The user can see a contact section at the bottom of the support page in case of experiencing any issues and in the event that the user would want to receive the support of the contact center.

