

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://helloverb.com>

Contact Info: hello@helloverb.com

Version: 1.0.0

BasePath:/e-invoicing/developer-portal

All rights reserved

<http://apache.org/licenses/LICENSE-2.0.html>

Access

1. HTTP Basic Authentication

Methods

[[Jump to Models](#)]

Table of Contents

[ComplianceInvoice](#)

- [POST /compliance/invoices](#)

ComplianceInvoice

POST /compliance/invoices

[Up](#)

It performs compliance checks on einvoice documents (**reportSingleInvoice**)

It performs compliance checks on einvoice documents such as:

- Standard invoice.
- Standard debit note.
- Standard credit note.
- Simplified Invoice.
- Simplified credit note.
- Simplified debit note.

Consumes

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

- application/json

Request body

Body Parameter —

$a \in \mathcal{O}_K \setminus \mathcal{O}_K^\times$ and $\mathfrak{p} \in \text{Spec}(\mathcal{O}_K)$ such that $\mathfrak{p} \nmid \text{ord}(a)$. Then $\mathfrak{p} \nmid \text{ord}(a)$ and $\mathfrak{p} \nmid \text{ord}(a)$.

[illegible]

[illegible]


```
{
  "validationResults": {
    "infoMessages": [
      {
        "type": "INFO",
        "code": "XSD_ZATCA_VALID",
        "category": "XSD validation",
        "message": "Complied with UBL 2.1 standards in line with ZATCA"
      }
    ]
  }
}
```

```
specifications","status":"PASS"]],"warningMessages":[],"errorMessages":[{"type":"ERROR","code":"BR-KSA-37","category":"KSA","message":"The seller address building number must contain 4 digits."},"status":"ERROR"},{"type":"ERROR","code":"BR-KSA-09","category":"KSA","message":"Seller address must contain additional number (KSA-23), street name (BT-35), building number (KSA-17), postal code (BT-38), city (BT-37), Neighborhood (KSA-3), country code (BT-40).For more information please access this link: https://www.address.gov.sa/en/address-format/overview","status":"ERROR"}],"status":"ERROR"},"reportingStatus":"NOT_REPORTED","clearanceStatus":null,"qrSellertStatus":null,"qrBuyertStatus":null}
```

Example data

Content-Type: Missing QR Code

```
{"invoiceHash":"TODO add invoice hash","status":"Not Reported","warnings":null,"errors":[{"category":"QR-Code-Errors","code":"Missing-QR-Code","message":"Please include a digital signature in the invoice"}]}
```

Example data

Content-Type: Invalid QR Code

```
{"invoiceHash":"TODO add invoice hash","status":"Not Reported","warnings":null,"errors":[{"category":"QR-Code-Errors","code":"Seller-Name","message":"seller name does not match with qr code seller name"},{"code":"QR-Hashed-XML","message":"Invalid The XML hash. The XML hash of the invoice does not match with QR Code xml hash"}]}
```

Example data

Content-Type: Invalid Authentication Certificate

```
{"invoiceHash":"TODO add invoice hash","status":"Not Reported","warnings":null,"errors":[{"category":"Authentication-Errors","code":"Invalid-Authentication-Certificate","message":"Please include a valid certificate in the header"}]}
```

Example data

Content-Type: Missing Authentication Certificate

```
{"invoiceHash":"TODO add invoice hash","status":"Not Reported","warnings":null,"errors":[{"category":"Authentication-Errors","code":"Missing-Authentication-Certificate","message":"Please include the missing certificate in the header"}]}
```

Example data

Content-Type: Missing Signature

```
{"invoiceHash":"TODO add invoice hash","status":"Not Reported","warnings":null,"errors":[{"category":"Signature-Errors","code":"Missing-Signature","message":"Please include a digital signature in the invoice"}]}
```

Example data

Content-Type: Invalid Signature

```
{"invoiceHash":"TODO add invoice hash","status":"Not Reported","warnings":null,"errors":[{"category":"Signature-Errors","code":"X-509-Issuer-Name","message":"Wrong X509IssuerName"},{"category":"Signature-Errors","code":"Certificate","message":"Wrong Invoice Certificate"},{"category":"Signature-Errors","code":"xades-Signed-Properties-Digest-Value","message":"Wrong xadesSignedPropertiesDigestValue"},{"category":"Signature-Errors","code":"X509-Serial-Number","message":"Wrong X509SerialNumber"},{"category":"Signature-Errors","code":"Signature-Value","message":"Wrong Signature Value"},{"category":"Signature-Errors","code":"Signing-Certificate-Digest-Value","message":"Wrong signingCertificateDigestValue"}]}
```

401

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

Example data

Content-Type: Unauthorized

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

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

1. [CSRRequest - CSRRequest](#)
2. [CertificatesErrorsResponse - CertificatesErrorsResponse](#)
3. [ClearedInvoiceResultModel - ClearedInvoiceResultModel](#)
4. [ErrorModel - ErrorModel](#)
5. [InfoModel - InfoModel](#)
6. [InvoiceRequest - InvoiceRequest](#)
7. [InvoiceResultModel - InvoiceResultModel](#)
8. [WarningModel - WarningModel](#)

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](#)