

# *API v1*

---

## *REFERENCE GUIDE*

*Incassoservicedesk*

TRIVION | EIKENDREEF 4, 2350 VOSSELAAR

## Change Log

[illegible]

## Table of Contents

|     |                                        |    |
|-----|----------------------------------------|----|
| 1.  | ENVIRONMENTS.....                      | 3  |
| 2.  | GET-TOKEN .....                        | 5  |
| 3.  | SUBMIT-INVOICE .....                   | 8  |
| 4.  | SUBMIT-FILE.....                       | 12 |
| 5.  | SUBMIT-PAYMENT .....                   | 14 |
| 6.  | GET-FEEDBACK.....                      | 16 |
| 7.  | GET-FILEINFO .....                     | 19 |
| 8.  | GET-FILE-PAYMENT-PLAN-CONDITIONS ..... | 24 |
| 9.  | SUBMIT-FILE-PAYMENT-PLAN .....         | 26 |
| 10. | GET-STATUSCODES.....                   | 28 |
| 11. | SUBMIT-REMARK.....                     | 32 |

## 1. Environments

There are 2 environments available for the api and the application.

LIVE: <https://ims.trivion.be>

The screenshot shows the TRIVION LIVE application interface in a web browser. The browser's address bar displays 'ims.trivion.be'. The page has a header with the TRIVION logo and language options (NL - FR - EN). A left sidebar contains navigation links: 'ALGEMEEN' (highlighted in orange), 'RAPPORTEN', and 'UITLOGGEN'. The main content area is titled 'DOSSIERS' and 'Weikom DEMO'. It features a search form with fields for 'dossier nr van', 'tot dossier nr', 'Referentie', 'Toestandscade', 'debiteur', and 'Factuur'. There are also dropdowns for 'bedrag' and 'gelijk', a checkbox for 'Toon eveneens afgesloten dossiers', and an orange button labeled 'ZOEKOPDRACHT UITVOEREN'. A breadcrumb trail 'Algemeen->Dossiers' is visible in the top right.

TEST: <https://ims-dev.trivion.be>

The screenshot shows the TRIVION TEST application interface in a web browser. The browser's address bar displays 'ims-dev.trivion.be'. The page layout is identical to the LIVE version, but the sidebar is blue, and the 'ALGEMEEN' link is highlighted in blue. The main content area is titled 'DOSSIERS' and 'Weikom DEMO'. The search form fields and buttons are the same, but the 'ZOEKOPDRACHT UITVOEREN' button is blue. A new element, 'Database Date 20191012', is visible in the sidebar. The breadcrumb trail 'Algemeen->Dossiers' is also present.

In this document you will find the word {ENVIRONMENT}. This placeholder needs to be replaced with the right environment at the time of using. So, when you are testing your interface with our application you need to use TEST. And when everything is working fine and you've got a GO from our side, you need to use the LIVE.

## 2. get-token

A token is necessary to get access to the api of [ims.trivion.be](https://ims.trivion.be). To get a token you first need to request a `client_id` and a `client_secret` by sending a mail to [it@incassoservicedesk.be](mailto:it@incassoservicedesk.be).

These `client_id` and `client_secret` are strict confident and can only be used by 1 application. So do not share it.

To get a token you need to make a POST request.

|               |                             |
|---------------|-----------------------------|
| <b>Method</b> | POST                        |
| <b>Url</b>    | {ENVIRONMENT}/api/get-token |

With the following fields in the header.

|                      |                                                                             |
|----------------------|-----------------------------------------------------------------------------|
| <b>client-id</b>     | {client-id}                                                                 |
| <b>client-secret</b> | {client-secret}                                                             |
| <b>scope</b>         | One or more scopes separated by comma.<br>Depending by the next request(s). |
| <b>grant-type</b>    | client_credentials                                                          |

### Postman Sample

|                                     |                                          |                    |             |
|-------------------------------------|------------------------------------------|--------------------|-------------|
| POST                                | https://ims-dev.trivion.be/api/get-token | Send               | Save        |
| Params                              | Authorization                            | Headers (4)        | Body        |
| Pre-request Script                  | Tests                                    | Settings           | Cookies     |
| Code                                |                                          |                    |             |
| ▼ Headers (4)                       |                                          |                    |             |
|                                     | KEY                                      | VALUE              | DESCRIPTION |
| <input checked="" type="checkbox"/> | client-id                                | 5e3132449836a      |             |
| <input checked="" type="checkbox"/> | client-secret                            | c6389442313e5      |             |
| <input checked="" type="checkbox"/> | scope                                    | invoice            |             |
| <input checked="" type="checkbox"/> | grant-type                               | client_credentials |             |
| ...                                 | ...                                      | ...                | ...         |

The success result will be a JSON response like this.

```
{
  "status": 200,
  "code": "OK",
  "data": {
    "token_type": "Bearer",
    "expires_in": 3600,
    "access_token": "f1tMuasrmnqaZFD7YII0Zn1SNwszN3WwpH4jwMAKSSQlHUnktt0kCaItzbJsbv9tgEE2Hf2nXE9750ToCEfaRaKFU4a9geMgz1fJt9BDU8gFtLaeIax3ukgw01pPJH1bKDwhSsoCotFKK7kGw0TSsWYOHEV104c24UF58prdKoNFC6LdkYJUSE03CDEzUCcg3SuwsnmyBpoEu4zkQtDnUZ344HwdVOflNHv6KGcDvynji2goVE6Md6Eb3PH057pK5IqYAsYqmFgvdDlh39Y58TnAW2ixE8R2qa2UMUQRshMryHTMJTOodMZcf65rj6KBZK5NvsVbQpNfiZZKVXSuorG8iQ7HadZR89bswHppzMbeAmduumMBV9DKQc10fUeE6R9Tcgc1lSAV7Sp00bIhukiDCyQeuGPD0TF4vOo4SU6nI12T10kw6q8NNLTmzWJOILaQdAPvJrd4QTN6TpVfM0Fo4QspaNsD8o6mUxB3jVs7PVxfrjJ5JD4iGubky2LxyFWPyQX3gHA7FU1xrVXURz8dDnIUXGvtxxpyi431c3UMr5Ia6NEkQURzu0DBiEM77Rtx0jwsynmyJmgX9vyiNrU3ZOqnb8AKsZx00wkjWMidLukIVw35tqBrAfKgf6XBxCpe7Ydfc6Jq3EpdY7C1JfUWZ7N6mkyp2Vy14fnx4qL76piqQTQwnoVUshEzaYDZlHn186qaRqfd7szFVW1H1RMkFqxWfVbulJNzfgjJD8zriauryEPCF4oWwQrXnr4xRt2vUDqrCxPBDeWx3iQgyiwymK0qNtjh1ZSsCYVFfSfiQ0i24nItYSfrX6mX0h00adPctQASZajct38myJJ7Zv2WrsGevBcSV0AQnr2cpS0QvoHj1lukAwYKbrm9iIzPnj7qGZzCoIaaqT0kZQXdEq6gY00oh9c72gyG6nBMBIDMBNtxSLIKsyiIeKJ5peo0JHA9v0PYR09pD1SCBi91c9NXFI03b1X1HaucPJmj"
  }
}
```

In the data part you will get the following values

|              |                                                                                              |
|--------------|----------------------------------------------------------------------------------------------|
| token_type   | The type of the token                                                                        |
| expires_in   | Every token is only valid for 3600 seconds                                                   |
| access_token | This is unique text token of 1000 characters and needs to be used for all the other request. |

A failure response will look like this.

```
{  
  "status": 400,  
  "code": "NOK",  
  "error": "Json Request: Client Id and Client Secret not valid."  
}
```

### 3. submit-invoice

To get invoices in our system, you can send them through the “submit-invoice” method of our api. This is done invoice by invoice. But within our system we can decide whether we create a file per invoice or per debtor.

This method will only create a file once, and if you sent it multiple times it will only check and update the payments.

You will always get the same response.

Instead of passing basic authentication you need to add a token as an Authorization key in the header of your request. The token must be generated with ‘invoice’ included in the scope. Otherwise you cannot access this method.

To send an invoice to our api you need to make a POST request.

|               |                                  |
|---------------|----------------------------------|
| <b>Method</b> | POST                             |
| <b>Url</b>    | {ENVIRONMENT}/api/submit-invoice |

With the following fields in the header.

|                      |                             |
|----------------------|-----------------------------|
| <b>Authorization</b> | {token type} {access token} |
| <b>Content-Type</b>  | application/json            |

And in the body you need to add all your data.

|                 |                                                                                                                                                                                  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>username</b> | This will be sent to your customer. They can use it to login on our website. (mandatory)                                                                                         |
| <b>password</b> | This will be sent to your customer. They can use it to login on our website. (mandatory)                                                                                         |
| <b>name</b>     | The name of your customer (mandatory)                                                                                                                                            |
| <b>vatno</b>    | VAT Registration No                                                                                                                                                              |
| <b>address</b>  | Address of you customer (mandatory)                                                                                                                                              |
| <b>country</b>  | ISO country code                                                                                                                                                                 |
| <b>zip</b>      | Zip code                                                                                                                                                                         |
| <b>city</b>     | City name                                                                                                                                                                        |
| <b>birthday</b> | Day of birth (Optional)                                                                                                                                                          |
| <b>email</b>    |                                                                                                                                                                                  |
| <b>phone</b>    |                                                                                                                                                                                  |
| <b>mobile</b>   |                                                                                                                                                                                  |
| <b>lang</b>     | Language code                                                                                                                                                                    |
| <b>remarks</b>  |                                                                                                                                                                                  |
| <b>custno</b>   | Your internal no of your customer. (mandatory)                                                                                                                                   |
| <b>custtype</b> | Classification of your customer. Values: B2C, B2B or B2G (Optional)<br>When not provided the system will use the vatno as an indicator. When vatno is empty = B2C otherwise B2B. |

|                    |                                                                                                                                                              |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>payoutref</b>   | A reference can be added. The will be used when your account is setup to payout “Per file” instead of “Per period” (Optional)                                |
| <b>invoiceno</b>   | Your invoice no (mandatory). This can only be sent once.                                                                                                     |
| <b>date</b>        | Invoice Date (mandatory)                                                                                                                                     |
| <b>duedate</b>     | Due date of your invoice (mandatory)                                                                                                                         |
| <b>amount</b>      | Initial Amount including VAT (mandatory)                                                                                                                     |
| <b>outstanding</b> | Outstanding Amount (mandatory). This will be used for the file.                                                                                              |
| <b>payments</b>    | An Array of payments can be added. Full payments or partial. Every payment will be checked everytime and updated or created.                                 |
| <b>type</b>        | Optional: Indication of the type of payment. Values: payment, creditnote. Default: payment when not provided                                                 |
| <b>paymentref</b>  | Internal reference for payment. This must be unique per invoiceno                                                                                            |
| <b>date</b>        | Date of your payment                                                                                                                                         |
| <b>amount</b>      | Total amount paid                                                                                                                                            |
| <b>files</b>       | An Array of files can be added in Base64 string. For example, your invoice and reminder document. One file must contain a filename and the filebase64 string |
| <b>filename</b>    | The name of your document                                                                                                                                    |
| <b>filetype</b>    | Indication of the type of your document. Values: invoice, reminder, other. Default: invoice                                                                  |
| <b>filebase64</b>  | Base64 value of your document                                                                                                                                |

[illegible]

```
{
  "status": 200,
  "code": "OK",
  "data": {
    "invoiceno": "K123456",
    "filenr": "102000134",
    "filedate": "2020-03-14",
    "statuscode": "NEW",
    "statusdescription": "Nieuw dossier - Nouveau dossier",
    "statusdate": "2020-03-14",
    "debtor": {
      "name": "Trivion",
      "vatnr": "",
      "address": "",
      "zip_code": "",
      "city": "",
      "country_code": "BE",
      "telephone": "",
      "gsm": "",
      "e_mail": ""
    }
  }
}
```

A failure response will look like this.

```
{  
  "status": 401,  
  "code": "NOK",  
  "error": "outstanding is missing."  
}
```

#### 4. submit-file

Files can also be sent to our system after the submit-invoice. This can be done by using the “submit-file” method of our api.

Instead of passing basic authentication you need to add a token as an Authorization key in the header of your request. The token must be generated with ‘file included in the scope. Otherwise you cannot access this method.

To send an invoice to our api you need to make a POST request.

|               |                                  |
|---------------|----------------------------------|
| <b>Method</b> | POST                             |
| <b>Url</b>    | {ENVIRONMENT}/api/submit-payment |

With the following fields in the header.

|                      |                             |
|----------------------|-----------------------------|
| <b>Authorization</b> | {token type} {access token} |
| <b>Content-Type</b>  | application/json            |

And in the body you need to add all your data.

|                   |                                                                                                              |
|-------------------|--------------------------------------------------------------------------------------------------------------|
| <b>username</b>   | This will be sent to your customer. They can use it to login on our website. (mandatory)                     |
| <b>password</b>   | This will be sent to your customer. They can use it to login on our website. (mandatory)                     |
| <b>custno</b>     | Your internal no of your customer. (mandatory)                                                               |
| <b>type</b>       | Optional: Indication of the type of payment. Values: payment, creditnote. Default: payment when not provided |
| <b>paymentref</b> | Your payment reference (mandatory).                                                                          |
| <b>invoiceno</b>  | Your invoice no (mandatory).                                                                                 |
| <b>date</b>       | Invoice Date (mandatory)                                                                                     |
| <b>amount</b>     | Total payment amount (mandatory)                                                                             |

#### Postman Sample

The image displays two screenshots of the Postman application. The top screenshot shows a POST request to the URL `https://ims-dev.trivion.be/api/submit-invoice`. The 'Headers' tab is selected, showing two headers: 'Authorization' with the value 'Bearer OoY5KW6aOJQOK9yJlCsBUQfaxBK5tsgcjr9w...' and 'Content-Type' with the value 'application/json'. The bottom screenshot shows a POST request to the URL `https://ims-dev.trivion.be/api/submit-payment`. The 'Body' tab is selected, showing a JSON body with the following fields: 'username' (value: 'LINDAVAREL'), 'password' (value: 'xxx'), 'custno' (value: 'K123456'), 'paymentref' (value: 'CODA123'), 'invoiceno' (value: 'VF19/123465'), 'date' (value: '2020-01-01'), and 'amount' (value: '1000').

The success result will be a JSON response like this.

```
{  
  "status": 200,  
  "code": "OK",  
}
```

A failure response will look like this.

```
{  
  "status": 401,  
  "code": "NOK",  
  "error": "FILE DOES NOT EXISTS FOR INVOICE VF19/123466"  
}
```

## 5. submit-payment

Payments can also be sent to our system. This can be done by using the “submit-payment” method of our api. The payment will be added to your file.

Instead of passing basic authentication you need to add a token as an Authorization key in the header of your request. The token must be generated with ‘payment’ included in the scope. Otherwise you cannot access this method.

To send an invoice to our api you need to make a POST request.

|               |                                  |
|---------------|----------------------------------|
| <b>Method</b> | POST                             |
| <b>Url</b>    | {ENVIRONMENT}/api/submit-payment |

With the following fields in the header.

|                      |                             |
|----------------------|-----------------------------|
| <b>Authorization</b> | {token type} {access token} |
| <b>Content-Type</b>  | application/json            |

And in the body you need to add all your data.

|                   |                                                                                                              |
|-------------------|--------------------------------------------------------------------------------------------------------------|
| <b>username</b>   | This will be sent to your customer. They can use it to login on our website. (mandatory)                     |
| <b>password</b>   | This will be sent to your customer. They can use it to login on our website. (mandatory)                     |
| <b>custno</b>     | Your internal no of your customer. (mandatory)                                                               |
| <b>type</b>       | Optional: Indication of the type of payment. Values: payment, creditnote. Default: payment when not provided |
| <b>paymentref</b> | Your payment reference (mandatory).                                                                          |
| <b>invoiceno</b>  | Your invoice no (mandatory).                                                                                 |
| <b>date</b>       | Invoice Date (mandatory)                                                                                     |
| <b>amount</b>     | Total payment amount (mandatory)                                                                             |

### Postman Sample

The first screenshot shows a POST request to `https://ims-dev.trivion.be/api/submit-invoice`. The Headers tab is selected, showing two headers: `Authorization` with value `Bearer OoY5KW6aOJQOK9yJlCsBUQfaxBK5tsgcjr9w...` and `Content-Type` with value `application/json`.

The second screenshot shows a POST request to `https://ims-dev.trivion.be/api/submit-payment`. The Body tab is selected, showing a JSON body with the following fields: `username`: "LINDAVAREL", `password`: "xxx", `custno`: "K123456", `paymentref`: "CODA123", `invoiceno`: "VF19/123465", `date`: "2020-01-01", and `amount`: 1000.

The success result will be a JSON response like this.

```
{  
  "status": 200,  
  "code": "OK",  
}
```

A failure response will look like this.

```
{  
  "status": 401,  
  "code": "NOK",  
  "error": "FILE DOES NOT EXISTS FOR INVOICE VF19/123466"  
}
```

## 6. get-feedback

If you would like to get only feedback of a certain invoice, you can use the “get-feedback” method of our api.

Instead of passing basic authentication you need to add a token as an Authorization key in the header of your request. The token must be generated with ‘feedback’ included in the scope. Otherwise you cannot access this method.

To send an invoice to our api you need to make a POST request.

|               |                                |
|---------------|--------------------------------|
| <b>Method</b> | POST                           |
| <b>Url</b>    | {ENVIRONMENT}/api/get-feedback |

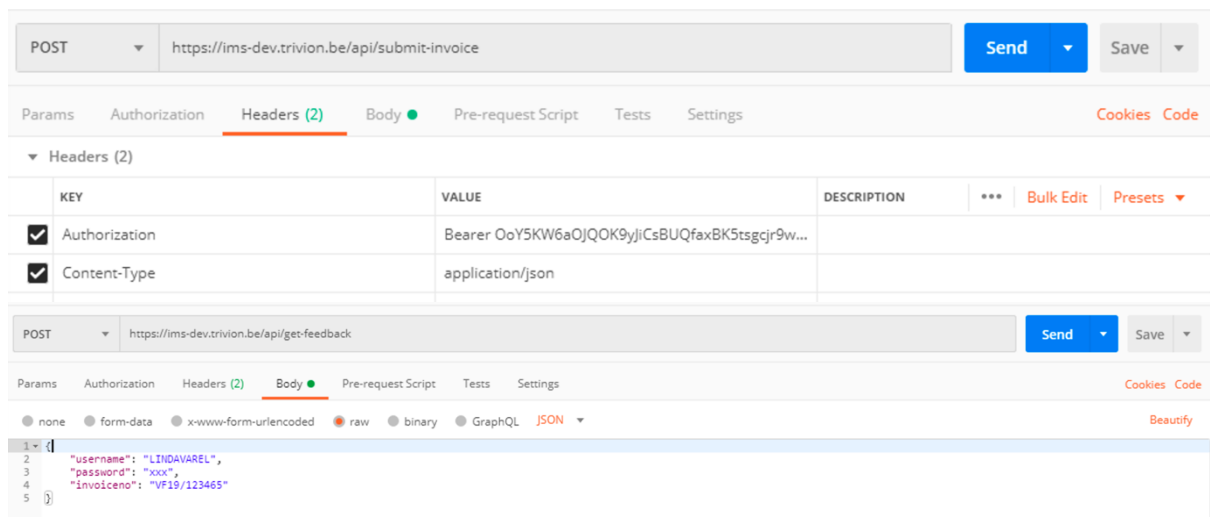
With the following fields in the header.

|                      |                             |
|----------------------|-----------------------------|
| <b>Authorization</b> | {token type} {access token} |
| <b>Content-Type</b>  | application/json            |

And in the body you need to add all your data.

|                  |                                                                                          |
|------------------|------------------------------------------------------------------------------------------|
| <b>username</b>  | This will be sent to your customer. They can use it to login on our website. (mandatory) |
| <b>password</b>  | This will be sent to your customer. They can use it to login on our website. (mandatory) |
| <b>invoiceno</b> | Your invoice no (mandatory).                                                             |

### Postman Sample



The success result will be a JSON response like this.

```
{
  "status": 200,
  "code": "OK",
  "data": {
    "invoiceno": "K123456",
    "filenr": "102000133",
    "filedate": "2020-02-12",
    "statuscode": "CLOSED",
    "statusdescription": "Gesloten dossier",
    "statusdate": "2020-02-12",
    "debtor": {
      "name": "Trivion",
      "vatnr": "",
      "address": "",
      "zip_code": "",
      "city": "",
      "country_code": "BE",
      "telephone": "",
      "gsm": "",
      "e_mail": ""
    }
  }
}
```

A failure response will look like this.

```
{
  "status": 401,
  "code": "NOK",
  "error": "FILE DOES NOT EXISTS FOR INVOICE VF19/123466"
}
```

## 7. get-fileinfo

With get-feedback you can already get feedback of a certain invoice, but when you want more details from a certain file you can use the “get-fileinfo” method of our api.

Instead of passing basic authentication you need to add a token as an Authorization key in the header of your request. The token must be generated with ‘fileinfo’ included in the scope. Otherwise, you cannot access this method.

To send an invoice to our api you need to make a POST request.

|               |                                |
|---------------|--------------------------------|
| <b>Method</b> | POST                           |
| <b>Url</b>    | {ENVIRONMENT}/api/get-fileinfo |

With the following fields in the header.

|                      |                             |
|----------------------|-----------------------------|
| <b>Authorization</b> | {token type} {access token} |
| <b>Content-Type</b>  | application/json            |

And in the body you need to add all your data.

|                 |                                                                                                       |
|-----------------|-------------------------------------------------------------------------------------------------------|
| <b>fileno</b>   | This is our internal fileno and can be found on our documents. (mandatory)                            |
| <b>password</b> | This is an unique password linked to the fileno. This can also be found on our documents. (mandatory) |

### Postman Sample

The first screenshot shows a Postman request configuration for a POST method to the URL `https://lms-dev.trivion.be/api/get-fileinfo`. The 'Headers' tab is selected, showing two headers: 'Authorization' with value 'Bearer {{authtoken}}' and 'Content-Type' with value 'application/json'. Both headers are checked as active.

The second screenshot shows the same request configuration but with the 'Body' tab selected. The body is set to 'raw' and 'JSON' format, containing the following JSON object:

```
1 {
2   "fileno": "257000011",
3   "password": "3Stoel5Kat7Maan"
4 }
```

The success result will be a JSON response like this.

```
{
  "status": 200,
  "code": "OK",
  "data": {
    "custno": "CL.005820",
    "fileno": 257000011,
    "accesscode": null,
    "creation_date": "2025-11-25",
    "statuscode": "TEL1",
    "statusdescription": "Telefonische actie wordt ingepland.",
    "statusdate": "2026-01-09",
    "closed": false,
    "closeddate": null,
    "amounts": {
      "total_amount": "40.00",
      "total_costs": "0.00",
      "total_interest": "0.00",
      "total_all": "40.00",
      "payed_amount": "0.00",
      "payed_costs": "0.00",
      "payed_interest": "0.00",
      "payed_unknown": "0.00",
      "payed_total": "0.00",
      "remaining_amount": "40.00",
      "remaining_costs": "0.00",
      "remaining_interest": "0.00",
      "remaining_total": "40.00"
    },
    "client": {
      "name": "DAP ANIMORETUS BV (NESTO)",
      "address": "JULES MORETUSLEI 171",
      "zip_code": "2610",
      "city": "WILRIJK"
    },
    "debtor": {
      "name": "FEJZAGIC ",
      "vatno": "",
      "address": "BOOMSESTEENWEG 85 BUS 3",
      "zip_code": "2610",
      "city": "WILRIJK (ANTWERPEN)",
      "country_code": "BE",
      "telephone": "",
      "mobile": "0465838105",
      "e_mail": "nihad_besic@hotmail.nl",
      "lang": "NL",
      "type": "B2C"
    },
    "references": [
      {
        "reference": "SI.069558",

```

```

        "creation_date": "2025-11-25",
        "start_date": "2024-07-12",
        "end_date": "2025-11-25",
        "invoice_date": "2024-07-02",
        "interest_percent": "12.00",
        "interest_minimum": "0.00",
        "cost_percent": "18.00",
        "cost_minimum": "25.00",
        "amount": "40.00",
        "costs": "0.00",
        "interest": "0.00",
        "total": "40.00"
    }
],
"payments": [],
"payment_plan": [],
"remarks": [],
"actions": [
    {
        "action_code": "NIEUW_VALIDATEER",
        "action_description": "Nieuw dossier (Data validatie)",
        "action_date": "2025-11-25",
        "remarks": ""
    },
    {
        "action_code": "NIEUW_CHECK",
        "action_description": "Nieuw dossier (na te kijken)",
        "action_date": "2025-11-25",
        "remarks": ""
    },
    {
        "action_code": "NIEUW",
        "action_description": "Nieuw dossier werd aangemaakt",
        "action_date": "2025-11-25",
        "remarks": ""
    },
    {
        "action_code": "BRF1",
        "action_description": "Eerste ingebrekestelling werd verstuurd.",
        "action_date": "2025-11-26",
        "remarks": ""
    },
    {
        "action_code": "BRF2",
        "action_description": "Tweede ingebrekestelling werd verstuurd.",
        "action_date": "2025-12-17",
        "remarks": ""
    }
]
}
}

```

## 8. get-file-payment-plan-conditions

To create a new payment plan, you need to know our conditions. These can be different per file so can use the “get-file-payment-plan-conditions” method of our api to get the correct info.

Instead of passing basic authentication you need to add a token as an Authorization key in the header of your request. The token must be generated with ‘fileinfo included in the scope. Otherwise, you cannot access this method.

To send an invoice to our api you need to make a POST request.

|               |                                                    |
|---------------|----------------------------------------------------|
| <b>Method</b> | POST                                               |
| <b>Url</b>    | {ENVIRONMENT}/api/get-file-payment-plan-conditions |

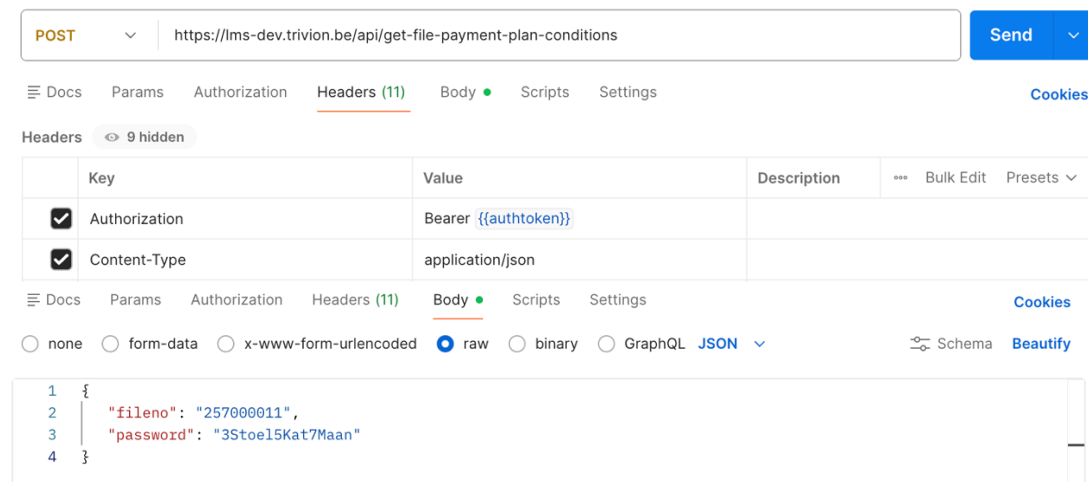
With the following fields in the header.

|                      |                             |
|----------------------|-----------------------------|
| <b>Authorization</b> | {token_type} {access_token} |
| <b>Content-Type</b>  | application/json            |

And in the body you need to add all your data.

|                 |                                                                                                       |
|-----------------|-------------------------------------------------------------------------------------------------------|
| <b>fileno</b>   | This is our internal fileno and can be found on our documents. (mandatory)                            |
| <b>password</b> | This is an unique password linked to the fileno. This can also be found on our documents. (mandatory) |

### Postman Sample



The success result will be a JSON response like this.

```
{
  "status": 200,
  "code": "OK",
  "data": {
    "min_payments": 1,
    "max_payments": 24,
    "days_until_first_payment": 10
  }
}
```

## 9. submit-file-payment-plan

After you have retrieved the payment plan conditions, you can simulate or create a new payment plan. The the “submit-file-payment-plan” method of our api can be used for this.

Instead of passing basic authentication you need to add a token as an Authorization key in the header of your request. The token must be generated with ‘fileinfo included in the scope. Otherwise, you cannot access this method.

To send an invoice to our api you need to make a POST request.

|               |                                            |
|---------------|--------------------------------------------|
| <b>Method</b> | POST                                       |
| <b>Url</b>    | {ENVIRONMENT}/api/submit-file-payment-plan |

With the following fields in the header.

|                      |                             |
|----------------------|-----------------------------|
| <b>Authorization</b> | {token_type} {access_token} |
| <b>Content-Type</b>  | application/json            |

And in the body you need to add all your data.

|                    |                                                                                                                      |
|--------------------|----------------------------------------------------------------------------------------------------------------------|
| <b>fileno</b>      | This is our internal fileno and can be found on our documents. (mandatory)                                           |
| <b>password</b>    | This is an unique password linked to the fileno. This can also be found on our documents. (mandatory)                |
| <b>start_date</b>  | The first day the paymentplan must start. From that day the system will repeat the no of payments each month.        |
| <b>no_payments</b> | This indicates the no of months you want to pay off. The max is indicated with the “get-file-payment-plan-conditions |
| <b>create</b>      | This boolean gives you the posibility to simulate the payment-plan (false), or accually create it (true)             |

### Postman Sample

POST https://lms-dev.trivion.be/api/submit-file-payment-plan

Headers (11)

| Key           | Value                | Description |
|---------------|----------------------|-------------|
| Authorization | Bearer {{authtoken}} |             |
| Content-Type  | application/json     |             |

Body (raw)

```
1 {
2   "fileno": "257000011",
3   "password": "3Stoe15Kat7Maan",
4   "start_date": "2026-01-31",
5   "no_payments": 4,
6   "create": false
7 }
```

The success result will be a JSON response like this.

```
{
  "status": 200,
  "code": "OK",
  "data": [
    {
      "payment_no": 1,
      "payment_date": "2026-01-31",
      "creation_date": "2026-01-25",
      "amount": 10
    },
    {
      "payment_no": 2,
      "payment_date": "2026-03-03",
      "creation_date": "2026-01-25",
      "amount": 10
    },
    {
      "payment_no": 3,
      "payment_date": "2026-03-31",
      "creation_date": "2026-01-25",
      "amount": 10
    },
    {
      "payment_no": 4,
      "payment_date": "2026-05-01",
      "creation_date": "2026-01-25",
      "amount": 10
    }
  ]
}
```

A failure response will look like this.

```
{
  "status": 400,
  "code": "NOK",
  "error": "no_payments is missing."
}
```

## 10. get-statuscodes

In the submit-invoice and get-feedback the system returns a “`statusCode`”. To get the full list of all the active statuscodes you can call the “get-statuscodes” method of our api.

Instead of passing basic authentication you need to add a token as an Authorization key in the header of your request. The token can be generated with all of the available scopes, but at least one. Otherwise you cannot access this method.

To send an invoice to our api you need to make a POST request.

|               |                                   |
|---------------|-----------------------------------|
| <b>Method</b> | POST                              |
| <b>Url</b>    | {ENVIRONMENT}/api/get-statuscodes |

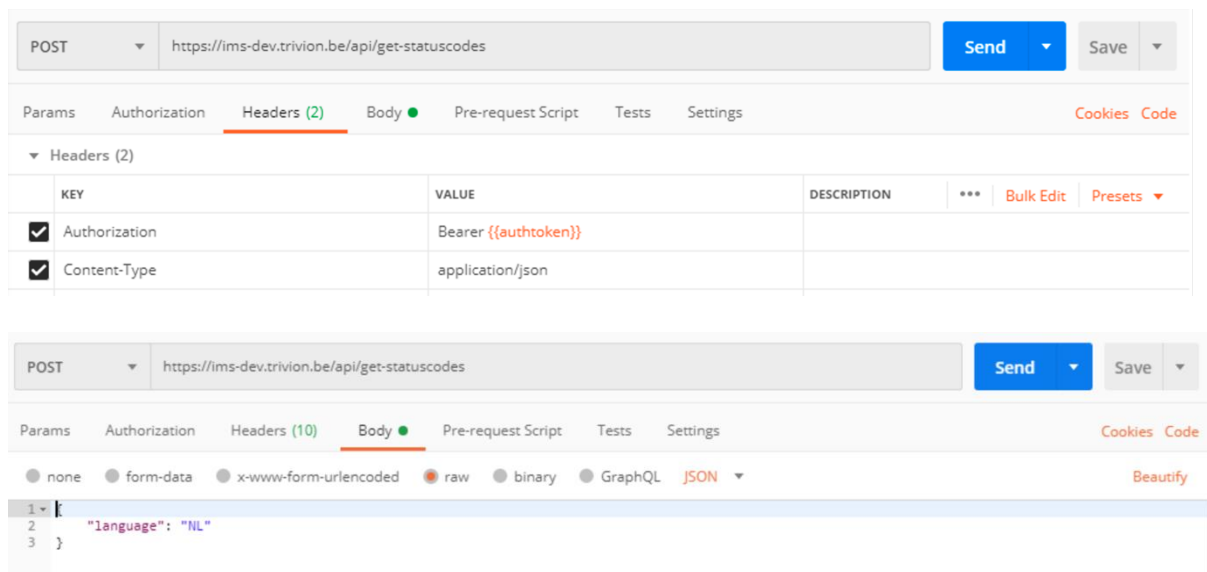
With the following fields in the header.

|                      |                             |
|----------------------|-----------------------------|
| <b>Authorization</b> | {token type} {access token} |
| <b>Content-Type</b>  | application/json            |

And in the body you need to add all your data.

|                 |                                                                 |
|-----------------|-----------------------------------------------------------------|
| <b>language</b> | Language code. NL = Dutch, FR = French, EN = English (Optional) |
|-----------------|-----------------------------------------------------------------|

### Postman Sample



The success result will be a JSON response like this.

```
{
  "status": 200,
  "code": "OK",
  "data": [
    {
      "code": "AANG SCHULDVORDERING",
      "description": "Aangifte schuldvordering"
    },
    {
      "code": "ADRES",
      "description": "Adres aanvraag"
    },
    {
      "code": "ADRES_HER",
      "description": "Adresaanvraag herinnering"
    },
    {
      "code": "ADV1",
      "description": "Dossier bij advocaat"
    },
    {
      "code": "AFBETALINGSOVEREENKO",
      "description": "AFBETALINGSOVEREENKOMST"
    },
    {
      "code": "AGENDA",
      "description": "Dossier te bekijken"
    },
    {
      "code": "AGENT",
      "description": "via agent"
    },
    {
      "code": "BETAALBELOFTE",
      "description": "Betaalbelofte"
    },
    {
      "code": "BETAALPLAN",
      "description": "Betaalplan overeengekomen"
    },
    {
      "code": "BEV_DAGV",
      "description": "Bevestig dagvaarding"
    },
    {
      "code": "BEZOEK",
      "description": "Bezoek"
    },
    {
      "code": "BEZOEKVERSLAG",
      "description": "Bezoekverslag"
    },
    {
      "code": "BEZOEKVERSLAG_AANM",
      "description": "Bezoekersverslag aanmaken"
    },
    {
      "code": "BRF1",
      "description": "Eerste aanmaning"
    },
    {
      "code": "BRF1_ZONDER_KOSTEN",
      "description": "Brief1 zonder kosten"
    },
    {
      "code": "BRF2",
      "description": "Tweede aanmaning"
    },
    {
      "code": "BRF3",
      "description": "Derde aanmaning"
    },
    {
      "code": "BRIEF",
      "description": "BRIEF AAN DEBITEUR"
    },
    {
      "code": "BUITENLAND",
      "description": "Buitenland"
    },
    {
      "code": "BZKVERSLAG_GEKOPPELD",
      "description": "Bezoekverslag gekoppeld"
    },
    {
      "code": "CLOSED",
      "description": "Gesloten dossier"
    },
    {
      "code": "DAGV_VOORSTEL",
      "description": "Voorstel tot dagvaarding"
    },
    {
      "code": "DISPUUT",
      "description": "Dispuut"
    },
    {
      "code": "DW",
      "description": "Doorgeven aan deurwaarder"
    },
    {
      "code": "DW2",
      "description": "Bij deurwaarder"
    },
    {
      "code": "E-MAIL",
      "description": "e-mail"
    },
    {
      "code": "EXPLOIT",
      "description": "Exploit toegevoegd"
    },
    {
      "code": "HUISBEZOEK",
      "description": "huisbezoek"
    },
    {
      "code": "HUISBEZOEK UITGV",
      "description": "Huisbezoek uitgevoerd"
    }
  ],
}
```

```

{
  "code": "KLACHT_CONSUMENTENB",
  "description": "Klacht consumentenbond"
},
{
  "code": "KLACHT ECC",
  "description": "Klacht ECC"
},
{
  "code": "KLACHT TESTAANKOOP",
  "description": "Klacht Testaankoop"
},
{
  "code": "LAATSTE AANMANING",
  "description": "ALLERLAATSTE INGEBREKESTELLING AANGET./ONTVANGKAART"
},
{
  "code": "NEW",
  "description": "Nieuw dossier - Nouveau dossier"
},
{
  "code": "NEW_TOCHECK",
  "description": "The credit info that was found needs to be looked at."
},
{
  "code": "PAUZE",
  "description": "Afwachtend op antwoord"
},
{
  "code": "REOPEN",
  "description": "Factuur opnieuw betaalbaar"
},
{
  "code": "RINGRING SMS LIST",
  "description": "Ringring sms lijst"
},
{
  "code": "RINGRING_TEL_LIST",
  "description": "Ringring telefoon lijst"
},
{
  "code": "SCHULDBEKENTENIS",
  "description": "Schuldbekentenis"
},
{
  "code": "TEL.CONTACT",
  "description": "Telefonisch contact"
},
{
  "code": "TEL1",
  "description": "Telefonische actie"
},
{
  "code": "TEL1_DONE",
  "description": "A phone call was logged."
},
{
  "code": "TEL2",
  "description": "Telefonische actie 2"
},
{
  "code": "TEL_TTS",
  "description": "Telephone text to speech"
},
{
  "code": "VONNIS",
  "description": "Vonnis bekomen"
}
]
}

```

A failure response will look like this.

```

{
  "status": 400,
  "code": "NOK",
  "error": "INVALID TOKEN"
}

```

## 11. submit-remark

Remarks can also be sent to our system. This can be done by using the “submit-remark” method of our api. The remark will be added to your file.

Instead of passing basic authentication you need to add a token as an Authorization key in the header of your request. The token must be generated with ‘remark’ included in the scope. Otherwise you cannot access this method.

To send an invoice to our api you need to make a POST request.

|               |                                 |
|---------------|---------------------------------|
| <b>Method</b> | POST                            |
| <b>Url</b>    | {ENVIRONMENT}/api/submit-remark |

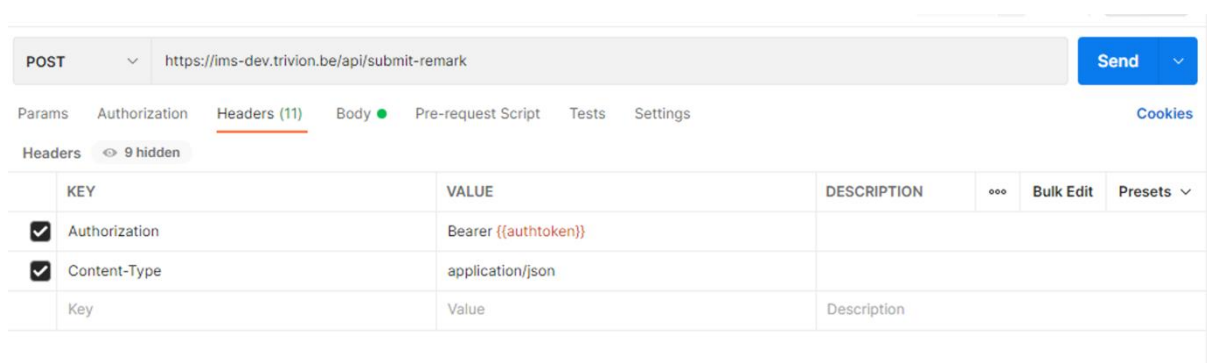
With the following fields in the header.

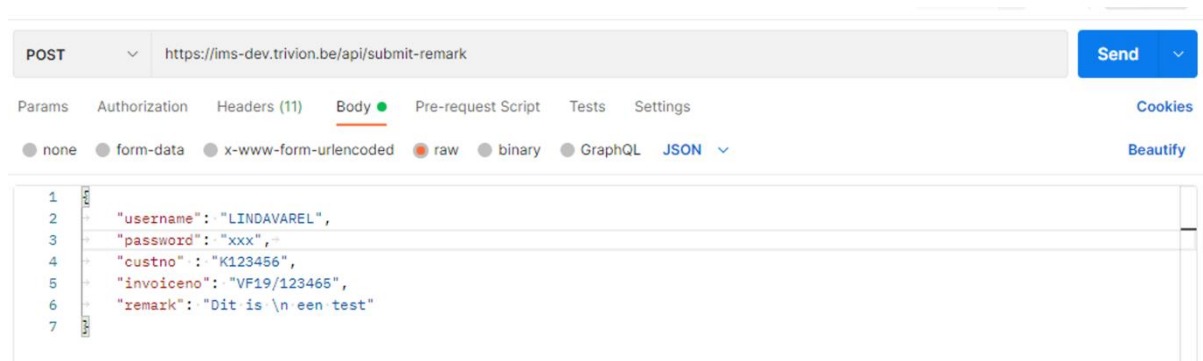
|                      |                             |
|----------------------|-----------------------------|
| <b>Authorization</b> | {token type} {access token} |
| <b>Content-Type</b>  | application/json            |

And in the body you need to add all your data.

|                  |                                                                                          |
|------------------|------------------------------------------------------------------------------------------|
| <b>username</b>  | This will be sent to your customer. They can use it to login on our website. (mandatory) |
| <b>password</b>  | This will be sent to your customer. They can use it to login on our website. (mandatory) |
| <b>custno</b>    | Your internal no of your customer. (mandatory)                                           |
| <b>invoiceno</b> | Your invoice no (mandatory). This can only be sent once.                                 |
| <b>date</b>      | Remark Date (mandatory).                                                                 |
| <b>remark</b>    | This can be a multiline text with \n as line delimiter. (mandatory)                      |

### Postman Sample





The success result will be a JSON response like this.

```
{
  "status": 200,
  "code": "OK",
}
```

A failure response will look like this.

```
{
  "status": 401,
  "code": "NOK",
  "error": "FILE DOES NOT EXISTS FOR INVOICE VF19/123466"
}
```