# Jinius API - Technical Guide

## Introduction

Welcome to the technical guide of the Jinius API.

The aim of this document is to be used along with the API's Postman collection and documentation, in order to provide instructions and examples on how to use the API.

The document covers various aspects of the API starting from general actions and features of the API and moving towards more domain specific actions.

## General API guidelines

### Authentication

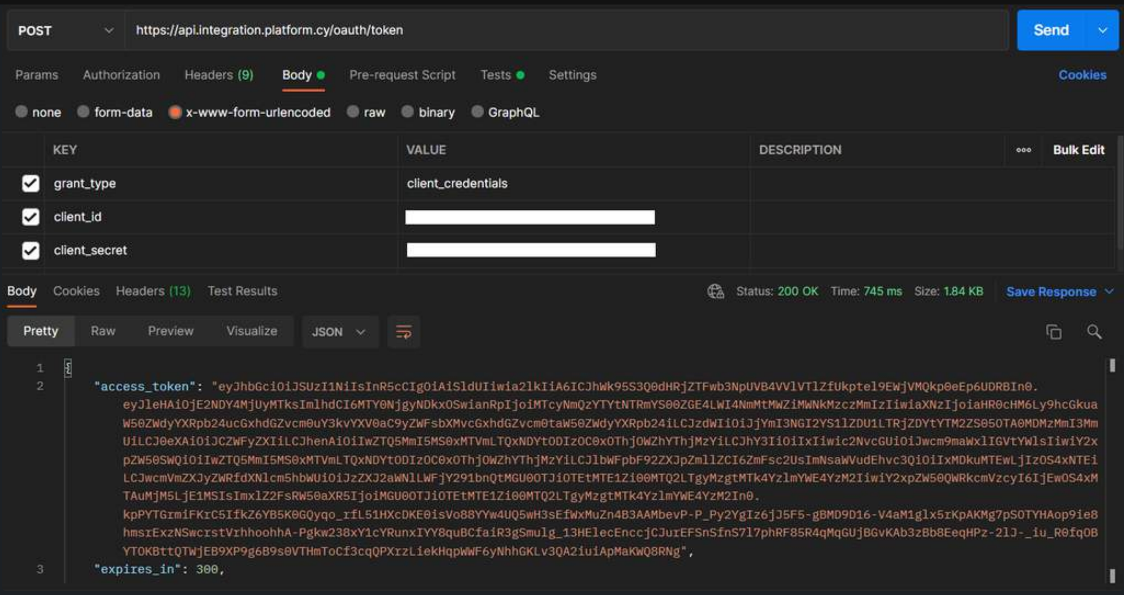
The APIs use OAuth 2.0 Client Credentials Grant to authenticate.

In order to access the Integrations API, clients must provide an **Authorization Header with a Bearer JWT Token.**

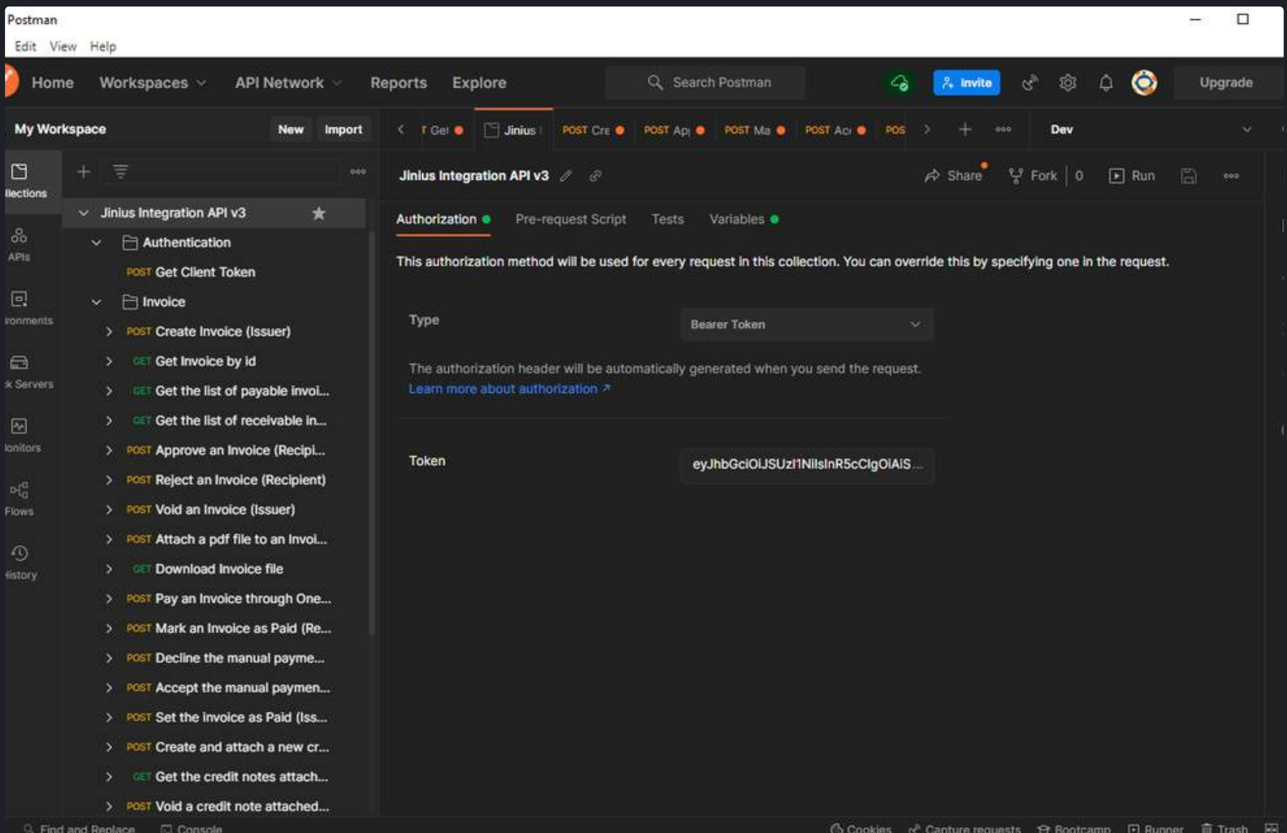
In order to obtain this token, clients first need to set up the **Client Id and Client Secret** for their Legal Entity. This can be set up in the **Developer Settings** page.


Then clients can call the **OAuth endpoint of the API** (please check the Postman collection for more details) providing these values in order to obtain the JWT Token.

*(Bear in mind the the token expires after some time, indicated in the endpoint's response (in seconds))*




In every subsequent request to the API, this token needs to be provided in the **Authorization Header** as a **Bearer Token**. To do so in Postman, edit the ‘Token’ value at the root of the collection and under the ‘Authorization’ tab.




Using the access token

## Pagination

To use pagination when retrieving the list of objects, the client needs to provide the ***page*** and the ***page size*** values, indicating the desired number of page and the number of results in the page. (ie page 2 and page size 5 will bring back results 6 - 10)

Along with the paginated results, the total number of results and pages is returned to the client, along with the number of the current page, in a *pageResponse* object.

```
"pageResponse": {
        "currentPage": 1,
        "totalPages": 35,
        "totalItems": 173
    }
```

If pagination fields are missing (ie both page and page size fields are not provided) then the API will respond with the first page of the result set with the page size set to 50.

*Please be aware that it is advised to **use the same page size between your requests**. This is in order to always get the next/previous subset of results regardless of the page size.*

If for example a request is sent to the API for the 2nd page with page size 25, the API will respond with items 26-50 of the result set. A request (using the same filters) for page 1 with page size 50, will return items 0-50 of the result set, hence including the same results (26-50) as the previous request and not items 51-100.

### Sorting

To sort the returned list of object in a specific order, clients must provide the ***orderBy*** and ***sortOrder*** values, indicating the field on which the sorting will occur and the type of ordering ('Asc' or 'Desc' for ascending or descending order respectively). The values are provided as string values. If the field requested in the orderBy fields is not valid, the service will respond with an error message.

### File Downloads

When downloading a file using the Jinius API, the service responds with the following format.

```
{
  "files": [
    {
      "url": "...",
      "id": "...",
      "filename": "..."
    }
  ]
}
```

The url returned is redirection link to the url that clients should visit in order to download the file.

*(Please be aware that the urls have an expiration time)*

## Errors

The API returns errors to the calling clients in the following format, which follows the GRPC struct format.

```
{
    "StatusCode": "InvalidArgument",
    "Detail": "Input validation error - 'VatAmount' must be greater or equal to zero"
}
```

All error messages related to the request will included in the error message response, separated by the **'-'** sign.

Along with the error message, standard HTTP Status codes are used in orderto indicate the nature of the error.

# Document payload description

This section describes the various fields of the document payload.

This payload is used in the Jinius API for both Invoices and Credit Notes. It is also being used both when creating Documents (Invoices or Credit Notes) and when getting Documents from the API.

For detailed examples of Invoice or Credit Note specifc payloads, please look at the following chapters of the guide.

## Document


```
{
  "kind": "in",
  "id": "elit",
  "documentNumber": "et ullamco culpa in sed",
  "status": "eu irure sed in",
  "issueDate": "2000-01-04T21:22:14.397Z",
  "dueDate": "1962-09-22T01:10:06.590Z",
  "modificationDate": "2000-01-04T21:22:14.397Z",
  "fileId": "cillum ullamco",
  "description": "dolore minim tempor u",
  "orderReference": "laboris minim",
  "issuerVatNumber": "minim in culpa consectetur",
  "issuerTaxIdNumber": "minim dolore eiusmod irure exercitation",
  "issuerCompanyRegistrationNumber": "ex",
  "issuerBusinessUnitId": "string",
  "issuerBusinessUnitIdentifiers": [
   {
      "identifierType": "FacilityId",
      "id": "string"
   },
   {
      "identifierType": "SystemUnitId",
       "id": "string"
    }
  ],
  "recipientVatNumber": "sed anim fugiat aute",
  "recipientTaxIdNumber": "nos",
  "recipientCompanyRegistrationNumber": "sit nulla laboris minim",
  "recipientBusinessUnitId": "string",
  "recipientBusinessUnitIdentifiers": [
   {
      "identifierType": "FacilityId",
      "id": "string"
   },
   {
      "identifierType": "SystemUnitId",
       "id": "string"
    }
  ],
  "lineItems": [
    {
      "code": "aliquip veniam mollit",
      "description": "id nulla ad irure aliquip",
      "quantity": 43763352.122098505,
      "unit": "Boxes",
      "price": -49996052.39116924,
      "discountPercentage": -20152090.464874923,
```

```
      "taxPercentage": "None",
      "discount": -68814942.9956471,
      "lineTotal": 61893531.794571996,
      "taxAmount": -47641055.93863839
    }
  ],
  "net": -52623740.28470358,
  "vatAmount": 90781688.9538669,
  "totalAmount": 25217237.777853936,
  "discount": -69431816.78434663,
  "initial": -12796362.147895038,
  "documents": [
    {
      "kind": "cillum Duis tempor proiden",
      "id": "nisi id",
      "documentNumber": "elit aliqua"
    }
  ],
  "payments": [
    {
      "kind": "ea fugiat ad magna",
      "id": "enim officia",
      "transactionId": "adipisicing culpa",
      "amount": 61591241.634348184
    }
  ],
  "supportingFilesIds": [
    "string"
  ]
}
```


| Field | Description | Mandatory for Creation | Included in Response |
|---|---|---|---|
| kind | The type of the Document ('Invoice' , 'CreditNote') | No | Invoices and Credit Notes |
| id | The id of the Document. ***Must not be populated in document creation requests. This value is populated by Jinius*** | No | Invoices and Credit Notes |
| documentNumber | The number of the Document | Invoices and Credit Notes | Invoices and Credit Notes |
| status | The Status of the Document. ***Must not be populated in document creation requests. This value is populated by Jinius*** | No | Invoices and Credit Notes |
| issueDate | The date that the document was issued | Invoices and Credit Notes | Invoices and Credit Notes |
| dueDate | The date by which the document (invoice) must be paid | Invoices | Invoices<br>Empty for Credit Notes |

| | | | |
|---|---|---|---|
| modificationDate | The date when the document was last updated. ***Must not be populated in document creation requests. This value is populated by Jinius*** | No | Invoices and Credit Notes |
| fileId | The id of a file attached to the document. ***Must not be populated in document creation requests*** | No | Invoices and Credit Notes |
| description | An ***optional*** brief desceription of the document | No | Invoices and Credit Notes |
| orderReference | An ***optional*** reference to an order associated with an Invoice | No | Invoices and Credit Notes |
| issuer (VatNumber / TaxIdNumber / CompanyRegistrationNumber ) | The taxation fields for the Legal Entity issuing the document. | Invoices and Credit Notes | Invoices and Credit Notes |
| issuerBusinessUnitId | The id of the issuer's business unit.<br><br>***Optionally*** included in cases the document was issued by a specific business unit.<br><br>***These ids can be obtained via the Legal Entity Search endpoint*** | No | Invoices and Credit Notes |
| issuerBusinessUnitIdentifiers | Any custom identifiers available that are related to the business unit issuing the document.<br><br>On Invoice/Credit Note creation the following options are available:<br><br>• any of these identifiers could be used instead of the issuerBusinessUnitId.<br>• if used together with a valid issuerBusinessUnitId, the custom business unit identifiers are ignored.<br>• if issuerBusinessUnitId is not provided, but more than one identifiers are provided, only one of them has to match. | No | Yes |

| | | | |
|---|---|---|---|
| recipient (VatNumber / TaxIdNumber / CompanyRegistrationNumber ) | The taxation fields for the Legal Entity to receive the document | Invoices and Credit Notes | Invoices and Credit Notes |
| recipientBusinessUnitId | The id of the recipient's business unit.<br><br>***Optionally*** included in cases the document was issued to a specific business unit.<br><br>***These ids can be obtained via the Legal Entity Search endpoint*** | No | Invoices and Credit Notes |
| recipientBusinessUnitIdentifiers | Any identifiers available that are related to the business unit.<br><br>Same conditions apply on creation as for *issuerBusinessUnitIdentifiers* | No | |
| net | The total net value of the document | Invoices and Credit Notes | Invoices and Credit Notes |
| vatAmount | The total vat amount of the document | Invoices and Credit Notes | Invoices and Credit Notes |
| totalAmount | The total amount of the document | Invoices and Credit Notes | Invoices and Credit Notes |
| discount | The total discount amount of the document | Yes for Invoices<br><br>Invalid for Credit Notes | Invoices and Credit Notes |
| initial | The initial amount of the document (pre-tax and discounts) | Invoices and Credit Notes | Invoices and Credit Notes |
| supportingFilesIds | A list of ids for any supporting files attached to an invoice. ***Must not be populated in document creation requests*** | No | Optional for Invoices |

## Line Items

| Field | Description | Mandatory for Creation | Included in Response |
|---|---|---|---|
| code | The unique code of the item | Invoices and Credit Notes | Invoices and Credit Notes |
| description | A short description of the line item | Invoices and Credit Notes | Invoices and Credit Notes |

| | | | |
|---|---|---|---|
| quantity | The quantity of the items in this line | Invoices and Credit Notes | Invoices and Credit Notes |
| unit | The unit of the items in this line ('Kilos','Boxes','Pieces','Litres', 'Activity','Months','Days','Hours') | Invoices and Credit Notes | Invoices and Credit Notes |
| price | The price for a single item | Invoices and Credit Notes | Invoices and Credit Notes |
| discountPercentage | A percentage of discount on the line. (Can optionally be provided along with the 'discount' value) | No for Invoices<br>No for Credit Notes | Invoices and Credit Notes.<br>If the discount of the line item has been provided only as an amount (using the 'discount' value) then the discountPercentage will be calculated by Jinius |
| taxPercentage | The tax band for the line | Invoices and Credit Notes | Invoices and Credit Notes |
| discount | The total discount amount of the line. Must be populated if the line item has a discount. It can optionally be accompanied by the 'discountPercentage' field | No for Invoices<br>No for Credit Notes | Invoices and Credit Notes<br>If the discount of the line item has been provided only as a percentage (possible only via the Jinius Platform) then the 'discount' amount will be calculated by Jinius |
| lineTotal | The total value of the line | Invoices and Credit Notes | Invoices and Credit Notes |
| taxAmount | The total tax amount of the line | Invoices and Credit Notes | Invoices and Credit Notes |

## Documents

This is a list of documents related to the current document. For example Credit Notes attached to an Invoice or the Invoice on which a Credit Note is attached to.

| Field | Description | Mandatory for Creation | Included in Response |
|---|---|---|---|
| kind | The type of the Document ('Invoice' , 'CreditNote') | No | Invoices and Credit Notes |
| id | The id of the Document | No | Invoices and Credit Notes |
| documentNumber | The number of the Document | No | Invoices and Credit Notes |

## Payments

| Field | Description | Mandatory for Creation | Included in Response |
|---|---|---|---|
| kind | The type of payment ('OneBankB2B','OneBankDirectLink','Manual','JccDirectLink') | No | Invoices and Credit Notes |
| id | The id of the payment in the Jinius platform. ***This id can be used by the clients in the 'Payments' section of the API and specifically in the 'Get Payment details by Id' endpoint in order to get more details about the payment.*** | No | Invoices and Credit Notes |
| transactionId | The if of the transaction in OneBank | No | Invoices and Credit Notes |
| amount | The amount paid | No | Invoices and Credit Notes |

### Payments visibility

Please note that the payments list is populated differently for issuers and receivers of a document.

**Regarding manual payments** both roles will see payments that are either Completed or Pending Approval  (approval from the document's issuer - Accept Manual Payment endpoints)

**Regarding OneBank payments**, issuers will only see CompletedOneBank payments while receivers will see those payments that are Pending Approval  (via the OneBank portal)

Finally, **rejected payments are not shown to either party.**

| Type of Payment | Payment Status | Visible to Issuer | Visible to Receiver |
|---|---|---|---|
| OneBank | Completed | Yes | Yes |
| OneBank | Pending Approval (via the recipients OneBank portal) | No | Yes |
| OneBank | Rejected | No | No |
| Manual | Completed | Yes | Yes |
| Manual | Pending Approval (can be approved by the issuer by using the 'Accept Manual Payment' endpoints) | Yes | Yes |
| Manual | Rejected | No | No |

## Rounding of total values and line item values

Jinius is a platform used by multiple different systems, that each can possibly use a different rounding up/down policy. In order to relax a document's validation rules and allow these systems to work seamlessly with Jinius, the platform uses the following approach.

Jinius uses the provided values and re-calculates the sums and total for each line item and for the document in total. These calculated values are used for cross checking the validity of the provided amounts. Jinius will always store the client provided amounts when available.

### Line items

Jinius will consider a line item valid if the provided **'discount', 'taxAmount' and 'lineTotal'** values have a **max 0.01€ discrepancy** from the values it (Jinius) calculates.

### Document Total values

#### Basic checks

For each of the document's total values (**'net','vatAmount','totalAmount','discount','initial'**) Jinius will consider a document valid if all values have a **max (Number of Line Items x 0.01€) discrepancy** from the values it (Jinius) calculates. This takes into consideration the chance that a line item may be provided with a rounded up value.

#### Fallback check for the 'totalAmount' field

Regarding the **'totalAmount'** value in particular, Jinius has a fallback validation check (in case this field's value has a discrepancy greater than (Number of Line Items x 0.01€). Jinius will accept a document with a max discrepancy of **(2 x Number of Line Items x 0.01€) on the sum of the 'net' and 'vatAmount' fields**. This takes into consideration the chance that both these fields may have been provided with the max allowed discrepancy, descibed in the "Basic checks" section.

## Document Enumeration fields

Some of the fields used when creating an Invoice (or a Credit Note) are parts of predefined enumeration, as shown below. As such, apart from providing the string value for these fields, it is also possible to provide the numerical value of the renumeration instead.

For example, specifying that the Line Item Unit is 'Kilos' can be done either by providing the string value ("Kilos") or the numerical value (1). The API will map both inputs to the correct Line Item Unit.

Below you can find the enumeration mappings for Tax Percentages and Line Item units.

### Tax Percentage

| Tax Percentage | Number Value | String Value |
|---|---|---|
| None | 0 | "None" |
| Exempted | 1 | "Exempted" |
| Five Percent | 2 | "FivePercent" |
| Nine Percent | 3 | "NinePercent" |
| Nineteen Percent | 4 | "NineteenPercent" |
| Three Percent | 5 | "ThreePercent" |

### Line Item Unit

| Line Item Unit | Number | String Value |
| --- | --- | --- |
| Boxes | 0 | “Boxes” |
| Kilos | 1 | “Kilos” |
| Pieces | 2 | “Pieces” |
| Litres | 3 | “Litres” |
| Activity | 4 | “Activity” |
| Months | 5 | “Months” |
| Days | 6 | “Days” |
| Hours | 7 | “Hours” |

### Business Units and Custom Identifiers in Documents

Custom identifiers for Business Units are explained thoroughly in the Business Units section.

The following conditions apply when creating document using custom identifiers for business units:

- Any identifier can be used instead of the issuerBusinessUnitId.
- If used together with a valid issuerBusinessUnitId/receiverBusinessUnitId, the custom business unit identifiers are ignored.
- If issuerBusinessUnitId/receiverBusinessUnitId are not provided, but more than one identifiers are provided (for each case respectively), only one of these identifiers has to match.
- If no identifier (custom or not) is provided for the issuer or recepient business unit, then the document will be assigned to the theoretical “Unspecified” business unit. The only effect of this assignement is that specific keywords can be used when searching for documents that are not related to any business unit.

## Invoice API guidelines

### Invoice document

This is an example of a document used for creating an invoice on the Jinius Platform.

```
{
    "documentNumber": "TaxRatesCombo2",
    "issueDate": "2023-09-09T21:00:00Z",
    "dueDate": "2023-09-29T21:00:00Z",
    "description": "",
    "issuerVatNumber": "10101011C",
    "issuerTaxIdNumber": "10101011C",
    "issuerCompanyRegistrationNumber": "CY10101011",
    "issuerBusinessUnitId": "",
    "issuerBusinessUnitIdentifiers": [],
    "recipientVatNumber": "10101010C",
    "recipientTaxIdNumber": "10101010C",
    "recipientCompanyRegistrationNumber": "CY10101010",
    "recipientBusinessUnitId": "",
    "recipientBusinessUnitIdentifiers": [
        {
            "identifierType": "SystemUnitId",
            "id": "0102"
```

```
        }
    ],
    "orderReference": "",
    "lineItems": [
        {
            "code": "LI001",
            "description": "Line Item 1",
            "quantity": 10,
            "unit": "Pieces",
            "price": 10,
            "discountPercentage": 25,
            "taxPercentage": "FivePercent",
            "discount": 25,
            "lineTotal": 78.75,
            "taxAmount": 3.75
        },
        {
            "code": "LI002",
            "description": "Line Item 2",
            "quantity": 10,
            "unit": "Pieces",
            "price": 10,
            "discountPercentage": 10,
            "taxPercentage": "NineteenPercent",
            "discount": 10,
            "lineTotal": 107.1,
            "taxAmount": 17.1
        }
    ],
    "net": 165,
    "vatAmount": 20.85,
    "totalAmount": 185.85,
    "discount": 35,
    "initial": 200,
    "documents": [],
    "payments": [],
    "supportingFilesIds": []
}
```


## Invoice with Negative Line Items and Discounts

- ***Initial value*** is calculated as the sum of the initial values of positive line items
  (2500+100 = 2600)
- The total ***discount*** is calculated as the sum of negative line items and discounts on line items
  (250+100 = 350)
- For a negative line item (discount) to be valid, the tax group of the item must match the tax group of at least one of the positive line items. (ie you cannot have a discount with 9% tax unless you have a positive line item with 9% tax)
- For an invoice to be issued, the recipient's Tax, Vat and Company Registration Number must be provided (at least two out of three). The recipient must be registered to Jinius and the provided values must match with those of the recipient's.


```
{
    "documentNumber": "V30000000016",
    "issuedate": "2022-02-22T12:42:12.000Z",
    "duedate": "2022-04-21T12:42:12.000Z",
    "issuerVatNumber": "12345678K",
    "issuerTaxIdNumber": "12345678K",
```

```
    "issuerCompanyRegistrationNumber": "K12345678",
    "issuerBusinessUnitId": "string",
    "recipientVatNumber": "12345678L",
    "recipientTaxIdNumber": "12345678L",
    "recipientCompanyRegistrationNumber": "L12345678",
    "recipientBusinessUnitId": "string",
    "lineitems": [
        {
            "code": "LHP002",
            "description": "Hydraulic Lift (persons)",
            "quantity": "1",
            "unit": 1,
            "price": "100.00",
            "discountpercentage": "0.00",
            "taxpercentage": 0,
            "taxamount": "0.00",
            "discount": "0.00",
            "linetotal": "100.00"
        },
        {
            "code": "LHP001",
            "description": "Hydraulic Lift (persons)",
            "quantity": "1",
            "unit": 1,
            "price": "2500.00",
            "discountpercentage": "10.00",
            "taxpercentage": 4,
            "taxamount": "427.50",
            "discount": "250.00",
            "linetotal": "2677.50"
        },
        {
            "code": "CSD",
            "description": "Cash Discount",
            "quantity": "1",
            "unit": 1,
            "price": "-100.00",
            "discountpercentage": "0.00",
            "taxpercentage": 4,
            "taxamount": "-19.00",
            "discount": "0.00",
            "linetotal": "-119.00"
        }
    ],
    "discount": "350.00",
    "net": "2250.00",
    "initial": "2600.00",
    "vatamount": "408.50",
    "totalamount": "2658.50"
}
```


## Invoice with payments and attached Credit Note (Response payload)


```
{
    "kind": "Invoice",
    "id": "25818b12-465e-4dda-8577-6684ffc91692",
    "documentNumber": "V300000001245",
```

```
    "status": "Approved",
    "issueDate": "2019-07-31T12:42:12Z",
    "fileId": "",
    "dueDate": "2019-08-30T12:42:12Z",
    "description": "",
    "issuerVatNumber": "99999911C",
    "issuerTaxIdNumber": "99999911C",
    "issuerCompanyRegistrationNumber": "CY99999911",
    "issuerBusinessUnitId": "",
    "issuerBusinessUnitIdentifiers": [],
    "recipientVatNumber": "99999922C",
    "recipientTaxIdNumber": "99999922C",
    "recipientCompanyRegistrationNumber": "CY99999922",
    "recipientBusinessUnitId": "e695368b-b796-4799-b8c3-7a6423f0cdea",
    "recipientBusinessUnitIdentifiers": [
        {
            "identifierType": "SystemUnitId",
            "id": "0102"
        }
    ],
    "orderReference": "0210001272",
    "lineItems": [
        {
            "code": "001",
            "description": "LINE DESCRIPTION",
            "quantity": 1,
            "unit": "Kilos",
            "price": 100.1,
            "discountPercentage": 0,
            "taxPercentage": "NinePercent",
            "discount": 0,
            "lineTotal": 109.11,
            "taxAmount": 9.01
        }
    ],
    "net": 90.1,
    "vatAmount": 8.11,
    "totalAmount": 98.21,
    "discount": 0,
    "initial": 100.1,
    "documents": [
        {
            "kind": "CreditNote",
            "id": "e59ed3b8-4dc2-49b5-82be-b0174f3e1ac1",
            "documentNumber": "CN0000000000236"
        }
    ],
    "payments": [
        {
            "kind": "OneBankB2B",
            "id": "692156ca-7c56-43fb-a027-822d44a84bf1",
            "transactionId": "692156ca-7c56-43fb-a027-822d44a84ff7",
            "amount": 10,
            "status": "Successful"
        },
        {
            "kind": "Manual",
            "id": "c2ea6479-1f79-4080-9955-e72e29ab576c",
```

```
            "transactionId": "",
            "amount": 20,
            "status": "Successful"
        }
    ]
}
```

## Get Receivable/Payable Invoices

### Filtering options

#### Issue Date

Clients can filter the list of returned object by Issue Date. The ***issueDate*** field must be provided with a date value following the YYYY-MM-DD format (ie 2022-03-25). The service will return the incoices issued on that day.

#### Modification date

Clients can filter the list of returned object by Modification Date, which is the date that the document was last updated (created, changed status or had any other update performed on it (for example a file attached)). Clients can use any combination of the ***modifiedAfterDate*** and ***modifiedBeforeDate*** fields to define the desired time range.

#### Status

Clients can filter the list of returned object by status. The ***status*** must be provided with string value indicating the invoices' status *(WaitingApproval, Approved, Rejected, Voided, InReview, Paid)*

#### Other Legal Entity

Clients can filter the list of returned object by narrowing them down to those related to a specific legal entity.

For receivable invoices for example, clients can narrow the results down to those issued to a specific legal entity and for payable invoices, to those issued by a specific legal entity.

To use this filter, clients must provide the taxation fields (Vat, Tax and Company Registration Number) of the specific legal entity in the respective request fields ***vatNumber, taxNumber, companyRegistrationNumber.***

#### Business Units

Clients can filter the list of returned objects by narrowing them down to those addressed to and from specific Business Units.

In order to filter documents by a specific business unit of the client's own company, clients have to populate the **'ownBusinessUnitId'** field with the id of the desired Business Unit.

In order to filter documents by a specific business unit of other party's legal entity, clients have to populate the **'businessUnitId'** field with the id of the desired Business Unit.

Business Unit Ids can be identified via the **'Legal Entity Search'** endpoint which is thoroughly documented in this guide.

Please bear in mind that custom business unit identifiers are not supported for filtering.

Not providing a value in the above mentioned Business Unit fields (ownBusinessUnitId and businessUnitId) will bring back invoices related to all business units of the issuer and the recepient including those not related to any business unit (Unspecified)

In order to only fetch the invoices that are not related to any business unit, please use the '**notspecified**' keyword as the value of the search parameter.

#### Sorting Options

Results returned from the Get Invoices queries can be sorted on the following fields:

| Field Name |
|---|
| documentNumber |
| orderReference |
| issueDate |
| dueDate |
| status |
| totalAmount |
| initial |
| net |
| vatAmount |
| discount |
| modificationDate |

## Single Invoice Payments

OneBank and Manual Invoice Payment methods support partial payments.

This means that the clients using the manual payment endpoints (Mark as Paid, Accept Manual Payment and Set As Paid) and the OneBank payment endpoint (Pay Invoice with OneBank) have to provide the exact payment amount along with their request.

```json
{
  "paymentAmount": 5,
  "comments": "test mark as paid",
  "paymentMethod": "ManualCheque",
  "invoicePaymentDate": "2022-06-23T20:40:42.796Z",
  "referenceNumber": "1234",
  "chequeDate": "1985-02-05T22:07:40.610Z",
  "bankCode": "BOC"
}
```

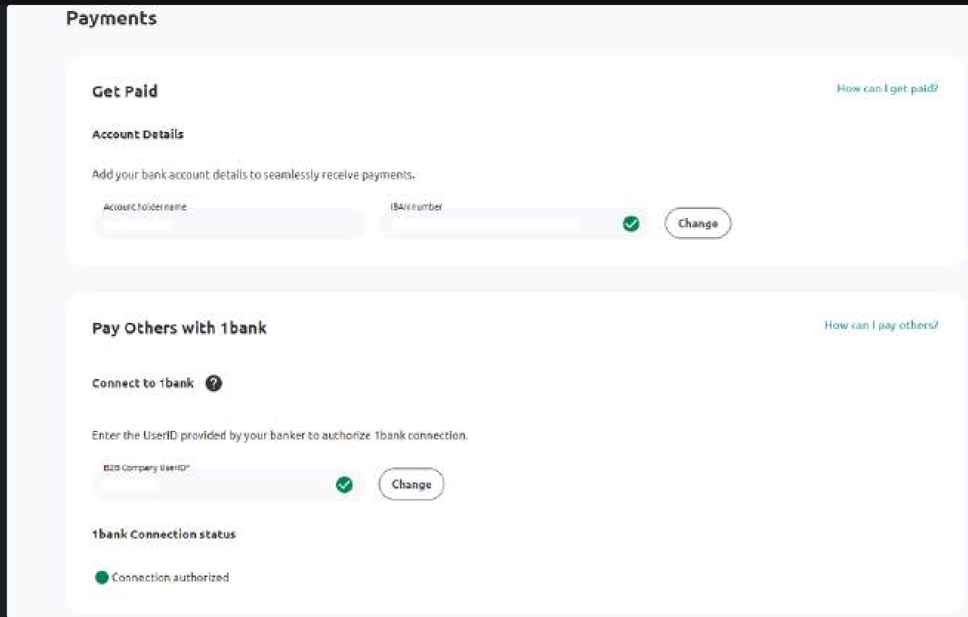
If no request body is provided, the service will respond with an error.

Please be aware that only the “**paymentAmount**” field is required for making payments.

*The rest of the fields are optional for manual payments and will be completely ignored by the API if provided when making a OneBank payment.*

The accepted values for the “**paymentMethod**” field, when making a manual payment are: “Other”, “ManualCash” and “ManualCheque”

Please be aware that in order to use OneBank as a payment method, clients need to set up their payment settings using the ‘Payment Settings’ page on the Platform.




### Manual Cheque Payments

When making a manual payment with the payment method specified as 'ManualCheque' it is possible for clients to optionally specify the Bank tha issued the cheque and the date that the cheque was settled.

To do so, clients need to populate the BankCode and ChequeDate fields.

The BankCode field will accept value from the table below:

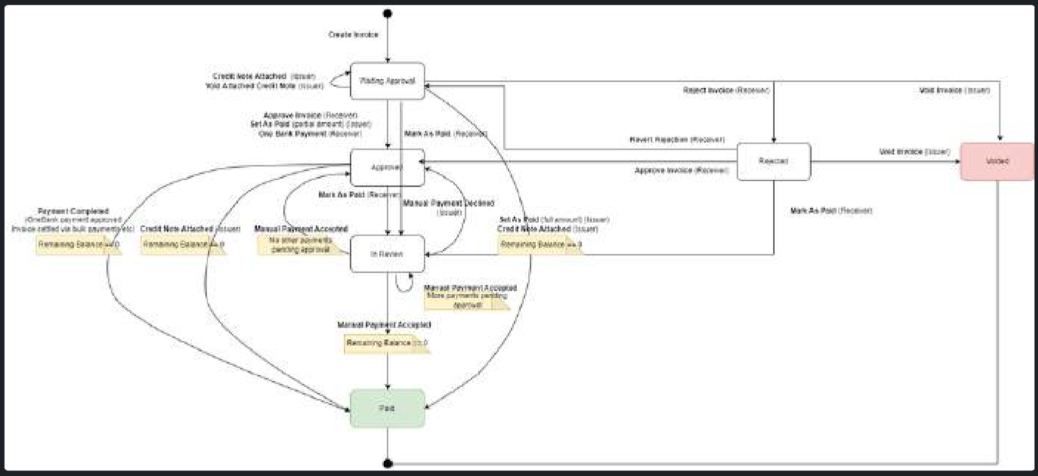
| Bank Code value | Bank’s full name |
|---|---|
| ANC | Ancoria Bank Limited |
| AST | Astrobank Public Company Limited |
| BOC | Bank of Cyprus Public Company Ltd |
| CDB | Cyprus Development Bank Public Company Limited |
| HBP | Hellenic Bank Public Company Limited |
| HFC | Housing Finance Corporation |

## Invoice Flows and Statuses

The following table lists the possible statuses of an invoice

| Status |
|---|
| WaitingApproval |
| Approved |
| Paid |
| Rejected |
| Voided |
| InReview |

Transitioning between the various statuses can be archived by taking various actions on the invoice, as depicted in the flow diagram below




Invoice Flow Diagram

# Credit Note API guidelines

## Credit Note document

This is an example of a document used for creating a Credit Note on the Jinius Platform.

```
{
   "documentNumber":"CRN00001111111",
   "issueDate":"2019-07-31T12:42:12Z",
   "issuerVatNumber":"10101011C",
   "issuerTaxIdNumber":"10101011C",
   "issuerCompanyRegistrationNumber":"CY10101011",
   "recipientVatNumber":"10101010C",
   "recipientTaxIdNumber":"10101010C",
   "recipientCompanyRegistrationNumber":"CY10101010",
   "recipientBusinessUnitIdentifiers":[
      {
         "identifierType":"SystemUnitId",
         "id":"0102"
      }
   ],
   "lineItems":[
      {
         "code":"001",
         "description":"LINE DESCRIPTION",
         "quantity":1,
         "unit":1,
         "price":10,
         "discountPercentage":0,
         "taxPercentage":3,
         "discount":0,
         "lineTotal":10.9,
         "taxAmount":0.9
      }
```

```
  ],
  "discount":0,
  "net":10,
  "initial":10,
  "vatAmount":0.9,
  "totalAmount":10.9
}
```

## Creating a Credit Note

Credit Notes can be created either by attaching them directly to an invoice, or by being issued to a another Legal Entity (standalone Crdit Note)

To use the first approach, please use the **'Create and attach a new credit note on an existing invoice'** enpoint, in which you need to specify the id of the invoice in the enpoint's url

For standalone Credit Notes, please use the **'Create a new standalone credit note'** endpoint.

## Get Issued/Received Credit Notes

### Filtering options

#### Issue Date

Clients can filter the list of returned object by Issue Date. The ***issueDate*** field must be provided with a date value following the YYYY-MM-DD format (ie 2022-03-25). The service will return the credit notes issued on that day.

#### Status

Clients can filter the list of returned object by status. The ***status*** must be provided with string value indicating the Credit Note's status *(Created, Voided, Approved, Settled)*

#### Other Legal Entity

Clients can filter the list of returned object by narrowing them down to those related to a specific legal entity.

For received credit notes for example, clients can narrow the results down to those issued to a specific legal entity and for issued credit notes, to those issued by a specific legal entity.

To use this filter, clients must provide the taxation fields (Vat, Tax and Company Registration Number) of the specific legal entity in the respective request fields ***vatNumber, taxNumber, companyRegistrationNumber.***

#### Business Units

Clients can filter the list of returned objects by narrowing them down to those addressed to and from specific Business Units.

In order to filter documents by a specific business unit of the client's own company, clients have to populate the **'ownBusinessUnitId'** field with the id of the desired Business Unit.

In order to filter documents by a specific business unit of other party's legal entity, clients have to populate the **'businessUnitId'** field with the id of the desired Business Unit.

Business Unit Ids can be identified via the **'Legal Entity Search'** endpoint which is thoroughly documented in this guide.

Please bear in mind that custom business unit identifiers are not supported for filtering.

Not providing a value in the above mentioned Business Unit fields (ownBusinessUnitId and businessUnitId) will bring back credit notes related to all business units of the issuer and the recepient including those not related to any business unit (Unspecified)

In order to only fetch the credit notesthat are not related to any business unit, please use the '**notspecified**' keyword as the value of the search parameter.

### Sorting Options

Results returned from the Get Credit Notes queries can be sorted on the following fields:

| Field Name |
|---|
| documentNumber |
| issueDate |
| total |
| status |
| modificationDate |

## Credit Note Flows and Statuses

The following table lists the possible statuses of a credit note

| Status |
|---|
| Created |
| Voided |
| Approved |
| Settled |
| Reserved |

A "**Created"** Credit Note can be approved or voided via the respective endpoints, resulting in the respective statuses.

An **"Approved"** Credit Note can still be voided via the respective endpoint, resulting in its' status becoming **"Voided"**.

A **"Reserved"** Credit Note is a temporary status of a credit note (while an invoice settlement is being processed) before becoming **"Settled"**

A **"Settled"** Credit Note has been used in an invoice settlement (could include one or more invoices)

# Legal Entity API guidelines

The Legal Entity API offer wasy for the Jinius API clients to find informations on the Legal Entities onboarded to the Jinius Platform.

The API offers two methods for getting Legal Entity information.

To begin with, It is possible for the API's clients to check whether a company is registered in the platform.

## Check if a Legal Entity exists on the Platform

To do so, clients can use the ***'Check if a Legal Entity exists on the Platform'*** endpoint and provide the following information for the company that they are interested in.

- The company's Tax id
- The company's Vat id

- The company’s Registration Number

The endpoint will respond with a ‘true’ or return a **'404 Not Found'** response, depending on whether or not the company is registered in the Jinius Platform.

The endpoint can help in cases where calling clients want to verify that one of their customers is onboarded to Jinius, before digitally sending them a new document.

## Legal Entity Search

****** Under development ******

The Legal Entity Search is an endpoint that can provide richer information regarding Legal Entities. It allows users to search for Legal Entities using various filters and returns information regading the Legal Entities and their Business Units, in a paginated response, as shown below.

```
{
    "legalEntities": [{
        "name": "string",
        "taxIdNumber": "string",
        "vatNumber": "string",
        "companyRegistrationNumber": "string",
        "businessUnits": [{
            "id": "string",
            "name": "string",
            "address": "string",
            "city": "string",
            "postcode": "string"
            "identifiers": [{
                "identifierType": "BusinessUnitIdentifierType"
                "id": "string"
            }]
        }]
    }],
    "pageResponse": {
        "currentPage": 0,
        "totalPages": 0,
        "totalItems": 0
    }
}
```

***Please be aware that the ids of the business units returned by this endpoints, are the ids that can be used when creating a new document, in order to specify the issuer or recipient business unit.***

| Parameter Name | Description |
| --- | --- |
| `name` | Legal Entity Registration/Trading name |
| `taxIdNumber` | The tax  identification of the legal entity |
| `vatNumber` | The vat number of the legal entity |
| `companyRegistrationNumber` | The registration number of the legal entity |
| `businessUnits` | List of Legal Entity’s business units |
| `id` | The identifier of the business unit |
| `name` | The name of the business unit |

| | |
|---|---|
| `address` | The address of the business unit |
| `city` | The city of the business unit |
| `postcode` | The postcode of the business unit |
| `identifiers` | A list of identifiers (for now only used for tobacco related identifiers) |
| `identifierType` | An Enum with the values:<br>'FacilityId' : (the id of the  of the business unit as provided by the EU tobacco traceability system).<br><br>'SystemUnitId' : (the id provided by the business unit for identifying its tobacco store - applies to a local level only) |
| `id` | The actual identifier id value |

### Filtering options

#### Legal Entity taxation fields

Clients can retrieve a specific legal entity by providing the ***vatNumber, taxNumber, companyRegistrationNumber*** fields.

Please be aware that as the endpoint is under development, clients must provide the above mentioned fields (at least two) in order to use the endpoint. It is not possible at the moment to leave the filters empty in order to get the full list of onboarded Legal Entities

Please be aware that at least two of the above listed fields must be provided. All of the provided fields (either two or three) must match those of the registered company. If for example Vat and Tax match but the Registration Number does not, the endpoint will return 'false'

## Business Units

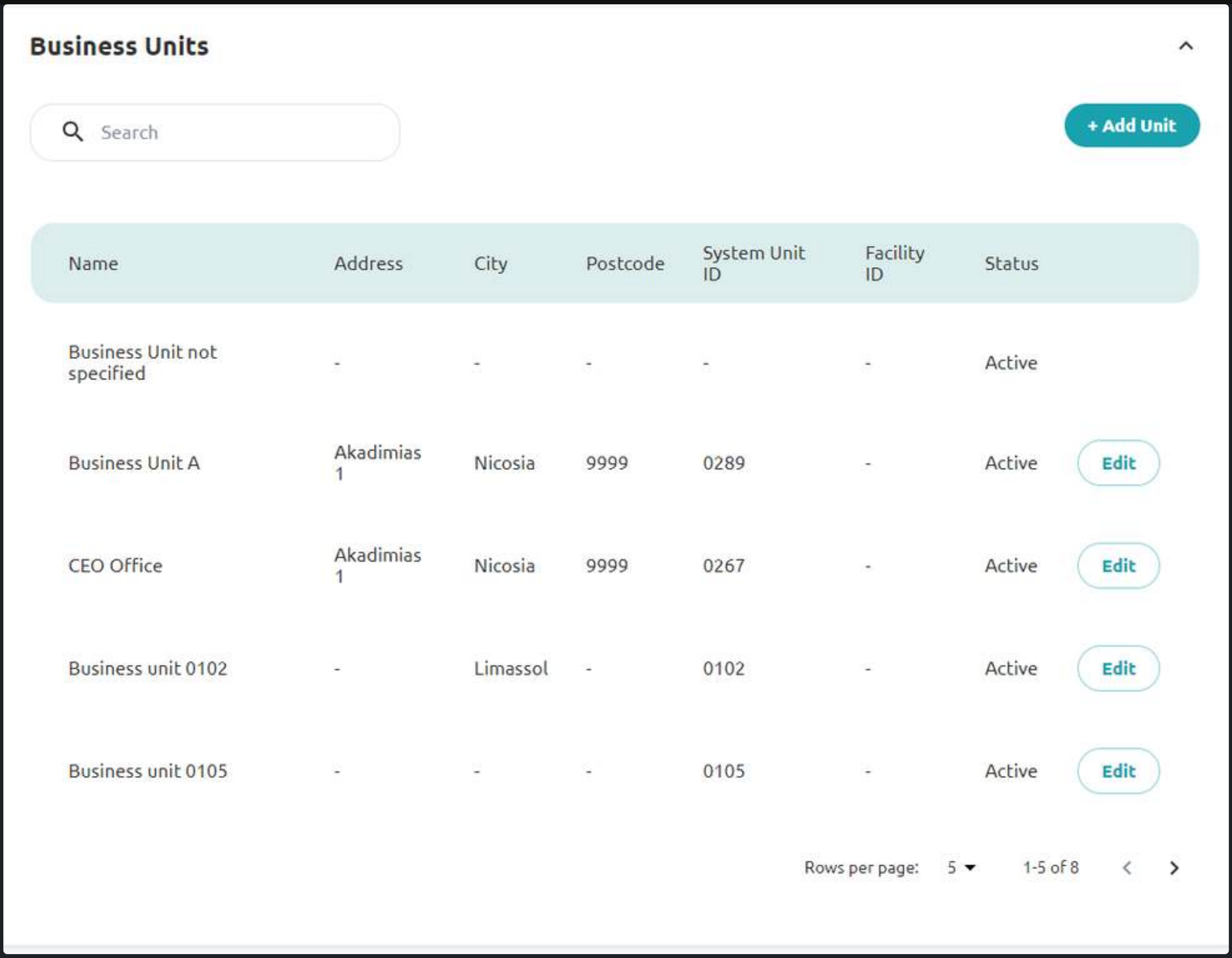
In the Jinius API, clients can refer to business units either via their unique id, or via their custom identifiers.

Unique Ids are assigned by the Jinius platform and can be retrived via the Legal Entity Search endpoint as shown above.

Custom Identifiers are extra identifiers that can be used to hold specific values that are used in various domains in order to refer to a specific business units.

At the moment, it is possible to configure two custom identifiers for a business unit via the Jinius Platform. These are the **System Unit Id** and the **Facility Id.**

This can be achieved by the Legal Entity's setting page as shown below



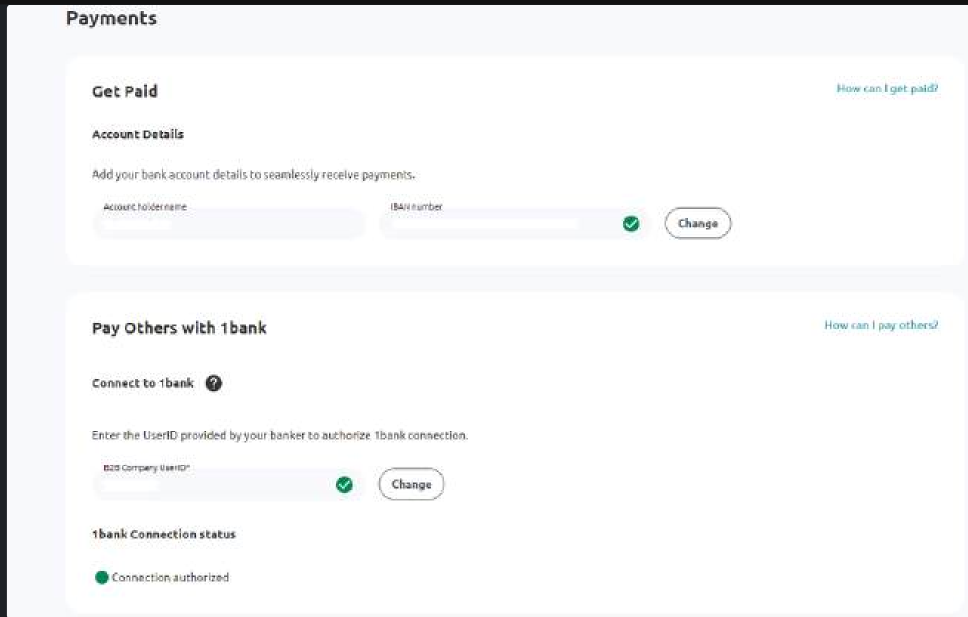

| Name | Address | City | Postcode | System Unit ID | Facility ID | Status | |
|---|---|---|---|---|---|---|---|
| Business Unit not specified | - | - | - | - | - | Active | |
| Business Unit A | Akadimias 1 | Nicosia | 9999 | 0289 | - | Active | Edit |
| CEO Office | Akadimias 1 | Nicosia | 9999 | 0267 | - | Active | Edit |
| Business unit 0102 | - | Limassol | - | 0102 | - | Active | Edit |
| Business unit 0105 | - | - | - | 0105 | - | Active | Edit |

Documents (Invoice and Credit Notes) that are not adressed to/from a business unit are treated in Jinius as if they were assigned to a theoretical “Unassigned” business unit. This does not affect the document itself and it ractically only helps in searching for those documents.

Conditions that are applied when creating and searching documents using business units are described in the respective sections.

## Remittance API guidelines

Please be aware that in order to submit any remittance advice files, clients need to set up their payment settings using the ‘Payment Settings’ page on the Platform.




## Remittance advice file

In order to craeate a Remittance, clients have to submit a Remittance advice file.

This is an example of a document used for creating a Remittance advice file on the Jinius Platform.

```
{
    "groupHeader": {
        "remittanceAdviceId": "BOC3574387",
        "creationDateTime": "2022-06-09",
        "executionDate": "2022-10-11",
        "numberOfPayments": 1,
        "totalAmount": 15.0
    },
    "payments": [
        {
            "paymentDetails": {
                "debtorDetails": {
                    "accountNumber": "351012345671"
                },
                "creditorDetails": {
                    "iban": "CY17002001280000001200527600",
                    "creditorName": "Test"
                },
                "amount": 15.0,
                "paymentReference": "1372468"
            },
            "paymentOptions":
            {
              "paymentType": "Invoices"
            },
            "supplier": {
                "taxIdentificationNumber": "11111121W",
                "vatNumber": "11111121W",
                "companyRegistrationNumber": "W20221117",
                "companyName": "Test Company",
                "email": "test@platform.cy"
            },
            "invoices": [
```

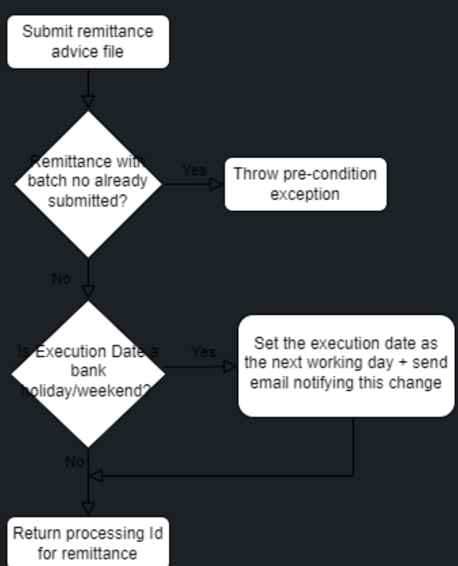
```
                {
                    "buyerReferenceNumber": "REF001",
                    "supplierInvoiceNumber": "1111",
                    "amount": 2.0,
                    "discount": 0.0,
                    "issueDate": "2022-03-09"
                },
                {
                    "buyerReferenceNumber": "REF002",
                    "supplierInvoiceNumber": "2222",
                    "amount": 13.0,
                    "discount": 0.0,
                    "issueDate": "2022-06-03"
                }
            ]
        }
    ]
}
```

### Remittance File submission

Submitting a remirrance advice file to Jinius can be achieved by using the ‘Submit Remittance’ endpoint.

Along with the file, clients can optionally provide their selected execution method (Mass/Group payment) as well as the selected company group id, in case of a group payment.

### Remittance advice file submission flow



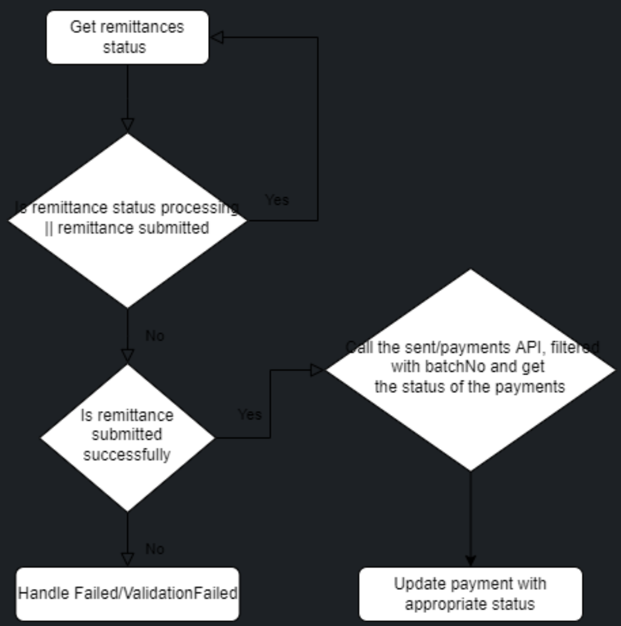

Upon submitting a remittance file, the API responds with a ‘**processingStatusId**’

This id can be used to poll for the progress of the submitted file (processing of the payments to be initiated)

In order to poll for the Remittance status, clients can use the **'Get remittance and payment statuses by processing id'** endpoint and follow the polling flow desrcibed below.

# Remittance polling

### Remittance Polling Flow Diagram




Upon calling the **'Get remittance and payment statuses by processing id'** endpoint, client get the following response, describing the progress of their submission.

```
{
  "batchNo": "<string>",
  "submittedAt": "<dateTime>",
  "totalAmount": "<double>",
  "remittanceStatus": "Processing",
  "validationErrors": [
    {
      "errorDescription": "<string>",
      "errorCode": "InvalidDebitAccount",
      "company": "<string>",
      "field": "<string>"
    },
    {
      "errorDescription": "<string>",
      "errorCode": "InvalidDebitAccount",
      "company": "<string>",
      "field": "<string>"
    }
  ],
  "failedDescription": "<string>",
  "errorCode": "<string>",
  "remittancePaymentsProgress": {
    "allRemittancePaymentsStatus" : "Pending"
  }
}
```

| Parameter Name | Description |
| --- | --- |

| | |
|---|---|
| `batchNo` | The 'remittanceAdviceId' of the submitted file |
| `submittedAt` | The date on which the file was submitted |
| `totalAmount` | The total amount of the payments to be initiated by the submitted file |
| `remittanceStatus` | The status of the Remittance file submission. More details in the Remittance Status table |
| `validationErrors` | |
| `failedDescription` | |
| `errorCode` | |
| `remittancePaymentsProgress` | The progress of the payments initiated. Fields detailed below. |

**All Remittance Payments Progress payload**

| Parameter Name | Description |
|---|---|
| `allRemittancePaymentsStatus` | The general status of all payments initiated by the remittance file submission. More details in the AllRemittancePaymentsStatus table |

### Remittance Statuses

The following table lists the possible statuses of a remittance

| Status | Description |
|---|---|
| Processing | The submission of the file is being proccessed |
| ValidationSuccessful | The file has passed input validations |
| Submitted | The file has been processed and the payments have been initiated |
| Failed | Initiating the various payments failed |
| ValidationFailed | The file has failed input validations |

### AllRemittancePayments Statuses

| Status | Description |
|---|---|
| Unspecified | No payment has been initiated yet |
| Pending | One or more payments have not been completed yet |
| Failed | All payments have failed. **Clients can resubmit the Remittance Advice file** |
| Completed | All payments have a final status (Paid or Failed (breakdown of failed payments listed in the Remittance Payment statuses table)). Clients can use the 'Get Sent/Received Remittances' endpoints to fetch their remittances and get more details on the generated payments |

Please be aware that the Remittance Status corresponds to the progress of the client's submission and whether you it passed validations and successfully initiated payments. It is possible for a remittance file to be successfully submitted and then having all of its initiated payments fail.

The AllRemittancePaymentsStatus corresponds to the progress of the initiated payments.

***Please note that it is only possible to submit a Remittance Advice File with the same RemittanceAdviceId/BatchNumber only if the AllRemittancePayments status is Failed.***

## Remittance Payments

Remittance payments are payments that are initiated as part of Remittance Advice File submission, after it successfully passed input validation.

Payments are linked to the file submission and clients can get more details on the individual payments by querying for the remittance that initiated them (using the 'Get Sent/Received Remittances' endpoint)

Related endpoints:

- [Sent] remittances/sent/payments
- [Sent] remittances/payments/:paymentReferenceId
- [Received] remittances/received/payments

| Fields | Format | Description | Example value |
|---|---|---|---|
| `paymentId` | UUID - 36 characters | | db9899ce-0da7-487f-a459-7d28a862f928 |
| `legalEntityTaxationFields` | Object | Supplier details submitted in Remittance file (Sent endpoints)<br><br>Business details who sent the Remittance (Received endpoints) | {<br>"vatNumber": "",<br>"companyRegistrationNumber": "",<br>"taxIdentificationNumber": ""<br>}, |
| `legalEntityTaxationFields`<br><br>`-> taxIdentificationNumber` | Alphanumeric | TIN information for related business | 11223344A |
| `legalEntityTaxationFields`<br><br>`-> vatNumber` | Alphanumeric | VAT value for related business | 11223344A |
| `legalEntityTaxationFields`<br><br>`->`<br>`companyRegistrationNumber` | Alphanumeric | Company registration number | HE11111 |
| `paymentReferenceId` | Alphanumeric Selected special chars allowed (-) | | PAY1000000000001 |
| `dueDate` | Dateime with timezone - | | 2024-06-13T00:00:00Z |

| | | | |
|---|---|---|---|
| | UTC 00:00 | | |
| `amount` | Decimal | | 307.66 |
| `status` | String | Payment status<br>Refer to [Remittance Payment Statuses] | Paid |
| `toAccountIban` | String | Creditor IBAN account | |
| `company` | string | Business name submitted in Supplier details during remittance submission (Sent endpoints)<br><br>Business name who submitted the remittance (Received endpoints) | Company Ltd |
| `fromAccountId` | string | Debtor account<br>(only for Send endpoints) | |
| `toBeneficiaryName` | string | Business name submitted in Supplier details during remittance submission (Sent endpoints) | Company Ltd |
| `paidOn` | Date | Payment execution date | 2024-06-17 |
| `batchNo` | Alphanumeric | Unique number to identify batch file | REMLE00000370 |
| `invoiceCount` | Integer | Count of how many invoices have been included in this payment.<br>In case of General payment, this value is 0. | 5 |
| `submittedOn` | Datetime - with timezone UTC | Datetime of submission request | 2024-05-17T14:23:23.95246Z |
| `bankTransactionNumber` | Alphanumeric | Transaction number returned by the bank | 342244 |
| `email` | Email | Creditor email | credito@email.com |
| `actualExecutionDate` | Date | Payment execution date | 2024-06-17 |
| `modifiedOn` | Datetime - UTC | Latest modification datetime | 2024-06-17 14:23:28 |
| `paymentDescription` | String | Payment description for general payments | PAY1000000000001 - HE111 - Jinius Payment |
| `paymentType` | string | Payment type to define if payment includes invoicing information or not | General |

### Example RemittanceIntegrationPayment


```
{
  "PaymentId": "921aabeb-e434-4632-861b-95840e12d2d2",
  "LegalEntityTaxationFields": {
    "VatNumber": "VatNumbere0566b52",
    "CompanyRegistrationNumber": "CompanyRegistrationNumber4bee3210",
    "TaxIdentificationNumber": "TaxIdentificationNumber72db0c56"
```

```
  },
  "PaymentReferenceId": "ReferenceId94f9e89c",
  "Amount": 68.000000086,
  "Status": 0,
  "ToAccountIban": "ToAccountIban9e5f44de",
  "Company": "Company685ea7c9",
  "PaidOn": "1970-01-01",
  "BatchNo": "BatchNo3a724c3b-b026-4d04-80ac-06d1f5f09ef7",
  "InvoiceCount": 146,
  "BankTransactionNumber": "BankTransactionNumber5cfbeeb6",
  "ModifiedOn": "1970-01-01 00:03:39",
  "PaymentDescription": "ReferenceId94f9e89c - CreditorCompNo - DebtorCompNo - Jinius Payment - FilePaymentDescription605530d1",
  "PaymentType": "Invoices"
}
```

### Remittance Payment Statuses

The following table lists the possible statuses of a remittance payment

| Status | Description |
|---|---|
| RejectedByUser | The payment has been rejected by the approver |
| PendingApproval | The payment is pending approval |
| Paid | The payment has been completed |
| Failed | The payment has failed |
| DocumentValidationFailed | Payment didn’t process because of invalid document information |
| Expired | Payment has expired without being approved |

### Remittance Payment Type

The following table lists the possible payment types of a remittance payment

| Payment Type | Description |
|---|---|
| Unspecified | Not set - Unexpected type |
| General | General payment is a remittance payment without invoicing information (e.g pre-payment) |
| Invoices | Invoices type defined that the remittance payment has the related invoicing information and remittance advices is provided. |

# Get Sent Remittances

## Filtering options

### Submission Date From/To

Clients can filter the list of returned object by Submission Date. The ***submissionDateFrom*** and ***submissionDateTo*** (either or both) fields must be provided with a date value following the YYYY-MM-DD format (ie 2022-03-25). The service will return the remittances sent between those dates.

### Payment Status

Clients can filter the list of returned object by Payment Status. The ***paymentStatus*** must be provided with string value indicating the remittance payment status *( PendingApproval, Paid, Failed, RejectedByUser, DocumentValidationFailed, Expired)*

### Batch Number

Clients can filter the list of returned object by remittance batchNo. The ***batchNo*** must be provided with string value indicating the remittance file which the payments belong.

### ModifiedDate After/Before

Clients can filter the list of returned object by ModifiedDate. ModifiedDate is the date a remittance payment record changed , usually when the status of the payment change. The ***modifiedAfterDate*** and ***modifiedBeforeDate*** (either or both) fields must be provided with a date value following the YYYY-MM-DD format (ie 2022-03-25). The service will return the remittances sent between those dates.

# Get Received Remittances

## Filtering options

### Submission Date From/To

Clients can filter the list of returned object by Submission Date. The ***submissionDateFrom*** and ***submissionDateTo*** (either or both) fields must be provided with a date value following the YYYY-MM-DD format (ie 2022-03-25). The service will return the remittances received between those dates.

### ModifiedDate After/Before

Clients can filter the list of returned object by ModifiedDate. ModifiedDate is the date a remittance payment record changed , usually when the status of the payment change. The ***modifiedAfterDate*** and ***modifiedBeforeDate***(either or both) fields must be provided with a date value following the YYYY-MM-DD format (ie 2022-03-25). The service will return the remittances received between those dates.

# Get Sent/Received Remittances sorting options

The results of the Get Sent/Received Remittances queries can be sorted on the following fields

| Field Name |
| --- |
| batchNo |
| paymentReferenceId |
| company |
| submittedOn |
| dueDate |

| |
|---|
| invoiceCount |
| amount |
| status |
| paidOn |
| modifiedOn |

## Get Remittance Invoices

The Get Remittance Invoices is an endpoint that can provide richer information regarding invoices created by a remittance submission. It allows users to search for Remittance Invoices using various filters and returns information regarding the Remittance Invoices, in a paginated response, as shown below.

### Example Response

```
{
  "pageResponse": {
    "currentPage": 1,
    "totalPages": 1,
    "totalItems": 1
  },
  "invoices": [
    {
      "issueDate": "2021-11-07",
      "buyerInvoiceNumber": "4e5d5508-5645-43dd-867c-71b11039f271",
      "supplierInvoiceNumber": "INV0002",
      "amount": 7,
      "discount": 0,
      "paymentStatus": "Paid",
      "executionDate": "2022-11-14",
      "referenceNumber": "20221114090101",
      "documentId": "68f7c7e4-b391-40e8-acd0-3bbdd52336c1",
      "documentType": "Invoice",
      "paymentId": "244745b7-baa1-4662-bb81-a319a1d7beb6"
    }
  ]
}
```

### Filtering options

#### Payment Id

This is a unique payment id for each remittance payment (**paymentId**). Clients can take the unique paymentId from the Get Remittance Sent or Received Payments api responses. **This field is mandatory.**

#### Invoice Issue Date From/To

Clients can filter the list of returned object by Submission Date. The ***invoiceIssueDateFrom*** and ***invoiceIssueDateTo*** (either or both) fields must be provided with a date value following the YYYY-MM-DD format (ie 2022-03-25).

## Get Remittance Received Invoices

The Get Remittance Received Invoices is an endpoint that can provide information regarding invoices created by a remittance submission. This endpoint returns remittance received payment invoices, only with status paid. It allows users to search for Remittance Invoices using execution date from/to filters and returns information regarding the Remittance Invoices, in a paginated response, as shown below.

### Example Response

```
 {
  "pageResponse": {
    "currentPage": 1,
    "totalPages": 1,
    "totalItems": 1
  },
  "invoices": [
    {
      "issueDate": "2021-11-07",
      "buyerInvoiceNumber": "4e5d5508-5645-43dd-867c-71b11039f271",
      "supplierInvoiceNumber": "INV0002",
      "amount": 7,
      "discount": 0,
      "paymentStatus": "Paid",
      "executionDate": "2022-11-14",
      "referenceNumber": "20221114090101",
      "documentId": "68f7c7e4-b391-40e8-acd0-3bbdd52336c1",
      "documentType": "Invoice",
      "paymentId": "244745b7-baa1-4662-bb81-a319a1d7beb6"
    }
  ]
}
```

### Filtering options

#### Payment Id

This is a unique payment id for each remittance payment (**paymentId**). Clients can take the unique paymentId from the Get Remittance Sent or Received Payments api responses.

#### Invoice Issue Date From/To

Clients can filter the list of returned object by Submission Date. The ***invoiceIssueDateFrom*** and ***invoiceIssueDateTo*** (either or both) fields must be provided with a date value following the YYYY-MM-DD format (ie 2022-03-25).

#### Payment Execution Date From/To

Clients can filter the list of returned object by Invoices Execution Date. The ***paymentExecutionDateFrom*** and ***paymentExecutionDateTo*** (either or both) fields must be provided with a date value following the YYYY-MM-DD format (ie 2022-03-25).

#### Supplier Invoice Number

This is the invoice number of the supplier (***supplierInvoiceNumber***). Clients can retrieve the invoice of a specific invoice number.

## Get Processing Ids

In the case there was an error or the processing id that was returned was lost after the submission of the remittance advice id this endpoint is provided to recover the processing ids associated with the remittances submitted.

#### Filtering options

##### Batch Number

Clients can filter the list of returned object by Batch Number. The ***batchNo*** must be provided with string value indicating the remittance batch number.

##### Submission Date From/To

Clients can filter the list of returned object by Submission Date. The ***submissionDateFrom*** and ***submissionDateTo*** (either or both) fields must be provided with a date value following the YYYY-MM-DD format (ie 2022-03-25). The service will return the processing ids along with the batch numbers associated with the processing ids between those dates.

##### Remittance Status

Clients can filter the list of returned object by Remittance Status. The ***remittanceStatus*** must be provided with string value indicating the remittance status (*Unset*, *Validated*, *Submitted*, *Failed*, *ValidationFailed*)

## Remittance File Validation

The Remittance File Validation endpoint accepts a remittance file as a parameter and only runs the validations without submitting the file for payment. If it has validation errors then it returns a list of errors with a status of ValidationFailed, otherwise if the file is valid then the list of errors will be null and the status will be ValidationSuccessful.

Optionally, along with the file, clients can provided their slected execution method for Group or Mass payments.

# Payments API Guidelines

The payments section of the Jinius API allows you to settle multiple invoices and credit notes (bulk payments) as well as retrieving information about individual payments (coming soon)

Using the **Set multiple invoices as paid (Issuer)** endpoint you can (as an issuer) record a payment and set multiple invoices as paid. Please note that all documents used in the payment (invoices and credit notes) must have been issued to the same buyer legal entity)

## Bulk Payments payload


```
{
  "invoiceAmounts": [
    {
      "id": "63710b26-7cbd-4169-9b40-0f2791b0efd7",
      "amount": 10
    },
    {
      "id": "52978f88-4cbd-4b46-a0b7-6660fd169208",
      "amount": 10
    }
  ],
  "creditNoteAmounts": [
    {
      "id": "52978f88-4cbd-4b46-a0b7-6660fd169632",
      "amount": 10
    }
  ],
  "totalPaymentAmount": 10,
  "invoicePaymentDate": "2022-08-08T23:03:24.707Z",
  "referenceNumber": "This is a reference number",
```

```
  "comments": "This is a comment",
  "paymentMethod": "ManualCheque",
  "chequeDate": "1985-02-05T22:07:40.610Z",
  "bankCode": "BOC"
}
```

*Please note that using credit notes in order to settle a payment is not mandatory, so the list of credit notes can be left empty. If however you choose to do so, you need to provide the credit note id and the amount to be used towards settling the payment*

When using “ManualCheque” as the payment method it is possible to specify the Bank that issued the checque as well as the date that the cheque was settled. These values are optional. To see the full list of Bank Codes that can be used, please refer to the Jinius API - Technical Guide | Manual Cheque Payments section of the guide.

| Field | Description | Mandatory |
|---|---|---|
| invoiceAmounts.Id | The id of the invoice to be paid | Yes |
| invoiceAmounts.Amount | The amount to be paid towards the invoice (must be the outstanding amount of the invoice) | Yes |
| creditNoteAmounts.Id | The id of the credit note (standalone) to be used in this payment | Yes when using credit notes for the payment |
| creditNoteAmounts.Amount | The amount of the credit note to be used in this payment | Yes when using credit notes for the payment |
| totalPaymentAmount | The total amount to be paid towards the invoices in this payment.<br>The value must equal the sum of invoice amounts minus the sum of credit note amounts.<br>(In the example above: (10+10) - (10) = 10) | Yes |
| invoicePaymentDate | The date the the payment being recorded took place (optional) | No |
| referenceNumber | An optional reference number | No |
| comments | Any optional comments for this payment | No |
| paymentMethod | The method used for this payment<br>Available options (ManualCash, ManualCheque, Other) | No |
| chequeDate | The date that the cheque used for the payment was settled. This can be optionally provided only when the payment method is set to “ManualCheque” | No |
| bankCode | The code of the bank that issued the check. This can be optionally provided | No |

| | | | |
|---|---|---|---|
| | only when the payment method is set to “ManualCheque” (please refer to the Jinius API - Technical Guide \| Manual Cheque Payments section of the guide) | | |

### Payment Reports

Payment Reports are detailed reports on payments, listing various details about the payment, along with a list of documents (invoices and credit notes) that were settled by the payment.

Payment Reports can be obtained by providing a payment id ***(included in an invoice’s payments list)*** to the ‘Get Payment Details by Id’ endpoint

| Field | Description |
|---|---|
| invoiceIds | The ids of the invoices  paid |
| CreditNoteIds | The ids of the credit note (standalone) used in this payment |
| paymentAmount | The total amount to be paid towards the invoices in this payment. |
| kind | The payment method used for this payment |
| status | The status of the payment (OneBank, OneBankDirectLink, Manual, ManualCash, ManualCheque, JccDirectLink) |
| oneBankEndToEndId | The OneBank id (in case the payment was executed via OneBank) |
| invoicePaymentDate | The date the the payment being recorded took place (optional) |
| referenceNumber | An optional reference number |
| comments | Any optional comments for this payment |
| bankCode | The code of the bank tha issued the cheque. This can be optionally populated only when the payment method is set to “ManualCheque” |
| chequeDate | The date that the cheque was settled. This can be optionally populated only when the payment method is set to “ManualCheque” |

## Orders API guidelines

The marketplace orders section of the Jinius API allows you to retrieve order information, create carrier waybills, and notify the carrier to send the orders.

### Search Orders

The Order Search is an endpoint that can provide  information regarding the seller Orders. It allows users to search for Orders using various filters and returns information regarding the Orders and their shipment details, in a paginated response, as shown below.

```
{
```

```
    "orders": [
        {
            "orderId": "00069224-A",
            "shopId": "2289",
            "orderLines": [
                {
                    "orderLineId": "00069224-A-1",
                    "shipmentDetails": {
                        "id": "414cc6a3-d641-4708-b9f0-8847776c1ab6",
                        "tracking": {
                            "carrierCode": "ACS",
                            "carrierName": "ACS",
                            "trackingNumber": "7401084154",
                            "trackingUrl": "https://www.acscourier.net/en/track-and-trace?trackingNumber=7401084154"
                        }
                    },
                    "orderState": "Received"
                }
            ],
            "orderState": "Received",
            "shippingDetails": {
                "shippingDeadLine": "2024-01-04T14:04:00.4775052Z",
                "shippingAddress": "59 Gladstonos, 804659 Gladstonos, 3333",
                "shippingCity": "Nicosia",
                "shippingAddressNumber": "59 Gladstonos, 804659 Gladstonos, 3333",
                "shippingPostalCode": "2450",
                "shippingTrackingNumber": "7401084154",
                "shippingStatus": "CarrierNotified",
                "shippingType": "Standard"
            },
            "createdDate": "2024-01-02T14:04:00.499Z"
        }
    ],
    "pageResponse": {
        "currentPage": 1,
        "totalPages": 28,
        "totalItems": 28
    }
}
```


## Filtering options

### Created Date

Clients can filter the list of returned object by Order Created Date. The ***createdDateFrom and createdDateTo*** field must be provided with a date value following the YYYY-MM-DD format (ie 2022-03-25). The service will return the orders created within the From to To range.

### Order State

Clients can filter the list of returned object by order status. The ***orderState*** must be provided with string value indicating the Order’s state *(Staging, WaitingAcceptance, WaitingDebit, WaitingDebitPayment, Shipping, Shipped, ToCollect, Received, Closed, Refused, Canceled, Refunded).*

#### Shipping Status

Clients can filter the list of returned object by shipping status. The **shippingStatus** must be provided with string value indicating the status of the order shipment *(CarrierNotified, WaybillCreated, PendingCreationOfWaybill).*

#### Shipping Type

Clients can filter the list of returned object by shipping type. The **shippingType** must be provided with string value indicating the type of the order shipment *(Standard, Express).*

#### Business Units

Clients can filter the list of returned objects by narrowing them down to those addressed from specific Business Units.

Business Unit Ids can be identified via the **'Legal Entity Search'** endpoint which is thoroughly documented in this guide.

## Get Shops

This endpoint (v3/marketplace/shops) returns all shops that are attached to the logged-in legal entity. No payload or/and query parameters are needed as the legal entity is extracted from the headers.

Sample response:


```
{
    "shops": [
        {
            "shopName": "emily's shop1",
            "shopId": "2105",
            "address": {
                "street": "test address 1",
                "city": "nicosia",
                "zipCode": "7040",
                "country": "CYP"
            }
        }
    ],
    "pageResponse": {
        "currentPage": 1,
        "totalPages": 1,
        "totalItems": 1
    }
}
```


## Create Waybill for ACS Courier

### Endpoint

`POST /v3/marketplace/orders/shop/:shopId/order/:orderNumber/waybill`

### Description

This endpoint creates a waybill for ACS courier for a specified shop and order number. The address for the waybill is derived from the Mirakl address associated with the provided shop.

### Path Parameters

| Parameter | Type | Description |
|---|---|---|

| | | |
|---|---|---|
| `shopId` | string | The unique identifier of the shop. |
| `orderNumber` | string | The unique number of the order. |

### Request

This endpoint does not require a request body, as the necessary information is provided through the path parameters.

### Response

#### Successful Response (200 OK)

A JSON object containing the details of the created waybill.

`{ "VoucherNumber": "string", "FileBase64": "string" }`

##### Response Fields

- `VoucherNumber` (string): The voucher number of the created waybill.
- `FileBase64` (string): The waybill file in PDF format, encoded as a Base64 string.

### Example Request

```
POST /v3/marketplace/orders/shop/2289/order/00071100-A/waybill
```

### Example Response



```
{
    "VoucherNumber": "ABC123456",
    "FileBase64": "JVBERi0xLjQKJaqrr..."
}
```



### Notes

- Ensure that the `shopId` and `orderNumber` are valid and associated with each other.
- Ensure that the `shopId` is associated with the legal entity that makes the request.
- The waybill is created using the Mirakl address linked to the specified shop.

## Create Waybill for ACS Courier for A Specific Business Unit Address

### Endpoint

`POST /v3/marketplace/orders/shop/:shopId/businessunit/:businessUnitId/order/:orderNumber/waybill`

### Description

This endpoint creates a waybill for ACS courier for a specified shop a specified business unit and order number. The address for the waybill is derived from the business unit address associated.

### Path Parameters

| Parameter | Type | Description |
|---|---|---|

| | | |
|---|---|---|
| `shopId` | string | The unique identifier of the shop. |
| `businessUnitId` | string | The unique number of the business unit. |
| `orderNumber` | string | The unique number of the order. |

### Request

This endpoint does not require a request body, as the necessary information is provided through the path parameters.

### Response

#### Successful Response (200 OK)

A JSON object containing the details of the created waybill.

`{ "VoucherNumber": "string", "FileBase64": "string" }`

##### Response Fields

- `VoucherNumber` (string): The voucher number of the created waybill.
- `FileBase64` (string): The waybill file in PDF format, encoded as a Base64 string.

### Example Request

```
POST /v3/marketplace/orders/shop/2289/order/00071100-A/waybill
```

### Example Response

```
{
    "VoucherNumber": "ABC123456",
    "FileBase64": "JVBERi0xLjQKJaqrr..."
}
```

### Notes

- Ensure that the `shopId` and `orderNumber` are valid and associated with each other.
- Ensure that the `shopId` is associated with the legal entity that makes the request.
- Ensure that the `shopId` and `businessUnitId` are valid and associated with each other.
- The waybill is created using the Business unit address provided.
-

## Get Waybill for a Specific Order

### Endpoint

`GET /v3/marketplace/orders/shop/:shopId/order/:orderNumber/waybill`

### Description

This endpoint retrieves the waybill for a specific order number within a specified shop. The response contains the waybill details in the same format as the waybill creation endpoint.

### Path Parameters

| Parameter | Type | Description |
|---|---|---|
| `shopId` | string | The unique identifier of the shop. |
| `orderNumber` | string | The unique number of the order. |

### Request

This endpoint does not require a request body, as the necessary information is provided through the path parameters.

### Response

#### Successful Response (200 OK)

A JSON object containing the details of the requested waybill.

```
json { "VoucherNumber": "string", "FileBase64": "string" }
```

##### Response Fields

- `VoucherNumber` (string): The voucher number of the waybill.
- `FileBase64` (string): The waybill file in PDF format, encoded as a Base64 string.

### Example Request

```
GET /v3/marketplace/orders/shop/2289/order/00071100-A/waybill
```

### Example Response

```
{ "VoucherNumber": "ABC123456", "FileBase64": "JVBERi0xLjQKJaqrr..." }
```

### Notes

- Ensure that the `shopId` and `orderNumber` are valid and associated with each other.
- The waybill details are returned in the same format as when a waybill is created.

## Notify Carrier For All Orders with Created Waybills

### Endpoint

`POST /v3/marketplace/shops/shop/:shopId/notifyallcarrier`

### Description

This endpoint notifies the carrier to pick up all orders with created waybills for a specified shop. It requires specifying a date and a time range for the pick-up.

### Path Parameters

| Parameter | Type | Description |
|---|---|---|

| | | |
|---|---|---|
| `shopId` | string | The unique identifier of the shop. |

### Request Body

A JSON object specifying the date and time range for the carrier pick-up.

```
{ "date": "2024-06-22", "timeFrom": "09:00", "timeTo": "17:00" }
```

#### Request Fields

- `date` (string): The date for the carrier pick-up. Must be a future date (Cyprus date and time), not a weekend or bank holiday.
- `timeFrom` (string): The start time for the pick-up window. Must be in the range 09:00 to 17:00.
- `timeTo` (string): The end time for the pick-up window. Must be in the range 09:00 to 17:00.

### Response

#### Successful Response (200 OK)

A JSON object containing the pick-up ID returned by the carrier.

```
{ "PickUpId": "string" }
```

##### Response Fields

- `PickUpId` (string): The pick-up ID returned by the carrier.

### Example Request

```
POST /v3/marketplace/shops/shop/2289/notifyallcarrier Content-Type: application/json { "date": "2024-06-22", "timeFrom": "09:00", "timeTo": "17:00" }
```

### Example Response

```
{ "PickUpId": "PU123456789" }
```

### Notes

- The `date` must be a future date according to Cyprus date and time, and it must not be a weekend or a bank holiday.
- The `timeFrom` and `timeTo` must be within the range 09:00 to 17:00.
- Ensure that the `shopId` is valid and has orders with created waybills ready for pick-up.

## Notify Carrier to Pick Up All Orders with Created Waybills for a Business Unit

### Endpoint

`POST /v3/marketplace/shops/shop/:shopId/businessunit/:businessUnitId/notifyallcarrier`

### Description

This endpoint notifies the carrier to pick up all orders with created waybills for a specified business unit within a shop. It requires specifying a date and a time range for the pick-up.

### Path Parameters

| Parameter | Type | Description |
| --- | --- | --- |
| `shopId` | string | The unique identifier of the shop. |
| `businessUnitId` | string | The unique identifier (GUID) of the business unit. |

### Request Body

A JSON object specifying the date and time range for the carrier pick-up.

```
{ "date": "2024-06-22", "timeFrom": "09:00", "timeTo": "17:00" }
```

#### Request Fields

- `date` (string): The date for the carrier pick-up. Must be a future date (Cyprus date and time), not a weekend or bank holiday.
- `timeFrom` (string): The start time for the pick-up window. Must be in the range 09:00 to 17:00.
- `timeTo` (string): The end time for the pick-up window. Must be in the range 09:00 to 17:00.

### Response

#### Successful Response (200 OK)

A JSON object containing the pick-up ID returned by the carrier.

```
{ "PickUpId": "string" }
```

##### Response Fields

- `PickUpId` (string): The pick-up ID returned by the carrier.

### Example Request

```
POST /v3/marketplace/shops/shop/2289/businessunit/123e4567-e89b-12d3-a456-426614174000/notifyallcarrier
Content-Type: application/json { "date": "2024-06-22", "timeFrom": "09:00", "timeTo": "17:00" }
```

### Example Response

```
{ "PickUpId": "PU123456789" }
```

### Notes

- The `date` must be a future date according to Cyprus date and time, and it must not be a weekend or a bank holiday.
- The `timeFrom` and `timeTo` must be within the range 09:00 to 17:00.
- Ensure that the `shopId` and `businessUnitId` are valid and that the business unit has orders with created waybills ready for pick-up.
- Ensure that the `businessUnitId` belongs to the shop.

## Cancel Notification for a Specific Order

### Endpoint

`PUT /v3/marketplace/shops/shop/:shopId/order/:orderNumber/cancelnotification`

### Description

This endpoint cancels the notification for a specific order within a specified shop. No request body is needed, and there is no response body.

### Path Parameters

| Parameter | Type | Description |
|---|---|---|
| `shopId` | string | The unique identifier of the shop. |
| `orderNumber` | string | The unique number of the order. |

### Request

This endpoint does not require a request body. The necessary information is provided through the path parameters.

### Response

#### Successful Response (200 OK)

This endpoint does not return a response body. The status code `200 OK` indicates that the notification for the specified order has been successfully cancelled.

### Example Request

```
PUT /v3/marketplace/shops/shop/2289/order/00071100-A/cancelnotification
```

### Example Response

```
HTTP/1.1 200 OK
```

### Notes

- Ensure that the `shopId` and `orderNumber` are valid and associated with each other.
- This endpoint is used to cancel a previously sent notification for an order.

---

# Changelog

## v1.0.0 (08-04-2022)

Initial version

## v1.0.1 (20-04-2022)

Adding description of default API's behaviour when no pagination fields are provided

## v1.0.2 (24-05-2022)

Adding description for modification date fields and filters

### v1.0.3 (01-06-2022)

Invoice with payments and attached Credit Note

### v1.0.4 (20-06-2022)

Wording fix in the pagination section

### v1.0.5 (21-06-2022)

Adding remittance section with remittance advice file example and description on queries with filters

Fix typos

### v1.0.6 (20-06-2022)

Added supporting files ids section in Document payload

### v1.0.7 (28-06-2022)

Added documentation for payment method specification for manual payments

### v1.0.8 (30-06-2022)

Added description of the updated Credit Note flow and states

### v1.0.9 (11-07-2022)

Added description for Credit Note creation (standalone and attached to an Invoice)

### v1.0.10 (12-08-2022)

Introduction to the Payments API

### v1.0.11 (12-08-2022)

Payment Reports and linkage to invoice payments (via payment id)

### v1.0.12 (06-09-2022)

New get remittance invoices for a specific payment end point (via payment id)

Remove payments from get remittance submission status end point response.

Add new query parameter (batchNo) for get sent remittance payments end point.

### v1.0.13 (30-09-2022)

Update Invoice and Credit Note Enumeration values sections with the string representation of the enumeration values

### v1.0.14 (20-10-2022)

Updated Legal Entity API section with details on the new (Under Construction) ‘Legal Entity Search’ Endpoint.

Updated the ‘Check if Legal Entity Exists' endpoint with a 404 Not Found response in case no Legal Entity is found. 'False’ was previously been returned for this case.

Added Business Unit ids and their respective description inside the Document payload and the Invoice and Credit Note examples.

Added clear explanation on which fields are mandatory or optional during a document’s creation.

Added clear description on cross-enpoint dependencies

1. Payment id available in the invoice payments section can be used for getting the details of a specific payment
2. Business Unit id available in the Legal Entity Search response can be used when creating a new document to specify any specific Business Units involved

## v1.0.14 (24-10-2022)

Updated payments section with the various special cases of the payments visibility

Remittance: New payment status ‘DocumentValidationFailed’

## v1.0.15 (01-11-2022)

Added description of the sorting options for Invoices, Credit Notes and Remittances

## v1.0.16 (08-11-2022)

Added description of the new rounding up/down policy used by Jinius for values provided in Financial Documents

## v1.0.17 (13-12-2022)

Remittance: New endpoint for file validation only

## v1.0.18 (11-01-2023)

Payments: New payment type : JccDirectLink

## v1.0.19 (19-01-2023)

Manual Payments: Bank Code and Cheque Date

## v1.0.20 (23-01-2023)

Remittance: New payment status ‘Expired’

## v1.0.21 (14-02-2023)

Adding ‘modificationDate’ as Sorting Option for Invoices, Credit Notes and Remittance

## v1.0.22 (06-03-2023)

New Get Remittance Received Invoices endpoint

## v1.0.23 (20-03-2023)

Added clarification on the ‘ChequeDate’ field of invoice manual payments referring to a cheque’s settlement date.

Added description on Business Unit filtering funcationality for Invoice and Credit Notes search endpoints.

## v1.0.24 (20-04-2023)

Added supplierInvoiceNumber filter in Get Remittance Received Invoices endpoint.

## v1.0.25 (08-06-2023)

Added documentation regarding the use of business unit custom identifiers

## v1.0.26 (14/09/2023)

Added additional documentation on Business Units, their custom identifiers and how they can be used in creating and searching for documents

Updated Enumeration values for line item units

Added extra clarifications for Authentication, Creation of Invoices etc.

Typos

Updated payloads for Invoice and Credit Notes

## v1.0.27 (06/10/2023)

Introduced 3% Tax Rate

## v1.0.28 (10/10/2023)

Added Credit Note statuses

## v1.0.29 (05/02/2024)

Orders API

## v1.0.30 (05/02/2024)

Added discount-specific details in line items. It is now possible to provide discounts by amount only.

## v1.0.31 (21/02/2024)

Added get shops endpoint on marketplace APIs

## v1.0.32 (06/03/2024)

Added errors desccription section

## v1.0.33 (07/06/2024)

Added paymentDescription and paymentType in Remittance Payments section

## v1.0.34 (12/07/2024)

Added description for the optionbal fields for Group/Mass payments on Remittance Submition and Validation

## v1.0.35 (07/08/2024)

Updated the Invoices flow diagram