e-Invoicing Sandbox Release (2.1.0)

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

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

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

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

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

More information: https://helloreverb.com
Contact Info: hello@helloreverb.com
Version: 1.0.0
BasePath:/e-invoicing/developer-portal

Access

1. HTTP Basic Authentication

Methods

[Jump to Models]

Table of Contents

CryptographicStampIdentifierCertificateEndpointS

POST /production/csids

Up

Issues an X509 Production Cryptographic Stamp Identifier (PCSID/Certificate) (CSID) based on submitted CSR. (**productionCsidsPost**)

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

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body object (optional)
Body Parameter —

Request headers

Return type

String

Example data

Content-Type: application/json

```
""
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json
- text/plain;charset=UTF-8

Responses

200

returns a Base64 encoded X509 certificate String

Example data

Content-Type: application/json

```
{"requestID":1642424139872,"dispositionMessage":"ISSUED","binarySecurityToken":"TUlJQjVqQ0NBWTJnQXdJQkFnSUdBWDVvRzRCZ01Bb0dDQ3FHU000OUJBTUNNQlV4RXpBUkJnTlZCQU1NQ21WSmJuWnZhV05wYm1jd0hoY05Nakl3TVRFM01USTFOVE0wV2hjTk1qY3dNVEUzTURBd01EQXdXakJKTVFzd0NRWURWUVFHRXdKVFFURVZNQk1HQTFVRUN3d01ZVzF0WVc0Z1FuSmhibU5vTVE4d0RRWURWUVFLREFab1lYbGhlV0V4RWpBUUJnTlZCQU1NQ1RFeU55NHdMakF1TVRCV01CQUdCeXFHU000OUFnRUdCU3VCQkFBS0EwSUFCQkNRTUlKeE9mUy9uWVIwZ21LZXVCdm9YdU5GckNjZTNualF2Y3VwaXNTVlArSzA3MW9VSS9YejJDa0NhTzhnK3RDeWVLSFFJRUhUMUpnSFkzWmNpZG1qZ1pjd2daUXdEQVlEVlIwVEFRSC9CQUl3QURDQmd3WURWUjBSQkh3d2VxUjRNSFl4R0RBV0JnTlZCQVFNRHpJek5ESXpORE15TURNeU16RTVPVEVmTUIwR0NnbVNKb21UOGl4a0FRRU1Eek13TURBMU5URTRORFF3TURBd016RU5NQXNHQTFVRURBd0VNREF4TURFUU1BNEdBMVVFR2d3SFdtRjBZZMkVnTXpFWU1CWUdBMVVFRHd3UFJtOXZaQ0JDZFhOemFXNWxjM016TUFvR0NDcUdTTTQ5QkFNQ0EwY0FNRVFDSUhRSG54bnF1bDM3WmJ0ZGpFaUEvOUI4MVBNN0VtYzlJaE52SjRyd1JBZHFBaUFKVjk1Nm5DbGE2d0cwR0Y1YStkNy8vT0lTdjVQcEhNUXNNVndwQSs5aW9nPT0=","secret":"yXs8Sgn46CC3lsj2sKW1DQ1eJ4+dQ20bjqZ1hod4Zxw="}
```

400

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

Example data

Content-Type: Invalid ComplianceRequest Id

```
{"errors":[{"code":"Invalid-ComplianceRequestId","message":"The provided compliance_request_id is invalid"}]}
```

Example data

Content-Type: Missing ComplianceSteps

```
{"errors":[{"code":"Missing-ComplianceSteps","message":"Compliance steps for this CSID are not yet complete"}]}
```

Example data

Content-Type: Invalid CurrentCCSID

```
{"errors":[{"code":"Invalid-CurrentCCSID","message":"currentCCSID is invalid"}]}
```

Example data

Content-Type: Missing compliance request id

```
{"errors":[{"code":"Missing-compliance_request_id","message":"compliance_request_id is a required header"}]}
```

Example data

Content-Type: Missing CurrentCCSID

```
{"errors":[{"code":"Missing-CurrentCCSID","message":"currentCCSID is a required header"}]}
```

401

Returned when username and password are not added or added as wrong values.

Example data

Content-Type: Unuthorized

```
{"timestamp":1654514661409,"status":401,"error":"Unauthorized","message":""}
```

406

Example data

Content-Type: Not Acceptable

```
This Version is not supported or not provided in the header.
```

500
HTTP Internal Server Error. Returned when the service faces internal errors. ErrorModel
Example data
Content-Type: InternalServerError

```
{"code":"Invalid-Request","message":"System failed to process your request"}
```

Models

[Jump to Methods]

Table of Contents

CSRRequest - CSRRequest

Up

An object representing the structure of the CSR request that is used to generate a CSID.

csr (optional)
String

CertificatesErrorsResponse - CertificatesErrorsResponse

Up

errors (optional)
array[ErrorModel]

ClearedInvoiceResultModel - ClearedInvoiceResultModel

Up

An object representing the structure of the clearance endpoint response. Specifically, it is an object that contains the hash of the document, status, the cleared document, warnings (if any), and errors (if any).

invoiceHash (optional)
String

clearedInvoice (optional)
String

status (optional)
String
Enum:
Cleared
Not Cleared

warnings (optional)
array[WarningModel]

erros (optional)
array[ErrorModel]

ErrorModel - ErrorModel

Up

An object representing the structure of the error object returned by the API endpoints. Specifically, it includes the Category of the error, its code and message.

category (optional)
String

code (optional)
String

message (optional)
String

InfoModel - InfoModel

Up

An object representing the result of the clearance or reporting API endpoints when the clearance flag is turned on or off. Basically, it shows an informational message instructing the client to see the other api.

message (optional)
String

InvoiceRequest - InvoiceRequest

Up

An object representing the structure of the clearance endpoint request. Specifically, it has the the submitted document hash and the base64 representation of the invoice.

invoiceHash (optional)
String

invoice (optional)
String

InvoiceResultModel - InvoiceResultModel

Up

An Object the represents the response of the API endpoint where it shows the results including status, warnings (if any), and error (if any) in addition to the submitted document hash

invoiceHash (optional)
String

status (optional)
String

Enum:
Reported
Not Reported
Accepted with Warnings

warnings (optional)
array[WarningModel]

erros (optional)
array[ErrorModel]

WarningModel - WarningModel

Up

An object representing the structure of the warning object returned by the API endpoints. Specifically, it includes the Category of the warning, its code and message.

category (optional)
String

code (optional)
String

message (optional)
String