FATOORA

هيئة الزكاة والضريبة والجمارك
Zakat, Tax and Customs Authority

User Manual

Developer Portal Manual
Version 3

فـاتورة
Fatoora

Nov 2022

Contents

1. General Information

1.1 Introduction

The Developer Portal is a dedicated portal provided by ZATCA for the developers of E-invoicing Generation Solutions (EGS). It contains two development tools aimed at supporting developers build compliant EGS units which are:

- **The Compliance and Enablement Toolbox Software Development Kit (SDK):** an offline downloadable tool which can be used to validate an XML based e-invoice, credit or debit note files in accordance with the ZATCA published requirements, standards and guidelines. 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 Command Line Interface (CLI).
- **Integration Sandbox:** a test ZATCA backend system which EGS units can integrate with to make API calls to simulate and test the Onboarding process followed by the submission of test e-invoices, credit and debit notes for Reporting and Clearance in accordance with the ZATCA published requirements, standards and guidelines.

In addition to the above, the Developer Portal has a third tool aimed at intermediate or non-technical users to validate an XML based e-invoice, credit or debit note files from the portal directly. This is referred to as the **Compliance and Enablement Toolbox Portal Based Validator**.

Finally, the Developer Portal has a dedicated support page containing a list of Frequently Asked Questions (FAQs) to help developers troubleshoot during development and testing of their EGS units as well as provide guidance on E-invoicing requirements in general.

1.1.1 Objectives

The primary objective of the Developer Portal is to support the Developer community in building EGS units that are compliant with ZATCA's XML implementation standards as well as the Security Features and Implementation Standards.

- The objective of the SDK is to ensure that XML e-invoices, credit or debit notes being generated by the EGS units are compliant
- The objective of the Web based validator is to allow less or non-technical users to be able to independently validate the compliance of XML e-invoices, credit or debit notes and share the results with their developers.
- The objective of the Integration Sandbox is to test the EGS units / solutions / applications are able to integrate with a test ZATCA backend system via APIs covering the following:
    - Submit a test Certificate Signing Request (CSR) to obtain a test Compliance Cryptographic Stamp Identifier (CCSID) and test Request ID
    - Submit a test Request ID to obtain a test Production CSID
    - Submit test Standard Documents using the test Clearance API (or using a variant of the test Reporting API to mimic the process when Clearance is turned off)
    - Submit Simplified Documents using the Reporting API

1.1.2 Scope

This user manual acts as a guide for Developers in order to help them access and use the Developer Portal features and functionalities. It explains in detail the user journey including steps, requirements and processes needed for accessing the Developer Portal. Moreover, it provides guidance for the relevant technical aspects and methods to be used to solve common issues that might be faced by the Portal users. This document covers the following functionalities that are of relevant to the Developer Portal:

- Accessing the Developer Portal
    - Website address and browsers supported
    - Main dashboard elements

- Registration for the Developer Portal
    - Functionalities that require registration
    - Authentication and verification process

- Log in to the Developer Portal
    - Entering user credentials
    - Password reset / Forget Password
    - Successful log in

- Accessing and downloading the Compliance and Enablement Toolbox SDK
    - Understanding of the SDK and how to use it
    - Downloading the SDK

- Accessing and using the Web Based Validator
    - Understanding of and the use of the Web Based Validator
    - Validating XMLs directly on the Web Based Validator

- Accessing and using the Integration Sandbox APIs
    - Accessing documentation for the APIs
    - Using the APIs to integrate with the test ZATCA backend system
    - Test the integration calls for Onboarding, Renewal, Reporting, Clearance

1.1.3 Intended Audience

This document is intended to be used by:

- Solution Developers
- Taxpayers
- Other users of relevance

1.1.4 Recommended Reading

Although not a pre-requisite for accessing and using the Developer Portal functionalities, it is strongly advised that users go through the following documentation:

1. XML Implementation Standards (E-Invoice XML Implementation Standard)
2. Security Features and Implementation Standards (E-Invoice Security Features and Implementation Standards)
3. Data Dictionary (E-Invoice Data Dictionary)
4. E-Invoicing Resolution (E-Invoicing Resolution)

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:

<table>
<tr><th colspan="4">Developer Portal</th></tr>
<tr><td colspan="2">Login</td><td>Access Portal Based Validator</td><td>Access Developer Portal Support Page</td></tr>
<tr><td>Access SDK Page</td><td>Access Integration Sandbox Page</td><td></td><td></td></tr>
<tr><td>• Download SDK
• SDK Support
• SDK Documentation
• SDK Version History</td><td>• Access API Documentation (Swagger Files)
• Test APIs for Onboarding, Renewal, Reporting and Clearance</td><td>Validate XMLs</td><td>Access FAQs</td></tr>
<tr><td colspan="4">Outside Developer Portal</td></tr>
<tr><td>Using the SDK</td><td colspan="2">Using the Integration Sandbox (APIs)</td><td></td></tr>
<tr><td>• Test compliance of XML
• Test compliance of QR Code (Generation Phase)
• Test Compliance of QR Code (Integration Phase)</td><td colspan="2">• 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</td><td></td></tr>
</table>

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:

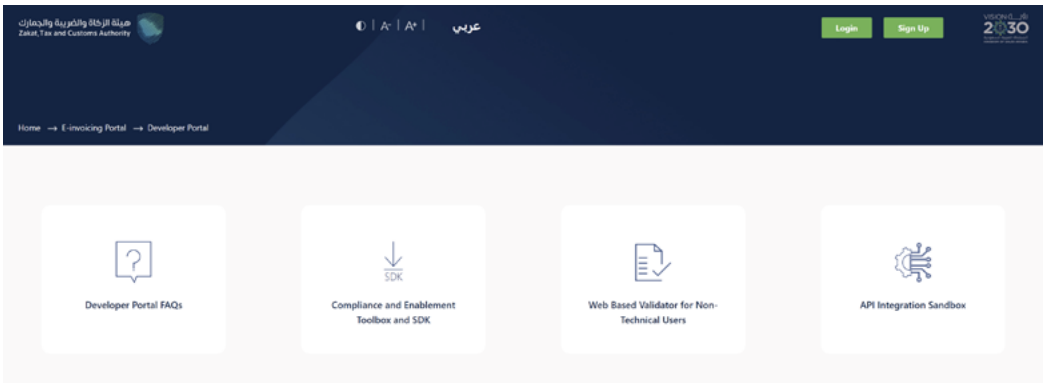
1. Access the Developer Portal
2. Accessing the Compliance and Enablement Toolbox Portal Page
3. Test XML compliance
4. Provide the error messages / responses (if any) to Solution Developers
5. Leveraging the Developer Portal Support page FAQs for troubleshooting

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

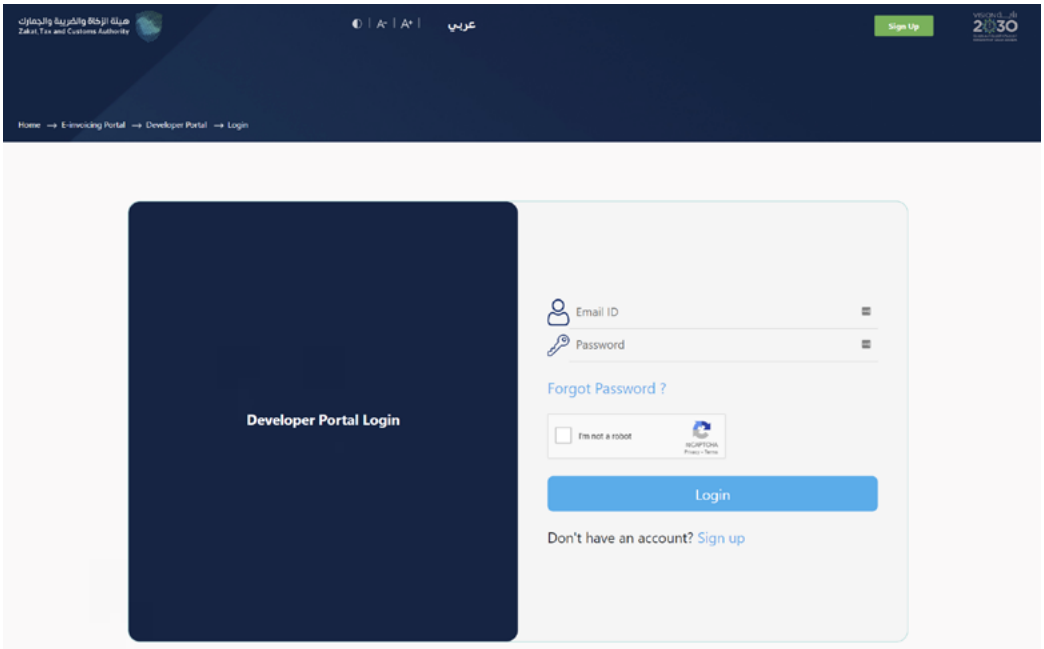
2.3.2 Creating a Developer Portal Account

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

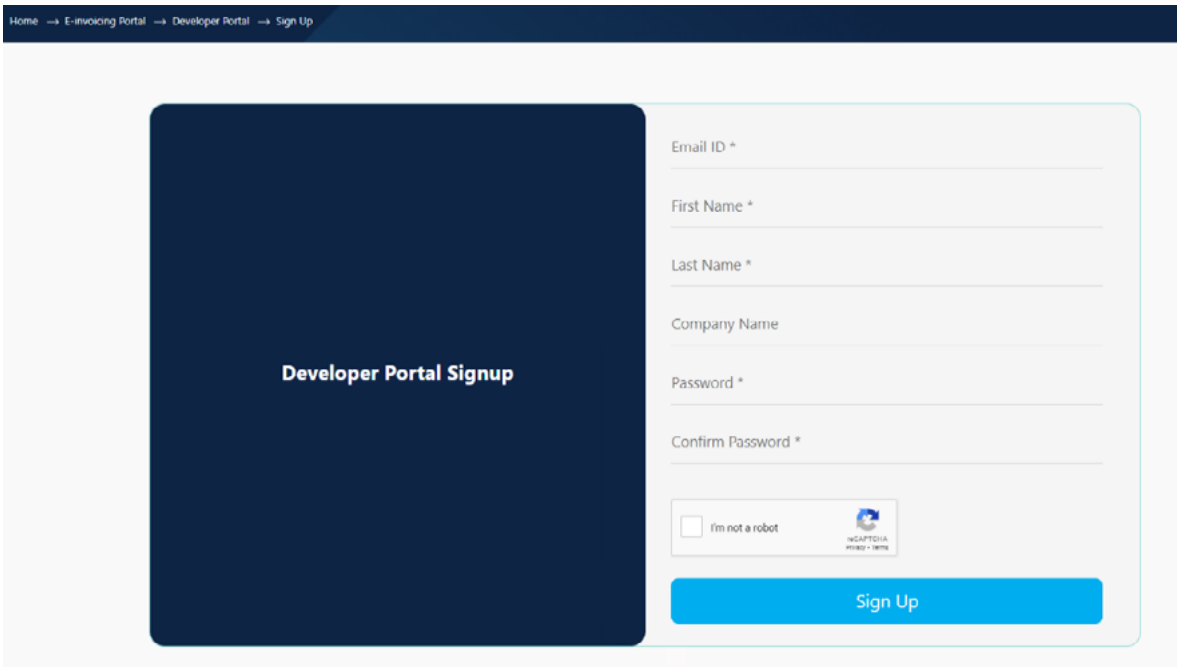
In the Sign Up page (as seen in the Figure below), the user will be prompted to create a new account by providing the following details:
The email must be a valid email and the password must be at least 8 characters comprising of at least one number, one letter each in lower and upper case, and one symbol.
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.

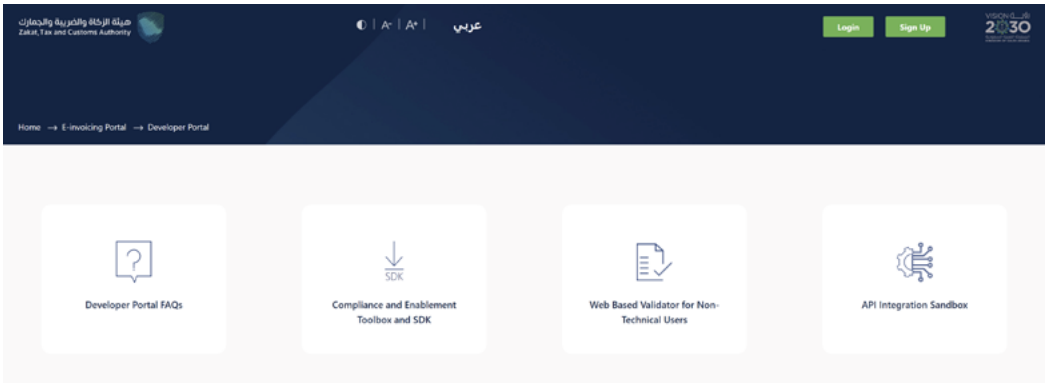


2.3.3 Accessing the Compliance and Enablement Toolbox SDK Page

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:

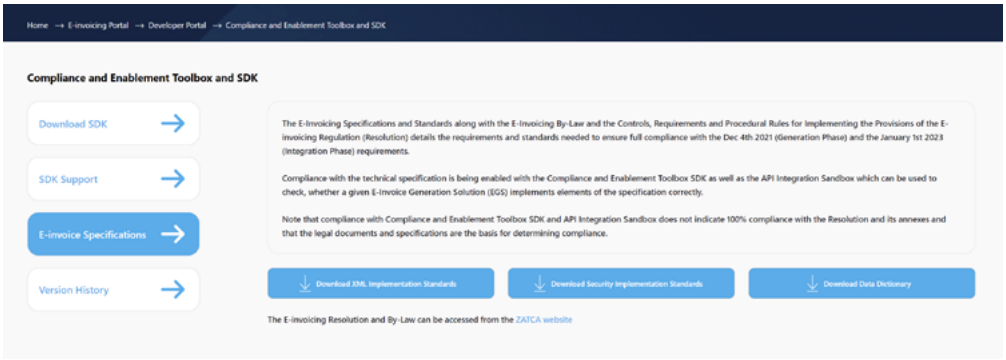
- The user should be registered and logged into the Developer Portal successfully
- The user should click on "Compliance and Enablement Toolbox and SDK" to view the SDK functionalities.



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.

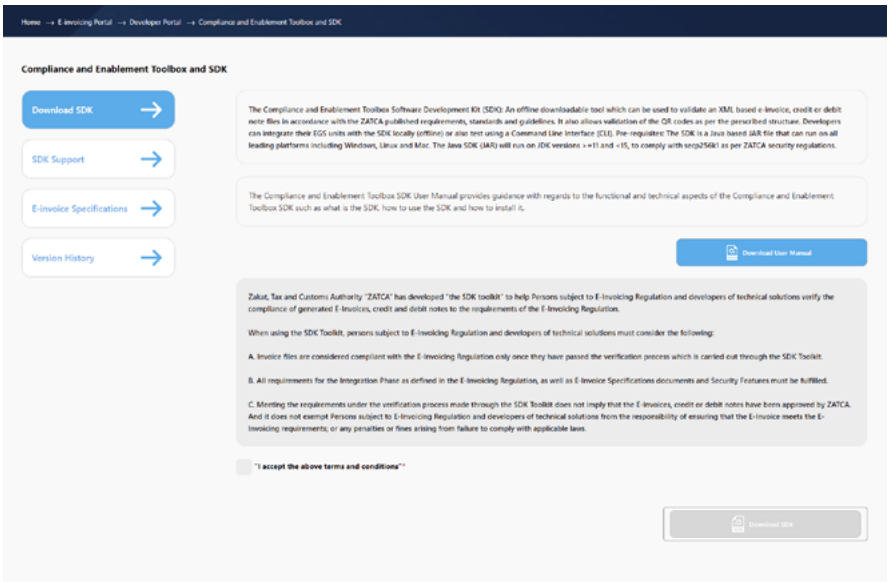


Accessing the E-Invoicing specification documents

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.

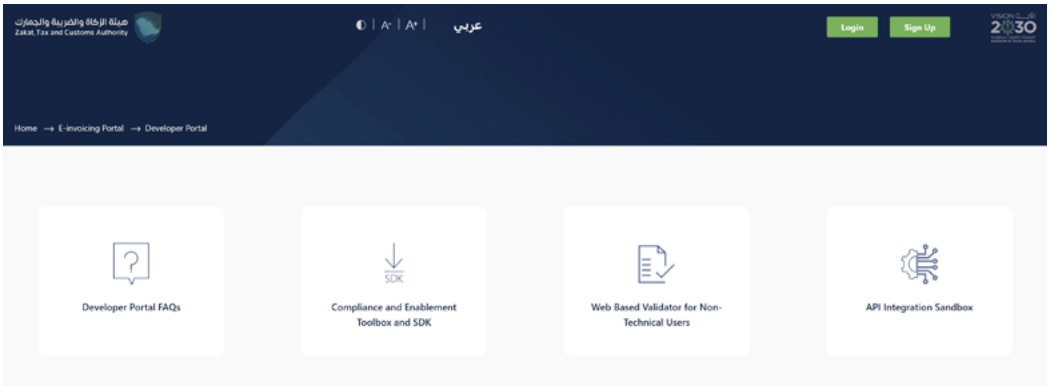
2.3.6 Accessing the Web Based Validator for Non-Technical Users

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:

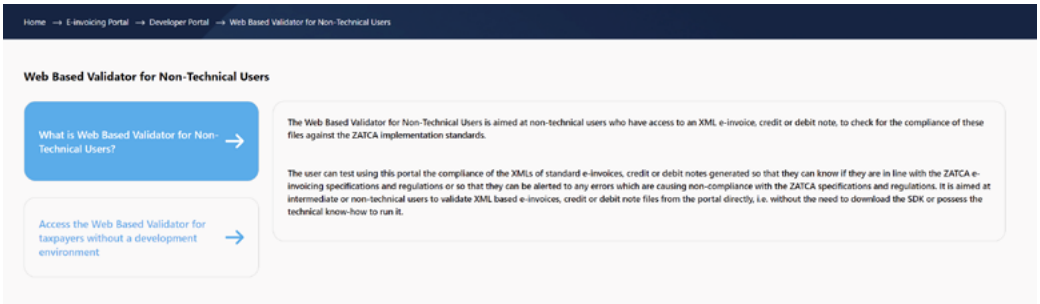
- The User accesses the "Web Based Validator for Non-Technical Users" on the Developer Portal (no prior registration or log in required).



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.

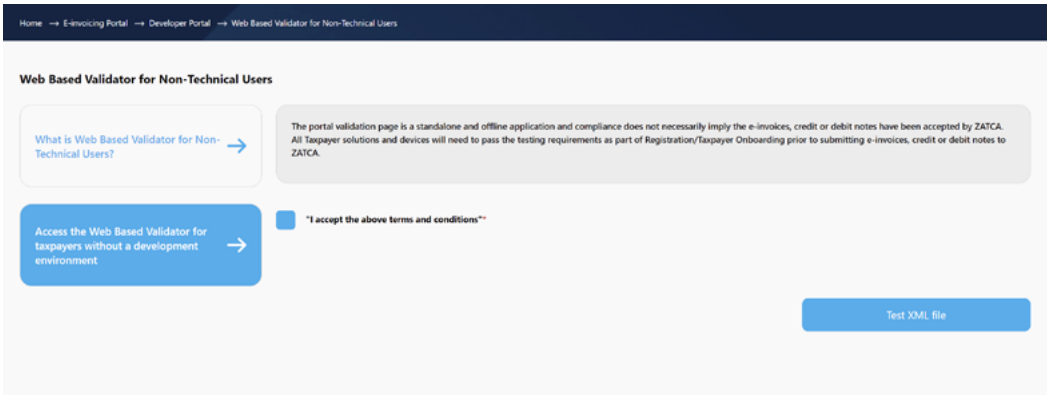


Web based validator Page

- 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.



Web based validator Disclaimer

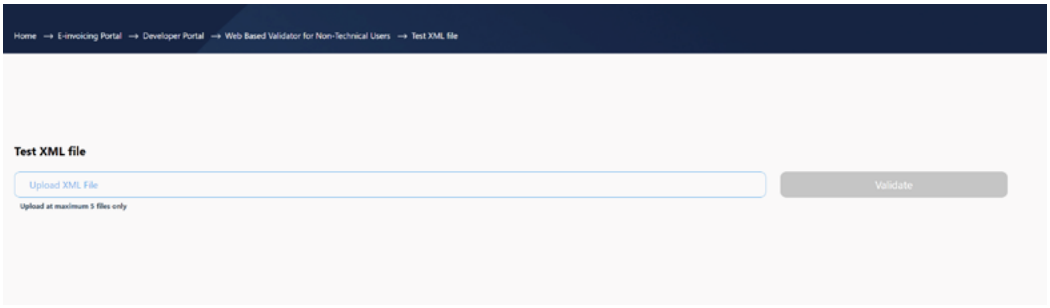
2.3.7 Using the Web Based Validator for Non-Technical Users

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.

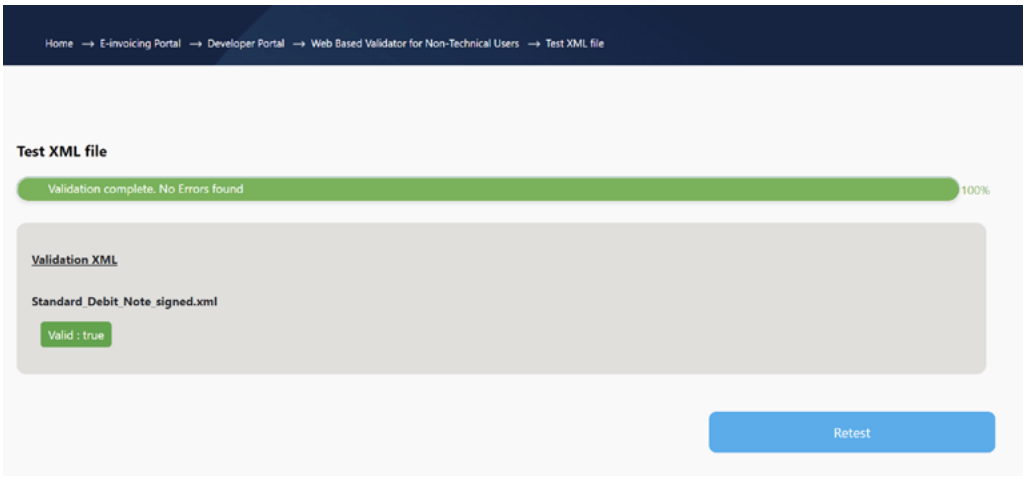
The process for validating XMLs from the Web Based Validator for Non-Technical Users page is as follows:

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



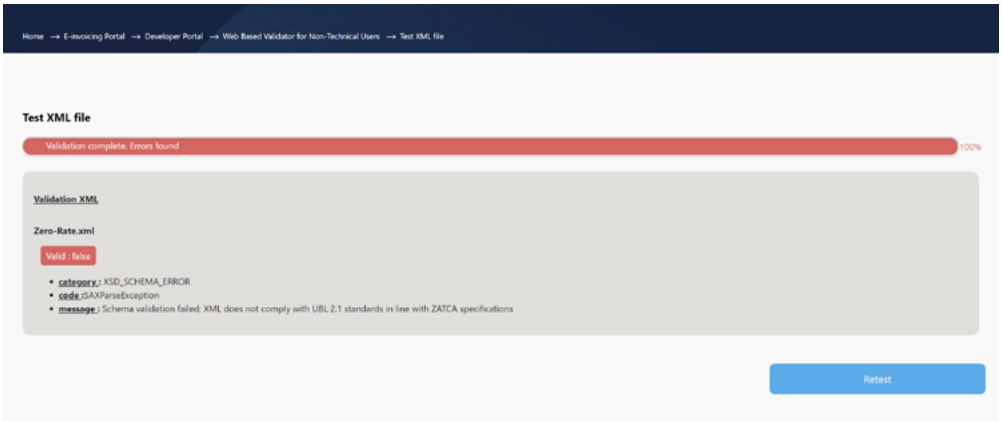
Uploading an XML file on the web based validator

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



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.

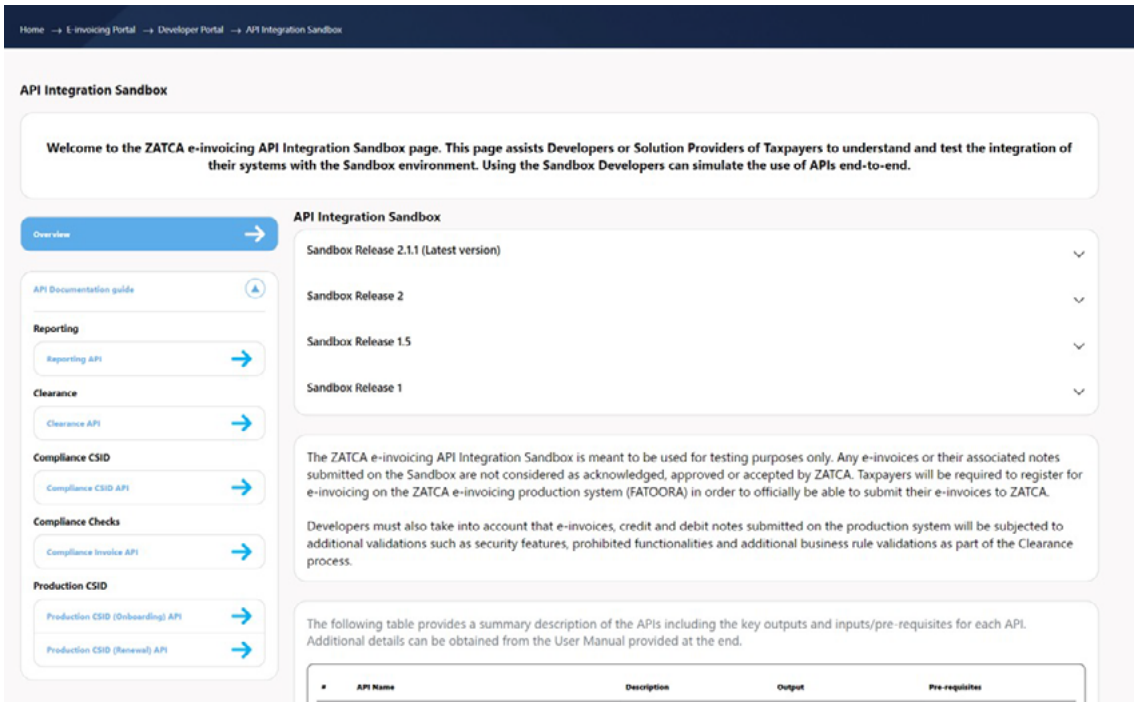
2.3.8 Accessing the Integration Sandbox Page

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.



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.

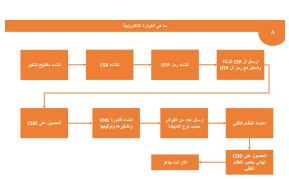
- Test request for a new Production CSID as part of renewal (requires a test Compliance CSID to be submitted)
- Test submission of documents for Clearance (requires a test Production CSID)
- Test submission of documents for Reporting (requires a test Production CSID)

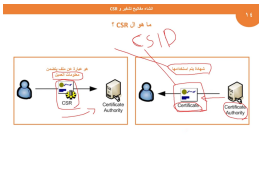
Although the Sandbox uses test CSIDs, it is important to note that the VAT Registration number used to obtain the test CSID must match with the VAT Registration number in the Renewal CSR and/or e-invoices, credit notes, debit notes and QR codes submitted in all subsequent calls made using that specific 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.

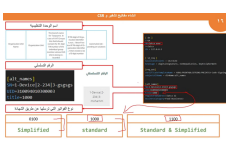
https://www.youtube.com/watch?v=vGUBzZKInZo
https://www.youtube.com/watch?v=-R20tuc9DR0
https://www.youtube.com/watch?v=xJ4sDrVtB0Q
https://www.youtube.com/watch?v=0E5DHw-4B90

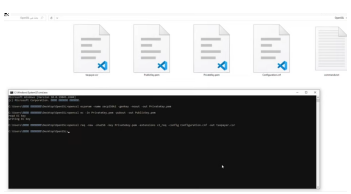
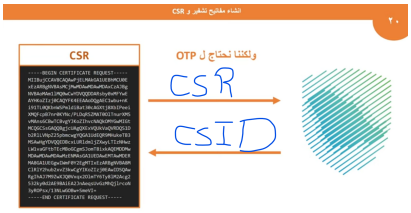
fer to the API Documentation through the following LINK.

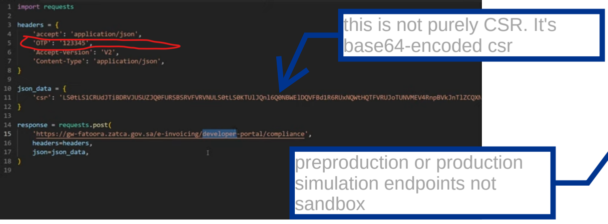
make sure to log-in in order to view the API documentation

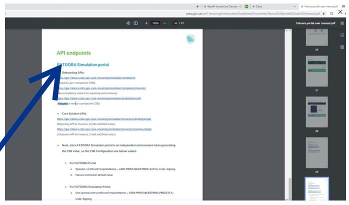


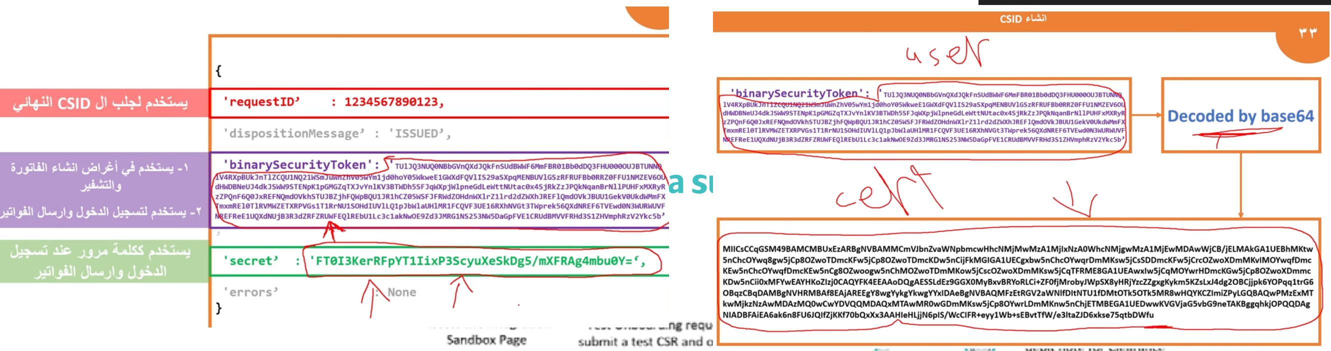





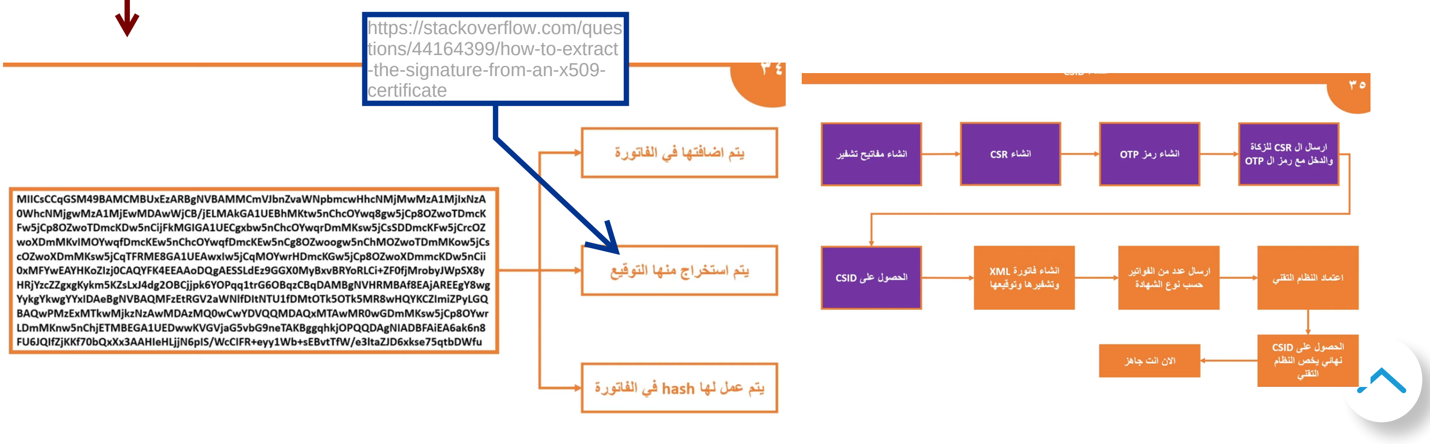



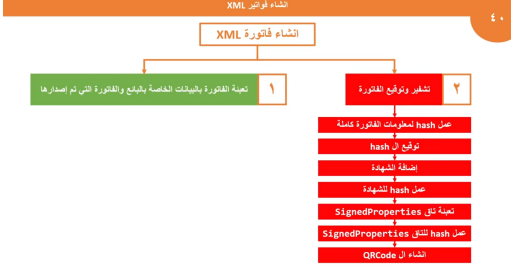






1. For Reporting and Clearance (testing the submission of E-invoices, credit and debit notes)
   - The users' E-Invoice Generation Solution (EGS) needs to generate compliant XML documents. For more details on generating compliant XML documents please refer to the XML Implementation Standards and the Data Dictionary (E-Invoice specifications (zatca.gov.sa)). It is also recommend to test the compliance using the Compliance and Enablement Toolbox SDK (Download SDK (zatca.gov.sa) or Portal based validator for non-technical users (Compliance and Enablement Toolbox portal).
   - 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).





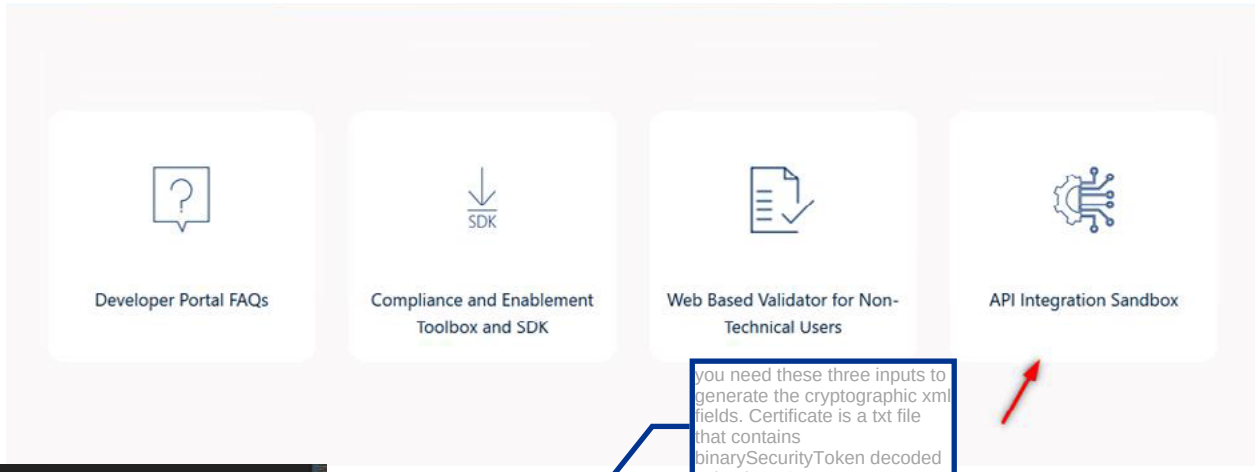
https://stackoverflow.com/questions/45873832/how-do-i-create-and-sign-certificates-with-pythons-pyopenssl

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

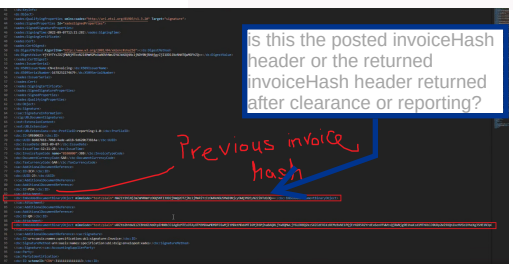
2. For Cryptographic Stamp Identifier (testing the Onboarding and Renewal processes).
   - The users' EGS needs to generate a compliant CSR to obtain a test CSID. For more details on generating a compliant CSR and CSID specifications please refer to (E-Invoice Security Features and Implementation Standards).
   - 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.

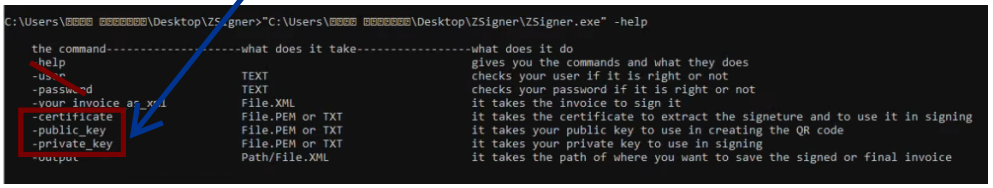
2.3.10.1 Compliance CSID

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

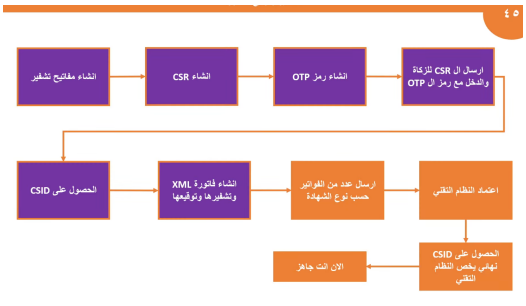


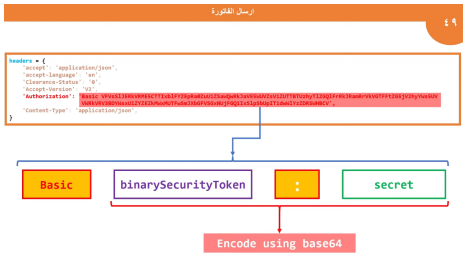
you need these three inputs to generate the cryptographic xml fields. Certificate is a txt file that contains binarySecurityToken decoded using base64

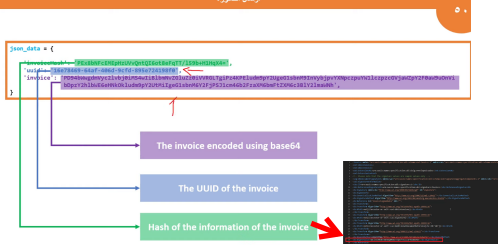




self-signed certificate from openssl



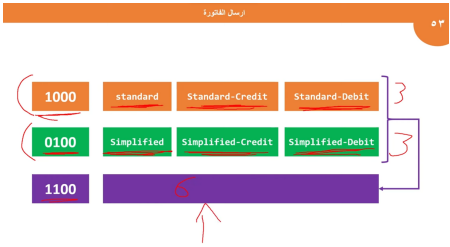


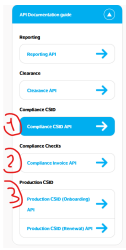


مداخلة :

https://gw-fatoora.zatca.gov.sa/e-invoicing/simulation/compliance/invoices

https://gw-fatoora.zatca.gov.sa/e-invoicing/simulation/core/csids/compliance

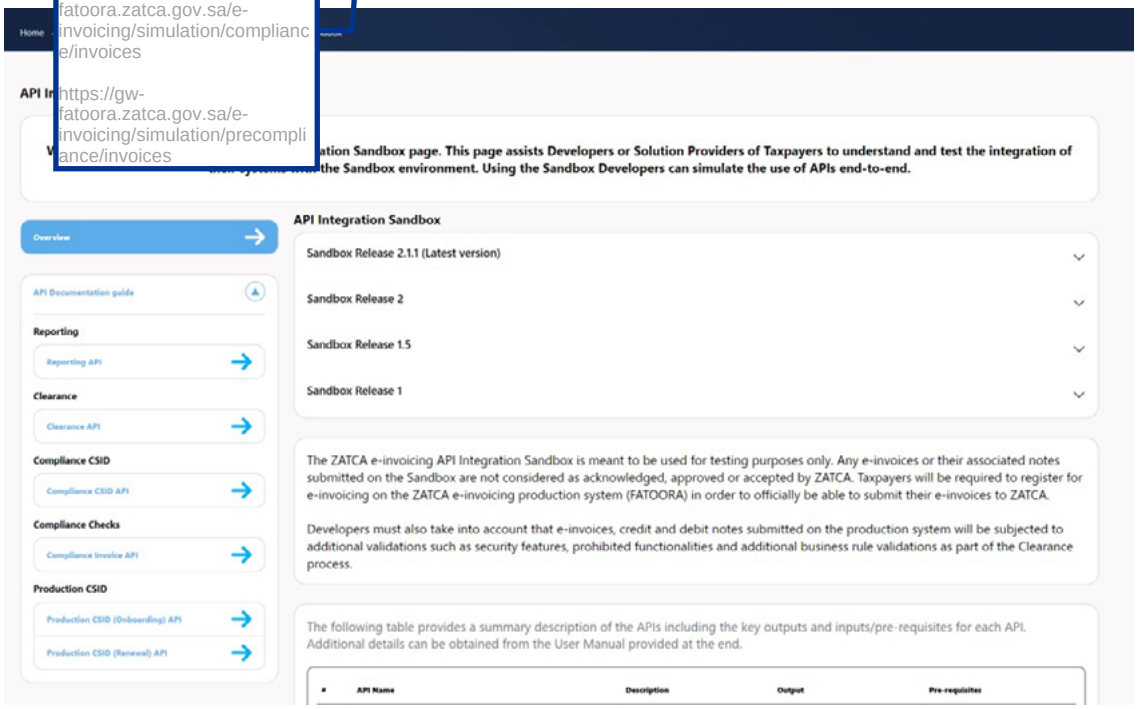




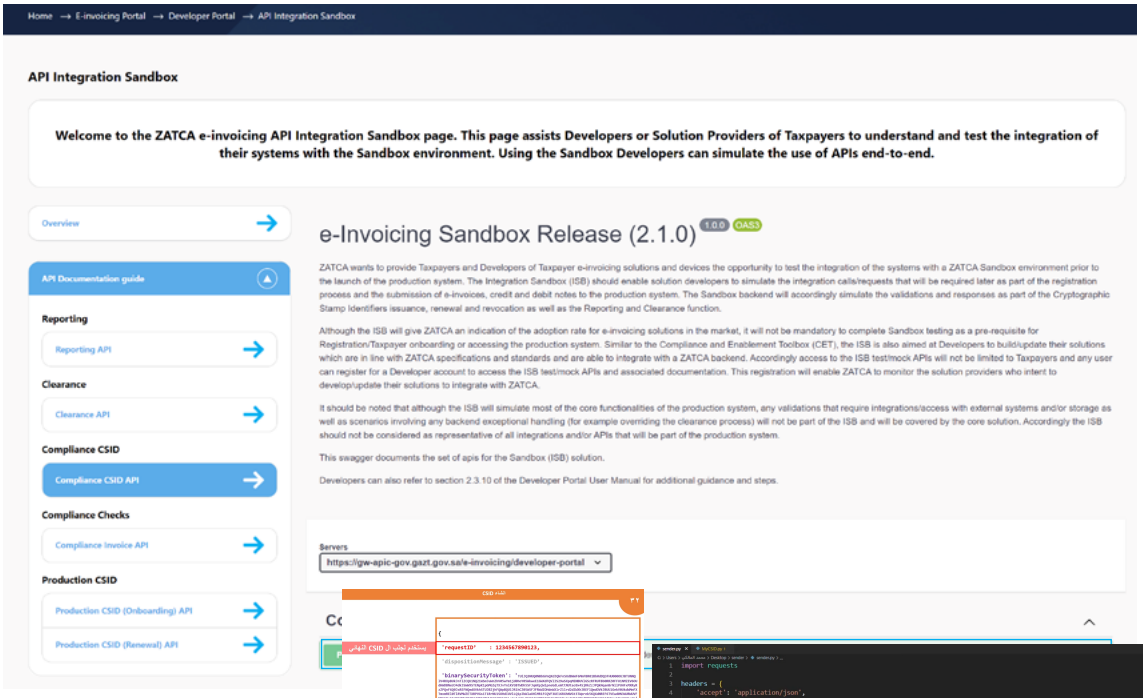
use this link instead of the above in the step of compliance check. You might try those:
https://gw-fatoora.zatca.gov.sa/e-invoicing/simulation/compliance/invoices

https://gw-fatoora.zatca.gov.sa/e-invoicing/simulation/precompliance/invoices

- St[illegible]mentation guides

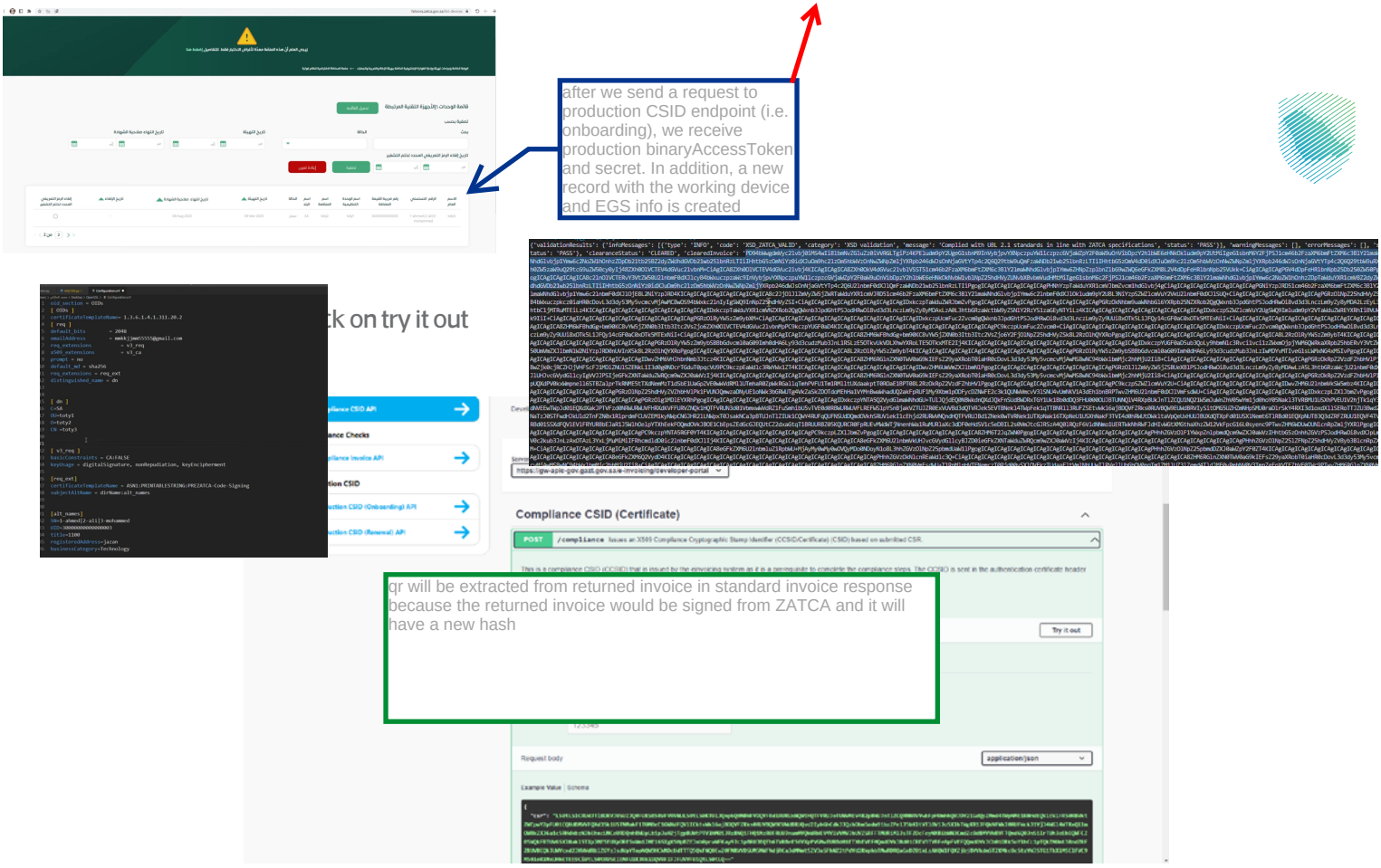


- Step 5: Click on Compliance CSID API drop-down

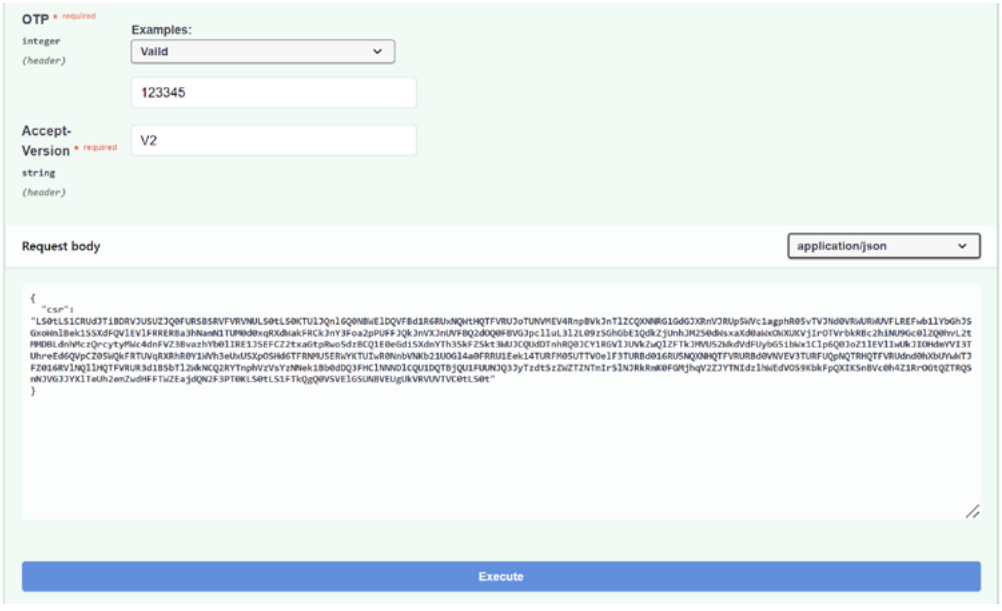


this value is replaced by requestID property in response of compliance CSID request (i.e. csr)

مثال عملي على جلب امر اعتماد الحل التقني وجلب ال CSID النهائي

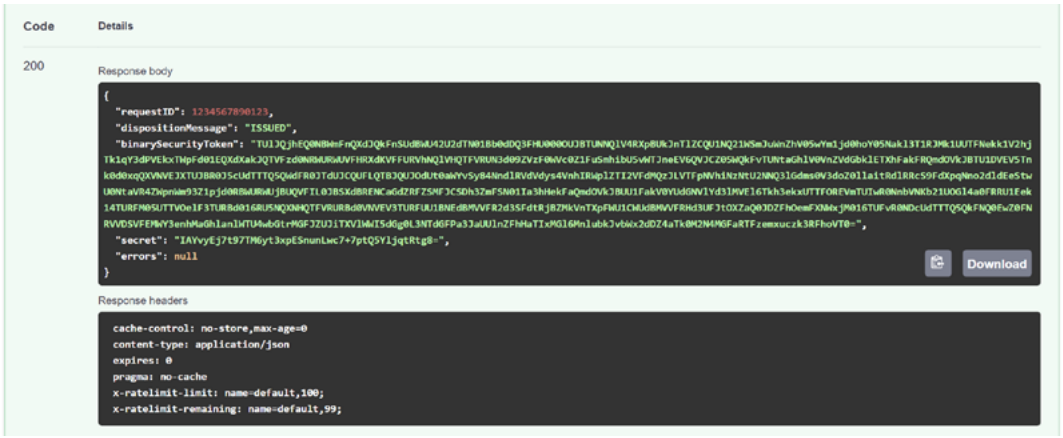


- Step 7: Insert valid OTP and fill the request body (CSR)
- Step 8: Click on Execute



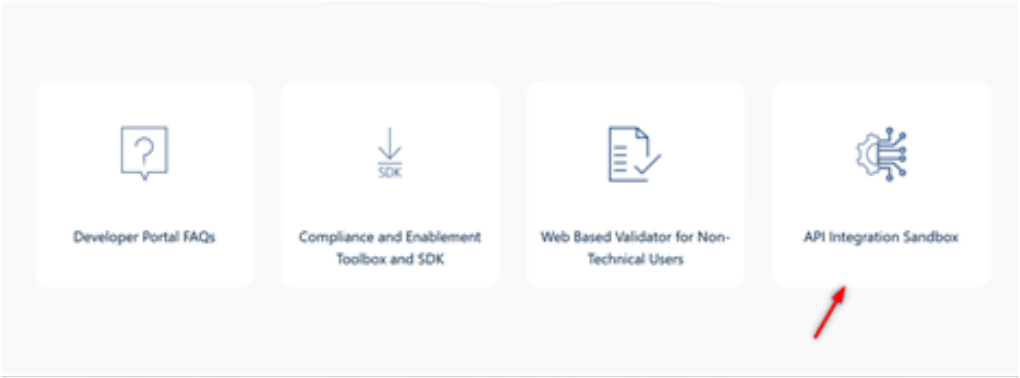
Note: V2 refers to the Version of the APIs used and should be mentioned in the API calls (V2 is currently the only valid version).

- Result (200)

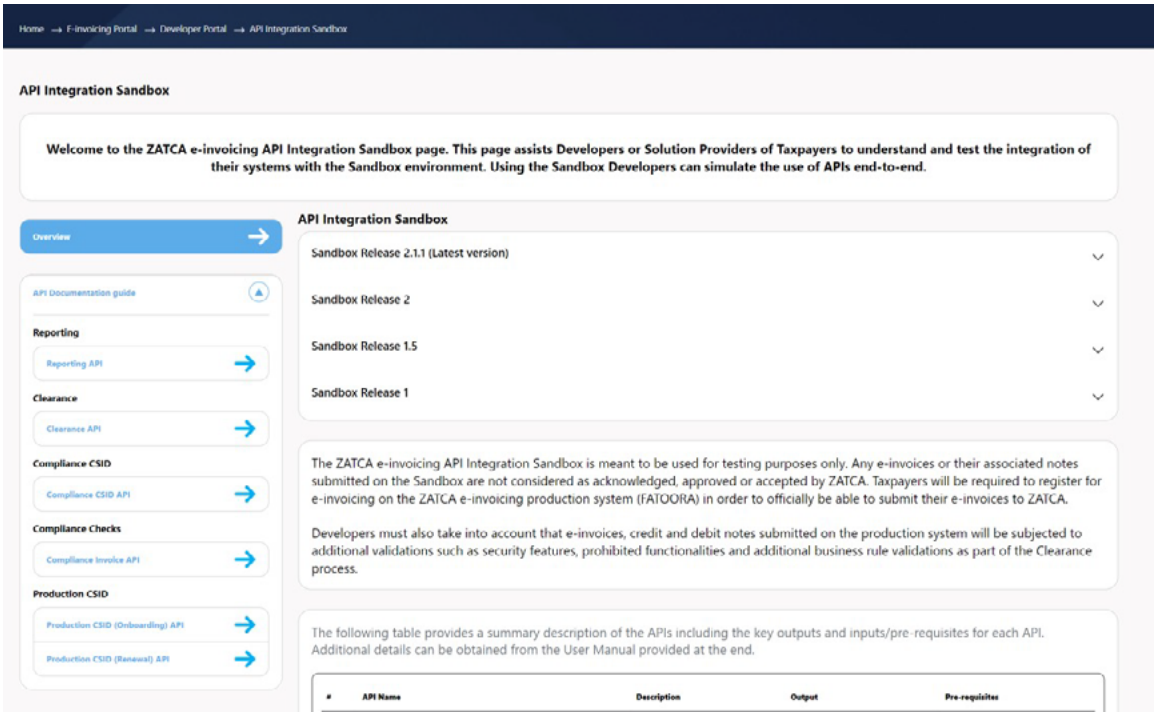


2.3.10.2 Compliance Invoice API

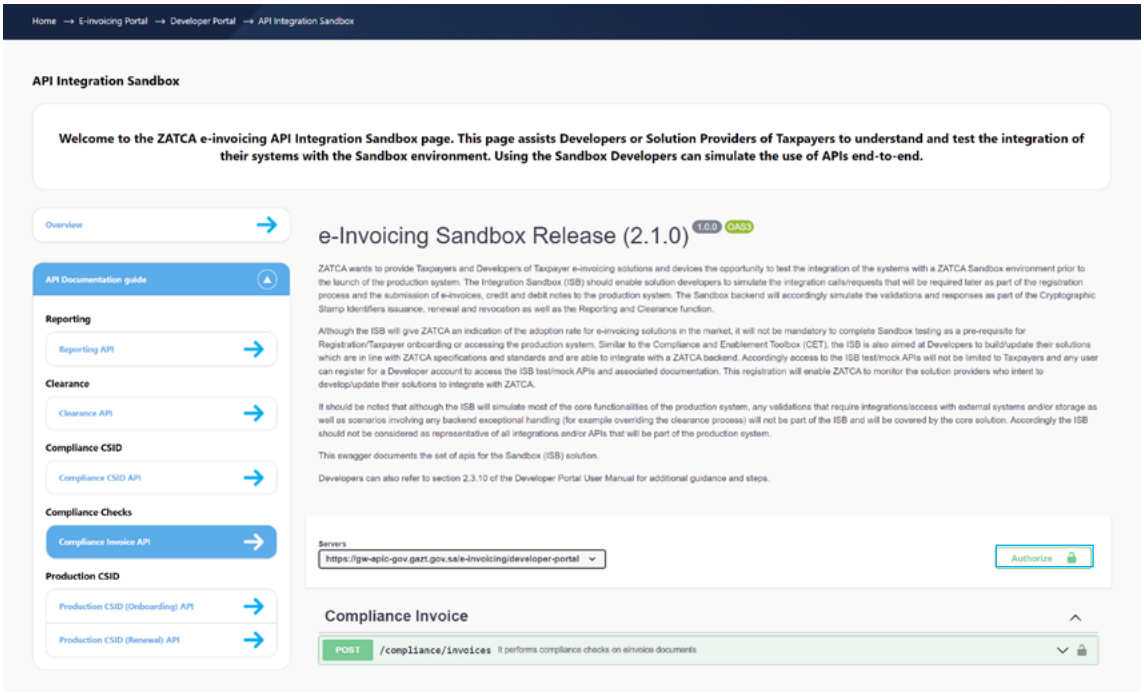
- Step 1: Navigate to Developer Portal link
- Step 2: Login with correct credentials
- Step 3: Navigate to API Integration Sandbox



- Step 4: Click on API documentation guide



- Step 5: Click on Authorize Compliance Invoice API



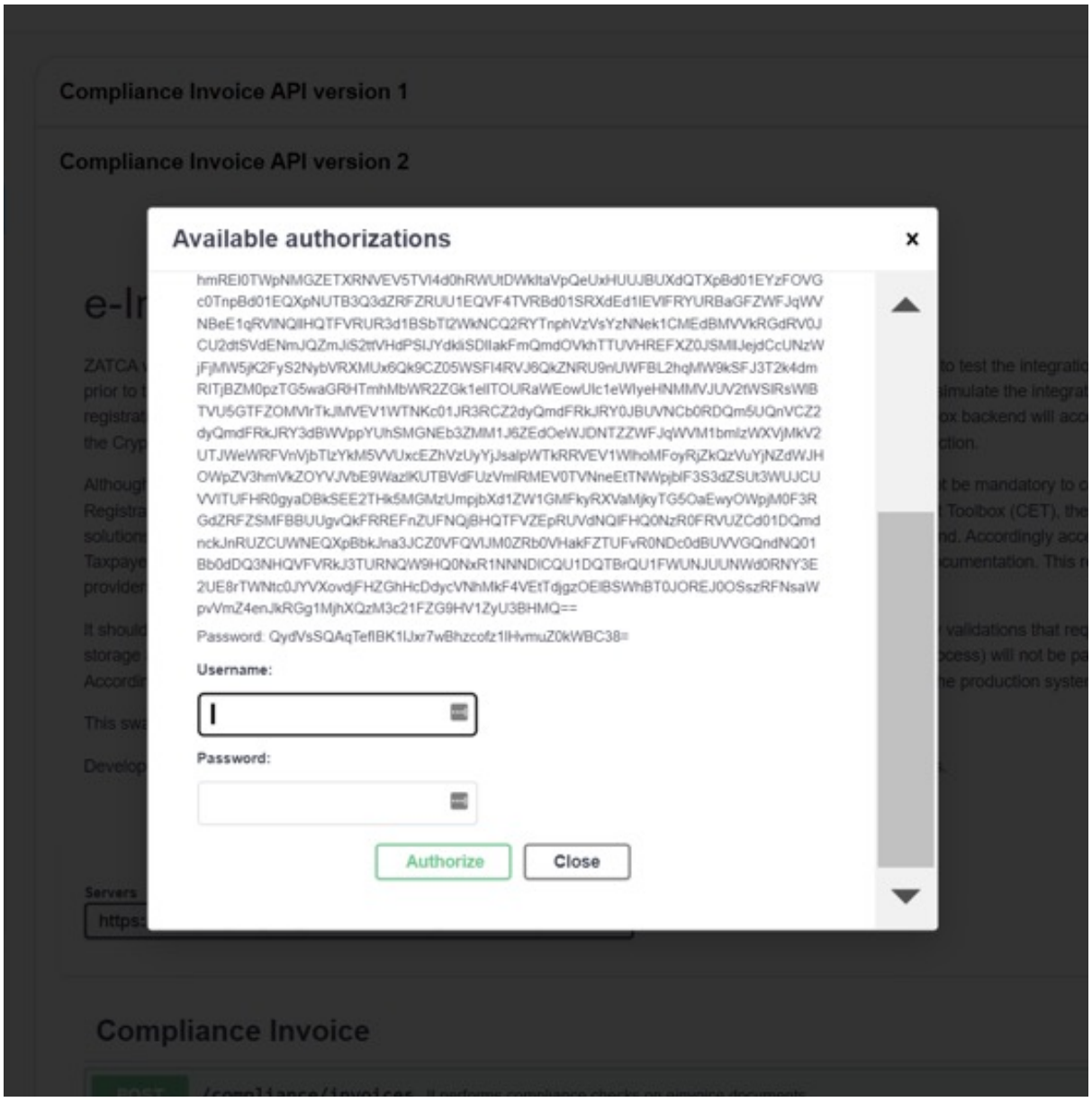
Note: This step is to be repeated on the number of invoices to be sent as part of the compliance checks.

- Step 6:

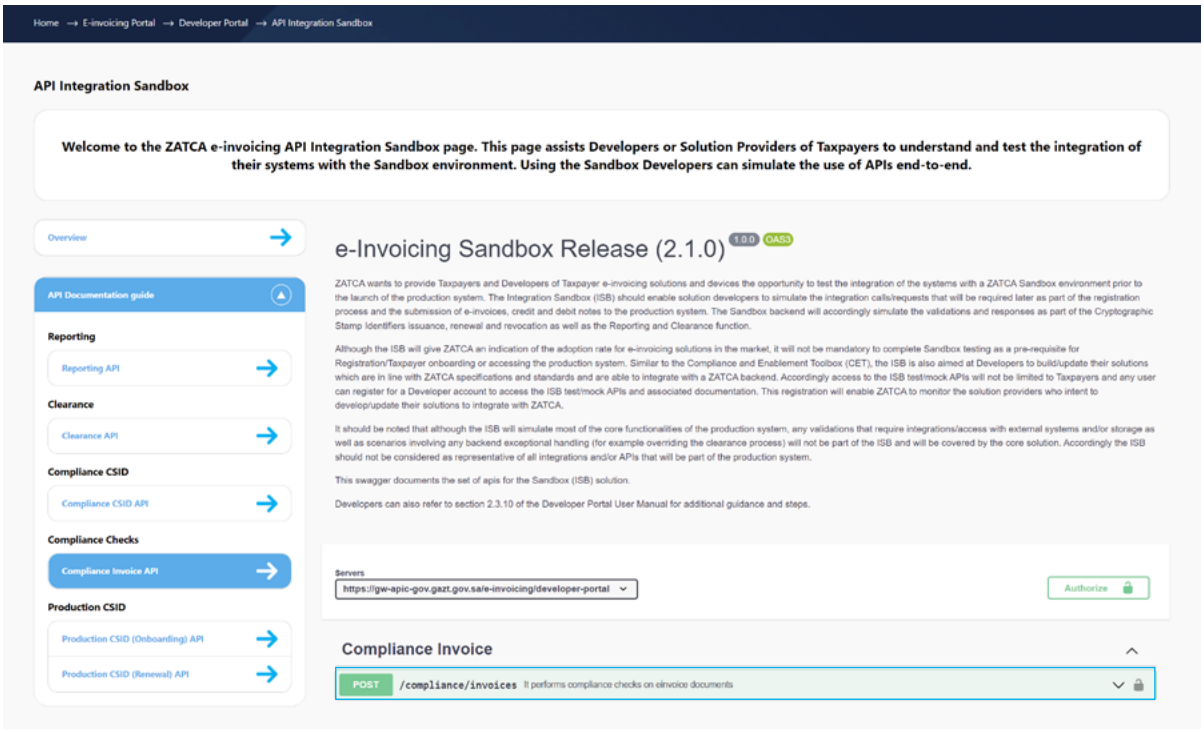
For the Sandbox, use the sample dummy Username and Password provided to you on the Authorization screen.

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

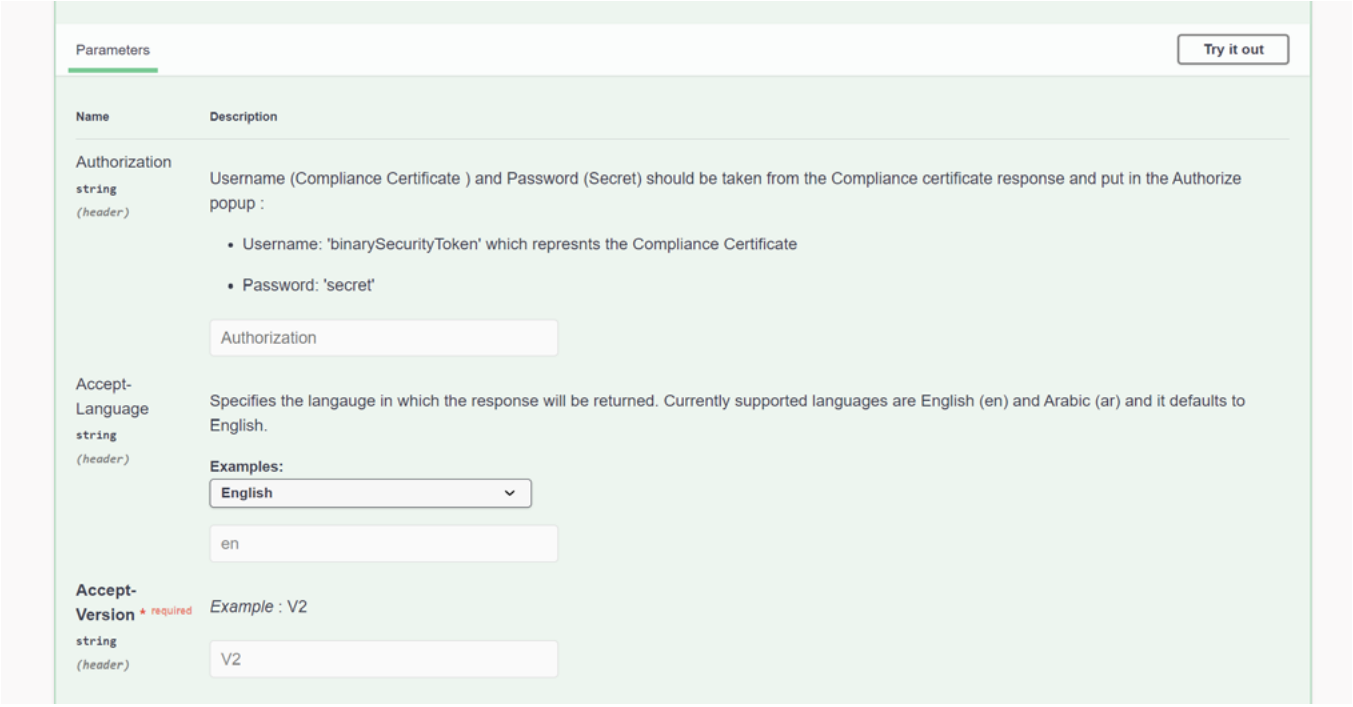
Please refer to Chapter 3 of this document for further details.



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

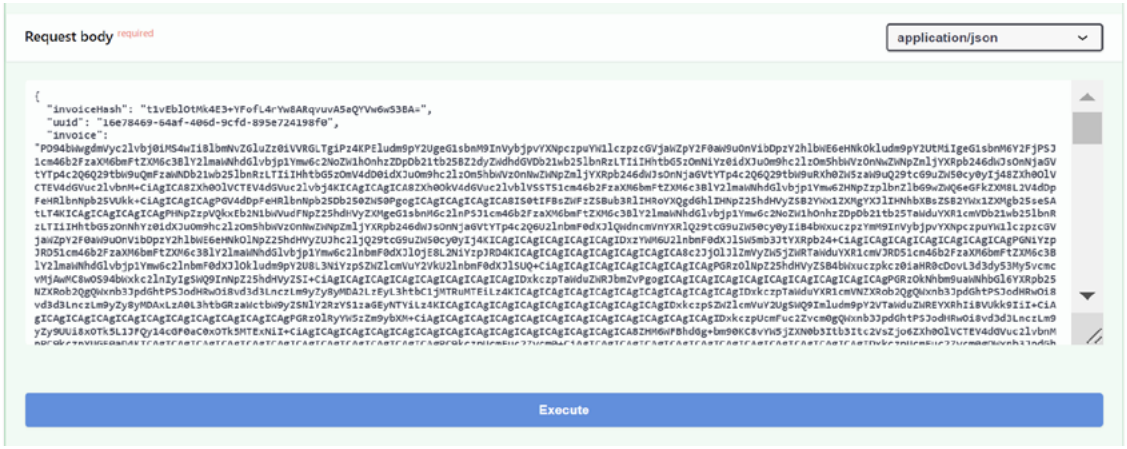


- Step 8: Click on Try it out button

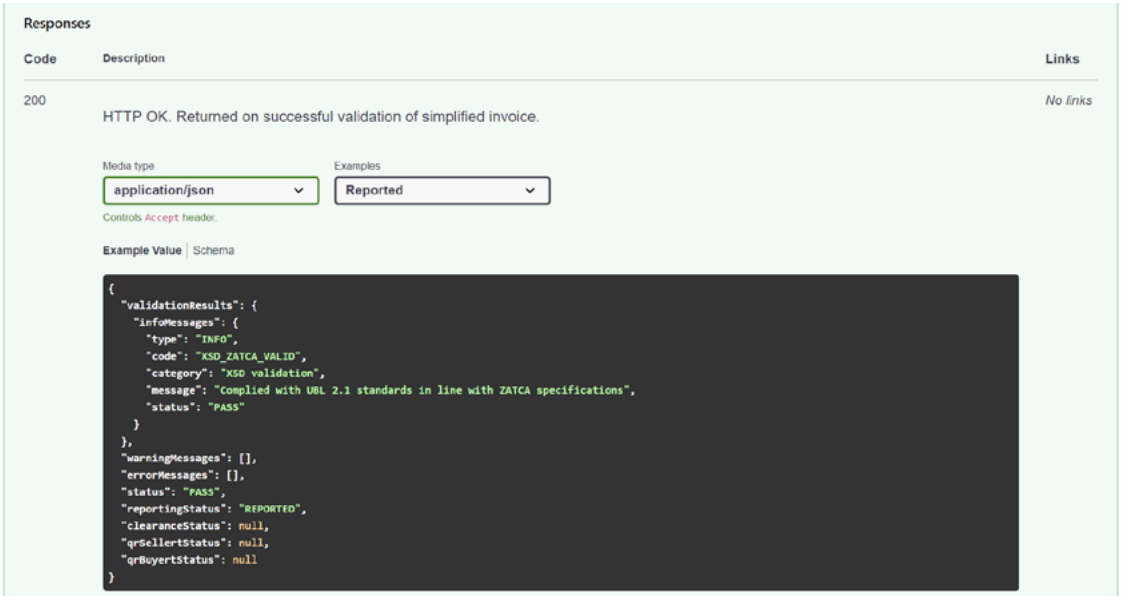


Note: V2 refers to the Version of the APIs used and should be mentioned in the API calls (V2 is currently the only valid version).

- Step 9: Fill the request body (invoice hash, UUID, encoded XML invoice)
- Step 10: Click on Execute

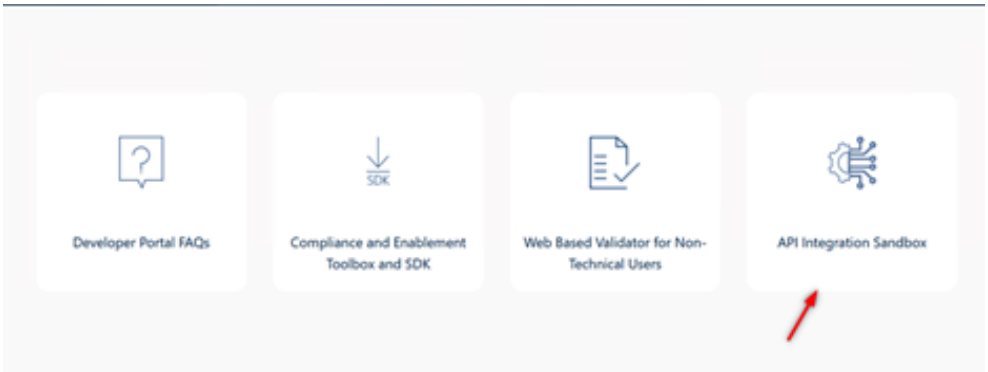


- Result (200)

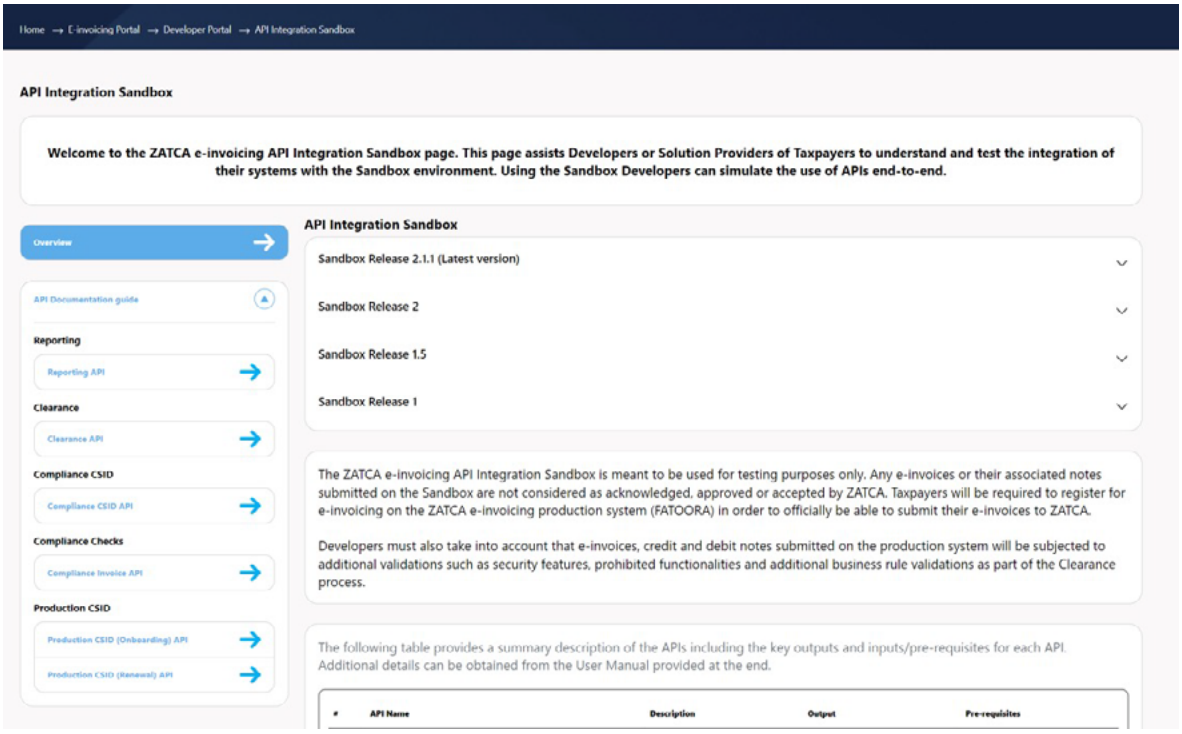


2.3.10.3 Production CSID (Onboarding) API

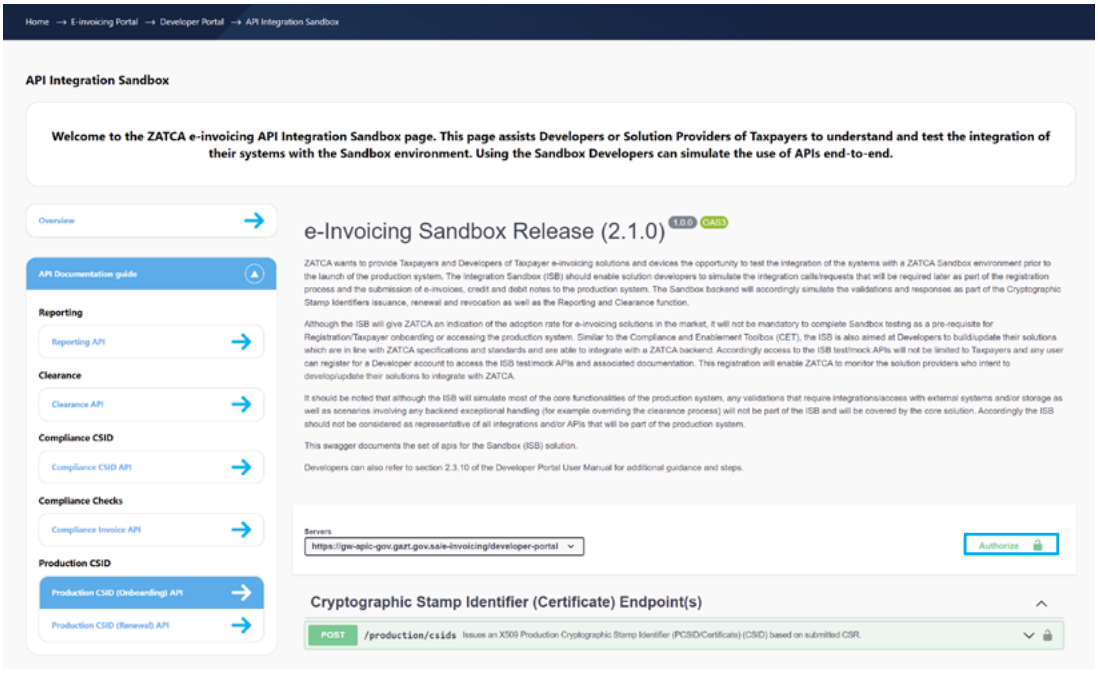
- Step 1: Navigate to Developer Portal link
- Step 2: Login with correct credentials
- Step 3: Navigate to API Integration Sandbox



- Step 4: Click on API documentation guide



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

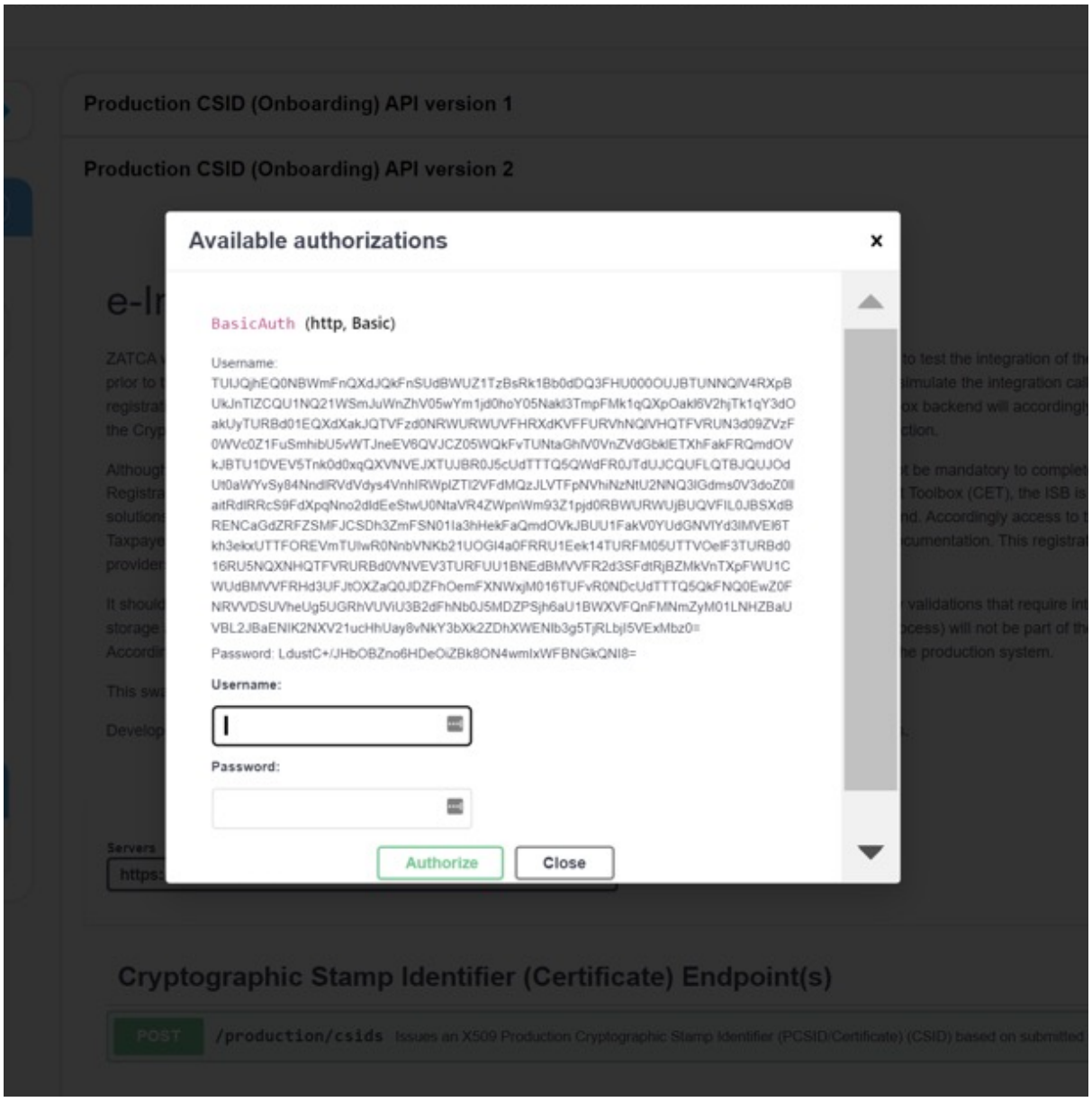


- Step 6:

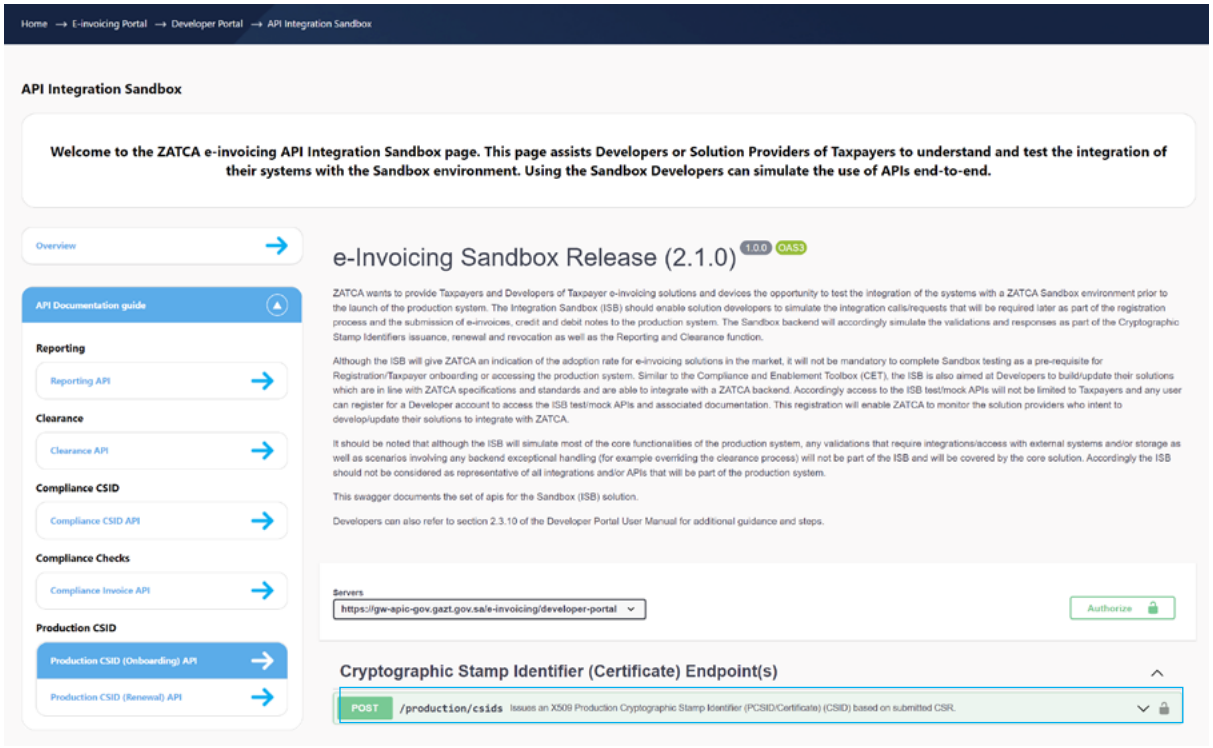
For the Sandbox, use the sample dummy Username and Password provided to you on the Authorization screen.

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

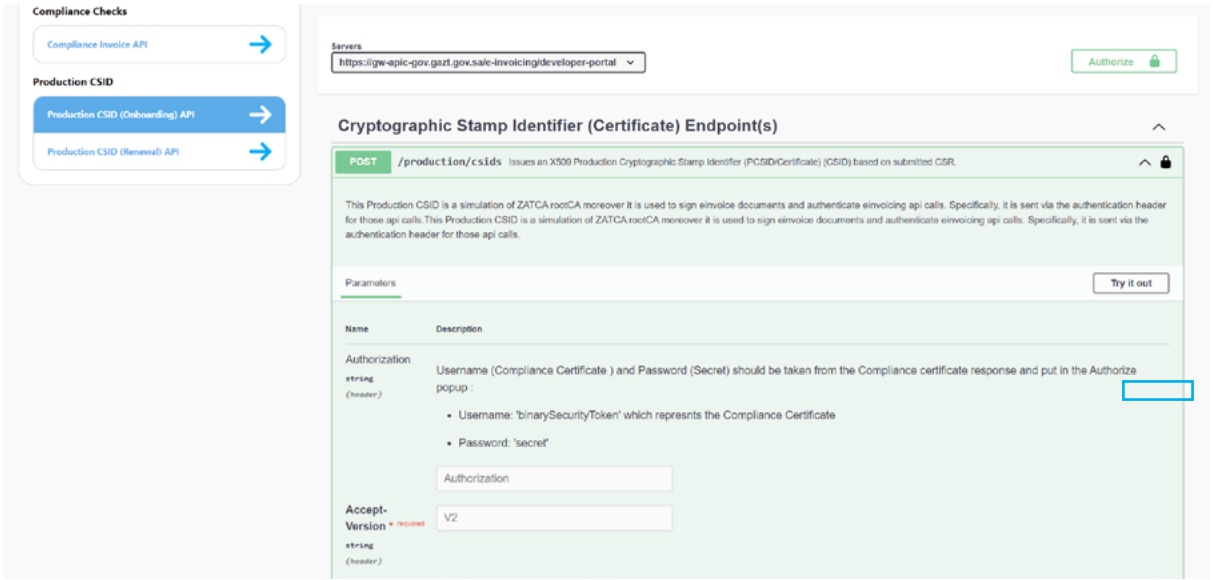
Please refer to Chapter 3 of this document for further details.



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

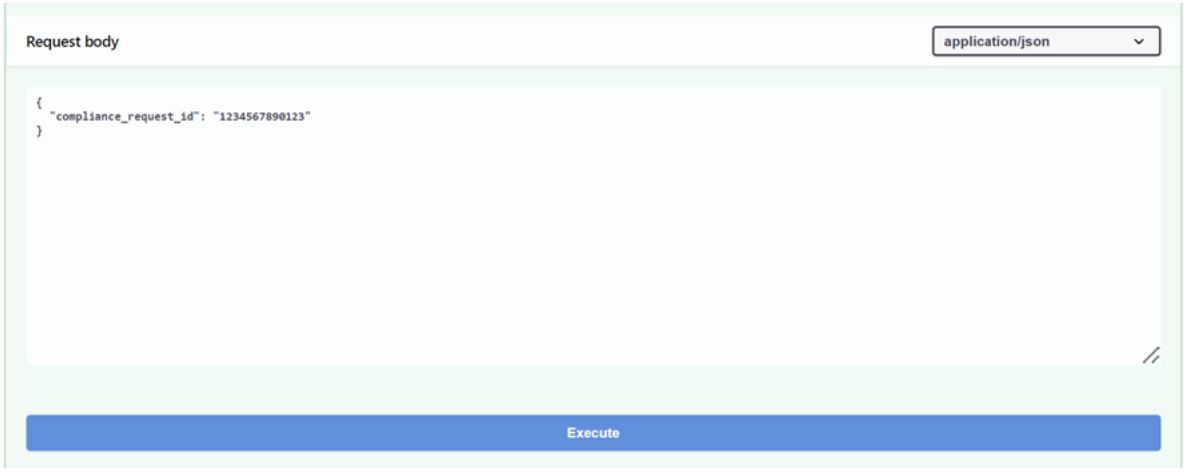


- Step 8: Click on Try it now button

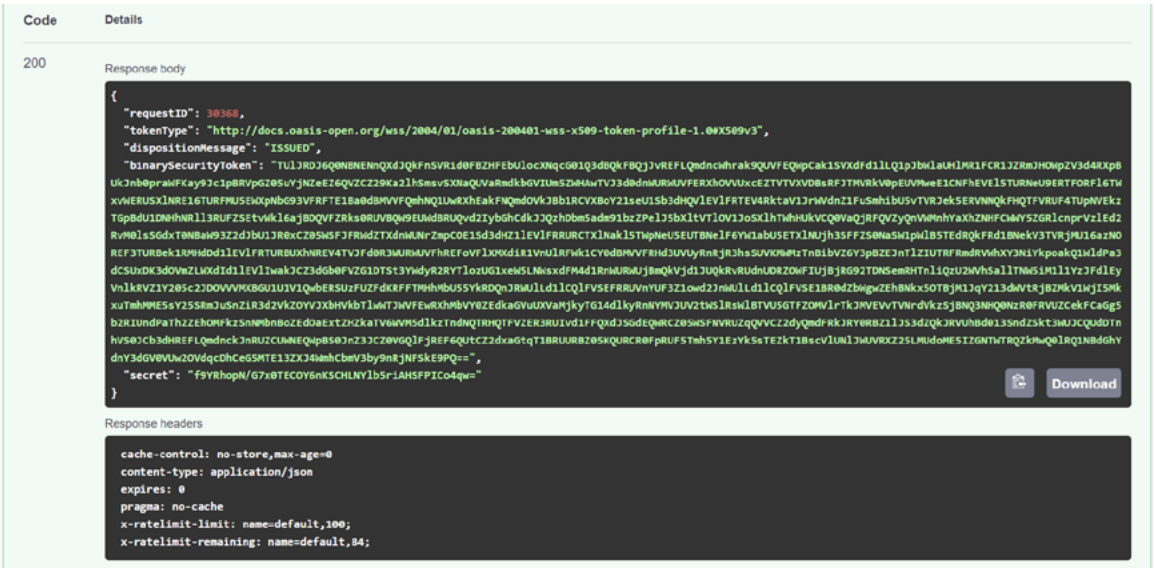


Note: V2 refers to the Version of the APIs used and should be mentioned in the API calls (V2 is currently the only valid version).

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

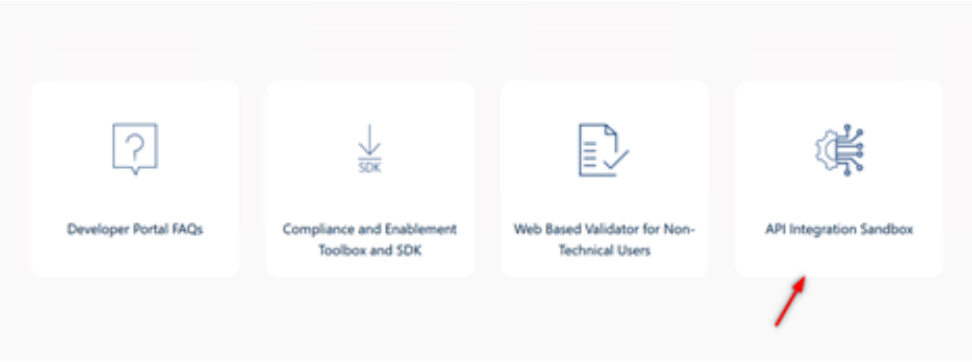


- Result (200)

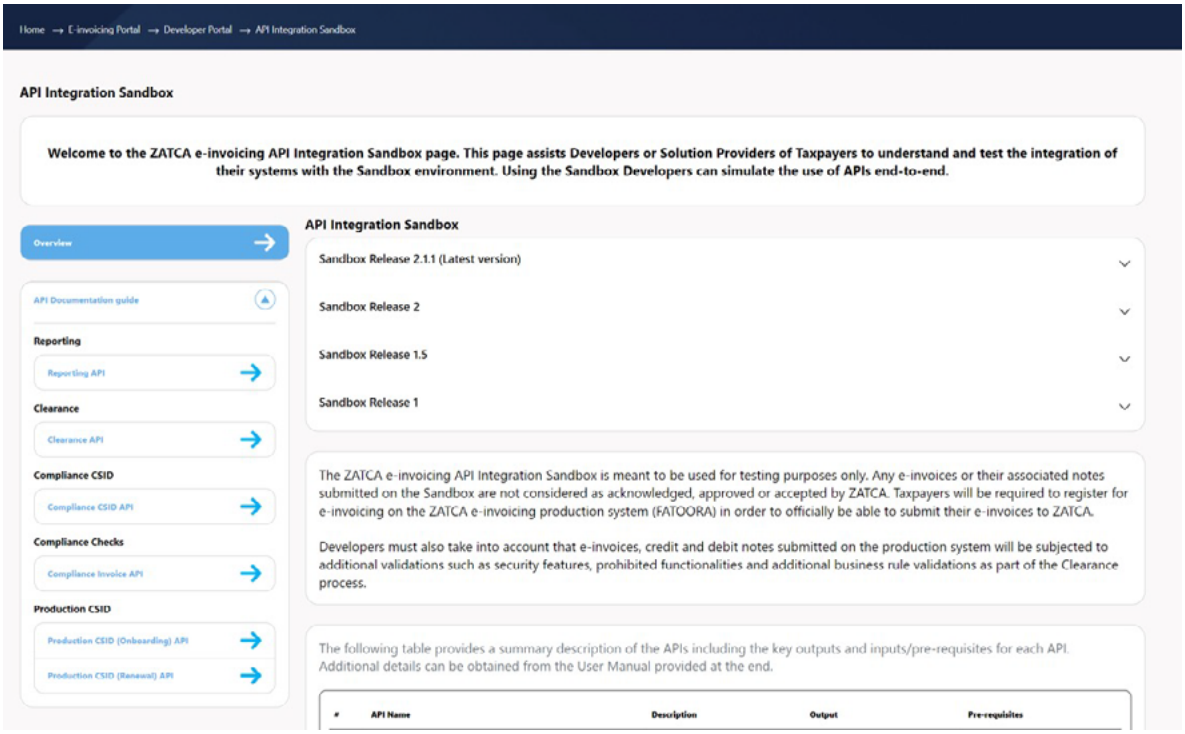

Code
Details
200
Response body
{
"requestID": 30368,
"tokenType": "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3",
"dispositionMessage": "ISSUED",
"binarySecurityToken": "TUlJRDJ6Q0NBNENnQXdJQkFnSVR1d0FBZHFEbUlocXNqcG01Q3dBQkFBQjJvREFLQmdncWhrak9QUVFEQWpCak1SVXdFd1lLQ1pJbWlaUHlMR1FCR1JZRmJHOWpZV3d4RXpB
UkJnb0praWFKay9Jc1pBRVpGZ05uYjNZeEZ6QVZCZ29Ka2lhSmsvSXNaQUVaRmdkbGVIUm5ZWHAwTVJ3d0dnWURWUVFERXhOVVUxcEZTVTVXVDBsRFJTMVRkV0pEUVMweE1CNFhEVEl5TURNeU9ERTFORl6TW
xvWERUSXlNRE16TURFMU5EWXpNbG93VFRFTE1Ba0dBMVVFQmhNQ1UwRXhEakFNQmdOVkJBb1RCVXBoY21seU1Sb3dHQVlEVlFRTEV4RktaV1JrWVdnZ1FuSmhibU5vTVRJek5ERVNNQkFHQTFVRUF4TUpNVEkz
TGpBdU1DNHhNRll3RUFZSEtvWkl6ajBDQVFZRks0RUVBQW9EUWdBRUQvd2IybGhCdkJQzh0bm5adm91bzZPel35bXltVTlOV1JoSXlhTWhHUkVCQ0VaQjRFQVZyQnVWMnhYaXhZNHFCWWY5ZGRlcnprVzlEd2
RvM0lsSGdxT0NBaW93Z2dJbU1JR0xCZ05WSFJFRWdZTXdnWUNrZmpCOE1Sd3dHZ1lEVlFRRURCTXlNak15TWpNeU5EUTBNelF6YW1abU5ETXlNUjh3SFFZS0NaSW1pWlB5TEdRQkFRd1BNekV3TVRjMU16azNO
REF3TURBek1RMHdDd1lEVlFTURBUXhNREV4TVJFd0R3WURWUVFhREFoVFlXMXdiR1VnUlRFWk1CY0dBMVVFRHd3UVUyRnRjR3hsSUVKMWMzTnBibVZGY3pBZEJnTlZIUTRFRmdRVWhXY3NiYkpoakQ1WldPa3
dCSUxDK3dOVmZLWXdId1lEVlIwakJCZ3dGb0FVZG1DTSt3YWdyR2RYTlozUG1xeW5LNWsxdFM4d1RnWURWUj8mQkVjdlJUQkRvRUdnUDRZOWFIUjBjRG92TDNSemRHTnliQzU2WVhSallTNW5iM1l1YzJFdlEy
VnlkRVZ1Y205c2JDOVVVMXBGU1U1V1QwbERSUzFUZFdKRFFTMHhMbU55YkRDQnJRWUlldllCQlFVSEFRRUVnYUF3Z1owd2JnWUlldllCQlFVSE1BR0dZbWgwZEhBNkx5OTBjM1JqY213dWVtRjBZMkV1WjISMk
xuTmhMME5sY25SRmJuSnZiR3d2VkZOYVJXbHVkbTlwWTJWVFEwRXhMbVY0ZEdkaGVuUXVaMjkyTG14dlkyRnNYMVJUV2tWSlRsWlBTVU5GTFZOMVlrTkJMVEVVTVNrdVkzSjBNQ3NHQ0NzR0FRVUZCekFCaGg5
bzRIUndPaTh2ZEhOMFkzSnNMbnBoZEdOaExtZHZkaTV6WVM5dlkzTndNQTRMQTFVZER3RUIvd1FFQXdJSGdEQWRCZ05WSFNVRUZqQVVCZ2dyQmdFRkJRY0RBZilJS3dzQkJRVUhBd013SndZSktJWUJCQUdDTn
hvS0JCb3dHREFLQmdncktnRUZCUWNEQWpBS0JnZ3JCZ0VGQlFjREF6QUtCZ2dxaGtqT1BRUURBZ05KQURCR0FpRUF5TmhSY1EzYk5sTEZkT1BscVlUNlJWUVRXZ25LMUdoME5IZGNTWTRQZkMwQ0lRQ1NBdGhY
dnY3dGV0VUw2OVdqcDhCeG5MTE13ZXJ4WmhCbmV3by9nRjNFSkE9PQ==",
"secret": "f9YRhopN/G7x0TECOY6nK5CHLNYlbSriAH5FPICo4qw="
}
Download
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

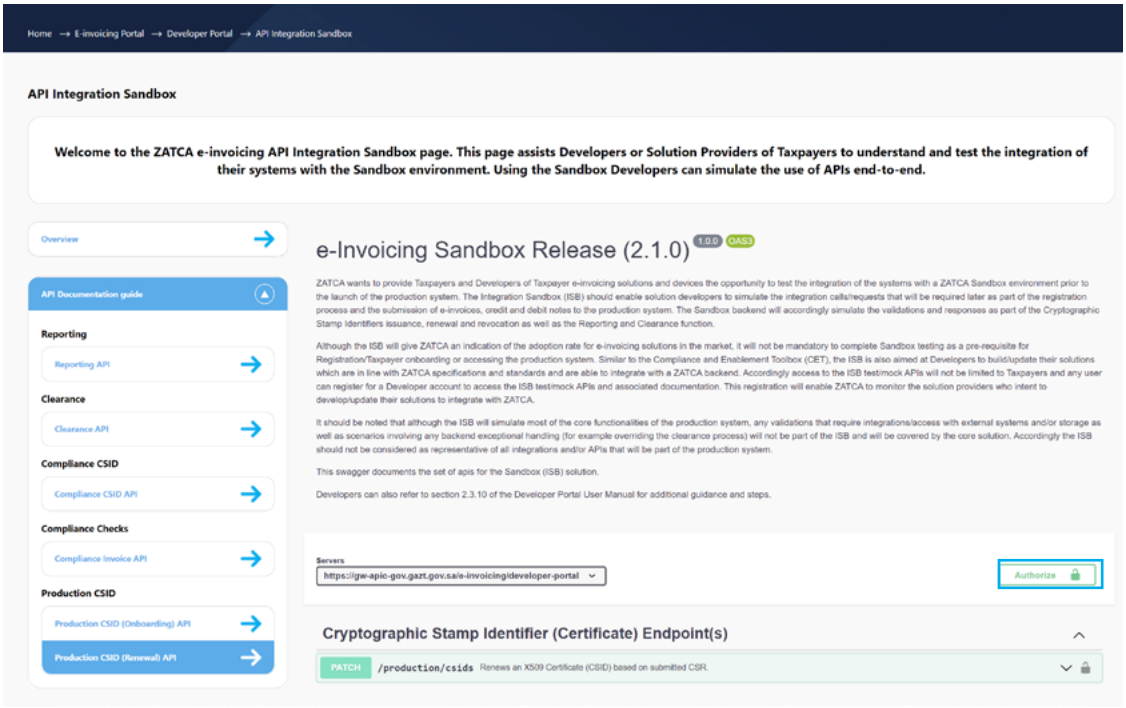
- Step 1: Navigate to Developer Portal link
- Step 2: Login with correct credentials
- Step 3: Navigate to API Integration Sandbox



- Step 4: Click on API documentation guide



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

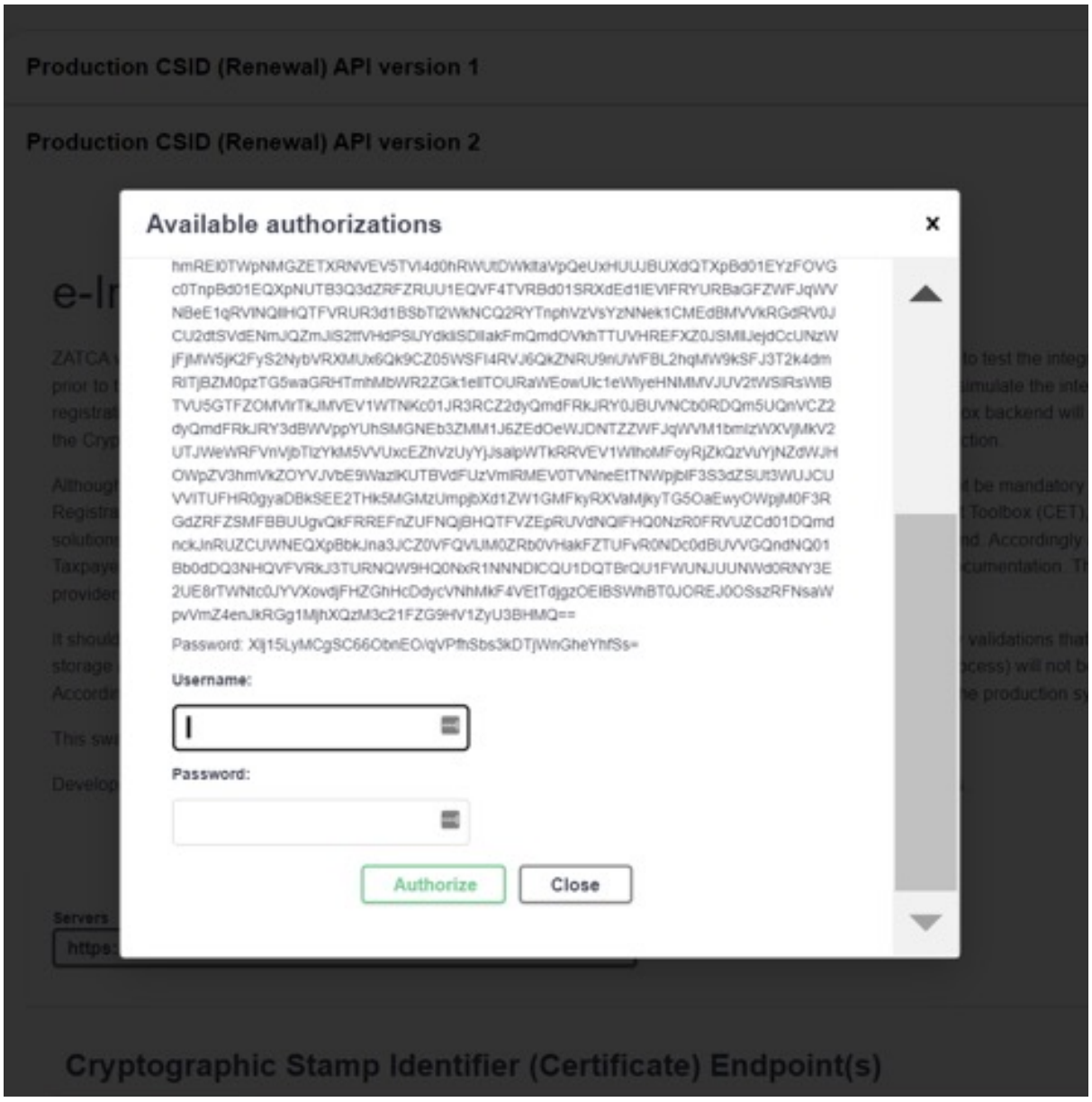


- Step 6:

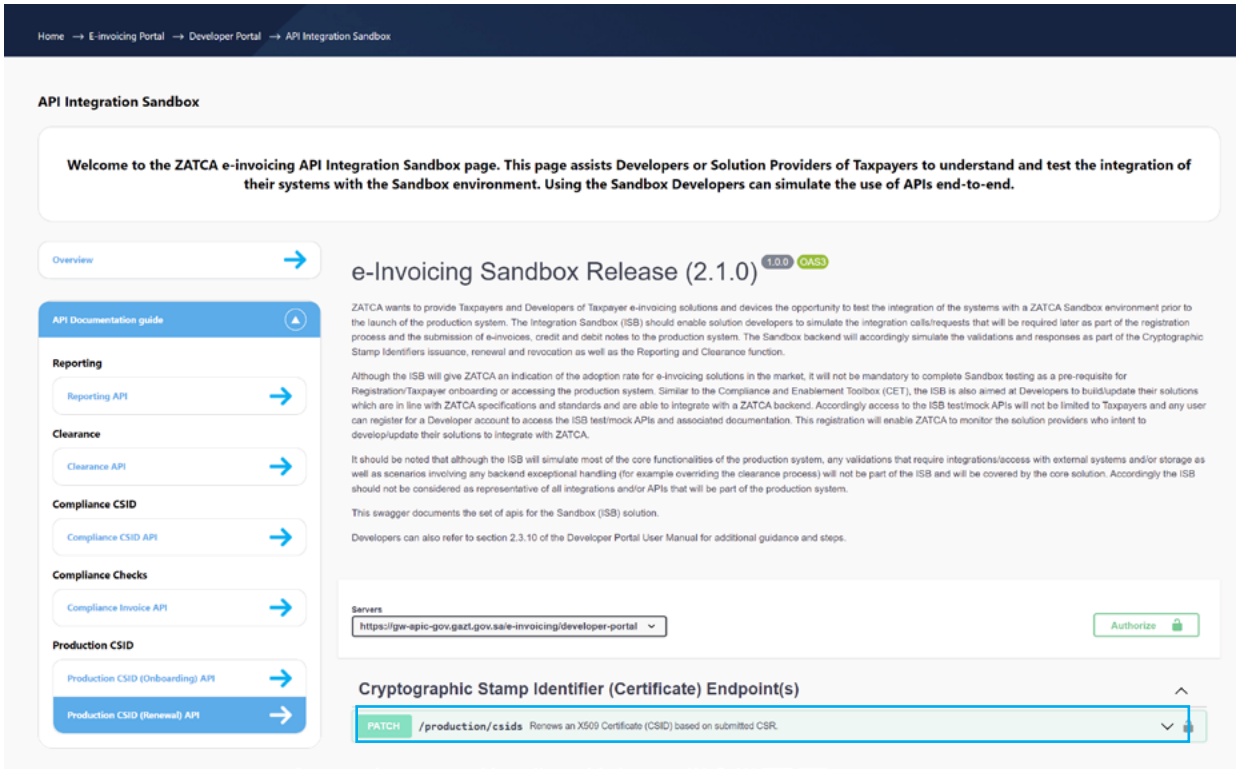
For the Sandbox, use the sample dummy Username and Password provided to you on the Authorization screen.

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

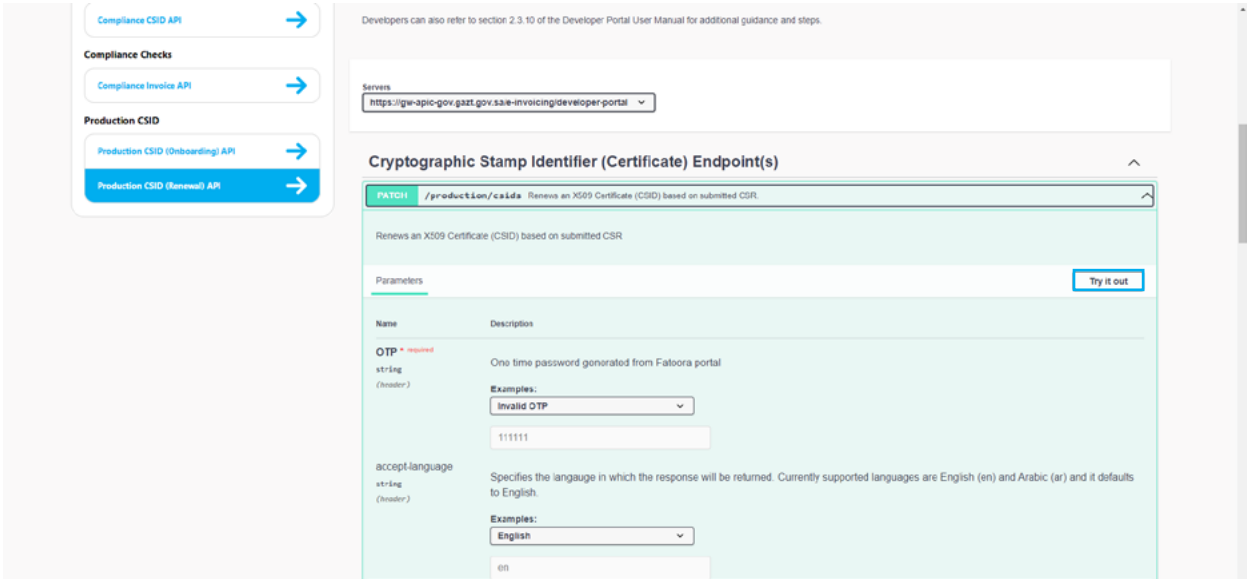
Please refer to Chapter 3 of this document for further details.



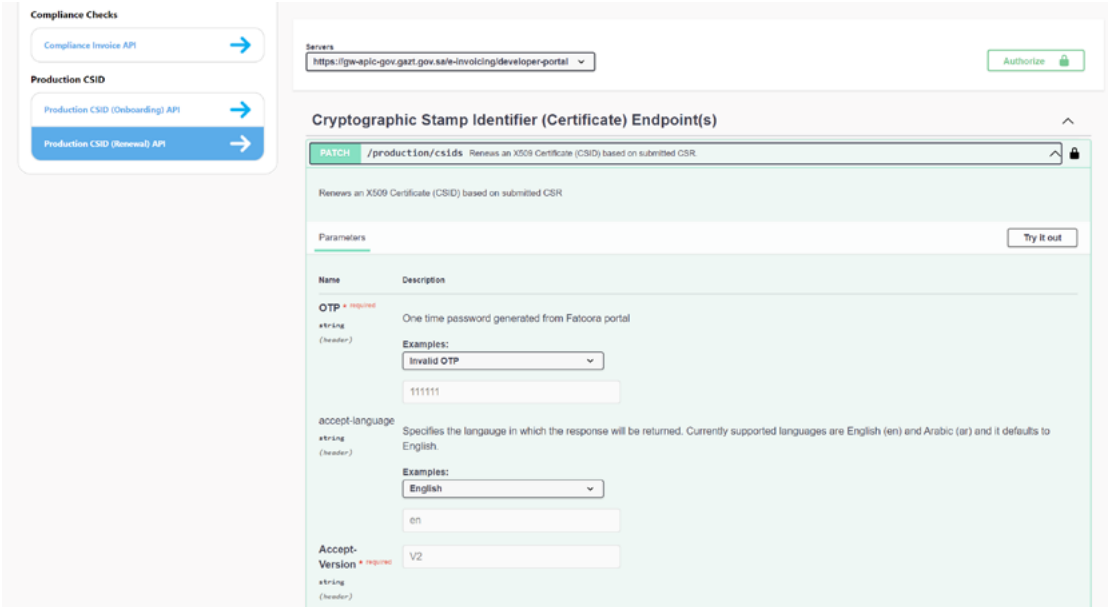
- Step 7: Click on Production CSID (renewal) API drop-down



- Step 8: Click on Try it now

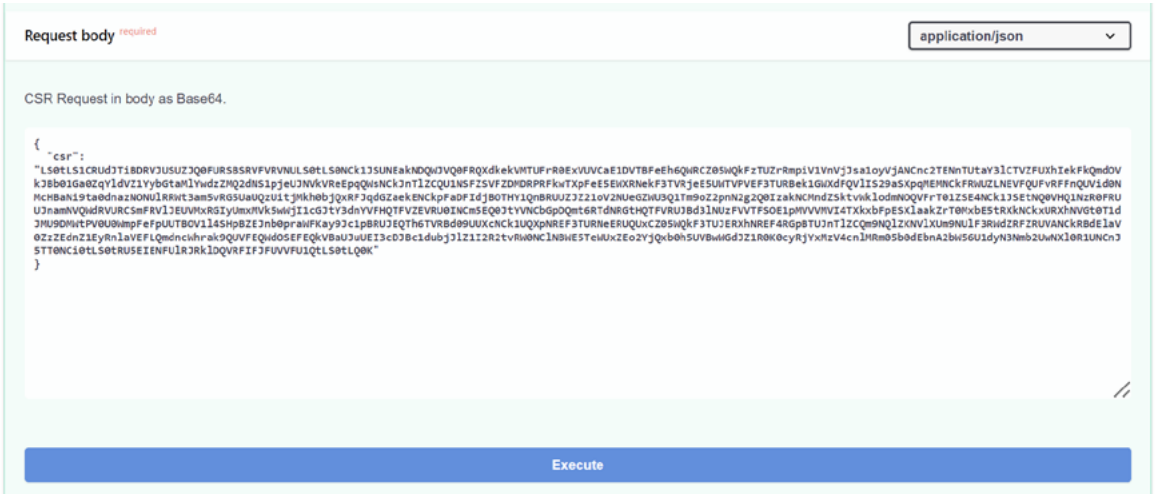


- Step 9: Insert valid OTP

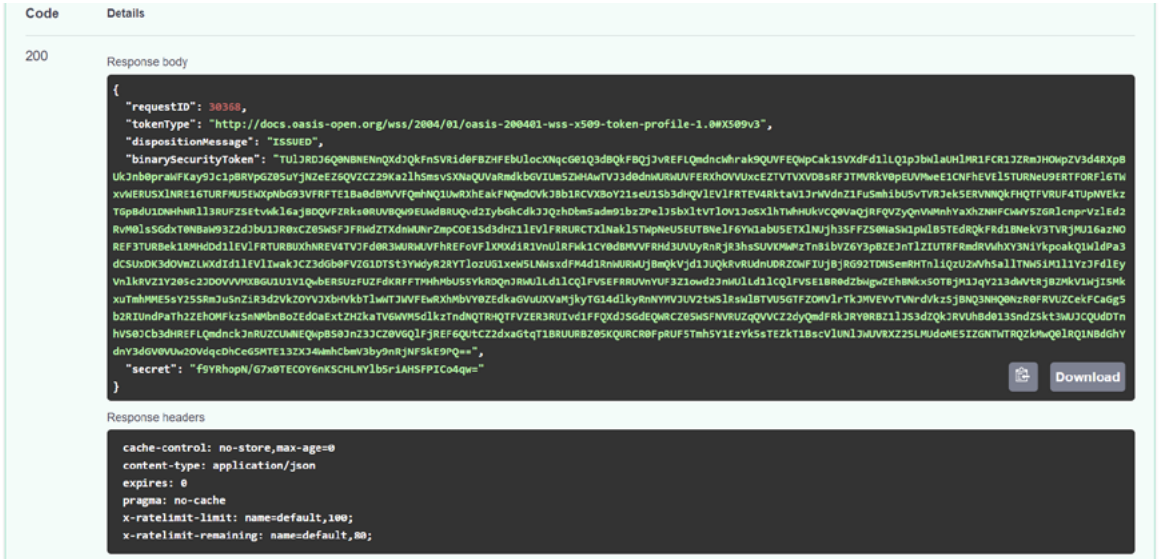


Note: V2 refers to the Version of the APIs used and should be mentioned in the API calls (V2 is currently the only valid version).

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



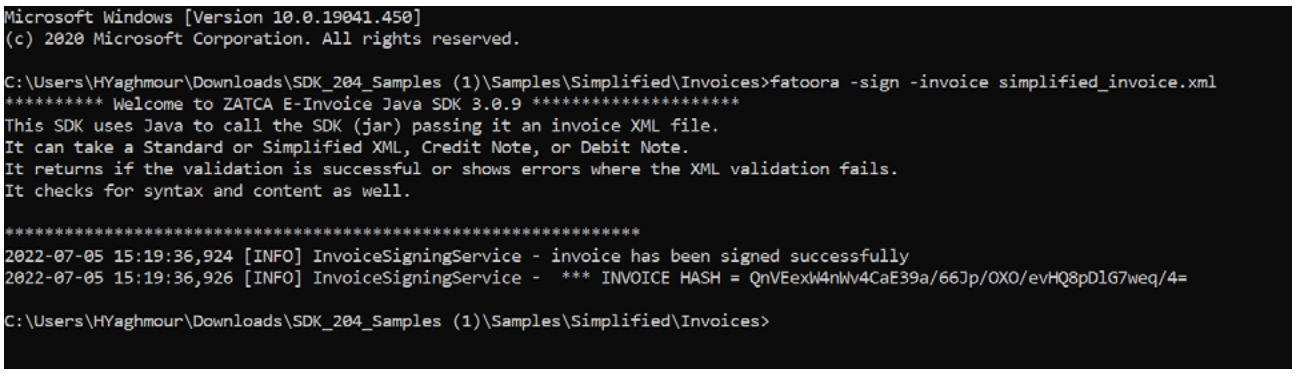
- Result (200)

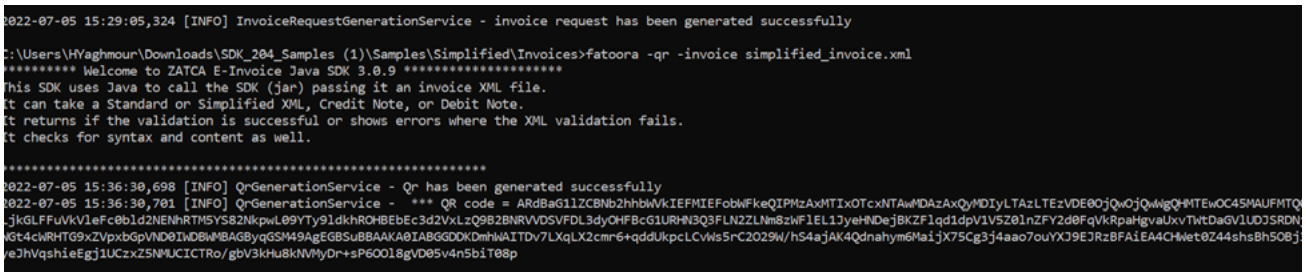


2.3.10.5 REPORTING

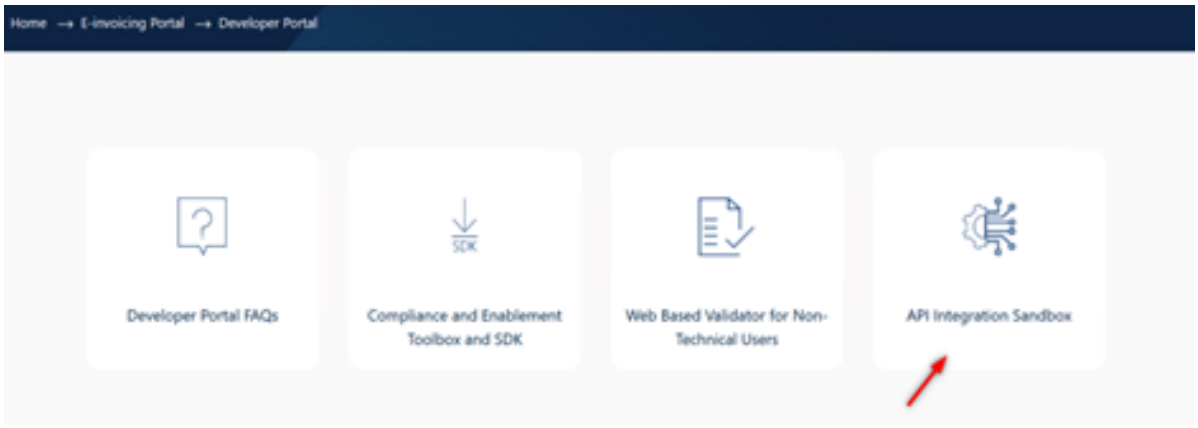
- Step 1: Open CMD and Generate simplified invoice
- Step 2: On SDK, sign the XML invoice and get the hash value using the following:
  - fatoora -sign -qr -invoice invoiceName.xml -signedinvoice signedinvoiceName.xml (hash is returned and signed invoice is generated)
  - fatoora -generateHash -invoice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

    Afterwards, validate the XML invoice.

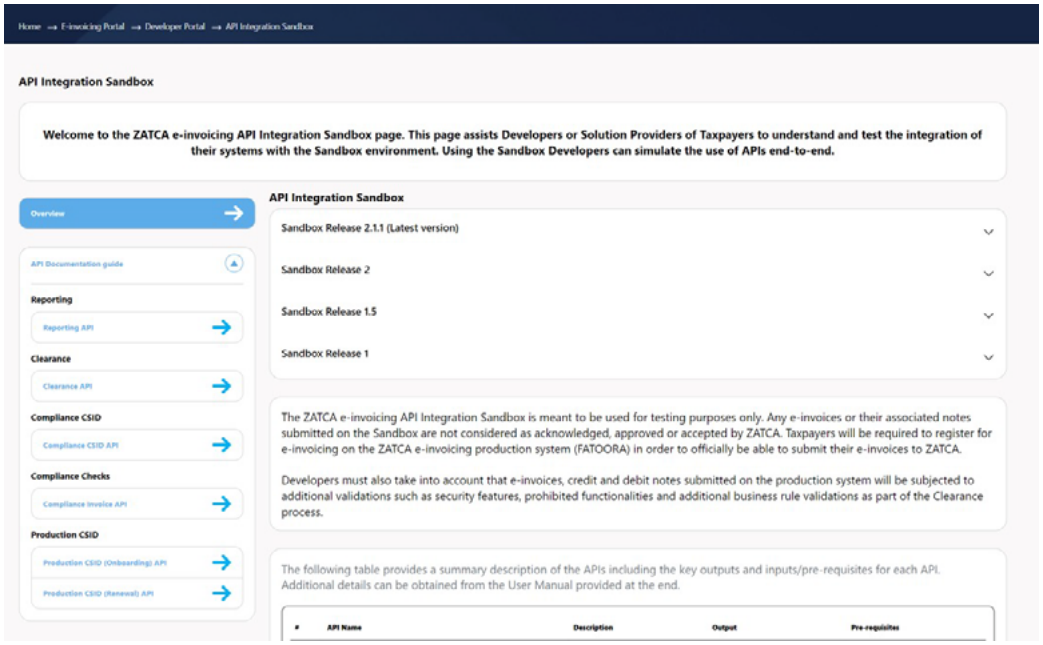




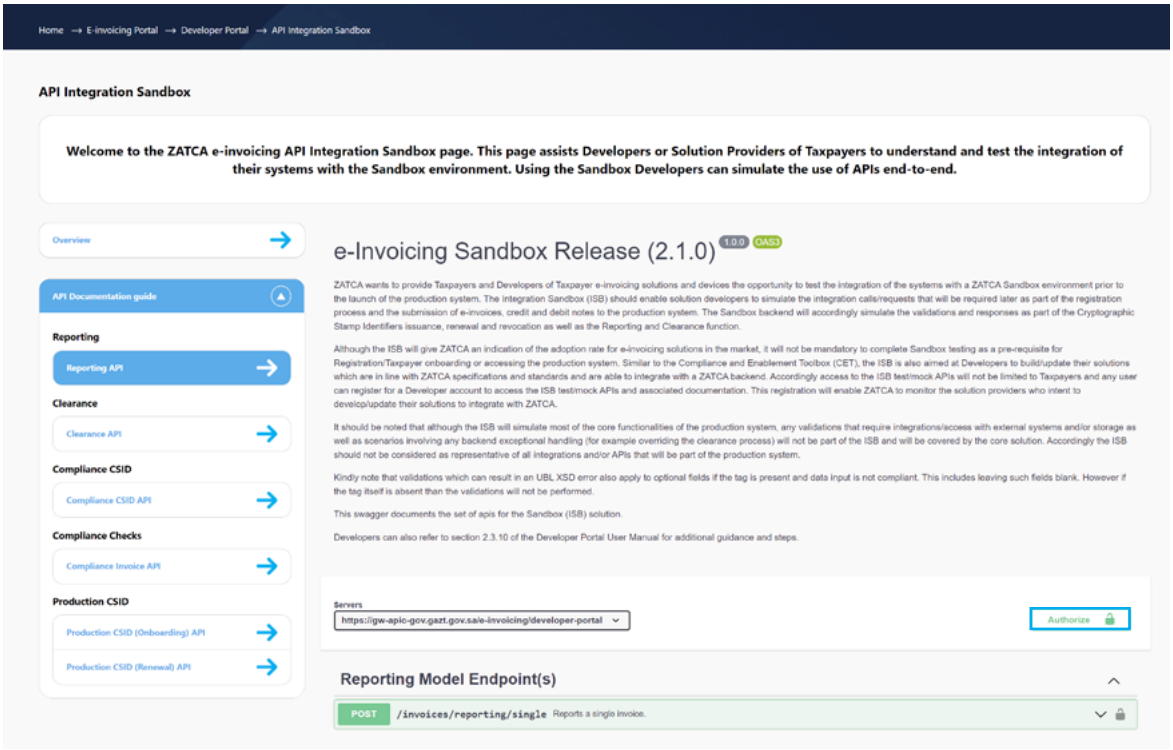
- Step 3: Encode the Signed XML invoice using Base 64
- Step 4: Open Developer Portal and choose integration sandbox



- Step 5: Click on API documentation guides



- Step 6: Click on Authorize Reporting API

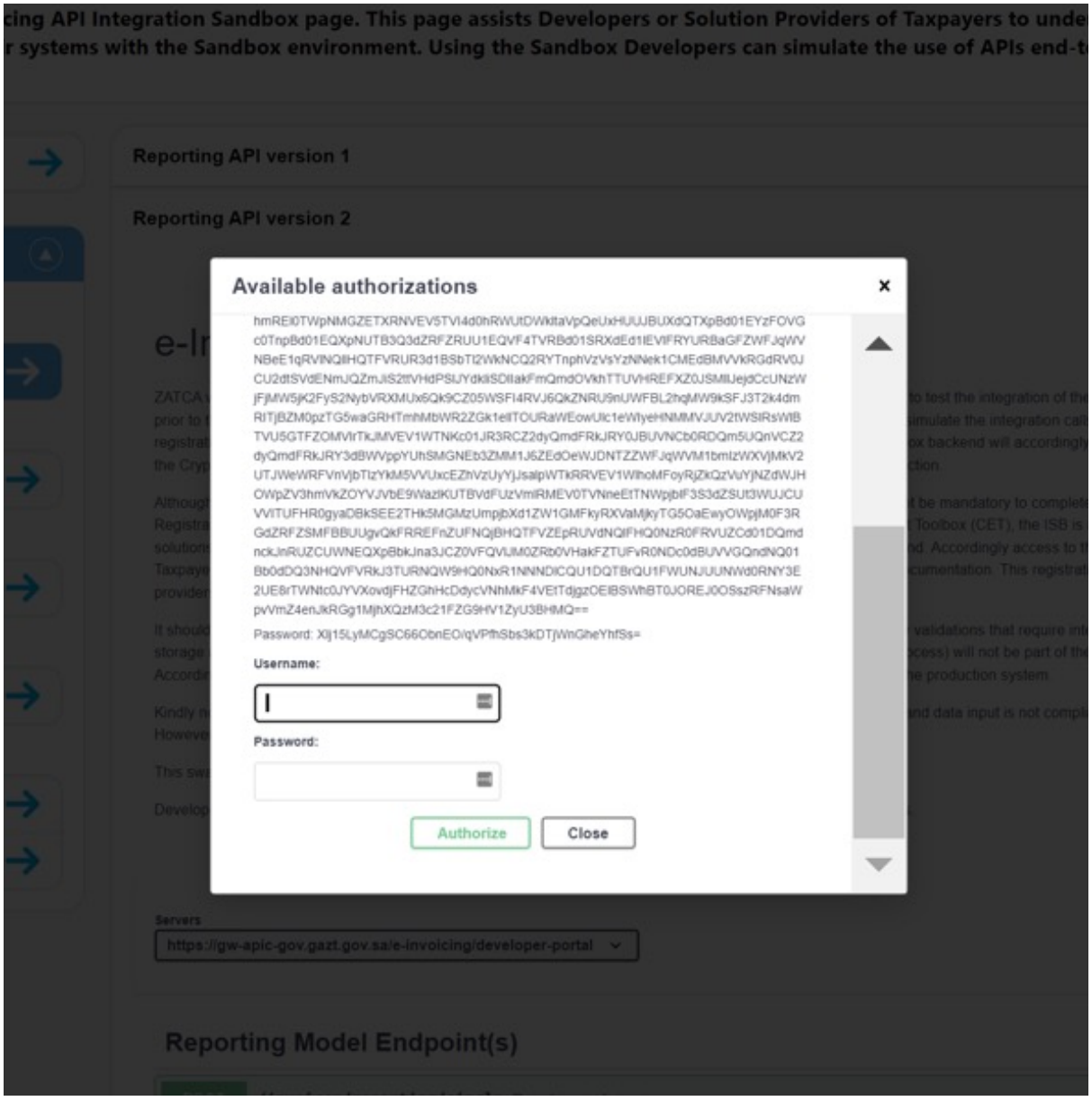


- Step 7:

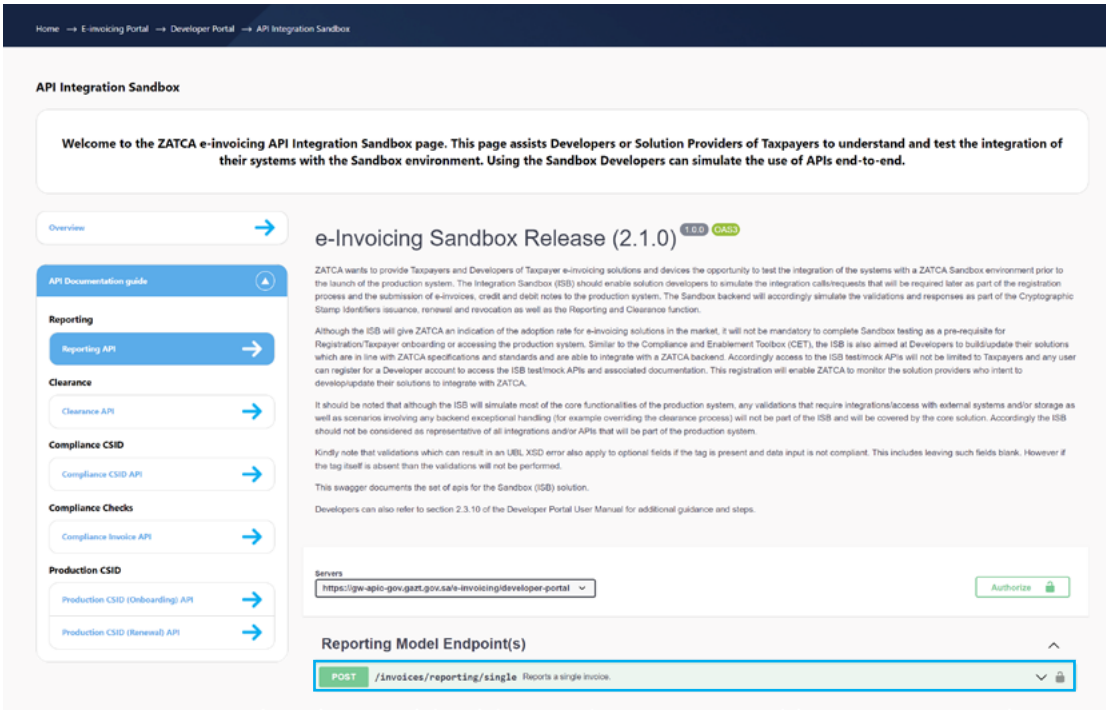
For the Sandbox, use the sample dummy Username and Password provided to you on the Authorization screen.

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

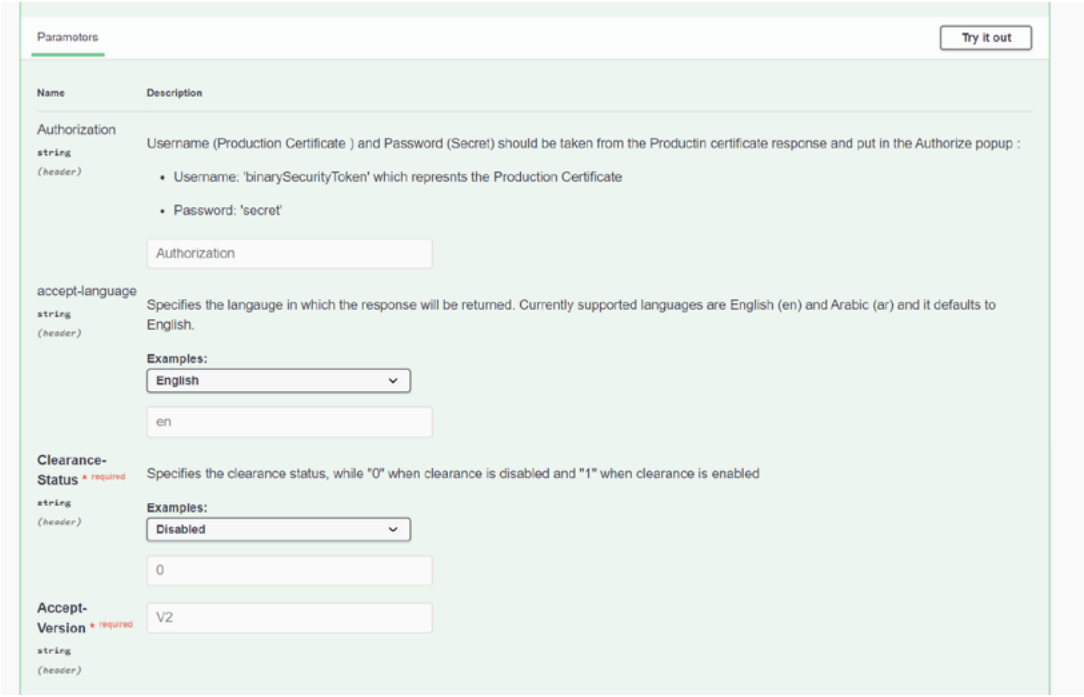
Please refer to Chapter 3 of this document for further details.



- Step 8: Click on Reporting API drop-down

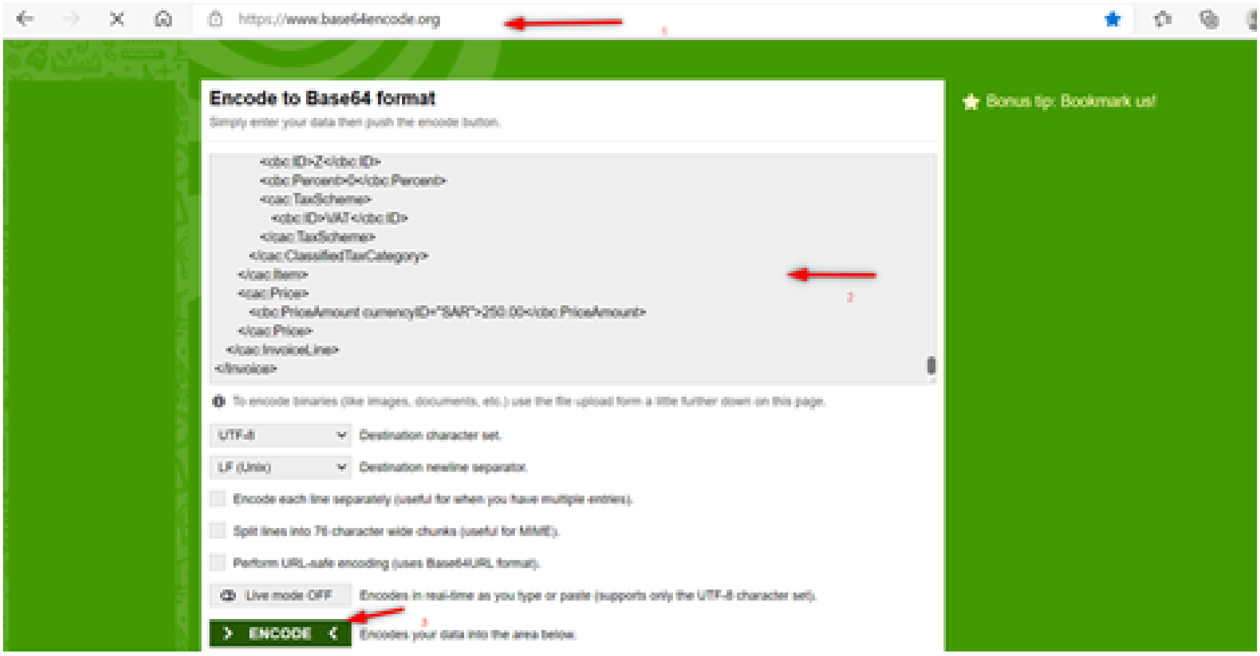


- Step 9: Click on try it out



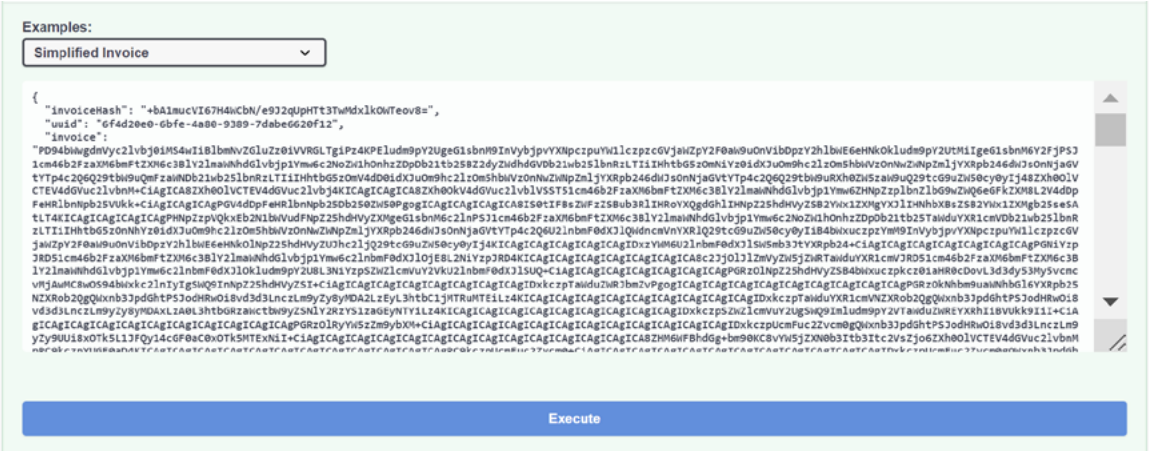
Note: V2 refers to the Version of the APIs used and should be mentioned in the API calls (V2 is currently the only valid version).

- Step 10: Replace the Invoice hash in xml and encode the xml using Base 64

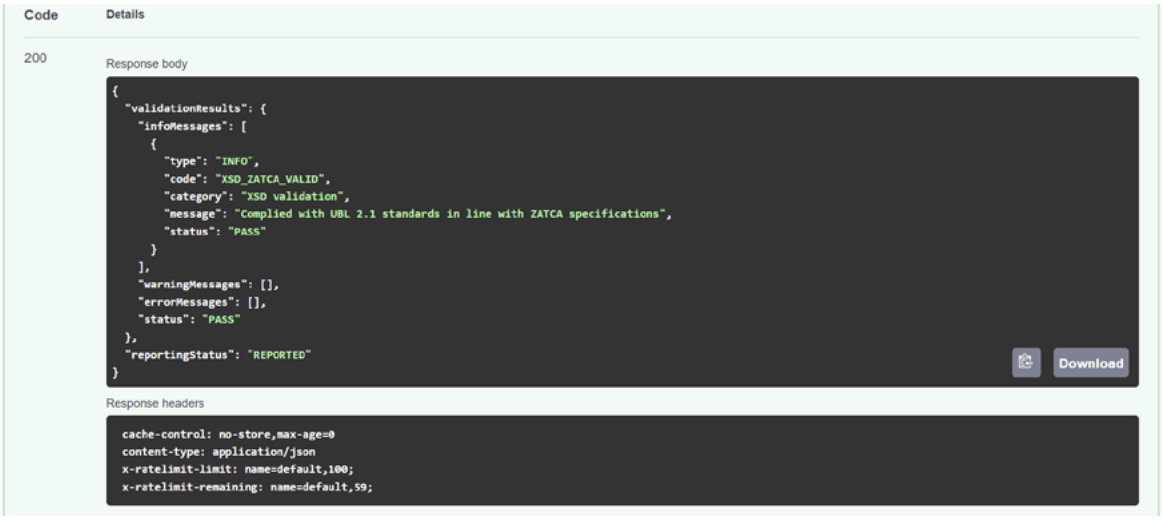


- Step 11:Fill the request body (invoice hash, UUID, Base 64 encoded XML invoice)

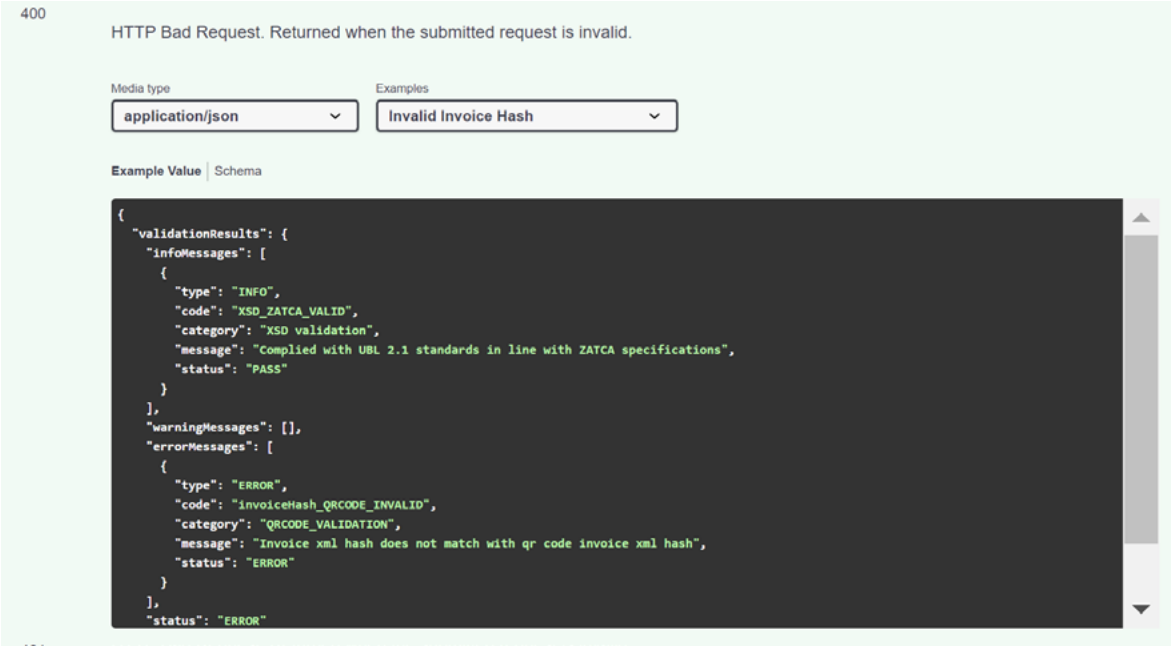
- Step 12: Execute



- Result (200)



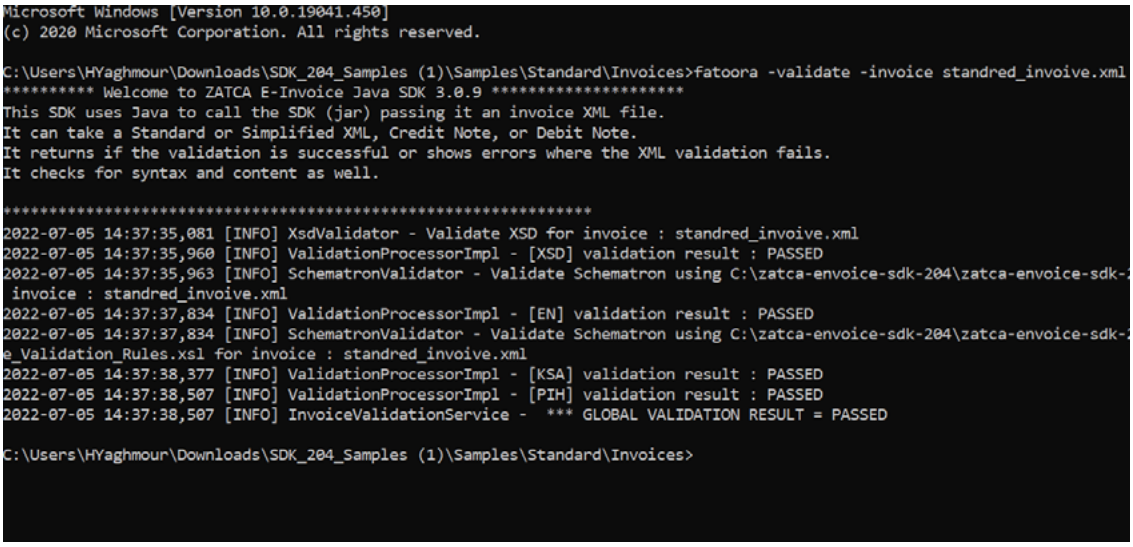
- Result (400)



2.3.10.6 CLEARANCE

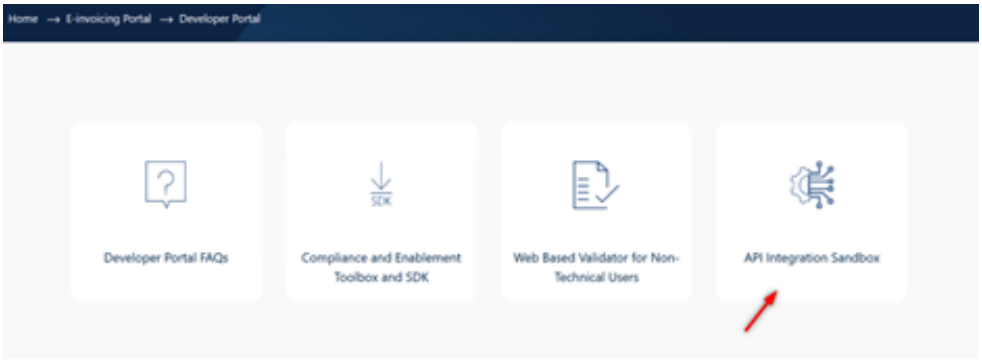
- Step 1: Open CMD and Generate standard invoice

- Step 2: On SDK, get the hash value using the following:
  - fatoora -generateHash -invoice invoiceName.xml
  Afterwards, validate the XML invoice.

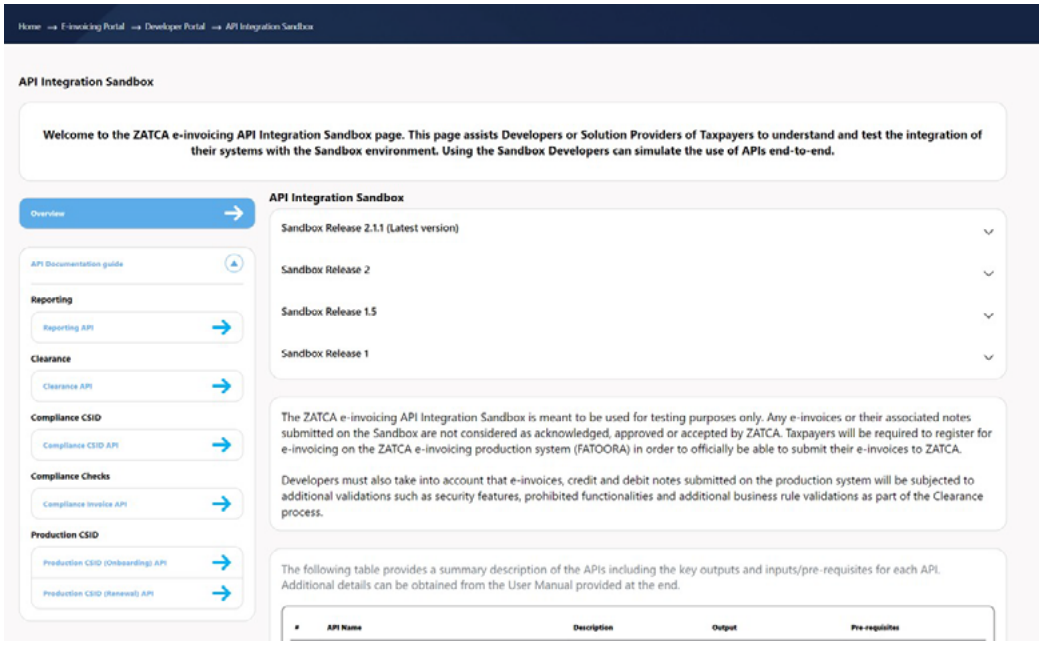


- Step 3: Encode the XML invoice using Base 64

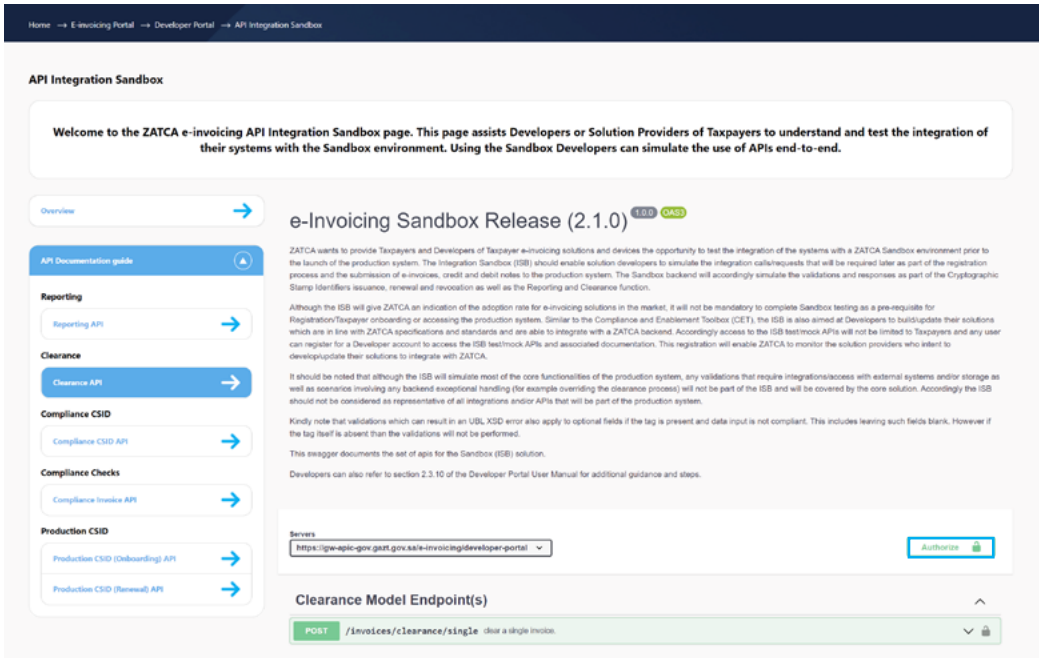
- Step 4: Open Developer Portal and choose integration sandbox



- Step 5: Click on API documentation guides



- Step 6: Click on Authorize Clearance API

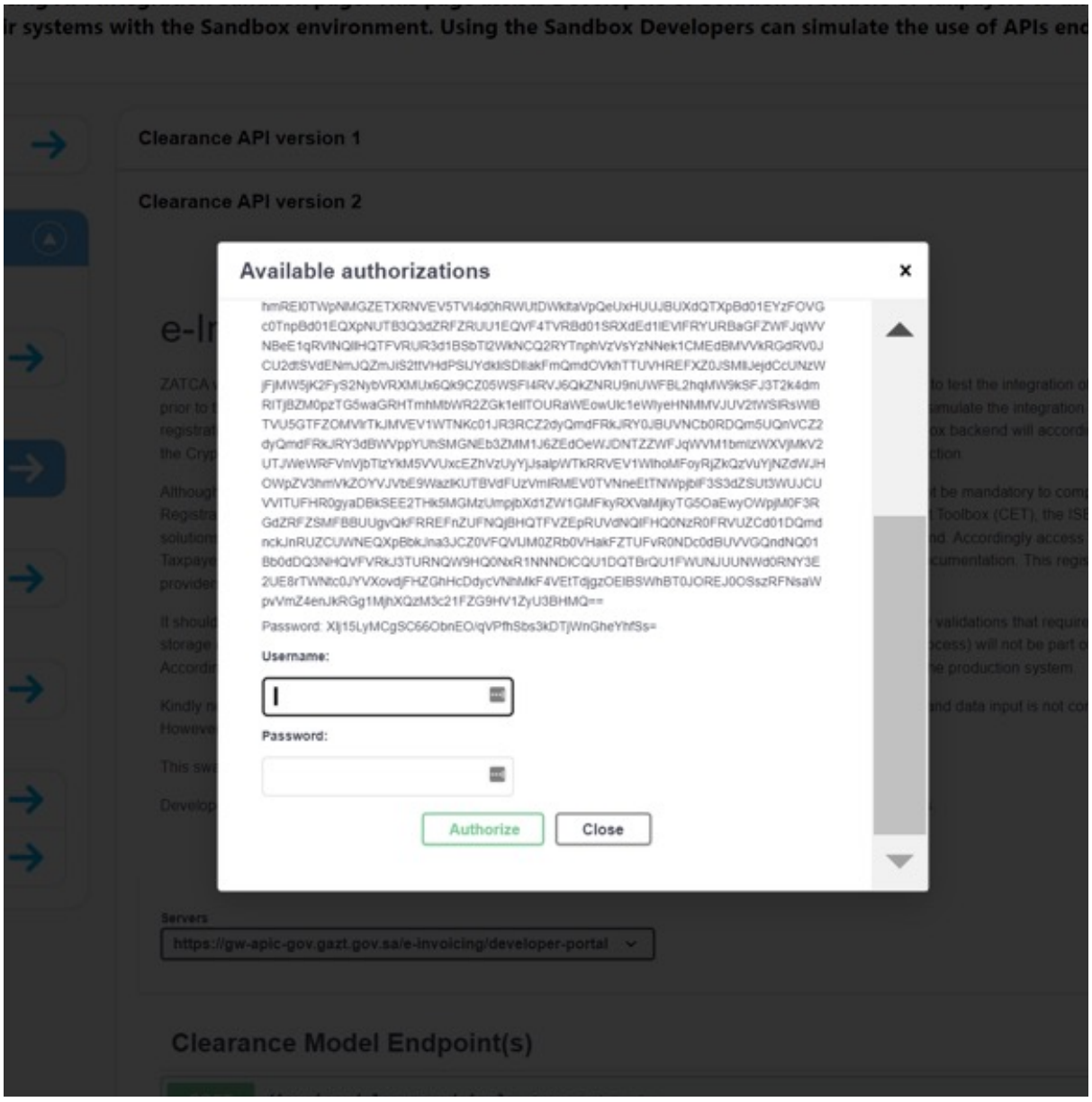


- Step 7:

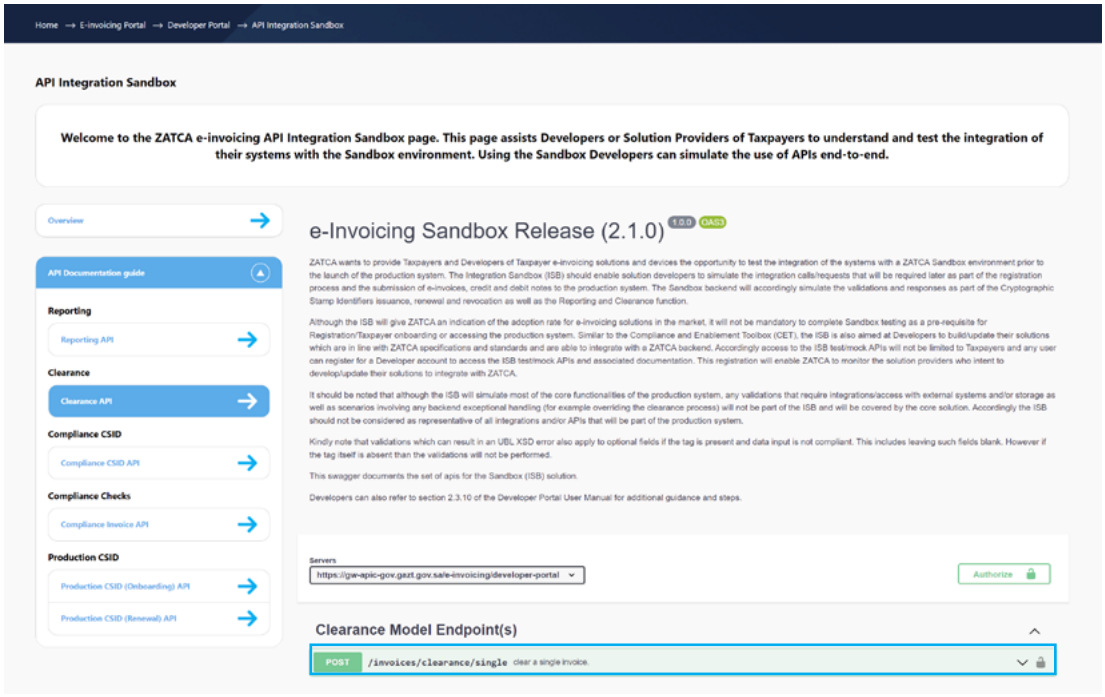
For the Sandbox, use the sample dummy Username and Password provided to you on the Authorization screen.

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

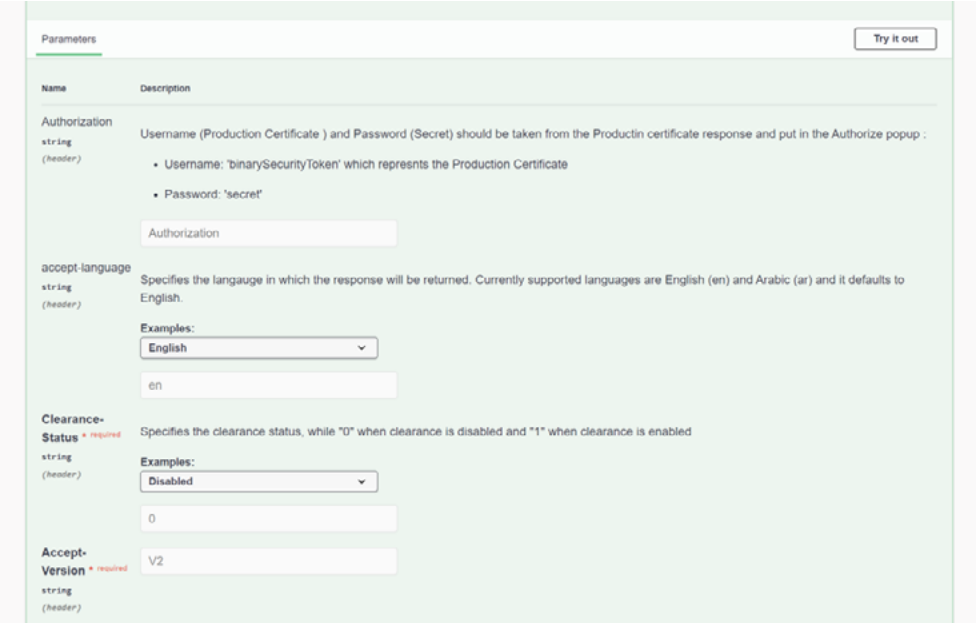
Please refer to Chapter 3 of this document for further details.



- Step 8: Click on Clearance API drop-down

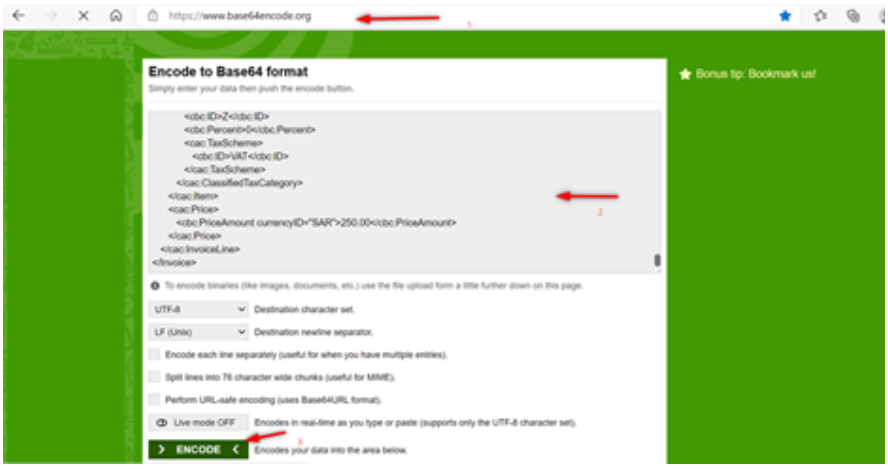


- Step 9: Click on try it out



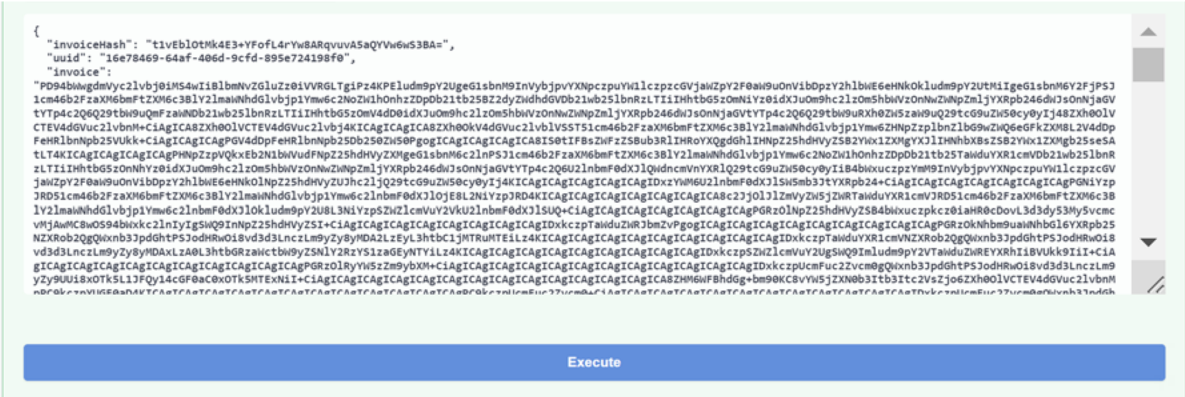
Note: V2 refers to the Version of the APIs used and should be mentioned in the API calls (V2 is currently the only valid version).

- Step 10: Replace the Invoice hash in xml and encode the xml using Base 64

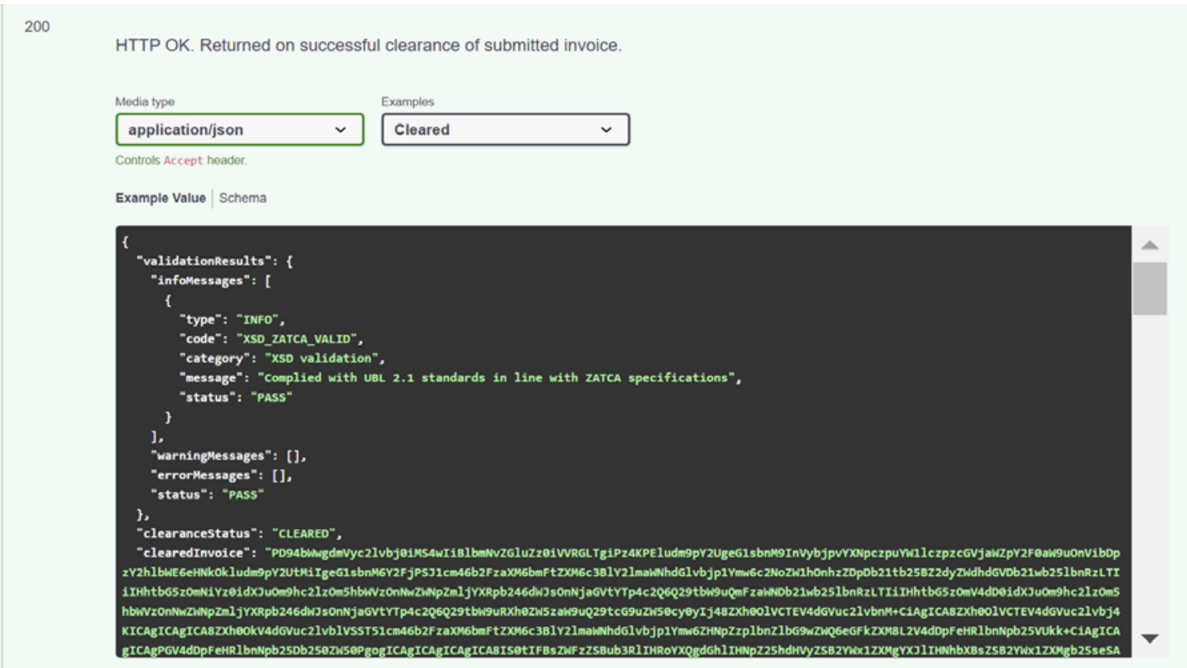


- Step 11: Fill the request body (invoice hash, UUID, Base 64 encoded XML invoice)

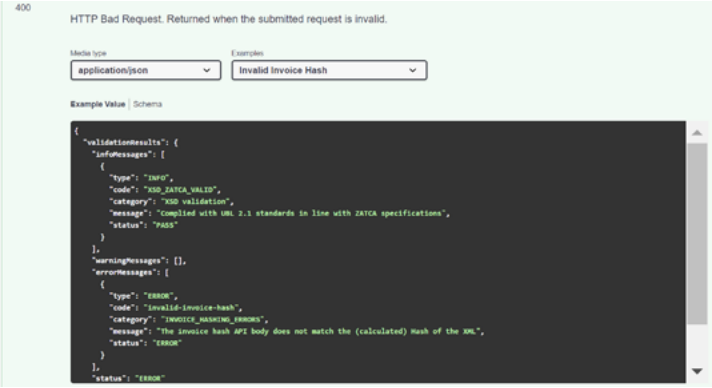
- Step 12: Execute



- Result (200)



- 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.

Function-ality	Description	Release 1 (No-vember 2021)	Release 1.5 (Feb-ruary 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:** • Reporting API • Clearance API **Onboarding APIs:** • CSID API (for • Onboarding) • CSID API (for • Renewal)	**Invoices APIs:** • Reporting API • Clearance API **Onboarding APIs:** • Compliance CSID API • Production CSID API • (for Onboarding) • Production CSID API (for Renewal) • Compliance Checks APIs (for Onboard-ing / Renewal)	**Invoices APIs:** • Reporting API • Clearance API **Onboarding APIs:** • Compliance CSID API • Production CSID API (for On-boarding) • Production CSID API (for Re-newal) • 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.	• 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	• 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	• 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 Einvoicing Solution there will only be one API each for Reporting and Clearance which at any point of time willeither 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.	• Covers the APIs mentioned above • No provisions for Exceptions or turning off Clearance	• 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	• 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	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 When Clearance is disabled, this API can also be used to test submitting Standard e-invoices, credit or debit notes for Reporting 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)	• **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**	• **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	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 When Clearance is disabled, this API will return a 303 Response indicating that the Reporting API be used to submit Standard documents as well 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)	• 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	• 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	• Valid request: Test Compliance CSID and a test Request ID are obtained • Invalid request: Error message(s)	• 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	• Valid request: Test Production CSID is obtained • Invalid request: Error message(s)	• 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	• Valid request: Test Production CSID is obtained • Invalid request: Error message(s)	• A test Compliance CSID obtained from APIs #3 above • A test (dummy) request ID
Compliance Checks APIs (for Onboarding / Renewal)	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 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);	• All Compliance checks passed • One or more compliance checks failed with error messages	• 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.

3. Security Requirements

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

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

Basic Authentication Format:

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

4. Frequently Asked Questions (FAQs)

4.1 Business FAQs

4.1.1 Developer Portal Business FAQs

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

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

4.1.2 SDK Business FAQs

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

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

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

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

4.1.3 Web Based Validator Business FAQs

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

4.1.4 Integration Sandbox Business FAQs

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

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

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

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

4.2 Technical FAQs

4.2.1 Developer Portal Technical FAQs

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

4.2.2 SDK Technical FAQs

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

3	What is a Java and JAR?	Java is a programming language that runs on different operating systems (OS), such as Windows and Linux. A JAR is a package file format that is generally used to aggregate many Java class files and associated metadata and resources (text, images, etc.) into one file for distribution.
4	Can I validate the Arabic language fields in a QR code within the CLI?	No, since the CLI does not support Arabic characters display.
5	What JAVA version should I install before using the SDK?	The prerequisite is using the Java SDK (JAR) versions >=11 and <15.
6	What should the user do when faced with a JAVA error?	When faced with a JAVA error, the user needs to install JAVA (versions >=11 and <15) before running and using the SDK.
7	Where can I find examples of XML files?	Sample XML files are included in SDK. You can download the latest SDK from Developers portal (Sandbox).

4.2.3 Web Based Validator for Non-Technical Users Technical FAQs

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

4.2.4 Integration Sandbox Technical FAQs

#	Question	Answer
1	What is the "Reporting API"?	The "Reporting API" reports a single simplified invoice, credit note, or debit note. Specifically, it accepts a simplified invoice, credit note, or debit note encoded in base64 and validates it to ensure: 1. Compliance to the UBL2 XSD. 2. EN 16931 Rules subset. 3. KSA Specific Rules set. KSA Rules set will override EN 16931 Rules set in case the same rule exists in both sets. 4. QR Code validation 5. Cryptographic Stamp validation
2	How can the user access "Reporting API"?	The user will need to do a POST Method on endpoint / invoices/reporting/single and pass it on "authentication-certificate" and accept-language as a parameter in the header. More information can be found on the Integration Sandbox section of the Developer Portal.

3	What's the Request Body the user should send while calling "Reporting API"?	The body object should Contain 2 Values: the first one is called "invoiceHash" and the second one is called "invoice".
		Example:
		{
		"invoiceHash": "string",
		"invoice": "string"
		}
		More information can be found on the Integration Sandbox section of the Developer Portal.

4	What should the user expect as a response if calling the "Reporting API" was a success?	The response will be 200 HTTP Ok with a Retrieved object containing 4 values : "invoiceHash","Status" ,"Warnings", "errors" .
		Retrieved object Example:
		{
		"invoiceHash": "TODO Add Invoice Hash",
		"status": "Reported",
		"warnings": null,
		"errors": null
		}
		More information can be found on the Integration Sandbox section of the Developer Portal.

5	What is the "Clearance API"?	The "Clearance API" clears a single standard invoice, credit note, or debit note. Specifically, it accepts standard invoice, credit note, or debit note encoded in base64 and validates it to ensure: 1. Compliance to the UBL2 XSD. 2. EN 16931 Rules subset. 3. KSA Specific Rules set. KSA Rules set will override EN 16931 Rules set in case the same rule exists in both sets. On successful validation, the api then applies a cryptographic stamp from ZATCA side and generates a QR Code string. After that the XML is returned back.
6	How can the user access "Clearance API"?	The user will need to do a POST Method on endpoint /invoices/clearance/single and pass it on "authentication-certificate" and accept-language as a parameter in the header. More information can be found on the Integration Sandbox section of the Developer Portal.

7	What's the Request Body the user should send while calling "Clearance API"?	The body object should Contain 2 Values: the first one is called "invoiceHash" and the second one is called "invoice".
		Example:
		{
		"invoiceHash": "string",
		"invoice": "string"
		}
		More information can be found on the Integration Sandbox section of the Developer Portal.

8	What should the user expect as a response if calling "Clearance API" was a Success?	The response will be 200 HTTP Ok with a Retrieved object containing 4 values : "invoiceHash","Status" ,"Warnings", "errors" .
		Retrieved object Example:
		{
		"invoiceHash": "TODO Add Invoice Hash",
		"status": "Reported",
		"warnings": null,
		"errors": null
		}
		More information can be found on the Integration Sandbox section of the Developer Portal.

9	What are the Response causes (Code & Description) that can appear while calling "Reporting single API"?	Code - Description • 200- HTTP OK • 202- Accepted with Errors, simplified invoice accepted with warning errors • 303- HTTP See Other. Returned when the submitted invoice is a Standard Invoice while clearance is activated • 400- HTTP Bad Request. Returned when the submitted request is invalid • 500- HTTP Internal Server Error. Returned when the service faces internal errors

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

5. Appendix

5.1 Glossary

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

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

5.2 Developer Portal Security Information

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

5.3 Generate CSR

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

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

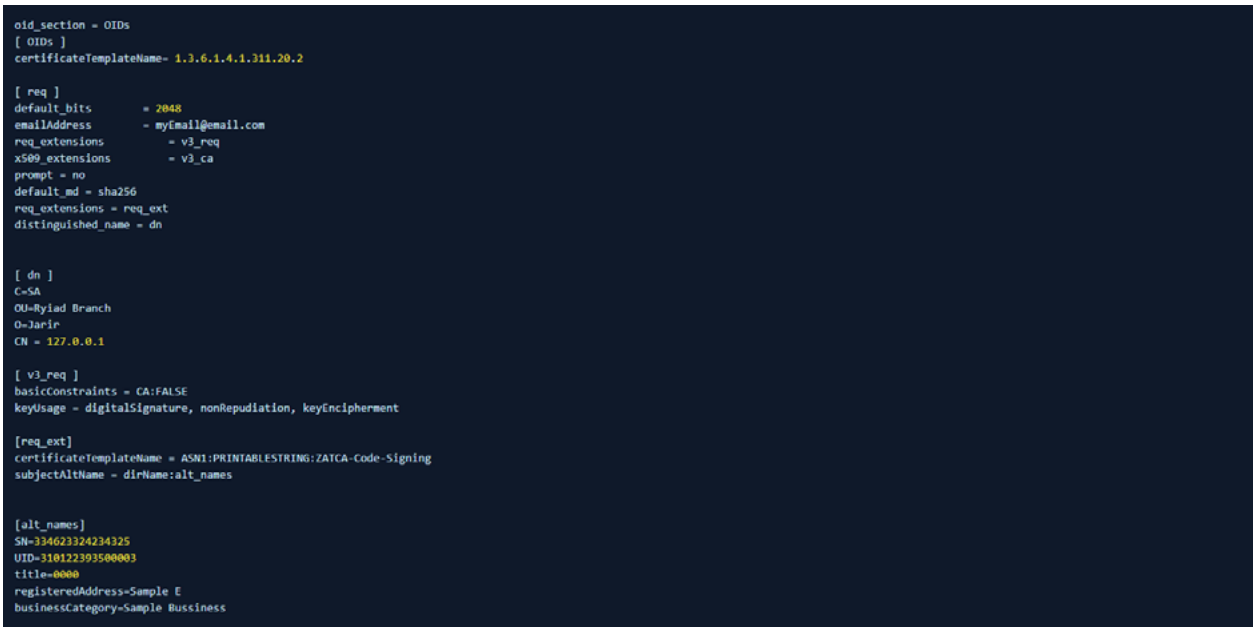
The CSR inputs (Open SSL Config. File) are as follows:

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

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

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

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



5.3.2 Generate public/private key pair

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

5.3.2.1 Generate Private Key

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

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

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

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

5.3.2.2 Generate Public Key

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

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

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

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

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

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

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

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

5.3.3 Generate a Certificate Signing Request

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

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

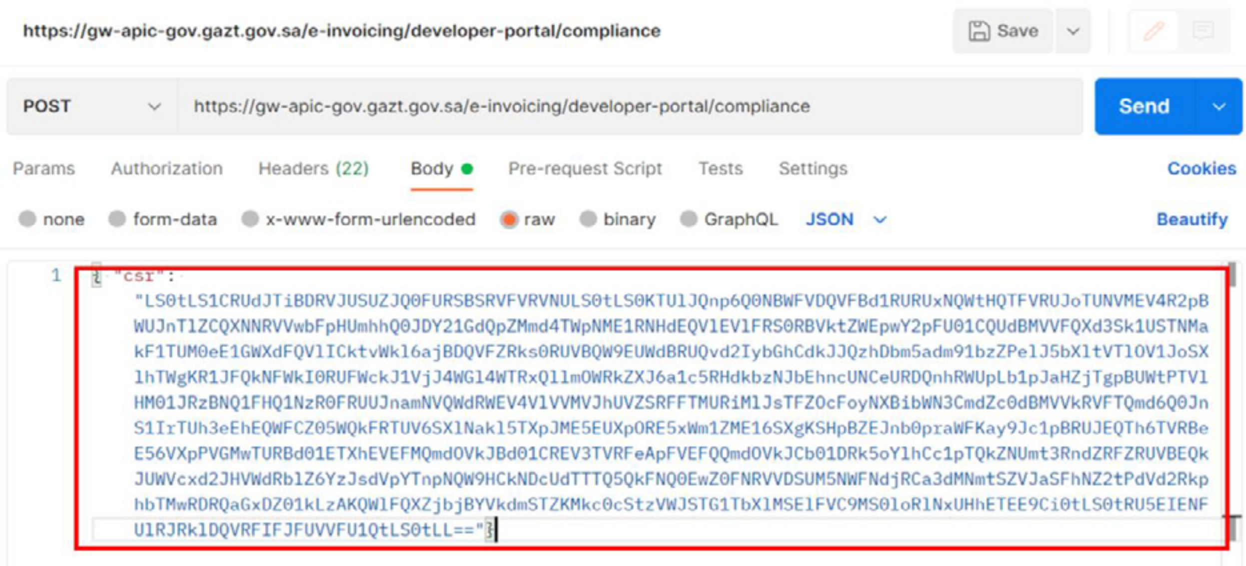
Sample contents of the CSR:

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

5.3.4 Testing the certificate

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

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



External Document

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

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