# API Integration Sandbox

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

## Overview →

## API Documentation guide ▲

### Reporting

Reporting API →

### Clearance

Clearance API →

### Compliance CSID

Compliance CSID API

### Compliance ChecKs

Compliance Invoice API

### Production CSID

Production CSID (Onboarding) API

Production CSID (Renewal) API

## API Integration Sandbox releases

**Sandbox Release 2.1.1 (Latest version)**

Fixes to:

- QR Code generation, validation and clearance
- CSR Validation
- Rounding of Decimal Numbers on invoices
- QR character limit changed from 500 to 700

In addition to:

- Enhances to error messages maKing them more specific
- Security enhancements (please refer to the User Manual for enhancements done on security authentication)

**Sandbox Release 2**

Changes to existing APIs:

- Change to Clearance API to enable turning off clearance, causing Response 303 and Reporting API to accept standard invoices instead. See Swagger files for new flags in Reporting and Clearance API for details.
- Enhanced Exception Handling (Warnings are now displayed as Response 202 — Please refer to the Swagger files for the Reporting and Clearance APIs for more details)
- Updates made to hash validation and hash generation to enforce C14N11 Canonicalization of the XML file prior to hashing, as per the published standards

- The ZATCA email address from which all Developer Portal related notifications are sent was updated to noreply@zatca.gov.sa so as to reduce the possibility of it being sent to recipient's spam folders. Please add this email address to your safe sender's list
- Other minor updates and enhancements made to the Developer Portal screens without any impact to the APIs
- UUID has been introduced in the request as a parameter to help with referencing invalid submissions
- Command line interface has been changed from fatoorah to fatoora

### Sandbox Release 1.5

- Updates to the Onboarding APIs to include the Compliance checKs
- Updates and enhancements made to the Developer Portal screens

### Sandbox Release 1

First release of the Integration Sandbox to test the following APIs:

- Onboarding
- Renewal
- Reporting
- Clearance

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

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

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

| # | API Name | Description | Output | Pre-requisites |
|---|---|---|---|---|
| 1 | Reporting API | This API will be used to submit test Simplified e-invoices, credit or debit note to a test | • If no errors or warnings: Accepted | • A test Production CSID obtained from API #5 or #6 below |

| | | | | |
|---|---|---|---|---|
| | | ZATCA bacKend system as part of the Reporting process<br>When Clearance is disabled, this API can also be used to submit test Standard e-invoices, credit or debit notes for Reporting<br>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 error in Seller Address: Accepted with warning message<br>• If errors other than Seller Address: Rejected with error messages | • Simplified invoice, credit or debit note in XML format<br>• Standard invoice, credit or debit note in XML format when Clearance is disabled |
| 2 | Clearance API | This API will be used to submit test Standard e-invoices, credit or debit note to a test ZATCA bacKend system as part of the Clearance process<br>When Clearance is disabled, this API will return a 303 Response indicating that the Reporting API be used to submit Standard documents as well<br>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 | • If no errors or warnings: Accepted and document is returned with test ZATCA stamp and QR code<br>• If error in Seller Address: Accepted with warning message and document is returned with test ZATCA stamp and QR code<br>• If errors other than Seller Address: Rejected with error messages<br>• 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<br>• Standard invoice, credit or debit note in XML format |

| | | | | |
|---|---|---|---|---|
| | | and the Reporting API must be used to submit Standard documents as well) | | |
| 3 | Compliance CSID API | This API will be used to submit test CSRs (Certificate Signing Requests) to a test ZATCA bacKend system as part of the Onboarding and Renewal process | • Valid request: Test Compliance CSID and a test Request ID are obtained<br>• Invalid request: Error message(s) | • Public Private Key pair<br>• Signed CSR |
| 4 | 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<br>• Invalid request: Error message(s) | • A test Compliance CSID obtained from APIs #3 above<br>• A test (dummy) request ID |
| 5 | 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 Renewal process | • Valid request: Test Production CSID is obtained<br>• Invalid request: Error message(s) | • A test Compliance CSID obtained from APIs #3 above<br>• A test (dummy) request ID |
| 6 | Compliance ChecKs APIs (for Onboarding / Renewal) | These APIs are used to test the compliance of the test device / solution unit (EGS) as part of the Onboarding and/or Renewal processes<br>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); compliance of QR codes as part of 2-way Clearance (Compliance Buyer QRs API) or Seller Acceptance in the case of | • All Compliance checKs passed<br>• One or more compliance checKs failed with error messages | • A test Compliance CSID obtained from APIs #3 above<br>• Standard and/or Simplified invoices, credit or debit notes in XML format |

Self-Billing (compliance Seller

QRs API)

Download User Manual

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

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.

The Clearance API will return a response type 303 when Clearance has been disabled (Standard documents need to be submitted using the Reporting API in this case). Please checK the Swagger file of the Clearance API for more details. Note that the disabling clearance functionality has not been activated in Sandbox Release 1.5.

The "Onboarding API" receives a test CSR (Certificate Signing Request) along with a (dummy) OTP as input, validates them in line with the specifications and standards and returns a test CSID. This test CSID can be used to maKe subsequent integration calls using the Renewal, Reporting or Clearance APIs.

The "Renewal API" receives a test CSR (Certificate Signing Request) along with a (dummy) OTP and an existing test CSID as input, validates them in line with the specifications and standards and returns a new test CSID. The existing test CSID can be any previously obtained test CSID using either the Onboarding or Renewal API.

All the APIs mentioned above can be directly tested from within the Swagger files by clicKing on the "Try It" button on the Swagger itself.

Recruitment

Privacy policy

Information Security

Site Map

Data Initiative

Complaints and suggestions

## Customer Service Center

(Local) 19993

(International) 00966112048998

24/7

# 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.

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

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

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

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

**ClearanceModelEndpointS**

# ClearanceModelEndpointS

## POST /invoices/clearance/single

Up

clear a single invoice. (**clearSingleInvoice**)

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 the below. On successful validation, the API then signs the invoice, applies a QR code and returns back.

- Compliance to the UBL2 XSD.
- EN 16931 Rules set.
- KSA Specific Rules set.

  KSA Rules set will override EN 16931 Rules set in case the same rule exists in both sets.

- QR Code validation (if any)
- Cryptographical Stamp validation (if any)

- Previous Invoice Hash Validation (PIH)

### Consumes

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

- application/json

### Request body

**body InvoiceRequest (required)**

*Body Parameter —*

example: { "summary" : "Standard Invoice", "value" : { "invoiceHash" : "V4U5qlZ3yXQ/Si1AC/R8SLc3F+iN
y27wdVe8IWRqFAQ=", "uuid" : "8d487816-70b8-4ade-a618-9d620b73814a", "invoice" : "PD94bWwgdmV
yc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPEludm9pY2UgeG1sbnM9InVybjpvYXNpczpuYW1lczpz
cGVjaWZpY2F0aW9uOnVibDpzY2hlbWE6eHNkOkludm9pY2UtMiIgeG1sbnM6Y2FjPSJ1cm46b2FzaXM6b
mFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2NoZW1hOnhzZDpDb21tb25BZ2dyZWdhdGVDb21wb25lbnRzL
TIiIHhtbG5zOmNiYz0idXJuOm9hc2lzOm5hbWVzOnNwZWNpZmljYXRpb246dWJsOnNjaGVtYTp4c2Q6Q
29tbW9uQmFzaWNDb21wb25lbnRzLTIiIHhtbG5zOmV4dD0idXJuOm9hc2lzOm5hbWVzOnNwZWNpZmlj
YXRpb246dWJsOnNjaGVtYTp4c2Q6Q29tbW9uRXh0ZW5zaW9uQ29tcG9uZW50cy0yIj48ZXh0OlVCTEV
4dGVuc2lvbnM+CiAgICA8ZXh0OlVCTEV4dGVuc2lvbj4KICAgICAgICA8ZXh0OkV4dGVuc2lvblVSST51cm
46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6ZHNpZzplbnZlbG9wZWQ6eGFkZXM8L2V4dDpFe
HRlbnNpb25VUkk+CiAgICAgICAgPGV4dDpFeHRlbnNpb25Db250ZW50PgogICAgICAgICAgICA8c2lnOlV
CTERvY3VtZW50U2lnbmF0dXJlcyB4bWxuczpzaWc9InVybjpvYXNpczpuYW1lczpzcGVjaWZpY2F0aW9u
OnVibDpzY2hlbWE6eHNkOkNvbW1vblNpZ25hdHVyZUNvbXBvbmVudHMtMiIgeG1sbnM6c2FjPSJ1cm46
b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2NoZW1hOnhzZDpTaWduYXR1cmVBZ2dyZWdhd
GVDb21wb25lbnRzLTIiIHhtbG5zOnNiYz0idXJuOm9hc2lzOm5hbWVzOnNwZWNpZmljYXRpb246dWJsO
nNjaGVtYTp4c2Q6U2lnbmF0dXJlQmFzaWNDb21wb25lbnRzLTIiPgogICAgICAgICAgICAgICAgPHNhYzp
TaWduYXR1cmVJbmZvcm1hdGlvbj4gCiAgICAgICAgICAgICAgICAgICAgPGNiYzpJRD51cm46b2FzaXM6
bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2lnbmF0dXJlOjE8L2NiYzpJRD4KICAgICAgICAgICAgICAgICA
gICA8c2JjOlJlZmVyZW5jZWRTaWduYXR1cmVJRD51cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp
1Ymw6c2lnbmF0dXJlOkludm9pY2U8L3NiYzpSZWZlcmVuY2VkU2lnbmF0dXJlSUQ+CiAgICAgICAgICAgI
CAgICAgICAgPGRzOlNpZ25hdHVyZSB4bWxuczpkcz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94
bWxkc2lnIyIgSWQ9InNpZ25hdHVyZSI+CiAgICAgICAgICAgICAgICAgICAgICAgIDxkczpTaWduZWRJbm
ZvPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOkNhbm9uaWNhbGl6YXRpb25NZXRob2QgQ
Wxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDA2LzEyL3htbC1jMTRuMTEiLz4KICAgICAgICAgICA
gICAgICAgICAgICAgIDxkczpTaWduYXR1cmVNZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3Lnc
zLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNlY2RzYS1zaGEyNTYiLz4KICAgICAgICAgICAgICAgICAg
ICAgICAgICAgIDxkczpSZWZlcmVuY2UgSWQ9Imludm9pY2VTaWduZWREYXRhIiBVUkk9IiI+CiAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgPGRzOlRyYW5zZm9ybXM+CiAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgIDxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy9
UUi8xOTk5L1JFQy14cGF0aC0xOTk5MTExNiI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICA8ZHM6WFBhdGg+bm90KC8vYW5jZXN0b3Itb3Itc2VsZjo6ZXh0OlVCTEV4dGVuc2lvbnMpPC9
kczpYUGF0aD4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPC9kczpUcmFuc2Zvcm0+
CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPS
JodHRwOi8vd3d3LnczLm9yZy9UUi8xOTk5L1JFQy14cGF0aC0xOTk5MTExNiI+CiAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICA8ZHM6WFBhdGg+bm90KC8vYW5jZXN0b3Itb3Itc2VsZjo6Y2F
jOlNpZ25hdHVyZSk8L2RzOlhQYXRoPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8L2
RzOlRyYW5zZm9ybT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOlRyYW5zZm
9ybSBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnL1RSLzE5OTkvUkVDLXhwYXRoLTE5OTkxMTE2Ij
4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpYUGF0aD5ub3QoLy9hbmNlc
3Rvci1vci1zZWxmOjpjYWM6QWRkaXRpb25hbERvY3VtZW50UmVmZXJlbmNlW2NiYzpJRD0nUVInXSk
8L2RzOlhQYXRoPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8L2RzOlRyYW5zZm9y
bT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOlRyYW5zZm9ybSBBbGdvcml0a
G09Imh0dHA6Ly93d3cudzMub3JnLzIwMDYvMTIveG1sLWMxNG4xMSIvPgogICAgICAgICAgICAgIC
AgICAgICAgICAgIDwvZHM6VHJhbnNmb3Jtcz4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CA8ZHM6RGlnZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bW
xlbmMjc2hhMjU2Ii8+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOkRpZ2VzdFZhbHVlPl
Y0VTVxbFozeVhRL1NpMUFDL1I4U0xjM0YraU55Mjd3ZFZlOElXUnFGQVE9PC9kczpEaWdlc3RWYWx1
ZT4KICAgICAgICAgICAgICAgICAgICAgICAgIDwvZHM6UmVmZXJlbmNlPgogICAgICAgICAgICAgIC
AgICAgICAgICAgPGRzOlJlZmVyZW5jZSBUeXBlPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3
htbGRzaWcjU2lnbmF0dXJlUHJvcGVydGllcyIgVVJJPSIjeGFkZXNTaWduZWRQcm9wZXJ0aWVzIj4KICA
gICAgICAgICAgICAgICAgICAgICAgICA8ZHM6RGlnZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0
cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+CiAgICAgICAgICAgICAgICAgICA
gICAgICAgPGRzOkRpZ2VzdFZhbHVlPk9EUXdOVGcxTlRCaE1qTXpNMll4WTJaa1pqVmtZemRsTl
RaaVpqWTBPREpqTWpOa1lXSTRNVFV6TmpkbU5EVmpNakF3WlRCak9EYzJZVE5oTVdRMU5nPT08L
2RzOkRpZ2VzdFZhbHVlPgogICAgICAgICAgICAgICAgICAgICAgICAgPC9kczpSZWZlcmVuY2U+Ci
AgICAgICAgICAgICAgICAgIDwvZHM6U2lnbmVkSW5mbz4KICAgICAgICAgICAgICAgICAgICAgICAg
ICAgPGRzOlNpZ25hdHVyZVZhbHVlPk1FVUNJQnh5UjhyYzRLODcyOHdkU0Y0WFNEcVBzK3JJTCszV
EZoOW0rYU54UVB0U0FpRUE2Y0hhcEl0dnAxM3lNU3U2Nk5iT2cyQ3BvbUh3VVNuWUo5aDZ1R1E2N
WFZPTwvZHM6U2lnbmF0dXJlVmFsdWU+CiAgICAgICAgICAgICAgICAgICAgIDxkczpLZXlJbmZvPg
ogICAgICAgICAgICAgICAgICAgICAgICAgPGRzOlg1MDlEYXRhPgogICAgICAgICAgICAgICAgICAgI
CAgICAgICAgIDxkczpYNTA5Q2VydGlmaWNhdGU+TUlJRDNqQ0NBNFNnQXdJQkFnSVRFUUFB
T0FQRjkwQWpzL3hjWHdBQkFBQTRBekFLQmdncWhrak9QUVFEQWpCaU1SVXdFd1lLQ1pJbWlaUHlM

R1FCR1JZRmJHOWpZV3d4RXpBUkJnb0praWFKay9Jc1pBRVpGZ05uYjNZeEZ6QVZCZ29Ka2lhSmsvS
XNaQUVaRmdkbGVIUm5ZWHAwTVJzd0dRWURWUVFERXhKUVVscEZTVTVXVDBsRFJWTkRRVFF0
UTBFd0hoY05NalF3TVRFeE1Ea3hPVE13V2hjTk1qa3dNVEE1TURreE9UTXdXakIxTVFzd0NRWURWUV
FHRXdKVFFURW1NQ1FHQTFVRUNoTWRUV0Y0YVcxMWJTQlRjR1ZsWkNCVVpXTm9JRk4xY0hCc2
VTQk1WRVF4RmpBVUJnTlZCQXNURFZKcGVXRmthQ0JDY21GdVkyZ3hKakFrQmdOVkJBTVRIVlJUV
kMwNE9EWTBNekV4TkRVdE16azVPVGs1T1RrNU9UQXdNREF6TUZZd0VBWUhLb1pJemowQ0FRWU
ZLNEVFQUFvRFFnQUVvV0NLYTBTYTlGSUVyVE92MHVBa0MxVklLWHhVOW5QcHgydmxmNHloTWV
qeThjMDJYSmJsRHE3dFB5ZG84bXEwYWhPTW1Obzhnd25pN1h0MUtUOVVlS09DQWdjd2dnSURNSU
d0QmdOVkhSRUVnYVV3Z2FLa2daOHdnWnd4T3pBNUJnTlZCQVFNTWpFdFZGTlVmREl0VkZOVWZET
XRaV1F5TW1ZeFpEZ3RaVFpoTWkweE1URTRMVGxpTlRndFpEbGhPR1l4TVdVME5EVm1NUjh3SFFZS
0NaSW1pWlB5TEdRQkFRd1BNems1T1RrNU9UazVPVEF3TURBek1RMHdDd1lEVlFRTURBUXhNVEF3
TVJFd0R3WURWUVFhREFoU1VsSkVNamt5T1RFYU1CZ0dBMVVFRHd3UlUzVndjR3g1SUdGamRHbD
JhWFJwWlhNd0hRWURWUjBPQkJZRUZFWCtZdm1tdG5Zb0RmOUJHYktvN29jVEtZSzFNQjhHQTFVZ
El3UVlNQmFBRkp2S3FxTHRtcXdza0lGelZ2cFAyUHhUKzlObk1Ic0dDQ3NHQVFVRkJ3RUJCRzh3YlRC
ckJnZ3JCZ0VGQlFjd0FvWmZhSFIwY0RvdkwyRnBZVFF1ZW1GMFkyRXVaMjkyTG5OaEwwTmxjblJGY
m5KdmJHd3ZVRkphUlVsdWRtOXBZMlZUUTBFMExtVjRkR2RoZW5RdVoyOTJMbXh2WTJGc1gxQlNXa
1ZKVGxaUFNVTkZVME5CTkMxRFFFTZ3hLUzVqY25Rd0RnWURWUjBQQVFIL0JBUURBZ2VBTUR3R0
NTc0dBUVFCZ2pVKJ3UXZNQzBHSlNzR0FRUUJnamNWQ0lHR3FCMkUwUHNTaHUyZEpJZk8reG5Ud
0ZWbWgvcWxaWVhaaEQ0Q0FXUUNBUkl3SFFZRFZSMGxCQll3RkFZSUt3WUJCUVVIQXdNR0NDc0d
BUVVGQndNQ01DY0dDU3NHQVFRQmdqY1ZDZ1FhTUJnd0NnWUlLd1lCQlFVSEF3TXdDZ1lJS3dZQkJ
RVUhBd0l3Q2dZSUtvWkl6ajBFQXdJRFNBQXdSUUloQUxFL2ljaG1uV1hDVUtVYmNhM3ljaThvcXdhTHZ
GZEhWalFydmVJOXVxQWJBaUE5aEM0TThqZ01CQURQU3ptZDJ1aVBKQTZnS1IzTEUwM1U3NWVxY
kMvclhBPT08L2RzOlg1MDlDZXJ0aWZpY2F0ZT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgIDwvZH
M6WDUwOURhdGE+CiAgICAgICAgICAgICAgICAgICAgIDwvZHM6S2V5SW5mbz4KICAgICAgICAg
ICAgICAgICAgICAgPGRzOk9iamVjdD4KICAgICAgICAgICAgICAgICAgICAgICAgICAgIDx4YWRlczp
RdWFsaWZ5aW5nUHJvcGVydGllcyB4bWxuczp4YWRlcz0iaHR0cDovL3VyaS5ldHNpLm9yZy8wMTkwM
y92MS4zLjIjIiBUYXJnZXQ9InNpZ25hdHVyZSI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPH
hhZGVzOlNpZ25lZFByb3BlcnRpZXMgSWQ9InhhZGVzU2lnbmVkUHJvcGVydGllcyI+CiAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgIDx4YWRlczpTaWduZWRTaWduYXR1cmVQcm9wZXJ0aWVzPg
ogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOlNpZ25pbmdUaW1lPjIwM
jQtMDEtMTRUMTA6MjE6NDA8L3hhZGVzOlNpZ25pbmdUaW1lPgogICAgICAgICAgICAgICAgI
CAgICAgICAgICAgPHhhZGVzOlNpZ25pbmdDZXJ0aWZpY2F0ZT4KICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICA8eGFkZXM6Q2VydD4KICAgICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgPHhhZGVzOkNlcnREaWdlc3Q+CiAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6RGlnZXN0TWV0aG9kIEFsZ29yaX
RobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+CiAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6RGlnZXN0VmFsdWU+WkRNd01tS
TBNVEUxTnpWak9UVTJOVGs0WXpWbE9EaGhZbUkwT0RVMk5EVXlOVFUyWVRWaFlqaGhNREZtTjJ
GallqazFZVEEyT1dRME5qWTJNalE0TlE9PTwvZHM6RGlnZXN0VmFsdWU+CiAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgIDwveGFkZXM6Q2VydERpZ2VzdD4KICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOklzc3VlclNlcmlhbD4KICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpYNTA5SXNzdW
VyTmFtZT5DTj1QUlpFSU5WT0lDRVNDQTQtQ0EsIERDPWV4dGdhenQsIERDPWdvdiwgREM9bG9jYW
w8L2RzOlg1MDlJc3N1ZXJOYW1lPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgPGRzOlg1MDlTZXJpYWxOdW1iZXI+Mzc5MTEyNzQyODMxMzgwNDcxODM1MjYzO
TY5NTg3Mjg3NjYzNTIwNTI4Mzg3PC9kczpYNTA5U2VyaWFsTnVtYmVyPgogICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICA8L3hhZGVzOklzc3VlclNlcmlhbD4KICAgICAgICAgICA
gICAgICAgICAgICAgICA8L3hhZGVzOkNlcnQ+CiAgICAgICAgICAgICAgIC
AgICAgICAgICAgICA8L3hhZGVzOlNpZ25pbmdDZXJ0aWZpY2F0ZT4KICAgICA
gICAgICAgICAgICAgICAgPC94YWRlczpTaWduZWRTaWduYXR1cmVQcm9wZXJ0a
WVzPgogICAgICAgICAgICAgICAgICAgIDwveGFkZXM6U2lnbmVkUHJvcGVydGllcz4
KICAgICAgICAgICAgICAgICAgIDwveGFkZXM6UXVhbGlmeWluZ1Byb3BlcnRpZXM+CiAgI
CAgICAgICAgICAgIDwvZHM6T2JqZWN0PgogICAgICAgICAgICAgIDwvZHM6U
2lnbmF0dXJlPgogICAgICAgICAgPC9zYWM6U2lnbmF0dXJlSW5mb3JtYXRpb24+CiAgICA
gICAgIDwvc2lnOlVCTERvY3VtZW50U2lnbmF0dXJlcz4KICAgICAgICA8L2V4dDpFeHRlbnNpb25Db250Z
W50PgogICAgPC9leHQ6VUJMRXh0ZW5zaW9uPgo8L2V4dDpVQkxFeHRlbnNpb25zPgogICAgCiAgICA8
Y2JjOlByb2ZpbGVJRD5yZXBvcnRpbmc6MS4wPC9jYmM6UHJvZmlsZUlEPgogICAgPGNiYzpJRD5TTU
UwMDAyMzwvY2JjOklEPgogICAgPGNiYzpVVUlEPjhkNDg3ODE2LTcwYjgtNGFkZS1hNjE4LTlkNjIwYjcz
ODE0YTwvY2JjOlVVSUQ+CiAgICA8Y2JjOklzc3VlRGF0ZT4yMDIyLTA5LTA3PC9jYmM6SXNzdWVEYX
RlPgogICAgPGNiYzpJc3N1ZVRpbWU+MTI6MjE6Mjg8L2NiYzpJc3N1ZVRpbWU+CiAgICA8Y2JjOkludm9
pY2VUeXBlQ29kZSBuYW1lPSIwMTAwMDAwIj4zODg8L2NiYzpJbnZvaWNlVHlwZUNvZGU+CiAgICA8Y2
JjOkRvY3VtZW50Q3VycmVuY3lDb2RlPlNBUjwvY2JjOkRvY3VtZW50Q3VycmVuY3lDb2RlPgogICAgPG
NiYzpUYXhDdXJyZW5jeUNvZGU+U0FSPC9jYmM6VGF4Q3VycmVuY3lDb2RlPgogICAgPGNhYzpBZG
RpdGlvbmFsRG9jdW1lbnRSZWZlcmVuY2U+CiAgICAgICAgPGNiYzpJRD5JQ1Y8L2NiYzpJRD4KICAgIC
AgICA8Y2JjOlVVSUQ+MjM8L2NiYzpVVUlEPgogICAgPC9jYWM6QWRkaXRpb25hbERvY3VtZW50UmV
mZXJlbmNlPgogICAgPGNhYzpBZGRpdGlvbmFsRG9jdW1lbnRSZWZlcmVuY2U+CiAgICAgICAgPGNiYz
pJRD5QSUg8L2NiYzpJRD4KICAgICAgICA8Y2FjOkF0dGFjaG1lbnQ+CiAgICAgICAgICAgIDxjYmM6RW1i
ZWRkZWREb2N1bWVudEJpbmFyeU9iamVjdCBtaW1lQ29kZT0idGV4dC9wbGFpbiI+TldabFFkyVmlOalptW
m1NNE5tWXpPR1E1TlRJM09EWmpObVEyT1Raak56bGpNbVJpWXpJek9XUmtOR1U1TVdJME5qY3lPV
1EzTTJFeU4yWmlOVGRsT1E9PTwvY2JjOkVtYmVkZGVkRG9jdW1lbnRCaW5hcnlPYmplY3Q+CiAgICA
gICAgPC9jYWM6QXR0YWNobWVudD4KICAgIDwvY2FjOkFkZGl0aW9uYWxEb2N1bWVudFJlZmVyZW5j
ZT4KICAgIAogICAgCiAgICA8Y2FjOkFkZGl0aW9uYWxEb2N1bWVudFJlZmVyZW5jZT4KICAgICAgICA8Y

2JjOklEPlFSPC9jYmM6SUQ+CiAgICAgICAgPGNhYzpBdHRhY2htZW50PgogICAgICAgICAgICA8Y2JjOk
VtYmVkZGVkRG9jdW1lbnRCaW5hcnlPYmplY3QgbWltZUNvZGU9InRleHQvcGxhaW4iPkFXL1l0Tml4Mll
QWXFTRFlxdG1JMkxIWml0aXZJTmluMllUWXF0bUQyWWJaaU5tRTJZallyTm1LMktjZzJLallvOW1DMkx
YWmlTRFlzOWl4MkxuWXFTRFlwOW1FMllYWXJkaXYyWWpZcjlpcElId2dUV0Y0YVcxMWJTQlRjR1ZsW
kNCVVpXTm9JRk4xY0hCc2VTQk1WRVFDRHpNNU9UazVPVGs1T1Rrd01EQXdNd01UWWpBeU1pMH
dPUzB3TjFReE1qb3lNVG95T0FRRU5DNDJNQVVETUM0MkJpeG1LekJYUTNGdVVHdEpia2tyWlV3NVJ
6Tk1RWEo1TVRKbVZGQm1LM1J2UXpsVldEQTNSalJtU1N0elBRZGdUVVZWUTBsQ2VIbFNPSEpqTkV
zNE56STRkMlJUUmpSWVUwUnhVSE1yY2tsTUt6TlVSbWc1YlN0aFRuaFJVSFJJUUVdsRlFUWmpTR0Z3
U1hSMmNERXplVTFUZFRZMlRtSlBaekpEY0c5dFNIZFZMjVaU2psb05uVkhVVFkxWVZrOUNGZ3dWak
FRQmdjcWhrak9QUUlCQmdVcmdRUUFDZ05DQUFTaFJJcHJSSnIwVWdTdE02L1M0Q1FMVlVncGZGV
DJjK25IYStWL2pLRXg2UEx4elRaY2x1VU9ydTAvSjJqeWFyUnFFNHlZMmp5RENlTHRlM1VwUDFSNDw
vY2JjOkVtYmVkZGVkRG9jdW1lbnRCaW5hcnlPYmplY3Q+CiAgICAgICAgPC9jYWM6QXR0YWNobWVud
D4KPC9jYWM6QWRkaXRpb25hbERvY3VtZW50UmVmZXJlbmNlPjxjYWM6U2lnbmF0dXJlPgogICAgICA8
Y2JjOklEPnVybjpvYXNpczpuYW1lczpzcGVjaWZpY2F0aW9uOnVibDpzaWduYXR1cmU6SW52b2ljZTwv
Y2JjOklEPgogICAgICA8Y2JjOlNpZ25hdHVyZU1ldGhvZD51cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdG
lvbjp1Ymw6ZHNpZzplbnZlbG9wZWQ6eGFkZXM8L2NiYzpTaWduYXR1cmVNZXRob2Q+CjwvY2FjOlNpZ
25hdHVyZT48Y2FjOkFjY291bnRpbmdTdXBwbGllclBhcnR5PgogICAgICAgIDxjYWM6UGFydHk+CiAgICA
gICAgICAgIDxjYWM6UGFydHlJZGVudGlmaWNhdGlvbj4KICAgICAgICAgICAgICAgIDxjYmM6SUQgc2No
ZW1lSUQ9IkNSTiI+MTAxMDAxMDAwMDwvY2JjOklEPgogICAgICAgICAgICA8L2NhYzpQYXJ0eUlkZW5
0aWZpY2F0aW9uPgogICAgICAgICAgICA8Y2FjOlBvc3RhbEFkZHJlc3M+CiAgICAgICAgICAgICAgICA8Y
2JjOlN0cmVldE5hbWU+2KfZhNin2YXZitixINiz2YTYt9in2YYgfCBQcmluY2UgU3VsdGFuPC9jYmM6U3RyZ
WV0TmFtZT4KICAgICAgICAgICAgICAgIDxjYmM6QnVpbGRpbmdOdW1iZXI+MjMyMjwvY2JjOkJ1aWxka
W5nTnVtYmVyPgogICAgICAgICAgICAgICAgPGNiYzpDaXR5U3ViZGl2aXNpb25OYW1lPtin2YTZhdix2Kj
YuSB8IEFsLU11cmFiYmE8L2NiYzpDaXR5U3ViZGl2aXNpb25OYW1lPgogICAgICAgICAgICAgICAgPGNi
YzpDaXR5TmFtZT7Yp9mE2LHZitin2LYgfCBSaXlhZGg8L2NiYzpDaXR5TmFtZT4KICAgICAgICAgICAgIC
AgIDxjYmM6UG9zdGFsWm9uZT4yMzMzMzwvY2JjOlBvc3RhbFpvbmU+CiAgICAgICAgICAgICAgICA8Y
2FjOkNvdW50cnk+CiAgICAgICAgICAgICAgICAgICAgPGNiYzpJZGVudGlmaWNhdGlvbkNvZGU+U0E8L
2NiYzpJZGVudGlmaWNhdGlvbkNvZGU+CiAgICAgICAgICAgICAgICA8L2NhYzpDb3VudHJ5PgogICAgIC
AgICAgICA8L2NhYzpQb3N0YWxBZGRyZXNzPgogICAgICAgICAgICA8Y2FjOlBhcnR5VGF4U2NoZW1lP
gogICAgICAgICAgICAgICAgPGNiYzpDb21wYW55SUQ+Mzk5OTk5OTk5OTAwMDAzPC9jYmM6Q29tcG
FueUlEPgogICAgICAgICAgICAgICAgPGNhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgICAgICAgICAgPGNi
YzpJRD5WQVQ8L2NiYzpJRD4KICAgICAgICAgICAgICAgIDwvY2FjOlRheFNjaGVtZT4KICAgICAgICAgI
CAgPC9jYWM6UGFydHlUYXhTY2hlbWU+CiAgICAgICAgICAgIDxjYWM6UGFydHlMZWdhbEVudGl0eT4
KICAgICAgICAgICAgICAgIDxjYmM6UmVnaXN0cmF0aW9uTmFtZT7YtNix2YPYqSDYqtmI2LHZitivINin2Y
TYqtmD2YbZiNmE2YjYrYNmK2Kcg2KjYo9mC2LXZiSDYs9ix2LnYqSDYp9mE2YXYrdiv2YjYr9ipIHwgTWF4
aW11bSBTcGVlZCBUZWNoIFN1cHBseSBMVEQ8L2NiYzpSZWdpc3RyYXRpb25OYW1lPgogICAgICAgI
CAgICA8L2NhYzpQYXJ0eUxlZ2FsRW50aXR5PgogICAgICAgIDwvY2FjOlBhcnR5PgogICAgPC9jYWM6Q
WNjb3VudGluZ1N1cHBsaWVyUGFydHk+CiAgICAgPGNhYzpBY2NvdW50aW5nQ3VzdG9tZXJQYXJ0eT
4KICAgICAgICA8Y2FjOlBhcnR5PgogICAgICAgICAgICA8Y2FjOlBvc3RhbEFkZHJlc3M+CiAgICAgICAgIC
AgICAgICA8Y2JjOlN0cmVldE5hbWU+2LXZhNin2K0g2KfZhNiv2YrZhiB8IFNhbGFoIEFsLURpbjwvY2JjOlN
0cmVldE5hbWU+CiAgICAgICAgICAgICAgICA8Y2JjOkJ1aWxkaW5nTnVtYmVyPjExMTE8L2NiYzpCdWls
ZGluZ051bWJlcj4KICAgICAgICAgICAgICAgIDxjYmM6Q2l0eVN1YmRpdmlzaW9uTmFtZT7Yp9mE2YXYsd
mI2KwgfCBBbC1NdXJvb2o8L2NiYzpDaXR5U3ViZGl2aXNpb25OYW1lPgogICAgICAgICAgICAgICAgPGN
iYzpDaXR5TmFtZT7Yp9mE2LHZitin2LYgfCBSaXlhZGg8L2NiYzpDaXR5TmFtZT4KICAgICAgICAgICAgIC
AgIDxjYmM6UG9zdGFsWm9uZT4xMjIyMjwvY2JjOlBvc3RhbFpvbmU+CiAgICAgICAgICAgICAgICA8Y2Fj
OkNvdW50cnk+CiAgICAgICAgICAgICAgICAgICAgPGNiYzpJZGVudGlmaWNhdGlvbkNvZGU+U0E8L2Ni
YzpJZGVudGlmaWNhdGlvbkNvZGU+CiAgICAgICAgICAgICAgICA8L2NhYzpDb3VudHJ5PgogICAgICAgI
CAgICA8L2NhYzpQb3N0YWxBZGRyZXNzPgogICAgICAgICAgICA8Y2FjOlBhcnR5VGF4U2NoZW1lPgog
ICAgICAgICAgICAgICAgPGNiYzpDb21wYW55SUQ+Mzk5OTk5OTk5ODAwMDAzPC9jYmM6Q29tcGFu
eUlEPgogICAgICAgICAgICAgICAgPGNhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgICAgICAgICAgPGNiY
zpJRD5WQVQ8L2NiYzpJRD4KICAgICAgICAgICAgICAgIDwvY2FjOlRheFNjaGVtZT4KICAgICAgICAgICAgIC
AgPC9jYWM6UGFydHlUYXhTY2hlbWU+CiAgICAgICAgICAgIDxjYWM6UGFydHlMZWdhbEVudGl0eT4KI
CAgICAgICAgICAgICAgIDxjYmM6UmVnaXN0cmF0aW9uTmFtZT7YtNix2YPYqSDZhtmF2F2KfYsNisINmB2
KfYqtmI2LHYqSDYp9mE2YXYrdiv2YjYr9ipIHwgRmF0b29yYSBTYW1wbGVzIExURDwvY2JjOlJlZ2lzdHJ
hdGlvbk5hbWU+CiAgICAgICAgICAgIDwvY2FjOlBhcnR5TGVnYWxFbnRpdHk+CiAgICAgICAgPC9jYWM6
UGFydHk+CiAgICA8L2NhYzpBY2NvdW50aW5nQ3VzdG9tZXJQYXJ0eT4KICAgIDxjYWM6RGVsaXZlcn
nk+CiAgICAgICAgPGNiYzpBY3R1YWxEZWxpdmVyeURhdGU+MjAyMi0wOS0wNzwvY2JjOkFjdHVhbE
RlbGl2ZXJ5RGF0ZT4KICAgIDwvY2FjOkRlbGl2ZXJ5PgogICAgPGNhYzpQYXltZW50TWVhbnM+CiAgICA
gICAgPGNiYzpQYXltZW50TWVhbnNDb2RlPjEwPC9jYmM6UGF5bWVudE1lYW5zQ29kZT4KICAgIDwvY
2FjOlBheW1lbnRNZWFucz4KICAgIDxjYWM6QWxsb3dhbmNlQ2hhcmdlPgogICAgICAgIDxjYmM6Q2hhcm
dlSW5kaWNhdG9yPmZhbHNlPC9jYmM6Q2hhcmdlSW5kaWNhdG9yPgogICAgICAgIDxjYmM6QWxsb3dh
bmNlQ2hhcmdlUmVhc29uPmRpc2NvdW50PC9jYmM6QWxsb3dhbmNlQ2hhcmdlUmVhc29uPgogICAgIC
AgIDxjYmM6QW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+MC4wMDwvY2JjOkFtb3VudD4KICAgICAgICA8Y2
FjOlRheENhdGVnb3J5PgogICAgICAgICAgICA8Y2JjOklEIHNjaGVtZUlEPSJVTi9FQ0UgNTMwNSIgc2No
ZW1lQWdlbmN5SUQ9IjYiPlM8L2NiYzpJRD4KICAgICAgICAgICAgPGNiYzpQZXJjZW50PjE1PC9jYmM6U
GVyY2VudD4KICAgICAgICAgICAgPGNhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgICAgICA8Y2JjOklEIH
NjaGVtZUlEPSJVTi9FQ0UgNTE1MyIgc2NoZW1lQWdlbmN5SUQ9IjYiPlZBVDwvY2JjOklEPgogICAgICAg
ICAgICA8L2NhYzpUYXhTY2hlbWU+CiAgICAgICAgPC9jYWM6VGF4Q2F0ZWdvcnk+CiAgICA8L2NhYzp
BbGxvd2FuY2VDaGFyZ2U+CiAgICA8Y2FjOlRheFRvdGFsPgogICAgICAgIDxjYmM6VGF4QW1vdW50IG
N1cnJlbmN5SUQ9IlNBUiI+MC42PC9jYmM6VGF4QW1vdW50PgogICAgPC9jYWM6VGF4VG90YWw+Ci
AgICA8Y2FjOlRheFRvdGFsPgogICAgICAgIDxjYmM6VGF4QW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+MC
42PC9jYmM6VGF4QW1vdW50PgogICAgICAgIDxjYWM6VGF4U3VidG90YWw+CiAgICAgICAgICAgIDxjY

mM6VGF4YWJsZUFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjQuMDA8L2NiYzpUYXhhYmxlQW1vdW50Pgog
ICAgICAgICAgICA8Y2JjOlRheEFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjAuNjA8L2NiYzpUYXhBbW91bnQ
+CiAgICAgICAgICAgICA8Y2FjOlRheENhdGVnb3J5PgogICAgICAgICAgICAgICAgIDxjYmM6SUQgc2NoZ
W1lSUQ9IlVOL0VDRSA1MzA1IiBzY2hlbWVBZ2VuY3lJRD0iNiI+UzwvY2JjOklEPgogICAgICAgICAgICAgI
CAgIDxjYmM6UGVyY2VudD4xNS4wMDwvY2JjOlBlcmNlbnQ+CiAgICAgICAgICAgICAgICA8Y2FjOlRheF
NjaGVtZT4KICAgICAgICAgICAgICAgICAgIDxjYmM6SUQgc2NoZW1lSUQ9IlVOL0VDRSA1MTUzIiBzY2hl
bWVBZ2VuY3lJRD0iNiI+VkFUPC9jYmM6SUQ+CiAgICAgICAgICAgICAgICA8L2NhYzpUYXhTY2hlbWU+
CiAgICAgICAgICA8L2NhYzpUYXhDYXRlZ29yeT4KICAgICAgICA8L2NhYzpUYXhTdWJ0b3RhbD4KI
CAgIDwvY2FjOlRheFRvdGFsPgogICAgPGNhYzpMZWdhbE1vbmV0YXJ5VG90YWw+CiAgICAgICAgPG
NiYzpMaW5lRXh0ZW5zaW9uQW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+NC4wMDwvY2JjOkxpbmVFeHRl
bnNpb25BbW91bnQ+CiAgICAgICAgPGNiYzpUYXhFeGNsdXNpdmVBbW91bnQgY3VycmVuY3lJRD0iU0
FSIj40LjAwPC9jYmM6VGF4RXhjbHVzaXZlQW1vdW50PgogICAgICAgIDxjYmM6VGF4SW5jbHVzaXZlQ
W1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+NC42MDwvY2JjOlRheEluY2x1c2l2ZUFtb3VudD4KICAgICAgICA
8Y2JjOkFsbG93YW5jZVRvdGFsQW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+MC4wMDwvY2JjOkFsbG93Y
W5jZVRvdGFsQW1vdW50PgogICAgICAgIDxjYmM6UHJlcGFpZEFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPj
AuMDA8L2NiYzpQcmVwYWlkQW1vdW50PgogICAgICAgIDxjYmM6UGF5YWJsZUFtb3VudCBjdXJyZW5j
eUlEPSJTQVIiPjQuNjA8L2NiYzpQYXlhYmxlQW1vdW50PgogICAgPC9jYWM6TGVnYWxNb25ldGFyeVR
vdGFsPgogICAgPGNhYzpJbnZvaWNlTGluZT4KICAgICAgICA8Y2JjOklEPjE8L2NiYzpJRD4KICAgICAgI
CA8Y2JjOkludm9pY2VkUXVhbnRpdHkgdW5pdENvZGU9IlBDRSI+Mi4wMDAwMDA8L2NiYzpJbnZvaWNl
ZFF1YW50aXR5PgogICAgICAgIDxjYmM6TGluZUV4dGVuc2lvbkFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPj
QuMDA8L2NiYzpMaW5lRXh0ZW5zaW9uQW1vdW50PgogICAgICAgIDxjYWM6VGF4VG90YWw+CiAgIC
AgICAgICAgICA8Y2JjOlRheEFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjAuNjA8L2NiYzpUYXhBbW91bnQ+Ci
AgICAgICAgICAgICA8Y2JjOlJvdW5kaW5nQW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+NC42MDwvY2JjOlJ
vdW5kaW5nQW1vdW50PgogICAgICAgIDwvY2FjOlRheFRvdGFsPgogICAgICAgIDxjYWM6SXRlbT4KICA
gICAgICAgICAgPGNiYzpOYW1lPtmC2YTZhSDYsdi12KfYtTwvY2JjOk5hbWU+CiAgICAgICAgICAgIDxjY
WM6Q2xhc3NpZmllZFRheENhdGVnb3J5PgogICAgICAgICAgICAgICAgPGNiYzpJRD5TPC9jYmM6SUQ+
CiAgICAgICAgICAgICAgICA8Y2JjOlBlcmNlbnQ+MTUuMDA8L2NiYzpQZXJjZW50PgogICAgICAgICAgIC
AgICAgPGNhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgICAgICAgICAgPGNiYzpJRD5WQVQ8L2NiYzpJR
D4KICAgICAgICAgICAgICAgIDwvY2FjOlRheFNjaGVtZT4KICAgICAgICAgICAgPC9jYWM6Q2xhc3NpZmll
ZFRheENhdGVnb3J5PgogICAgICAgIDwvY2FjOkl0ZW0+CiAgICAgICAgPGNhYzpQcmljZT4KICAgICAgIC
AgICAgPGNiYzpQcmljZUFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjIuMDA8L2NiYzpQcmljZUFtb3VudD4KIC
AgICAgICA8L2NhYzpQcmljZT4KICAgIDwvY2FjOkludm9pY2VMaW5lPgo8L0ludm9pY2U+" } }

### Request headers

### Return type

ClearedInvoiceResultModel

### Example data

Content-Type: application/json

{ "clearanceStatus" : "CLEARED", "clearedInvoice" : "clearedInvoice", "validationResults" : [ { "warningMessages" : [ { "code" : "code", "category" : "category", "message" : "message" }, { "code" : "code", "category" : "category", "message" : "message" } ], "infoMessages" : [ { "message" : "message" }, { "message" : "message" } ], "erroMessages" : [ { "code" : "code", "category" : "category", "message" : "message" }, { "code" : "code", "category" : "category", "message" : "message" } ], "status" : "PASS" } , { "warningMessages" : [ { "code" : "code", "category" : "category", "message" : "message" }, { "code" : "code", "category" : "category", "message" : "message" } ], "infoMessages" : [ { "message" : "message" }, { "message" : "message" } ], "erroMessages" : [ { "code" : "code", "category" : "category", "message" : "message" }, { "code" : "code", "category" : "category", "message" : "message" } ], "status" : "PASS" } ] }

### 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

### Responses

**200**

HTTP OK. Returned on successful clearance of submitted invoice. ClearedInvoiceResultModel

### Example data

Content-Type: cleared

{"validationResults":{"infoMessages":[{"type":"INFO","code":"XSD_ZATCA_VALID","category":"XSD validation","message":"Complied with UBL 2.1 standards in line with ZATCA specifications","status":"PASS"}],"warningMessages":[],"errorMessages":[],"status":"PASS"},"clearanceStatus":"CLEARED","clearedInvoice":"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPEludm9pY2UgeG1sbnM9InVybjpvYXNp
czpuYW1lczpzcGVjaWZpY2F0aW9uOnVibDpzY2hlbWE6eHNkOkludm9pY2UtMiIgeG1sbnM6Y2FjPSJ1cm46b2FzaXM6bm
FtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2NoZW1hOnhzZDpDb21tb25BZ2dyZWdhdGVDb21wb25lbnRzLTIiIHhtbG5zOmNiY
z0idXJuOm9hc2lzOm5hbWVzOnNwZWNpZmljYXRpb246dWJsOnNjaGVtYTp4c2Q6Q29tbW9uQmFzaWNDb21wb25lbnRz
LTIiIHhtbG5zOmV4dD0idXJuOm9hc2lzOm5hbWVzOnNwZWNpZmljYXRpb246dWJsOnNjaGVtYTp4c2Q6Q29tbW9uRXh0
ZW5zaW9uQ29tcG9uZW50cy0yIj48ZXh0OlVCTEV4dGVuc2lvbnM+CiAgICA8ZXh0OlVCTEV4dGVuc2lvbj4KICAgICAgICA

8ZXh0OkV4dGVuc2lvblVSST51cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6ZHNpZzplbnZlbG9wZWQ6eGFk
ZXM8L2V4dDpFeHRlbnNpb25VUkk+CiAgICAgICAgPGV4dDpFeHRlbnNpb25Db250ZW50PgogICAgICAgICAgICA8c2lnOl
VCTERvY3VtZW50U2lnbmF0dXJlcyB4bWxuczpzaWc9InVybjpvYXNpczpuYW1lczpzcGVjaWZpY2F0aW9uOnVibDpzY2hlb
WE6eHNkOkNvbW1vblNpZ25hdHVyZUNvbXBvbmVudHMtMiIgeG1sbnM6c2FjPSJ1cm46b2FzaXM6bmFtZXM6c3BlY2lma
WNhdGlvbjp1Ymw6c2NoZW1hOnhzZDpTaWduYXR1cmVBZ2dyZWdhdGVDb21wb25lbnRzLTIiIHhtbG5zOnNiYz0idXJuOm
9hc2lzOm5hbWVzOnNwZWNpZmljYXRpb246dWJsOnNjaGVtYTp4c2Q6U2lnbmF0dXJlQmFzaWNDb21wb25lbnRzLTIiPgog
ICAgICAgICAgICAgICAgPHNhYzpTaWduYXR1cmVJbmZvcm1hdGlvbj4gCiAgICAgICAgICAgICAgICAgICAgPGNiYzpJRD5
1cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2lnbmF0dXJlOjE8L2NiYzpJRD4KICAgICAgICAgICAgICAg
ICA8c2JjOlJlZmVyZW5jZWRTaWduYXR1cmVJRD51cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2lnbmF0d
XJlOkludm9pY2U8L3NiYzpSZWZlcmVuY2VkU2lnbmF0dXJlSUQ+CiAgICAgICAgICAgICAgICAgICAgPGRzOlNpZ25hdHVy
ZSB4bWxuczpkcz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnIyIgSWQ9InNpZ25hdHVyZSI+CiAgICAgICA
gICAgICAgICAgICAgICAgIDxkczpTaWduZWRJbmZvPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOkNhbm9ua
WNhbGl6YXRpb25NZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDA2LzEyL3htbC1jMTRuMTEiLz4KI
CAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpTaWduYXR1cmVNZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3Ln
czLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNlY2RzYS1zaGEyNTYiLz4KICAgICAgICAgICAgICAgICAgICAgICAgICAgI
DxkczpSZWZlcmVuY2UgSWQ9Imludm9pY2VTaWduZWREYXRhIiBVUkk9IiI+CiAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgPGRzOlRyYW5zZm9ybXM+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpUcmFuc2Zvcm0
gQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy9UUi8xOTk5L1JFQy14cGF0aC0xOTk5MTExNiI+CiAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6WFBhdGg+bm90KC8vYW5jZXN0b3Itb3Itc2VsZjo6ZXh0OlVCTEV4d
GVuc2lvbnMpPC9kczpYUGF0aD4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPC9kczpUcmFuc2Zvcm0+
CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRwOi8vd3d3
LnczLm9yZy9UUi8xOTk5L1JFQy14cGF0aC0xOTk5MTExNiI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICA8ZHM6WFBhdGg+bm90KC8vYW5jZXN0b3Itb3Itc2VsZjo6Y2FjOlNpZ25hdHVyZSk8L2RzOlhQYXRoPgogICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICA8L2RzOlRyYW5zZm9ybT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgPGRzOlRyYW5zZm9ybSBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnL1RSLzE5OTkvUkVDLXhwYXRo
LTE5OTkxMTE2Ij4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpYUGF0aD5ub3QoLy9hbmNlc
3Rvci1vci1zZWxmOjpjYWM6QWRkaXRpb25hbERvY3VtZW50UmVmZXJlbmNlW2NiYzpJRD0nUVInXSk8L2RzOlhQYXRo
PgogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8L2RzOlRyYW5zZm9ybT4KICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgPGRzOlRyYW5zZm9ybSBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDYvMTIveG
1sLWMxNG4xMSIvPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgIDwvZHM6VHJhbnNmb3Jtcz4KICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICA8ZHM6RGlnZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjA
wMS8wNC94bWxlbmMjc2hhMjU2Ii8+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOkRpZ2VzdFZhbHVlPm
YrMFdDcW5Qa0luSStlTDlHM0xBcnkxMmZUUGYrdG9DOVVYMDdGNGZJK3M9PC9kczpEaWdlc3RWYWx1ZT4KICAgICA
gICAgICAgICAgICAgICAgIDwvZHM6UmVmZXJlbmNlPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOlJ
lZmVyZW5jZSBUeXBlPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjU2lnbmF0dXJlUHJvcGVydGllcyIgVV
JJPSIjeGFkZXNTaWduZWRQcm9wZXJ0aWVzIj4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6RGlnZXN
0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+CiAgICAgICAgIC
AgICAgICAgICAgICAgICAgPGRzOkRpZ2VzdFZhbHVlPk1HRTVZbUU1T1RjeFlUSmlaV1F5TkROa1pUWmtZbU1
6T1dGaVpHWTRaVEF4WkRnMVpERXlNR0l6T0dZMk1tWTJNbVk1TkdaaU5HTTFNekZqTTJZMU5nPT08L2RzOkRpZ2Vz
dFZhbHVlPgogICAgICAgICAgICAgICAgICAgICAgPC9kczpSZWZlcmVuY2U+CiAgICAgICAgICAgICAgICAgICAgI
CAgIDwvZHM6U2lnbmVkSW5mbz4KICAgICAgICAgICAgICAgICAgICAgPGRzOlNpZ25hdHVyZVZhbHVlPk1FUUNJU
URSOE4wVTQySzBOcFFFM3lwUXJTRmwzWERpbVFMV2ZNa01xL0t5MWRNQVNBSWZRdjZSOGp5VjdhUzJkWDNWYz
NnbzBtT1ByQVlJb2hvQ0VyY09zSVZxZlE9PTwvZHM6U2lnbmF0dXJlVmFsdWU+CiAgICAgICAgICAgICAgICAgICAgI
DxkczpLZXlJbmZvPgogICAgICAgICAgICAgICAgICAgICAgPGRzOlg1MDlEYXRhPgogICAgICAgICAgICAgICAgI
CAgICAgICAgIDxkczpYNTA5Q2VydGlmaWNhdGU+TUlJRDNqQ0NBNFNnQXdJQkFnSVRFUUFBT0FQRjkwQW
pzL3hjWHdBQkFBQTRBekFLQmdncWhrak9QUVFEQWpCaU1SVXdFd1lLQ1pJbWlaUHlMR1FCR1JZRmJHOWpZV3d4R
XpBUkJnb0praWFKazl/Jc1pBRVpGZ05uYjNZeEZ6QVZCZ29Ka2lhSmsvSXNaQUVaRmdkbGVIUm5ZWHAwTVJzd0dRWU
RWUVFERXhKUVVscEZTVTVXVDBsRFJJTkRRVFF0UTBFd0hoY05NalF3TVRFeE1Ea3hPVE13V2hjTk1qa3dNVEE1TU
RreE9UXdXakIxTVFzd0NRWURWUVFHRXdKVFFURW1NQ1FHQTFVRUNoTWRUUV0Y0YVcxMWJTQlRjR1ZsWkNCVV
pXTm9JRk4xY0hCc2VTQk1WRVF4RmpBVUJnTlZCQXNURFZKcGVXRmthQ0JDY21GdGkyZ3hKakFrQmdOVkJBTVRIVlJ
UVkMwNE9EWTBNekV4TkRVdE16azVPVGs1T1RrNU9UQXdNREF6TUZZd0VBWUhLb1pJemowQ0FRWUZLNEVFQUFv
RFFnQUVvV0NLYTBYTlYTlGSUVyVE92MHVBa0MxVklLWHhVOW5QcHgydmxmNHloTWVqeThjMDJYSmJsRHE3dFB5ZG
84bXEwYWhPTW1Obzhnd25pN1h0MUtUOVVlS09DQWdjd2dnSURNSUd0QmdOVkhSRUVnYVV3Z2FLa2daOHdnWnd4T
3pBNUJnTlZCQVFNTWpFdFZGTlVmREl0VkZOVWZETXRaV1F5TW1ZeFpEZ3RaVFpoTWkweE1URTRMVGxpTlRndFpE
bGhPR1l4TVdVME5EVm1NUjh3SFFZS0NaSW1pWlB5TEdRQkFRd1BNems1T1RrNU9UazVPVEF3TURBek1RMHdDd1lE
VlFRTURBUXhNVEF3TVJFd0R3WURWUVFhREFoU1VsSkVVamt5T1RFYU1CZ0dBMVVFRHd3UlUzVndjR3g1SUdGamR
HbDJhWFJwWlhNd0hRWURWUjBPQkJZRUZEZWWCtZdm1tdG5Zb0RmOUJHYktvN29jVEtZSzFNQjhHQTFVZEl3UVlNQmF
BRkp2S3FxTHRtcXdza0lGelZ2cFAyUHhUKzlObk1Ic0dDQ3NHQVFVRkJ3RUJCRzh3YlRCckJnZ3JCZ0VGQlFjd0FvWmZh
SFIwY0RvdkwyRnBZVFF1ZW1GMGFkyRXVaMjkyTG5OaEwwTmxjblJGYm5KdmJHd3ZVRkphRVdsWRtOXBZMlZUUTBFM
ExtVjRkR2R2ZW5RdVoyOTJMbXh2WTJGc1gxQlNXVXZxUGxaUFNVTkZVME5CTkMxRFFTZ3hLUzVqY25Rd0RnWURW
UjBQQVFIL0JBUURBZ2VBTUR3R0NTc0dBUVFCZ2pjVkJ3UXZNQzBHSlNnR0FRUUJnamNWQ0lHR3FCMkUwUHNTaHU
yZEpJZk8reG5Ud0ZWbWgvcWxaWVhaaEQ0Q0FXUUNBUkl3SFFZRFZSMGxCQll3RkFZSUt3WUJCUVVIQXdNR0NDc0d
BUVVGQndNQ01DY0dDU3NHQVFRQmdqY1ZDZ1FhTUJnd0NnWUlLd1lCQlFVSEF3TXdDZ1lJS3dZQkJRVUhBd0l3Q2dZ
SUtvWkl6ajBFQXdJRFNBQXdSUUloQUxFL2ljaG1uV1hDVUtVYmNhM3ljaThvcXdhTHZGZEhWalFydmVJOXVxQWJBaUE
5aEM0TThqZ01CQURQU3ptZDJ1aVBKQTZnS1IzTEUwM1U3NWVxYkMvclhBPT08L2RzOlg1MDlDZXJ0aWZpY2F0ZT4KI
CAgICAgICAgICAgICAgICAgICAgIDwvZHM6WDUwOURhdGE+CiAgICAgICAgICAgICAgICAgICAgIDwvZH
M6S2V5SW5mbz4KICAgICAgICAgICAgICAgICAgICAgPGRzOk9iamVjdD4KICAgICAgICAgICAgICAgICAgICAgI
CAgIDx4YWRlczpRdWFsaWZ5aW5nUHJvcGVydGllcyB4bWxuczp4YWRlcz0iaHR0cDovL3VyaS5ldHNpLm9yZy8wMTkwM
y92MS4zLjIjIiBUYXJnZXQ9InNpZ25hdHVyZSI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOlNpZ25
lZFByb3BlcnRpZXMgSWQ9InhhZGVzU2lnbmVkUHJvcGVydGllcyI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgIDx4YWRlczpTaWduZWRTaWduYXR1cmVQcm9wZXJ0aWVzPgogICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgPHhhZGVzOlNpZ25pbmdUaW1lPjIwMjUtMDUtMDhUMTE6NTc6MzI8L3hhZGVzOlNpZ25pbmdUaW1lPgogI

CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOlNpZ25pbmdDZXJ0aWZpY2F0ZT4KICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8eGFkZXM6Q2VydD4KICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOkNlcnREaWdlc3Q+CiAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6RGlnZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmc
vMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICA8ZHM6RGlnZXN0VmFsdWU+WkRNd01tSTBNVEUxTnpWak9UVTJOVGs0WXpWbE9EaGhZbUkwT0RVMk5EVXl
OVFUyWVRWaFlqaGhNREZtTjJGallqazFZVEEyT1dRME5qWTJNalE0TlE9PTwvZHM6RGlnZXN0VmFsdWU+CiAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDwveGFkZXM6Q2VydERpZ2VzdD4KICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOklzc3VlclNlcmlhbD4KICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpYNTA5SXNzdWVyTmFtZT5DTj1QUlpFSU5WT0lDRV
NDQTQtQ0EsIERDPWV4dGdhenQsIERDPWdvdiwgREM9bG9jYWw8L2RzOlg1MDlJc3N1ZXJOYW1lPgogICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOlg1MDlTZXJpYWxOdW1iZXI+Mzc5MTEyNzQ
yODMxMzgwNDcxODM1MjYzOTY5NTg3Mjg3NjYzNTIwNTI4Mzg3PC9kczpYNTA5U2VyaWFsTnVtYmVyPgogICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8L3hhZGVzOklzc3VlclNlcmlhbD4KICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8L3hhZGVzOkNlcnQ+CiAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICA8L3hhZGVzOlNpZ25pbmdDZXJ0aWZpY2F0ZT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgPC94YWRlczpTaWduZWRTaWduYXR1cmVQcm9wZXJ0aWVzPgogICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgIDwveGFkZXM6U2lnbmVkUHJvcGVydGllcz4KICAgICAgICAgICAgICAgICAgICAgICAgICAgIDwveGFkZXM6UXVhbGl
meWluZ1Byb3BlcnRpZXM+CiAgICAgICAgICAgICAgICAgICAgIDwvZHM6T2JqZWN0PgogICAgICAgICAgICAgICAgIC
AgIDwvZHM6U2lnbmF0dXJlPgogICAgICAgICAgICAgICAgPC9zYWM6U2lnbmF0dXJlSW5mb3JtYXRpb24+CiAgICAgICAgI
CAgIDwvc2lnOlVCTERvY3VtZW50U2lnbmF0dXJlcz4KICAgICAgICA8L2V4dDpFeHRlbnNpb25Db250ZW50PgogICAgPC9l
eHQ6VUJMRXh0ZW5zaW9uPgo8L2V4dDpVQkxFeHRlbnNpb25zPgogICAgCiAgICA8Y2JjOlByb2ZpbGVJRD5yZXBvcnRp
bmc6MS4wPC9jYmM6UHJvZmlsZUlEPgogICAgPGNiYzpJRD5TTUUwMDAyMzwvY2JjOklEPgogICAgPGNiYzpVVUlEPjhk
NDg3ODE2LTcwYjgtNGFkZS1hNjE4LTlkNjIwYjczODE0YTwvY2JjOlVVSUQ+CiAgICA8Y2JjOklzc3VlRGF0ZT4yMDIyLTA5
LTA3PC9jYmM6SXNzdWVEYXRlPgogICAgPGNiYzpJc3N1ZVRpbWU+MTI6MjE6Mjg8L2NiYzpJc3N1ZVRpbWU+CiAgICA
8Y2JjOkludm9pY2VUeXBlQ29kZSBuYW1lPSIwMTAwMDAwIj4zODg8L2NiYzpJbnZvaWNlVHlwZUNvZGU+CiAgICA8Y2Jj
OkRvY3VtZW50Q3VycmVuY3lDb2RlPlNBUjwvY2JjOkRvY3VtZW50Q3VycmVuY3lDb2RlPgogICAgPGNiYzpUYXhDdXJyZ
W5jeUNvZGU+U0FSPC9jYmM6VGF4Q3VycmVuY3lDb2RlPgogICAgPGNhYzpBZGRpdGlvbmFsRG9jdW1lbnRSZWZlcmV
uY2U+CiAgICAgICAgPGNiYzpJRD5JQ1Y8L2NiYzpJRD4KICAgICAgICA8Y2JjOlVVSUQ+MjM8L2NiYzpVVUlEPgogICAgP
C9jYWM6QWRkaXRpb25hbERvY3VtZW50UmVmZXJlbmNlPgogICAgPGNhYzpBZGRpdGlvbmFsRG9jdW1lbnRSZWZlcm
VuY2U+CiAgICAgICAgPGNiYzpJRD5QSUg8L2NiYzpJRD4KICAgICAgICA8Y2FjOkF0dGFjaG1lbnQ+CiAgICAgICAgICAgI
DxjYmM6RW1iZWRkZWREb2N1bWVudEJpbmFyeU9iamVjdCBtaW1lQ29kZT0idGV4dC9wbGFpbiI+TldabFkyVmlOalptWm
1NNE5tWXpPR1E1TlRJM09EWmpObVEyT1Raak56bGpNbVJpWXpJek9XUmtOR1U1TVdJME5qY3lPV1EzTTJFeU4yWml
OVGRsT1E9PTwvY2JjOkVtYmVkZGVkRG9jdW1lbnRCaW5hcnlPYmplY3Q+CiAgICAgICAgPC9jYWM6QXR0YWNobWVu
dD4KICAgIDwvY2FjOkFkZGl0aW9uYWxEb2N1bWVudFJlZmVyZW5jZT4KICAgIAogICAgCiAgICA8Y2FjOkFkZGl0aW9uY
WxEb2N1bWVudFJlZmVyZW5jZT4KICAgICAgICA8Y2JjOklEPlFSPC9jYmM6SUQ+CiAgICAgICAgPGNhYzpBdHRhY2htZ
W50PgogICAgICAgICAgICA8Y2JjOkVtYmVkZGVkRG9jdW1lbnRCaW5hcnlPYmplY3QgbWltZUNvZGU9InRleHQvcGxhaW
4iPkFXL1l0Tml4MllQWXFTRFlxdG1JMkxIWml0aXZJTmluMllUWXF0bUQyWWJaaU5tRTJZallyTm1LMktjZzJLallvOW1DMk
xYWmlTRFlzOWl4MkxuWXFTRFlwOW1FMllYWXJkaXYyWWpZcjlpcElId2dUV0Y0YVcxMWJTQlRjR1ZsWkNCVVpXTm9J
Rk4xY0hCc2VTQk1WRVFDRHpNNU9UazVPVGs1T1Rrd01EQXdNd01UTWpBeU1pMHdPUzB3TjFReE1qb3lNVG95T0FR
RU5DNDJNQVVETUM0MkJpeG1LekJYUTNGdVVHdEpia2tyWlV3NVJ6Tk1RWEo1TVRKbVZGQm1LM1J2UXpsVldEQTN
SalJtU1N0elBRZGdUVVZSUTBsUlJGSTRUakJWTkRKTE1FNXdVVVV6ZVhCUmNsTkdiRE5ZUkdsdFVVeFhaazFyVFhFdl
Mza3haRTFCVTBGSlpsRjJObEk0YW5sV04yRlRNbVJZTTFaak0yZHZNRzFQVUhKQldVbHZhRzlEUlhKalQzTkpWbkZtVV
QwOUNGZ3dWakFRQmdjcWhrak9QUUlCQmdVcmdRUUFDZ05DQUFTaFlJcHJSSnIwVWdTdE02L1M0Q1FMVlVncGZGV
DJjK25IYStWL2pLRXg2UEx4elRaY2x1VU9ydTAvSjJqeWFyUnFFNHlZMmp5RENlTHRlM1VwUDFSNDwvY2JjOkVtYmVkZ
GVkRG9jdW1lbnRCaW5hcnlPYmplY3Q+CiAgICAgICAgPC9jYWM6QXR0YWNobWVudD4KPC9jYWM6QWRkaXRpb25hb
ERvY3VtZW50UmVmZXJlbmNlPjxjYWM6U2lnbmF0dXJlPgogICAgICA8Y2JjOklEPnVybjpvYXNpczpuYW1lczpzcGVjaWZp
Y2F0aW9uOnVibDpzaWduYXR1cmU6SW52b2ljZTwvY2JjOklEPgogICAgICA8Y2JjOlNpZ25hdHVyZU1ldGhvZD51cm46b2
FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6ZHNpZzplbnZlbG9wZWQ6eGFkZXM8L2NiYzpTaWduYXR1cmVNZXRob
2Q+CjwvY2FjOlNpZ25hdHVyZT48Y2FjOkFjY291bnRpbmdTdXBwbGllclBhcnR5PgogICAgICAgIDxjYWM6UGFydHk+CiAgI
CAgICAgICAgIDxjYWM6UGFydHlJZGVudGlmaWNhdGlvbj4KICAgICAgICAgICAgIDxjYmM6SUQgc2NoZW1lSUQ9Ik
NSTiI+MTAxMDAxMDAwMDwvY2JjOklEPgogICAgICAgICAgICA8L2NhYzpQYXJ0eUlkZW50aWZpY2F0aW9uPgogICAgIC
AgICAgICA8Y2FjOlBvc3RhbEFkZHJlc3M+CiAgICAgICAgICAgICA8Y2JjOlN0cmVldE5hbWU+2KfZhNin2YXZitixINiz2Y
TYt9in2YYgfCBQcmluY2UgU3VsdGFuPC9jYmM6U3RyZWV0TmFtZT4KICAgICAgICAgICAgIDxjYmM6QnVpbGRpbm
dOdW1iZXI+MjMyMjwvY2JjOkJ1aWxkaW5nTnVtYmVyPgogICAgICAgICAgICAgPGNiYzpDaXR5U3ViZGl2aXNpb25O
YW1lPtin2YTZhdix2KjYuSB8IEFsLU11cmFiYmE8L2NiYzpDaXR5U3ViZGl2aXNpb25OYW1lPgogICAgICAgICAgICAg
PGNiYzpDaXR5TmFtZT7Yp9mE2LHZitin2LYgfCBSaXlhZGg8L2NiYzpDaXR5TmFtZT4KICAgICAgICAgICAgIDxjYmM
6UG9zdGFsWm9uZT4yMzMzMzwvY2JjOlBvc3RhbFpvbmU+CiAgICAgICAgICAgICA8Y2FjOkNvdW50cnk+CiAgICAgI
CAgICAgICAgICAgPGNiYzpJZGVudGlmaWNhdGlvbkNvZGU+U0E8L2NiYzpJZGVudGlmaWNhdGlvbkNvZGU+CiAgI
CAgICAgICAgICA8L2NhYzpDb3VudHJ5PgogICAgICAgICAgICA8L2NhYzpQb3N0YWxBZGRyZXNzPgogICAgICAgIC
AgICA8Y2FjOlBhcnR5VGF4U2NoZW1lPgogICAgICAgICAgICAgPGNiYzpDb21wYW55SUQ+Mzk5OTk5OTk5OTAwM
DAzPC9jYmM6Q29tcGFueUlEPgogICAgICAgICAgICAgPGNhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgICAgICA
gPGNiYzpJRD5WQVQ8L2NiYzpJRD4KICAgICAgICAgICAgIDwvY2FjOlRheFNjaGVtZT4KICAgICAgICAgPC9jY
WM6UGFydHlUYXhTY2hlbWU+CiAgICAgICAgIDxjYWM6UGFydHlMZWdhbEVudGl0eT4KICAgICAgICAgICAgIDx
jYmM6UmVnaXN0cmF0aW9uTmFtZT7YtNix2YPYqSDYqtmI2LHZitivINin2YTYqtmD2YbZiNmE2YjYrNmK2Kcg2KjYo9mC2L
XZiSDYs9ix2LnYqSDYp9mE2YXYrdiv2YjYr9ipIHwgTWF4aW11bSBTcGVlZCBUZWNoIFN1cHBseSBMVEQ8L2NiYzpSZW
dpc3RyYXRpb25OYW1lPgogICAgICAgICAgICA8L2NhYzpQYXJ0eUxlZ2FsRW50aXR5PgogICAgICAgIDwvY2FjOlBhcnR5
PgogICAgPC9jYWM6QWNjb3VudGluZ1N1cHBsaWVyUGFydHk+CiAgICAgPGNhYzpBY2NvdW50aW5nQ3VzdG9tZXJQY
XJ0eT4KICAgICAgICA8Y2FjOlBhcnR5PgogICAgICAgICAgICA8Y2FjOlBvc3RhbEFkZHJlc3M+CiAgICAgICAgICAgICA
8Y2JjOlN0cmVldE5hbWU+2LXZhNin2K0g2KfZhNiv2YrZhiB8IFNhbGFoIEFsLURpbjwvY2JjOlN0cmVldE5hbWU+CiAgICAgI
CAgICAgICA8Y2JjOkJ1aWxkaW5nTnVtYmVyPjExMTE8L2NiYzpCdWlsZGluZ051bWJlcj4KICAgICAgICAgICAgI
DxjYmM6Q2l0eVN1YmRpdmlzaW9uTmFtZT7Yp9mE2YXYsdmI2KwgfCBBbC1NdXJvb2o8L2NiYzpDaXR5U3ViZGl2aXNpb

```
25OYW1lPgogICAgICAgICAgICAgICAgPGNiYzpDaXR5TmFtZT7Yp9mE2LHZitin2LYgfCBSaXlhZGg8L2NiYzpDaXR5TmFt
ZT4KICAgICAgICAgICAgICAgIDxjYmM6UG9zdGFsWm9uZT4xMjIyMjwvY2JjOlBvc3RhbFpvbmU+CiAgICAgICAgICAgICAg
ICA8Y2FjOkNvdW50cnk+CiAgICAgICAgICAgICAgICAgICAgPGNiYzpJZGVudGlmaWNhdGlvbkNvZGU+U0E8L2NiYzpJZG
VudGlmaWNhdGlvbkNvZGU+CiAgICAgICAgICAgICAgICA8L2NhYzpDb3VudHJ5PgogICAgICAgICAgICA8L2NhYzpQb3N0
YWxBZGRyZXNzPgogICAgICAgICAgICA8Y2FjOlBhcnR5VGF4U2NoZW1lPgogICAgICAgICAgICAgICAgPGNiYzpDb21wY
W55SUQ+Mzk5OTk5OTk5ODAwMDAzPC9jYmM6Q29tcGFueUlEPgogICAgICAgICAgICAgICAgPGNhYzpUYXhTY2hlbWU
+CiAgICAgICAgICAgICAgICAgICAgPGNiYzpJRD5WQVQ8L2NiYzpJRD4KICAgICAgICAgICAgICAgIDwvY2FjOlRheFNjaG
VtZT4KICAgICAgICAgICAgPC9jYWM6UGFydHlUYXhTY2hlbWU+CiAgICAgICAgICAgIDxjYWM6UGFydHlMZWdhbEVudGl
0eT4KICAgICAgICAgICAgICAgIDxjYmM6UmVnaXN0cmF0aW9uTmFtZT7YtNix2YPYqSDZhtmF2KfYsNisINmB2KfYqtmI2L
HYqSDYp9mE2YXYrdiv2YjYr9ipIHwgRmF0b29yYSBTYW1wbGVzIExURDwvY2JjOlJlZ2lzdHJhdGlvbk5hbWU+CiAgICAgIC
AgICAgIDwvY2FjOlBhcnR5TGVnYWxFbnRpdHk+CiAgICAgICAgPC9jYWM6UGFydHk+CiAgICA8L2NhYzpBY2NvdW50aW
5nQ3VzdG9tZXJQYXJ0eT4KICAgIDxjYWM6RGVsaXZlcnk+CiAgICAgICAgPGNiYzpBY3R1YWxEZWxpdmVyeURhdGU+M
jAyMi0wOS0wNzwvY2JjOkFjdHVhbERlbGl2ZXJ5RGF0ZT4KICAgIDwvY2FjOkRlbGl2ZXJ5PgogICAgPGNhYzpQYXltZW50
TWVhbnM+CiAgICAgICAgPGNiYzpQYXltZW50TWVhbnNDb2RlPjEwPC9jYmM6UGF5bWVudE1lYW5zQ29kZT4KICAgIDw
vY2FjOlBheW1lbnRNZWFucz4KICAgIDxjYWM6QWxsb3dhbmNlQ2hhcmdlPgogICAgICAgIDxjYmM6Q2hhcmdlSW5kaWNh
dG9yPmZhbHNlPC9jYmM6Q2hhcmdlSW5kaWNhdG9yPgogICAgICAgIDxjYmM6QWxsb3dhbmNlQ2hhcmdlUmVhc29uPm
Rpc2NvdW50PC9jYmM6QWxsb3dhbmNlQ2hhcmdlUmVhc29uPgogICAgICAgIDxjYmM6QW1vdW50IGN1cnJlbmN5SUQ9Il
NBUiI+MC4wMDwvY2JjOkFtb3VudD4KICAgICAgICA8Y2FjOlRheENhdGVnb3J5PgogICAgICAgICAgICA8Y2JjOklEIHNjaG
VtZUlEPSJVTi9FQ0UgNTMwNSIgc2NoZW1lQWdlbmN5SUQ9IjYiPlM8L2NiYzpJRD4KICAgICAgICAgICAgPGNiYzpQZXJj
ZW50PjE1PC9jYmM6UGVyY2VudD4KICAgICAgICAgICAgPGNhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgICAgICA8Y2JjO
klEIHNjaGVtZUlEPSJVTi9FQ0UgNTE1MyIgc2NoZW1lQWdlbmN5SUQ9IjYiPlZBVDwvY2JjOklEPgogICAgICAgICAgICA8L2
NhYzpUYXhTY2hlbWU+CiAgICAgICAgPC9jYWM6VGF4Q2F0ZWdvcnk+CiAgICA8L2NhYzpBbGxvd2FuY2VDaGFyZ2U+C
iAgICA8Y2FjOlRheFRvdGFsPgogICAgICAgIDxjYmM6VGF4QW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+MC42PC9jYmM6VG
F4QW1vdW50PgogICAgPC9jYWM6VGF4VG90YWw+CiAgICA8Y2FjOlRheFRvdGFsPgogICAgICAgIDxjYmM6VGF4QW1v
dW50IGN1cnJlbmN5SUQ9IlNBUiI+MC42PC9jYmM6VGF4QW1vdW50PgogICAgICAgIDxjYWM6VGF4U3VidG90YWw+CiA
gICAgICAgICAgIDxjYmM6VGF4YWJsZUFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjQuMDA8L2NiYzpUYXhhYmxlQW1vdW50
PgogICAgICAgICAgICA8Y2JjOlRheEFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjAuNjA8L2NiYzpUYXhBbW91bnQ+CiAgICAgI
CAgICAgICA8Y2FjOlRheENhdGVnb3J5PgogICAgICAgICAgICAgICAgIDxjYmM6SUQgc2NoZW1lSUQ9IlVOL0VDRSA1Mz
A1IiBzY2hlbWVBZ2VuY3lJRD0iNiI+UzwvY2JjOklEPgogICAgICAgICAgICAgICAgIDxjYmM6UGVyY2VudD4xNS4wMDwvY2
JjOlBlcmNlbnQ+CiAgICAgICAgICAgICAgICA8Y2FjOlRheFNjaGVtZT4KICAgICAgICAgICAgICAgICAgICAgIDxjYmM6SUQgc2No
ZW1lSUQ9IlVOL0VDRSA1MTUzIiBzY2hlbWVBZ2VuY3lJRD0iNiI+VkFUPC9jYmM6SUQ+CiAgICAgICAgICAgICAgICA8L2
NhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgICA8L2NhYzpUYXhDYXRlZ29yeT4KICAgICAgICA8L2NhYzpUYXhTdWJ0b3R
hbD4KICAgIDwvY2FjOlRheFRvdGFsPgogICAgPGNhYzpMZWdhbE1vbmV0YXJ5VG90YWw+CiAgICAgICAgPGNiYzpMa
W5lRXh0ZW5zaW9uQW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+NC4wMDwvY2JjOkxpbmVFeHRlbnNpb25BbW91bnQ+CiAg
ICAgICAgPGNiYzpUYXhFeGNsdXNpdmVBbW91bnQgY3VycmVuY3lJRD0iU0FSIj40LjAwPC9jYmM6VGF4RXhjbHVzaXZl
QW1vdW50PgogICAgICAgIDxjYmM6VGF4SW5jbHVzaXZlQW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+NC42MDwvY2JjOlRh
eEluY2x1c2l2ZUFtb3VudD4KICAgICAgICA8Y2JjOkFsbG93YW5jZVRvdGFsQW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+MC4
wMDwvY2JjOkFsbG93YW5jZVRvdGFsQW1vdW50PgogICAgICAgIDxjYmM6UHJlcGFpZEFtb3VudCBjdXJyZW5jeUlEPSJT
QVIiPjAuMDA8L2NiYzpQcmVwYWlkQW1vdW50PgogICAgICAgIDxjYmM6UGF5YWJsZUFtb3VudCBjdXJyZW5jeUlEPSJT
QVIiPjQuNjA8L2NiYzpQYXlhYmxlQW1vdW50PgogICAgPC9jYWM6TGVnYWxNb25ldGFyeVRvdGFsPgogICAgPGNhYzpJ
bnZvaWNlTGluZT4KICAgICAgICA8Y2JjOklEPjE8L2NiYzpJRD4KICAgICAgICA8Y2JjOkludm9pY2VkUXVhbnRpdHkgdW5p
dENvZGU9IlBDRSI+Mi4wMDAwMDA8L2NiYzpJbnZvaWNlZFF1YW50aXR5PgogICAgICAgIDxjYmM6TGluZUV4dGVuc2lvb
kFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjQuMDA8L2NiYzpMaW5lRXh0ZW5zaW9uQW1vdW50PgogICAgICAgIDxjYWM6V
GF4VG90YWw+CiAgICAgICAgICAgICA8Y2JjOlRheEFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjAuNjA8L2NiYzpUYXhBbW91
bnQ+CiAgICAgICAgICAgICA8Y2JjOlJvdW5kaW5nQW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+NC42MDwvY2JjOlJvdW5kaW
5nQW1vdW50PgogICAgICAgIDwvY2FjOlRheFRvdGFsPgogICAgICAgIDxjYWM6SXRlbT4KICAgICAgICAgICAgPGNiYzpO
YW1lPtmC2YTZhSDYsdi12KfYtTwvY2JjOk5hbWU+CiAgICAgICAgICAgIDxjYWM6Q2xhc3NpZmllZFRheENhdGVnb3J5Pgo
gICAgICAgICAgICAgICAgPGNiYzpJRD5TPC9jYmM6SUQ+CiAgICAgICAgICAgICAgICA8Y2JjOlBlcmNlbnQ+MTUuMDA8L
2NiYzpQZXJjZW50PgogICAgICAgICAgICAgICAgPGNhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgICAgICAgICAgPGNiYzpJ
RD5WQVQ8L2NiYzpJRD4KICAgICAgICAgICAgICAgIDwvY2FjOlRheFNjaGVtZT4KICAgICAgICAgICAgPC9jYWM6Q2xhc3
NpZmllZFRheENhdGVnb3J5PgogICAgICAgIDwvY2FjOkl0ZW0+CiAgICAgICAgPGNhYzpQcmljZT4KICAgICAgICAgICAgP
GNiYzpQcmljZUFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjIuMDA8L2NiYzpQcmljZUFtb3VudD4KICAgICAgICAgICAgPGNhYz
pBbGxvd2FuY2VDaGFyZ2U+CiAgICAgICAgICAgICAgICAgIDxjYmM6Q2hhcmdlSW5kaWNhdG9yPnRydWU8L2NiYzpDaGFyZ2
VJbmRpY2F0b3I+CiAgICAgICAgICAgICAgICAgIDxjYmM6QWxsb3dhbmNlQ2hhcmdlUmVhc29uPmRpc2NvdW50PC9jYmM6Q
Wxsb3dhbmNlQ2hhcmdlUmVhc29uPgogICAgICAgICAgICAgICAgICA8Y2JjOkFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjAuMDA8L
2NiYzpBbW91bnQ+CiAgICAgICAgICAgIDwvY2FjOkFsbG93YW5jZUNoYXJnZT4KICAgICAgICA8L2NhYzpQcmljZT4KICAg
IDwvY2FjOkludm9pY2VMaW5lPgo8L0ludm9pY2U+"}
```

**202**

HTTP Accepted. Returned on clearance of submitted invoice with warnings. ClearedInvoiceResultModel

**Example data**

Content-Type: cleared

```
{"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":[{"type":"WARNING","code":"BR-KSA-
51","category":"KSA","message":"The line amount with VAT (KSA-12) must be Invoice line net amount (BT-131) + Line VAT
amount (KSA-11).","status":"WARNING"}],"errorMessages":[],"status":"WARNING"},"clearanceStatus":"CLEARED","clearedI
nvoice":"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPEludm9pY2UgeG1sbnM9InVybjpvYXNpczpuY
W1lczpzcGVjaWZpY2F0aW9uOnVibDpzY2hlbWE6eHNkOkludm9pY2UtMiIgeG1sbnM6Y2FjPSJ1cm46b2FzaXM6bmFtZXM
6c3BlY2lmaWNhdGlvbjp1Ymw6c2NoZW1hOnhzZDpDb21tb25BZ2dyZWdhdGVDb21wb25lbnRzLTIiIHhtbG5zOmNiYz0idXJ
uOm9hc2lzOm5hbWVzOnNwZWNpZmljYXRpb246dWJsOnNjaGVtYTp4c2Q6Q29tbW9uQmFzaWNDb21wb25lbnRzLTIiIHh
```

tbG5zOmV4dD0idXJuOm9hc2lzOm5hbWVzOnNwZWNpZmljYXRpb246dWJsOnNjaGVtYTp4c2Q6Q29tbW9uRXh0ZW5za
W9uQ29tcG9uZW50cy0yIj48ZXh0OlVCTEV4dGVuc2lvbnM+CiAgICA8ZXh0OlVCTEV4dGVuc2lvbj4KICAgICAgICA8ZXh0
OkV4dGVuc2lvblVSST51cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6ZHNpZzpzbnZlbG9wZWQ6eGFkZXM8L
2V4dDpFeHRlbnNpb25VUkk+CiAgICAgICAgPGV4dDpFeHRlbnNpb25Db250ZW50PgogICAgICAgICAgICA8c2lnOlVCTER
vY3VtZW50U2lnbmF0dXJlcyB4bWxuczpzaWc9InVybjpvYXNpczpuYW1lczpzcGVjaWZpY2F0aW9uOnVibDpzY2hlbWE6eH
NkOkNvbW1vblNpZ25hdHVyZUNvbXBvbmVudHMtMiIgeG1sbnM6c2FjPSJ1cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGl
vbjp1Ymw6c2NoZW1hOnhzZDpTaWduYXR1cmVBZ2dyZWdhdGVDb21wb25lbnRzLTIiIHhtbG5zOnNiYz0idXJuOm9hc2lzO
m5hbWVzOnNwZWNpZmljYXRpb246dWJsOnNjaGVtYTp4c2Q6U2lnbmF0dXJlQmFzaWNDb21wb25lbnRzLTIiPgogICAgIC
AgICAgICAgICAgPHNhYzpTaWduYXR1cmVJbmZvcm1hdGlvbj4gCiAgICAgICAgICAgICAgICAgPGNiYzpJRD51cm46
b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2lnbmF0dXJlOjE8L2NiYzpJRD4KICAgICAgICAgICAgICAgICA8c
2JjOlJlZmVyZW5jZWRTaWduYXR1cmVJRD51cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2lnbmF0dXJlOkl
udm9pY2U8L3NiYzpSZWZlcmVuY2VkU2lnbmF0dXJlSUQ+CiAgICAgICAgICAgICAgICAgPGRzOlNpZ25hdHVyZSB4b
Wxuczpkcz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnIyIgSWQ9InNpZ25hdHVyZSI+CiAgICAgICAgI
CAgICAgICAgICAgIDxkczpTaWduZWRJbmZvPgogICAgICAgICAgICAgICAgICAgICAgICAgPGRzOkNhbm9uaWNhbG
l6YXRpb25NZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDA2LzEyL3htbC1jMTRuMTEiLz4KICAgICAg
ICAgICAgICAgICAgICAgICAgIDxkczpTaWduYXR1cmVNZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy
8yMDAxLzA0L3htbGRzaWctbW9yZSNlY2RzYS1zaGEyNTYiLz4KICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpSZ
WZlcmVuY2UgSWQ9Imludm9pY2VTaWduZWREYXRhIiBVUkk9IiI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
PGRzOlRyYW5zZm9ybXM+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpUcmFuc2Zvcm0gQWxnb3
JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy9UUi8xOTk5L1JFQy14cGF0aC0xOTk5MTExNiI+CiAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICA8ZHM6WFBhdGg+bm90KC8vYW5jZXN0b3Itb3Itc2VsZjo6ZXh0OlVCTEV4dGVuc2lvbn
MpPC9kczpYUGF0aD4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPC9kczpUcmFuc2Zvcm0+CiAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgIDxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9y
Zy9UUi8xOTk5L1JFQy14cGF0aC0xOTk5MTExNiI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8Z
HM6WFBhdGg+bm90KC8vYW5jZXN0b3Itb3Itc2VsZjo6Y2FjOlNpZ25hdHVyZSk8L2RzOlhQYXRoPgogICAgICAgICAg
ICAgICAgICAgICAgICAgICA8L2RzOlRyYW5zZm9ybT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgPGRzOlRyYW5zZm9ybSBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnL1RSLzE5OTkvUkVDLXhwYXRoLTE5OTkx
MTE2Ij4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpYUGF0aD5ub3QoLy9hbmNlc3Rvci1vci1
zZWxmOjpjYWM6QWRkaXRpb25hbERvY3VtZW50UmVmZXJlbmNlW2NiYzpJRD0nUVInXSk8L2RzOlhQYXRoPgogICAgI
CAgICAgICAgICAgICAgICAgICAgICA8L2RzOlRyYW5zZm9ybT4KICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgPGRzOlRyYW5zZm9ybSBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDYvMTIveG1sLWMxN
G4xMSIvPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgIDwvZHM6VHJhbnNmb3Jtcz4KICAgICAgICAgICAgI
CAgICAgICAgICAgICA8ZHM6RGlnZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wN
C94bWxlbmMjc2hhMjU2Ii8+CiAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOkRpZ2VzdFZhbHVlPmtFLzhDZ
0QwcVhlRmdEc09RVnNPOVZlc3Z1TGFWMTloWVR4ZG15Sm9hODg9PC9kczpEaWdlc3RWYWx1ZT4KICAgICAgICAgIC
AgICAgICAgICAgICAgIDwvZHM6UmVmZXJlbmNlPgogICAgICAgICAgICAgICAgICAgICAgICAgPGRzOlJlZmVyZ
W5jZSBUeXBlPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjU2lnbmF0dXJlUHJvcGVydGllcyIgVVJJPSIje
GFkZXNTaWduZWRQcm9wZXJ0aWVzIj4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6RGlnZXN0TWV0a
G9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+CiAgICAgICAgI
CAgICAgICAgICAgICAgPGRzOkRpZ2VzdFZhbHVlPll6YzNZamMwWlRWaE9HTTJOVFUzWkRGbE56TXdZak14WVR
RME1XVmhPVFZrTW1JMFpqSTBNakF4TUdWbU5qUTJNVE14TnpNMk1UYzNZalJsWmpnek5BPT08L2RzOkRpZ2VzdFZ
hbHVlPgogICAgICAgICAgICAgICAgICAgICAgPC9kczpSZWZlcmVuY2U+CiAgICAgICAgICAgICAgICAgICAg
IDwvZHM6U2lnbmVkSW5mbz4KICAgICAgICAgICAgICAgICAgICAgPGRzOlNpZ25hdHVyZVZhbHVlPk1FUUNJQmd
mbUl0YTdhT3lOZ3VUWlF1NU92QlBYdjNBOW5SbG93c2YyUk5xOW95ekFpQmJ6OGtWTDlpSHRGMDUxYlZiVE94a2c3S
DN1MmxnK3g0UmQzN0NVbTQ1eUE9PTwvZHM6U2lnbmF0dXJlVmFsdWU+CiAgICAgICAgICAgICAgICAgIDxk
czpLZXlJbmZvPgogICAgICAgICAgICAgICAgICAgICAgPGRzOlg1MDlEYXRhPgogICAgICAgICAgICAgICAgI
CAgICAgIDxkczpYNTA5Q2VydGlmaWNhdGU+TUlJRVJqQ0NBK3lnQXdJQkFnSVRFUUFBRjByZENEL05PbXdP
bGdBQkFBQVhTakFLQmdncWhrak9QUVFEQWpCaU1SVXdFd1lLQ1pJbWlaUHlMR1FCR1JZRmJHOWpZV3d4RXpBUkJ
nb0praWFKay9Jc1pBRVpGZ05uYjNZeEZ6QVZCZ29Ka2lhSmsvSXNaQUVaRmdkbGVIUm5ZWHAwTVJzd0dRWURWUV
FERXhKUVVscEZTVTVXVDBsRFJWTkRRVFF0UTBFd0hoY05Nak13T0RFek1USTFPVEU1V2hjTk1qZ3dPREV4TVRJMU
9URTVXakJhTVFzd0NRWURWUVFHRXdKVFFURVZNQk1HQTFVRUNoTU1RVUpEUkNCTWFXMXBkR1ZrTVF3d0NnW
URWUVFMRXdOTFUwRXhKakFrQmdOVkJBTVRIVlJUVkMwNE9EWTBNekV4TkRVdE16RXhNVEV4TVRFeE1UQXhNVE
V6TUZZd0VBWUhLb1pJemowQ0FRWUZLNEVFQUFvRFFnQUVhZy9qQ1NRWmJVSjVPMzg2MUlKZVB6ZGdqM0FIa3ow
d0ErMFdyUW9wakk2N3BlRkRFTGxoK0JhM2JsVGduSlZxeFpRNVBHaVZkclFKa1lGTVc0alFPYU9DQW9vd2dnS0dNSUd5
QmdOVkhSRUVnYW93Z2Fla2dhUXdnYUV4T3pBNUJnTlZCQVFNTWpFdFZGTlVmREl0VkZOVWZETXRaV1F5TW1ZeFp
EZ3RaVFpoTWkweE1URTRMVGxpTlRndFpEbGhPR1l4TVdVME5EVm1NUjh3SFFZS0NaSW1pWlB5TEdRQkFRd1BNekV
3TURrME1ERXdNekF3TURBek1RMHdDd1lEVlFRTURBUXhNVEF3TVJFd0R3WURWUVFhREFoU1VsSkVNamt5T1RFZk
1CMEdBMVVFRHd3V1VtVmhiQ0JsYzNSaGRHVWdZV04wYVhacGRHbGxjekFkQmdOVkhRNEVGZ1FVOHZraEF4ME5u
QTJZR28rdkZWZHF6SnRKSVJVd0h3WURWUjBqQkJnd0ZvQVVtOHFxb3UyYXJDeVFnWE5XK2svWS9GUDcwMmN3VG
dZRFZSMGZCRWN3UlRCRG9FR2dQNFk5YUhSMGNEb3ZMMk55YkRRdWVtRjBZMkV1WjI5MkxuTmhMME5sY25SRmJ
uSnZiR3d2VUZKYVJVbE9WazlKUTBWVFEwRTBMVU5CS0RFcExtTnliRENCcUFZSUt3WUJCUVVIQVFFRWdac3dnWmd
3YXdZSUt3WUJCUVVITUFLR1gyaDBkSEE2THk5aGFXRTBMbnBoZEdOaExtZHZkaTV6WVM5RFpYSjBSVzV5YjJ4c0wx
QlNXa1ZKYm5admFXTmxVME5CTkM1bGVIUm5ZWHAwTG1kdmRpNXNiMk5oYkY5UVVscEZTVTVXVDBsRFJWTkRRV
FF0UTBFb01Ta3VZM0owTUNrR0NDc0dBUVVGQnpBQmhoMW9kSFJ3T2k4dlkzSnNOQzU2WVhSallTNW5iM1l1YzJFdmI
yTnpjREFPQmdOVkhROEJBZjhFQkFNQ0I0QXdQQVlKS3dZQkJBR0NOeFVIQkM4d0xRWWxLd1lCQkFHQ054VUlnWWF
vSFlUUSt4S0c3WjBraDg3N0dkUEFWV2FIK3FWbGhkbUVQZ0lCWkFJQkVEQWRCZ05WSFNVRUZqQVVCZ2dyQmdFRk
JRY0RBd1lJS3dZQkJRVUhBd0l3SndZSkt3WUJCQUdDTnhVS0JCb3dHREFLQmdnckJnRUZCUWNEQXpBS0JnZ3JCZ0V
GQlFjREFqQUtCZ2dxaGtqT1BRUURBZ05JQURCRkFpQkdpcVZGR2xhamwxTDkvNVErMWZmVHZzdTZyNE5RdzJ4Rnd
KL08wRkxIYVFJaEFLeFB5U1F3MnZsdnpiWW5ic0RvK0lNYTZ1dzV3TU1aczUyMEhhUkpMMlNnPC9kczpYNTA5Q2VydGl
maWNhdGU+CiAgICAgICAgICAgICAgICAgICAgICA8L2RzOlg1MDlEYXRhPgogICAgICAgICAgICAgICAgICAgIC
AgICA8L2RzOktleUluZm8+CiAgICAgICAgICAgICAgICAgIDxkczpPYmplY3Q+CiAgICAgICAgICAgICAgICAgI
CAgICA8eGFkZXM6UXVhbGlmeWluZ1Byb3BlcnRpZXMgeG1sbnM6eGFkZXM9Imh0dHA6Ly91cmkuZXRzaS5vcmcv

MDE5MDMvdjEuMy4yIyIgVGFyZ2V0PSJzaWduYXR1cmUiPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDx4YW
RlczpTaWduZWRQcm9wZXJ0aWVzIElkPSJ4YWRlc1NpZ25lZFByb3BlcnRpZXMiPgogICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICA8eGFkZXM6U2lnbmVkU2lnbmF0dXJlUHJvcGVydGllcz4KICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgIDx4YWRlczpTaWduaW5nVGltZT4yMDI1LTA1LTA4VDEzOjE2OjEyPC94YWRlczpTaWduaW5nVGlt
ZT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDx4YWRlczpTaWduaW5nQ2VydGlmaWNhdGU+Ci
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOkNlcnQ+CiAgICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgIDx4YWRlczpDZXJ0RGlnZXN0PgogICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgPGRzOkRpZ2VzdE1ldGhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzM
ub3JnLzIwMDEvMDQveG1sZW5jI3NoYTI1NiIvPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgPGRzOkRpZ2VzdFZhbHVlPk0yWm1NR1l6TXpabU1qWXhaV0psT0Raak1EUXdNVEUyWW1Vd1pEUmtZMl
ZoWkRRM1lXUTVORGN5TVRnell6WTJaREJpTnpZNFpEWTVNalV3WVRJelpRPT08L2RzOkRpZ2VzdFZhbHVlPgogICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8L3hhZGVzOkNlcnREaWdlc3Q+CiAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDx4YWRlczpJc3N1ZXJTZXJpYWw+CiAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6WDUwOUlzc3Vlck5hbWU+Q049UFJaRUlOVk9J
Q0VTQ0E0LUNBLCBEQz1leHRnYXp0LCBEQz1nb3YsIERDPWxvY2FsPC9kczpYNTA5SXNzdWVyTmFtZT4KICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpYNTA5U2VyaWFsTnVtYmVyPjM3OTExMj
Y5OTMzNTk3NzUzMDA4NDExNzk2NzUzMzM2NjYyOTE2MDE5NTkxNDwvZHM6WDUwOVNlcmlhbE51bWJlcj4KICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPC94YWRlczpJc3N1ZXJTZXJpYWw+CiAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPC94YWRlczpDZXJ0PgogICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgPC94YWRlczpTaWduaW5nQ2VydGlmaWNhdGU+CiAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgIDwveGFkZXM6U2lnbmVkU2lnbmF0dXJlUHJvcGVydGllcz4KICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgICA8L3hhZGVzOlNpZ25lZFByb3BlcnRpZXM+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8L3hhZGVzOlF1YWx
pZnlpbmdQcm9wZXJ0aWVzPgogICAgICAgICAgICAgICAgICAgICA8L2RzOk9iamVjdD4KICAgICAgICAgICAgICAgICAgICA
gICA8L2RzOlNpZ25hdHVyZT4KICAgICAgICAgICAgIDwvc2FjOlNpZ25hdHVyZUluZm9ybWF0aW9uPgogICAgICAgIC
AgICA8L3NpZzpVQkxEb2N1bWVudFNpZ25hdHVyZXM+CiAgICAgICAgPC9leHQ6RXh0ZW5zaW9uQ29udGVudD4KICAgI
DwvZXh0OlVCTEV4dGVuc2lvbj4KPC9leHQ6VUJMRXh0ZW5zaW9ucz4KICAgIAogICAgPGNiYzpQcm9maWxlSUQ+cmVw
b3J0aW5nOjEuMDwvY2JjOlByb2ZpbGVJRD4KICAgIDxjYmM6SUQ+U01FMDAwMTU8L2NiYzpJRD4KICAgIDxjYmM6VVV
JRD4zMjJlZmQ3NC01YjFiLTQxZTAtODQzYy0xZDEzYmVjNDQ3NWU8L2NiYzpVVUlEPgogICAgPGNiYzpJc3N1ZURhdG
U+MjAyMi0wOS0wNTwvY2JjOklzc3VlRGF0ZT4KICAgIDxjYmM6SXNzdWVUaW1lPjE2OjUwOjQwPC9jYmM6SXNzdWVUa
W1lPgogICAgPGNiYzpJbnZvaWNlVHlwZUNvZGUgbmFtZT0iMDEwMDAwMCI+MzgxPC9jYmM6SW52b2ljZVR5cGVDb2Rl
PgogICAgPGNiYzpEb2N1bWVudEN1cnJlbmN5Q29kZT5TQVI8L2NiYzpEb2N1bWVudEN1cnJlbmN5Q29kZT4KICAgIDxjY
mM6VGF4Q3VycmVuY3lDb2RlPlNBUjwvY2JjOlRheEN1cnJlbmN5Q29kZT4KICAgIDxjYWM6QmlsbGluZ1JlZmVyZW5jZT4
KICAgICAgICA8Y2FjOkludm9pY2VEb2N1bWVudFJlZmVyZW5jZT4KICAgICAgICAgICAgPGNiYzpJRD5TTUUwMDAwMjw
vY2JjOklEPgogICAgICAgIDwvY2FjOkludm9pY2VEb2N1bWVudFJlZmVyZW5jZT4KICAgIDwvY2FjOkJpbGxpbmdSZWZlcm
VuY2U+CiAgICA8Y2FjOkFkZGl0aW9uYWxEb2N1bWVudFJlZmVyZW5jZT4KICAgICAgICA8Y2JjOklEPklDVjwvY2JjOklEPg
ogICAgICAgIDxjYmM6VVVJRD4xNTwvY2JjOlVVSUQ+CiAgICA8L2NhYzpBZGRpdGlvbmFsRG9jdW1lbnRSZWZlcmVuY2
U+CiAgICA8Y2FjOkFkZGl0aW9uYWxEb2N1bWVudFJlZmVyZW5jZT4KICAgICAgICA8Y2JjOklEPlBJSDwvY2JjOklEPgogI
CAgICAgIDxjYWM6QXR0YWNobWVudD4KICAgICAgICAgICAgPGNiYzpFbWJlZGRlZERvY3VtZW50QmluYXJ5T2JqZWN
0IG1pbWVDb2RlPSJ0ZXh0L3BsYWluIj5OV1psWTJWaU5qWm1abU00Tm1Zek9HUTVOVEkzT0Raak5tUTJPVFpqTnpsak1
tUmlZekl6T1dSa05HVTVNV0kwTmpjeU9XUTNNMkV5TjJaaU5UZGxPUT09PC9jYmM6RW1iZWRkZWREb2N1bWVudEJp
bmFyeU9iamVjdD4KICAgICAgICA8L2NhYzpBdHRhY2htZW50PgogICAgPC9jYWM6QWRkaXRpb25hbERvY3VtZW50Um
VmZXJlbmNlPgogICAgCiAgICAKICAgIDxjYWM6QWRkaXRpb25hbERvY3VtZW50UmVmZXJlbmNlPgogICAgICAgIDxjYmM
6SUQ+UVI8L2NiYzpJRD4KICAgICAgICA8Y2FjOkF0dGFjaG1lbnQ+CiAgICAgICAgICAgIDxjYmM6RW1iZWRkZWREb2N1
bWVudEJpbmFyeU9iamVjdCBtaW1lQ29kZT0idGV4dC9wbGFpbiI+QVcvWXROaXgyWVBZcVNEWXF0bUkyTEhaaXRpdkl
OaW4yWVRZcXRtRDJZYlppTm1FMllqWXJObUsyS2NnMktqWW85bUMyTFhaaVNEWXF0bUMyTG5ZcVNEWXA5bUUyW
VhZcmRpdjJZallyOWlwSUh3Z1RXRjRhVzExYlNCVGNHVmxaQ0JVWldOb0lGTjFjSEJzZVNCTVZFUUNEek14TURBNU5E
QXhNRE13TURBd013TVRNakF5TWkwd09TMHdOVlF4TmpvMU1EbzBNQVFFTkM0Mk1BVURNQzQyQml4clJTODRRMm
RFTUhGWVpVWm5SSE5QVVZaelR6bFdaWE4yZFV4aFZqRTVhRmxVZUdSdGVVcHZZVGc0UFFkZ1RVVlJRMGxDWWJa
dFNYUmhOMkZQZVU1bmRWUmFVWFUxVDNaQ1VGaDJNMEU1YmxKc2IzZHpaakpTVG5FNWIzbDZRV2xDWW5vNGE
xWk1PV2xJZEVZd05URmlWbUpUVVDNoclp6ZElNM1V5YkdjcmVEUlNaRE0zUTFWdE5EVjRVRDA5Q0Znd1ZqQVFCZ2Nx
aGtqT1BRSUJCZ1VyZ1FRUUNnTkNBQVJxRCtNSkpCbHRRbms3ZnpyVWdsNC9OMkNQY0FlVFBUQUQ3UmF0Q2ltTWp
ydWw0VU1RdVdINEZyZHVWT0NjbFdyRmxEazhhSlYydEFtUmdVeGJpTkE1PC9jYmM6RW1iZWRkZWREb2N1bWVudEJ
pbmFyeU9iamVjdD4KICAgICAgICA8L2NhYzpBdHRhY2htZW50Pgo8L2NhYzpBZGRpdGlvbmFsRG9jdW1lbnRSZWZlcmVu
Y2U+PGNhYzpTaWduYXR1cmU+CiAgICAgIDxjYmM6SUQ+dXJuOm9hc2lzOm5hbWVzOnNwZWNpZmljYXRpb246dWJs
OnNpZ25hdHVyZTpJbnZvaWNlPC9jYmM6SUQ+CiAgICAgIDxjYmM6U2lnbmF0dXJlTWV0aG9kPnVybjpvYXNpczpuYW1lc
zpzcGVjaWZpY2F0aW9uOnVibDpkc2lnOmVudmVsb3BlZDp4YWRlczwvY2JjOlNpZ25hdHVyZU1ldGhvZD4KPC9jYWM6U2
lnbmF0dXJlPjxjYWM6QWNjb3VudGluZ1N1cHBsaWVyUGFydHk+CiAgICAgICAgPGNhYzpQYXJ0eT4KICAgICAgICAgICA
gPGNhYzpQYXJ0eUlkZW50aWZpY2F0aW9uPgogICAgICAgICAgICAgICAgPGNiYzpJRCBzY2hlbWVJRD0iQ1JOIj4xMDE
wMDEwMDAwPC9jYmM6SUQ+CiAgICAgICAgICAgIDwvY2FjOlBhcnR5SWRlbnRpZmljYXRpb24+CiAgICAgICAgICAgIDxj
YWM6UG9zdGFsQWRkcmVzcz4KICAgICAgICAgICAgICAgIDxjYmM6U3RyZWV0TmFtZT7Yp9mE2KfZhdmK2LEg2LPZhN
i32KfZhiB8IFByaW5jZSBTdWx0YW48L2NiYzpTdHJlZXROYW1lPgogICAgICAgICAgICAgICAgPGNiYzpCdWlsZGluZ051b
WJlcj4yMzIyPC9jYmM6QnVpbGRpbmdOdW1iZXI+CiAgICAgICAgICAgICAgICA8Y2JjOkNpdHlTdWJkaXZpc2lvbk5hbWU+
2KfZhNmF2LHYqNi5IHwgQWwtTXVyYWJiYTwvY2JjOkNpdHlTdWJkaXZpc2lvbk5hbWU+CiAgICAgICAgICAgICAgICA8Y2J
jOkNpdHlOYW1lPtin2YTYsdmK2KfYtiB8IFJpeWFkaDwvY2JjOkNpdHlOYW1lPgogICAgICAgICAgICAgICAgPGNiYzpQb3N
0YWxab25lPjIzMzMzPC9jYmM6UG9zdGFsWm9uZT4KICAgICAgICAgICAgICAgIDxjYWM6Q291bnRyeT4KICAgICAgICAgI
CAgICAgICAgICA8Y2JjOklkZW50aWZpY2F0aW9uQ29kZT5TQTwvY2JjOklkZW50aWZpY2F0aW9uQ29kZT4KICAgICAgIC
AgICAgICAgIDwvY2FjOkNvdW50cnk+CiAgICAgICAgICAgIDwvY2FjOlBvc3RhbEFkZHJlc3M+CiAgICAgICAgICAgIDxjYWM
6UGFydHlUYXhTY2hlbWU+CiAgICAgICAgICAgICAgICA8Y2JjOkNvbXBhbnlJRD4zMTAwOTQwMTAzMDAwMDM8L2NiYz
pDb21wYW55SUQ+CiAgICAgICAgICAgICAgICA8Y2FjOlRheFNjaGVtZT4KICAgICAgICAgICAgICAgICAgICA8Y2JjOklEPlZ
BVDwvY2JjOklEPgogICAgICAgICAgICAgICAgPC9jYWM6VGF4U2NoZW1lPgogICAgICAgICAgICA8L2NhYzpQYXJ0eVRh
eFNjaGVtZT4KICAgICAgICAgICAgPGNhYzpQYXJ0eUxlZ2FsRW50aXR5PgogICAgICAgICAgICAgICAgPGNiYzpSZWdpc3
RyYXRpb25OYW1lPti02LHZg9ipINiq2YjYsdmK2K8g2KfZhNiq2YPZhtmI2YTZiNis2YrYpyDYqNij2YLYtdmJINiz2LHYudipINin

```
2YTZhdit2K/ZiNiv2KkgfCBNYXhpbXVtIFNwZWVkIFRlY2ggU3VwcGx5IExURDwvY2JjOlJlZ2lzdHJhdGlvbk5hbWU+CiAgIC
AgICAgICAgIDwvY2FjOlBhcnR5TGVnYWxFbnRpdHk+CiAgICAgICAgPC9jYWM6UGFydHk+CiAgICA8L2NhYzpBY2NvdW
50aW5nU3VwcGxpZXJQYXJ0eT4KPGNhYzpBY2NvdW50aW5nQ3VzdG9tZXJQYXJ0eT4KICAgICAgICA8Y2FjOlBhcnR5P
gogICAgICAgICAgICA8Y2FjOlBvc3RhbEFkZHJlc3M+CiAgICAgICAgICAgICA8Y2JjOlN0cmVldE5hbWU+2LXZhNin2K
0g2KfZhNiv2YrZhiB8IFNhbGFoIEFsLURpbjwvY2JjOlN0cmVldE5hbWU+CiAgICAgICAgICAgICA8Y2JjOkJ1aWxkaW5
nTnVtYmVyPjExMTE8L2NiYzpCdWlsZGluZ051bWJlcj4KICAgICAgICAgICAgIDxjYmM6Q2l0eVN1YmRpdmlzaW9uTmF
tZT7Yp9mE2YXYsdmI2KwgfCBBbC1NdXJvb2o8L2NiYzpDaXR5U3ViZGl2aXNpb25OYW1lPgogICAgICAgICAgICAgICAgP
GNiYzpDaXR5TmFtZT7Yp9mE2LHZitin2LYgfCBSaXlhZGg8L2NiYzpDaXR5TmFtZT4KICAgICAgICAgICAgIDxjYmM6
UG9zdGFsWm9uZT4xMjIyMjwvY2JjOlBvc3RhbFpvbmU+CiAgICAgICAgICAgICA8Y2FjOkNvdW50cnk+CiAgICAgICAg
ICAgICAgICAgPGNiYzpJZGVudGlmaWNhdGlvbkNvZGU+U0E8L2NiYzpJZGVudGlmaWNhdGlvbkNvZGU+CiAgICAgI
CAgICAgICA8L2NhYzpDb3VudHJ5PgogICAgICAgICA8L2NhYzpQb3N0YWxBZGRyZXNzPgogICAgICAgIC
A8Y2FjOlBhcnR5VGF4U2NoZW1lPgogICAgICAgICAgICAgPGNiYzpDb21wYW55SUQ+Mzk5OTk5OTk5ODAwMDAz
PC9jYmM6Q29tcGFueUlEPgogICAgICAgICAgICAgPGNhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgICAgICAgPG
NiYzpJRD5WQVQ8L2NiYzpJRD4KICAgICAgICAgICAgIDwvY2FjOlRheFNjaGVtZT4KICAgICAgICAgPC9jYWM6
UGFydHlUYXhTY2hlbWU+CiAgICAgICAgIDxjYWM6UGFydHlMZWdhbEVudGl0eT4KICAgICAgICAgICAgIDxjYm
M6UmVnaXN0cmF0aW9uTmFtZT7YtNix2YPYqSDZhtmF2KfYsNisINmB2KfYqtmI2LHYqSDYp9mE2YXYrdiv2YjYr9ipIHwg
RmF0b29yYSBTYW1wbGVzIExURDwvY2JjOlJlZ2lzdHJhdGlvbk5hbWU+CiAgICAgICAgIDwvY2FjOlBhcnR5TGVnYW
xFbnRpdHk+CiAgICAgICAgPC9jYWM6UGFydHk+CiAgICA8L2NhYzpBY2NvdW50aW5nQ3VzdG9tZXJQYXJ0eT4KICAgID
xjYWM6RGVsaXZlcnk+CiAgICAgICAgPGNiYzpBY3R1YWxEZWxpdmVyeURhdGU+MjAyMi0wOS0wNTwvY2JjOkFjdHVhb
ERlbGl2ZXJ5RGF0ZT4KICAgIDwvY2FjOkRlbGl2ZXJ5PgogICAgPGNhYzpQYXltZW50TWVhbnM+CiAgICAgICAgPGNiYzp
QYXltZW50TWVhbnNDb2RlPjEwPC9jYmM6UGF5bWVudE1lYW5zQ29kZT4KICAgICAgICA8Y2JjOkluc3RydWN0aW9uTm
90ZT5DQU5DRUxMQVRJT05fT1JfVEVSTUlOQVRJT048L2NiYzpJbnN0cnVjdGlvbk5vdGU+CiAgICA8L2NhYzpQYXltZW5
0TWVhbnM+CiAgICA8Y2FjOkFsbG93YW5jZUNoYXJnZT4KICAgICAgICA8Y2JjOkNoYXJnZUluZGljYXRvcj5mYWxzZTwvY
2JjOkNoYXJnZUluZGljYXRvcj4KICAgICAgICA8Y2JjOkFsbG93YW5jZUNoYXJnZVJlYXNvbj5kaXNjb3VudDwvY2JjOkFsbG
93YW5jZUNoYXJnZVJlYXNvbj4KICAgICAgICA8Y2JjOkFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjAuMDA8L2NiYzpBbW91bn
Q+CiAgICAgICAgPGNhYzpUYXhDYXRlZ29yeT4KICAgICAgICAgICAgPGNiYzpJRCBzY2hlbWVJRD0iVU4vRUNFIDUzMD
UiIHNjaGVtZUFnZW5jeUlEPSI2Ij5TPC9jYmM6SUQ+CiAgICAgICAgICAgIDxjYmM6UGVyY2VudD4xNTwvY2JjOlBlcmNlbn
Q+CiAgICAgICAgICAgIDxjYWM6VGF4U2NoZW1lPgogICAgICAgICAgICAgPGNiYzpJRCBzY2hlbWVJRD0iVU4vRUN
FIDUxNTMiIHNjaGVtZUFnZW5jeUlEPSI2Ij5WQVQ8L2NiYzpJRD4KICAgICAgICAgICAgPC9jYWM6VGF4U2NoZW1lPgogI
CAgICAgIDwvY2FjOlRheENhdGVnb3J5PgogICAgPC9jYWM6QWxsb3dhbmNlQ2hhcmdlPgogICAgPGNhYzpUYXhUb3Rhb
D4KICAgICAgICA8Y2JjOlRheEFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjAuNjwvY2JjOlRheEFtb3VudD4KICAgIDwvY2FjOlRh
eFRvdGFsPgogICAgPGNhYzpUYXhUb3RhbD4KICAgICAgICA8Y2JjOlRheEFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjAuNjw
vY2JjOlRheEFtb3VudD4KICAgICAgICA8Y2FjOlRheFN1YnRvdGFsPgogICAgICAgICAgICA8Y2JjOlRheGFibGVBbW91bnQ
gY3VycmVuY3lJRD0iU0FSIj40LjAwPC9jYmM6VGF4YWJsZUFtb3VudD4KICAgICAgICAgICAgPGNiYzpUYXhBbW91bnQg
Y3VycmVuY3lJRD0iU0FSIj4wLjYwPC9jYmM6VGF4QW1vdW50PgogICAgICAgICAgICAgPGNhYzpUYXhDYXRlZ29yeT4KI
CAgICAgICAgICAgICAgICA8Y2JjOklEIHNjaGVtZUlEPSJVTi9FQ0UgNTMwNSIgc2NoZW1lQWdlbmN5SUQ9IjYiPlM8L2NiY
zpJRD4KICAgICAgICAgICAgICAgICA8Y2JjOlBlcmNlbnQ+MTUuMDA8L2NiYzpQZXJjZW50PgogICAgICAgICAgICAg
PGNhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgICAgICA8Y2JjOklEIHNjaGVtZUlEPSJVTi9FQ0UgNTE1MyIgc2NoZW1
lQWdlbmN5SUQ9IjYiPlZBVDwvY2JjOklEPgogICAgICAgICAgICAgPC9jYWM6VGF4U2NoZW1lPgogICAgICAgICAgIC
AgPC9jYWM6VGF4Q2F0ZWdvcnk+CiAgICAgICAgPC9jYWM6VGF4U3VidG90YWw+CiAgICA8L2NhYzpUYXhUb3RhbD4K
ICAgIDxjYWM6TGVnYWxNb25ldGFyeVRvdGFsPgogICAgICAgIDxjYmM6TGluZUV4dGVuc2lvbkFtb3VudCBjdXJyZW5jeUl
EPSJTQVIiPjQuMDA8L2NiYzpMaW5lRXh0ZW5zaW9uQW1vdW50PgogICAgICAgIDxjYmM6VGF4RXhjbHVzaXZlQW1vd
W50IGN1cnJlbmN5SUQ9IlNBUiI+NC4wMDwvY2JjOlRheEV4Y2x1c2l2ZUFtb3VudD4KICAgICAgICA8Y2JjOlRheEluY2x1c2
l2ZUFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjQuNjA8L2NiYzpUYXhJbmNsdXNpdmVBbW91bnQ+CiAgICAgICAgPGNiYzpBb
Gxvd2FuY2VUb3RhbEFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjAuMDA8L2NiYzpBbGxvd2FuY2VUb3RhbEFtb3VudD4KICAg
ICAgICA8Y2JjOlByZXBhaWRBbW91bnQgY3VycmVuY3lJRD0iU0FSIj4wLjAwPC9jYmM6UHJlcGFpZEFtb3VudD4KICAgIC
AgICA8Y2JjOlBheWFibGVBbW91bnQgY3VycmVuY3lJRD0iU0FSIj40LjYwPC9jYmM6UGF5YWJsZUFtb3VudD4KICAgIDw
vY2FjOkxlZ2FsTW9uZXRhcnlUb3RhbD4KICAgIDxjYWM6SW52b2ljZUxpbmU+CiAgICAgICAgPGNiYzpJRD4xPC9jYmM6S
UQ+CiAgICAgICAgPGNiYzpJbnZvaWNlZFF1YW50aXR5IHVuaXRDb2RlPSJQQ0UiPjIuMDAwMDAwPC9jYmM6SW52b2lj
ZWRRdWFudGl0eT4KICAgICAgICA8Y2JjOkxpbmVFeHRlbnNpb25BbW91bnQgY3VycmVuY3lJRD0iU0FSIj40LjAwPC9jY
mM6TGluZUV4dGVuc2lvbkFtb3VudD4KICAgICAgICA8Y2FjOlRheFRvdGFsPgogICAgICAgICAgICAgPGNiYzpUYXhBbW9
1bnQgY3VycmVuY3lJRD0iU0FSIj4wLjIwPC9jYmM6VGF4QW1vdW50PgogICAgICAgICAgICAgPGNiYzpSb3VuZGluZ0Ftb
3VudCBjdXJyZW5jeUlEPSJTQVIiPjQuNjA8L2NiYzpSb3VuZGluZ0Ftb3VudD4KICAgICAgICA8L2NhYzpUYXhUb3RhbD4KI
CAgICAgICA8Y2FjOkl0ZW0+CiAgICAgICAgICAgIDxjYmM6TmFtZT7ZgtmE2YU8L2NiYzpOYW1lPgogICAgICAgICAgICA8Y
2FjOkNsYXNzaWZpZWRUYXhDYXRlZ29yeT4KICAgICAgICAgICAgICAgIDxjYmM6SUQ+UzwvY2JjOklEPgogICAgICAgIC
AgICAgICAgPGNiYzpQZXJjZW50PjE1LjAwPC9jYmM6UGVyY2VudD4KICAgICAgICAgICAgICAgIDxjYWM6VGF4U2NoZW
1lPgogICAgICAgICAgICAgICAgIDxjYmM6SUQ+VkFUPC9jYmM6SUQ+CiAgICAgICAgICAgICA8L2NhYzpUYXhT
Y2hlbWU+CiAgICAgICAgICAgIDwvY2FjOkNsYXNzaWZpZWRUYXhDYXRlZ29yeT4KICAgICAgICA8L2NhYzpJdGVtPgogI
CAgICAgIDxjYWM6UHJpY2U+CiAgICAgICAgICAgIDxjYmM6UHJpY2VBbW91bnQgY3VycmVuY3lJRD0iU0FSIj4yLjAwPC9
jYmM6UHJpY2VBbW91bnQ+CiAgICAgICAgICAgIDxjYWM6QWxsb3dhbmNlQ2hhcmdlPgogICAgICAgICAgICAgICA8Y2JjO
kNoYXJnZUluZGljYXRvcj50cnVlPC9jYmM6Q2hhcmdlSW5kaWNhdG9yPgogICAgICAgICAgICAgICA8Y2JjOkFsbG93YW5j
ZUNoYXJnZVJlYXNvbj5kaXNjb3VudDwvY2JjOkFsbG93YW5jZUNoYXJnZVJlYXNvbj4KICAgICAgICAgICAgICAgPGNiYzp
BbW91bnQgY3VycmVuY3lJRD0iU0FSIj4wLjAwPC9jYmM6QW1vdW50PgogICAgICAgICAgICA8L2NhYzpBbGxvd2FuY2V
DaGFyZ2U+CiAgICAgICAgPC9jYWM6UHJpY2U+CiAgICA8L2NhYzpJbnZvaWNlTGluZT4KPC9JbnZvaWNlPg=="}
```

**208**

Invoice Hash Previously Submitted ClearedInvoiceResultModel

**Example data**

Content-Type: cleared

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

specifications","status":"PASS"}],"warningMessages":[],"errorMessages":[],"status":"PASS"},"clearanceStatus":"CLEARED","
clearedInvoice":"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPEludm9pY2UgeG1sbnM9InVybjpvYXNp
czpuYW1lczpzcGVjaWZpY2F0aW9uOnVibDpzY2hlbWE6eHNkOkludm9pY2UtMiIgeG1sbnM6Y2FjPSJ1cm46b2FzaXM6bm
FtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2NoZW1hOnhzZDpDb21tb25BZ2dyZWdhdGVDb21wb25lbnRzLTIiIHhtbG5zOmNiY
z0idXJuOm9hc2lzOm5hbWVzOnNwZWNpZmljYXRpb246dWJsOnNjaGVtYTp4c2Q6Q29tbW9uQmFzaWNDb21wb25lbnRz
LTIiIHhtbG5zOmV4dD0idXJuOm9hc2lzOm5hbWVzOnNwZWNpZmljYXRpb246dWJsOnNjaGVtYTp4c2Q6Q29tbW9uRXh0
ZW5zaW9uQ29tcG9uZW50cy0yIj48ZXh0OlVCTEV4dGVuc2lvbnM+CiAgICA8ZXh0OlVCTEV4dGVuc2lvbj4KICAgICAgICA
8ZXh0OkV4dGVuc2lvblVSST51cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6ZHNpZzplbnZlbG9wZWQ6eGFk
ZXM8L2V4dDpFeHRlbnNpb25VUkk+CiAgICAgICAgPGV4dDpFeHRlbnNpb25Db250ZW50PgogICAgICAgICAgICA8c2lnOl
VCTERvY3VtZW50U2lnbmF0dXJlcyB4bWxuczpzaWc9InVybjpvYXNpczpuYW1lczpzcGVjaWZpY2F0aW9uOnVibDpzY2hlb
WE6eHNkOkNvbW1vblNpZ25hdHVyZUNvbXBvbmVudHMtMiIgeG1sbnM6c2FjPSJ1cm46b2FzaXM6bmFtZXM6c3BlY2lma
WNhdGlvbjp1Ymw6c2NoZW1hOnhzZDpTaWduYXR1cmVBZ2dyZWdhdGVDb21wb25lbnRzLTIiIHhtbG5zOnNiYz0idXJuOm
9hc2lzOm5hbWVzOnNwZWNpZmljYXRpb246dWJsOnNjaGVtYTp4c2Q6U2lnbmF0dXJlQmFzaWNDb21wb25lbnRzLTIiPgog
ICAgICAgICAgICAgICAgPHNhYzpTaWduYXR1cmVJbmZvcm1hdGlvbj4gCiAgICAgICAgICAgICAgICAgICAgPGNiYzpJRD5
1cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2lnbmF0dXJlOjE8L2NiYzpJRD4KICAgICAgICAgICAgICAgICAg
ICA8c2JjOlJlZmVyZW5jZWRTaWduYXR1cmVJRD51cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2lnbmF0d
XJlOkludm9pY2U8L3NiYzpSZWZlcmVuY2VkU2lnbmF0dXJlSUQ+CiAgICAgICAgICAgICAgICAgICAgPGRzOlNpZ25hdHVy
ZSB4bWxuczpkcz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnIyIgSWQ9InNpZ25hdHVyZSI+CiAgICAgICA
gICAgICAgICAgICAgICAgIDxkczpTaWduZWRJbmZvPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOkNhbm9ua
WNhbGl6YXRpb25NZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDA2LzEyL3htbC1jMTRuMTEiLz4KI
CAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpTaWduYXR1cmVNZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3Ln
czLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNlY2RzYS1zaGEyNTYiLz4KICAgICAgICAgICAgICAgICAgICAgICAgICAgI
DxkczpSZWZlcmVuY2UgSWQ9Imludm9pY2VTaWduZWREYXRhIiBVUkk9IiI+CiAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgPGRzOlRyYW5zZm9ybXM+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpUcmFuc2Zvcm0
gQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy9UUi8xOTk5L1JFQy14cGF0aC0xOTk5MTExNiI+CiAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6WFBhdGg+bm90KC8vYW5jZXN0b3Itb3Itc2VsZjo6ZXh0OlVCTEV4d
GVuc2lvbnMpPC9kczpYUGF0aD4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPC9kczpUcmFuc2Zvcm0+
CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRwOi8vd3d3
LnczLm9yZy9UUi8xOTk5L1JFQy14cGF0aC0xOTk5MTExNiI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICA8ZHM6WFBhdGg+bm90KC8vYW5jZXN0b3Itb3Itc2VsZjo6Y2FjOlNpZ25hdHVyZSk8L2RzOlhQYXRoPgogICAgICAg
ICAgICAgICAgICAgICAgICAgICA8L2RzOlRyYW5zZm9ybT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgPGRzOlRyYW5zZm9ybSBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnL1RSLzE5OTkvUkVDLXhwYXRo
LTE5OTkxMTE2Ij4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpYUGF0aD5ub3QoLy9hbmNlc
3Rvci1vci1zZWxmOjpjYWM6QWRkaXRpb25hbERvY3VtZW50UmVmZXJlbmNlW2NiYzpJRD0nUVInXSk8L2RzOlhQYXRo
PgogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8L2RzOlRyYW5zZm9ybT4KICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgPGRzOlRyYW5zZm9ybSBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDYvMTIveG
1sLWMxNG4xMSIvPgogICAgICAgICAgICAgICAgICAgICAgICAgIDwvZHM6VHJhbnNmb3Jtcz4KICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICA8ZHM6RGlnZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjA
wMS8wNC94bWxlbmMjc2hhMjU2Ii8+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOkRpZ2VzdFZhbHVlPm
YrMFdDcW5Qa0luSStlTDlHM0xBcnkxMmZUUGYrdG9DOVVYMDdGNGZJK3M9PC9kczpEaWdlc3RWYWx1ZT4KICAgICA
gICAgICAgICAgICAgICAgIDwvZHM6UmVmZXJlbmNlPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOlJ
lZmVyZW5jZSBUeXBlPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjU2lnbmF0dXJlUHJvcGVydGllcyIgVV
JJPSIjeGFkZXNTaWduZWRQcm9wZXJ0aWVzIj4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6RGlnZXN
0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+CiAgICAgICAgIC
AgICAgICAgICAgICAgICAgPGRzOkRpZ2VzdFZhbHVlPk1EWTBZbUppTTJJMU5tTTFORFF6T1dZME9ESTJNb
UkyTXpBNU1qSTVNVFpqTVRoaFpqZ3pabVl6WkdVME1UWTVPRGsxTmpaak16TTVOakJrWlROaE5nPT08L2RzOkRpZ2
VzdFZhbHVlPgogICAgICAgICAgICAgICAgICAgICAgICAgPC9kczpSZWZlcmVuY2U+CiAgICAgICAgICAgICAgIC
AgICAgIDwvZHM6U2lnbmVkSW5mbz4KICAgICAgICAgICAgICAgICAgICAgPGRzOlNpZ25hdHVyZVZhbHVlPk1FUUN
JRUF2TkVXMjJnbjFXOEd4dC9SSUdjQ1NuSzUwVXBLV1oxOWhaell3REpoOEFpQkMvTXZreVZCNlZ5WnZzQ3IzNGNSb
EV0YUFMcVJHWWQydjFLajFCLythekE9PTwvZHM6U2lnbmF0dXJlVmFsdWU+CiAgICAgICAgICAgICAgICAgICAgID
xkczpLZXlJbmZvPgogICAgICAgICAgICAgICAgICAgICAgPGRzOlg1MDlEYXRhPgogICAgICAgICAgICAgIC
AgICAgICAgICAgIDxkczpYNTA5Q2VydGlmaWNhdGU+TUlJRDNqQ0NBNFNnQXdJQkFnSVRFUUFBT0FQRjkwQWp
zL3hjWHdBQkFBQTRBekFLQmdncWhrak9QUVFEQWpCaU1SVXdFd1lLQ1pJbWlaUHlMR1FCR1JZRmJHOWpZV3d4RX
pBUkJnb0praWFKay9Jc1pBRVpGZ05uYjNZeEZ6QVZCZ29Ka2lhSmsvSXNaQUVaRmdkbGVIUm5ZWHAwTVJzd0dRWU
RWUVFERXhKUVVscEZTVTVXVDBsRFJWTkRRVFF0UTBFd0hoY05NalF3TVRFeE1Ea3hPVE13V2hjTk1qa3dNVEE1TU
RreE9UTXdXakIxTVFzd0NRWURWUVFHRXdKVFFURW1NQ1FHQTFFVRUNoTWRUV0Y0YVcxMWJTQlRjR1ZsWkNCVV
pXTm9JRk4xY0hCc2VTQk1WRVF4RmpBVUJnTlZCQXNURFZKcGVXRmthQ0JDY21GdVkyZ3hKakFrQmdOVkJBTVRIVlJ
UVkMwNE9EWTBNekV4TkRVdE16azVPVGs1T1RrNU9UQXdNREF6TUZZd0VBWUhLb1pJemowQ0FRWUZLNEVFQUFv
RFFnQUVvV0NLYTBTYTlGSUVyVE92MHVBa0MxVklLWHhVOW5QcHgydmxmNHloTWVqeThjMDJYSmJsRHE3dFB5ZG
84bXEwYWhPTW1Obzhnd25pN1h0MUtUOVVlS09DQWdjd2dnSURNSUd0QmdOVkhSRUVnYVV3Z2FLa2daOHdnWnd4T
3pBNUJnTlZCQVFNTWpFdFZGTlVmREl0VkZOVWZETXRaV1F5TW1ZeFpEZ3RaVFpoTWkweE1URTRMVGxpTlRndFpE
bGhPR1l4TVdVME5EVm1NUjh3SFFZS0NaSW1pWlB5TEdRQkFRd1BNems1T1RrNU9UazVPVEF3TURBek1RMHdDd1lE
VlFRTURBUXhNVEF3TVJFd0R3WURWUVFhREFoU1VsSkVNamt5T1RFYU1CZ0dBMVVFRHd3UlUzVndjR3g1SUdGamR
HbDJhWFJwWlhNd0hRWURWUjBPQkJZRUZFWCtZdm1tdG5Zb0RmOUJHYktvN29jVEtZSzFNQjhHQTFVZEl3UVlNQmF
BRkp2S3FxTHRtcXdza0lGelZ2cFAyUHhUKzlObk1Ic0dDQ3NHQVFVRkJ3RUJCRzh3YlRCckJnZ3JCZ0VGQlFjd0FvWmZh
SFIwY0RvdkwyRnBZVFF1ZW1GMFkyRXVaMjkyTG5OaEwwTmxjblJGYm5KdmJHd3ZVRkphUlVsdWRtOXBZMlZUUTBFM
ExtVjRkR2RoZW5RdVoyOTJMbXh2WTJGc1gxQlNXa1ZKVGxaUFNVTkZVME5CTkMxRFFTZ3hLUzVqY25Rd0RnWURW
UjBQQVFIL0JBUURBZ2VBTUR3R0NTc0dBUVFCZ2pjVkJ3UXZNQzBHSlNzR0FRUUJnamNWQ0lHR3FCMkUwUHNTaHU
yZEpJZk8reG5Ud0ZWbWgvcWxaWVhaaEQ0Q0FXUUNBUkl3SFFZRFZSMGxCQll3RkFZSUt3WUJCUVVIQXdNR0NDc0d
BUVVGQndNQ01DY0dDU3NHQVFRQmdqY1ZDZ1FhTUJnd0NnWUlLd1lCQlFVSEF3TXdDZ1lJS3dZQkJRVUhBd0l3Q2dZ
SUtvWkl6ajBFQXdJRFNBQXdSUUloQUxFL2ljaG1uV1hDVUtVYmNhM3ljaThvcXdhTHZGZEhWalFydmVJOXVxQWJBaUE
5aEM0TThqZ01CQURQU3ptZDJ1aVBKQTZnS1IzTEUwM1U3NWVxYkMvclhBPT08L2RzOlg1MDlDZXJ0aWZpY2F0ZT4KI

CAgICAgICAgICAgICAgICAgICAgICAgICAgIDwvZHM6WDUwOURhdGE+CiAgICAgICAgICAgICAgICAgICAgICAgICAgIDwvZH
M6S2V5SW5mbz4KICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOk9iamVjdD4KICAgICAgICAgICAgICAgICAgICAgICAgI
CAgIDx4YWRlczpRdWFsaWZ5aW5nUHJvcGVydGllcyB4bWxuczp4YWRlcz0iaHR0cDovL3VyaS5ldHNpLm9yZy8wMTkwM
y92MS4zLjIjIiBUYXJnZXQ9InNpZ25hdHVyZSI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOlNpZ25
lZFByb3BlcnRpZXMgSWQ9InhhZGVzU2lnbmVkUHJvcGVydGllcyI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgIDx4YWRlczpTaWduZWRTaWduYXR1cmVQcm9wZXJ0aWVzPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgPHhhZGVzOlNpZ25pbmdUaW1lPjIwMjQtMDEtMTVUMTM6NTI6MDI8L3hhZGVzOlNpZ25pbmdUaW1lPgogI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOlNpZ25pbmdDZXJ0aWZpY2F0ZT4KICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8eGFkZXM6Q2VydD4KICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOkNlcnREaWdlc3Q+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6RGlnZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmc
vMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICA8ZHM6RGlnZXN0VmFsdWU+WkRNd01tSTBNVEUxTnpWak9UVTJOVGs0WXpWbE9EaGhZbUkwT0RVMk5EVXl
OVFUyWVRWaFlqaGhNREZtTjJGallqazFZVEEyT1dRME5qWTJNalE0TlE9PTwvZHM6RGlnZXN0VmFsdWU+CiAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDwveGFkZXM6Q2VydERpZ2VzdD4KICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOklzc3VlclNlcmlhbD4KICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpYNTA5SXNzdWVyTmFtZT5DTj1QUlpFSU5WT0lDRV
NDQTQtQ0EsIERDPWV4dGdhenQsIERDPWdvdiwgREM9bG9jYWw8L2RzOlg1MDlJc3N1ZXJOYW1lPgogICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOlg1MDlTZXJpYWxOdW1iZXI+Mzc5MTEyNzQ
yODMxMzgwNDcxODM1MjYzOTY5NTg3Mjg3NjYzNTIwNTI4Mzg3PC9kczpYNTA5U2VyaWFsTnVtYmVyPgogICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8L3hhZGVzOklzc3VlclNlcmlhbD4KICAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICA8L3hhZGVzOkNlcnQ+CiAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICA8L3hhZGVzOlNpZ25pbmdDZXJ0aWZpY2F0ZT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgPC94YWRlczpTaWduZWRTaWduYXR1cmVQcm9wZXJ0aWVzPgogICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgIDwveGFkZXM6U2lnbmVkUHJvcGVydGllcz4KICAgICAgICAgICAgICAgICAgICAgICAgICAgIDwveGFkZXM6UXVhbGl
meWluZ1Byb3BlcnRpZXM+CiAgICAgICAgICAgICAgICAgICAgIDwvZHM6T2JqZWN0PgogICAgICAgICAgICAgICAgIC
AgIDwvZHM6U2lnbmF0dXJlPgogICAgICAgICAgICAgICAgPC9zYWM6U2lnbmF0dXJlSW5mb3JtYXRpb24+CiAgICAgICAgI
CAgIDwvc2lnOlVCTERvY3VtZW50U2lnbmF0dXJlcz4KICAgICAgICA8L2V4dDpFeHRlbnNpb25Db250ZW50PgogICAgPC9l
eHQ6VUJMRXh0ZW5zaW9uPgo8L2V4dDpVQkxFeHRlbnNpb25zPgogICAgCiAgICA8Y2JjOlByb2ZpbGVJRD5yZXBvcnRp
bmc6MS4wPC9jYmM6UHJvZmlsZUlEPgogICAgPGNiYzpJRD5TTUUwMDAyMzwvY2JjOklEPgogICAgPGNiYzpVVUlEPjhk
NDg3ODE2LTcwYjgtNGFkZS1hNjE4LTlkNjIwYjczODE0YTwvY2JjOlVVSUQ+CiAgICA8Y2JjOklzc3VlRGF0ZT4yMDIyLTA5
LTA3PC9jYmM6SXNzdWVEYXRlPgogICAgPGNiYzpJc3N1ZVRpbWU+MTI6MjE6Mjg8L2NiYzpJc3N1ZVRpbWU+CiAgICA
8Y2JjOkludm9pY2VUeXBlQ29kZSBuYW1lPSIwMTAwMDAwIj4zODg8L2NiYzpJbnZvaWNlVHlwZUNvZGU+CiAgICA8Y2Jj
OkRvY3VtZW50Q3VycmVuY3lDb2RlPlNBUjwvY2JjOkRvY3VtZW50Q3VycmVuY3lDb2RlPgogICAgPGNiYzpUYXhDdXJyZ
W5jeUNvZGU+U0FSPC9jYmM6VGF4Q3VycmVuY3lDb2RlPgogICAgPGNhYzpBZGRpdGlvbmFsRG9jdW1lbnRSZWZlcmV
uY2U+CiAgICAgICAgPGNiYzpJRD5JQ1Y8L2NiYzpJRD4KICAgICAgICA8Y2JjOlVVSUQ+MjM8L2NiYzpVVUlEPgogICAgP
C9jYWM6QWRkaXRpb25hbERvY3VtZW50UmVmZXJlbmNlPgogICAgPGNhYzpBZGRpdGlvbmFsRG9jdW1lbnRSZWZlcm
VuY2U+CiAgICAgICAgPGNiYzpJRD5QSUg8L2NiYzpJRD4KICAgICAgICA8Y2FjOkF0dGFjaG1lbnQ+CiAgICAgICAgI
DxjYmM6RW1iZWRkZWREb2N1bWVudEJpbmFyeU9iamVjdCBtaW1lQ29kZT0idGV4dC9wbGFpbiI+TldabFkyVmlOalptWm
1NNE5tWXpPR1E1TlRJM09EWmpObVEyT1Raak56bGpNbVJpWXpJek9XUmtOR1U1TVdJME5qY3lPV1EzTTJFeU4yWml
OVGRsT1E9PTwvY2JjOkVtYmVkZGVkRG9jdW1lbnRCaW5hcnlPYmplY3Q+CiAgICAgICAgPC9jYWM6QXR0YWNobWVu
dD4KICAgIDwvY2FjOkFkZGl0aW9uYWxEb2N1bWVudFJlZmVyZW5jZT4KICAgIAogICAgCiAgICA8Y2FjOkFkZGl0aW9uY
WxEb2N1bWVudFJlZmVyZW5jZT4KICAgICAgICA8Y2JjOklEPlFSPC9jYmM6SUQ+CiAgICAgICAgPGNhYzpBdHRhY2htZ
W50PgogICAgICAgICAgICA8Y2JjOkVtYmVkZGVkRG9jdW1lbnRCaW5hcnlPYmplY3QgbWltZUNvZGU9InRleHQvcGxhaW
4iPkFXL1l0Tml4MllQWXFTRFlxdG1JMkxIWml0aXZJTmluMllUWXF0bUQyWWJaaU5tRTJZallyTm1LMktjZzJLallvOW1DMk
xYWmlTRFlzOWl4MkxuWXFTRFlwOW1FMllYWXJkaXYyWWpZcjlpcElId2dUV0Y0YVcxMWJTQlRjR1ZsWkNCVVpXTm9J
Rk4xY0hCc2VTQk1WRVFDRHpNNU9UazVPVGs1T1Rrd01EQXdNd01UTWpBeU1pMHdPUzB3TjFReE1qb3lNVG95T0FR
RU5DNDJNQVVETUM0MkJpeG1LekJYUTNGdVVHdEpia2tyWlV3NVJ6Tk1RWEo1TVRKbVZGQm1LM1J2UXpsVldEQTN
SalJtU1N0elBRZGdUVVZSUTBsRlFYWk9SVmN5TW1kdU1WYzRSM2gwTDFKSlIyTkRVMjVMTlRCVmNFdFhXakU1YUZ
wNlVqZEVTbWc0UVdsQ1F5OU5kbXQ1VmtJMlZubGFkbk5EY2pNMFkxSnNSWFJoUVV4eFVrZFpaREoyTVV0cU1VSXZL
MkY2UVQwOUNGZ3dWakFRQmdjcWhrak9QUUlCQmdVcmdRUUFDZ05DQUFTaFlJcHJSSnIwVWdTdE02L1M0Q1FMVlV
ncGZGVDJjK25IYStWL2pLRXg2UEx4elRaY2x1VU9ydTAvSjJqeWFyUnFFNHlZMmp5RENlTHRlM1VwUDFSNDwvY2JjOk
VtYmVkZGVkRG9jdW1lbnRCaW5hcnlPYmplY3Q+CiAgICAgICAgPC9jYWM6QXR0YWNobWVudD4KPC9jYWM6QWRkaX
Rpb25hbERvY3VtZW50UmVmZXJlbmNlPjxjYWM6U2lnbmF0dXJlPgogICAgICA8Y2JjOklEPnVybjpvYXNpczpuYW1lczpz
GVjaWZpY2F0aW9uOnVibDpzaWduYXR1cmU6SW52b2ljZTwvY2JjOklEPgogICAgICA8Y2JjOlNpZ25hdHVyZU1ldGhvZD5
1cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6ZHNpZzplbnZlbG9wZWQ6eGFkZXM8L2NiYzpTaWduYXR1cm
VNZXRob2Q+CjwvY2FjOlNpZ25hdHVyZT48Y2FjOkFjY291bnRpbmdTdXBwbGllclBhcnR5PgogICAgICAgIDxjYWM6UGFyd
Hk+CiAgICAgICAgICAgIDxjYWM6UGFydHlJZGVudGlmaWNhdGlvbj4KICAgICAgICAgICAgICAgIDxjYmM6SUQgc2NoZW1
lSUQ9IkNSTiI+MTAxMDAxMDAwMDwvY2JjOklEPgogICAgICAgICAgICA8L2NhYzpQYXJ0eUlkZW50aWZpY2F0aW9uPgo
gICAgICAgICAgICA8Y2FjOlBvc3RhbEFkZHJlc3M+CiAgICAgICAgICAgICA8Y2JjOlN0cmVldE5hbWU+2KfZhNin2YXZit
ixINiz2YTYt9in2YYgfCBQcmluY2UgU3VsdGFuPC9jYmM6U3RyZWV0TmFtZT4KICAgICAgICAgICAgIDxjYmM6QnVp
bGRpbmdOdW1iZXI+MjMyMjwvY2JjOkJ1aWxkaW5nTnVtYmVyPgogICAgICAgICAgICAgPGNiYzpDaXR5U3ViZGl2a
XNpb25OYW1lPtin2YTZhdix2KjYuSB8IEFsLU11cmFiYmE8L2NiYzpDaXR5U3ViZGl2aXNpb25OYW1lPgogICAgICAgI
CAgICAgPGNiYzpDaXR5TmFtZT7Yp9mE2LHZitin2LYgfCBSaXlhZGg8L2NiYzpDaXR5TmFtZT4KICAgICAgICAgICAg
IDxjYmM6UG9zdGFsWm9uZT4yMzMzMzwvY2JjOlBvc3RhbFpvbmU+CiAgICAgICAgICAgICA8Y2FjOkNvdW50cnk+C
iAgICAgICAgICAgICAgICAgPGNiYzpJZGVudGlmaWNhdGlvbkNvZGU+U0E8L2NiYzpJZGVudGlmaWNhdGlvbkNvZG
U+CiAgICAgICAgICAgICA8L2NhYzpDb3VudHJ5PgogICAgICAgICAgICA8L2NhYzpQb3N0YWxBZGRyZXNzPgogICA
gICAgICA8Y2FjOlBhcnR5VGF4U2NoZW1lPgogICAgICAgICAgICAgICAgPGNiYzpDb21wYW55SUQ+Mzk5OTk5OTk5
OTAwMDAzPC9jYmM6Q29tcGFueUlEPgogICAgICAgICAgICAgPGNhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgICAgI
CAgICAgPGNiYzpJRD5WQVQ8L2NiYzpJRD4KICAgICAgICAgIDwvY2FjOlRheFNjaGVtZT4KICAgICAgICAgICAg
PC9jYWM6UGFydHlUYXhTY2hlbWU+CiAgICAgICAgIDxjYWM6UGFydHlMZWdhbEVudGl0eT4KICAgICAgICAgICAgI
CAgIDxjYmM6UmVnaXN0cmF0aW9uTmFtZT7YtNix2YPYqSDYqtmI2LHZitivINin2YTYqtmD2YbZiNmE2YjYrNmK2Kcg2KjY

o9mC2LXZiSDYs9ix2LnYqSDYp9mE2YXYrdiv2YjYr9ipIHwgTWF4aW11bSBTcGVlZCBUZWNoIFN1cHBseSBMVEQ8L2Ni
YzpSZWdpc3RyYXRpb25OYW1lPgogICAgICAgICAgICA8L2NhYzpQYXJ0eUxlZ2FsRW50aXR5PgogICAgICAgIDwvY2FjO
lBhcnR5PgogICAgPC9jYWM6QWNjb3VudGluZ1N1cHBsaWVyUGFydHk+CiAgICAgPGNhYzpBY2NvdW50aW5nQ3VzdG9
tZXJQYXJ0eT4KICAgICAgICA8Y2FjOlBhcnR5PgogICAgICAgICAgICA8Y2FjOlBvc3RhbEFkZHJlc3M+CiAgICAgICAgICAgI
CAgICA8Y2JjOlN0cmVldE5hbWU+2LXZhNin2K0g2KfZhNiv2YrZhiB8IFNhbGFoIEFsLURpbjwvY2JjOlN0cmVldE5hbWU+Ci
AgICAgICAgICAgICAgICA8Y2JjOkJ1aWxkaW5nTnVtYmVyPjExMTE8L2NiYzpCdWlsZGluZ051bWJlcj4KICAgICAgICAgIC
AgICAgIDxjYmM6Q2l0eVN1YmRpdmlzaW9uTmFtZT7Yp9mE2YXYsdmI2KwgfCBBbC1NdXJvb2o8L2NiYzpDaXR5U3ViZGl
2aXNpb25OYW1lPgogICAgICAgICAgICAgICAgPGNiYzpDaXR5TmFtZT7Yp9mE2LHZitin2LYgfCBSaXlhZGg8L2NiYzpDaX
R5TmFtZT4KICAgICAgICAgICAgICAgIDxjYmM6UG9zdGFsWm9uZT4xMjIyMjwvY2JjOlBvc3RhbFpvbmU+CiAgICAgICAgI
CAgICAgICA8Y2FjOkNvdW50cnk+CiAgICAgICAgICAgICAgICAgICAgPGNiYzpJZGVudGlmaWNhdGlvbkNvZGU+U0E8L2
NiYzpJZGVudGlmaWNhdGlvbkNvZGU+CiAgICAgICAgICAgICAgICA8L2NhYzpDb3VudHJ5PgogICAgICAgICAgICA8L2Nh
YzpQb3N0YWxBZGRyZXNzPgogICAgICAgICAgICA8Y2FjOlBhcnR5VGF4U2NoZW1lPgogICAgICAgICAgICAgICAgPGNiY
zpDb21wYW55SUQ+Mzk5OTk5OTk5ODAwMDAzPC9jYmM6Q29tcGFueUlEPgogICAgICAgICAgICAgICAgPGNhYzpUYXh
TY2hlbWU+CiAgICAgICAgICAgICAgICAgICAgPGNiYzpJRD5WQVQ8L2NiYzpJRD4KICAgICAgICAgICAgICAgIDwvY2FjOl
RheFNjaGVtZT4KICAgICAgICAgICAgPC9jYWM6UGFydHlUYXhTY2hlbWU+CiAgICAgICAgICAgIDxjYWM6UGFydHlMZW
dhbEVudGl0eT4KICAgICAgICAgICAgICAgIDxjYmM6UmVnaXN0cmF0aW9uTmFtZT7YtNix2YPYqSDZhtmF2KfYsNisINmB
2KfYqtmI2LHYqSDYp9mE2YXYrdiv2YjYr9ipIHwgRmF0b29yYSBTYW1wbGVzIExURDwvY2JjOlJlZ2lzdHJhdGlvbk5hbWU+
CiAgICAgICAgICAgIDwvY2FjOlBhcnR5TGVnYWxFbnRpdHk+CiAgICAgICAgPC9jYWM6UGFydHk+CiAgICA8L2NhYzpBY
2NvdW50aW5nQ3VzdG9tZXJQYXJ0eT4KICAgIDxjYWM6RGVsaXZlcnk+CiAgICAgICAgPGNiYzpBY3R1YWxEZWxpdmVy
eURhdGU+MjAyMi0wOS0wNzwvY2JjOkFjdHVhbERlbGl2ZXJ5RGF0ZT4KICAgIDwvY2FjOkRlbGl2ZXJ5PgogICAgPGNhYz
pQYXltZW50TWVhbnM+CiAgICAgICAgPGNiYzpQYXltZW50TWVhbnNDb2RlPjEwPC9jYmM6UGF5bWVudE1lYW5zQ29kZ
T4KICAgIDwvY2FjOlBheW1lbnRNZWFucz4KICAgIDxjYWM6QWxsb3dhbmNlQ2hhcmdlPgogICAgICAgIDxjYmM6Q2hhcmd
lSW5kaWNhdG9yPmZhbHNlPC9jYmM6Q2hhcmdlSW5kaWNhdG9yPgogICAgICAgIDxjYmM6QWxsb3dhbmNlQ2hhcmdlU
mVhc29uPmRpc2NvdW50PC9jYmM6QWxsb3dhbmNlQ2hhcmdlUmVhc29uPgogICAgICAgIDxjYmM6QW1vdW50IGN1cnJl
bmN5SUQ9IlNBUiI+MC4wMDwvY2JjOkFtb3VudD4KICAgICAgICA8Y2FjOlRheENhdGVnb3J5PgogICAgICAgICAgICA8Y2J
jOklEIHNjaGVtZUlEPSJVTi9FQ0UgNTMwNSIgc2NoZW1lQWdlbmN5SUQ9IjYiPlM8L2NiYzpJRD4KICAgICAgICAgICAgPG
NiYzpQZXJjZW50PjE1PC9jYmM6UGVyY2VudD4KICAgICAgICAgICAgPGNhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgICA
gICA8Y2JjOklEIHNjaGVtZUlEPSJVTi9FQ0UgNTE1MyIgc2NoZW1lQWdlbmN5SUQ9IjYiPlZBVDwvY2JjOklEPgogICAgICAg
ICAgICA8L2NhYzpUYXhTY2hlbWU+CiAgICAgICAgPC9jYWM6VGF4Q2F0ZWdvcnk+CiAgICA8L2NhYzpBbGxvd2FuY2VD
aGFyZ2U+CiAgICA8Y2FjOlRheFRvdGFsPgogICAgICAgIDxjYmM6VGF4QW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+MC42P
C9jYmM6VGF4QW1vdW50PgogICAgPC9jYWM6VGF4VG90YWw+CiAgICA8Y2FjOlRheFRvdGFsPgogICAgICAgIDxjYmM
6VGF4QW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+MC42PC9jYmM6VGF4QW1vdW50PgogICAgICAgIDxjYWM6VGF4U3Vid
G90YWw+CiAgICAgICAgICAgIDxjYmM6VGF4YWJsZUFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjQuMDA8L2NiYzpUYXhhYm
xlQW1vdW50PgogICAgICAgICAgICA8Y2JjOlRheEFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjAuNjA8L2NiYzpUYXhBbW91bn
Q+CiAgICAgICAgICAgIDxjYWM6VGF4Q2F0ZWdvcnk+CiAgICAgICAgICAgICAgICA8Y2JjOklEIHNjaGVtZUlEPSJVTi9FQ0UgNTMwNSIgc2NoZW1lQWdlbmN5SUQ9IjYiPlM8L2NiYzpJRD4K

303

HTTP See Other. Returned when the submitted invoice is a Standard Invoice while clearance is deactivated. InfoModel

**Example data**

Content-Type: application/json

```
{"message":"Clearance is deactiviated. Please use the /invoices/reporting/single endpoint instead."}
```

**400**
HTTP Bad Request. Returned when the submitted request is invalid. InvoiceResultModel
**Example data**
Content-Type: KSA Rules Violation

{"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-14","category":"KSA","message":"The buyer identification (BT-46), required only if buyer is not VAT registered, then the buyer identification (BT-46) must be provided with one of the scheme IDs (BT-46-1) (TIN, CRN, MOM, MLS, 700, SAG, NAT, GCC, IQA, OTH) and must contain only alphanumeric characters. Tax Identification Number 'TIN' as schemeID. Commercial registration number with 'CRN' as schemeID. MOMRAH license with 'MOM' as schemeID. MHRSD license with 'MLS' as schemeID. 700 Number with '700' as schemeID. MISA license with 'SAG' as schemeID. National ID with 'NAT' as schemeID. GCC ID with 'GCC' as schemeID. Iqama Number with 'IQA' as schemeID. Passport ID with 'PAS' as schemeID. Other ID with 'OTH' as schemeID.In case of multiple commercial registrations, the seller should fill the commercial registration of the branch in respect of which the Tax Invoice is being issued. In case multiple IDs exist then one of the above must be entered following the sequence specified above.","status":"ERROR"}],"status":"ERROR"},"clearanceStatus":"NOT_CLEARED","clearedInvoice":null}

**Example data**
Content-Type: EN Rules Violation

{"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-F-06-C17","category":"EN_16931","message":"[BR-KSA-F-06-C17] - Field character limits for Invoice line identifier field (BT-126) have not been met. The minimum limit is 1 character and the maximum limit is 6 characters.","status":"ERROR"}],"status":"ERROR"},"clearanceStatus":"NOT_CLEARED","clearedInvoice":null}

**Example data**
Content-Type: XML Schema Error

{"validationResults":{"infoMessages":[],"warningMessages":[],"errorMessages":[{"type":"ERROR","code":"XSD_ZATCA_INVALID","category":"XSD validation","message":"Schema validation failed; XML does not comply with UBL 2.1 standards in line with ZATCA specifications. ERROR: org.xml.sax.SAXParseException; lineNumber: 203; columnNumber: 20; cvc-complex-type.2.4.b: The content of element 'cac:TaxTotal' is not complete. One of '{\"urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2\":TaxAmount}' is expected.","status":"ERROR"}],"status":"ERROR"},"clearanceStatus":"NOT_CLEARED","clearedInvoice":null}

**Example data**
Content-Type: InvalidInvoiceHashInvalidBase64

{"validationResults":{"infoMessages":[],"warningMessages":[],"errorMessages":[{"type":"ERROR","code":"InvoiceHash-Errors","category":"Invalid-InvoiceHash","message":"Document hash format in the API body is not valid","status":"ERROR"}],"status":"ERROR"},"clearanceStatus":"NOT_CLEARED","clearedInvoice":null}

**Example data**
Content-Type: InvalidInvoiceHash

{"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":"invalid-invoice-hash","category":"INVOICE_HASHING_ERRORS","message":"The invoice hash API body does not match the (calculated) Hash of the XML","status":"ERROR"}],"status":"ERROR"},"clearanceStatus":"NOT_CLEARED","clearedInvoice":null}

**Example data**
Content-Type: Missing Invoice

{"validationResults":{"infoMessages":[],"warningMessages":[],"errorMessages":[{"type":"ERROR","code":"XSD_ZATCA_INVALID","category":"XSD validation","message":"Schema validation failed; XML does not comply with UBL 2.1 standards in line with ZATCA specifications","status":"ERROR"}],"status":"ERROR"},"clearanceStatus":"NOT_CLEARED","clearedInvoice":null}

**Example data**
Content-Type: Missing Invoice Hash

{"validationResults":{"infoMessages":[],"warningMessages":[],"errorMessages":[{"type":"ERROR","code":"InvoiceHash-Errors","category":"Invalid-InvoiceHash","message":"Document hash is not present in the API body","status":"ERROR"}],"status":"ERROR"},"clearanceStatus":"NOT_CLEARED","clearedInvoice":null}

**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":""}
```

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

```
{"category":"HTTP-Errors","code":"500","message":"Something went wrong and caused an Internal Server Error."}
```

## 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 status, the cleared document, warnings (if any), and errors (if any).

**validationResults (optional)**
*array[validationResultsModel]*

**clearedInvoice (optional)**
*String*

**clearanceStatus (optional)**
*String*
**Enum:**
*CLEARED*
*NOT_CLEARED*

### 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*

### validationResultsModel - validationResultsModel

Up

An object representing the structure of the validation results returned by the API endpoints. Specifically, it includes the invoice hash, status, and lists of info, warning, and error messages.

**infoMessages (optional)**
*array[InfoModel]*

**warningMessages (optional)**
*array[WarningModel]*

**erroMessages (optional)**
*array[ErrorModel]*

**status (optional)**
*String*

**Enum:**
*PASS*
*WARNING*
*ERROR*

# 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.

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

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

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

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

**ReportingModelEndpointS**

# ReportingModelEndpointS

## POST /invoices/reporting/single

Up

Reports a single invoice. (**reportSingleInvoice**)

Reports a single SIMPLIFIED invoice, credit note, or debit note. Specifically, it accepts 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 set.
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. Cryptographical Stamp validation

6. Previous Invoice Hash Validation (PIH)

### Consumes

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

- application/json

### Request body

**body [InvoiceRequest](#) (required)**

*Body Parameter —*

example: { "summary" : "Simplified Invoice", "value" : { "invoiceHash" : "vLGQoYNoM3tf1XAxKpoNTSz/8p
kdidXy47HWh0VQmu8=", "uuid" : "8e6000cf-1a98-4174-b3e7-b5d5954bc10d", "invoice" : "PD94bWwgdm
Vyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPEludm9pY2UgeG1sbnM9InVybjpvYXNpczpuYW1lczp
zcGVjaWZpY2F0aW9uOnVibDpzY2hlbWE6eHNkOkludm9pY2UtMiIgeG1sbnM6Y2FjPSJ1cm46b2FzaXM6
bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2NoZW1hOnhzZDpDb21tb25BZ2dyZWdhdGVDb21wb25lbnRz
LTIiIHhtbG5zOmNiYz0idXJuOm9hc2lzOm5hbWVzOnNwZWNpZmljYXRpb246dWJsOnNjaGVtYTp4c2Q6
Q29tbW9uQmFzaWNDb21wb25lbnRzLTIiIHhtbG5zOmV4dD0idXJuOm9hc2lzOm5hbWVzOnNwZWNpZml
jYXRpb246dWJsOnNjaGVtYTp4c2Q6Q29tbW9uRXh0ZW5zaW9uQ29tcG9uZW50cy0yIj48ZXh0OlVCTEV
4dGVuc2lvbnM+CiAgICA8ZXh0OlVCTEV4dGVuc2lvbj4KICAgICAgICA8ZXh0OkV4dGVuc2lvblVSST51cm
46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6ZHNpZzpbnZlbG9wZWQ6eGFkZXM8L2V4dDpFe
HRlbnNpb25VUkk+CiAgICAgICAgPGV4dDpFeHRlbnNpb25Db250ZW50PgogICAgICAgICAgICA8c2lnOlV
CTERvY3VtZW50U2lnbmF0dXJlcyB4bWxuczpzaWc9InVybjpvYXNpczpuYW1lczpzcGVjaWZpY2F0aW9u
OnVibDpzY2hlbWE6eHNkOkNvbW1vblNpZ25hdHVyZUNvbXBvbmVudHMtMiIgeG1sbnM6c2FjPSJ1cm46
b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2NoZW1hOnhzZDpTaWduYXR1cmVBZ2dyZWdhd
GVDb21wb25lbnRzLTIiIHhtbG5zOnNiYz0idXJuOm9hc2lzOm5hbWVzOnNwZWNpZmljYXRpb246dWJsO
nNjaGVtYTp4c2Q6U2lnbmF0dXJlQmFzaWNDb21wb25lbnRzLTIiPgogICAgICAgICAgICAgICAgPHNhYzp
TaWduYXR1cmVJbmZvcm1hdGlvbj4gCiAgICAgICAgICAgICAgICAgICAgPGNiYzpJRD51cm46b2FzaXM6
bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2lnbmF0dXJlOjE8L2NiYzpJRD4KICAgICAgICAgICAgICAgICA
gICA8c2JjOlJlZmVyZW5jZWRTaWduYXR1cmVJRD51cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp
1Ymw6c2lnbmF0dXJlOkludm9pY2U8L3NiYzpSZWZlcmVuY2VkU2lnbmF0dXJlSUQ+CiAgICAgICAgICAgI
CAgICAgICAgPGRzOlNpZ25hdHVyZSB4bWxuczpkcz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94
bWxkc2lnIyIgSWQ9InNpZ25hdHVyZSI+CiAgICAgICAgICAgICAgICAgICAgICAgIDxkczpTaWduZWRJbm
ZvPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOkNhbm9uaWNhbGl6YXRpb25NZXRob2QgQ
Wxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDA2LzEyL3htbC1jMTRuMTEiLz4KICAgICAgICAgICA
gICAgICAgICAgICAgIDxkczpTaWduYXR1cmVNZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3Lnc
zLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNlY2RzYS1zaGEyNTYiLz4KICAgICAgICAgICAgICAgICAg
ICAgICAgIDxkczpSZWZlcmVuY2UgSWQ9Imludm9pY2VTaWduZWREYXRhIiBVUkk9IiI+CiAgICAgI
CAgICAgICAgICAgICAgICAgICAgPGRzOlRyYW5zZm9ybXM+CiAgICAgICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgIDxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy9
UUi8xOTk5L1JFQy14cGF0aC0xOTk5MTExNiI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgICA8ZHM6WFBhdGg+bm90KC8vYW5jZXN0b3Itb3Itc2VsZjo6ZXh0OlVCTEV4dGVuc2lvbnMpPC9
kczpYUGF0aD4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPC9kczpUcmFuc2Zvcm0+
CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPS
JodHRwOi8vd3d3LnczLm9yZy9UUi8xOTk5L1JFQy14cGF0aC0xOTk5MTExNiI+CiAgICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICA8ZHM6WFBhdGg+bm90KC8vYW5jZXN0b3Itb3Itc2VsZjo6Y2F
jOlNpZ25hdHVyZSk8L2RzOlhQYXRoPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8L2
RzOlRyYW5zZm9ybT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOlRyYW5zZm
9ybSBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnL1RSLzE5OTkvUkVDLXhwYXRoLTE5OTkxMTE2Ij
4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpYUGF0aD5ub3QoLy9hbmNlc
3Rvci1vci1zZWxmOjpjYWM6QWRkaXRpb25hbERvY3VtZW50UmVmZXJlbmNlW2NiYzpJRD0nUVInXSk
8L2RzOlhQYXRoPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8L2RzOlRyYW5zZm9y
bT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOlRyYW5zZm9ybSBBbGdvcml0a
G09Imh0dHA6Ly93d3cudzMub3JnLzIwMDYvMTIveG1sLWMxNG4xMSIvPgogICAgICAgICAgICAgIC
AgICAgICAgICAgIDwvZHM6VHJhbnNmb3Jtcz4KICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CA8ZHM6RGlnZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bW
xlbmMjc2hhMjU2Ii8+CiAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOkRpZ2VzdFZhbHVlPn
ZMR1FvWU5vTTN0ZjFYQXhLcG9OVFN6Lzhwa2RpZFh5NDdIV2gwVlFtdTg9PC9kczpEaWdlc3RWYWx
1ZT4KICAgICAgICAgICAgICAgICAgICAgIDwvZHM6UmVmZXJlbmNlPgogICAgICAgICAgI
CAgICAgICAgICAgPGRzOlJlZmVyZW5jZSBUeXBlPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L
3htbGRzaWcjU2lnbmF0dXJlUHJvcGVydGllcyIgVVJJPSIjeGFkZXNTaWduZWRQcm9wZXJ0aWVzIj4KIC
AgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6RGlnZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR
0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+CiAgICAgICAgICAgICAgIC
AgICAgICAgPGRzOkRpZ2VzdFZhbHVlPllYRTFaRFV3TXpBd1pEVm1PREpsTUdGaFpETTRNamxrT
0RJd1kySmtNV0psTWpneE16ZGhZbVZqT1RFNE5qQmlOV0ppT0RRM09XRTVZekJrWVRjd1pBPT08L2
RzOkRpZ2VzdFZhbHVlPgogICAgICAgICAgICAgICAgICAgICAgPC9kczpSZWZlcmVuY2U+CiA
gICAgICAgICAgICAgICAgIDwvZHM6U2lnbmVkSW5mbz4KICAgICAgICAgICAgICAgICAgI
CAgPGRzOlNpZ25hdHVyZVZhbHVlPk1FUUNJRmlSZjliYW5mZVpYRWY4Q1k0TVRmMW1SY1FsbXpB
MVpodDBBazVHdXkxbUFpQm9MWGlkNjFLQUVOb0dod3YyZmh5S3IzNTBNMG1XWGlOUk5MQUc5ZW
FFVUE9PTwvZHM6U2lnbmF0dXJlVmFsdWU+CiAgICAgICAgICAgICAgICAgICAgIDxkczpLZXlJbmZ
vPgogICAgICAgICAgICAgICAgICAgICAgPGRzOlg1MDlEYXRhPgogICAgICAgICAgICAgICAgIC
AgICAgICAgIDxkczpYNTA5Q2VydGlmaWNhdGU+TUlJRDNqQ0NBNFNnQXdJQkFnSVRFUU
FBT0FQRjkwQWpzL3hjWHdBQkFBQTRBekFLQmdncWhrak9QUVFEQWpCaU1SVXdFd1lLQ1pJbWlaU

HlMR1FCR1JZRmJHOWpZV3d4RXpBUkJnb0praWFKay9Jc1pBRVpGZ05uYjNZeEZ6QVZCZ29Ka2lhSm
svSXNaQUVaRmdkbGVlUm5ZWHAwTVJzd0dRWURWUVFERXhKUVVscEZTVTVXVDBsRFJWTkRRV
FF0UTBFd0hoY05NalF3TVRFeE1Ea3hPVE13V2hjTk1qa3dNVEE1TURreE9UTXdXakIxTVFzd0NRWUR
WUVFHRXdKVFFURW1NQ1FHQTFVRUNoTWRUV0Y0YVcxMWJTQlRjR1ZsWkNCVVpXTm9JRk4xY0
hCc2VTQk1WRVF4RmpBVUJnTlZCQXNURFZKcGVXRmthQ0JDY21GdVkyZ3hKakFrQmdOVkJBTVRI
VlJUVkMwNE9EWTBNekV4TkRVdE16azVPVGs1T1RrNU9UQXdNREF6TUZZd0VBWUhLb1pJemowQ0
FRWUZLNEVFQUFvRFFnQUVvV0NLYTBTYTlGSUVyVE92MHVBa0MxVklLWHhVOW5QcHgydmxmNH
loTWVqeThjMDJYSmJsRHE3dFB5ZG84bXEwYWhPTW1Obzhnd25pN1h0MUtUOVVlS09DQWdjd2dnSU
RNSUd0QmdOVkhSRUVnYVV3Z2FLa2daOHdnWnd4T3pBNUJnTlZCQVFFNTWpFdFZGTlVmREl0VkZOV
WZETXRaV1F5TW1ZeFpEZ3RaVFpoTWkweE1URTRMVGxpTlRndFpEbGhPR1l4TVdVME5EVm1NUjh3
SFFZS0NaSW1pWlB5TEdRQkFRd1BNems1T1RrNU9UazVPVEF3TURBek1RMHdDd1lEVlFRTURBUXh
NVEF3TVJFd0R3WURWUVFhREFoU1VsSkVNamt5T1RFYU1CZ0dBMVVFRHd3UlUzVndjR3g1SUdGa
mRHbDJhWFJwWlhNd0hRWURWUjBPQkJZRUZCWCtZdm1tdG5Zb0RmOUJHYktvN29jVEtZSzFNQjhH
QTFVZEl3UVlNQmFBRkp2S3FxTHRtcXdza0lGelZ2cFAyUHhUKzlObk1Ic0dDQ3NHQVFVRkJ3RUJCRz
h3YlRCckJnZ3JCZ0VGQlFjd0FvWmZhSFIwY0RvdkwzQnZaM0Z0Y1RiUzdmI5a0ZkyRXVaMjkyTG5OaEwwTm
xjblJGYm5KdmJHd3ZVRkphUlVsT1ZsUlBNUlZERUNBMExVTkJMVjRLVkR3TUJxWXh2WTJGc0 
xQlNXa1ZKVGxaUFNVTkZVME5CTkMxRFFTZ3hLUzVqY25Rd0RnWURWUjBQQVFIL0JBUURBZ2VBT
UR3R0NTc0dBUVFCZ2pjVkJ3UXZNQzBHSlNzR0FRUUJnamNWQ0lHR3FCMkUwUHNaSUQ5ZEpJZk8r
eG5Ud0ZWbWgvcWxaWVhaaEQ0Q0FXUUNBUkl3SFFZRFZSMGxCQlI3RkFZSUt3WUJCUVVIQXdNR0
NDc0dBUVVGQndNQ01DY0dDU3NHQVFRQmdqY1ZDZ1FhTUJnd0NnWUlLd1lCQlFVSEF3TXdDZ1lJS
3dZQkJRVUhBd0l3Q2dZSUtvWkl6ajBFQXdJRFNBQXdSUUloQUxFL2ljaG1uV1hhDVUtVYmNhM2ljaThvc
XdhTHZGZEhWalFydmVJOXVxQWJBaUE5aEM0TThqZ01CQURQQU3ptZDJ1aVBKQTZnS1IzTEUwM1U
3NWVxYkMvclhBPT08L2RzOlg1MDlDZXJ0aWZpY2F0ZT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
IDwvZHM6WDUwOURhdGE+CiAgICAgICAgICAgICAgICAgICAgICAgIDwvZHM6S2V5SW5mbz4KICAgI
CAgICAgICAgICAgICAgICAgICAgPGRzOk9iamVjdD4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDx4
YWRlczpRdWFsaWZ5aW5nUHJvcGVydGllcyB4bWxuczp4YWRlcz0iaHR0cDovL3VyaS5ldHNpLm9yZy8
wMTkwMy92MS4zLjIjIiBUYXJnZXQ9InNpZ25hdHVyZSI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICA
gICAgPHhhZGVzOlNpZ25lZFByb3BlcnRpZXMgSWQ9InhhZGVzU2lnbmVkUHJvcGVydGllcyI+CiAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgIDx4YWRlczpTaWduZWRTaWduYXR1cmVQcm9wZXJ
0aWVzPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOlNpZ25pbmdU
aW1lPjIwMjUtMDYtMDJUMTU6NDQ6MDE8L3hhZGVzOlNpZ25pbmdUaW1lPgogICAgICAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOlNpZ25pbmdDZXJ0aWZpY2F0ZT4KICAgICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8eGFkZXM6Q2VydD4KICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOkNlcnREaWdlc3Q+CiAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6RGlnZXN0TWV0aG9kIE
FsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+CiAgICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6RGlnZXN0VmFsdWU+W
kRNd01tSTBNVEUxTnpWak9UVTJOVGs0WXpWbE9EaGhZbUkwT0RVMk5EVXlOVFUyWVRWaFlqaGh
NREZtTjJGallqazFZVEEyT1dRME5qWTJNalE0TlE9PTwvZHM6RGlnZXN0VmFsdWU+CiAgICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgIDwveGFkZXM6Q2VydERpZ2VzdD4KICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOklzc3VlclNlcmlhb
D4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpYNTA5
SXNzdWVyTmFtZT5DTj1QUlpFSU5WT0lDRVNDQTQtQ0EsIERDPWV4dGdhenQsIERDPWdvdiwgREM
9bG9jYWw8L2RzOlg1MDlJc3N1ZXJOYW1lPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgPGRzOlg1MDlTZXJpYWxOdW1iZXI+Mzc5MTEyNzQyODMxMzgwNDcxO
DM1MjYzOTY5NTg3Mjg3NjYzNTIwNTI4Mzg3PC9kczpYNTA5U2VyaWFsTnVtYmVyPgogICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICA8L3hhZGVzOklzc3VlclNlcmlhbD4KICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICA8L3hhZGVzOkNlcnQ+CiAgICAgICAgI
CAgICAgICAgICAgICAgICAgICA8L3hhZGVzOlNpZ25pbmdDZXJ0aWZpY2F0ZT4KIC
AgICAgICAgICAgICAgICAgICAgICAgPC94YWRlczpTaWduZWRTaWduYXR1cmVQc
m9wZXJ0aWVzPgogICAgICAgICAgICAgICAgICAgICAgIDwveGFkZXM6U2lnbmVkUHJvc
GVydGllcz4KICAgICAgICAgICAgICAgICAgICAgIDwveGFkZXM6UXVhbGlmeWluZ1Byb3BlcnR
pZXM+CiAgICAgICAgICAgICAgICAgIDwvZHM6T2JqZWN0PgogICAgICAgICAgICAgICAgI
DwvZHM6U2lnbmF0dXJlPgogICAgICAgICAgICAgPC9zYWM6U2lnbmF0dXJlSW5mb3JtYXRpb24+Ci
AgICAgICAgIDwvc2lnOlVCTERvY3VtZW50U2lnbmF0dXJlcz4KICAgICAgICA8L2V4dDpFeHRlbnNpb
25Db250ZW50PgogICAgPC9leHQ6VUJMRXh0ZW5zaW9uPgo8L2V4dDpVQkxFeHRlbnNpb25zPgoKICA
gIDxjYmM6UHJvZmlsZUlEPnJlcG9ydGluZzoxLjA8L2NiYzpQcm9maWxlSUQ+CiAgICA8Y2JjOklEPlNNR
TAwMDEwPC9jYmM6SUQ+CiAgICA8Y2JjOlVVSUQ+OGU2MDAwY2YtMWE5OC00MTc0LWIzZTctYjVk
NTk1NGJjMTBkPC9jYmM6VVVJRD4KICAgIDxjYmM6SXNzdWVEYXRlPjIwMjUtMDYtMDI8L2NiYzpJc3N
1ZURhdGU+CiAgICA8Y2JjOklzc3VlVGltZT4xNzo0MTowODwvY2JjOklzc3VlVGltZT4KICAgIDxjYmM6SW
52b2ljZVR5cGVDb2RlIG5hbWU9IjAyMDAwMDAiPjM4ODwvY2JjOkludm9pY2VUeXBlQ29kZT4KICAgIDxj
YmM6Tm90ZSBsYW5ndWFnZUlEPSJhciI+QUJDPC9jYmM6Tm90ZT4KICAgIDxjYmM6RG9jdW1lbnRDd
XJyZW5jeUNvZGU+U0FSPC9jYmM6RG9jdW1lbnRDdXJyZW5jeUNvZGU+CiAgICA8Y2JjOlRheEN1cnJl
bmN5Q29kZT5TQVI8L2NiYzpUYXhDdXJyZW5jeUNvZGU+CiAgICA8Y2FjOkFkZGl0aW9uYWxEb2N1bW
VudFJlZmVyZW5jZT4KICAgICAgICA8Y2JjOklEPklDVjwvY2JjOklEPgogICAgICAgIDxjYmM6VVVJRD4xM
DwvY2JjOlVVSUQ+CiAgICA8L2NhYzpBZGRpdGlvbmFsRG9jdW1lbnRSZWZlcmVuY2U+CiAgICA8Y2Fj
OkFkZGl0aW9uYWxEb2N1bWVudFJlZmVyZW5jZT4KICAgICAgICA8Y2JjOklEPlBJSDwvY2JjOklEPgogI
CAgICAgIDxjYWM6QXR0YWNobWVudD4KICAgICAgICAgICAgPGNiYzpFbWJlZGRlZERvY3VtZW50Qml
uYXJ5T2JqZWN0IG1pbWVDb2RlPSJ0ZXh0L3BsYWluIj5OV1psWTJWaU5qWm1abU00Tm1Zek9HUTVO
VEkzT0Raak5tUTJPVFpqTnpsak1tUmlZekl6T1dSa05HVTVNV0kwTmpjeU9XUTNNMkV5TjJaaU5UZGxP
UT09PC9jYmM6RW1iZWRkZWREb2N1bWVudEJpbmFyeU9iamVjdD4KICAgICAgICA8L2NhYzpBdHRhY
2htZW50PgogICAgPC9jYWM6QWRkaXRpb25hbERvY3VtZW50UmVmZXJlbmNlPgogICAgCiAgICAKICAg

IDxjYWM6QWRkaXRpb25hbERvY3VtZW50UmVmZXJlbmNlPgogICAgICAgIDxjYmM6SUQ+UVI8L2NiYzp
JRD4KICAgICAgICA8Y2FjOkF0dGFjaG1lbnQ+CiAgICAgICAgICAgIDxjYmM6RW1iZWRkZWREb2N1bW
VudEJpbmFyeU9iamVjdCBtaW1lQ29kZT0idGV4dC9wbGFpbiI+QVcvWXROaXgyWVBZcVNEWXF0bUky
TEhaaXRpdklOaW4yWVRZcXRtRDJZYlppTm1FMllqWXJObUsyS2NnMktqWW85bUMyTFhaaVNEWXM5
aXgyTG5ZcVNEWXA5bUUyWVhZcmRpdjJZallyOWlwSUh3Z1RXRjRhVzExYlNCVGNHVmxaQ0JVWldO
b0lGTjFjSEJzZVNCTVZFUUNEek01T1RrNU9UazVPVGt3TURBd013TVRNakF5TlMwd05pMHdNbFF4Tn
pvME1Ub3dPQVFHTWpNeExqRTFCUVV6TUM0eE5RWXNka3hIVVc5WlRtOU5NM1JtTVZoQmVFdHdiM
DVVVTNvdk9IQnJaR2xrV0hrME4waFhhREJXVVcxMU9EMEhZRTFGVVVOSlJtbFNaamxpWVc1bVpWc
FlSV1k0UTFrMFRWUWJmNVVzWnFTWTFGc2JYcEJNVnBvZERCQmF6VkhkWGt4YlVGcFFtOU1XUkFhZ
TFFVVk9iMGRvZDNZeVptaDVTM0l6TlRCTk1HMVhXR2xPVWs1TVFVYzVaV0ZZZGVlVFFOVBRaFlNRll3R
UFZSEtvWkl6ajBDQVFZRks0RUVBQW9EUWdBRW9XQ0thMFNhOUZJRXJJUT3YwdUFrQzFWSUtYeF
U5blBweDJ2bGY0eWhNZWp5OGMwMlhKYmxEcTd0UHlkbzhtcTBhaE9NbU5vOGd3bmk3WHQxS1Q5V
WVBbEhNRVVDSVFDeFA0bklacDFsd2xDbEczR3Q4bkl2S0tzR2k3eFhSMVkwSzczaVBicWdHd0lnUFlR
dURQSTREQVFBejBzNW5kcm9qeVFPb0NrZHl4Tk4xTytYcW13djYxdz08L2NiYzpFbWJlZGRlZERvY3Vt
ZW50QmluYXJ5T2JqZWN0PgogICAgICAgIDwvY2FjOkF0dGFjaG1lbnQ+CjwvY2FjOkFkZGl0aW9uYWxE
b2N1bWVudFJlZmVyZW5jZT48Y2FjOlNpZ25hdHVyZT4KICAgICAgPGNiYzpJRD51cm46b2FzaXM6bmFt
ZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2lnbmF0dXJlOkludm9pY2U8L2NiYzpJRD4KICAgICAgPGNiYzpTa
WduYXR1cmVNZXRob2Q+dXJuOm9hc2lzOm5hbWVzOnNwZWNpZmljYXRpb246dWJsOmRzaWc6ZW5
2ZWxvcGVkOnhhZGVzPC9jYmM6U2lnbmF0dXJlTWV0aG9kPgo8L2NhYzpTaWduYXR1cmU+PGNhYzp
BY2NvdW50aW5nU3VwcGxpZXJQYXJ0eT4KICAgICAgICA8Y2FjOlBhcnR5PgogICAgICAgICAgICA8Y2F
jOlBhcnR5SWRlbnRpZmljYXRpb24+CiAgICAgICAgICAgICAgICA8Y2JjOklEIHNjaGVtZUlEPSJDUk4iPjE
wMTAwMTAwMDA8L2NiYzpJRD4KICAgICAgICAgICAgPC9jYWM6UGFydHlJZGVudGlmaWNhdGlvbj4KI
CAgICAgICAgICAgPGNhYzpQb3N0YWxBZGRyZXNzPgogICAgICAgICAgICAgICAgPGNiYzpTdHJlZXR
OYW1lPtin2YTYp9mF2YrYsSDYs9mE2LfYp9mGIHwgUHJpbmNlIFN1bHRhbjwvY2JjOlN0cmVldE5hbWU
+CiAgICAgICAgICAgICAgICA8Y2JjOkJ1aWxkaW5nTnVtYmVyPjIzMjI8L2NiYzpCdWlsZGluZ051bWJlcj4K
ICAgICAgICAgICAgICAgIDxjYmM6Q2l0eVN1YmRpdmlzaW9uTmFtZT7Yp9mE2YXYsdio2LkgfCBBbC1Nd
XJhYmJhPC9jYmM6Q2l0eVN1YmRpdmlzaW9uTmFtZT4KICAgICAgICAgICAgICAgIDxjYmM6Q2l0eU5hb
WU+2KfZhNix2YrYp9i2IHwgUml5YWRoPC9jYmM6Q2l0eU5hbWU+CiAgICAgICAgICAgICAgICA8Y2JjOlB
vc3RhbFpvbmU+MjMzMzM8L2NiYzpQb3N0YWxab25lPgogICAgICAgICAgICAgICAgPGNhYzpDb3VudHJ
5PgogICAgICAgICAgICAgICAgICAgIDxjYmM6SWRlbnRpZmljYXRpb25Db2RlPlNBPC9jYmM6SWRlbnRp
ZmljYXRpb25Db2RlPgogICAgICAgICAgICAgICAgPC9jYWM6Q291bnRyeT4KICAgICAgICAgICAgPC9jYW
M6UG9zdGFsQWRkcmVzcz4KICAgICAgICAgICAgPGNhYzpQYXJ0eVRheFNjaGVtZT4KICAgICAgICAg
ICAgICAgIDxjYmM6Q29tcGFueUlEPjM5OTk5OTk5OTkwMDAwMzwvY2JjOkNvbXBhbnlJRD4KICAgICA
gICAgICAgICAgIDxjYWM6VGF4U2NoZW1lPgogICAgICAgICAgICAgICAgICAgIDxjYmM6SUQ+VkFUPC9j
YmM6SUQ+CiAgICAgICAgICAgICAgICA8L2NhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgIDwvY2FjOlBhc
nR5VGF4U2NoZW1lPgogICAgICAgICAgICA8Y2FjOlBhcnR5TGVnYWxFbnRpdHk+CiAgICAgICAgICAgIC
AgICA8Y2JjOlJlZ2lzdHJhdGlvbk5hbWU+2LTYsdmD2Kkg2KrZiNix2YrYryDYp9mE2KrZg9mG2YjZhNmI2K
zZitinINio2KPZgti12Ykg2LPYsdi52Kkg2KfZhNmF2K3Yr9mI2K/YqSB8IE1heGltdW0gU3BlZWQgVGVjaCBT
dXBwbHkgTFREPC9jYmM6UmVnaXN0cmF0aW9uTmFtZT4KICAgICAgICAgICAgPC9jYWM6UGFydHlM
ZWdhbEVudGl0eT4KICAgICAgICA8L2NhYzpQYXJ0eT4KICAgIDwvY2FjOkFjY291bnRpbmdTdXBwbGllcl
BhcnR5PgogICAgIDxjYWM6QWNjb3VudGluZ0N1c3RvbWVyUGFydHk+CiAgICAgICAgPGNhYzpQYXJ0
eT4KICAgICAgICAgICAgPGNhYzpQb3N0YWxBZGRyZXNzPgogICAgICAgICAgICAgICAgPGNiYzpTdHJ
lZXROYW1lPti12YTYp9itINin2YTYr9mK2YYgfCBTYWxhaCBBbC1EaW48L2NiYzpTdHJlZXROYW1lPgogI
CAgICAgICAgICAgICAgPGNiYzpCdWlsZGluZ051bWJlcj4xMTExPC9jYmM6QnVpbGRpbmdOdW1iZXI+C
iAgICAgICAgICAgICAgICA8Y2JjOkNpdHlTdWJkaXZpc2lvbk5hbWU+2KfZhNmF2LHZiNisIHwgQWwtTXV
yb29qPC9jYmM6Q2l0eVN1YmRpdmlzaW9uTmFtZT4KICAgICAgICAgICAgICAgIDxjYmM6Q2l0eU5hbWU
+2KfZhNix2YrYp9i2IHwgUml5YWRoPC9jYmM6Q2l0eU5hbWU+CiAgICAgICAgICAgICAgICA8Y2JjOlBvc3
RhbFpvbmU+MTIyMjI8L2NiYzpQb3N0YWxab25lPgogICAgICAgICAgICAgICAgPGNhYzpDb3VudHJ5Pgo
gICAgICAgICAgICAgICAgICAgIDxjYmM6SWRlbnRpZmljYXRpb25Db2RlPlNBPC9jYmM6SWRlbnRpZmljY
XRpb25Db2RlPgogICAgICAgICAgICAgICAgPC9jYWM6Q291bnRyeT4KICAgICAgICAgICAgPC9jYWM6U
G9zdGFsQWRkcmVzcz4KICAgICAgICAgICAgPGNhYzpQYXJ0eVRheFNjaGVtZT4KICAgICAgICAgICAgICAg
ICAgIDxjYmM6Q29tcGFueUlEPjM5OTk5OTk5OTgwMDAwMzwvY2JjOkNvbXBhbnlJRD4KICAgICAgICAgICA
gICAgIDxjYWM6VGF4U2NoZW1lPgogICAgICAgICAgICAgICAgICAgIDxjYmM6SUQ+VkFUPC9jYmM
6SUQ+CiAgICAgICAgICAgICAgICA8L2NhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgIDwvY2FjOlBhcnR
5VGF4U2NoZW1lPgogICAgICAgICAgICA8Y2FjOlBhcnR5TGVnYWxFbnRpdHk+CiAgICAgICAgICAgICAg
ICA8Y2JjOlJlZ2lzdHJhdGlvbk5hbWU+2LTYsdmD2Kkg2YbZhdin2LDYrCDZgdin2KrZiNix2Kkg2KfZhNmF2
K3Yr9mI2K/YqSB8IEZhdG9vcmEgU2FtcGxlcyBMVEQ8L2NiYzpSZWdpc3RyYXRpb25OYW1lPgogICAgI
CAgICA8L2NhYzpQYXJ0eUxlZ2FsRW50aXR5PgogICAgICAgIDwvY2FjOlBhcnR5PgogICAgPC9jY
WM6QWNjb3VudGluZ0N1c3RvbWVyUGFydHk+CiAgICA8Y2FjOlBheW1lbnRNZWFucz4KICAgICAgICA8
Y2JjOlBheW1lbnRNZWFuc0NvZGU+MTA8L2NiYzpQYXltZW50TWVhbnNDb2RlPgogICAgPC9jYWM6UG
F5bWVudE1lYW5zPgogICAgPGNhYzpBbGxvd2FuY2VDaGFyZ2U+CiAgICAgICAgPGNiYzpDaGFyZ2VJ
bmRpY2F0b3I+ZmFsc2U8L2NiYzpDaGFyZ2VJbmRpY2F0b3I+CiAgICAgICAgPGNiYzpBbGxvd2FuY2VD
aGFyZ2VSZWFzb24+ZGlzY291bnQ8L2NiYzpBbGxvd2FuY2VDaGFyZ2VSZWFzb24+CiAgICAgICAgPG
NiYzpBbW91bnQgY3VycmVuY3lJRD0iU0FSIj4wLjAwPC9jYmM6QW1vdW50PgogICAgICAgIDxjYWM6V
GF4Q2F0ZWdvcnk+CiAgICAgICAgICAgIDxjYmM6SUQgc2NoZW1lSUQ9IlVOL0VDRSA1MzA1IiBzY2hlb
WVBZ2VuY3lJRD0iNiI+UzwvY2JjOklEPgogICAgICAgICAgICA8Y2JjOlBlcmNlbnQ+MTU8L2NiYzpQZXJjZ
W50PgogICAgICAgICAgICA8Y2FjOlRheFNjaGVtZT4KICAgICAgICAgICAgICAgIDxjYmM6SUQgc2NoZW
1lSUQ9IlVOL0VDRSA1MTUzIiBzY2hlbWVBZ2VuY3lJRD0iNiI+VkFUPC9jYmM6SUQ+CiAgICAgICAgICA
gIDwvY2FjOlRheFNjaGVtZT4KICAgICAgICA8L2NhYzpUYXhDYXRlZ29yeT4KICAgICAgICA8Y2FjOlRheE
NhdGVnb3J5PgogICAgICAgICAgICA8Y2JjOklEIHNjaGVtZUlEPSJVTi9FQ0UgNTMwNSIgc2NoZW1lQWd
lbmN5SUQ9IjYiPlM8L2NiYzpJRD4KICAgICAgICAgICAgPGNiYzpQZXJjZW50PjE1PC9jYmM6UGVyY2Vu
dD4KICAgICAgICAgICAgPGNhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgICAgICA8Y2JjOklEIHNjaGVtZUl

*EPSJVTi9FQ0UgNTE1MyIgc2NoZW1lQWdlbmN5SUQ9IjYiPlZBVDwvY2JjOklEPgogICAgICAgICAgICA8L
2NhYzpUYXhTY2hlbWU+CiAgICAgICAgPC9jYWM6VGF4Q2F0ZWdvcnk+CiAgICA8L2NhYzpBbGxvd2Fu
Y2VDaGFyZ2U+CiAgICA8Y2FjOlRheFRvdGFsPgogICAgICAgIDxjYmM6VGF4QW1vdW50IGN1cnJlbmN
5SUQ9IlNBUiI+MzAuMTU8L2NiYzpUYXhBbW91bnQ+CiAgICA8L2NhYzpUYXhUb3RhbD4KICAgIDxjYW
M6VGF4VG90YWw+CiAgICAgICAgPGNiYzpUYXhBbW91bnQgY3VycmVuY3lJRD0iU0FSIj4zMC4xNTwv
Y2JjOlRheEFtb3VudD4KICAgICAgICA8Y2FjOlRheFN1YnRvdGFsPgogICAgICAgICAgICA8Y2JjOlRheGFi
bGVBbW91bnQgY3VycmVuY3lJRD0iU0FSIj4yMDEuMDA8L2NiYzpUYXhhYmxlQW1vdW50PgogICAgIC
AgICAgICA8Y2JjOlRheEFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjMwLjE1PC9jYmM6VGF4QW1vdW50Pgo
gICAgICAgICAgICAgPGNhYzpUYXhDYXRlZ29yeT4KICAgICAgICAgICAgICAgICA8Y2JjOklEIHNjaGVtZ
UlEPSJVTi9FQ0UgNTMwNSIgc2NoZW1lQWdlbmN5SUQ9IjYiPlM8L2NiYzpJRD4KICAgICAgICAgICAgIC
AgICA8Y2JjOlBlcmNlbnQ+MTUuMDA8L2NiYzpQZXJjZW50PgogICAgICAgICAgICAgICAgPGNhYzpUYX
hTY2hlbWU+CiAgICAgICAgICAgICAgICAgICA8Y2JjOklEIHNjaGVtZUlEPSJVTi9FQ0UgNTE1MyIgc2NoZ
W1lQWdlbmN5SUQ9IjYiPlZBVDwvY2JjOklEPgogICAgICAgICAgICAgICAgPC9jYWM6VGF4U2NoZW1lPg
ogICAgICAgICAgICAgPC9jYWM6VGF4Q2F0ZWdvcnk+CiAgICAgICAgPC9jYWM6VGF4U3VidG90YWw+
CiAgICA8L2NhYzpUYXhUb3RhbD4KICAgIDxjYWM6TGVnYWxNb25ldGFyeVRvdGFsPgogICAgICAgIDxj
YmM6TGluZUV4dGVuc2lvbkFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjIwMS4wMDwvY2JjOkxpbmVFeHRlb
nNpb25BbW91bnQ+CiAgICAgICAgPGNiYzpUYXhFeGNsdXNpdmVBbW91bnQgY3VycmVuY3lJRD0iU0F
SIj4yMDEuMDA8L2NiYzpUYXhFeGNsdXNpdmVBbW91bnQ+CiAgICAgICAgPGNiYzpUYXhJbmNsdXNp
dmVBbW91bnQgY3VycmVuY3lJRD0iU0FSIj4yMzEuMTU8L2NiYzpUYXhJbmNsdXNpdmVBbW91bnQ+Ci
AgICAgICAgPGNiYzpBbGxvd2FuY2VUb3RhbEFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjAuMDA8L2NiYzpB
bGxvd2FuY2VUb3RhbEFtb3VudD4KICAgICAgICA8Y2JjOlByZXBhaWRBbW91bnQgY3VycmVuY3lJRD0i
U0FSIj4wLjAwPC9jYmM6UHJlcGFpZEFtb3VudD4KICAgICAgICA8Y2JjOlBheWFibGVBbW91bnQgY3Vyc
mVuY3lJRD0iU0FSIj4yMzEuMTU8L2NiYzpQYXlhYmxlQW1vdW50PgogICAgPC9jYWM6TGVnYWxNb25l
dGFyeVRvdGFsPgogICAgPGNhYzpJbnZvaWNlTGluZT4KICAgICAgICA8Y2JjOklEPjE8L2NiYzpJRD4KI
CAgICAgICA8Y2JjOkludm9pY2VkUXVhbnRpdHkgdW5pdENvZGU9IlBDRSI+MzMuMDAwMDAwPC9jYm
M6SW52b2ljZWRRdWFudGl0eT4KICAgICAgICA8Y2JjOkxpbmVFeHRlbnNpb25BbW91bnQgY3VycmVuY
3lJRD0iU0FSIj45OS4wMDwvY2JjOkxpbmVFeHRlbnNpb25BbW91bnQ+CiAgICAgICAgPGNhYzpUYXhUb
3RhbD4KICAgICAgICAgICAgIDxjYmM6VGF4QW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+MTQuODU8L2Ni
YzpUYXhBbW91bnQ+CiAgICAgICAgICAgICA8Y2JjOlJvdW5kaW5nQW1vdW50IGN1cnJlbmN5SUQ9IlNB
UiI+MTEzLjg1PC9jYmM6Um91bmRpbmdBbW91bnQ+CiAgICAgICAgPC9jYWM6VGF4VG90YWw+CiAgIC
AgICAgPGNhYzpJdGVtPgogICAgICAgICAgICA8Y2JjOk5hbWU+2YPYqtin2Kg8L2NiYzpOYW1lPgogICAg
ICAgICAgICA8Y2FjOkNsYXNzaWZpZWRUYXhDYXRlZ29yeT4KICAgICAgICAgICAgICAgIDxjYmM6SUQ
+UzwvY2JjOklEPgogICAgICAgICAgICAgICAgPGNiYzpQZXJjZW50PjE1LjAwPC9jYmM6UGVyY2VudD4
KICAgICAgICAgICAgICAgIDxjYWM6VGF4U2NoZW1lPgogICAgICAgICAgICAgICAgICAgIDxjYmM6SUQ+
VkFUPC9jYmM6SUQ+CiAgICAgICAgICAgICAgICA8L2NhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgIDwv
Y2FjOkNsYXNzaWZpZWRUYXhDYXRlZ29yeT4KICAgICAgICA8L2NhYzpJdGVtPgogICAgICAgIDxjYWM
6UHJpY2U+CiAgICAgICAgICAgIDxjYmM6UHJpY2VBbW91bnQgY3VycmVuY3lJRD0iU0FSIj4zLjAwPC9j
YmM6UHJpY2VBbW91bnQ+CiAgICAgICAgPC9jYWM6UHJpY2U+CiAgICA8L2NhYzpJbnZvaWNlTGluZT
4KICAgIDxjYWM6SW52b2ljZUxpbmU+CiAgICAgICAgPGNiYzpJRD4yPC9jYmM6SUQ+CiAgICAgICAgP
GNiYzpJbnZvaWNlZFF1YW50aXR5IHVuaXRDb2RlPSJQQ0UiPjMuMDAwMDAwPC9jYmM6SW52b2ljZW
RRdWFudGl0eT4KICAgICAgICA8Y2JjOkxpbmVFeHRlbnNpb25BbW91bnQgY3VycmVuY3lJRD0iU0FSIj4
xMDIuMDA8L2NiYzpMaW5lRXh0ZW5zaW9uQW1vdW50PgogICAgICAgIDxjYWM6VGF4VG90YWw+CiA
gICAgICAgICAgICA8Y2JjOlRheEFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjE1LjMwPC9jYmM6VGF4QW1vd
W50PgogICAgICAgICAgICAgPGNiYzpSb3VuZGluZ0Ftb3VudCBjdXJyZW5jeUlEPSJTQVIiPjExNy4zMDw
vY2JjOlJvdW5kaW5nQW1vdW50PgogICAgICAgIDwvY2FjOlRheFRvdGFsPgogICAgICAgIDxjYWM6SXRl
bT4KICAgICAgICAgICAgPGNiYzpOYW1lPtmC2YTZhTwvY2JjOk5hbWU+CiAgICAgICAgICAgIDxjYWM6
Q2xhc3NpZmllZFRheENhdGVnb3J5PgogICAgICAgICAgICAgICAgPGNiYzpJRD5TPC9jYmM6SUQ+CiAg
ICAgICAgICAgICAgICA8Y2JjOlBlcmNlbnQ+MTUuMDA8L2NiYzpQZXJjZW50PgogICAgICAgICAgICAgIC
AgPGNhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgICAgICAgICAgPGNiYzpJRD5WQVQ8L2NiYzpJRD4KI
CAgICAgICAgICAgICAgIDwvY2FjOlRheFNjaGVtZT4KICAgICAgICAgICAgPC9jYWM6Q2xhc3NpZmllZFR
heENhdGVnb3J5PgogICAgICAgIDwvY2FjOkl0ZW0+CiAgICAgICAgPGNhYzpQcmljZT4KICAgICAgICAgI
CAgPGNiYzpQcmljZUFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjM0LjAwPC9jYmM6UHJpY2VBbW91bnQ+Ci
AgICAgICAgPC9jYWM6UHJpY2U+CiAgICA8L2NhYzpJbnZvaWNlTGluZT4KPC9JbnZvaWNlPg==" } }*

### Request headers

### Return type

InvoiceResultModel

### Example data

Content-Type: application/json

{ "validationResults" : [ { "warningMessages" : [ { "code" : "code", "category" : "category", "message" : "message" }, { "code" : "code", "category" : "category", "message" : "message" } ], "infoMessages" : [ { "message" : "message" }, { "message" : "message" } ], "erroMessages" : [ { "code" : "code", "category" : "category", "message" : "message" }, { "code" : "code", "category" : "category", "message" : "message" } ], "status" : "PASS" }, { "warningMessages" : [ { "code" : "code", "category" : "category", "message" : "message" }, { "code" : "code", "category" : "category", "message" : "message" } ], "infoMessages" : [ { "message" : "message" }, { "message" : "message" } ], "erroMessages" : [ { "code" : "code", "category" : "category", "message" : "message" }, { "code" : "code", "category" : "category", "message" : "message" } ], "status" : "PASS" } ], "reportingStatus" : "REPORTED" }

**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

**Responses**

**200**

HTTP OK. Returned on successful validation of simplified invoice. InvoiceResultModel

**Example data**

Content-Type: reported

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

**202**

HTTP Accepted. Returned when the invoice is reported with warnings InvoiceResultModel

**Example data**

Content-Type: reported

{"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":[{"type":"WARNING","code":"BR-CO-17","category":"EN_16931","message":"VAT category tax amount (BT-117) = VAT category taxable amount (BT-116) x (VAT category rate (BT-119) / 100), rounded to two decimals.","status":"WARNING"},{"type":"WARNING","code":"BR-KSA-98","category":"KSA","message":"[BR-KSA-98] - The simplified invoice should be submitted within 24 hours of issuing the invoice.","status":"WARNING"}],"errorMessages":[],"status":"WARNING"},"reportingStatus":"REPORTED"}

**400**

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

**Example data**

Content-Type: KSA Rules Violation

{"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-06","category":"KSA","message":"The invoice transaction code (KSA-2) must exist and respect the following structure: NNPNESB where NN (positions 1 and 2) = invoice subtype: - 01 for tax invoice - 02 for simplified tax invoice P (position 3) = 3rd Party invoice transaction, 0 for false, 1 for true N (position 4) = Nominal invoice transaction, 0 for false, 1 for true E (position 5) = Exports invoice transaction, 0 for false, 1 for true S (position 6) = Summary invoice transaction, 0 for false, 1 for true B (position 7) = Self billed invoice","status":"ERROR"}],"status":"ERROR"},"reportingStatus":"NOT_REPORTED"}

**Example data**

Content-Type: Missing QR Code

{"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":"QRCODE_INVALID","category":"QRCODE_VALIDATION","message":"Invalid QR code format, Please follow the ZATCA QR code specifications","status":"ERROR"}],"status":"ERROR"},"reportingStatus":"NOT_REPORTED"}

**Example data**

Content-Type: Invalid QR Code

{"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":"QRCODE_INVALID","category":"QRCODE_VALIDATION","message":"Failed to validate QR code","status":"ERROR"}],"status":"ERROR"},"reportingStatus":"NOT_REPORTED"}

**Example data**

Content-Type: EN Rules Violation

{"validationResults":{"infoMessages":[],"warningMessages":[],"errorMessages":[{"type":"ERROR","code":"BR-26","category":"EN_16931","message":"Each Invoice line (BG-25) shall contain the Item net price (BT-146).","status":"ERROR"}],"status":"ERROR"},"reportingStatus":"NOT_REPORTED"}

**Example data**

Content-Type: XML Schema Error

```
{"validationResults":{"infoMessages":[],"warningMessages":[],"errorMessages":[{"type":"ERROR","code":"XSD_ZATCA_INVALID","category":"XSD validation","message":"Schema validation failed; XML does not comply with UBL 2.1 standards in line with ZATCA specifications. ERROR: org.xml.sax.SAXParseException; lineNumber: 204; columnNumber: 19; cvc-complex-type.2.4.b: The content of element 'cac:Price' is not complete. One of '{\"urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2\":PriceAmount}' is expected.","status":"ERROR"}],"status":"ERROR"},"reportingStatus":"NOT_REPORTED"}
```

**Example data**
Content-Type: InvalidInvoiceHash

```
{"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":"invalid-invoice-hash","category":"INVOICE_HASHING_ERRORS","message":"The invoice hash API body does not match the (calculated) Hash of the XML","status":"ERROR"},
{"type":"ERROR","code":"invoiceHash_QRCODE_INVALID","category":"QRCODE_VALIDATION","message":"Invoice xml hash does not match with qr code invoice xml hash","status":"ERROR"}],"status":"ERROR"},"reportingStatus":"NOT_REPORTED"}
```

**Example data**
Content-Type: Missing Invoice

```
{"validationResults":{"infoMessages":[],"warningMessages":[],"errorMessages":[{"type":"ERROR","code":"XSD_ZATCA_INVALID","category":"XSD validation","message":"Schema validation failed; XML does not comply with UBL 2.1 standards in line with ZATCA specifications","status":"ERROR"}],"status":"ERROR"},"reportingStatus":"NOT_REPORTED"}
```

**Example data**
Content-Type: Missing Invoice Hash

```
{"validationResults":{"infoMessages":[],"warningMessages":[],"errorMessages":[{"type":"ERROR","code":"InvoiceHash-Errors","category":"Invalid-InvoiceHash","message":"Document hash is not present in the API body","status":"ERROR"}],"status":"ERROR"},"reportingStatus":"NOT_REPORTED"}
```

**Example data**
Content-Type: Invalid Signature

```
{"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":"invalid-digital-signature","category":"SIGNATURE_ERRORS","message":"Invalid digital signature","status":"ERROR"}],"status":"ERROR"},"reportingStatus":"NOT_REPORTED"}
```

**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":""}
```

**409**
Invoice was already Reported successfully earlier. InvoiceResultModel
**Example data**
Content-Type: conflict

```
{"validationResults":{"infoMessages":[],"warningMessages":[],"errorMessages":[{"type":"ERROR","code":null,"category":null,"message":"Invoice was already Reported successfully earlier.","status":"ERROR"}],"status":"ERROR"},"reportingStatus":"NOT_REPORTED"}
```

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

```
{"category":"HTTP-Errors","code":"500","message":"Something went wrong and caused an Internal Server Error."}
```

## Models

[ Jump to Methods ]

### Table of Contents

### 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)

**validationResults (optional)**
*array[validationResultsModel]*

**reportingStatus (optional)**
*String*

**Enum:**
*REPORTED*
*NOT_REPORTED*

### 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*

### validationResultsModel - validationResultsModel

Up

An object representing the structure of the validation results returned by the API endpoints. Specifically, it includes the invoice hash, status, and lists of info, warning, and error messages.

**infoMessages (optional)**
*array[InfoModel]*

**warningMessages (optional)**
*array[WarningModel]*

**erroMessages (optional)**
*array[ErrorModel]*

**status (optional)**
*String*

**Enum:**

*PASS*
*WARNING*
*ERROR*

# 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

## PATCH /production/csids

Up

Renews an X509 Certificate (CSID) based on submitted CSR. (**renewCertificatesUsingPOST**)

Renews an X509 Certificate (CSID) based on submitted CSR

### Consumes

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

- application/json

### Request body

**body CSRRequest (required)**
*Body Parameter* — CSR Request in body as Base64.
*example:* { "csr" : "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFVRVNULS0tLS0NCk1JSUNGekNDQWI0Q0FRQXdkVEVMTUFrR0ExVUVCaE1DVTBFeEZqQVVCZ05WQkFzTURWSnBlV0ZrYUNCQ2NtRnUNClkyZ3hKakFrQmdOVkJBb01IVTFoZUdsdGRXMGdVM0JsWldRZ1ZHVmphQ0JUZFhCd2JIa2dURlJF
TVNZd0pBWUQNClZRUUREQjFVVFRdE9EZzJORE14TVRRMUxUTTVPVGs1T1RrNU9Ua3dNREF3T
XpCV01CQUdCeXFHU000OUFnRUNCkJTdUJCQUFLQTBJQUJPVlQvWFVGcDVJdU0wWUZHSTZUU
StBTGxWT0Z3RldxckNXMVU3M1NpUFJDSGM2Q1V4UXcNCmxoeG14aFRjNUdEOU1xRFM3YVArZi9O
R0F3ZWJ6UmJvRjJpZ2dla3dnZVlHQ1NxR1NJYjNEUUVKRGpFQ0INCjFUQWhCZ2tyQmdFRUFZSS

*TNGQUlFRkF3U1drRlVRMEV0UTI5a1pTMVRhV2R1YVc1bk1JR3ZCZ05WSFJFRWdhY3cNCmdhU2tnY
UV3Z1o0eE96QTVCZ05WQkFRTU1qRXRWRk5VZkRJdFZGTlVmRE10WldReU1tWXhaRGd0WlRaaE1p
MHgNCk1URTRMVGxpTlRndFpEbGhPR1l4TVdVME5EVm1NUjh3SFFZS0NaSW1pWlB5TEdRQkFRd1B
Nems1T1RrNU9UazUNCk9UQXdNREF6TVEwd0N3WURWUVFNREFReE1UQXdNUkV3RHdZRFZRU
WFEQWhTVWxKRU1qa3lPVEVjTUJvR0ExVUUNCkR3d1QyWTFUZFhCd2JIa2dZV04wYVhacGRHbGxj
ekFLQmdncWhrak9QUVFEQWdOSEFEQkVBaUJCWXNyRkZLUGYNCml4SkhSRVNxRlF0cmI0L0orMX
RNczlrcWNGdEJ2eGdiR3dJZ0tXaWE0ZVcyZkhYY1d4amp5T3RqdVQ5MmhMUDMNCjRKOFBNVnFqeD
R4TDFVTT0NCi0tLS0tRU5EIENFUlRJRklDQVRFIFJFUVVFU1QtLS0tLQ0K" }*

### Request headers

### Return type

Object

### 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 Object

### Example data

Content-Type: application/json

{"requestID":347,"tokenType":"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3","dispositionMessage":"ISSUED","binarySecurityToken":"TUlJRDRUQ0NBNGFnQXdJQkFnSVRGd0FBTzJVV
XE4RmR0Z1lLNWdBQkFBQTdaVEFLQmdncWhrak9QUVFEQWpCaU1SVXdFd1lLQ1pJbWlaUHlMR1FCR1JZRmJHOWp
ZV3d4RXpBUkJnb0praWFKay9Jc1pBRVpGZ05uYjNZeEZ6QVZCZ29Ka2lhSmsvSXNaQUVaRmdkbGVIUm5ZWHAwTVJz
d0dRWURWUVFERXhKUVVscEZTVTVXVDBsRFJWTkRRVEV0UTBFd0hoY05aelF3TVRFMU1UY3lOVEU1V2hjTk1qa3d
NVEV6TVRjeU5URTVXaklxTVFzd0NRWURWUVFHRXdKVFFURW1NQ1FHQTFVRUNoTWRUV0Y0YVcxMWJTQlRjR1Zs
WkNCVVpXTm9JRk4xY0hCc2VTQk1WRVF4RmpBVUJnTlZCQXNURFZKcGVXRmthQ0JDY21GdVkyZ3hKakFrQmdOVkJ
BTVRIVlJUVkMwNE9EWTBNekV4TkRVdE16azVPVGs1T1RrNU9UQXdNREF6TUZZd0VBWUhLb1pJemowQ0FRWUZLN
EVFQUFvRFFnQUU1VlA5ZFFXbmtpNHpSZ1VZanBORDRBdVZVNFhBVmFxc0piVlR2ZEtJOUVJZHpvSlRGRENXSEdiR0
ZOemtZUDB5b05MdG8vNS84MFlEQjV2TkZ1Z1hhS09DQWdrd2dnSUZNSUd2QmdOVkhSRUVnYWN3Z2FTa2dhRXdnWj
R4T3pBNUJnTlZCQVFNTWpFdFZGTlVmREl0VkZOVWZETXRaV1F5TW1ZeFpEZ3RaVFpoTWkweE1URTRMVGxpTlRnd
FpEbGhPR1l4TVdVME5EVm1NUjh3SFFZS0NaSW1pWlB5TEdRQkFRd1BNems1T1RrNU9UazVPVEF3TURBek1RMHdD
d1lEVlFRTURBUXhNVEF3TVJFd0R3WURWUVFhREFoU1VsSkVNamt5T1RFY01Cb0dBMVVFRHd3VDJZMVRkWEJ3Ykh
rZ1lXTjBhWFpwZEdsbGN6QWRCZ05WSFE0RUZnUVVadkJOcHdmMFJzWTBvU2QyWXo2Tjg0aXhCRll3SHdZRFZSMGp
CQmd3Rm9BVWNwUFJEbXY2SkZzVGhlckJGZk80RmZzYkJZMHdld1lJS3dZQkJRVUhBUUVFYnpCdE1Hc0dDQ3NHQVF
VRkJ6QUNobDlvZEhSd09pOHZZV2xoTVM1NllYUmpZUzVuYjNZdWMyRXZRMlZ5ZEVWdWNtOXNiQzlRVWxwRmFXNTJi
MmxqWlZORFFURXVaWGgwWjJGNmRDNW5iM1l1Ykc5allXeGZVRkphUlVsT1ZrOUpRMFZUUTBFeExVTkJLREVwTG1
OeWREQU9CZ05WSFE4QkFmOEVCQU1DQjRBd1BBWUpLd1lCQkFHQ054VUhCQzh3TFFZbEt3WUJCQUdDTnhVSWd
ZYW9lWVRRK3hLRzdaMGtoODc3R2RQQVZXYUgrcVZsaGRtRVBnSUJaQUlCRWpBZEJnTlZIU1VFRmpBVUJnZ3JCZ0V
GQlFjREF3WUlLd1lCQlFVSEF3SXdKd1lKS3dZQkJBR0NOeFVLQkJvd0dEQUtCZ2dyQmdFRkJRY0RBekFLQmdnckJnRU
ZCUWNEQWpBS0JnZ3Foa2pPUFFRREFnTkpBREJHQWlFQS9vaDRIb2FlTGh6SDFNN2YrTjBrSmZoSW42RHlzQkZaWE
ZNcGdnK3poeG9DSVFDVWwweEtyTGxuZEM5V25QdGVSNUx1dVF2amdQQUpvVklFd2JDeVJpSXk2dz09","secret":"1np
mzeTq4VnKFQaZ5/9SwUxNhoLtuMWZVLVWUm3MTVU="}

**400**

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

### Example data

Content-Type: Invalid OTP

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

### Example data

Content-Type: Missing CSR

```
{"errors":[{"code":"Missing-CSR","message":"CSR is required field"}]}
```

### Example data

Content-Type: Invalid CSR

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

**Example data**
Content-Type: Missing currentCSID

```
{"errors":[{"code":"Missing-currentCSID","message":"currentCSID is required field"}]}
```

**Example data**
Content-Type: Invalid currentCSID

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

**Example data**
Content-Type: Missing OTP

```
{"errors":[{"code":"Missing-OTP","message":"OTP is required field"}]}
```

**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**
Returned when accept version header is anything other than V2
**Example data**
Content-Type: Not Acceptable

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

**428**
returns a Base64 encoded X509 compliance certificate Object
**Example data**
Content-Type: application/json

```
{"value":
{"requestID":1234567890123,"tokenType":null,"dispositionMessage":"NOT_COMPLIANT","binarySecurityToken":"TUlJQ1FE
Q0NBZVdnQXdJQkFnSUdBWTBPTFNiWk1Bb0dDQ3FHU000OUJBTUNNQlV4RXpBUkJnTlZCQU1NQ21WSmJuWnZhV0
5wYm1jd0hoY05NalF3TVRFMU1UY3pNRFV4V2hjTk1qa3dNVEUwTWpFd01EQXdXaklxTVFzd0NRWURWUVFHRXdKVF
FURVdNQlFHQTFVRUN3d05VbWw1WVdSb0lFSnlZVzVqYURFbU1DUUdBMVVFQ2d3ZFRXRjRhVzExYlNCVGNHVmxa
Q0JVWldOb0lGTjFjSEJzZVNCTVZFUXhKakFrQmdOVkJBTU1IVlJUVkMwNE9EWTBNekV4TkRVdE16azVPVGs1T1RrNU
9UQXdNREF6TUZZd0VBWUhLb1pJemowQ0FRWUZLNEVFQUFvRFFnQUU1VlA5ZFFXbmtpNHpSZ1VZanBORDRBdVZ
VNFhBVmFxc0piVlR2ZEtJOUVJZHpvSlRGRENXSEdiR0ZOemtZUDB5b05MdG8vNS84MFlEQjV2TkZ1Z1hhS09Cd3pDQn
dEQU1CZ05WSFJNQkFmOEVBakFBTUlHdkJnTlZIUkVFZ2Fjd2dhU2tnYUV3Z1o0eE96QTVCZ05WQkFRTU1qRXRWRk5
VZkRJdFZGTlVmRE10WldReU1tWXhaRGd0WlRaaE1pMHhNVEU0TFRsaU5UZ3RaRGxoT0dZeE1XVTBORFZtTVI4d0hR
WUtDWkltaVpQeUxHUUJBUXdQTXprNU9UazVPVGs1T1RBd01EQXpNUTB3Q3dZRFZRUU1EQVF4TVRBd01SRXdEd1l
EVlFRYURBaFNVbEpFTWpreU9URWNNQm9HQTFVRUR3d1QyWTFUZFhCd2JIa2dZV04wYVhacGRHbGxjekFLQmdnc
Whrak9QUVFEQWdOSkFEQkdBaUVBM1JsTTJlaGZaMzFmdk5yRGlJKzI5c0crNGJVVlg2QWZ1eEJuNUJiLzZ4TUNJUUQ
5ZmxVNTc4U0htdEdZeTNWaW9LSU1VMFpMVHJaT2liOXdHVTliTFJiYll3PT0=","secret":"goDgeIdM5mkfTThl1unu8rP9Xhk
nKWAc24hafXZS1f4=","errors":null}}
```

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

```
{"category":"HTTP-Errors","code":"500","message":"Something went wrong and caused an Internal Server Error."}
```

## 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*

# 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":"TUlJRDNqQ0NBNFNnQXdJQkFnSV
RFUUFBT0FQRjkwQWpzL3hjWHdBQkFBQTRBekFLQmdncWhrak9QUVFEQWpCaU1SVXdFd1lLQ1pJbWlaUHlMR1FCR
1JZRmJHOWpZV3d4RXpBUkJnb0praWFKay9Jc1pBRVpGZ05uYjNZeEZ6QVZCZ29Ka2lhSmsvSXNaQUVaRmdkbGVlUm
5ZWHAwTVJzd0dRWURWUVFERXhKUVVscEZTVTVXVDBsRFJWTkRRVFF0UTBFd0hoY05NalF3TVRFeE1Ea3hPVE13
V2hjTk1qa3dNVEE1TURreE9UTXdXakIxTVFzd0NRWURWUVFHRXdKVFFURW1NQ1FHQTFVRUNoTWRUV0Y0YVcxM
WJTQlRjR1ZsWkNCVVpXTm9JRk4xY0hCc2VTQk1WRVF4RmpBVUJnTlZCQXNURFZKcGVXRmthQ0JDY21GdVkyZ3hK
akFrQmdOVkJBTVRIVlJUVkMwNE9EWTBNekV4TkRVdE16azVPVGs1T1RrNU9UQXdNREF6TUZZd0VBWUhLb1pJemow
Q0FRWUZLNEVFQUFvRFFnQUVvV0NLYTBTYTlGSUVyVE92MHVBa0MxVklLWHhVOW5QcHgydmxmNHloTWVqeThjM
DJYSmJsRHE3dFB5ZG84bXEwYWhPTW1Obzhnd25pN1h0MUtUOVVlS09DQWdjd2dnSURNSUd0QmdOVkhSRUVnYVV
3Z2FLa2daOHdnWnd4T3pBNUJnTlZCQVFNTWpFdFZGTlVmREl0VkZOVWZETXRaV1F5TW1ZeFpEZ3RaVFpoTWkweE1
URTRMVGxpTlRndFpEbGhPR1l4TVdVME5EVm1NUjh3SFFZS0NaSW1pWlB5TEdRQkFRd1BNems1T1RrNU9UazVPVEF
3TURBek1RMHdDd1lEVlFRTURBUXhNVEF3TVJFd0R3WURWUVFhREFoU1VsSkVNamt5T1RFYU1CZ0dBMVVFRHd3Ul
UzVndjR3g1SUdGamRHbDJhWFJwWlhNd0hRWURWUjBPQkJZRUZFWCtZdm1tdG5Zb0RmOUJHYktvN29jVEtZSzFNQjh
HQTFVZEl3UVlNQmFBRkp2S3FxTHRtcXdza0lGelZ2cFAyUHhUKzlObk1Ic0dDQ3NHQVFVRkJ3RUJCRzh3YlRCckJnZ3J
CZ0VGQlFjd0FvWmZhSFIwY0RvdkwyRnBZVFF1ZW1GMFkyRXVaMjkyTG5OaEwwTmxjblJGYm5KdmJHd3ZVRkphUlVsd
WRtOXBZMlZUUTBFMExtVjRkR2RoZW5RdVoyOTJMbXh2WTJGc1gxQlNXa1ZKVGxaUFNVTkZVME5CTkMxRFFTZ3hL
UzVqY25Rd0RnWURWUjBQQVFIL0JBUURBZ2VBTUR3R0NTc0dBUVFCZ2pjVkJ3UXZNQzBHSlNzR0FRUUJnamNWQ0l
HR3FCMkUwUHNTaHUyZEpJZk8reG5Ud0ZWbWgvcWxaWVhaaEQ0Q0FXUUNBUkl3SFFZRFZSMGxCQll3RkFZSUt3WU
JCUVVIQXdNR0NDc0dBUVVGQndNQ01DY0dDU3NHQVFRQmdqY1ZDZ1FhTUJnd0NnWUlLd1lCQlFVSEF3TXdDZ1lJS3
dZQkJRVUhBd0l3Q2dZSUtvWkl6ajBFQXdJRFNBQXdSUUloQUxFL2ljaG1uV1hDVUtVYmNhM3ljaThvcXdhTHZGZEhWalF
ydmVJOXVxQWJBaUE5aEM0TThqZ01CQURQU3ptZDJ1aVBKQTZnS1IzTEUwM1U3NWVxYkMvclhBPT0=","secret":"SX3
P87hpTma5qUsOEQWv46fHL9uGcKFow90i9ercnSY="}

**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: Unthorized

```
{"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*

# 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

# 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 InvoiceRequest (required)**

*Body Parameter —*

example: { "invoiceHash" : "V4U5qlZ3yXQ/Si1AC/R8SLc3F+iNy27wdVe8IWRqFAQ=", "uuid" : "8d487816
-70b8-4ade-a618-9d620b73814a", "invoice" :
"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPEludm9pY2UgeG1sbnM9InVybjpvYX
NpczpuYW1lczpzcGVjaWZpY2F0aW9uOnVibDpzY2hlbWE6eHNkOkludm9pY2UtMiIgeG1sbnM6Y2FjPSJ
1cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2NoZW1hOnhzZDpDb21tb25BZ2dyZWdhdG
VDb21wb25lbnRzLTIiIHhtbG5zOmNiYz0idXJuOm9hc2lzOm5hbWVzOnNwZWNpZmljYXRpb246dWJsOn
NjaGVtYTp4c2Q6Q29tbW9uQmFzaWNDb21wb25lbnRzLTIiIHhtbG5zOmV4dD0idXJuOm9hc2lzOm5hbW
VzOnNwZWNpZmljYXRpb246dWJsOnNjaGVtYTp4c2Q6Q29tbW9uRXh0ZW5zaW9uQ29tcG9uZW50cy0y
Ij48ZXh0OlVCTEV4dGVuc2lvbnM+CiAgICA8ZXh0OlVCTEV4dGVuc2lvbj4KICAgICAgICA8ZXh0OkV4dG
Vuc2lvblVSST51cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6ZHNpZzplbnZlbG9wZWQ6eGF
kZXM8L2V4dDpFeHRlbnNpb25VUkk+CiAgICAgICAgPGV4dDpFeHRlbnNpb25Db250ZW50PgogICAgICA
gICAgICA8c2lnOlVCTERvY3VtZW50U2lnbmF0dXJlcyB4bWxuczpzaWc9InVybjpvYXNpczpuYW1lczpzcG
VjaWZpY2F0aW9uOnVibDpzY2hlbWE6eHNkOkNvbW1vblNpZ25hdHVyZUNvbXBvbmVudHMtMiIgeG1sb
nM6c2FjPSJ1cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2NoZW1hOnhzZDpTaWduYXR1
cmVBZ2dyZWdhdGVDb21wb25lbnRzLTIiIHhtbG5zOnNiYz0idXJuOm9hc2lzOm5hbWVzOnNwZWNpZmlj
YXRpb246dWJsOnNjaGVtYTp4c2Q6U2lnbmF0dXJlQmFzaWNDb21wb25lbnRzLTIiPgogICAgICAgICAgIC
AgICAgPHNhYzpTaWduYXR1cmVJbmZvcm1hdGlvbj4gCiAgICAgICAgICAgICAgICAgICAgPGNiYzpJRD
51cm46b2FzaXM6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6c2lnbmF0dXJlOjE8L2NiYzpJRD4KICAgICAgI
CAgICAgICAgICAgICA8c2JjOlJlZmVyZW5jZWRTaWduYXR1cmVJRD51cm46b2FzaXM6bmFtZXM6c3Bl
Y2lmaWNhdGlvbjp1Ymw6c2lnbmF0dXJlOkludm9pY2U8L3NiYzpSZWZlcmVuY2VkU2lnbmF0dXJlSUQ+Ci
AgICAgICAgICAgICAgICAgPGRzOlNpZ25hdHVyZSB4bWxuczpkcz0iaHR0cDovL3d3dy53My5vcmc
vMjAwMC8wOS94bWxkc2lnIyIgSWQ9InNpZ25hdHVyZSI+CiAgICAgICAgICAgICAgICAgICAgICAgIDxkcz
pTaWduZWRJbmZvPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOkNhbm9uaWNhbGl6YXRpb
25NZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDA2LzEyL3htbC1jMTRuMTEiLz4KI
CAgICAgICAgICAgICAgICAgICAgICAgIDxkczpTaWduYXR1cmVNZXRob2QgQWxnb3JpdGhtPSJod
HRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNlY2RzYS1zaGEyNTYiLz4KICAgICAgI
CAgICAgICAgICAgIDxkczpSZWZlcmVuY2UgSWQ9Imludm9pY2VTaWduZWREYXRhIiBV
Ukk9IiI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOlRyYW5zZm9ybXM+CiAgICAgICA
gICAgICAgICAgICAgICAgIDxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRwOi8vd
3d3LnczLm9yZy9UUi8xOTk5L1JFQy14cGF0aC0xOTk5MTExNiI+CiAgICAgICAgICAgICAgICAgICA
gICAgICAgICAgICA8ZHM6WFBhdGg+bm90KC8vYW5jZXN0b3Itb3Itc2VsZjo6ZXh0OlVCTEV4d
GVuc2lvbnMpPC9kczpYUGF0aD4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPC9kczp
UcmFuc2Zvcm0+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpUcmFuc2Zvcm0gQ
Wxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy9UUi8xOTk5L1JFQy14cGF0aC0xOTk5MTExNiI+CiAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6WFBhdGg+bm90KC8vYW5jZXN0b3I
tb3Itc2VsZjo6Y2FjOlNpZ25hdHVyZSk8L2RzOlhQYXRoPgogICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICA8L2RzOlRyYW5zZm9ybT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPG
RzOlRyYW5zZm9ybSBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnL1RSLzE5OTkvUkVDLXhwYXRo
LTE5OTkxMTE2Ij4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxkczpYUGF0aD5
ub3QoLy9hbmNlc3Rvci1vci1zZWxmOjpjYWM6QWRkaXRpb25hbERvY3VtZW50UmVmZXJlbmNlW2NiYz
pJRD0nUVInXSk8L2RzOlhQYXRoPgogICAgICAgICAgICAgICAgICAgICAgICAgICA8L2Rz
OlRyYW5zZm9ybT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOlRyYW5zZm9y
bSBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDYvMTIveG1sLWMxNG4xMSIvPgogICAgICAg
ICAgICAgICAgICAgICAgIDwvZHM6VHJhbnNmb3Jtcz4KICAgICAgICAgICAgICAgICAgICA
gICAgICAgICA8ZHM6RGlnZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjA
wMS8wNC94bWxlbmMjc2hhMjU2Ii8+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPGRzOkRpZ
2VzdFZhbHVlPlY0VTVxbFozeVhRL1NpMUFDL1I4U0xjM0YraU55Mjd3ZFZlOElXUnFGQVE9PC9kczpEa
Wdlc3RWYWx1ZT4KICAgICAgICAgICAgICAgICAgICAgICAgIDwvZHM6UmVmZXJlbmNlPgogICAgI
CAgICAgICAgICAgICAgPGRzOlJlZmVyZW5jZSBUeXBlPSJodHRwOi8vd3d3LnczLm9yZy
8yMDAwLzA5L3htbGRzaWcjU2lnbmF0dXJlUHJvcGVydGllcyIgVVJJPSIjeGFkZXNTaWduZWRQcm9wZ
XJ0aWVzIj4KICAgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6RGlnZXN0TWV0aG9kIEFsZ29
yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+CiAgICAgICAgI
CAgICAgICAgICAgICAgPGRzOkRpZ2VzdFZhbHVlPk9EUXdOVGcxTlRCaE1qTXpNMll4WTJa
a1pqVmtZemRsTlRaaVpqWTBPREpqTWpOa1lXSTRNVFV6TmpkbU5EVmpNakF3WlRCak9EYzJZVE5o
TVdRMU5nPT08L2RzOkRpZ2VzdFZhbHVlPgogICAgICAgICAgICAgICAgICAgICAgPC9kczpSZ
WZlcmVuY2U+CiAgICAgICAgICAgICAgICAgIDwvZHM6U2lnbmVkSW5mbz4KICAgICAgICAgIC
AgICAgICAgPGRzOlNpZ25hdHVyZVZhbHVlPk1FVUNJQnh5UjhyYzRLODcyOHdkU0Y0WFNE
cVBzK3JJTCszVEZoOW0rYU54UVB0U0FpRUE2Y0hhcEl0dnAxM3lNU3U2Nk5iT2cyQ3BvbUh3VVNuW
Uo5aDZ1R1E2NWFZPTwvZHM6U2lnbmF0dXJlVmFsdWU+CiAgICAgICAgICAgICAgICAgIDxkc
zpLZXlJbmZvPgogICAgICAgICAgICAgICAgICAgICAgPGRzOlg1MDlEYXRhPgogICAgICAgICA
gICAgICAgICAgICAgIDxkczpYNTA5Q2VydGlmaWNhdGU+TUlJRDNqQ0NBNFNnQXdJQ
kFnSVRFUUFBT0FQRjkwQWpzL3hjWHdBQkFBQTRBekFLQmdncWhrak9QUVFEQWpCaU1SVXdFd1lL
Q1pJbWlaUHlMR1FCR1JZRmJHOWpZV3d4RXpBUkJnb0praWFKay9Jc1pBRVpGZ05uYjNZeEZ6QVZC
Z29Ka2lhSmsvSXNaQUVaRmdkbGVIUm5ZWHAwTVJzd0dRWURWUVFERXhKUVVscEZTVTVXVDBs
RFJWTkRRVFF0UTBFd0hoY05NalF3TVRFeE1Ea3hPVE13V2hjTk1qa3dNVEE1TURreE9UTXdXakIxTVF
zd0NRWURWUVFHRXdKVFFURW1NQ1FHQTFVRUNoTWRUV0Y0YVcxMWJTQlRjR1ZsWkNCVVpXT
m9JRk4xY0hCc2VTQk1WRVF4RmpBVUJnTlZCQXNURFZKcGVXRmthQ0JDY21GdVkyZ3hKakFrQmd
OVkJBTVRIVlJUVkMwNE9EWTBNekV4TkRVdE16azVPVGs1T1RrNU9UQXdNREF6TUZZd0VBWUhLb
1pJemowQ0FRWUZLNEVFQUFvRFFnQUVvV0NLYTBTYTlGSUVyVE92MHVBa0MxVklLWHhVOW5Qc
HgydmxmNHloTWVqeThjMDJYSmJsRHE3dFB5ZG84bXEwYWhPTW1Obzhnd25pN1h0MUtUOVVlS09D

QWdjd2dnSURNSUd0QmdOVkhSRUVnYVV3Z2FLa2daOHdnWnd4T3pBNUJnTlZCQVFNTWpFdFZGTlV
mREl0VkZOVWZETXRaV1F5TW1ZeFpEZ3RaVFpoTWkweE1URTRMVGxpTlRndFpEbGhPR1l4TVdVME
5EVm1NUjh3SFFZS0NaSW1pWlB5TEdRQkFRd1BNems1T1RrNU9UazVPVEF3TURBek1RMHdDd1lEVl
FRTURBUXhNVEF3TVJFd0R3WURWUVFhREFoU1VsSkVNamt5T1RFYU1CZ0dBMVVFRHd3UlUzVndj
R3g1SUdGamRHbDJhWFJwWlhNd0hRWURWUjBPQkJZRUZFWCtZdm1tdG5Zb0RmOUJHYktvN29jVEt
ZSzFNQjhHQTFVZEl3UVlNQmFBRkp2S3FxTHRtcXdza0lGelZ2cFAyUHhUKzlObk1Ic0dDQ3NHQVFVRk
J3RUJCRzh3YlRCckJnZ3JCZ0VGQlFjd0FvWmZhSFIwY0RvdkwyRnBZVFF1ZW1GMFkyRXVaMjkyTG5
OaEwwTmxjblJGYm5KdmJHd3ZVRkphUlVsdWRtOXBZMlZUUTBFMExtVjRkR2RoZW5RdVoyOTJMbXh
2WTJGc1gxQlNXa1ZKVGxaUFNVTkZVME5CTkMxRFFTZ3hLUzVqY25Rd0RnWURWUjBQQVFIL0JBU
URBZ2VBTUR3R0NTc0dBUVFCZ2pjVkJ3UXZNQzBHSlNzR0FRUUJnamNWQ0lHR3FCMkUwUHNTaH
UyZEpJZk8reG5Ud0ZWbWgvcWxaWVhaaEQ0Q0FXUUNBUkl3SFFZRFZSMGxCQll3RkFZSUt3WUJCU
VVIQXdNR0NDc0dBUVVGQndNQ01DY0dDU3NHQVFRQmdqY1ZDZ1FhTUJnd0NnWUlLd1lCQlFVSEF
3TXdDZ1lJS3dZQkJRVUhBd0l3Q2dZSUtvWkl6ajBFQXdJRFNBQXdSUUloQUxFL2ljaG1uV1hDVUtVYm
NhM3ljaThvcXdhTHZGZEhWalFydmVJOXVxQWJBaUE5aEM0TThqZ01CQURQU3ptZDJ1aVBKQTZnS1I
zTEUwM1U3NWVxYkMvclhBPT08L2RzOlg1MDlDZXJ0aWZpY2F0ZT4KICAgICAgICAgICAgICAgICAgIC
AgICAgICAgIDwvZHM6WDUwOURhdGE+CiAgICAgICAgICAgICAgICAgICAgICAgIDwvZHM6S2V5SW5
mbz4KICAgICAgICAgICAgICAgICAgICAgPGRzOk9iamVjdD4KICAgICAgICAgICAgICAgICAgICAgIC
AgICAgIDx4YWRlczpRdWFsaWZ5aW5nUHJvcGVydGllcyB4bWxuczp4YWRlcz0iaHR0cDovL3VyaS5ldH
NpLm9yZy8wMTkwMy92MS4zLjIjIiBUYXJnZXQ9InNpZ25hdHVyZSI+CiAgICAgICAgICAgICAgICAgI
CAgICAgICAgICAgPHhhZGVzOlNpZ25lZFByb3BlcnRpZXMgSWQ9InhhZGVzU2lnbmVkUHJvcGVydGllc
yI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDx4YWRlczpTaWduZWRTaWduYXR1cm
VQcm9wZXJ0aWVzPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOl
NpZ25pbmdUaW1lPjIwMjQtMDEtMTRUMTA6MjE6NDA8L3hhZGVzOlNpZ25pbmdUaW1lPgogICAgICAgIC
AgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOlNpZ25pbmdDZXJ0aWZpY2F0ZT4KICA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8eGFkZXM6Q2VydD4KICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOkNlcnREaWdlc3Q+CiA
gICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6RGlnZXN0T
WV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+Ci
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8ZHM6RGlnZXN0
VmFsdWU+WkRNd01tSTBNVEUxTnpWak9UVTJOVGs0WXpWbE9EaGhZbUkwT0RVMk5EVXlOVFUy
WVRWaFlqaGhNREZtTjJGallqazFZVEEyT1dRME5qWTJNalE0TlE9PTwvZHM6RGlnZXN0VmFsdWU+Ci
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDwveGFkZXM6Q2VydERp
Z2VzdD4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPHhhZGVzOkl
zc3VlclNlcmlhbD4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
DxkczpYNTA5SXNzdWVyTmFtZT5DTj1QUlpFSU5WT0lDRVNDQTQtQ0EsIERDPWV4dGdhenQsIERDP
WdvdiwgREM9bG9jYWw8L2RzOlg1MDlJc3N1ZXJOYW1lPgogICAgICAgICAgICAgICAgIC
AgICAgICAgICAgICAgICAgPGRzOlg1MDlTZXJpYWxOdW1iZXI+Mzc5MTEyNzQyODMx
MzgwNDcxODM1MjYzOTY5NTg3Mjg3NjYzNTIwNTI4Mzg3PC9kczpYNTA5U2VyaWFsTnVtYmVyPgogIC
AgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8L3hhZGVzOklzc3VlclNlcmlh
bD4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8L3hhZGVzOkNlcnQ+CiA
gICAgICAgICAgICAgICAgICAgICAgICA8L3hhZGVzOlNpZ25pbmdDZXJ0aWZpY
2F0ZT4KICAgICAgICAgICAgICAgICAgICAgICAgPC94YWRlczpTaWduZWRTaWduY
XR1cmVQcm9wZXJ0aWVzPgogICAgICAgICAgICAgICAgICAgICAgICAgIDwveGFkZXM6U2lnb
mVkUHJvcGVydGllcz4KICAgICAgICAgICAgICAgICAgICAgIDwveGFkZXM6UXVhbGlmeWluZ1
Byb3BlcnRpZXM+CiAgICAgICAgICAgICAgICAgIDwvZHM6T2JqZWN0PgogICAgICAgICAg
ICAgICAgIDwvZHM6U2lnbmF0dXJlPgogICAgICAgICAgICAgICAgPC9zYWM6U2lnbmF0dXJlSW5mb3JtY
XRpb24+CiAgICAgICAgIDwvc2lnOlVCTERvY3VtZW50U2lnbmF0dXJlcz4KICAgICAgICA8L2V4dDpF
eHRlbnNpb25Db250ZW50PgogICAgPC9leHQ6VUJMRXh0ZW5zaW9uPgo8L2V4dDpVQkxFeHRlbnNpb2
5zPgogICAgCiAgICA8Y2JjOlByb2ZpbGVJRD5yZXBvcnRpbmc6MS4wPC9jYmM6UHJvZmlsZUlEPgogIC
AgPGNiYzpJRD5TTUUwMDAyMzwvY2JjOklEPgogICAgPGNiYzpVVUlEPjhkNDg3ODE2LTcwYjgtNGFk
ZS1hNjE4LTlkNjIwYjczODE0YTwvY2JjOlVVSUQ+CiAgICA8Y2JjOklzc3VlRGF0ZT4yMDIyLTA5LTA3PC9
jYmM6SXNzdWVEYXRlPgogICAgPGNiYzpJc3N1ZVRpbWU+MTI6MjE6Mjg8L2NiYzpJc3N1ZVRpbWU+C
iAgICA8Y2JjOkludm9pY2VUeXBlQ29kZSBuYW1lPSIwMTAwMDAwIj4zODg8L2NiYzpJbnZvaWNlVHlwZU
NvZGU+CiAgICA8Y2JjOkRvY3VtZW50Q3VycmVuY3lDb2RlPlNBUjwvY2JjOkRvY3VtZW50Q3VycmVuY
3lDb2RlPgogICAgPGNiYzpUYXhDdXJyZW5jeUNvZGU+U0FSPC9jYmM6VGF4Q3VycmVuY3lDb2RlPgo
gICAgPGNhYzpBZGRpdGlvbmFsRG9jdW1lbnRSZWZlcmVuY2U+CiAgICAgICAgPGNiYzpJRD5JQ1Y8L
2NiYzpJRD4KICAgICAgICA8Y2JjOlVVSUQ+MjM8L2NiYzpVVUlEPgogICAgPC9jYWM6QWRkaXRpb25h
bERvY3VtZW50UmVmZXJlbmNlPgogICAgPGNhYzpBZGRpdGlvbmFsRG9jdW1lbnRSZWZlcmVuY2U+Ci
AgICAgICAgPGNiYzpJRD5QSUg8L2NiYzpJRD4KICAgICAgICA8Y2FjOkF0dGFjaG1lbnQ+CiAgICAgICA
gICAgIDxjYmM6RW1iZWRkZWREb2N1bWVudEJpbmFyeU9iamVjdCBtaW1lQ29kZT0idGV4dC9wbGFpbiI
+TldabFkyVmlOalptWm1NNE5tWXpPR1E1TlRJM09EWmpObVEyT1Raak56bGpNbVJpWXpJek9XUmtO
R1U1TVdJME5qY3lPV1EzTTJFeU4yWmlOVGRsT1E9PTwvY2JjOkVtYmVkZGVkRG9jdW1lbnRCaW5hc
nlPYmplY3Q+CiAgICAgICAgPC9jYWM6QXR0YWNobWVudD4KICAgIDwvY2FjOkFkZGl0aW9uYWxEb2
N1bWVudFJlZmVyZW5jZT4KICAgIAogICAgCiAgICA8Y2FjOkFkZGl0aW9uYWxEb2N1bWVudFJlZmVyZ
W5jZT4KICAgICAgICA8Y2JjOklEPlFSPC9jYmM6SUQ+CiAgICAgICAgPGNhYzpBdHRhY2htZW50PgogI
CAgICAgICA8Y2JjOkVtYmVkZGVkRG9jdW1lbnRCaW5hcnlPYmplY3QgbWltZUNvZGU9InRleHQvc
GxhaW4iPkFXL1l0Tml4MllQWXFTRFlxdG1JMkxIWml0aXZJTmluMllUWXF0bUQyWWJaaU5tRTJZallyTm
1LMktjZzJLallvOW1DMkxYWmlTRFlzOWl4MkxuWXFTRFlwOW1FMllYWXJkaXYyWWpZcjlpcElId2dUV0
Y0YVcxMWJTQlRjR1ZsWkNCVVpXTm9JRk4xY0hCc2VTQk1WRVFDRHpNNU9UazVPVGs1T1Rrd01E
QXdNd01UTWpBeU1pMHdPUzB3TjFReE1qb3lNVG95T0FRRU5DNDJNQVVETUM0MkJpeG1LekJYUT
NGdVVHdEpia2tyWlV3NVJ6Tk1RWEo1TVRKbVZGQm1LM1J2UXpsVldEQTNSalJtU1N0elBRZGdUVVZ
WUTBsQ2VIbFNPSEpqTkVzNE56STRkMlJUUmpSWVUwUnhVSE1yY2tsTUt6TlVSbWc1YlN0aFRuaFJV

SFJUUVdsRlFUWmpTR0Z3U1hSMmNERXplVTFUZFRZMlRtSlBaekpEY0c5dFNIZFZVMjVaU2psb05uVk
hVVFkxWVZrOUNGZ3dWakFRQmdjcWhrak9QUUlCQmdVcmdRUUFDZ05DQUFTaFlJcHJSSnIwVWdT
dE02L1M0Q1FMVlVncGZGVDJjK25IYStWL2pLRXg2UEx4elRaY2x1VU9ydTAvSjJqeWFyUnFFNHlZMmp
5RENlTHRlM1VwUDFSNDwvY2JjOkVtYmVkZGVkRG9jdW1lbnRCaW5hcnlPYmplY3Q+CiAgICAgICAgP
C9jYWM6QXR0YWNobWVudD4KPC9jYWM6QWRkaXRpb25hbERvY3VtZW50UmVmZXJlbmNlPjxjYWM
6U2lnbmF0dXJlPgogICAgICA8Y2JjOklEPnVybjpvYXNpczpuYW1lczpzcGVjaWZpY2F0aW9uOnVibDpza
WduYXR1cmU6SW52b2ljZTwvY2JjOklEPgogICAgICA8Y2JjOlNpZ25hdHVyZU1ldGhvZD51cm46b2FzaX
M6bmFtZXM6c3BlY2lmaWNhdGlvbjp1Ymw6ZHNpZzplbnZlbG9wZWQ6eGFkZXM8L2NiYzpTaWduYXR1c
mVNZXRob2Q+CjwvY2FjOlNpZ25hdHVyZT48Y2FjOkFjY291bnRpbmdTdXBwbGllclBhcnR5PgogICAgICA
gIDxjYWM6UGFydHk+CiAgICAgICAgICAgIDxjYWM6UGFydHlJZGVudGlmaWNhdGlvbj4KICAgICAgICAg
ICAgICAgIDxjYmM6SUQgc2NoZW1lSUQ9IkNSTiI+MTAxMDAxMDAwMDwvY2JjOklEPgogICAgICAgICA
gICA8L2NhYzpQYXJ0eUlkZW50aWZpY2F0aW9uPgogICAgICAgICAgICA8Y2FjOlBvc3RhbEFkZHJlc3M+
CiAgICAgICAgICAgICAgICA8Y2JjOlN0cmVldE5hbWU+2KfZhNin2YXZitixINiz2YTYt9in2YYgfCBQcmluY2
UgU3VsdGFuPC9jYmM6U3RyZWV0TmFtZT4KICAgICAgICAgICAgICAgIDxjYmM6QnVpbGRpbmdOdW1i
ZXI+MjMyMjwvY2JjOkJ1aWxkaW5nTnVtYmVyPgogICAgICAgICAgICAgICAgPGNiYzpDaXR5U3ViZGl2a
XNpb25OYW1lPtin2YTZhdix2KjYuSB8IEFsLU11cmFiYmE8L2NiYzpDaXR5U3ViZGl2aXNpb25OYW1lPgo
gICAgICAgICAgICAgICAgPGNiYzpDaXR5TmFtZT7Yp9mE2LHZitin2LYgfCBSaXlhZGg8L2NiYzpDaXR5T
mFtZT4KICAgICAgICAgICAgICAgIDxjYmM6UG9zdGFsWm9uZT4yMzMzMzwvY2JjOlBvc3RhbFpvbmU+
CiAgICAgICAgICAgICAgICA8Y2FjOkNvdW50cnk+CiAgICAgICAgICAgICAgICAgICAgPGNiYzpJZGVudGl
maWNhdGlvbkNvZGU+U0E8L2NiYzpJZGVudGlmaWNhdGlvbkNvZGU+CiAgICAgICAgICAgICAgICA8L2
NhYzpDb3VudHJ5PgogICAgICAgICAgICA8L2NhYzpQb3N0YWxBZGRyZXNzPgogICAgICAgICAgICA8Y
2FjOlBhcnR5VGF4U2NoZW1lPgogICAgICAgICAgICAgICAgPGNiYzpDb21wYW55SUQ+Mzk5OTk5OTk5
OTAwMDAzPC9jYmM6Q29tcGFueUlEPgogICAgICAgICAgICAgICAgPGNhYzpUYXhTY2hlbWU+CiAgICA
gICAgICAgICAgICAgICAgPGNiYzpJRD5WQVQ8L2NiYzpJRD4KICAgICAgICAgICAgICAgIDwvY2FjOlRh
eFNjaGVtZT4KICAgICAgICAgICAgPC9jYWM6UGFydHlUYXhTY2hlbWU+CiAgICAgICAgICAgIDxjYWM6
UGFydHlMZWdhbEVudGl0eT4KICAgICAgICAgICAgICAgIDxjYmM6UmVnaXN0cmF0aW9uTmFtZT7YtNix
2YPYqSDYqtmI2LHZitivINin2YTYqtmD2YbZiNmE2YjYrNmK2Kcg2KjYo9mC2LXZiSDYs9ix2LnYqSDYp9m
E2YXYrdiv2YjYr9ipIHwgTWF4aW11bSBTcGVlZCBUZWNoIFN1cHBseSBMVEQ8L2NiYzpSZWdpc3RyY
XRpb25OYW1lPgogICAgICAgICAgICA8L2NhYzpQYXJ0eUxlZ2FsRW50aXR5PgogICAgICAgIDwvY2FjOl
BhcnR5PgogICAgPC9jYWM6QWNjb3VudGluZ1N1cHBsaWVyUGFydHk+CiAgICAgPGNhYzpBY2NvdW5
0aW5nQ3VzdG9tZXJQYXJ0eT4KICAgICAgICA8Y2FjOlBhcnR5PgogICAgICAgICAgICA8Y2FjOlBvc3Rhb
EFkZHJlc3M+CiAgICAgICAgICAgICAgICA8Y2JjOlN0cmVldE5hbWU+2LXZhNin2K0g2KfZhNiv2YrZhiB8IF
NhbGFoIEFsLURpbjwvY2JjOlN0cmVldE5hbWU+CiAgICAgICAgICAgICAgICA8Y2JjOkJ1aWxkaW5nTnVt
YmVyPjExMTE8L2NiYzpCdWlsZGluZ051bWJlcj4KICAgICAgICAgICAgICAgIDxjYmM6Q2l0eVN1YmRpdm
lzaW9uTmFtZT7Yp9mE2YXYsdmI2KwgfCBBbC1NdXJvb2o8L2NiYzpDaXR5U3ViZGl2aXNpb25OYW1lPg
ogICAgICAgICAgICAgICAgPGNiYzpDaXR5TmFtZT7Yp9mE2LHZitin2LYgfCBSaXlhZGg8L2NiYzpDaXR5
TmFtZT4KICAgICAgICAgICAgICAgIDxjYmM6UG9zdGFsWm9uZT4xMjIyMjwvY2JjOlBvc3RhbFpvbmU+Ci
AgICAgICAgICAgICAgICA8Y2FjOkNvdW50cnk+CiAgICAgICAgICAgICAgICAgICAgPGNiYzpJZGVudGlm
aWNhdGlvbkNvZGU+U0E8L2NiYzpJZGVudGlmaWNhdGlvbkNvZGU+CiAgICAgICAgICAgICAgICA8L2N
hYzpDb3VudHJ5PgogICAgICAgICAgICA8L2NhYzpQb3N0YWxBZGRyZXNzPgogICAgICAgICAgICA8Y2
FjOlBhcnR5VGF4U2NoZW1lPgogICAgICAgICAgICAgICAgPGNiYzpDb21wYW55SUQ+Mzk5OTk5OTk5
ODAwMDAzPC9jYmM6Q29tcGFueUlEPgogICAgICAgICAgICAgICAgPGNhYzpUYXhTY2hlbWU+CiAgIC
AgICAgICAgICAgICAgICAgPGNiYzpJRD5WQVQ8L2NiYzpJRD4KICAgICAgICAgICAgICAgIDwvY2FjOlR
heFNjaGVtZT4KICAgICAgICAgICAgPC9jYWM6UGFydHlUYXhTY2hlbWU+CiAgICAgICAgICAgIDxjYWM6
UGFydHlMZWdhbEVudGl0eT4KICAgICAgICAgICAgICAgIDxjYmM6UmVnaXN0cmF0aW9uTmFtZT7YtNix
2YPYqSDZhtmF2KfYsNisINmB2KfYqtmI2LHYqSDYp9mE2YXYrdiv2YjYr9ipIHwgRmF0b29yYSBTYW1wb
GVzIExURDwvY2JjOlJlZ2lzdHJhdGlvbk5hbWU+CiAgICAgICAgICAgIDwvY2FjOlBhcnR5TGVnYWxFbnR
pdHk+CiAgICAgICAgPC9jYWM6UGFydHk+CiAgICA8L2NhYzpBY2NvdW50aW5nQ3VzdG9tZXJQYXJ0e
T4KICAgIDxjYWM6RGVsaXZlcnk+CiAgICAgICAgPGNiYzpBY3R1YWxEZWxpdmVyeURhdGU+MjAyMi0
wOS0wNzwvY2JjOkFjdHVhbERlbGl2ZXJ5RGF0ZT4KICAgIDwvY2FjOkRlbGl2ZXJ5PgogICAgPGNhYzp
QYXltZW50TWVhbnM+CiAgICAgICAgPGNiYzpQYXltZW50TWVhbnNDb2RlPjEwPC9jYmM6UGF5bWVud
E1lYW5zQ29kZT4KICAgIDwvY2FjOlBheW1lbnRNZWFucz4KICAgIDxjYWM6QWxsb3dhbmNlQ2hhcmdlP
gogICAgICAgIDxjYmM6Q2hhcmdlSW5kaWNhdG9yPmZhbHNlPC9jYmM6Q2hhcmdlSW5kaWNhdG9yPgo
gICAgICAgIDxjYmM6QWxsb3dhbmNlQ2hhcmdlUmVhc29uPmRpc2NvdW50PC9jYmM6QWxsb3dhbmNlQ
2hhcmdlUmVhc29uPgogICAgICAgIDxjYmM6QW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+MC4wMDwvY2JjO
kFtb3VudD4KICAgICAgICA8Y2FjOlRheENhdGVnb3J5PgogICAgICAgICAgICA8Y2JjOklEIHNjaGVtZUlEP
SJVTi9FQ0UgNTMwNSIgc2NoZW1lQWdlbmN5SUQ9IjYiPlM8L2NiYzpJRD4KICAgICAgICAgICAgPGNiY
zpQZXJjZW50PjE1PC9jYmM6UGVyY2VudD4KICAgICAgICAgICAgPGNhYzpUYXhTY2hlbWU+CiAgICAg
ICAgICAgICAgICA8Y2JjOklEIHNjaGVtZUlEPSJVTi9FQ0UgNTE1MyIgc2NoZW1lQWdlbmN5SUQ9IjYiPlZ
BVDwvY2JjOklEPgogICAgICAgICAgICA8L2NhYzpUYXhTY2hlbWU+CiAgICAgICAgPC9jYWM6VGF4Q2F
0ZWdvcnk+CiAgICA8L2NhYzpBbGxvd2FuY2VDaGFyZ2U+CiAgICA8Y2FjOlRheFRvdGFsPgogICAgICAg
IDxjYmM6VGF4QW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+MC42PC9jYmM6VGF4QW1vdW50PgogICAgP
C9jYWM6VGF4VG90YWw+CiAgICA8Y2FjOlRheFRvdGFsPgogICAgICAgIDxjYmM6VGF4QW1vdW50IG
N1cnJlbmN5SUQ9IlNBUiI+MC42PC9jYmM6VGF4QW1vdW50PgogICAgICAgIDxjYWM6VGF4U3VidG90Y
Ww+CiAgICAgICAgICAgIDxjYmM6VGF4YWJsZUFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjQuMDA8L2NiYz
pUYXhhYmxlQW1vdW50PgogICAgICAgICAgICA8Y2JjOlRheEFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjAu
NjA8L2NiYzpUYXhBbW91bnQ+CiAgICAgICAgICAgICA8Y2FjOlRheENhdGVnb3J5PgogICAgICAgICAgICAgIC
AgICAgIDxjYmM6SUQgc2NoZW1lSUQ9IlVOL0VDRSA1MzA1IiBzY2hlbWVBZ2VuY3lJRD0iNiI+UzwvY2Jj
OklEPgogICAgICAgICAgICAgICAgIDxjYmM6UGVyY2VudD4xNS4wMDwvY2JjOlBlcmNlbnQ+CiAgICAgI
CAgICAgICAgICA8Y2FjOlRheFNjaGVtZT4KICAgICAgICAgICAgICAgIDxjYmM6SUQgc2NoZW1lSU
Q9IlVOL0VDRSA1MTUzIiBzY2hlbWVBZ2VuY3lJRD0iNiI+VkFUPC9jYmM6SUQ+CiAgICAgICAgICAgICA
gICA8L2NhYzpUYXhTY2hlbWU+CiAgICAgICAgICA8L2NhYzpUYXhDYXRlZ29yeT4KICAgICAgICA8

*L2NhYzpUYXhTdWJ0b3RhbD4KICAgIDwvY2FjOlRheFRvdGFsPgogICAgPGNhYzpMZWdhbE1vbmV0YX
J5VG90YWw+CiAgICAgICAgPGNiYzpMaW5lRXh0ZW5zaW9uQW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+
NC4wMDwvY2JjOkxpbmVFeHRlbnNpb25BbW91bnQ+CiAgICAgICAgPGNiYzpUYXhFeGNsdXNpdmVBb
W91bnQgY3VycmVuY3lJRD0iU0FSIj40LjAwPC9jYmM6VGF4RXhjbHVzaXZlQW1vdW50PgogICAgICAgI
DxjYmM6VGF4SW5jbHVzaXZlQW1vdW50IGN1cnJlbmN5SUQ9IlNBUiI+NC42MDwvY2JjOlRheEluY2x1c
2l2ZUFtb3VudD4KICAgICAgICA8Y2JjOkFsbG93YW5jZVRvdGFsQW1vdW50IGN1cnJlbmN5SUQ9IlNBUi
I+MC4wMDwvY2JjOkFsbG93YW5jZVRvdGFsQW1vdW50PgogICAgICAgIDxjYmM6UHJlcGFpZEFtb3Vu
dCBjdXJyZW5jeUlEPSJTQVIiPjAuMDA8L2NiYzpQcmVwYWlkQW1vdW50PgogICAgICAgIDxjYmM6UGF
5YWJsZUFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjQuNjA8L2NiYzpQYXlhYmxlQW1vdW50PgogICAgPC9jY
WM6TGVnYWxNb25ldGFyeVRvdGFsPgogICAgPGNhYzpJbnZvaWNlTGluZT4KICAgICAgICA8Y2JjOklE
PjE8L2NiYzpJRD4KICAgICAgICA8Y2JjOkludm9pY2VkUXVhbnRpdHkgdW5pdENvZGU9IlBDRSI+Mi4wM
DAwMDA8L2NiYzpJbnZvaWNlZFF1YW50aXR5PgogICAgICAgIDxjYmM6TGluZUV4dGVuc2lvbkFtb3Vud
CBjdXJyZW5jeUlEPSJTQVIiPjQuMDA8L2NiYzpMaW5lRXh0ZW5zaW9uQW1vdW50PgogICAgICAgIDxjY
WM6VGF4VG90YWw+CiAgICAgICAgICAgICA8Y2JjOlRheEFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjAuNjA
8L2NiYzpUYXhBbW91bnQ+CiAgICAgICAgICAgICA8Y2JjOlJvdW5kaW5nQW1vdW50IGN1cnJlbmN5SU
Q9IlNBUiI+NC42MDwvY2JjOlJvdW5kaW5nQW1vdW50PgogICAgICAgIDwvY2FjOlRheFRvdGFsPgogIC
AgICAgIDxjYWM6SXRlbT4KICAgICAgICAgICAgPGNiYzpOYW1lPtmC2YTZhSDYsdi12KfYtTwvY2JjOk5h
bWU+CiAgICAgICAgICAgIDxjYWM6Q2xhc3NpZmllZFRheENhdGVnb3J5PgogICAgICAgICAgICAgICAgP
GNiYzpJRD5TPC9jYmM6SUQ+CiAgICAgICAgICAgICAgICA8Y2JjOlBlcmNlbnQ+MTUuMDA8L2NiYzpQZ
XJjZW50PgogICAgICAgICAgICAgICAgPGNhYzpUYXhTY2hlbWU+CiAgICAgICAgICAgICAgICAgICAgPG
NiYzpJRD5WQVQ8L2NiYzpJRD4KICAgICAgICAgICAgICAgIDwvY2FjOlRheFNjaGVtZT4KICAgICAgICA
gICAgPC9jYWM6Q2xhc3NpZmllZFRheENhdGVnb3J5PgogICAgICAgIDwvY2FjOkl0ZW0+CiAgICAgICAg
PGNhYzpQcmljZT4KICAgICAgICAgICAgPGNiYzpQcmljZUFtb3VudCBjdXJyZW5jeUlEPSJTQVIiPjIuMDA
8L2NiYzpQcmljZUFtb3VudD4KICAgICAgICA8L2NhYzpQcmljZT4KICAgIDwvY2FjOkludm9pY2VMaW5lPg
o8L0ludm9pY2U+" }*

### Request headers

### Return type

InvoiceResultModel

### Example data

Content-Type: application/json

```
{ "invoiceHash" : "invoiceHash", "warnings" : [ { "code" : "code", "category" : "category", "message" : "message" }, { "code" : "code", "category" : "category", "message" : "message" } ], "erros" : [ { "code" : "code", "category" : "category", "message" : "message" }, { "code" : "code", "category" : "category", "message" : "message" } ], "status" : "Reported" }
```

### 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

### Responses

### 200

HTTP OK. Returned on successful validation of simplified invoice. InvoiceResultModel

### Example data

Content-Type: reported

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

### 400

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

### Example data

Content-Type: KSA Rules Violation

```
{"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: Unuthorized

```
{"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

### 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*

# 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

# ComplianceCSIDCertificate

## POST /compliance

Up

Issues an X509 Compliance Cryptographic Stamp Identifier (CCSID/Certificate) (CSID) based on submitted CSR. (**complianceCertificate**)

This is a compliance CSID (CCSID) that is issued by the einvoicing system as it is a prerequisite to complete the compliance steps. The CCSID is sent in the authentication certificate header in the compliance api calls.

The CSR specification required to perform the Compliance API call is covered in section 4.3 of the Developer Portal user manual.

### 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**
Successful response String
**Example data**
Content-Type: application/json

```
{"requestID":1234567890123,"dispositionMessage":"ISSUED","binarySecurityToken":"TUlJQ1BUQ0NBZU9nQXdJQkFnSUd
BWXp6Z0VoTk1Bb0dDQ3FHU000OUJBTUNNQlV4RXpBUkJnTlZCQU1NQ21WSmJuWnZhV05wYm1jd0hoY05NalF3TVR
Fd01UTXhNVFUwV2hjTk1qa3dNVEE1TWpFd01EQXdXaklxTVFzd0NRWURWUVFHRXdKVFFURVdNQlFHQTFVRUN3d0
5VbWw1WVdSb0lFSnlZVzVqYURFbU1DUUdBMVVFQ2d3ZFRXRjRhVzExYlNCVGNHVmxaQ0JVWldOb0lGTjFjSEJzZVN
CTVZFUXhKakFrQmdOVkJBTU1IVlJUVkMwNE9EWTBNekV4TkRVdE16azVPVGs1T1RrNU9UQXdNREF6TUZZd0VBWU
hLb1pJemowQ0FRWUZLNEVFQUFvRFFnQUVvV0NLYTBTYTlGSUVyVE92MHVBa0MxVklLWHhVOW5QcHgydmxmNHlo
TWVqeThjMDJYSmJsRHE3dFB5ZG84bXEwYWhPTW1Obzhnd25pN1h0MUtUOVVlS09Cd1RDQnZqQU1CZ05WSFJNQk
FmOEVBakFBTUlHdEJnTlZIUkVFZ2FVd2dhS2tnWjh3Z1p3eE96QTVCZ05WQkFRTU1qRXRWRk5VZkRJdFZGTlVmRE10
WldReU1tWXhaRGd0WlRaaE1pMHhNVEU0TFRsaU5UZ3RaRGxoT0dZeE1XVTBORFZtTVI4d0hRWUtDWkltaVpQeUxHU
UJBUXdQTXprNU9UazVPVGs1T1RBd01EQXpNUTB3Q3dZRFZRUU1EQVF4TVRBd01SRXdEd1lEVlFRYURBaFNVbEpF
TWpreU9URWFNQmdHQTFVRUR3d1JVM1Z3Y0d4NUlHRmpkR2wyYVhScFpYTXdDZ1lJS29aSXpqMEVBd0lEU0FBd1JR
SWhBSUY4akljeHp2Q3lxVURUcDVPbXY3MlVweFBBTG1vUnl0OURZMjRqV21CUUFpQTBiYVo2WXJwcDV5SjRhaG9vb1
czK09hOGtrYjMxZXZBb0hkdmdEODA2M3c9PQ==","secret":"Dehvg1fc8GF6Jwt5bOxXwC6enR93VxeNEo2mlUatfgw="}
```

**400**
HTTP Bad Request. Returned when the submitted request is invalid. CertificatesErrorsResponse
**Example data**
Content-Type: Invalid OTP

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

**Example data**
Content-Type: Missing CSR

```
{"errors":[{"code":"Missing-CSR","message":"CSR is required field"}]}
```

**Example data**
Content-Type: Invalid CSR

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

**Example data**
Content-Type: Missing OTP

```
{"errors":[{"code":"Missing-OTP","message":"OTP is required field"}]}
```

**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*