



e-Business APIs

Version 2.0.6

25th June 2021

OZEDI Holdings Pty Ltd

ABN 47 167 142 672

Level 2, 1 Queens Road
Melbourne VIC 3004

Tel + 61 1300 737 614

www.OZEDI.com.au



Peppol
Access Point
CERTIFIED PROVIDER

Contents

Document Version Control.....	6
Section A) General	7
Introduction.....	7
Connecting to OZEDI.....	7
Tasks using APIs	7
Test and Production Environments	7
API Version Control	7
Webhooks versus Polling.....	7
Polling	8
Webhooks.....	8
Accessible Ports and Geo-Blocking.....	8
Section B) API Summary and Description	9
1. Authentication.....	9
2. OZEDI Client.....	9
2.1 OZEDI Registration System	9
2.2 Client Webhooks	9
3. Sending Documents.....	10
3.1 Authentication.....	10
3.2 Generate an upload token.....	10
3.3 Upload document	10
4. Receiving documents.....	11
4.1 Authentication.....	11
4.2 Message Receive	11
4.3 Message Received Confirmation	11
4.4 Receive UBL document as PDF	12
4.5 Attachment Download	12
5. Queries	12
5.1 Authentication.....	12
5.2 Query sent document status at OZEDI	12
5.3 Retrieve Sent Document Validation Errors	12
5.4 Query a list of documents received and their status for a client	12
5.5 Message Pending List	13
5.6 Query Account Balance	13
5.7 Participant document type support query (SML/SMP).....	13
6. Webhooks.....	13
6.1 Authentication.....	13

- 6.2 Register a client for webhook notifications..... 13
- 6.3 List..... 13
- 6.4 Update 13
- 6.5 Post 13
- 7. Errors 13
- Section C) API Details and Examples Base URL 14**
 - Test Environment 14
 - Production Environment 14
- 1. Authentication API..... 14
 - API-1 Authenticate API 14
 - Request parameters 14
 - Request example 14
 - Response..... 14
- 2. Registration APIs..... 15
 - API-2.1 Authenticate 15
 - API-2.2 OZEDI Client Registration..... 15
 - Request parameters 15
 - Request example 16
 - Response..... 16
 - API-2.3 OZEDI Client Update 17
 - Request parameters 17
 - Response..... 18
 - API-2.4 OZEDI Client disable/delete 18
 - Request parameters 18
 - Response..... 18
- 3. Sending APIs 19
 - API-3.1. Authenticate 19
 - API-3.2 Message Upload Token..... 19
 - Request parameters 19
 - Request example 19
 - Response..... 19
 - API-3.3 Message Send 20
 - Request parameters 20
 - Response..... 20
- 4. Receiving APIs..... 21
 - API-4.1 Authenticate 21
 - API-4.2 Message Receive..... 21

- Request parameters 21
- Response parameters..... 21
- Response..... 21
- API-4.3 Message Receive Confirmation 21
 - Request parameters 21
 - Response..... 22
- Business Level Responses..... 22
 - Request example 22
- API-4.4 Receive UBL document as PDF..... 24
 - Request parameters 24
 - Response parameters..... 24
 - Response..... 24
- API-4.5 Attachment Download..... 24
 - Request parameters 24
 - Response parameters..... 25
 - Response..... 25
- 5. Query APIs 26
 - API-5.1. Authenticate 26
 - API-5.2 Message Status 26
 - Request parameters 26
 - Response..... 26
 - Message status names (messageStatusName): 27
 - API-5.3 Validation Errors 27
 - Request parameters 27
 - Response..... 28
 - Request parameters 29
 - Request example 29
 - Response..... 30
 - API-5.5 Message Pending List..... 31
 - By Client..... 31
 - Request parameters 31
 - Response..... 31
 - By Account 32
 - Request parameters 32
 - Response..... 32
 - 5.6 Query Account Balance 33
 - Request parameters 33

- Response..... 33
- 5.7 Participant document type support query (SML/SMP) 34
 - Request parameters 34
 - Response example 34
 - Response..... 34
- 6. Webhooks..... 35
 - API-6.1. Authenticate 35
 - API-6.2 Register Webhooks 35
 - Request parameters 35
 - Request example 35
 - Response..... 36
 - Processing Errors 36
 - API-6.3 Webhook - List 36
 - Request parameters 36
 - Response..... 36
 - Response Example 37
 - API-6.4 Webhook - Update..... 37
 - Request parameters 37
 - Request example 37
 - Response..... 38
- Error Examples..... 38
 - Input parameters validation error example 38
 - Processing error example 38
- Appendix A: Document Type Specifications 39**

Document Version Control

<i>Version Number</i>	<i>Release Date</i>	<i>Who</i>	<i>Sections Affected</i>	<i>Changes</i>
2.0.0	16/09/20	Dev/AW	All	First version of production document, using v0.4.6 as starting point. Refers to v2 in base URL.
2.0.1	2/10/20	AW	API 5.2	Slash missing from GET snd method
2.0.2	14/10/20	AK	API 5.4	Message Status added to the filter parameters
2.0.2.2	10/11/20	ML	API 5.6	Updated for final release. Included balance API.
2.0.2.3	11/11/20	JB	API 6.4	Second usage changed to rcv
2.0.4	06/01/21	AK	API 4.3	Message Receive Confirmation optional status update request which generates Document Response.
	4/01/21	CB	API 5.5	documentDetails and documentCategoryName added to message details response.
			All	Change invoice references in text content to be document to cover all possible document types.
2.0.5	23/03/21	AK	API 2.3 API 2.4 API 4.4 API 5.7	Client update Client disable/delete Receive UBL document as PDF added Peppol SML/SMP query added
	19/04/21	ML	B1, B1 General B4.3 B5 Appendix A	Information included on: <ul style="list-style-type: none"> • Duration of tokens • Recommendations for re-authentication • Accessible ports and geo-blocking • Business Level Response explanation • Default query size • Document Type specifications
2.0.5.1	15/05/21	ML	C4.3	Business level responses explanation updated.
2.0.6	9/06/21	ML	C4.4 B&C 4.5 Appendix A B&C 3.2	Added Client id to path and parameter for API. Added Attachment Download API. Added AU/NZ Self-Billing Invoice and Credit Note Generate upload token usage with document type enhanced.

Section A) General

Introduction

REST APIs are required by applications interfacing seamlessly to OZEDI for e-Invoicing; this includes sending and receiving documents as well as registration and account query functions.

Connecting to OZEDI

To use REST API to connect to OZEDI, you will need to be registered as an OZEDI user and create an account on the OZEDI portal. Refer to Registration User Guide for details. You will be required to provide these registered user credentials:

- user name and password
- account number – users can be connected to more than one account
- client ID – for sending and receiving documents, query lists and webhook APIs (not required for Registration)

Tasks using APIs

The following tasks have detailed REST APIs to perform them:

1. **Authentication** to validate the user credentials against the OZEDI Registration system.
2. **Registering a new Client** for e-invoicing against an existing account – optional as this process can be performed manually on the Registration portal.
3. **Sending documents**
4. **Receiving documents**
5. **Queries** for documents sent and received to retrieve specified data
6. **Webhooks** to receive data updates

Test and Production Environments

OZEDI provides both test and production environments for e-invoicing. The APIs use different base URLs for each environment.

API Version Control

The API call base URL has a version number embedded to allow multiple versions to be used and maintained concurrently.

Webhooks versus Polling

Some tasks require a follow up call to retrieve data related to the initial message. For example, sending a document must be followed up by a request for the document status. This is required because a document may fail XML validation and have errors. Or the document may not have been sent to the receiver because the receiver's access point is down or has rejected it or OZEDI processing has paused or any number of reasons.

There are two ways to retrieve the document status at any time:

- Polling
- Webhooks

Polling

When a document has been sent to OZEDI, a record for the Message ID can be added to a queue which is then used to drive a request for the document status on a regular interval until a response is received. This method requires continually asking for the document status until you get it back. Then you stop asking.

Webhooks

An alternative is to register with OZEDI that you want to receive notification when a document status changes. This is performed by OZEDI sending a webhook to the address registered on the Client ID. The receiving system must be online 24/7 to receive webhooks sent by OZEDI. Your software can action the webhooks received by creating a document response API call; thereby only asking for document status once.

Webhooks can have separate addresses (URLs) for sending document processes and receiving document processes. The webhook contains the Message ID, enabling your software to match the webhook to the originating document and to the response pulled.

The following webhooks will be sent if registered:

- Sending documents – document status is available after upload and changes with XML validation and sending to receiver as well as response back from recipient
- Receiving documents – a document has been received and is awaiting download.

Accessible Ports and Geo-Blocking

OZEDI supports sending/receiving only via standard ports for HTTP/HTTPS. i.e. 80/443.

Network traffic coming from outside Australia and New Zealand is geo-blocked. If you require this, please contact us to arrange whitelisting of specific ports.

Section B) API Summary and Description

1. Authentication

Every interaction with our API endpoints will require authentication using the user name and password credentials as per the OZEDI Registration system. Successful authentication will return a token, which is required in the header of each subsequent API call.

The authentication token is valid for 24 hours. It is recommended that the authentication is only refreshed daily or when it expires. We recommend that you attempt to reuse the existing authentication, and if it has expired then re-authenticate. Re-authentication should **not** be done for every transaction.

2. OZEDI Client

2.1 OZEDI Registration System

Sending and receiving documents is performed by a Client registered under an OZEDI account. A Client is an organisation that has a Peppol participant ID and that belongs to the OZEDI account. In Production, you don't register your customers or suppliers – they must register on the Peppol network themselves. When using the OZEDI test environment, you may choose to have two Clients; one for sending and one for receiving – you will need to register both as the test environment is internal only to OZEDI – it does not connect to other Access Points for testing.

Registering a client against an existing account can be done manually via the OZEDI registration system as well as via an API embedded in your application. The following are created/updated:

- Peppol SML - locator
- OZEDI's SMP - publisher
- A new Client is added to the OZEDI account.

The details require the participant ID of the Client; this has two parts:

- Participant type; eg 0151 for ABN, 0088 for NZBN, 0088 for GS1 number (same as NZBN), 0060 for Dun's number (Dun and Bradstreet)
- Participant number – depends on the participant type

Participant IDs can only exist once on the SML – this is a central registry and it will point to the SMP which holds details of that Participant. Receivers of documents MUST be on the SML/SMP; senders of documents should also be registered because when document responses are implemented they will want to receive them.

Refer to API instructions below for details and this link for Peppol participant technical information:

https://peppol.helger.com/public/locale-en_US/menuitem-tools-participant

You can manually check Peppol Participant registration at:

https://test-directory.peppol.eu/public/locale-en_US/menuitem-search

2.2 Client Webhooks

An OZEDI client can have webhooks registered to report on any changes to document status; one webhook for sent documents, another for documents received.

3. Sending Documents

Sending documents to OZEDI is a 3-step process:

3.1 Authentication

A valid authentication token is required. See “1. Authentication” above.

3.2 Generate an upload token

Prior to sending a document to OZEDI, the application is required to generate a valid upload token. To receive an upload token the receiver’s e-Invoicing participant identifier (usually the ABN in Australia or NZBN in New Zealand) must be currently registered on the SML. This query also looks up the SMP (publisher) for the receiver to check if the document type being sent is accepted by the receiver. Your application must be able to create UBL in the receiver’s defined version to be accepted by OZEDI. OZEDI will only accept documents for receivers that are registered on the Peppol network and can receive the required document type and version.

The upload token is valid for 24 hours. The upload token is valid only for single usage of an upload or upload attempt.

API will return the following results:

- SML look up fail – not found on Locator
- SMP look up fail – receiver’s Access Point not found or document UBL does not match receiver’s accepted versions
- OZEDI account check fail – overdrawn
- If successful, a valid upload token

3.3 Upload document

API call will push a document in UBL format to OZEDI for on-forwarding.

OZEDI will validate the UBL and if errors are found they can be retrieved by your application.

API will return the following results:

- OZEDI credentials check failed
- Token not allocated or expired
- If all the above are ok, successful or unsuccessful upload
- If successful upload, a Message ID is allocated and provided in the response

Once the document has been successfully uploaded it will be validated prior to on-forwarding to the receiver. If the document fails the validation, it will not be forwarded and errors will be available for retrieval. Retrieving errors and message status is detailed under API Queries below.

Refer to API instructions below for details of each step.

Document status

Once a document has been uploaded to OZEDI, it has the following possible status settings. If webhooks for sending documents are registered against the Client, then this status change will result in a webhook sent to the sender location. Alternatively, the sender can call the API for document status. Refer to 5 Queries below.

Status Name	Description
SEND_UPLOAD_TOKEN	Token has been retrieved – pending final upload
SEND_IN_PROGRESS	Sending is in progress
SEND_SUCCESS	Sending completed successfully
SEND_FAILED	Sending failed – a non-validation error occurred (<i>failure condition retrieval API pending</i>)
SEND_FAILED_VALIDATION	Sending failed – validation error (<i>failure condition retrieval API pending</i>)

4. Receiving documents

Your application will need to use your client’s specific credentials with the OZEDI REST API for downloading a list of documents and their Message IDs according to desired filter options; eg all whitelisted supplier documents that haven’t been downloaded before. These Message UUIDs can be received via webhook or by getting list of filtered messages. Refer to API instructions in the API Queries and/or Webhooks sections below for details.

Downloading documents using the Message UUID is a 3-step process:

4.1 Authentication

A valid authentication token is required. See “1. Authentication” above.

4.2 Message Receive

Download document received by OZEDI via Peppol network for particular client and Message UUID. This API will return the UBL XML document or an error response.

The message UUID for download is obtained by running one of the query APIs. Eg. The Message Pending List.

4.3 Message Received Confirmation

Clients should acknowledge successful receipt of payload by calling this endpoint. Message status will change to RECEIVE_DELIVERED.

Refer to API instructions below for details of each step.

Document Status

Once a document has been received at OZEDI, it has the following possible status settings. If webhooks for receiving documents are registered against the Client, then this status change will result in a webhook sent to the receiver location. Alternatively, the receiver can call the API for document status. Refer to 5 Queries below.

Status Name	Description
RECEIVE_IN_PROGRESS	Document received and it is processing
RECEIVE_READY	Document successfully processed and available for download
RECEIVE_FAILED	Receive process been failed/interrupted and cause should be investigated
RECEIVE_FAILED_VALIDATION	Document been successfully received and processed, but failed validation
RECEIVE_DELIVERED	Status indicate document been downloaded by the client

4.4 Receive UBL document as PDF

XML invoice can be converted to PDF by OZEDI on request and downloaded via API. PDF layout is designed to suit most of the clients and customisation is possible on request.

4.5 Attachment Download

Extract and download embedded attachments from a document sent or received by OZEDI via the Peppol network for a particular client. The required Message UUID can be retrieved via webhook or by getting list of filtered messages as described under Query APIs and Webhooks below.

5. Queries

There are a number of Query/List APIs that are used to retrieve data. Some relate to the sending of documents, others are connected to documents received.

Note that by default the number of records returned by query APIs is 20. If more records are required to be returned, this can be done by specifying the “size” parameter.

5.1 Authentication

A valid authentication token is required. See “1. Authentication” above.

5.2 Query sent document status at OZEDI

Applications can query the status of a document by API using the document Message UUID – this status can then be attached to your application transaction for queries.

By referencing the document Message ID, an API may request the document status which will return one of the following status’:

- This message ID not known at OZEDI
- Document has validation errors and was not sent
- Document is error-free but has not been forwarded to customer yet
- Document has been forwarded to customer (their access point or OZEDI receiver pick up point)
- Document has been downloaded by customer (only available if receiver is also registered with OZEDI as access point)

5.3 Retrieve Sent Document Validation Errors

If errors are detected in the document validation process, they can be retrieved from OZEDI by the client using an API call by Message ID. This call can be initiated in the following ways:

- Webhook sent by OZEDI to the client triggering an API call to pull the document status and error list
- Client uses API to poll OZEDI by Message ID to pull the document status and error list

NB document errors can also be emailed to clients in PDF format.

5.4 Query a list of documents received and their status for a client

Additional query option to return a list of documents and their current status with options to filter the list including:

- Sender (optional)
- Message type (invoice, purchase order etc)
- Classification (whitelisted suppliers, blacklisted suppliers, not classified) (future)
- Date range
- Message status

5.5 Message Pending List

List of messages ready to be received can be obtained using this endpoint. It is recommended to use this method instead of Message Filtered List when implementing regular polling for new, not yet received documents e.g. documents with status RECEIVE_READY.

5.6 Query Account Balance

Query to show the account balance for an account in dollars.

5.7 Participant document type support query (SML/SMP)

Query to check if given participant is registered on Peppol network and which document types it supports.

6. Webhooks

APIs can place calls to OZEDI without knowing whether they will get any updated data as a response or not (polling), whereas webhooks receive calls through HTTP POSTs from external systems only when there are data updates. The webhook destination is held against the Client and you can have a separate target URL for the status of documents sent and for documents received.

API's are used to maintain the webhook details against the OZEDI Client:

6.1 Authentication

As with queries, APIs that maintain webhooks require authentication and resulting token. See "1. Authentication" above.

6.2 Register a client for webhook notifications

Send a request to register a webhook URL for message related events for a nominated ClientID. Separate URLs can be held for sending document messages and receiving document messaging.

6.3 List

Retrieve a list of all registered webhooks for a ClientID.

6.4 Update

Update an existing webhook. For example, if a webhook needs to be disabled or the target URL is changed.

6.5 Post

After the webhook is registered with OZEDI via the Create, a POST request will be sent to the specified `targetUrl` each time an event with type and triggerType defined in the webhook takes place. The content of this request depends on the webhook type; eg message status. Refer to the message status settings that trigger webhooks for sending documents and receiving documents above.

7. Errors

Unsuccessful calls will return an error code. For example, if the user and password authentication fails then a 401 – "Unauthorised" error is returned. For data validation errors, you can get a response back with further details using the [Validation Errors API](#) detailed under the Query APIs section. There are examples under the [Error Examples](#) section at the end of the document showing how they will be presented for a call like self-registration or a backend error related to processing.

Section C) API Details and Examples

Base URL

Test Environment

Base URL: <https://api-ebusiness.sandpit.OZEDI.com.au/v2/>

The version number is embedded in the base URL, which means that multiple versions can be supported concurrently.

Production Environment

Base URL: <https://api-ebusiness.OZEDI.com.au/v2/>

1. Authentication API

API-1 Authenticate API

Authenticate and obtain JWT token.

Method	Operation
POST	/authenticate

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
BODY (JSON)	username	<API_username>	OZEDI username
	password	<API_password>	OZEDI password

Request example

```
{
  "username": "api@abc_software.com.au",
  "password": "secret!pass"
}
```

Response

Status	Response
200	{ "id_token": "eyJhbGciOiJIUzUxMiB... (removed for brevity) ... CRXaQ" }
401	Unauthorized (BODY: TBD)
403	Forbidden (BODY: TBD)
500	System error (BODY: TBD)
503	Service unavailable (Maintenance mode) (BODY: TBD)

2. Registration APIs

API-2.1 Authenticate

Refer to API-1 above

API-2.2 OZEDI Client Registration

Register new OZEDI client to existing account

Method	Operation
POST	/client/self-register

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token from Authenticate call
BODY (JSON)	account	<account>	OZEDI account number – 15 digit number
	participant	<participant>	Peppol participant id
	companyName	<companyName>	Participant company name
	companyInfo	<companyInfo>	Optional. Company description
	countryCode	<countryCode>	Two letters ISO 3166-1 country code
	geographicalInfo	<geographicalInfo>	Optional. Business address string
	businessIdentifiers[]	businessIdentifiers	ONE or more business identifiers
businessIdentifiers element	identifierType	<identifierType>	Peppol network participant identifier type: "ABN", "NZBN", "GS1", "DUNS" ...
businessIdentifiers element	identifier	<identifier>	Identifier value
	contacts[]	contacts	ONE or more contacts
contacts element	contactEmail	<contactEmail>	Contact email. Used for OZEDI client as well as for Peppol directory
contacts element	contactType	<contactType>	Optional. eg "Reception"
contacts element	contactName	<contactName>	Optional.
contacts element	contactPhoneNumber	<contactPhoneNumber>	Optional.
	documentIdentifiers[]	documentIdentifiers	ONE or more supported document identifiers
documentIdentifiers element	documentType	<documentType>	Peppol document type Refer to Appendix A for the specification of the documentType.
documentIdentifiers element	process	<process>	Peppol document process identifier
	level	<level>	Participant registration level in OZEDI. Will client be sending/receiving messages or both. If receiving is required, Peppol network registration will be performed. Values: "RECEIVING_ONLY" "SENDING_ONLY" "SENDING_AND_RECEIVING"

Request example

```

{
  "account": "002821888100703",
  "participant": "0151:45004189708",
  "companyName": "Company Name",
  "businessIdentifiers": [
    { "identifierType": "ABN", "identifier": "45004189708" }
  ],
  "countryCode": "AU",
  "contacts": [{ "contactEmail": "contact@example.com" }],
  "documentIdentifiers": [
    {
      "process": "cenbii-procid-ubl::urn:fdc:peppol.eu:2017:poacc:billing:01:1.0",
      "documentType": "busdox-docid-
qns::urn:oasis:names:specification:ubl:schema:xsd:Invoice-
2::Invoice#urn:cen.eu:en16931:2017#conformant#urn:fdc:peppol.eu:2017:poacc:billing:int
ernational:aunz:3.0::2.1"
    }
  ],
  "level": "SENDING_AND_RECEIVING"
}
    
```

Response

Status	Response
200	{ "client": "1888100703" }
400	Processing Error (BODY: See Errors (6))
401	Unauthorized (BODY: TBD)
403	Forbidden (BODY: TBD)
500	System error (BODY: TBD)
503	Service unavailable (Maintenance mode) (BODY: TBD)

API-2.3 OZEDI Client Update

Update selected properties of existing OZEDI client. List of supported document types and company data can be updated.

Method	Operation
PUT	/client/self-register/<client>

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token from Authenticate call
BODY (JSON)	account	<account>	OZEDI account number – 15 digit number. Validated. Read only.
	participant	<participant>	Peppol participant id. Validated. Read only.
	companyName	<companyName>	Participant company name. (Peppol Directory)
	companyInfo	<companyInfo>	<i>Optional.</i> Company description (Peppol Directory)
	countryCode	<countryCode>	Two letters ISO 3166-1 country code where business is registered (Peppol Directory)
	geographicalInfo	<geographicalInfo>	<i>Optional.</i> Business address string (Peppol Directory)
	businessIdentifiers[]	businessIdentifiers	ONE or more business identifiers. First one should have primary identifier issued by the regulator in the country where business is registered. E.g. ATO issued ABN for Australia. Others are optional and stored in Peppol Directory only.
businessIdentifiers element	identifierType	<identifierType>	Supported business identifier type: “ABN”, “NZBN” ...
businessIdentifiers element	identifier	<identifier>	Identifier value
	contacts[]	contacts	ONE or more contacts
contacts element	contactEmail	<contactEmail>	Contact email is mandatory. Used for OZEDI client as well as for Peppol Directory
contacts element	contactType	<contactType>	<i>Optional.</i> eg “Reception”
contacts element	contactName	<contactName>	<i>Optional.</i>
contacts element	contactPhoneNumber	<contactPhoneNumber>	<i>Optional.</i>
	documentIdentifiers[]	documentIdentifiers	ONE or more supported document identifiers. (SMP)
documentIdentifiers element	documentType	<documentType>	Peppol document type
documentIdentifiers element	process	<process>	Peppol document process identifier

	level	<level>	Participant registration level in OZEDI. Will client be sending/receiving messages or both. If receiving is required, Peppol network registration will be performed. Values: "RECEIVING_ONLY" "SENDING_ONLY" "SENDING_AND_RECEIVING"
--	-------	---------	--

Response

Status	Response
200	{ "client": "1888100703" }
400	Processing Error (BODY: See Errors (6))
401	Unauthorized (BODY: TBD)
403	Forbidden (BODY: TBD)
500	System error (BODY: TBD)
503	Service unavailable (Maintenance mode) (BODY: TBD)

API-2.4 OZEDI Client disable/delete

A Client can be deleted from the system if no messages have been sent/received via the following request.
A Client can be disabled if it had messages sent/received via the following request.

Method	Operation
DELETE	/client/deactivate/<client>

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token received from Authenticate call
PATH	client	<client>	OZEDI Client ID - 10-digit number

Response

Status	Response
204	No Content on successful delete/disable of the client
400	Processing Error (BODY: See Errors (6))
401	Unauthorized (BODY: TBD)
403	Forbidden (BODY: TBD)
500	System error (BODY: TBD)
503	Service unavailable (Maintenance mode) (BODY: TBD)

3. Sending APIs

API-3.1. Authenticate

Refer to API-1 above

API-3.2 Message Upload Token

A valid upload token is required to upload a document. It should be obtained before using **Message Send** operation. This is used to ensure the recipient is on Peppol network and can accept the document type specified.

Note: The document type must be specified. The ubl being sent will be validated for the specified document type. Refer to Appendix A for the specification of the documentType and the list currently supported by OZEDI.

Method	Operation
POST	/snd/message/token/<client>

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token from Authenticate call
PATH	client	<client>	OZEDI Client ID - 10-digit number
	sender	<sender>	Peppol participant identifier of sender
	recipient	<recipient>	Peppol participant identifier of recipient
	process	<process>	ID of the Peppol document process
	documentType	<documentType>	Peppol document type. Refer to Appendix A for valid values.

Request example

```
{
  "sender": "0151:78330347529",
  "recipient": "0151:31008779511",
  "process": "cenbii-procid-ubl::urn:fdc:peppol.eu:2017:poacc:billing:01:1.0",
  "documentType": "busdox-docid-
  qns::urn:oasis:names:specification:ubl:schema:xsd:Invoice-
  2::Invoice##urn:cen.eu:en16931:2017#conformant#urn:fdc:peppol.eu:2017:poacc:billing:int
  ernational:aunz:3.0::2.1"
}
```

Response

Status	Response
200	{ uploadToken: "be6f2060-7603-4faf-b344-7925eee77b37@OZEDIB2BToken" }
400	Processing Error (<i>BODY: See Errors (6)</i>)
401	Unauthorized (<i>BODY: TBD</i>)
403	Forbidden (<i>BODY: TBD</i>)
500	System error (<i>BODY: TBD</i>)
503	Service unavailable (Maintenance mode) (<i>BODY: TBD</i>)

API-3.3 Message Send

Send XML document via Peppol network.

Method	Operation
POST	/snd/message/send/<client>

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	multipart/form-data	
HEAD	Authorization	Bearer <id_token>	Token received from Authenticate call
FORM	uploadToken	<uploadToken>	Token received from Message Upload Token (3.2) call
PATH	client	<client>	OZEDI Client ID – 10 digit number
FORM	sender	<sender>	Peppol participant identifier of sender
FORM	recipient	<recipient>	Peppol participant identifier of recipient
FORM	payloadFile	<payloadFile>	UBL Document XML to upload

Response

Status	Response
200	{ messageUuid: "fd108618-9e67-424a-b0ee-8bef0770f123@OZEDIB2BMessage" }
400	<i>Processing Error (BODY: See Errors (6))</i>
401	<i>Unauthorized (BODY: TBD)</i>
403	<i>Forbidden (BODY: TBD)</i>
500	<i>System error (BODY: TBD)</i>
503	<i>Service unavailable (Maintenance mode) (BODY: TBD)</i>

4. Receiving APIs

API-4.1 Authenticate

Refer to API-1 above

API-4.2 Message Receive

Download document received by OZEDI via Peppol network for particular client. The required Message UUID can be retrieved via webhook or by getting list of filtered messages as described under Query APIs and Webhooks below.

Method	Operation
GET	/rcv/message/receive/<client>/<messageUuid>

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token received from Authenticate call
PATH	client	<client>	OZEDI Client ID - 10-digit number
PATH	messageUuid	<messageUuid>	UUID of the message

Response parameters

Type	Params	Values	Description
HEAD	Content-Type	application/xml	
BODY (XML)			UBL XML document

Response

Status	Response
200	<Invoice xmlns:cac="urn:oasis:names... ... </Invoice>
400	Processing Error (BODY: See Errors (6))
401	Unauthorized (BODY: TBD)
403	Forbidden (BODY: TBD)
500	System error (BODY: TBD)
503	Service unavailable (Maintenance mode) (BODY: TBD)

API-4.3 Message Receive Confirmation

Clients should acknowledge successful receiving of payload by calling this endpoint. Message status will change to RECEIVE_DELIVERED.

Method	Operation
PUT	/rcv/message/receive/confirm/<client>/<messageUuid>

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token received from Authenticate call
PATH	client	<client>	OZEDI Client ID - 10-digit number
PATH	messageUuid	<messageUuid>	UUID of the message

Response

Status	Response
200	Ok
400	Processing Error (BODY: See Errors (6))
401	Unauthorized (BODY: TBD)
403	Forbidden (BODY: TBD)
500	System error (BODY: TBD)
503	Service unavailable (Maintenance mode) (BODY: TBD)

Business Level Responses

The message receive confirm API can also be used to generate business level responses that communicate the result of processing by the recipients accounting system. A business level response transaction will be sent back to the invoice sender when there is a message request body. The business level response is sent as a document of type INVOICE_RESPONSE_3. A business level response can only be sent if the invoice sender's Peppol id is registered to receive documents of type INVOICE_RESPONSE_3.

Note: When the message receive confirm API is used with a body to send a BLR, the invoice status is not changed to RECEIVE_DELIVERED.

Request example

```
{
  "responseCode": "RE",
  "status": [
    {
      "statusReasonCode": "OTH",
      "statusReason": "GST Missing on Line Item"
    }
  ]
}
```

For Business Level Responses, the following are the accepted codes.

Accepted two letter responseCode list:

AB - Message acknowledgement
 IP - In process
 UQ - Under query (may be repeated before moving forward)
 CA - Conditionally accepted
 RE - Rejected
 AP - Accepted
 PD - Paid, can be in steps, partially paid and then paid.

Accepted three letter statusReasonCode list:

NON - No Issue. Indicates that receiver of the documents sends the message just to update the status and there are no problems with document processing
REF - References incorrect. Indicates that the received document did not contain references as required by the receiver for correctly routing the document for approval or processing.
LEG - Legal information incorrect. Information in the received document is not according to legal requirements.
REC - Receiver unknown. The party to which the document is addressed is not known.
QUA - Item quality insufficient. Unacceptable or incorrect quality
DEL - Delivery issues. Delivery proposed or provided is not acceptable.
PRI - Prices incorrect. Prices not according to previous expectation.
QTY - Quantity incorrect. Quantity not according to previous expectation.
ITM - Items incorrect. Items not according to previous expectation.
PAY - Payment terms incorrect. Payment terms not according to previous expectation.
UNR - Not recognized. Commercial transaction not recognized.
FIN - Finance incorrect. Finance terms not according to previous expectation.
OTH - Other

statusReason is optional, but required for certain response codes (CA, UQ or RE)

Status information will be used to generate Peppol Document Response document and attempt to deliver it via Peppol network back to the sender if sender is registered with Peppol as accepting this type of documents.

Additional Business Level Responses and responses with more detail can be generated by sending a document type of INVOICE_RESPONSE_3. This can be used to show progressive stages of processing through the ERP system, such as advising that payment has been approved and payment has been processed.

API-4.4 Receive UBL document as PDF

Message UBL can be converted to PDF form and downloaded via this API.

Method	Operation
GET	/message/download/direct/pdf/<client_id>/<messageUuid>

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token received from Authenticate call
PATH	Client id	<client_id>	OZEDI Client ID - 10-digit number
PATH	messageUuid	<messageUuid>	UUID of the message

Response parameters

Type	Params	Values	Description
HEAD	Content-Type	application/pdf	
BODY (PDF)			PDF document based on message UBL

Response

Status	Response
200	PDF body
400	Processing Error (BODY: See Errors (6))
401	Unauthorized (BODY: TBD)
403	Forbidden (BODY: TBD)
500	System error (BODY: TBD)
503	Service unavailable (Maintenance mode) (BODY: TBD)

API-4.5 Attachment Download

Extract and download embedded attachments from a document sent or received by OZEDI via Peppol network for particular client. The required Message UUID can be retrieved via webhook or by getting list of filtered messages as described under [Query APIs](#) and [Webhooks](#) below.

Method	Operation
GET	(Receiver) /rcv/message/receive/attachment/<client>/<messageUuid> (Sender) /snd/message/receive/attachment/<client>/<messageUuid>

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token received from Authenticate call
PATH	client	<client>	OZEDI Client ID - 10-digit number
PATH	messageUuid	<messageUuid>	UUID of the message

Response parameters

Type	Params	Values	Description
HEAD	Content-Type	One of: application/pdf, application/vnd.oasis.opendocument.spreadsheet, application/vnd.openxmlformats-officedocument.spreadsheetml.sheet, image/jpeg, image/png, text/csv, application/zip	Dependent on the attachment(s) contained within a UBL document, a single attachment will be downloaded as its own content type. Where multiple attachments exist, these will be returned as part of a single zip archive
BODY (Binary)			Binary file for extracted attachment(s)

Response

Status	Response
200	Binary file of attachment(s)
204	<i>No Content (BODY: Empty)</i>
400	<i>Processing Error (BODY: See Errors (6))</i>
401	<i>Unauthorized (BODY: TBD)</i>
403	<i>Forbidden (BODY: TBD)</i>
500	<i>System error (BODY: TBD)</i>
503	<i>Service unavailable (Maintenance mode) (BODY: TBD)</i>

5. Query APIs

The maximum number of records returned by query APIs is 20 by default. If more records are required to be returned, this can be changed via the “size” parameter.

Eg.

```
{{base_url_b2breceiver}}/message/pending/{{receive_client_id}}?size=30&sort=auditCreated,desc
```

API-5.1. Authenticate

Refer to API-1 above

API-5.2 Message Status

List the message status for a particular messageUUID.

Method	Operation
GET	/snd/message/status/<client>/<messageUuid>
GET	/rcv/message/status/<client>/<messageUuid>

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token from Authenticate call
PATH	client	<client>	OZEDI Client ID - 10-digit number
PATH	messageUuid	<messageUuid>	UUID of the message

Response

Status	Response
200	<pre>{ "auditCreated": "2020-03-13T10:15:41+11:00", "auditModified": "2020-03-13T10:15:50+11:00", "messageUuid": "19465642-cf65-4313-b313-bc04a0de811b@OZEDIB2BMessage", "sender": "0088:1235", "recipient": "0088:9429039618191", "messageStatusName": "SEND_FAILED" }</pre>
400	Processing Error (BODY: See Errors (6))
401	Unauthorized (BODY: TBD)
403	Forbidden (BODY: TBD)
500	System error (BODY: TBD)
503	Service unavailable (Maintenance mode) (BODY: TBD)

Message status names (messageStatusName):

Status Name	Description
SEND_UPLOAD_TOKEN	Token has been retrieved – pending final upload
SEND_IN_PROGRESS	Sending is in progress
SEND_SUCCESS	Sending completed successfully
SEND_FAILED	Sending failed – a non-validation error occurred (<i>failure condition retrieval API pending</i>)
SEND_FAILED_VALIDATION	Sending failed – validation error (<i>failure condition retrieval API pending</i>)
RECEIVE_IN_PROGRESS	Document received and it is processing
RECEIVE_READY	Document successfully processed and available for download
RECEIVE_FAILED	Receive process been failed/interrupted and cause should be investigated
RECEIVE_FAILED_VALIDATION	Document been successfully received and processed, but failed validation
RECEIVE_DELIVERED	Status indicate document been downloaded by the client

API-5.3 Validation Errors

List of messages sent and received. Use query parameters to filter the result set.

Method	Operation
GET	/snd/validation-errors/<messageUuid>
GET	/rcv/validation-errors/<messageUuid>

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token received from Authenticate call
PATH	messageUuid	<messageUuid>	UUID of the message

Response

Status	Response
200	<pre>{ "id": 63, "auditCreated": "2020-05-08T12:51:55+10:00", "auditModified": "2020-05-08T12:51:55+10:00", "code": "B2BERROR-LN2-CL00105", "description": "cvc-elt.1.a: Cannot find the declaration of element 'StandardBusinessDocument'.", "location": "Starting at line 2 entity or value is missing", "messageId": 55 }, { "id": 64, "auditCreated": "2020-05-08T12:51:55+10:00", "auditModified": "2020-05-08T12:51:55+10:00", "code": "PEPPOL-EN16931-R007", "description": "Business process MUST be in the format 'urn:fdc:peppol.eu:2017:poacc:billing:NN:1.0' where NN indicates the process number.\n", "location": "StandardBusinessDocument:Invoice", "messageId": 55 }, { "id": 65, "auditCreated": "2020-05-08T12:51:55+10:00", "auditModified": "2020-05-08T12:51:55+10:00", "code": "PEPPOL-EN16931-R051-AUNZ", "description": "All currencyID attributes must have the same value as the invoice currency code (BT-5), except for the invoice total tax amount in accounting currency (BT-111).\n", "location": "StandardBusinessDocument:Invoice:InvoiceLine:Price:PriceAmount", "messageId": 55 } }</pre>
400	Processing Error (BODY: See Errors (6))
401	Unauthorized (BODY: TBD)
403	Forbidden (BODY: TBD)
500	System error (BODY: TBD)
503	Service unavailable (Maintenance mode) (BODY: TBD)

Get a list of messages for a client, with filters to restrict the list.

Method	Operation
POST	/snd/message/filter/<client>
POST	/rcv/message/filter/<client>

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token received from Authenticate call
PATH	client	<client>	OZEDI Client ID - 10-digit number
	sender	<sender>	<i>Optional.</i> Peppol participant identifier of sender
	recipient	<recipient>	<i>Optional.</i> Peppol participant identifier of recipient.
	documentCategory	<documentCategory>	<i>Optional.</i> The type of document i.e. invoice, purchase order etc.. type: "AUNZ_INVOICE_3", "SG_INVOICE_3", "EU_INVOICE_3".. <i>Optional</i>
	dateFrom	<dateFrom>	<i>Optional, but mandatory if dateTo is included.</i> The date from message are filtered.
	dateTo	<dateTo>	<i>Optional, but mandatory if dateFrom is included.</i> The date up to which message are filtered.
	messageStatusName	<messageStatus>	<i>Optional.</i> Message status as per Message status names list .

Request example

```
{
  "sender": "0151:45004189708",
  "recipient": "0151:73076521161",
  "documentCategory": "AUNZ_INVOICE_3",
  "dateFrom": "2020-05-07T07:36:42+10:00",
  "dateTo": "2020-05-07T18:38:59+10:00"
  "messageStatusName": "SEND_FAILED_VALIDATION"
}
```

Response

Status	Response
200	<pre>{ "auditCreated": "2020-05-07T02:37:56Z", "auditModified": "2020-05-07T02:37:56Z", "messageUuid": "29b8f11d-655a-4796-9ce0-3b5814d1fc50@OZEDIB2BMessage", "sender": "0151:45004189708", "recipient": "0151:73076521161", "accountNo": "002821888100703", "client": "1928538050", "messageStatusName": "SEND_FAILED_VALIDATION" }, { "auditCreated": "2020-05-07T04:04:06Z", "auditModified": "2020-05-07T04:04:06Z", "messageUuid": "d884063d-a89b-4ae1-818c-2956ee11e5c4@OZEDIB2BMessage", "sender": "0151:45004189708", "recipient": "0151:73076521161", "accountNo": "002821888100703", "client": "1928538050", "messageStatusName": "SEND_FAILED_VALIDATION" }, { "auditCreated": "2020-05-07T04:06:07Z", "auditModified": "2020-05-07T04:06:07Z", "messageUuid": "3eac1103-c607-454c-a0e0-f874cbe42a6d@OZEDIB2BMessage", "sender": "0151:45004189708", "recipient": "0151:73076521161", "accountNo": "002821888100703", "client": "1928538050", "messageStatusName": "SEND_FAILED_VALIDATION" } }</pre>
400	Processing Error (<i>BODY: See Errors (6)</i>)
401	Unauthorized (<i>BODY: TBD</i>)
403	Forbidden (<i>BODY: TBD</i>)
500	System error (<i>BODY: TBD</i>)
503	Service unavailable (Maintenance mode) (<i>BODY: TBD</i>)

API-5.5 Message Pending List

By Client

List of messages ready to be received can be obtained using this endpoint. It is recommended to use this method instead of Message Filtered List when implementing regular polling for new not yet received documents e.g. documents with status RECEIVE_READY.

Method	Operation
GET	/rcv/message/pending/<client>

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token received from Authenticate call
PATH	client	<client>	OZEDI Client ID - 10-digit number
PATH (query param)	?sort=<value>,<sort order>	<value> Any field in status response to sort on. E.g. auditCreated, sender and etc. <sort order> is `asc` or `desc`	Optional. Default sort order is ascending and can be omitted leaving just field to sort on.

Response

Status	Response
200	[<pre> { "auditCreated": "2020-07-22T09:24:32Z", "auditModified": "2020-07-22T09:24:32Z", "messageUuid": "23f11cef-371d-4895-ae2b-55399bfcc6be@OZEDIB2BMessage", "sender": "0151:45004189708", "recipient": "0151:73076521161", "accountNo": "002821888100703", "client": "1928538050", "messageStatusName": "RECEIVE_READY", "direction": "IN", "documentCategoryName": "AUNZ_INVOICE_3", "documentDetails": { "invoiceDate": "2020-07-22", "customerName": "Trotters Trading Co Ltd", "supplierName": "Supplier Trading Name Ltd.", "invoiceNumber": "Snippet1", "invoiceDueDate": "2020-08-30", "invoiceTypeCode": "380", "purchaseOrderNumber": "SOMEBLERB", "invoiceTaxInclusiveAmount": "4455.85", "invoiceTaxInclusiveCurrency": "AUD" } } </pre>
400	Processing Error (BODY: See Errors (6))
401	Unauthorized (BODY: TBD)
403	Forbidden (BODY: TBD)
500	System error (BODY: TBD)
503	Service unavailable (Maintenance mode) (BODY: TBD)

By Account

List of messages ready to be received can be obtained using this endpoint. It is recommended to use this method instead of Message Filtered List when implementing regular polling for new not yet received documents e.g. documents with status RECEIVE_READY.

Method	Operation
GET	/rcv/message/pending/all/<account>

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token received from Authenticate call
PATH	client	<account>	OZEDI Account ID - 15-digit number
PATH (query param)	?sort=<value>,<sort order>	<value> Any field in status response to sort on. E.g. auditCreated, sender and etc. <sort order> is `asc` or `desc`	Optional. Default sort order is ascending and can be omitted leaving just field to sort on.

Response

Status	Response
200	<pre>[{ "auditCreated": "2020-07-22T09:24:32Z", "auditModified": "2020-07-22T09:24:32Z", "messageUuid": "23f11cef-371d-4895-ae2b-55399bfcc6be@OZEDIB2BMessage", "sender": "0151:45004189708", "recipient": "0151:73076521161", "accountNo": "002821888100703", "client": "1928538050", "messageStatusName": "RECEIVE_READY", "direction": "IN", "documentCategoryName": "AUNZ_INVOICE_3", "documentDetails": { "invoiceDate": "2020-07-22", "customerName": "Trotters Trading Co Ltd", "supplierName": "Supplier Trading Name Ltd.", "invoiceNumber": "Snippet1", "invoiceDueDate": "2020-08-30", "invoiceTypeCode": "380", "purchaseOrderNumber": "SOMEBLERB", "invoiceTaxInclusiveAmount": "4455.85", "invoiceTaxInclusiveCurrency": "AUD" } }, { "auditCreated": "2020-07-27T07:10:32Z", "auditModified": "2020-07-27T07:10:32Z", "messageUuid": "82e7284e-6061-4f32-b50e-e992b12c0248@OZEDIB2BMessage", "sender": "0151:45004189708", "recipient": "0151:66216527732", "accountNo": "002821888100703", "client": "1888100703", "messageStatusName": "RECEIVE_READY",</pre>

	<pre> "direction": "IN", "documentCategoryName": "AUNZ_INVOICE_3", "documentDetails": { "invoiceDate": "2020-07-27", "customerName": "Trading Co Ltd", "supplierName": "Supplier Trading Name Ltd.", "invoiceNumber": "Snippet2", "invoiceDueDate": "2020-09-30", "invoiceTypeCode": "380", "purchaseOrderNumber": "SOMEBLERB", "invoiceTaxInclusiveAmount": "64.35", "invoiceTaxInclusiveCurrency": "AUD" } } </pre>
400	Processing Error (BODY: See Errors (6))
401	Unauthorized (BODY: TBD)
403	Forbidden (BODY: TBD)
500	System error (BODY: TBD)
503	Service unavailable (Maintenance mode) (BODY: TBD)

Document details consist of information extracted from UBL and are different per document category. List of fields can be extended in later releases of the API.

5.6 Query Account Balance

Query to display the account balance.

Method	Operation
GET	/balance/<account>

Request parameters

Type	Params	Values	Description
HEAD	Content Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token from Authenticate call
PATH	Account	<account>	OZEDI Account ID - 15-digit number

Response

Status	Response
200	<pre> { "auditCreated": "2020-10-07T22:18:50Z", "auditModified": "2020-11-06T02:29:42Z", "dollarAmount": "48.48" } </pre>
400	Processing Error (BODY: See Errors (6))
401	Unauthorized (BODY: TBD)
403	Forbidden (BODY: TBD)
500	System error (BODY: TBD)
503	Service unavailable (Maintenance mode) (BODY: TBD)

5.7 Participant document type support query (SML/SMP)

Query to check if participant is on Peppol network and get list of document types it can receive it is.

Method	Operation
GET	/snd/peppol/documents/<account>/<participant>

Request parameters

Type	Params	Values	Description
HEAD	Content Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token from Authenticate call
PATH	Account	<account>	OZEDI Account ID - 15-digit number
PATH	Participant	<participant>	Peppol participant identifier

Response example

```
{
  "documentIdentifiers": [
    {
      "process": "cenbii-procid-ubl::urn:fdc:peppol.eu:poacc:trns:invoice_response:3",
      "documentType": "busdox-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:ApplicationResponse-2::ApplicationResponse##urn:fdc:peppol.eu:poacc:trns:invoice_response:3::2.1"
    },
    {
      "process": "cenbii-procid-ubl::urn:cen.eu:en16931:2017#conformant#urn:fdc:peppol.eu:2017:poacc:billing:international:aunz:3.0",
      "documentType": "busdox-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##urn:cen.eu:en16931:2017#conformant#urn:fdc:peppol.eu:2017:poacc:billing:international:aunz:3.0::2.1"
    }
  ]
}
```

Response

Status	Response
200	List of document types supported by given participant. See above response example.
204	Participant not found on Peppol Network
400	<i>Processing Error (BODY: See Errors (6))</i>
401	<i>Unauthorized (BODY: TBD)</i>
403	<i>Forbidden (BODY: TBD)</i>
500	<i>System error (BODY: TBD)</i>
503	<i>Service unavailable (Maintenance mode) (BODY: TBD)</i>

6. Webhooks

API-6.1. Authenticate

Refer to API-1 above

API-6.2 Register Webhooks

Register webhooks to be notified about message related events. Separate webhooks for sending and receiving documents can be registered.

After a webhook is registered with OZEDI, a POST request will be sent to the specified `targetUrl` each time an event with the type and triggerType defined in the webhook takes place. Content of this request depends on the webhook type. For MESSAGE_STATUS it is the same as [Message Status response](#). Each webhook request triggered by OZEDI has a header parameter `x-oz-signature` which can be used to verify the origin of the request. See [response](#) definition for additional details.

Method	Operation
POST	/snd/webhook/<account>
POST	/rcv/webhook/<account>

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
HEAD	Authorization	Bearer <authorization-token>	Token received from Authenticate call
BODY	type	<type>	ONE of supported webhook notification types: MESSAGE_STATUS
BODY	enabled	<enabled>	Webhook enabled state: true/false
BODY	triggerType	<triggerType>	ONE of the supported types: CLIENT, ACCOUNT
BODY	triggerValue	<triggerValue>	Values for selected trigger type. Client id for triggerType: CLIENT

Request example

```

{
  "targetUrl": "https://webhook.site/130eaa91-8773-421f-b316-760973a68176",
  "type": "MESSAGE_STATUS",
  "enabled": true,
  "triggerType": "CLIENT",
  "triggerValue": "1136958912"
}
    
```

Response

Secret can be used to verify webhook been called by OZEDI. Every webhook request will have `x-oz-signature` in the request header which have Hash-based Message Authentication Code (HMAC) created by applying the SHA256 hash function on `secret` provided in webhook creation response. This code also acts as webhook identifier and have to be used in update requests.

Status	Response
200	<pre>{ "auditCreated": "2020-05-27T01:49:22Z", "auditModified": "2020-07-27T01:49:22Z", "targetUrl": "https://webhook.site/945f6cad-4679-4cbe-b197-e6d65f716bfd", "type": "MESSAGE_STATUS", "enabled": true, "account": "123454721718889", "triggerType": "CLIENT", "triggerValue": "7436958942", "secret": "d03e65b13686457282d100f8fedc5a73" }</pre>
400	Processing Error. See below.

Processing Errors

Error code	Error description
400	Webhook for the account, type, triggerType and triggerValue is already registered.

API-6.3 Webhook - List

Retrieve list of all registered webhooks.

Method	Operation
GET	/snd/webhook/<account>
GET	/rcv/webhook/<account>

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token received from Authenticate (1) call
PATH	account	<account>	OZEDI Client ID - 15-digit account number

Response

Status	Response
200	Ok
400	Processing Error (BODY: See Errors (6))
401	Unauthorized (BODY: TBD)
403	Forbidden (BODY: TBD)
500	System error (BODY: TBD)
503	Service unavailable (Maintenance mode) (BODY: TBD)

Response Example

```
[
  {
    "auditCreated": "2020-05-27T01:49:22Z",
    "auditModified": "2020-07-27T01:49:22Z",
    "targetUrl": "https://webhook.site/945f6cad-4679-4cbe-b197-e6d65f716bfd",
    "type": "MESSAGE_STATUS",
    "enabled": true,
    "account": "123454721718889",
    "triggerType": "CLIENT",
    "triggerValue": "7436958942",
    "secret": "d03e65b13686457282d100f8fedc5a73"
  }
]
```

API-6.4 Webhook - Update

Update existing webhook. For example, if webhook need to be disabled or target URL is changed.

Method	Operation
PUT	/snd/webhook/<account>
PUT	/rcv/webhook/<account>

Request parameters

Type	Params	Values	Description
HEAD	Content-Type	application/json	
HEAD	Authorization	Bearer <id_token>	Token received from Authenticate (1) call
PATH	account	<account>	OZEDI Client ID - 15-digit account number
BODY	secret	<secret>	Mandatory webhook identifier
BODY	targetUrl	<targetUrl>	Optional parameter to update
BODY	enabled	<enabled>	Optional parameter to update

Request example

```
{
  "secret": "d76058d15677444382d799dce94dbdf3",
  "enabled": false
}
```

Response

Status	Response
200	<pre>{ "auditCreated": "2020-05-27T01:49:22Z", "auditModified": "2020-06-20T06:27:36Z", "targetUrl": "https://webhook.site/945f6cad-4679-4cbe-b197-e6d65f716bfd", "type": "MESSAGE_STATUS", "enabled": false, "account": "123454721718889", "triggerType": "CLIENT", "triggerValue": "7436958942", "secret": "d76058d15677444382d799dce94dbdf3" }</pre>
400	Processing Error (BODY: See Errors (6))
401	Unauthorized (BODY: TBD)
403	Forbidden (BODY: TBD)
500	System error (BODY: TBD)
503	Service unavailable (Maintenance mode) (BODY: TBD)

Error Examples

Input parameters validation error example

<pre>{ "type": "https://e-invoicing.OZEDI.com.au/problem/message-error/constraint-violation", "title": "Method argument not valid", "status": 400, "path": "/api/message/token", "message": "error.validation", "fieldErrors": [{ "objectName": "messageTokenRequest", "field": "client", "message": "must match \"[0-9]{10}\"" }] }</pre>
--

Processing error example

<pre>{ "timestamp": "2020-03-13T10:08:11.2969901+11:00", "type": "https://e-invoicing.OZEDI.com.au/problem/message-error/problem-with-message", "title": "Validation Error", "status": 400, "detail": "Peppol network issue: Error locating participant on Peppol network. ParticipantId: 0088:9429039618191" }</pre>

Appendix A: Document Type Specifications

When a Client for receiving is registered with OZEDI, the following are the Processes and Document Type specifications that are registered with Peppol as accepted to be received.

#	Name	Process	Document Type	Description
1	AUNZ_INVOICE_3	cenbii-procid-ubl::urn:fdc:peppol.eu:2017:poacc:billing:01:1.0	busdox-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##urn:cen.eu:en16931:2017#conformant#urn:fdc:peppol.eu:2017:poacc:billing:international:aunz:3.0::2.1	AU/NZ Invoice
2	AUNZ_CREDIT_NOTE_3	cenbii-procid-ubl::urn:fdc:peppol.eu:2017:poacc:billing:01:1.0	busdox-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:CreditNote-2::CreditNote##urn:cen.eu:en16931:2017#conformant#urn:fdc:peppol.eu:2017:poacc:billing:international:aunz:3.0::2.1	AU/NZ Credit Note
3	MESSAGE_LEVEL_RESPONSE_3	cenbii-procid-ubl::urn:fdc:peppol.eu:poacc:bis:mlr:3	busdox-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:ApplicationResponse-2::ApplicationResponse##urn:fdc:peppol.eu:poacc:trns:mlr:3::2.1	Message Acknowledgement
4	INVOICE_RESPONSE_3	cenbii-procid-ubl::urn:fdc:peppol.eu:poacc:bis:invoice_response:3	busdox-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:ApplicationResponse-2::ApplicationResponse##urn:fdc:peppol.eu:poacc:trns:invoice_response:3::2.1	Business Response
5	AUNZ_SB_INVOICE_3	cenbii-procid-ubl::urn:fdc:peppol.eu:2017:poacc:selfbilling:01:1.0	busdox-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##urn:cen.eu:en16931:2017#conformant#urn:fdc:peppol.eu:2017:poacc:selfbilling:international:aunz:3.0::2.1	AU/NZ Self-Billing Invoice
6	AUNZ_SB_CREDIT_NOTE_3	cenbii-procid-ubl::urn:fdc:peppol.eu:2017:poacc:selfbilling:01:1.0	busdox-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:CreditNote-2::CreditNote##urn:cen.eu:en16931:2017#conformant#urn:fdc:peppol.eu:2017:poacc:selfbilling:international:aunz:3.0::2.1	AU/NZ Self-Billing Credit Note

We are progressively adding additional document types.