



PAROCHOS

— ΛΥΣΕΙΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΤΙΜΟΛΟΓΙΣΗΣ



API Documentation - Provider

Contents

Versions	3
API Services	4
1. Authentication	4
1.1 Login.....	4
1.2 Refresh token.....	6
1.3 API User.....	7
2. AADE Provider API Calls.....	8
2.1 Request Payment.....	9
Code identifier of the EFT-POS service provider. Values described in Appendix 3.....	10
2.2 Invoice Transmission.....	11
2.3 Polling process status	16
2.4 Grant write access to Azure Storage	17
2.5 Verify upload to Azure Storage.....	19
2.6 Download files.....	20
2.7 Download publish logs (B2G only)	22
2.8 Submit Payment.....	24
2.9 Cancel Payment.....	26
2.10 Get Payments	27
3. Invoice Source Data analysis	29
3.1 Tax Authorities' & UBL2.1 configured in External System	30
3.2 Tax Authorities' & UBL2.1 configured in web portal.....	37
3.3 Invoice Transmission to end customer.....	41
3.4 Invoice Delayed Transmission.....	42
Appendix 1	43
Appendix 2.....	44
Appendix 3.....	47

Versions

Version	Date	Changes
2.1.0	01.12.2021	
2.1.1	11.03.2022	2.6 Download files added
2.1.2	30.03.2022	Design enhancements
2.1.3	07.04.2022	Add method on each error code
2.1.4	15.04.2022	Add error category
2.1.5	27.06.2022	Counterpart email mandatory under specific conditions
2.1.6	13.10.2022	Add required fields for B2G Electronic Invoice Transmission
2.1.7	24.10.2022	url1 at Appendix 1
2.1.8	17.10.2023	Add "Download publish logs" and "B2G Validation Error Messages" (Appendix 2) documentation
2.1.9	24.10.2023	Add invoiceNote, organizationalUnitName, organizationalUnitCode in "Invoice Source Analysis"
3.0.0	14.03.2024	EFT-POS Payments Initial Implementation – RequestPayment/ SubmitPayment/ CancelPayment API calls
3.0.1	01.04.2024	Remove ProcessId from EFT-POS Payments related request models (RequestPayment/ SubmitPayment/ CancelPayment). Identify invoice by ExternalSystemId. Update missing payment response "status" field in documentation. Add missing B2G status values in documentation. Allow for dummy InvoiceAA upon pre-transmission payment request.
3.0.2	02.04.2024	Update transmission diagram to depict the EFT-POS payment transaction flow. New method GetPayments to retrieve all requested payments and their status.
3.0.3	10.04.2024	EFT-POS Payments: method Send extended with a new transmissionType (4) to allow requesting Provider's signature(s). Update transmission diagram to depict updated EFT-POS payment transaction flow. Update response models of methods Send and GetPayments to include UID and terminalId (values required by POS terminal). Update 3.1 and 3.2 with terminalId under paymentMethods.
3.0.4	18.04.2024	Update RequestPayment with nspCode input parameter. Add new Appendix 5, describing the allowed values on the nspCode field.
3.0.5	19.04.2024	Update 3.1 and 3.2 with nspCode under paymentMethods.

API Services

1. Authentication

The API communicates with the authentication server of the system. The identity server is responsible for generating the auth token necessary for all API methods. Therefore, there are two end points one for the authentication server and one for API. Identity server base URL in the DEV env is the following: <https://beta-account.parochos.gr/>

1.1 Login

Login method is responsible for authenticating the user to the system and requires user's credentials as well as the activated subscription's secret key, as displayed in the screenshot below, which uniquely identifies and validates the user's access. Generates and returns the auth token (jwtoken), a refresh token and the subscription info of the user.



Request

Method	URL
POST	/api/account/loginToSubscription

Parameter	Values	Description
subscriptionKey	string	Secret key of the user subscription.
Email	string	User's email.
password	string	User's password.

Response

Status	Response	Values	Description
200	jwt	string	The access token. Once the user is logged in, each subsequent request will include the jwt, allowing the user to access routes, services, and resources that are permitted with that token.
	jwtExpiration	DateTime	The expiration DateTime of the jwt.
	jwtRefreshToken	string	A refresh token that carries the information necessary to get a new access token. Whenever an access token is required to access a specific method, the client may use a refresh token to get a new access token issued by the authentication server.
	jwtRefreshTokenExpiration	DateTime	The expiration DateTime of the refresh token. The refresh token can also expire but are rather long-lived.
	url1	string	The base url of API.
400			When parameters are invalid.
403			If the username, password or secret key is invalid.

Example

REQUEST	POST /api/account/loginToSubscription
	<pre>{ "subscriptionKey": "1EE21B81418434A851E7AF0CED9B28C", "email": "user@test.gr", "password": "123456" }</pre>
RESPONSE	HTTP/1.1 200 OK
	<pre>{ "jwt": "eyJz93a...k4laUWw", "jwtExpiration": "2022-09-12T00:51:07Z", "jwtRefreshToken": "GEbRxBN...edjnXbL", "jwtRefreshTokenExpiration": "2022-10-21T21:51:07.9270559Z", "url1": "api url" }</pre>

1.2 Refresh token

Whenever an access token is required to access a specific method, the client must check if this token has expired. If so, refresh method is called passing a refresh token and the expired jwt to the authentication server. The server generates both new access token (jwt) and refresh token with updated expiration date. In case the refresh token expires or whenever call to refresh method fails, the client has to authenticate again using the login method.

Request

Method	URL
POST	/api/token/refresh

Parameter	Values	Description
token	string	The expired jwt.
refreshToken	string	A valid refresh token.

Response

Status	Response	Values	Description
200	jwt	string	The new access token.
	jwtExpiration	DateTime	The expiration DateTime of the jwt.
	jwtRefreshToken	string	The new refresh token.
	jwtRefreshTokenExpiration	DateTime	The expiration DateTime of the refresh token.
400			When parameters are invalid.
403			When either token or refresh token is invalid.

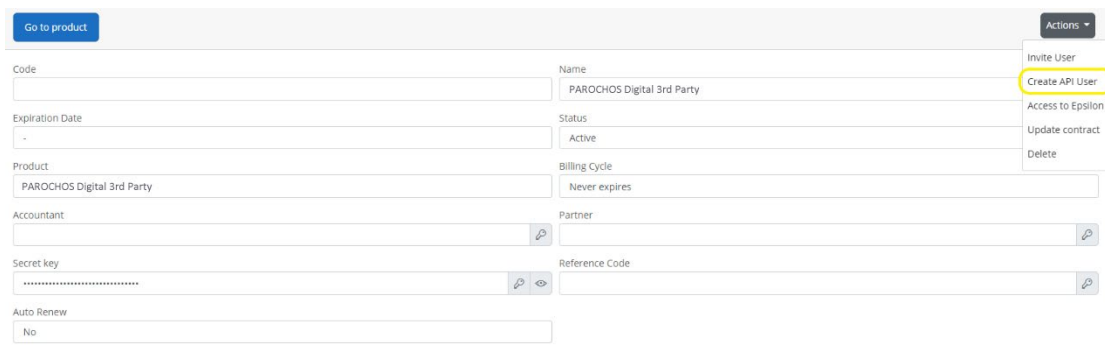
Example

REQUEST	POST /api/token/refresh
	<pre>{ "token": "eyJz93a...k41aUWw", "refreshToken": "GEbRxBN...edjnXbl" }</pre>
RESPONSE	HTTP/1.1 200 OK
	<pre>{ "jwt": "eyJz93a...kjd12az", "jwtExpiration": "2022-09-15T10:12:19Z", "jwtRefreshToken": "AroAv...NGtHpawGN", "jwtRefreshTokenExpiration": "2022-10-22T01:12:19.1768954Z", }</pre>

1.3 API User

API User is a personalized service account, created through web portal, with the following characteristics:

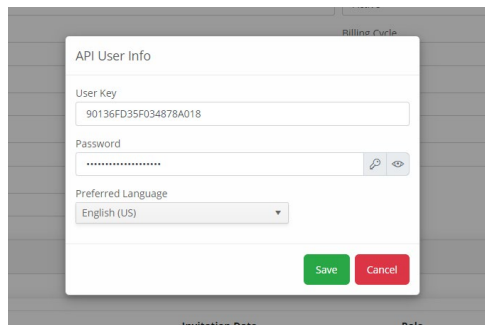
- The minimum set of access rights are assigned to this user role in order to access API but not web portal.
- This user is unique per subscription and cannot be used in more than one subscriptions.
- Separate culture definition is assigned to this user so that the transmission through API is not affected by the choices of the actual users.



The screenshot shows a web portal interface for managing API users. At the top left is a 'Go to product' button. The main area contains a form with the following fields:

- Code:** Empty text input.
- Name:** PAROCHOS Digital 3rd Party
- Expiration Date:** -
- Status:** Active
- Product:** PAROCHOS Digital 3rd Party
- Billing Cycle:** Never expires
- Accountant:** Empty text input with a link icon.
- Partner:** Empty text input with a link icon.
- Secret key:** Masked text input with a link icon and an eye icon.
- Reference Code:** Empty text input with a link icon.
- Auto Renew:** No

An 'Actions' dropdown menu is open on the right side, listing the following options: Invite User, **Create API User** (highlighted with a yellow circle), Access to Epsilon, Update contract, and Delete.

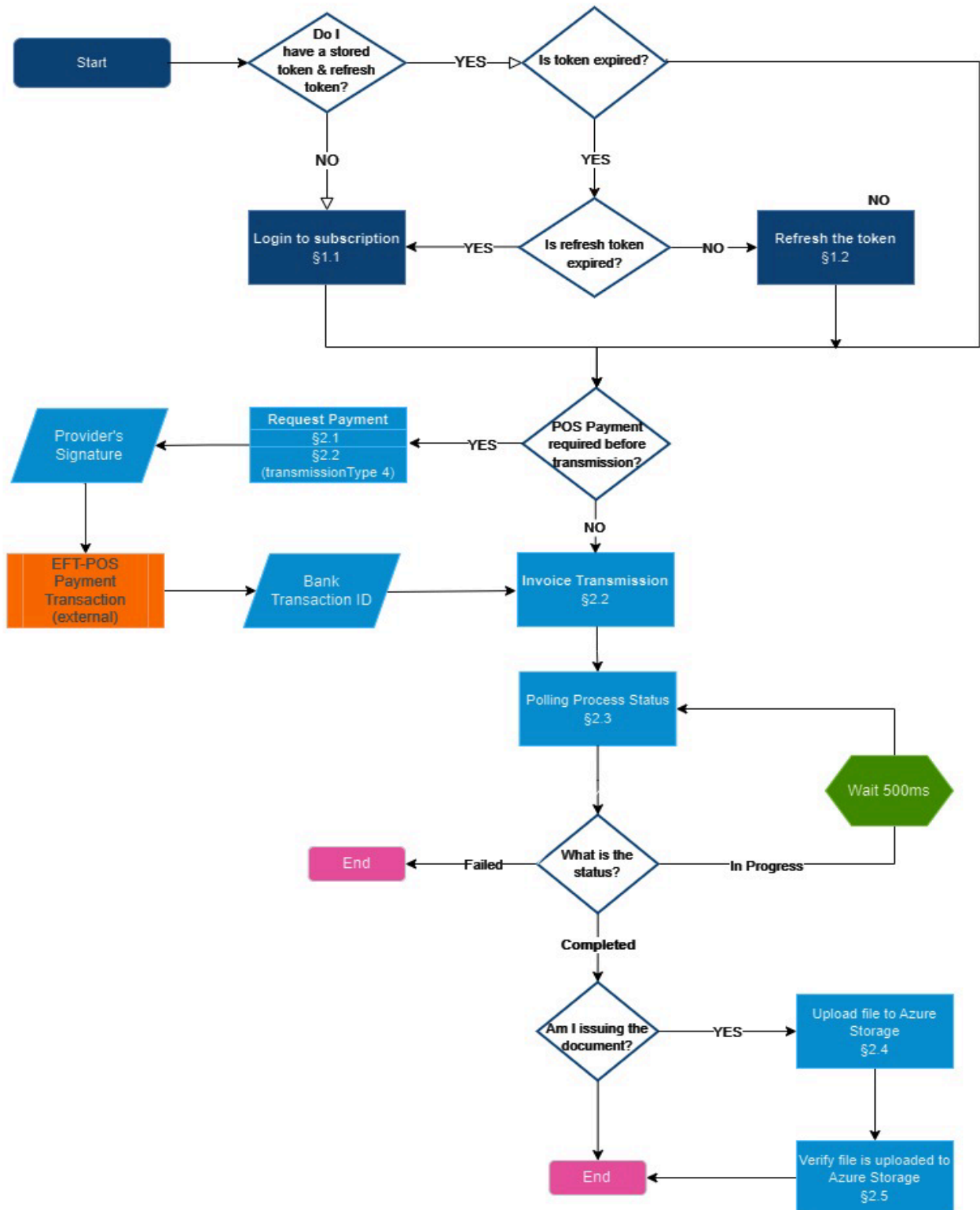


The screenshot shows a modal dialog box titled 'API User Info'. It contains the following fields:

- User Key:** 90136FD35F034878A018
- Password:** Masked text input with a link icon and an eye icon.
- Preferred Language:** English (US) (dropdown menu)

At the bottom of the dialog are two buttons: 'Save' (green) and 'Cancel' (red).

2. AADE Provider API Calls



API provides a collection of methods for sending invoices to AADE Platform. All the following methods require authorization header using the Bearer scheme. The jwt returned by the [loginToSubscription](#) should be used to the authorization header. The url1 returned by the [loginToSubscription](#) is the base address to all below methods.

As of version 3.0, every call to the API must denote the desired API version in the HTTP header, under key *X-Version*.

Provider API calls can be found on <https://beta-srv.parochos.gr/swagger/index.html>

2.1 Request Payment

Invoice transmissions involving EFT-POS payments, require a Provider's signature, verifying the POS payment, to be committed to the final POS terminal for the bank transaction to be completed. This method natively precedes invoice transmission (API method [Send](#)) in the transmission flow ^[1].

The returned Provider's signature must be included in the source invoice as presented in [3.1 Tax Authorities' & UBL2.1 configured in External System](#) and [3.2 Tax Authorities' & UBL2.1 configured in web portal](#) under the element `paymentMethods`.

NOTE: Invoking the RequestPayment method before transmitting the invoice, is required only if any payment methods with an EFT-POS payment type are defined in the invoice source.

Request

Method	URL
POST	/api/requestPayment
HTTP Header	Key: X-Version Value: 3.0 <i>*required</i>

Parameter	Values	Description
<code>externalSystemId</code>	string	An identifier of the invoice specified by the client. Each invoice is validated based on its status and this external ID. For example a successfully transmitted invoice with the same external ID is not allowed to resend.
<code>issuerVatNumber</code>	string	Issuer's TIN (VAT Number)
<code>invoiceIssueDate</code>	DateTime	Invoice Date ISO 8601
<code>companyBranch</code>	Number	Issuer's Company Branch N/A as registered in Tax Authorities' platform
<code>invoiceType</code>	string	Invoice Type according to Tax Authorities' documentation

¹ In the case of [2.8 Submit Payment](#), the RequestPayment API call will be invoked **after** invoice transmission.

invoiceSeries	string	Series Code of Invoice issued, use to resolve invoice type.
invoiceAA	string	Invoice Number
mark [optional]	long	Unique registration number of the invoice provided by AADE, if available. Required only in 2.8 Submit Payment scenario.
netValue	decimal	Invoice total net value
vatAmount	decimal	Invoice total vat amount
totalValue	decimal	Invoice total gross value
paymentAmount	decimal	Amount to be paid through the EFT-POS transaction
terminalId	string	The Id of the EFT-POS terminal that will perform the payment transaction
nspCode	string	Code identifier of the EFT-POS service provider. Values described in Appendix 3

Response

Status	Response	Values	Description	
200	externalSystemId	string	The external identifier provided to requestPayment method.	
	processId	guid	The identifier of the invoice. Will be null until the actual transmission is performed, through the Send method	
	uid	string	Identifier of the invoice required for the signature generation and for completing the EFT-POS payment.	
	paymentToken	timestamp	Datetime	The timestamp of signature creation in Greek time zone format
		signature	string	The payment signature generated by the Provider.
		amount	decimal	The requested payment amount
	status	number	The progress status of the procedure. Completed = 1, Failed = 2.	
	errorCode	string	In case invoice upload fails, errorCode indicates the type of the error.	
	errorMessage	string	Detailed description of the error.	
errorCategory	string	Indicates the category of the error. TED = Provider Technical error BED = Provider Business error TTA = Technical error by Tax Authorities BTA = Business error by Tax Authorities		

401			Unauthorized request. The jwt is either invalid or expired.
-----	--	--	---

Example

REQUEST	POST /api/requestPayment
<pre>{ "externalSystemId": "bcadd6ae-e8a6-5r84-g3da-23cbb7vs764k", "issuerVatNumber": "099788256", "invoiceIssueDate": "2024-03-13T14:22:22", "companyBranch": "0", "invoiceType": "1.1", "invoiceSeries": "ΤΔΑ", "invoiceAA": "32476", "netValue": 2628.38, "vatAmount": 630.81, "totalValue": 3259.19, "paymentAmount": 3259.19, "terminalId": "01234567" }</pre>	
RESPONSE	HTTP/1.1 200 OK
<pre>{ "processId": null, "externalSystemId": "bcadd6ae-e8a6-5r84-g3da-23cbb7vs764k", "uid": "string", "paymentToken": { "timestamp": "2024-03-13T12:20:20.700Z", "signature": "6B42CBC7C1CC02724095E6169F17819D[...]75E49FBB90E9FFA4D22944B2BBC02F1E27CFE57C8", "amount": 3259.19 }, "status": 1, "errorCategory": null, "errorCode": null, "errorMessage": null }</pre>	
RESPONSE	HTTP/1.1 401 Unauthorized
<pre>{}</pre>	

2.2 Invoice Transmission

API provides the send method for more than one transmissions. The required property `transmissionType` resolves the type of the transmission, thus the invoice transmission is responsible to certify invoice's content and transmit it to AADE platform.

When issuing the invoice's pdf document is required the parameter `transmissionType` is set to value 1. In this case, the design of at least one print form is required. The design of the print form can be configured through UI configuration.

An alternative approach to [2.1 Request Payment](#), for getting the Provider's signature(s) required to complete EFT-POS payment transactions, is available by setting the parameter `transmissionType` to value 4. Under this setting, Parochos will return a list of payment tokens for each one of the payment methods specified for EFT-POS payment without transmitting the invoice to AADE platform.

Request

Method	URL
POST	/api/send
HTTP Header	Key: X-Version Value: 3.0 *required

Parameter	Values	Description	
<code>externalSystemId</code>	string	An identifier of the invoice specified by the client. Each invoice is validated based its status and this external ID. For example a signed invoice with the same external ID is not allowed to resend.	
<code>source</code>	string	Json serialized object that contains the data to be send. See section 3. Source explanatory	
<code>identifier</code> [optional]	string	The identifier of the template used to map the incoming invoice data to the model required by AADE Provider. This value is specified through UI configuration.	
<code>transmissionType</code> [optional]	Number	Resolves the required type of transmission. 0 - transmits invoice to Tax Authorities Platform. 1 - transmits invoice to Tax Authorities Platform and generates invoice pdf file. 4 - generates Provider's signature(s) without transmitting invoice to Tax Authorities Platform. (Default Value = 0)	
<code>attachments</code> [optional]	<code>content</code> [optional]	string	Base64 encoded binary data. A draft .txt file of the invoice. It is the base64 representation string of the byte array of the file.

Response

Status	Response	Values	Description
200	<code>externalSystemId</code>	string	The external identifier provided to send method.
	<code>processId</code>	guid	The identifier of the invoice
	<code>signing</code> <code>uid</code>	string	Identifier of the invoice provided by AADE.

	mark	long	Unique registration number of the invoice provided by AADE.
	authenticationCode	string	Authentication code of the invoice provided by AADE.
	qrCode	string	The url where the client can see the invoice and download its files produced by the system.
	pdfUploaded	Boolean	Specifies whether the pdf of the invoice is uploaded or not.
	pdfFileUrl	string	Azure storage url with read rights to download the invoice pdf file.
	publishStatus	number	The status of the B2G invoice transmission. Applies only to B2G invoices. Pending = 0, Sending = 1, Transmitted = 2, Processing = 3, Paid = 4, Rejected = 5, UnderQuery = 6, Failed = 10
	paymentTokens	array	
	timestamp	Datetime	The timestamp of signature creation in Greek time zone format
	signature	string	The payment signature generated by the Provider
	amount	decimal	The requested payment amount
	terminalId	string	The Id of the EFT-POS terminal that will perform the payment transaction
	uid	string	Identifier of the invoice required for the signature generation and for completing the EFT-POS payment.
	timeStamp	DateTime	The timestamp of invoice creation in UTC
	status	number	The progress status of the procedure. InProgress = 0, Completed = 1, Failed = 2.
	errorSeverity	number	Warning = 1, Error = 2.
	errorCode	string	In case invoice upload fails, errorCode indicates the type of the error.
	errorMessage	string	Detailed description of the error.

	<code>errorCategory</code>	string	Indicates the category of the error. TED = Provider Technical error BED = Provider Business error TTA = Technical error by Tax Authorities BTA = Business error by Tax Authorities
401			Unauthorized request. The jwt is either invalid or expired.

Example

REQUEST	POST /api/send
<pre>{ "externalSystemId": "185", "identifiier": " eInvoicing", "transmissionType": 0, "source": {} }</pre>	
RESPONSE	HTTP/1.1 200 OK
<pre>{ "processId": "fe340fa6-444a-4ce7-1a71-08d9c47b8688", "externalSystemId": "185", "signing": { "uid": null, "mark": null, "authenticationCode": null, "qrCode": "https://localhost:6001/FileDocument/Get/fe340fa6-444a-4ce7-1a71-08d9c47b8688", "pdfUploaded": false, "pdfFileUrl": null, "publishStatus": 0, "paymentTokens": [{ "timestamp": "2024-03-13T12:20:20.700Z", "signature": "6B42CBC7C1CC02724095E6169F178[...]49FBB90E9FFA4D22944B2BBC02F1E27CFE57C8", "amount": 3259.19, "terminalId": "1234567890", "uid": "14BFA565194F552B3A3462DE07A48B2E24484BC1" }] }, "status": 0, "errorCategory": null, "errorCode": null, "errorSeverity": null, "errorMessage": null "timeStamp": "2021-12-01T12:15:06.6338557Z", }</pre>	
RESPONSE	HTTP/1.1 401 Unauthorized
<pre>{}</pre>	


```

    },
    "status": 1,
    "timeStamp": "2021-12-21T12:15:06.6338557",
    "errorCategory": null,
    "errorCode": null,
    "errorSeverity": null,
    "errorMessage": null
  }

```

2.4 Grant write access to Azure Storage

When an invoice has successfully been transmitted to AADE and only when the client is responsible for generating invoice's pdf file, the following process must be followed. The client must create the pdf file of the invoice and request for a shared access signature (SAS) to grant limited access to Azure Storage and therefore upload the file there.

Request

Method	URL
POST	/api/uploadFileRequest
HTTP Header	Key: X-Version Value: 3.0 *required

Parameter	Values	Description
processId ^[1] [optional]	guid	Unique processId as returned by api/send.
externalSystemId ^[1] [optional]	number	The external identifier provided as input to api/send method
fileName	string	The name of the pdf file.
fileSize	number	The size in bytes of the pdf file.

[1] At least one of the above should be defined

Response

Status	Response	Values	Description
200	uploadUrl	string	Azure storage url. [2]
	status	number	Status of the upload request. Success =1, Failed =2
	errorCode	string	In case upload request fails, errorCode indicates the type of the error.
	errorMessage	string	Detailed description of the error.
	errorCategory	string	Indicates the category of the error. TED = Provider Technical error BED = Provider Business error TTA = Technical error by Tax Authorities BTA = Business error by Tax Authorities
401			Unauthorized request. The jwt is either invalid or expired.

Example

REQUEST	POST /api/uploadFileRequest
	<pre>{ "externalSystemId": "185", "fileName": "Invoice.pdf", "fileSize": 33564 }</pre>
RESPONSE	HTTP/1.1 200 OK
	<pre>{ "uploadurl": "url", "status": 1, "errorCode": null, "errorCategory": null, "errorMessage": null }</pre>

² Use the Azure storage url to send a new PUT httpRequest (See here).

Request Syntax: PUT <uploadurl> HTTP/1.1

Request Headers: Content-Type: application/pdf
x-ms-blob-type: BlockBlob

Request Body: Content of pdf file as binary

2.5 Verify upload to Azure Storage

After client completing the file's upload to Azure Storage, this method should be called to verify file's content and complete the process.

Request

Method	URL
POST	api/Finalize
HTTP Header	Key: X-Version Value: 3.0 *required

Parameter	Values	Description
<code>processId</code> ^[1] [optional]	Guid	Unique processId as returned by api/send.
<code>externalSystemId</code> ^[1] [optional]	string	The external identifier provided as input to api/send method

[1] At least one of the above should be defined

Response

Status	Response	Values	Description
200	<code>status</code>	number	Status of the verify request. Success =1, Failed =2
	<code>errorCode</code>	string	In case verify request fails, errorCode indicates the type of the error.
	<code>errorMessage</code>	string	Detailed description of the error.
	<code>errorCategory</code>	string	Indicates the category of the error. TED = Provider Technical error BED = Provider Business error TTA = Technical error by Tax Authorities BTA = Business error by Tax Authorities
401			Unauthorized request. The jwt is either invalid or expired.

Example

REQUEST	POST /api/Finalize
<pre>{ "externalSystemId": "185" }</pre>	
RESPONSE	HTTP/1.1 200 OK
<pre>{ "status": 1, "errorCode": null, "errorCategory": null, "errorMessage": null }</pre>	

2.6 Download files

When the whole process has been successfully completed, the client can therefore download all the files uploaded, at any time. The request returns download urls of the pdf uploaded by the client, together with the actual xml file transmitted to Tax Authorities' and the e-Invoice for PEPPOL.

Request

Method	URL
POST	api/getFiles
HTTP Header	Key: X-Version Value: 3.0 <i>*required</i>

Parameter	Values	Description
processId ^[1] [optional]	Guid	Unique processId as returned by api/send.
externalSystemId ^[1] [optional]	string	The external identifier provided as input to api/send method
fileIdentifiers[optional]	array	List of file identifiers. If empty, all files are returned. Accepted values: <ul style="list-style-type: none"> pdf for invoice pdf, mydataxml for Tax Authorities xml file, invoicexml for PEPPOL xml file

[1] At least one of the above should be defined

Response

Status	Response	Values	Description
200	files	downloadUrl	string Azure storage url. ^[3]
		fileName	string Name of the file uploaded
	status	number	Status of the request. Success =1, Failed =2
	errorCode	string	In case verify request fails, indicates the type of the error.
	errorMessage	string	Detailed description of the error.
	errorCategory	string	Indicates the category of the error. TED = Provider Technical error BED = Provider Business error TTA = Technical error by Tax Authorities BTA = Business error by Tax Authorities
401			Unauthorized request. The jwt is either invalid or expired.

Example

REQUEST	POST /api/getFiles
	<pre>{ "externalSystemId": "185" }</pre>
RESPONSE	HTTP/1.1 200 OK
	<pre>"files": [{ "downloadUrl": "url", "fileName": "myDataXML.xml" }, { "downloadUrl": "url", "fileName": "eInvoiceXML.xml", "fileIdentifier": "eInvoiceXML" }, { "downloadUrl": "url", "fileName": "invoice", "fileIdentifier": "PDF" }], "status": 1, "errorCode": null, "errorMessage": null,</pre>

³ Use the Azure storage url to send a new GET http request ([See here](#)).

Request Syntax: GET <downloadUrl> HTTP/1.1

```
"errorCategory": null
```

2.7 Download publish logs (B2G only)

When the invoice publish type is B2G (2) and the whole process has been successfully completed, the client can download all the logs of the publish process, at any time. The request returns a response status success indicator, together with a list of publish logs.

Request

Method	URL
POST	api/GetPublishLogs
HTTP Header	Key: X-Version Value: 3.0 *required

Parameter	Values	Description
<code>processId</code> ^[1] [optional]	Guid	Unique processId as returned by api/send.
<code>externalSystemId</code> ^[1] [optional]	string	The external identifier provided as input to api/send method

[1] At least one of the above should be defined

Response

Status	Response	Values	Description	
200	<code>publishLogs</code>	<code>message</code>	string	Publish log entry detailed message. Possible error messages described in Appendix 2 .
		<code>status</code>	number	Publish log entry status. Values and aliases: Pending = 0

			<p>AB = 1 (e-Invoice received by KE.D.) AP = 2 (Accepted) RE = 3 (Rejected) IP = 4 (In Process) UQ = 5 (Under Query) CA = 6 (Conditionally Accepted) PD = 7 (Paid)</p> <p>As described in Peppol BIS v3 UNCL4343 Subset.</p>
	timestamp	datetime	The log entry timestamp.
	status	number	Status of the request. Success =1, Failed =2
	errorCode	string	In case request fails, indicates the type of the error.
	errorMessage	string	Detailed description of the error.
	errorCategory	string	Indicates the category of the error. TED = Provider Technical error BED = Provider Business error TTA = Technical error by Tax Authorities BTA = Business error by Tax Authorities
401			Unauthorized request. The jwt is either invalid or expired.

Example

REQUEST	POST /api/GetPublishLogs
<pre>{ "externalSystemId": "185" }</pre>	
RESPONSE	HTTP/1.1 200 OK
<pre>{ "publishLogs": [{ "message": "Το Ηλεκτρονικό Τιμολόγιο έχει δρομολογηθεί προς αποστολή στην Αναθέτουσα αρχή", "status": 0, "timestamp": "2023-09-11T11:15:34.5998309" }, { "message": "Παραλαβή από ΚΕΔ", "status": 1, "timestamp": "2023-09-11T11:15:39.2524312" }], "status": 1, "errorCode": null, "errorMessage": null, "errorCategory": null }</pre>	

2.8 Submit Payment

Besides the requirement for a Provider's Signature upon invoice transmission with EFT-POS payment method specified, there also might be cases of EFT-POS payments occurring after the invoice transmission.

In order to handle such post-transmission EFT-POS payment requirements, this method may be used, which natively follows invoice transmission (API method [Send](#)) in the transmission flow. The method invocation must be combined with the [RequestPayment](#) method, so as to get the required Provider's Signature that verifies the payment at the POS terminal.

Thus, before invoking the `SubmitPayment` method, the user must first invoke the `RequestPayment` method to get the required payment token info (signature, timestamp, amount, uid). When the `RequestPayment` occurs after invoice transmission, the "mark" input parameter is considered required and must not be undefined.

Refer to [2.1 Request Payment](#) for completing the first step required to proceed with `SubmitPayment` method.

Request

Method	URL
POST	/api/submitPayment
HTTP Header	Key: X-Version Value: 3.0 <i>*required</i>

Parameter	Values	Description
<code>externalSystemId</code>	string	The external identifier provided to "send" or "requestPayment" method.
<code>mark</code>	long	Unique registration number of the invoice provided by AADE.
<code>paymentInfo</code>	array	
	<code>type</code>	number Payment method type code. At least one entry must be of type EFT-POS (type=7)
	<code>amount</code>	decimal Payable amount for the payment type specified
	<code>paymentMethodInfo</code> [optional]	string Text description for specified payment
	<code>tipAmount</code> [optional]	decimal An optional extra amount paid through the EFT-POS transaction as a tip. Applicable only for payment method type EFT-POS (type=7)
	<code>transactionId</code>	string Bank transaction identifier returned by the POS terminal. Required only for payment method type EFT-POS (type=7)

	signature	string	Provider's payment signature returned by RequestPayment method. Required only for payment method type EFT-POS (type=7)
--	-----------	--------	--

Response

Status	Response	Values	Description
200	externalSystemId	string	The external identifier provided to "send" or "requestPayment" method.
	processId	guid	The identifier of the invoice.
	paymentMark	string	Payment mark returned by the Greek Tax Authorities.
	status	number	The progress status of the procedure. InProgress = 0, Completed = 1, Failed = 2.
	errorCode	string	In case invoice upload fails, errorCode indicates the type of the error.
	errorMessage	string	Detailed description of the error.
	errorCategory	string	Indicates the category of the error. TED = Provider Technical error BED = Provider Business error TTA = Technical error by Tax Authorities BTA = Business error by Tax Authorities
401			Unauthorized request. The jwt is either invalid or expired.

Example

REQUEST	POST /api/submitPayment
	<pre>{ "mark": 400000018449479, "externalSystemId": "bcadd6ae-e8a6-5r84-g3da-23cbb7vs7641", "paymentInfo": [{ "type": 7, "amount": 100, "paymentMethodInfo": "", "tipAmount": 0, "transactionId": "1234567890987654326", "signature": "6B42CBC7C1CC02724095E6169F178[...]006D75E49FBB90E9FFA4D22944B2BBC02F1E27CFE57C8" }] }</pre>
RESPONSE	HTTP/1.1 200 OK
	<pre>{ "processId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "externalSystemId": "bcadd6ae-e8a6-5r84-g3da-23cbb7vs7641", }</pre>

```

"paymentMark": 400001923846602,
"status": 1,
"errorCategory": null,
"errorCode": null,
"errorMessage": null
}

```

RESPONSE	HTTP/1.1 401 Unauthorized
----------	---------------------------

```

{}

```

2.9 Cancel Payment

In the event of a requested payment token (Provider's Signature) not been used in a bank transaction and the user decides that this token is no longer needed, this method allows cancelling a previously requested payment token.

Request

Method	URL
POST	/api/cancelPayment
HTTP Header	Key: X-Version Value: 3.0 *required

Parameter	Values	Description
externalSystemId	string	The external identifier provided to "send" or "requestPayment" method.
signature	string	Provider's payment signature that will be cancelled.

Response

Status	Response	Values	Description
200	externalSystemId	string	The external identifier provided to "requestPayment" method.
	processId	guid	The identifier of the invoice.
	signature	string	Provider's payment signature.
	status	number	The progress status of the procedure. Completed = 1, Failed = 2.
	errorCode	string	In case invoice upload fails, errorCode indicates the type of the error.
	errorMessage	string	Detailed description of the error.
	errorCategory	string	Indicates the category of the error. TED = Provider Technical error

200	externalSystemId		string	The external identifier provided to "getPayments" method.
	payments		array	
		signature	string	Provider's payment signature.
		timestamp	datetime	Provider's Signature creation timestamp.
		amount	decimal	The payable amount bound to the signature.
		status	number	Payment request status. Pending = 0, Completed = 1, Cancelled = 2, PendingTransmission = 3, ^[4] Expired = 10
		transactionId	String	Bank transaction identifier returned by the POS terminal.
		tipAmount	decimal	Optional tip amount paid during the EFT-POS transaction.
		paymentMark	long	Unique registration number of the submitted payment provided by AADE. ^[4]
		terminalId	string	The Id of the EFT-POS terminal that will perform the payment transaction
		uid	string	Identifier of the invoice required for the signature generation and for completing the EFT-POS payment.
	status		number	The progress status of the procedure. Completed = 1, Failed = 2.
	errorCode		string	In case invoice upload fails, errorCode indicates the type of the error.
	errorMessage		string	Detailed description of the error.
errorCategory		string	Indicates the category of the error. TED = Provider Technical error BED = Provider Business error TTA = Technical error by Tax Authorities BTA = Business error by Tax Authorities	
401				Unauthorized request. The jwt is either invalid or expired.

Example

REQUEST	GET /api/getPayments/bcadd6ae-e8a6-5r84-g3da-23cbb7vs764l
RESPONSE	HTTP/1.1 200 OK
<pre>{ "errorCategory": "string", "errorCode": "string",</pre>	

⁴ Available only when payment is submitted via [2.8 Submit Payment](#)

```

"errorMessage": "string",
"status": 1,
"externalSystemId": "bcadd6ae-e8a6-5r84-g3da-23cbb7vs7641",
"payments": [
  {
    "timestamp": "2024-04-02T18:23:29.261Z",
    "signature":
"6B42CBC7C1CC02724095E6169F178[...]006D75E49FBB90E9FFA4D22944B2BBC02F1E27CFE57C8",
    "amount": 100,
    "status": 1,
    "transactionId": "string",
    "tipAmount": 0,
    "paymentMark": 0,
    "terminalId": "1234567890",
    "uid": "14BFA565194F552B3A3462DE07A48B2E24484BC1"
  }
]
}

```

RESPONSE	HTTP/1.1 401 Unauthorized
----------	---------------------------

```
{}
```

3. Invoice Source Data analysis

The source argument in API call requests describes the content data of the invoice that will be transmitted to Tax Authorities'. It should contain all information required by both Tax Authorities' and e-Invoicing for PEPPOL BIS Billing 3.0 (sequence UBL 2.1).

The basic elements of the invoice are explained in detail in next the sections.

- Invoice issuer
- Invoice counterpart
- Invoice representative, when applicable
- Shipment Address, when applicable
- Basic Invoice Information
- Invoice Lines
- Payment Terms
- Correlated Invoices, when applicable
- Invoice Totals

Tax Authorities' and UBL2.1 also require specific elements to be configured according to given code lists ([Tax Authorities Code lists](#) & [PEPPOL BIS Billing 3.0](#)).

The code lists can be configured either at the external system (e.g EPR) or at the web portal and resolved based on Invoice's details and are listed below.

- Country [ISO 3166-1 alpha-2](#)
- Currency [ISO 4217](#)
- Invoice Type
- VAT category
- VAT Exemption category

- Measurement Units
- Allowances / Charges Categories
- Move Purposes
- Payment Term Codes
- Classification Categories
- Classification Types
- VAT Classification Types

Source should also contain two additional elements in order to handle specific conditions of invoice's transmission, which are:

1. Invoice Transmission to end- customer
2. Invoice Delayed Transmission in case of connectivity issues.

More information about these conditions can be found on Section 3.3 and 3.4.

3.1 Tax Authorities' & UBL2.1 configured in External System

Provided that the integrated external system (e.g. ERP) has embedded configuration of Tax Authorities required information, this can be directly transmitted into the source invoice data structure without any further configuration on the web portal of the system.

A sample source invoice data structure for this intent is described below.

Group	Field	Level	Mandatory	Type	UBL ID	Description
invoice		root		object		Invoice to be transmitted to Tax Authorities' platform
issuer		1	YES	object		Issuer's information
	vatNumber	2	YES	string	BT-31	Issuer's TIN
	branch	2	NO	int		Issuer's Company Branch N/A as registered in Tax Authorities' platform
	country	2	YES	string	BT-40	Issuer's country ISO 3166-1 alpha-2
	city	2	NO	string	BT-37	Issuer's city
	street	2	NO	string	BT-35	Issuer's address street
	streetNumber	2	NO	string	BT-35	Issuer's address street number
	postalCode	2	NO	string	BT-38	Issuer's address postal code
counterpart		1	YES	object		Counterpart's information
	vatNumber	2	YES	string	BT-1 BT-48	Counterpart's TIN
	branch	2	NO	int	BT-1	Counterpart's Company Branch N/A as registered in Tax Authorities' platform
	name	2	YES	string	BT-44	Counterpart's Company Official Title
	country	2	YES	string	BT-55	Counterpart's country ISO 3166-1 alpha-2
	city	2	NO	string	BT-52	Counterpart's city
	street	2	NO	string	BT-50	Counterpart's address street
	streetNumber	2	NO	string	BT-50	Counterpart's address street number
	postalCode ^[2]	2	NO	string	BT-53	Counterpart's address postal code
	email	2	NO ^[1]	string	BT-49	Counterpart's Company contact email
representative		1	NO	object		Tax Representative's details if exists
	vatNumber	2	NO	string	BT-1 BT-63	Representative's TIN

	name	2	NO	string	BT-62	Representative's Title
deliveryAddress		1	NO	object		Shipping address details
	country	2	NO	string	BT-80	Shipping country ISO 3166-1 alpha-2
	city	2	NO	string	BT-77	Shipping address city
	street	2	NO	string	BT-75	Shipping address street
	streetNumber	2	NO	string	BT-75	Shipping address street number
	postalCode	2	NO	string	BT-78	Shipping address postal code
	partyName	2	NO	string	BT-70	The name of the party to which the goods and services are delivered. Required only for B2G Transmission.
invoiceHeader		1	YES	object		Header of the invoice
	series	2	YES	string		Series Code of Invoice issued.
	aa	2	YES	long	BT-1	Invoice Number
	issueDate	2	YES	date	BT-1 BT-2	Invoice Date ISO 8601
	dueDate ^[1]	2	NO	date	BT-9	Payment Due Date ISO 8601 . In case the Amount due for payment is positive, either the Payment due date (BT-9) or Payment terms (BT-20) shall be present.
	paymentTerms ^[1]	2	NO	String	BT-20	A textual description of the payment terms that apply to the amount due for payment (Including description of possible penalties). In case the Amount due for payment is positive, either the Payment due date (BT-9) or the Payment terms (BT-20) shall be present.
	dispatchDate	2	NO	date		Dispatch Date ISO 8601
	invoiceNote	2	NO	string	BT-22	Invoice note/comment as free text
	invoiceCode	2	NO	string		Invoice Code
	invoiceType	2	YES	string	BT-1	Invoice Type according to Tax Authorities' documentation
	invoiceTypeUbl	2	YES	string	BT-3	Invoice Type according to UBL 2.1 documentation
	currency	2	YES	string	BT-5	Currency of Invoice ISO 4217
	selfPricing	2	NO	boolean		Indicates whether the invoice is self-pricing. Default is false
	movePurpose	2	NO	int		Move purpose according to Tax Authorities' documentation
	fuelInvoice	2	NO	Boolean		Indicates whether the invoice is related to fuel ... Default is false
invoiceDetails		1	YES	object		Lines of the invoice
	lineNumber	2	NO	int		Invoice Line N/A
	recType	2	NO	int		Line type according to Tax Authorities' documentation
	quantity	2	YES	decimal	BT-129	Line quantity
	quantity15 ^[2]	2	NO	Decimal		Used only for Fuel Invoices and represents fuel quantity in 15°C
	entityName	2	YES	string	BT-153	Item's Description
	invoiceDetailType	2	NO	int		Invoice Detail Type when necessary according Tax Authorities' documentation (e.g invoice type 1.5)
	fuelCode ^[2]	2	NO	int		Fuel Code according to Tax Authorities' documentation.
	netValue	2	YES	decimal	BT-131	Line net amount

	totalValue	2	NO	decimal		Line total amount
	vatCategory	2	YES	int		Vat Category according to Tax Authorities' documentation
	vatCategoryUbl	2	YES	string	BT-118	Vat Category according to UBL 2.1 documentation
	vatExemption ^[2]	2	NO	int		Vat exemption category according to Tax Authorities' documentation
	vatExemptionUbl ^[2]	2	NO	string	BT-121	Vat Exemption category according to UBL 2.1 documentation
	vatAmount	2	YES	decimal	BT-110	VAT amount
	vatPercent	2	YES	decimal	BT-119	VAT percent
	measurementUnit	2	YES	Int		Measurement Unit according to Tax Authorities' documentation
	measurementUnitUbl	2	YES	string	BT-130	Measurement Unit according to UBL 2.1 documentation
	lineComments	2	NO	string		Comments of line
	classificationCategory	2	NO	string		Classification category according to Tax Authorities' documentation (The system automatically resolves whether it represents income or expense classification)
	classificationType	2	NO	string		Classification type according to Tax Authorities' documentation (The system automatically resolves whether it represents income or expense classification)
	classificationVatType	2	NO	string		VAT classification type of in case of expense classification according to Tax Authorities' documentation
	cpvCode	2	NO	string		Classification code for public procurement. Required only for B2G Transmission. (publishType = 2)
taxInfo		2	NO	object		Charge/allowance applied on line level or charge with VAT in correspondence with "recType".
	taxCategory	3	NO	int		Charge/allowance category according to Tax Authorities' documentation
	taxCategoryUbl	3	NO	string		Charge/allowance category according to UBL 2.1 documentation
	underlyingValue	3	NO	decimal		Total amount subject to a specific charge/allowance category
	taxPercent	3	NO	decimal		Charge/allowance percent applied
paymentMethods		1	YES	List		List of payment Methods applied with different type and portion of net value
	type	2	YES	Int		Payment method code according to Tax Authorities' documentation
	amount	2	YES	decimal		Portion of net amount
	signature ^[2]	2	NO	string		Provider's payment signature returned by RequestPayment method. Required only for payment method type EFT-POS (type=7)
	transactionId ^[2]	2	NO	string		Bank transaction identifier returned by the POS terminal. Required only for payment method type EFT-POS (type=7)
	tipAmount ^[2]	2	NO	decimal		Tip amount paid alongside the payable amount. Required only for payment method type EFT-POS (type=7) when a tip amount is returned by the POS

						terminal.
	terminalId	2	NO	string		The Id of the EFT-POS terminal that will perform the payment transaction. Required only when requesting payment via Send with TransmissionType 4.
	nspCode	2	NO	string		Code identifier of the EFT-POS service provider. Values described in Appendix 3 . Required only when requesting payment via Send with TransmissionType 4.
	taxesTotals ^[2]	1	NO	List		Fees and additional charges or allowances details, when additional charges are included
	taxType	2	NO	int		Type of the charge/allowance according to Tax Authorities' documentation
	taxCategory	2	NO	int		Charge/allowance category according to Tax Authorities' documentation
	taxCategoryUbl	2	NO	string	BT-98 BT-105	Charge/allowance category according to UBL 2.1 documentation
	underlyingValue	2	NO	decimal	BT-93 BT-100	Total amount subject to a specific charge/allowance category
	taxAmount	2	NO	decimal	BT-92 BT-99	Charge/allowance amount
	taxPercent	2		decimal	BT-94 BT-101	Charge/allowance percent applied
	correlatedInvoices ^[2]	1	NO	List		Specific Invoice Types needs correlation between the invoice issued and a previously issued related invoice, e.g. sales invoice-credit invoice
	extSystemId	2	NO	string		Original Invoice id as defined in your ERP System
	mark	2	NO	long		Original Invoice transmission M.AR.K
	invoiceSummary	1	YES	object		Summary of total amounts of the invoice
	totalNetValue	2	NO	decimal	BT-106	Total net amount of the invoice
	totalVatAmount	2	NO	decimal	BT-110	Total VAT amount of the invoice
	totalValue	2	YES	decimal	BT-112	Invoice's Payable Amount
	publishType	1	NO	int		Type of Electronic Invoice Transmission (None = 0, B2B = 1, B2G = 2)
	publishDetails	1	NO	object		Additional Information for B2B and B2G publishing
	contractingAuthorityID	2	NO	string	BT-46	Public Authority Identifier (Required only for B2G)
	contractIdentifier	2	NO	string	BT-12	The Identifier of the contract with the public Institution (Required only for B2G)
	organizationalUnitName	2	NO	string	BT-10	Name of internal organizational unit of public authority that will actually receive the invoice, if any.
	organizationalUnitCode	2	NO	string	BT-10	Code of internal organizational unit of public authority that will actually receive the invoice, if any.
	budget	2	NO	object		
	type	3	NO	int	BT-11	Type of the budget (Regular Budget = 1, e-PDE = 2, Other Budget = 3) (Required only for B2G)
	identifier	3	NO	string		Identifier number of the budget (Required only for B2G)

messages		1	NO	object		Recipients information & type of message template
	type	2	YES	int		Type of message to be sent (Email = 0, SMS = 1, Viber = 2)
	recipients	2	YES	string		Email or mobile number of the recipients separated by ";"
	cc	2	NO	string		Cc emails separated by ";"
	templatedentifier	2	NO	string		Identifier of message template, when not included the default message template is applied.
delayedProcessCode ^[2]		1	NO	string		In case of real time transmission failure, authenticated code must be included, as it was provided at issue to mobile application

^[1] According to [PEPPOL BIS Billing 3.0](#)

^[2] Mandatory under specific conditions according to Tax Authorities' documentation

json example

```
{
  "invoice": {
    "issuer": {
      "vatNumber": "00000000",
      "branch": 0,
      "street": "street",
      "streetNumber": "streetNumber",
      "postalCode": "postalCode",
      "city": "city",
      "country": "GR"
    },
    "counterpart": {
      "vatNumber": "00000000",
      "branch": 0,
      "name": "name",
      "street": "street",
      "streetNumber": "streetNumber",
      "postalCode": "postalCode",
      "city": "city",
      "country": "GR",
      "email": "test@email.com"
    },
    "representative": {
      "vatNumber": "00000000",
      "name": "name"
    },
    "deliveryAddress": {
      "street": "street",
      "streetNumber": "streetNumber",
      "postalCode": "postalCode",
      "city": "city",
      "country": "GR"
    },
    "publishType": 2,
    "publishDetails": {
      "contractingAuthorityID": "1000.E00904.0001",
      "budget": {
        "type": 1,
        "identifier": "2018EP06600057"
      },
      "contractIdentifier": "20SYMV006467658"
    },
    "organizationalUnitName": "Sample Org Unit",
    "organizationalUnitCode": "123456"
  },
  "invoiceHeader": {
    "series": "INV",
    "aa": 1,
    "issueDate": "yyyy-MM-dd",
    "dispatchDate": "yyyy-MM-dd",
    "invoiceNote": "Free text comment",
    "invoiceCode": "INV-00001",
    "invoiceType": "1.1",
    "invoiceTypeUbl": "80",
    "currency": "EUR",
    "selfPricing": false,
    "movePurpose": 1,
    "fuelInvoice": false
  },
  "invoiceDetails": [
    {
      "lineNumber": 1,
      "recType": 1,
      "quantity": 1,
      "entityName": "string",
      "invoiceDetailType": 0,
      "netValue": 50,
      "totalValue": 62,
      "vatCategory": 1,
      "vatCategoryUbl": "Z",
      "vatExemption": 16,
      "vatExemptionUbl": "VATEX-EU-79",
      "vatAmount": 12,
      "vatPercent": 24,
      "measurementUnit": 1,
      "measurementUnitUbl": "E48",
      "lineComments": "string",
      "classificationCategory": "string",
      "classificationType": "string",
      "classificationVatType": "string",
      "cpvCode": "45233222-1",
      "taxInfo": {

```


3.2 Tax Authorities' & UBL2.1 configured in web portal

In case the integrated external system (e.g. ERP) is agnostic of Tax Authorities code lists, source invoice data structure should contain the necessary native values so that the mapping configuration, to the corresponding Tax Authorities structures, can be performed on web portal.

A sample source invoice data structure for this intent is described below.

Group	Field	Level	Mandatory	Type	UBL ¹⁾	Description
invoice		root		object		Invoice to be transmitted to Tax Authorities' platform
issuer		1	YES	object		Issuer's information
	vatNumber	2	YES	string	BT-31	Issuer's TIN
	branch	2	NO	int		Issuer's Company Branch N/A as registered in Tax Authorities' platform
	country	2	YES	string	BT-40	Issuer's country ISO 3166-1 alpha-2
	city	2	NO	string	BT-37	Issuer's city
	street	2	NO	string	BT-35	Issuer's address street
	streetNumber	2	NO	string	BT-35	Issuer's address street number
	postalCode	2	NO	string	BT-38	Issuer's address postal code
counterpart		1	YES	object		Counterpart's information
	vatNumber	2	YES	string	BT-1 BT-48	Counterpart's TIN
	branch	2	NO	int	BT-1	Counterpart's Company Branch N/A as registered in Tax Authorities' platform
	name	2	YES	string	BT-44	Counterpart's Company Official Title
	country	2	YES	string	BT-55	Counterpart's country ISO 3166-1 alpha-2
	city	2	NO	string	BT-52	Counterpart's city
	street	2	NO	string	BT-50	Counterpart's address street
	streetNumber	2	NO	string	BT-50	Counterpart's address street number
	postalCode ²⁾	2	NO	string	BT-53	Counterpart's address postal code
	email	2	NO ¹⁾	string	BT-49	Counterpart's Company contact email
representative		1	NO	object		Tax Representative's details if exists
	vatNumber	2	NO	string	BT-1 BT-63	Representative's TIN
	name	2	NO	string	BT-62	Representative's Title
deliveryAddress		1	NO	object		Shipping address details
	country	2	NO	string	BT-80	Shipping country ISO 3166-1 alpha-2
	city	2	NO	string	BT-77	Shipping address city
	street	2	NO	string	BT-75	Shipping address street
	streetNumber	2	NO	string	BT-75	Shipping address street number
	postalCode	2	NO	string	BT-78	Shipping address postal code
	partyName	2	NO	string	BT-70	The name of the party to which the goods and services are delivered. Required only for B2G Transmission.
invoiceHeader		1	YES	object		Header of the invoice
	series	2	YES	string	BT-1 BT-3	Series Code of Invoice issued, use to resolve invoice type.
	aa	2	YES	long	BT-1	Invoice Number
	issueDate	2	YES	date	BT-1 BT-2	Invoice Date ISO 8601
	dueDate ¹⁾	2	NO	date	BT-9	Payment Due Date ISO 8601 . In case the

						Amount due for payment is positive, either the Payment due date (BT-9) or the Payment terms (BT-20) shall be present.
	paymentTerms ^[1]	2	NO	String	BT-20	A textual description of the payment terms that apply to the amount due for payment (Including description of possible penalties). In case the Amount due for payment is positive, either the Payment due date (BT-9) or the Payment terms (BT-20) shall be present.
	dispatchDate	2	NO	date		Dispatch Date ISO 8601
	invoiceCode	2	NO	string		Invoice Code
	currency	2	YES	string	BT-3	Currency of Invoice ISO 4217
	selfPricing	2	NO	boolean		Indicates whether the invoice is self-pricing. Default is false
	movePurpose	2	NO	string		Move purpose as configured in ERP
	fuelInvoice	2	NO	Boolean		Indicates whether the invoice is related to fuel... Default is false
invoiceDetails		1	YES	object		Lines of the invoice
	lineNumber	2	NO	int		Invoice Line N/A
	lineType	2	NO	string		Line descriptor property in order to apply specific behavior to the line
	quantity	2	YES	decimal	BT-129	Line quantity
	quantity15 ^[2]	2	NO	decimal		Used only for Fuel Invoices and represents fuel quantity in 15°C
	entityName	2	YES	string	BT-153	Item's Description
	netValue	2	YES	decimal	BT-131	Line net amount
	totalValue	2	NO	decimal		Line total amount
	vatCategory	2	YES	string	BT-118	Vat Category code as configured in ERP System
	vatExemption ^[2]	2	NO	string	BT-121	Vat Exemption as configured in ERP System
	vatAmount	2	YES	decimal	BT-110	VAT amount
	vatPercent	2	YES	decimal	BT-119	VAT percent
	measurementUnit	2	YES	string	BT-130	Measurement Unit as configured in ERP System
	lineComments	2	NO	string		Comments of line
	fuelCode ^[2]	2	NO	int		Fuel Code according to Tax Authorities' documentation.
taxInfo		2	NO	object		Charge/allowance applied on line level or charge with VAT in correspondence to line's behavior.
	taxCategory	3	NO	string		Charge/allowance category as configured in ERP System
	underlyingValue	3	NO	decimal		Total amount subject to a specific charge/allowance category
	taxPercent	3	NO	decimal		Charge/allowance percent applied
paymentMethods		1	YES	List		List of payment Methods applied with different type and portion of net value
	type	2	YES	int		Payment Method Code as configured in ERP
	amount	2	YES	decimal		Portion of net amount
	signature ^[2]	2	NO	string		Provider's payment signature returned by RequestPayment method. Required only for payment method type EFT-POS (type=7)

	transactionId ^[2]	2	NO	string		Bank transaction identifier returned by the POS terminal. Required only for payment method type EFT-POS (type=7)
	tipAmount ^[2]	2	NO	decimal		Tip amount paid alongside the payable amount. Required only for payment method type EFT-POS (type=7) when a tip amount is returned by the POS terminal.
	terminalId	2	NO	string		The Id of the EFT-POS terminal that will perform the payment transaction. Required only when requesting payment via Send with TransmissionType 4.
	nspCode	2	NO	string		Code identifier of the EFT-POS service provider. Values described in Appendix 3 . Required only when requesting payment via Send with TransmissionType 4.
taxesTotals ^[2]		1	NO	List		Fees and additional charges or allowances details, when additional charges are included
	taxType	2	NO	string		Type of the charge/allowance as configured in ERP System
	taxCategory	2	NO	string		Charge/allowance category as configured in ERP System
	underlyingValue	2	NO	decimal		Total amount subject to a specific charge/allowance category
	taxAmount	2	NO	decimal		Charge/allowance amount
	taxPercent	2	NO	decimal		Charge/allowance percent applied
correlatedInvoices ^[2]		1	NO	List		Specific Invoice Types needs correlation between the invoice issued and a previously issued related invoice, e.g. sales invoice-credit invoice
	extSystemId	2	NO	string		Original Invoice id as defined in your ERP System
	mark	2	NO	long		Original Invoice transmission M.AR.K
invoiceSummary		1	YES	object		Summary of total amounts of the invoice
	totalNetValue	2	NO	decimal	BT-106	Total net amount of the invoice
	totalVatAmount	2	NO	decimal	BT-110	Total VAT amount of the invoice
	totalValue	2	YES	decimal	BT-112	Invoice's Payable Amount
publishType		1	NO	int		Type of Electronic Invoice Transmission (None = 0, B2B = 1, B2G = 2)
publishDetails		1	NO	object		Additional Information for B2B and B2G publishing
	contractingAuthorityID	2	NO	string	BT-46	Public Authority Identifier (Required only for B2G)
	contractIdentifier	2	NO	string	BT-12	The Identifier of the contract with the public Institution (Required only for B2G)
budget		2	NO	object		
	type	3	NO	int	BT-11	Type of the budget (Regular Budget = 1, e-PDE = 2, Other Budget = 3) (Required only for B2G)
	identifier	3	NO	string		Identifier number of the budget (Required only for B2G)
messages		1	NO	object		Recipients information & type of message template
	type	2	YES	int		Type of message to be sent (Email = 0, SMS = 1, Viber = 2)
	recipients	2	YES	string		Email or mobile number of the recipients separated by ";"
	cc	2	NO	string		Cc emails separated by ";"
	templateIdentifier	2	NO	string		Identifier of message template, when not included the default message template is applied.
delayedProcessCode ^[2]		1	NO	string		In case of real time transmission failure, authenticated code must be included, as it was provided at issue to the mobile application

^[1] According to [PEPPOL BIS Billing 3.0](#)

^[2] Mandatory under specific conditions according to Tax Authorities' documentation

json example

```
{
  "invoice": {
    "issuer": {
      "vatNumber": "00000000",
      "branch": 0,
      "street": "street",
      "streetNumber": "streetNumber",
      "postalCode": "postalCode",
      "city": "city",
      "country": "GR"
    },
    "counterpart": {
      "vatNumber": "00000000",
      "branch": 0,
      "name": "name",
      "street": "street",
      "streetNumber": "streetNumber",
      "postalCode": "postalCode",
      "city": "city",
      "country": "GR",
      "email": "test@email.com"
    },
    "representative": {
      "vatNumber": "00000000",
      "name": "name"
    },
    "deliveryAddress": {
      "street": "street",
      "streetNumber": "streetNumber",
      "postalCode": "postalCode",
      "city": "city",
      "country": "GR"
    },
    "publishType": 2,
    "publishDetails": {
      "contractingAuthorityID": "1000.E00904.0001",
      "budget": {
        "type": 1,
        "identifier": "2018EP06600057"
      },
      "contractIdentifier": "20SYMV006467658"
    },
    "invoiceHeader": {
      "series": "INV",
      "aa": 1,
      "issueDate": "yyyy-MM-dd",
      "dispatchDate": "yyyy-MM-dd",
      "invoiceCode": "INV-00001",
      "currency": "EUR",
      "selfPricing": false,
      "movePurpose": "movePurpose1",
      "fuelInvoice": false
    },
    "invoiceDetails": [
      {
        "lineNumber": 1,
        "lineType": "1",
        "quantity": 1,
        "entityName": "string",
        "netValue": 50,
        "totalValue": 62,
        "vatCategory": "VT01",
        "vatExemption": "EX01",
        "vatAmount": 12,
        "vatPercent": 24,
        "measurementUnit": "PCS",
        "lineComments": "string",
        "cpvCode": "45233222-1",
        "taxInfo": {
          "taxCategory": "category_1",
          "underlyingValue": 1,
          "taxPercent": 20
        }
      }
    ],
    "taxesTotals": [
      {
        "taxType": "1",
        "taxCategory": "category_1",
        "underlyingValue": 1,
        "taxAmount": 10,
      }
    ]
  }
}
```



```

    "taxPercent": 20
  }
],
"paymentMethods": [
  {
    "type": 5,
    "amount": 0
  }
],
"correlatedInvoices": [
  {
    "extSystemId": "string",
    "mark": 0,
  }
],
"invoiceSummary": {
  "totalNetValue": 50,
  "totalVatAmount": 12,
  "totalValue": 52
},
"Messages": [
  {
    "type": 0,
    "recipients": "test1@email.com;test2@email.com",
    "cc": "test3@email.com;test4@email.com",
    "templateIdentifier": "000001"
  },
  {
    "type": 1,
    "recipients": "6983022568;6983022868"
  }
]
}

```

3.3 Invoice Transmission to end customer

The system provides a service of sending personalized invoice information and the signed invoice file to a list of end customer's contacts via email, SMS and Viber.

The messages are defined as follows.

1. UI configuration. In this case the recipients can be configured through "Contacts" page and automatically recognized by their Tax Identification Number. More than one message templates can be assigned to each contact (recipient).
2. Identify invoice recipients on transmission via API call. The "messages" entity should be included in the "source" argument as described above.

Example of using "messages".

```

"Messages": [
  {
    "type": 0
    "recipients": "test1@email.com;test2@email.com",
    "cc": "test3@email.com;test4@email.com",
    "templateIdentifier": "000001"
  },
  {
    "type": 1,
    "recipients": "6983022568;6983022868"
  }
]

```

Please note that the recipients specified via API override any configuration. Both cases requires messages templates to be set through UI configuration for each of the sending modes.

3.4 Invoice Delayed Transmission

Every invoice issued should be transmitted to Tax Authorities in real time. In case of connectivity failure different steps should be followed depending on the source of the problem and the implementation approach decided.

- Invoice file is generated by the External System.
 - Internet Failure. Invoice can be issued off-line after receiving approval via mobile application. After resolving the problem, invoice at hand must be transmitted and source data structure should include the approved code in "delayedProcessCode" attribute.
 - Transmission to Tax Authorities failure. Transmission's polling process will return specific error codes indicating the source of the problem. Invoice can be issued off-line after receiving approval via the mobile application. The invoice at hand must be re-transmitted and source data structure should include the approved code in "delayedProcessCode" attribute.
- Invoice file is generated by the system.
 - Internet Failure. Invoice can be issued off-line after receiving approval via the mobile application. After resolving the problem, invoice at hand must be transmitted and source data structure should include the approved code in "delayedProcessCode" attribute.
 - Transmission to Tax Authorities failure. Transmission's polling process will return specific error codes that indicate the source of the problem. According to issuer's configuration the process will automatically stop or proceed without signing. In case of the latter, once the problem is resolved the invoice at hand will automatically be re-transmitted to Tax Authorities with the corresponding indication.

Appendix 1

Operational Error Codes

Operational errors occur when a procedure does not come as expected and fails. Data validation and connectivity issues could cause an operation to fail. In these cases the call is considered technically successful and the HTTP response status is 200 (OK).

The response of the API call contains the “errorCode”, the “errorCategory” the “errorMessage” and the “errorSeverity” which describe the cause of the failure.

Error Code	Error Category	Method	Description
00010003	Technical ED	All	Subscription information is not valid
00080002	Business ED	All	ProcessId or ExternalSystemId is required
00080003	Business ED	All	The input provided does not correspond to a system registration
00010002	Business ED	All	Input data is not valid
00010004	Business ED	Get	Failure while retrieving invoice information
00010005	Business ED	Get	Invoice not available.
00010011	Business ED	Send	Invoice creation failed
00010014	Business ED	Get	Errors while processing invoice source data
00010015	Business ED	Get	Errors while validating invoice source data
00010020	Business ED	Get	Errors while mapping invoice source data to template
00010022	Business ED	Get	Invoice with the same invoice code already submitted to Tax Authorities platform
00010028	Business ED	Finalize	Failure while verifying pdf file uploaded
00010029	Business TA	Get	Tax Authorities business errors. Tax Authorities error codes are included in “errorMessage”
00010030	Technical TA	Get	Tax Authorities technical errors. Tax Authorities error codes are included in “errorMessage”
00010032	Technical TA	Get	Tax Authorities connection failed
00010033	Business ED	Get	Errors while processing invoice source data for UBL 2.1

00010034	Business ED	Get	Errors while processing invoice source data for UBL 2.1
00010037	Technical ED	All	Subscription is locked
00010039	Technical ED	Get	Print form could not be resolved. Only for transmissionType = 2.
00010040	Technical TA	Get	Transmission is currently suspended, due to technical issue on Tax Authorities platform
00010041	Business ED	Get	Document information related to delayed transmission is not valid
00010042	Business ED	GetFiles	Failure while getting files
00010043	Business ED	UploadFileRequest	Failure while requesting file SAS URI
00010050	Business ED	UploadFileRequest	File already exists
00010052	Technical ED	All	License failure by unauthorized user
00010053	Technical ED	Get	Issuer's Tax Identification Number validation failed

Appendix 2

B2G Validation Error Messages

When the invoice publish type is B2G (2), a separate validation process takes place in order to verify that data conform to PEPPOL rules. Error messages are fixed (Greek language only) and provided by KED.

Technical Message	User Message
ΑΠΟΡΡΙΨΗ ΕΙΔΟΣ ΠΡΟΫΠΟΛΟΓΙΣΜΟΥ (Project_Reference)	ΑΠΟΡΡΙΨΗ ΕΙΔΟΥΣ ΠΡΟΫΠΟΛΟΓΙΣΜΟΥ (PROJECT REFERENCE)
ΑΠΟΡΡΙΨΗ ΑΔΑΜ	ΑΠΟΡΡΙΨΗ ΑΔΑΜ
ΑΠΟΡΡΙΨΗ ΕΝΑΡΙΘΜΟΣ	ΑΠΟΡΡΙΨΗ ΕΝΑΡΙΘΜΟΣ
ΓΕΝΙΚΗ ΑΠΟΡΡΙΨΗ ΚΕΔ	ΓΕΝΙΚΗ ΑΠΟΡΡΙΨΗ ΚΕΔ
ΤΟ ΠΕΔΙΟ INVOICE.INVOICE_NUMBER ΕΙΝΑΙ NULL (BT-1)	ΔΕΝ ΑΝΑΦΕΡΕΤΑΙ ΑΡΙΘΜΟΣ ΤΙΜΟΛΟΓΙΟΥ
ΤΟ ΠΕΔΙΟ INVOICE.PROJECT_REFERENCE_ID ΕΙΝΑΙ	ΔΕΝ ΑΝΑΦΕΡΕΤΑΙ ΣΤΟΙΧΕΙΟ ΑΝΑΦΟΡΑΣ

NULL (BT-11)	ΑΓΑΘΟΥ/ΕΡΓΟΥ (ΤΥΠΟΣ ΠΡΟΥΠΟΛΟΓΙΣΜΟΥ)
ΤΟ ΠΕΔΙΟ INVOICE.CONTRACT_REFERENCE_ID ΕΙΝΑΙ NULL (BT-12)	ΔΕΝ ΑΝΑΦΕΡΕΤΑΙ ΣΤΟΙΧΕΙΟ ΑΝΑΦΟΡΑΣ ΣΥΜΒΑΣΗΣ
ΤΟ ΠΕΔΙΟ SELLER_TAX_SCHEME.TAX_REGISTRATION_ID ΕΙΝΑΙ NULL (BT-31)	ΔΕΝ ΑΝΑΦΕΡΕΤΑΙ ΑΝΑΓΝΩΡΙΣΤΙΚΟ ΦΠΑ ΠΩΛΗΤΗ
ΤΟ ΠΕΔΙΟ SELLER.ELECTRONIC_ADDRESS_ID ΕΙΝΑΙ NULL (BT-34)	ΔΕΝ ΑΝΑΦΕΡΕΤΑΙ ΗΛΕΚΤΡΟΝΙΚΗ ΔΙΕΥΘΥΝΣΗ ΠΩΛΗΤΗ
ΤΟ ΠΕΔΙΟ BUYER.PARTY_ID ΕΙΝΑΙ NULL (BT-46)	ΔΕΝ ΑΝΑΦΕΡΕΤΑΙ ΚΩΔΙΚΟΣ ΑΝΑΘΕΤΟΥΣΑΣ ΑΡΧΗΣ
ΤΟ ΠΕΔΙΟ BUYER.TAX_REGISTRATION_ID ΕΙΝΑΙ NULL (BT-48)	ΔΕΝ ΑΝΑΦΕΡΕΤΑΙ ΑΝΑΓΝΩΡΙΣΤΙΚΟ ΦΠΑ ΑΓΟΡΑΣΤΗ
ΤΟ ΠΕΔΙΟ BUYER.ELECTRONIC_ADDRESS_ID ΕΙΝΑΙ NULL (BT-49)	ΔΕΝ ΑΝΑΦΕΡΕΤΑΙ ΗΛΕΚΤΡΟΝΙΚΗ ΔΙΕΥΘΥΝΣΗ ΑΓΟΡΑΣΤΗ
ΤΟ ΠΕΔΙΟ DELIVERY.DELIVER_TO_PARTY ΕΙΝΑΙ NULL (BT-70)	ΔΕΝ ΑΝΑΦΕΡΕΤΑΙ ΟΝΟΜΑ ΠΑΡΑΛΑΜΒΑΝΟΝΤΟΣ ΜΕΡΟΥΣ
ΤΟ ΠΕΔΙΟ DELIVERY.ADDRESS_LINE_1 ΕΙΝΑΙ NULL (BT-75)	ΔΕΝ ΑΝΑΦΕΡΕΤΑΙ ΔΙΕΥΘΥΝΣΗ ΠΑΡΑΔΟΣΗΣ
ΤΟ ΠΕΔΙΟ DELIVERY.LOCATION_CITY ΕΙΝΑΙ NULL (BT-77)	ΔΕΝ ΑΝΑΦΕΡΕΤΑΙ ΠΟΛΗ ΠΑΡΑΔΟΣΗΣ
ΤΟ ΠΕΔΙΟ DELIVERY.LOCATION_POST_CODE ΕΙΝΑΙ NULL (BT-78)	ΔΕΝ ΑΝΑΦΕΡΕΤΑΙ ΤΑΧΥΔΡΟΜΙΚΟΣ ΚΩΔΙΚΑΣ ΠΑΡΑΔΟΣΗΣ
ΤΟ ΠΕΔΙΟ TAX_TOTAL.VAT_AMOUNT ΕΙΝΑΙ NULL (BT-110)	ΔΕΝ ΑΝΑΦΕΡΕΤΑΙ ΣΥΝΟΛΙΚΟ ΠΟΣΟ ΦΠΑ ΕΓΓΡΑΦΟΥ
ΤΟ ΠΕΔΙΟ ITEM_CLASSIFICATION.CLASSIFICATION_ID ΕΙΝΑΙ NULL (BT-158)	ΔΕΝ ΑΝΑΦΕΡΕΤΑΙ ΑΝΑΓΝΩΡΙΣΤΙΚΟ ΤΑΞΙΝΟΜΗΣΗΣ ΣΤΟΙΧΕΙΟΥ
ΤΟ ΠΕΔΙΟ INVOICE.INVOICE_NUMBER ΔΕΝ ΕΙΝΑΙ ΕΓΚΥΡΟ (BT-1)	Ο ΑΡΙΘΜΟΣ ΤΙΜΟΛΟΓΙΟΥ ΔΕΝ ΕΙΝΑΙ ΕΓΚΥΡΟΣ
ΤΟ ΠΕΔΙΟ BUYER.TAX_REGISTRATION_ID ΔΕΝ ΕΙΝΑΙ ΕΓΚΥΡΟ (BT-48)	Ο ΦΠΑ ΑΓΟΡΑΣΤΗ ΔΕΝ ΥΠΑΡΧΕΙ ΣΤΟ ΦΟΡΟΛΟΓΙΚΟ ΜΗΤΡΩΟ
ΤΟ ΠΕΔΙΟ INVOICE_ARG.SELLER_TAX_REPR_TAX_REG_ID ΔΕΝ ΕΙΝΑΙ ΕΓΚΥΡΟ (BT-63)	Ο ΦΠΑ ΦΟΡΟΛΟΓΙΚΟΥ ΑΝΤΙΠΡΟΣΩΠΟΥ ΔΕΝ ΥΠΑΡΧΕΙ ΣΤΟ ΦΟΡΟΛΟΓΙΚΟ ΜΗΤΡΩΟ
ΤΟ ΠΕΔΙΟ INVOICE_ARG.SELLER_TAX_SCHEME_TBL.TAX_REGISTRATION_ID ΔΕΝ ΕΙΝΑΙ ΕΓΚΥΡΟ (BT-31)	Ο ΦΠΑ ΠΩΛΗΤΗ ΔΕΝ ΥΠΑΡΧΕΙ ΣΤΟ ΦΟΡΟΛΟΓΙΚΟ ΜΗΤΡΩΟ
ΤΟ ΠΕΔΙΟ BUYER.TAX_REGISTRATION_ID ΔΕΝ ΥΦΙΣΤΑΤΑΙ (BT-48)	Ο ΦΠΑ ΑΓΟΡΑΣΤΗ ΔΕΝ ΥΠΑΡΧΕΙ ΣΤΗ ΛΙΣΤΑ ΑΝΑΘΕΤΟΥΣΩΝ ΑΡΧΩΝ
ΤΟ ΠΕΔΙΟ BUYER.PARTY_ID ΔΕΝ ΥΦΙΣΤΑΤΑΙ (BT-46)	Ο ΚΩΔΙΚΟΣ ΑΝΑΘΕΤΟΥΣΑΣ ΑΡΧΗΣ ΔΕΝ

	ΥΠΑΡΧΕΙ ΣΤΗ ΛΙΣΤΑ ΑΝΑΘΕΤΟΥΣΩΝ ΑΡΧΩΝ
ΤΟ ΠΕΔΙΟ INVOICE.CONTRACT_REFERENCE_ID ΔΕΝ ΥΦΙΣΤΑΤΑΙ (BT-12)	ΤΟ ΣΤΟΙΧΕΙΟ ΑΝΑΦΟΡΑΣ ΣΥΜΒΑΣΗΣ ΔΕΝ ΥΦΙΣΤΑΤΑΙ
ΤΟ ΠΕΔΙΟ MARK ΔΕΝ ΥΠΑΡΧΕΙ (BT-122)	ΤΟ ΠΕΔΙΟ MARK ΔΕΝ ΥΠΑΡΧΕΙ
ΤΟ ΠΕΔΙΟ AUTH CODE ΔΕΝ ΥΠΑΡΧΕΙ (BT-122)	ΤΟ ΠΕΔΙΟ AUTH CODE ΔΕΝ ΥΠΑΡΧΕΙ
ΤΟ ΠΕΔΙΟ MARK ΔΕΝ ΕΙΝΑΙ ΕΓΚΥΡΟ (BT-122)	ΤΟ ΠΕΔΙΟ MARK ΔΕΝ ΕΙΝΑΙ ΕΓΚΥΡΟ
ΤΟ ΠΕΔΙΟ AUTH CODE ΔΕΝ ΕΙΝΑΙ ΕΓΚΥΡΟ (BT-122)	ΤΟ ΠΕΔΙΟ AUTH CODE ΔΕΝ ΕΙΝΑΙ ΕΓΚΥΡΟ
ΤΟ UID ΔΕΝ ΕΙΝΑΙ ΕΓΚΥΡΟ (BT-122)	ΤΟ UID ΔΕΝ ΕΙΝΑΙ ΕΓΚΥΡΟ
ΔΕΝ ΒΡΕΘΗΚΕ ΑΝΤΙΣΤΟΙΧΟ ΧΡΕΩΣΤΙΚΟ (BILLING REF)	ΔΕΝ ΒΡΕΘΗΚΕ ΤΟ ΑΝΤΙΣΤΟΙΧΟ ΧΡΕΩΣΤΙΚΟ ΤΙΜΟΛΟΓΙΟ
ΔΕΝ ΥΠΑΡΧΕΙ ΠΛΗΡΟΦΟΡΙΑ ΣΥΣΧΕΤΙΣΗΣ ΜΕ ΤΟΝ ΠΡΟΫΠΟΛΟΓΙΣΜΟ (ΜΗ ΣΥΣΧ. ΠΙΣΤΩΤΙΚΑ Η BILLING REFERENCE)	ΔΕΝ ΥΠΑΡΧΕΙ ΠΛΗΡΟΦΟΡΙΑ ΣΥΣΧΕΤΙΣΗΣ ΜΕ ΤΟΝ ΠΡΟΫΠΟΛΟΓΙΣΜΟ (ΔΕΝ ΥΠΑΡΧΕΙ ΔΥΝΑΤΟΤΗΤΑ ΔΡΟΜΟΛΟΓΗΣΗΣ)
ΛΑΝΘΑΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΑ ΣΥΣΧΕΤΙΣΗΣ ΜΕ ΠΡΟΫΠΟΛΟΓΙΣΜΟ (PROJECT_REFERENCE_ID)	ΛΑΝΘΑΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΑ ΣΥΣΧΕΤΙΣΗΣ ΜΕ ΠΡΟΫΠΟΛΟΓΙΣΜΟ
ΑΦΜ ΑΓΟΡΑΣΤΗ ΚΑΙ ΚΩΔΙΚΟΣ ΑΝΑΘΕΤΟΥΣΑΣ ΜΗ ΣΥΜΒΑΤΑ (BUYER_PARTY_ID, TAX_REGISTRATION_ID)	ΑΦΜ ΑΓΟΡΑΣΤΗ ΚΑΙ ΚΩΔΙΚΟΣ ΑΝΑΘΕΤΟΥΣΑΣ ΜΗ ΣΥΜΒΑΤΑ
ΤΟ ID ΤΟΥ ΤΙΜΟΛΟΓΙΟΥ ΠΡΕΠΕΙ ΝΑ ΕΧΕΙ ΕΛΛΗΝΙΚΟ FORMAT (BT-1)	ΤΟ ID ΤΟΥ ΤΙΜΟΛΟΓΙΟΥ ΠΡΕΠΕΙ ΝΑ ΕΧΕΙ ΕΛΛΗΝΙΚΟ FORMAT

Appendix 3

EFT-POS Service Providers

Code	Provider
1	CARDLINK
2	EURONET
3	MELLON
4	NEXI
5	EDPS
6	VIVA
98	Other (HEX) – Other Provider using HEX encoding on Provider's Signature
99	Other (Base64) - Other Provider using Base64 encoding on Provider's Signature