

Unilog API 1.0 - Rest



Documentation Version 1.0.0



LOGISTICS INTEGRATION GUIDE

Unilog Express

API FEATURES

"TO BE EFFECTIVE IN FULFILLING YOUR GOALS, YOU MUST BREAK THEM DOWN INTO LITTLE TASKS AND THEN SPEND MOST OF YOUR TIME WORKING ON THE TASK IN FRONT OF YOU, INSTEAD OF BEING OVERWHELMED BY THE ENORMITY OF THE BIG GOAL." — SUNDAY ADELAJA,

Introduction

With the birth of new technologies and the emergence of Logistics 4.0, Unilog Express, always seeks to be at the forefront aiming at a continuous improvement in its logistics processes.

We offer our customers a technology designed with Cloud Microservices and Dockers using REST, always seeking Integrity, Reliability and security for our data transactions with our customers, in every logistical aspect.

Why REST?

REST or RESTful APIs were designed to take advantage of existing protocols. While REST - or Representational State Transfer - can be used over nearly any protocol, when used for web APIs it typically takes advantage of HTTP. ... One of the key advantages of REST APIs is that they provide a great deal of flexibility.

Victor Lima
Unilog Express Brazil

RESOURCES	page
- Introduction.....	05
- Authentication Resources.....	06 - 07
- Product Registration Resources.....	08 - 14
- How to query the processing result.....	15 - 16
- Goods Receipt Document Resources.....	17 - 25
- Confirmation of Goods Received.....	21 - 24
- Completing an integration as read.....	25
- Sales Order Resources.....	26 - 32
- Confirmation Sales Order Check out.....	33 - 36
- Completing an integration as read.....	37
- Invoice Resources.....	38 - 43
- Quarantine Transfer Resources.....	44 - 45
- Goods Return Resources.....	46 - 49
- Glossary.....	50

INTRODUCTION

Overview

Through the Logistics API it is possible to integrate your application with WMS. The API provides POST and GET methods.

Staging

Need for Homologation

The approval of each method is of paramount importance, since without it the integration will not be able to integrate with the WMS, meeting the quality requirements.

Link to call the resources in the Staging environment

API Resources :

<http://api.unilog-dev.cloud.bytecode.software/swagger-ui.html>

AUTHENTICATION RESOURCES

To authenticate with the API, the Bearer type will be used using a security token.

A username and password must be requested for the Projects / IT area to authenticate with the API.

Being authenticated in the API, you can make requests using POST / GET methods.

The authenticated session limit is 30 minutes, outside this time it will be necessary to authenticate again requesting a new token to access API resources.

How to authenticate

POST: /auth <http://api.unilog-dev.cloud.bytecode.software/auth>

Headers :

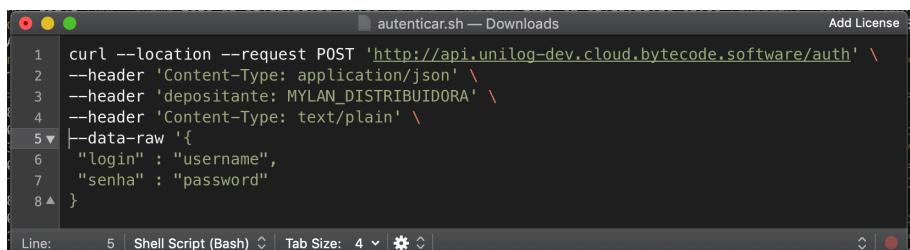
Key	Value	Description
Content-Type	application/json	Content type Json format
Depositante	MYLAN_DISTRIBUIDORA	The amount will be passed on by Unilog's technology team, after registering with the API.

Body :

```
1 ▼ {  
2   "login" : "username",  
3   "senha" : "password"  
4 }
```

You will receive a username and password generated by the Projects team to connect

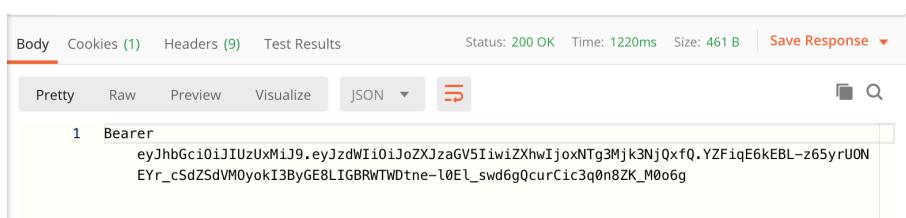
Example in code:



```
autenticar.sh — Downloads Add License  
1 curl --location --request POST 'http://api.unilog-dev.cloud.bytecode.software/auth' \  
2 --header 'Content-Type: application/json' \  
3 --header 'depositante: MYLAN_DISTRIBUIDORA' \  
4 --header 'Content-Type: text/plain' \  
5 ▼ |--data-raw '{  
6   "login" : "username",  
7   "senha" : "password"  
8 ▲ }  
Line: 5 | Shell Script (Bash) | Tab Size: 4 | ⚙ |
```

Result after sending a POST:

Body :



Body Cookies (1) Headers (9) Test Results Status: 200 OK Time: 1220ms Size: 461 B Save Response ▾

Pretty Raw Preview Visualize JSON  

```
1 Bearer  
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXJzaGV5IiwidXhwIjoxNTg3Mjk3NjQxfQ.YZFiqE6kEBL-z65yrU0N  
EYr_cSdZsdVM0yokI3ByGE8LIGBRWTWdtne-l0El_swd6gQcurCic3q0n8ZK_M0o6g
```

PRODUCT REGISTRATION RESOURCES

To submit a new product registration or change, you can use the POST method following the examples below:

POST: /produtos

Headers :

Key	Value	Description
Content-Type	application/json	Content type Json format
Depositante	MYLAN_DISTRIBUIDORA	The amount will be passed on by Unilog's technology team, after registering with the API.

Body :

The screenshot shows a code editor window with the following details:

- Title Bar:** products.json — Downloads
- Add License:** button in the top right corner.
- Code Content:** A JSON object with nested arrays and objects. The main object has a single key "produtos" which contains an array of product objects. Each product object has numerous properties like classeProduto, classificacaoFiscal, cnpjCpfDepositante, etc. There are also arrays for unidades (units) and tipos (types).
- Numbered Lines:** Line numbers from 1 to 42 are displayed on the left side of the code area.
- Status Bar:** Shows "Line: 35:23 | JSON" and "Tab Size: 4".

```
1 ▼ {  
2 ▼   "produtos": [{  
3     "classeProduto": "00",  
4     "classificacaoFiscal": "30032069",  
5     "cnpjCpfDepositante": "04429377000979",  
6     "codigo": "088828",  
7     "codigoCategoria": "1135",  
8     "codigoEmpresa": "04429377000979",  
9     "descricao": "BAR SP DRK CRANBERRY 4X10X100G",  
10    "descricaoCategoria": "BARRAS SPECIAL",  
11    "descricaoGrupo": "",  
12    "grupo": "",  
13    "informacaoAdicional1": "",  
14    "informacaoAdicional2": "",  
15    "ncm": "",  
16    "prazoValidade": 0,  
17    "produtoPesado": 0,  
18    "situacaoProduto": "1",  
19    "situacaoTributaria": 0,  
20    "temperaturaMaximaArmazenagem": 0,  
21    "temperaturaMediaArmazenagem": 0,  
22    "temperaturaMinimaArmazenagem": 0,  
23    "tipo": "UN",  
24    "unidades": [  
25      {"altura": 213,  
26        "codigoBarra": "7898292888283",  
27        "codigoEmpresa": "04429377000626",  
28        "codigoProduto": "088828",  
29        "comprimento": 408,  
30        "descricao": "CAIXA",  
31        "fator": 40,  
32        "informacaoAdicional1": "",  
33        "informacaoAdicional2": "",  
34        "largura": 125,  
35        "pesoBruto": 4.355,|  
36        "pesoLiquido": 4,  
37        "tipo": "CX",  
38        "tipoCodigoBarra": "GTIN14"  
39    }]  
40  }]  
41 }]  
42 }
```

Product - Body Description Header : produtos

Field	Value	Description
classeProduto	00	Product status, 00 - free
classificacaoFiscal	Fixed value = 30032069	obligatory
cnpjCpfDeposintante	Cod ID CNPJ registration Brazil	obligatory
codigo	Product cod	obligatory
codigoCategoria	Empty	Not obligatory
codigoEmpresa	Cod ID CNPJ registration Brazil	obligatory
descricao	Name product	obligatory
descricaoCategoria	Name category	Not obligatory
descricaoGrupo	Name group	Not obligatory
descrigrupocodigoGrupo	Cod group	Not obligatory
informacaoAdicional1	Note 1	Not obligatory

informacaoAdicional2	Note 2	Not obligatory
ncm	Cod NCM format 0000.00.00	NCM code is the acronym for Mercosur Common Nomenclature.
prazoValidade	Due date product	Not obligatory
produtoPesado	Heavy product value 0	Not obligatory
situacaoProduto	Product active = 1 or deactivate = 0	obligatory
situacaoTributaria	Value 0,	Not obligatory
temperaturaMaximaArmazenagem	Maximum storage temperature	Not obligatory
temperaturaMediaArmazenagem	Average storage temperature	Not obligatory
temperaturaMinimaArmazenagem	Minimum storage temperature	Not obligatory
temperaturaMediaArmazenagem	Minimum storage temperature	Not obligatory
tipo	Unit type Fixed value = "UN"	obligatory

Product - Body Description Detail : unidades

Field	Value	Description
altura	height value	obligatory
codigoBarra	Barcode number EAN or DUN14	obligatory
codigoEmpresa	Cod ID CNPJ registration Brazil	obligatory
codigoProduto	Product cod	obligatory
comprimento	length value	obligatory
descricao	Description value exemplo : CAIXA	obligatory
fator	factor number quantity Example: Box with 40 Units	obligatory
informacaoAdicional1	Note 1	Not obligatory
informacaoAdicional2	Note 2	Not obligatory
Largura	width value	obligatory
pesoBruto	gross weight value	obligatory
peloLiquido	weight product value	obligatory

tipo	Type. Example: If the detail record is a UN Unit, for a CX Box	obligatory
tipoCodigoBarra	Example: If the detail record is a UN Unit, the value is "GTIN13", for a CX Box, the value is "GTIN14"	obligatory

Code example POST:

```
curl --location --request POST 'http://api.unilog-dev.cloud.bytecode.software/produtos' \
--header 'Content-Type: application/json' \
--header 'depositante: MYLAN_DISTRIBUIDORA' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1nJ9.eyJwdWIiOiJpZSJzaGV5IiwiZXhwIjoxNTg3MzE2MjU4fQ. \
\0yuZL2GxXAHdfwq7J0fh6n2BgNmNQjXKCzdLn2ys7p0ieBCPx8RGiEA3__0mP5Rj1I4cGjg9USErF0p5Uw' \
--header 'Content-Type: text/plain' \
--data-raw '{

  "produtos": [
    {
      "classeProduto": "00",
      "classificacaoFiscal": "30032069",
      "cnpjDepositante": "04429377000979",
      "codigo": "088828",
      "codigoCategoria": "1135",
      "codigoEmpresa": "04429377000979",
      "descricao": "BAR SP DRK CRANBERRY 4X10X100G",
      "descricaoCategoria": "BARRAS SPECIAL",
      "descricaoGrupo": "",
      "grupo": "",
      "informacaoAdicional1": "",
      "informacaoAdicional2": "",
      "ncm": "",
      "prazoValidade": 0,
      "produtoPesado": 0,
      "situacaoProduto": "",
      "situacaoAduanaria": 0,
      "situacaoAduanariaOrigem": 0,
      "temperaturaMaximaArmazenagem": 0,
      "temperaturaMediaArmazenagem": 0,
      "temperaturaMinimaArmazenagem": 0,
      "tipo": "UN",
      "tipoIntegracao": 0,
      "umidadeMedia": 0,
      "unidades": [
        {
          "altura": 3,
          "codigoBarra": "78982928823",
          "codigoEmpresa": "04429377000626",
          "codigoProduto": "088828",
          "comprimento": 28,
          "descricao": "UNIDADE",
          "fator": 1,
          "informacaoAdicional1": "",
          "informacaoAdicional2": "",
          "largura": 25,
          "pesoBruto": 1.355,
          "pesoLiquido": 1,
          "pesoVariavel": 0,
          "sequenciaUnidadeAcondicionamento": 0,
          "tipo": "UN",
          "tipoCodigoBarra": "GTIN13"
        },
        {
          "altura": 213,
          "codigoBarra": "7898292888283",
          "codigoEmpresa": "04429377000626",
          "codigoProduto": "088828",
          "comprimento": 408,
          "descricao": "CAIXA",
          "fator": 40,
          "informacaoAdicional1": "",
          "informacaoAdicional2": "",
          "largura": 125,
          "pesoBruto": 4.355,
          "pesoLiquido": 4,
          "pesoVariavel": 0,
          "sequenciaUnidadeAcondicionamento": 0,
          "tipo": "CX",
          "tipoCodigoBarra": "GTIN14"
        }
      ]
    }
  ]
}'
```

Line: 27:37 | Shell Script (Bash) ◉ | Tab Size: 4 ▾ ⚙ ◉

Result after sending a POST:

Body : id sequence number

Status code http : 201 Created

A screenshot of a Postman interface. At the top, there are tabs for Body, Cookies (1), Headers (10), and Test Results. On the right, it shows 'Status: 201 Created', 'Time: 1837ms', 'Size: 384 B', and a 'Save Response' button. Below these, there are buttons for Pretty, Raw, Preview, Visualize, and Text. A red arrow points from the text above to the 'Status: 201 Created' text in the Postman interface. The main content area shows a single row with the value '1 2853904'.

Receiving created status 201, after 2 minutes, you can search if the material registration was processed successfully, you need to do a GET:

GET: /integracoes/{sequentialIntegracao}

Headers :

Key	Value	Description
Content-Type	application/json	Content type Json format
Depositante	MYLAN_DISTRIBUIDORA	Headers previously registered

Example:

A terminal window titled 'curl-integracoes.sh — mylan'. It contains a curl command with line numbers 1 to 4. The command is: curl --location --request GET 'http://api.unilog-dev.cloud.bytecode.software/integracoes/2853904' \ --header 'depositante: MYLAN_DISTRIBUIDORA' \ --header 'Authorization: Bearer eyJhbGciOiJIUzI1NjQ9.eyJzdWIiOiJ0ZXJzaGV5IiwidXNlIjoxNTg2NzAxNzQ4fQ.' \ RL63LATzBzA1uUiLix9wZ6tX38xFKFMHM5FnsgDIt_5UMJOXo83jIAH5633KX8KGg1TTo1rZNDrDdE-mr0fa' \

Obs : You need to pass in the parameter by the received sequence

GET : /integracoes/[2853904](#)

Result after sending a GET:

Body :

```
{  
    "sequencia": 2853904,  
    "tipo": 1,  
    "estado": 2,  
    "dataLog": "2020-04-08",  
    "romaneio": null,  
    "processadaComSucessoPeloWms": true,  
    "processadaComErroPeloWms": false  
}
```

Explanation of the received Json result :

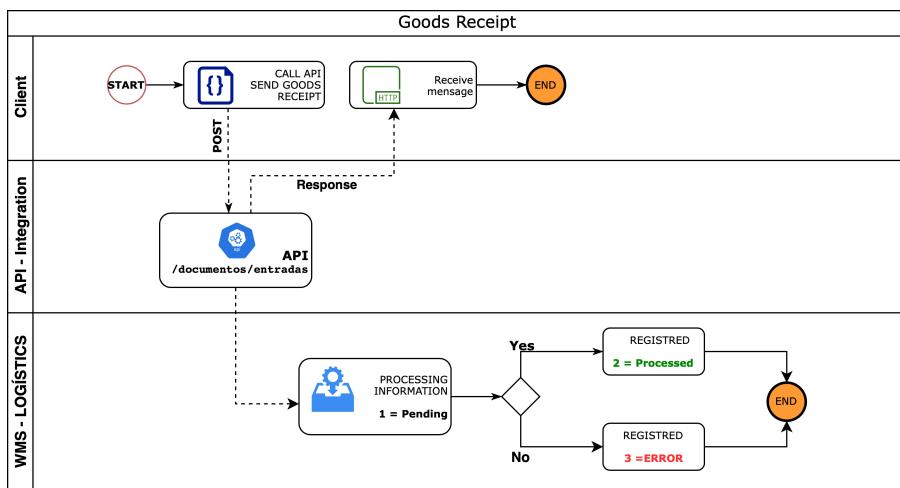
Field	Value	Description
sequencia	Number : 2853904	Number received from the integration sent previously
tipo	1 = Material Registration 101 = Goods Receipt /Goods Return 203 = Sales Order 206 = Sales Invoice 342 = batch approval process	internal WMS integration codes
estado	1 = Pending 2 = Processed 3 = Error	1 When processing is pending 2 When processing was successful
dataLog	Date	Registration date
processadaComSucessoPeloWms	True = when the integration was successful False = when integration has occurred an error	Checks whether the integration processed successfully or not
processadaComErroPeloWms	True = when the integration was successful False = when integration has occurred an error	Checks whether the integration processed successfully or not

GOODS RECEIPT DOCUMENT RESOURCES

THIS API FEATURE IS USED TO SEND GOODS TO LOGISTICS.

Steps:

- 1 - Unilog Express receives the goods receipt through the API.
- 2 - The Client receives an API response: 201 Created.
- 3 - The WMS Logistics System, processes the information in 2 minutes.

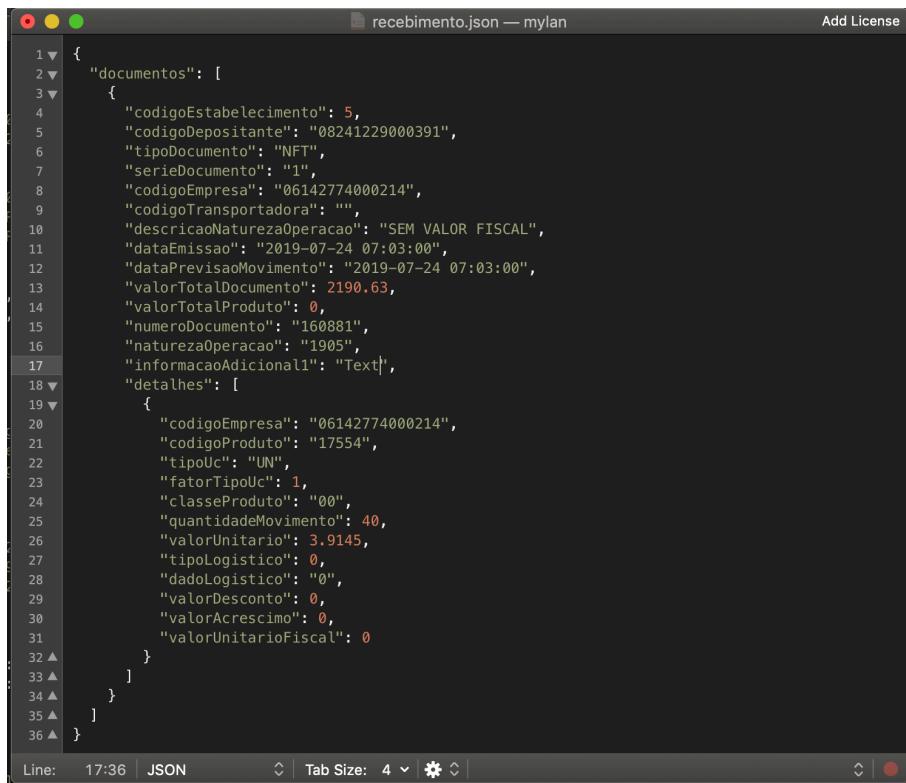


POST: /documentos/entradas

Headers :

Key	Value	Description
Content-Type	application/json	Content type Json format
Depositante	MYLAN_DISTRIBUIDORA	Headers previously registered

Body :



The screenshot shows a code editor window with the file 'recebimento.json' open. The JSON structure is as follows:

```
{ "documentos": [ { "codigoEstabelecimento": 5, "codigoDepositante": "08241229000391", "tipoDocumento": "NFT", "serieDocumento": "1", "codigoEmpresa": "06142774000214", "codigoTransportadora": "", "descricaoNaturezaOperacao": "SEM VALOR FISCAL", "dataEmissao": "2019-07-24 07:03:00", "dataPrevisaoMovimento": "2019-07-24 07:03:00", "valorTotalDocumento": 2190.63, "valorTotalProduto": 0, "numeroDocumento": "160881", "naturezaOperacao": "1905", "informacaoAdicional1": "Text", "detalhes": [ { "codigoEmpresa": "06142774000214", "codigoProduto": "17554", "tipoUc": "UN", "fatorTipoUc": 1, "classeProduto": "00", "quantidadeMovimento": 40, "valorUnitario": 3.9145, "tipoLogistico": 0, "dadosLogistico": "0", "valorDesconto": 0, "valorAcrescimo": 0, "valorUnitarioFiscal": 0 } ] } ] }
```

The code editor interface includes line numbers on the left, a status bar at the bottom showing 'Line: 17:36 | JSON', and various toolbars and status indicators.

Goods receipt - Boby Description Header : documentos

Field	Value	Description
codigoEstabelecimento	1	Customer Deposit Number in Logistics
codigoDepositante	Number cliente example (999999999999) 14 digits	Registered customer number in Brazil
tipoDocumento	Customer Document Type, example: "NFT"	NFT is Transfer Invoice
serieDocumento	Example : 1 ou 2	Invoice Serial Number
codigoEmpresa	Number example Cnpj (999999999999) 14 digits	number of the end customer registered in Brazil
codigoTransportadora	Number example (999999999999) 14 digits	Registered carrier number in Brazil
descricaoNaturezaOperacao	Fixed "SEM VALOR FISCAL"	Fixed value
dataEmissao	Date	Invoice Date
dataPrevisaoMovimento	Date	Movement Date
valorTotalDocumento	Example : 2190.63	Total amount of Invoice with Taxes
valorTotalProduto	Example : 2090.00	Total amount of invoice without taxes
numeroDocumento	Example : 0002313	Invoice number
naturezaOperacao	Fixes : "1905"	Fixed value
informacaoAdicional1	Note	Additional information

Goods receipt - Body Description Detail : details

Field	Value	Description
codigoEmpresa	Number example (999999999999) 14 digits	CNPJ number of the end customer registered in Brazil
codigoProduto	Product cod	obligatory
tipoUc	"UN"	Measure in Unit
fatorTipoUc	Values = 1	One unit
classeProduto	Example: "00"	Material Status, example: 00 is Free
quantidadeMovimento	Value Item Quantity	Item Quantity
valorUnitario	Unit value of the item	
tipoLogistico	Fixed Value = 3	Indicates that works with batch
dadoLogistico	Bath example : "L1234"	Product batch

Result after sending a POST:

Body : id sequence number
Status code http : 201 Created



```
Body Cookies (1) Headers (10) Test Results Status: 201 Created Time: 2.99s Size: 384 B Save Response ▾
```

Pretty Raw Preview Visualize Text

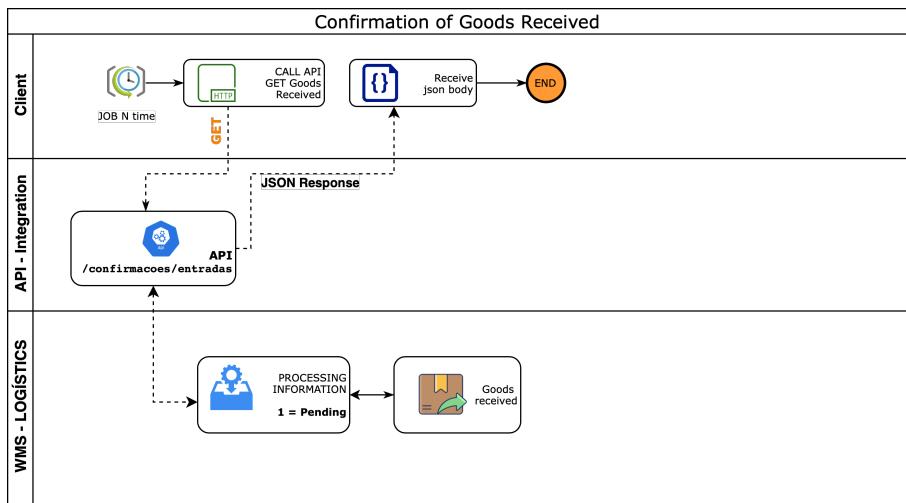
1 2853991



The asynchronous result of the processing is the same for all integrations, it can be retrieved with a GET method in the example specified on page 15 and 16.

Confirmation of Goods Received

After checking and entering the goods received in stock, you are available to retrieve the result of confirmation of Goods in stock.



Steps :

- 1 - With a job running from time to time, the customer calls the API with a GET to retrieve confirmations for Goods Received.
- 2 - The API queries the pending data in the WMS database.
- 3 - The result of the call in the API is given through a JSON format of the goods receipt data.

GET: /integracoes/{sequenciaIntegracao}

Headers :

Key	Value	Description
Content-Type	application/json	Content type Json format
Depositante	MYLAN_DISTRIBUIDORA	Headers previously registered

Example JSON Response:

```
[  
  {  
    "estadoIntegracao": "1",  
    "sequenciaIntegracao": 2851573,  
    "data": "2020-03-26 17:46:13",  
    "serieDocumento": "16",  
    "numeroDocumento": "000157812",  
    "quantidadeVolume": 0,  
    "codigoDepositante": "04429377000979",  
    "sentido": "E",  
    "peso": 0,  
    "detalhes": [  
      {  
        "classeProduto": "00",  
        "codigoProduto": "040022",  
        "fatorTipoUc": "192",  
        "quantidadeInicial": 384,  
        "quantidadeMovimento": 2,  
        "tipoUc": "CX",  
        "dataVencimento": "2021-07-30",  
        "loteFabricacao": "L1234"  
      },  
      {  
        "classeProduto": "00",  
        "codigoProduto": "040026",  
        "fatorTipoUc": "360",  
        "quantidadeInicial": 720,  
        "quantidadeMovimento": 2,  
        "tipoUc": "CX",  
        "dataVencimento": "2021-07-30",  
        "loteFabricacao": "L1234"  
      }  
    ]  
  }]
```

Confirmation of Goods Received - Boby Header

Field	Value	Description
estadoIntegracao	1 = Pending	Status integration
sequenciaIntegracao	Number integration sequence	Number integration
data	Date	Date
serieDocumento	Values = 1	One unit
numeroDocumento	Number invoice	Material Status, example: 00 is Free
quantidadeVolume	Value volume Quantity	Item Quantity
codigoDepositante	Unit value of the item	
sentido	Fixed "E"	Indicates Goods Receipt
peso	Bath example : "L1234"	Product batch

Confirmation of Goods Received - Boby detail

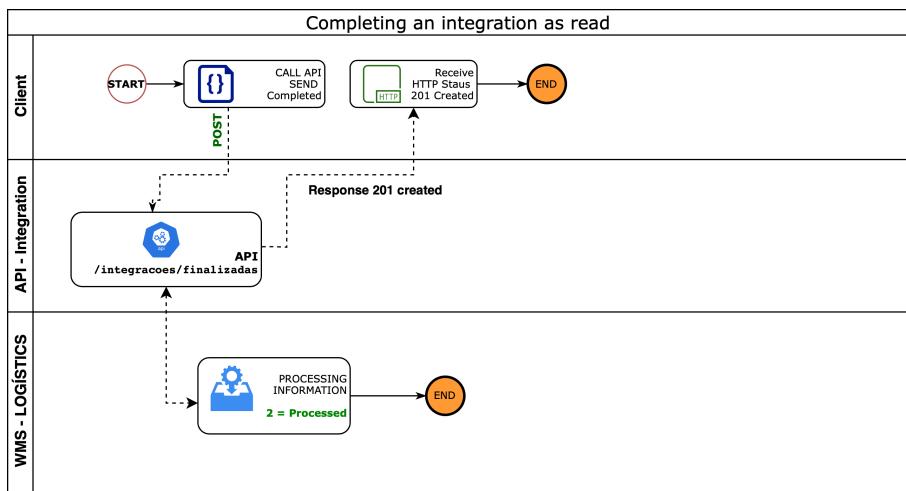
Field	Value	Description
classeProduto	1 = Pending	Status integration
sequenciaIntegracao	Number integration sequence	Number integration
codigoProduto	Date	Date
fatorTipoUc	Values = 1	One unit
quantidadeInicial	Number invoice	Material Status, example: 00 is Free
quantidadeMovimento	Value volume Quantity	Item Quantity
tipoUc	CX = Box	
dataVencimento	"2021-07-30"	batch Due Date
loteFabricacao	Bath example : "L1234"	Product batch

Completing an integration as read

How to complete a record reading in the integration ?

Should be called an API, with a **POST** method, make a call in:

POST : /integracoes/finalizadas



Steps:

Client: make a call to the API with a POST method and send a JSON with the integration sequence retrieved with GET.

API: receives the call and processes the update in the WMS database.

WMS: updates the integration record so that it does not appear in the next record readings calls, ending the cycle of any reading process.

Body :

```
{  
  "sequencia": 2851573  
}
```

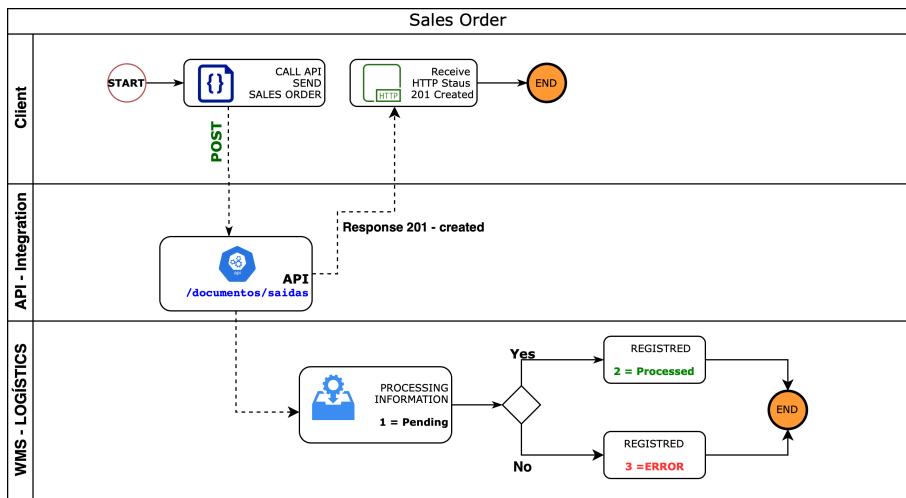
Response :

201 created - The request was fulfilled and resulted in an update to the previously created resource.

Sales Order Resources

How to send a sales order via API to Logistics ?

A Sales Order sent to Logistics will be registered in the WMS database and will be processed according to the flow below:



Client: make a call to the API with a POST method and send a JSON with the Sales Order.

API: receives the call and records the Sales Order in the WMS database.

WMS: records the integration and internally performs a 2m asynchronous process to process the registration, which can be Successfully Processed or any eventual error.

POST: `/documentos/saidas`

Headers :

Key	Value	Description
Content-Type	application/json	Content type Json format
Depositante	MYLAN_DISTRIBUIDORA	Headers previously registered

Sales Order Resources - Body JSON

```
 1 ▼ {  
 2   "documentos": [  
 3     {"agrupador": 1,  
 4      "codigoEstabelecimento": 1,  
 5      "codigoDepositante": "04429377000979",  
 6      "codigoEmpresa": "03476811042237",  
 7      "codigoTransportadora": "03476811042237",  
 8      "numeroDocumento": "000129",  
 9      "serieDocumento": "1",  
10      "tipoDocumento": "PED",  
11      "datamissao": "2020-03-31 19:42:19",  
12      "dataPrevisaoMovimento": "2020-03-31 19:42:19",  
13      "naturezaOperacao": "0000",  
14      "descricaoNaturezaOperacao": "SEM VALOR FISCAL",  
15      "quantidadeVolume": 0,  
16      "valorTotalDocumento": 30388.779,  
17      "valorTotalProduto": 30000.213,  
18      "nfeChaveAcesso": "",  
19      "marcaVolume": "",  
20      "centroCusto": "RODOVIARIO",  
21      "cnpjCpfEmpresa": "03476811042237",  
22      "tipoPessoa Empresa": "J",  
23      "nomeEmpresa": "NAME CUSTUMER ",  
24      "enderecoEmpresa": "ROD ALEXANDRE BALBO SN",  
25      "numeroEnderecoEmpresa": "233",  
26      "bairroEmpresa": "ENG CARLA DE",  
27      "complementoEmpresa": "BLOC AP",  
28      "municipioEmpresa": "VITORIA",  
29      "cepEmpresa": "140057800",  
30      "ufEmpresa": "ES",  
31      "inscricaoEmpresa": "string",  
32      "cnpjCpfTransportadora": "03476811042237",  
33      "tipoPessoaTransportadora": "J",  
34      "nomeTransportadora": "CARRIER NAME",  
35      "enderecoTransportadora": "RUA X , 231",  
36      "bairroTransportadora": "PORTAL JACARAPE",  
37      "municipioTransportadora": "SERRA",  
38      "cepTransportadora": "29000321",  
39      "ufTransportadora": "ES",  
40      "inscricaoTransportadora": "string",  
41      "informacaoAdicional1": "NOTE 1",  
42      "informacaoAdicional2": "NOTE 2",  
43      "informacaoAdicional3": "NOTE 3",  
44      "detalhes": [  
45        {"classeProduto": "00",  
46        "codigoEmpresa": "03476811042237",  
47        "codigoProduto": "088349",  
48        "dadoLogistico": "LT1234",  
49        "fatorTipoUc": 1,  
50        "quantidadeMovimento": 25,  
51        "tipoLogistico": 3,  
52        "tipoUc": "UN",  
53        "valorUnitario": 128.049737  
54      }]  
55    }]  
56  }  
Line: 50:29 | JSON      ◁ | Tab Size: 4 ▾ ⚙ ◁ |
```

Sales Order Resources - Boby Header

Field	Value	Description
agrupador	Number	Sales Order group number Used to group multiple orders
codigoEstabelecimento	1	Customer Deposit Number in Logistics
codigoDepositante	Number cliente example (9999999999999) 14 digits	Registered customer number in Brazil
codigoEmpresa	Number example Cnpj (9999999999999) 14 digits	number of the end customer registered in Brazil
codigoTransportadora	Number example (9999999999999) 14 digits	Registered carrier number in Brazil
numeroDocumento	Number example (9999999999999) 14 digits	Registered carrier number in Brazil
serieDocumento	Example : 1 ou 2	Order Serial Number or Invoice serial number
tipoDocumento	Value = "SO" or "INV"	Example: SO = Sales Order or INV = Invoice.
dataEmissao	Date	Sales order Date
dataPrevisaoMovimento	Date	Movement Date
naturezaOperacao	Fixed value : "0000"	Obligatory
descricaoNaturezaOperacao	Fixed value : "SEM VALOR FISCAL"	Obligatory
quantidadeVolume	Number	Sales Order Volume
valorTotalDocumento	Example : 2190.63	Total amount with Taxes

<code>valorTotalProduto</code>	Example : 2090.00	Total amount without taxes
<code>nfeChaveAcesso</code>	Value = "" or "number access key here"	If sales order = Empty If Invoice = Document access key
<code>marcaVolume</code>	Tracking number	Tracking number for post office
<code>centroCusto</code>	Example : "RODOVIARIO" = Road	Transport mode
<code>cnpjCpfEmpresa</code>	Number : "99999999999999"	CPF or CNPJ number of the final customer
<code>tipoPessoaEmpresa</code>	Legal Company = "J" Individual = "F"	Kind of person
<code>nomeEmpresa</code>	Value = "Company name"	Name
<code>enderecoEmpresa</code>	Value = "Street name"	Address
<code>numeroEnderecoEmpresa</code>	Value = "1241"	Number Street
<code>bairroEmpresa</code>	Value = "Neighborhood name"	Neighborhood
<code>complementoEmpresa</code>	Value = "complement"	Address complement
<code>municipioEmpresa</code>	Value = "City name"	City
<code>cepEmpresa</code>	Value = "Zip code"	Zip code
<code>ufEmpresa</code>	Value = "UF" Example : "RJ" = Rio de Janeiro	Federation unit
<code>inscricaoEmpresa</code>	Value Example = "123412"	State registration number in Brazil
<code>cnpjCpfTransportadora</code>	Number : "99999999999999"	CPF or CNPJ number of the final customer

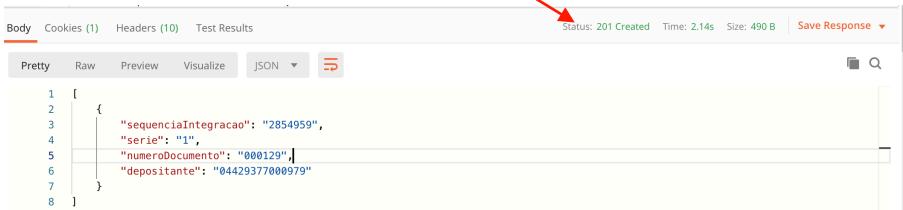
tipoPessoaTransportadora	Legal Company = "J"	Kind of person
nomeTransportadora	Value = "Carrier Name"	Carrier Name
enderecoTransportadora	Value = "Street name"	Address
bairroTransportadora	Value = "1241"	Number Street
municipioTransportadora	Value = "Neighborhood name"	Neighborhood
cepTransportadora	Value = "Zip code"	Zip code
ufTransportadora	Value = "UF" Example : "RJ" = Rio de Janeiro	Federation unit
inscricaoTransportadora	Value Example = "123412"	State registration number in Brazil
informacaoAdicional1	Value : "text 1"	Note 1
informacaoAdicional2	Value : "text 2"	Note 2
informacaoAdicional3	Value : "text 3"	Note 3

Sales Order Resources - Boby detail

Field	Value	Description
<code>classeProduto</code>	Example: "00"	Material Status, example: 00 is Free
<code>codigoEmpresa</code>	Number example (999999999999) 14 digits	CNPJ number of the end customer registered in Brazil
<code>codigoProduto</code>	Product cod	obligatory
<code>dadoLogistico</code>	Bath example : "L1234"	Product batch
<code>fatorTipoUc</code>	Values = 1	One unit
<code>quantidadeMovimento</code>	Value Item Quantity	Item Quantity
<code>tipoLogistico</code>	Fixed Value = 3	Indicates that works with batch
<code>tipoUc</code>	"UN"	Measure in Unit
<code>valorUnitario</code>	Unit value of the item	

Result after sending a POST:

Body : id sequence number



A screenshot of a browser or API tool interface showing a successful POST request. At the top, there are tabs for 'Body' (highlighted in red), 'Cookies (1)', 'Headers (10)', and 'Test Results'. On the right, it shows 'Status: 201 Created', 'Time: 2.14s', and 'Size: 490 B'. Below these are buttons for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'JSON'. The main area displays a JSON object with the following structure:

```
1 [  
2 {  
3   "sequenciaIntegracao": "2854959",  
4   "serie": "1",  
5   "numeroDocumento": "000129",  
6   "depositante": "04429377000979"  
7 }  
8 ]
```

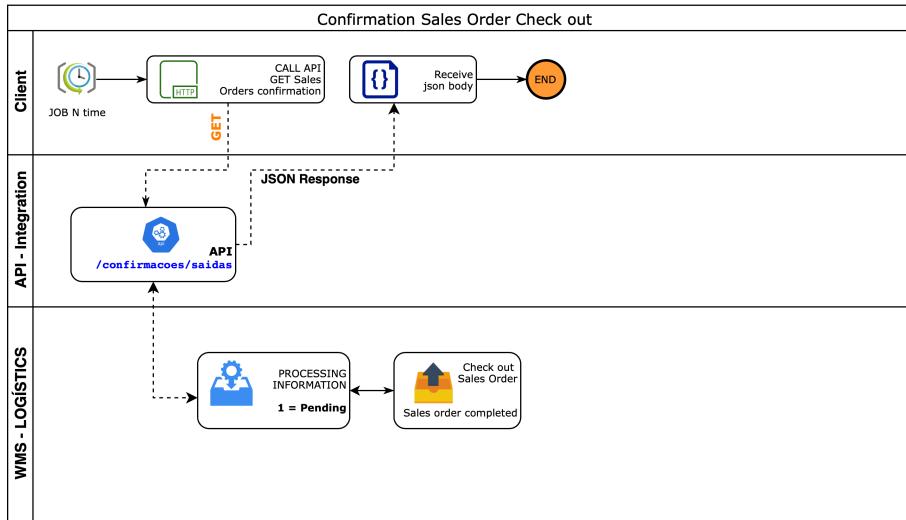
Status code http : 201 Created



The asynchronous result of the processing is the same for all integrations, it can be retrieved with a GET method in the example specified on page 15 and 16.

Confirmation Sales Order Check out

After picking and checking the product on the sales order, the result will be available in pending status.



Steps :

- 1 - With a job running from time to time, the customer calls the API with a GET to retrieve the check of the finished sales order products.
- 2 - The API queries the pending data in the WMS database.
- 3 - The result of the call in the API, data will be sent in a Json format of the sales order data.

GET: /integracoes/{sequenciaIntegracao}

Headers :

Key	Value	Description
Content-Type	application/json	Content type Json format
Depositante	MYLAN_DISTRIBUIDORA	Headers previously registered

Example JSON Response:

```
[  
  {  
    "estadoIntegracao": "1",  
    "sequenciaIntegracao": 2851573,  
    "data": "2020-03-26 17:46:13",  
    "serieDocumento": "16",  
    "numeroDocumento": "000157812",  
    "quantidadeVolume": 0,  
    "codigoDepositante": "04429377000979",  
    "sentido": "E",  
    "peso": 0,  
    "detalhes": [  
      {  
        "classeProduto": "00",  
        "codigoProduto": "040022",  
        "fatorTipoUc": "192",  
        "quantidadeInicial": 384,  
        "quantidadeMovimento": 2,  
        "tipoUc": "CX",  
        "dataVencimento": "2021-07-30",  
        "loteFabricacao": "L1234"  
      },  
      {  
        "classeProduto": "00",  
        "codigoProduto": "040026",  
        "fatorTipoUc": "360",  
        "quantidadeInicial": 720,  
        "quantidadeMovimento": 2,  
        "tipoUc": "CX",  
        "dataVencimento": "2021-07-30",  
        "loteFabricacao": "L1234"  
      }  
    ]  
  }]
```

Confirmation of Goods Received - Boby Header

Field	Value	Description
estadoIntegracao	1 = Pending	Status integration
sequenciaIntegracao	Number integration sequence	Number integration
data	Date	Date
serieDocumento	Values = 1	One unit
numeroDocumento	Number invoice	Material Status, example: 00 is Free
quantidadeVolume	Value volume Quantity	Item Quantity
codigoDepositante	Unit value of the item	
sentido	Fixed "E"	Indicates Goods Receipt
peso	Bath example : "L1234"	Product batch

Confirmation of Goods Received - Boby detail

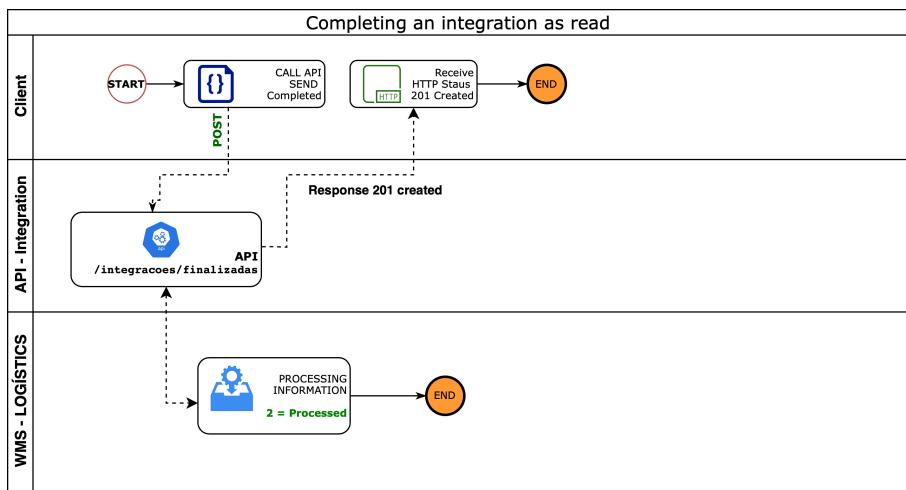
Field	Value	Description
classeProduto	1 = Pending	Status integration
sequenciaIntegracao	Number integration sequence	Number integration
codigoProduto	Date	Date
fatorTipoUc	Values = 1	One unit
quantidadeInicial	Number invoice	Material Status, example: 00 is Free
quantidadeMovimento	Value volume Quantity	Item Quantity
tipoUc	CX = Box	
dataVencimento	"2021-07-30"	batch Due Date
loteFabricacao	Bath example : "L1234"	Product batch

Completing an integration as read

How to complete a record reading in the integration ?

Should be called an API, with a POST method, make a call in:

POST : /integracoes/finalizadas



Steps:

Client: make a call to the API with a POST method and send a JSON with the integration sequence retrieved with GET.

API: receives the call and processes the update in the WMS database.

WMS: updates the integration record so that it does not appear in the next record readings calls, ending the cycle of any reading process.

Body :

```
{  
  "sequencia": 2851573  
}
```

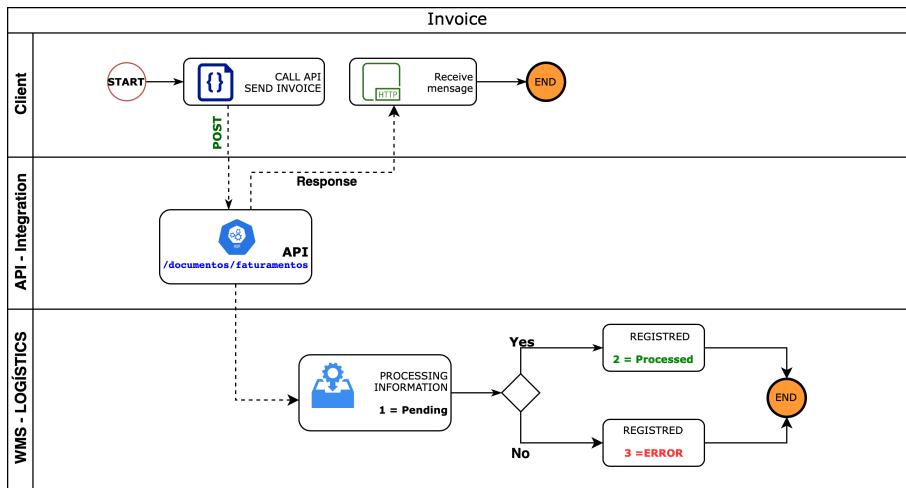
Response :

201 created - The request was fulfilled and resulted in an update to the previously created resource.

Invoice Resources

How to send an Invoice to Logistics ?

An invoice sent to Logistics will be registered in the WMS database and will be processed according to the flow below.



Steps:

Client: make a call to the API with a POST method and send a JSON with the Invoice

API: receives the call and records the Invoice in the WMS database.

WMS: records the integration and internally performs a 2m asynchronous process to process the registration, which can be Successfully Processed or any eventual error.

POST: </documentos/faturamentos>

Headers :

Key	Value	Description
Content-Type	application/json	Content type Json format
Depositante	MYLAN_DISTRIBUIDORA	Headers previously registered

Invoice Resources - Body JSON

To send an invoice through the API using the POST method, only the invoice header and the Sales Order header must be sent.

With this, the Logistics system will make a match of the Sales Order data and the Invoice.

Invoice Resources - Boby Header Invoice

Field	Value	Description
codigoEstabelecimento	1	Customer Deposit Number in Logistics
codigoDepositante	Number cliente example (999999999999) 14 digits	Registered customer number in Brazil
codigoEmpresa	Number example Cnpj (999999999999) 14 digits	number of the end customer registered in Brazil
codigoTransportadora	Number example (999999999999) 14 digits	Registered carrier number in Brazil
cnpjCpfEmpresa	Number : "99999999999999"	CPF or CNPJ number of the final customer
cnpjCpfTransportadora	Number : "99999999999999"	CPF or CNPJ number of the final customer
tipoDocumento	Value = "INV" or "NF" in Brazil	Example: INV = Invoice.
serieDocumento	Example : 1 ou 2	Invoice serial number
numeroDocumento	Example : "0000012345"	Invoice number
naturezaOperacao	Fixed value : "0000"	Obligatory
descricaoNaturezaOperacao	Fixed value : "SEM VALOR FISCAL"	Obligatory
dataEmissao	Date	Invoice Date
dataPrevisaoMovimento	Date	Movement Date
valorTotalDocumento	Example : 2190.63	Total amount with Taxes
valorTotalProduto	Example : 2090.00	Total amount without taxes

<code>nfeChaveAcesso</code>	Value = access key here	Invoice = Document access key
<code>quantidadeVolume</code>	Number	Invoice Volume
<code>centroCusto</code>	Example : "RODOVIARIO" = Road	Transport mode
<code>agrupador</code>	Fixed Number : "1"	Group value

Invoice Resources - Boby Header Sales Order

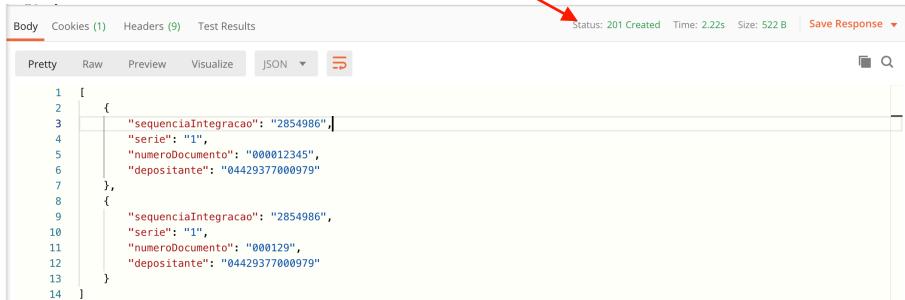
Field	Value	Description
<code>codigoEstabelecimento</code>	1	Customer Deposit Number in Logistics
<code>codigoDepositante</code>	Number cliente example (999999999999) 14 digits	Registered customer number in Brazil
<code>codigoEmpresa</code>	Number example Cnpj (999999999999) 14 digits	number of the end customer registered in Brazil
<code>codigoTransportadora</code>	Number example (999999999999) 14 digits	Registered carrier number in Brazil
<code>cnpjCpfEmpresa</code>	Number : "99999999999999"	CPF or CNPJ number of the final customer
<code>cnpjCpfTransportadora</code>	Number : "99999999999999"	CPF or CNPJ number of the final customer
<code>tipoDocumento</code>	Value = "PED"	Example: PED = Sales Order or "SO".
<code>serieDocumento</code>	Example : 1 ou 2	Sales Order serial number
<code>numeroDocumento</code>	Example : "000129"	Sales Order number

<code>naturezaOperacao</code>	Fixed value : "0000"	Obligatory
<code>descricaoNaturezaOperacao</code>	Fixed value : "SEM VALOR FISCAL"	Obligatory
<code>dataEmissao</code>	Date	Invoice Date
<code>dataPrevisaoMovimento</code>	Date	Movement Date
<code>valorTotalDocumento</code>	Example : 2190.63	Total amount with Taxes
<code>valorTotalProduto</code>	Example : 2090.00	Total amount without taxes
<code>nfeChaveAcesso</code>	Value = ""	No Obligatory
<code>quantidadeVolume</code>	Number	Invoice Volume
<code>centroCusto</code>	Example : "RODOVIARIO" = Road	Transport mode
<code>agrupador</code>	Fixed Number : "1"	Group value

Result after sending a POST:

Body

Status code http : 201 Created



Body Cookies (1) Headers (9) Test Results Status: 201 Created Time: 2.22s Size: 522 B Save Response ▾

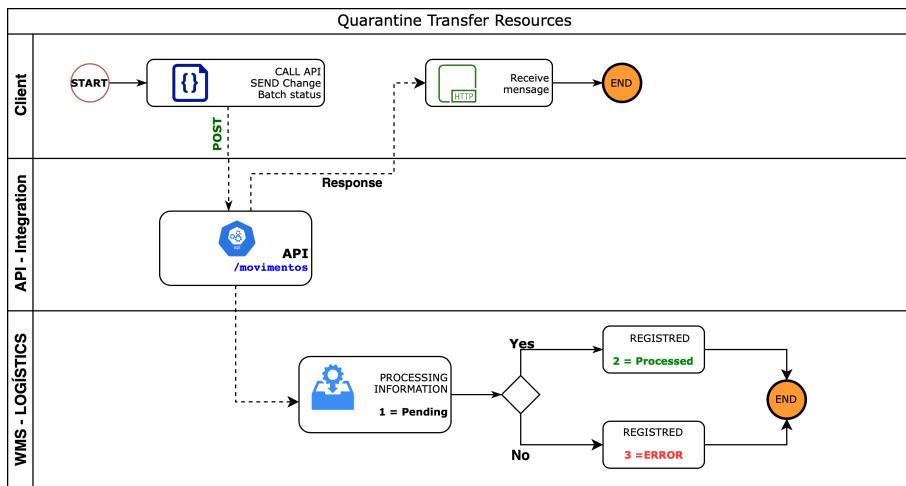
Pretty Raw Preview Visualize JSON

```
1 [  
2 {  
3   "sequenciaIntegracao": "2854986",  
4   "serie": "1",  
5   "numeroDocumento": "000012345",  
6   "depositante": "04429377000979"  
7 },  
8 {  
9   "sequenciaIntegracao": "2854986",  
10  "serie": "1",  
11  "numeroDocumento": "000129",  
12  "depositante": "04429377000979"  
13 }  
14 ]
```

The out bound process ends with the sending of the invoice. This is the last step.

Quarantine Transfer Resources

How to send a batch approval process to Logistics.
Quarantine movements for available for sale.



Steps:

Client: make a call to the API with a POST method and send a JSON with the lot and status information

API: receives the call and records the Invoice in the WMS database.

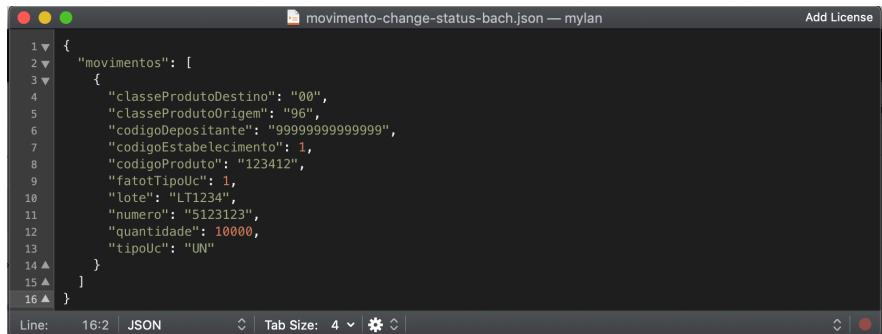
WMS: records the integration and internally performs a 2m asynchronous process to process the registration, which can be Successfully Processed or any eventual error.

POST: [/movimentos](#)

Response :

201 created - The request was fulfilled and resulted in a new resource being created.

Quarantine change Status batch - Boby



```
1 ▼ {  
2 ▼   "movimentos": [  
3 ▼     {  
4       "classeProdutoDestino": "00",  
5       "classeProdutoOrigem": "96",  
6       "codigoDepositante": "9999999999999999",  
7       "codigoEstabelecimento": 1,  
8       "codigoProduto": "123412",  
9       "fatotTipoUc": 1,  
10      "lote": "LT1234",  
11      "numero": "5123123",  
12      "quantidade": 10000,  
13      "tipoUc": "UN"  
14    }  
15  ]  
16 }  
Line: 16:2 | JSON | Tab Size: 4 | * |
```

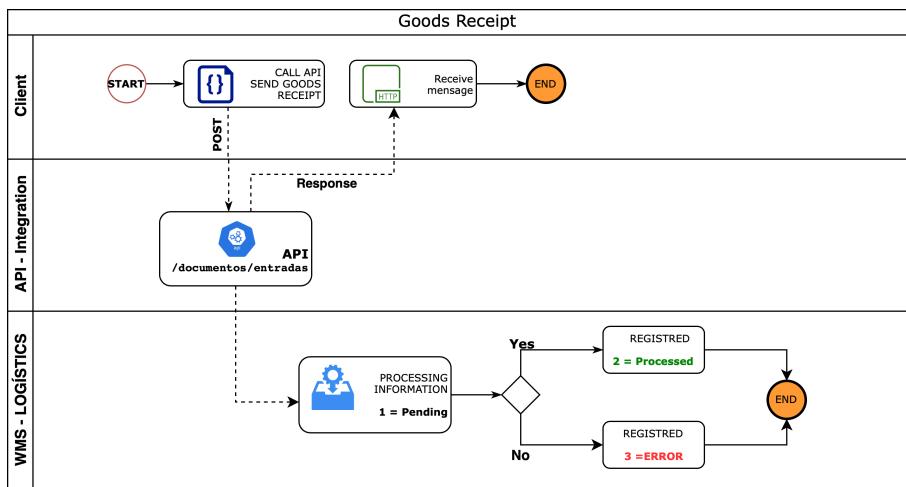
Field	Value	Description
classeProdutoDestino	Value = "00"	In this example I am removing from quarantine to available
classeProdutoOrigem	Value = "96"	Quanratin to available
codigoDepositante	Number example Cnpj (999999999999) 14 digits	number of the end customer registered in Brazil
codigoEstabelecimento	1	Customer Deposit Number in Logistics
codigoProduto	Number example : "123455"	Material code
fatotTipoUc	Values = 1	One unit
lote	Value example ="LT1234"	product batch
numero	Value example = "5123123"	request number
quantidade	Value example = 10000	quantity to move
tipoUc	"UN"	Measure in Unit

GOODS RETURN RESOURCES

THIS API FEATURE IS USED TO SEND GOODS RETURN TO LOGISTICS.

Steps:

- 1 - Unilog Express receives the goods receipt through the API.
- 2 - The Client receives an API response: 201 Created.
- 3 - The WMS Logistics System, processes the information in 2 minutes.



Important

Returning merchandise is the same process as receiving goods.

POST: /documentos/entradas

Headers :

Key	Value	Description
Content-Type	application/json	Content type Json format
Depositante	MYLAN_DISTRIBUIDORA	Headers previously registered

Body :

```
 1 ▼ { "documentos": [ 2 ▼   { 3 ▼     "codigoEstabelecimento": 5, 4      "codigoDepositante": "08241229000391", 5      "tipoDocumento": "NFT", 6      "serieDocumento": "1", 7      "codigoEmpresa": "06142774000214", 8      "codigoTransportadora": "", 9      "descricaoNaturezaOperacao": "SEM VALOR FISCAL", 10     "dataEmissao": "2019-07-24 07:03:00", 11     "dataPrevisaoMovimento": "2019-07-24 07:03:00", 12     "valorTotalDocumento": 2190.63, 13     "valorTotalProduto": 0, 14     "numeroDocumento": "160881", 15     "naturezaOperacao": "1905", 16     "informacaoAdicional1": "Text1", 17     "detalhes": [ 18 ▼       { 19 ▼         "codigoEmpresa": "06142774000214", 20         "codigoProduto": "17554", 21         "tipoUc": "UN", 22         "fatorTipoUc": 1, 23         "classeProduto": "00", 24         "quantidadeMovimento": 40, 25         "valorUnitario": 3.9145, 26         "tipoLogistico": 0, 27         "dadoLogistico": "0", 28         "valorDesconto": 0, 29         "valorAcrescimo": 0, 30         "valorUnitarioFiscal": 0 31       } 32     } 33   ] 34 } 35 } }
```

Line: 17:36 | JSON | Tab Size: 4 | ☰ | ↻ | 🔍 | ⌂

Goods receipt - Boby Description Header : documentos

Field	Value	Description
codigoEstabelecimento	1	Customer Deposit Number in Logistics
codigoDepositante	Number cliente example (999999999999) 14 digits	Registered customer number in Brazil
tipoDocumento	Customer Document Type, example: "NFT"	NFT is Transfer Invoice
serieDocumento	Example : 1 ou 2	Invoice Serial Number
codigoEmpresa	Number example Cnpj (999999999999) 14 digits	number of the end customer registered in Brazil
codigoTransportadora	Number example (999999999999) 14 digits	Registered carrier number in Brazil
descricaoNaturezaOperacao	Fixed "SEM VALOR FISCAL"	Fixed value
dataEmissao	Date	Invoice Date
dataPrevisaoMovimento	Date	Movement Date
valorTotalDocumento	Example : 2190.63	Total amount of Invoice with Taxes
valorTotalProduto	Example : 2090.00	Total amount of invoice without taxes
numeroDocumento	Example : 0002313	Invoice number
naturezaOperacao	Fixes : "1905"	Fixed value
informacaoAdicional1	Note	Additional information

Goods receipt - Body Description Detail : details

Field	Value	Description
codigoEmpresa	Number example (999999999999) 14 digits	CNPJ number of the end customer registered in Brazil
codigoProduto	Product cod	obligatory
tipoUc	"UN"	Measure in Unit
fatorTipoUc	Values = 1	One unit
classeProduto	Example: "00"	Material Status, example: 00 is Free
quantidadeMovimento	Value Item Quantity	Item Quantity
valorUnitario	Unit value of the item	
tipoLogistico	Fixed Value = 3	Indicates that works with batch
dadoLogistico	Bath example : "L1234"	Product batch

Result after sending a POST:

Body : id sequence number
Status code http : 201 Created



The asynchronous result of the processing is the same for all integrations, it can be retrieved with a GET method in the example specified on page 15 and 16.

Body Cookies (1) Headers (10) Test Results Status: 201 Created Time: 2.99s Size: 384 B Save Response ▾

Pretty Raw Preview Visualize Text ▾

```
1 2853991
```

The asynchronous result of the processing is the same for all integrations, it can be retrieved with a GET method in the example specified on page 15 and 16.

GLOSSARY

- 1 - **Client** : ERP system that sends requests and calls through the API.
- 2 - **API** : application program interface , An application program interface (**API**) is a set of routines, protocols, and tools for building software applications. Basically, an **API** specifies how software components should interact.
- 3 - **WMS** - A **warehouse management system** (WMS) is a software application that helps control and manage the day-to-day operations in a warehouse. WMS software guides inventory receiving and put-away, optimizes picking and shipping of orders and advises on inventory replenishment.
- 4 - **POST** : In computing, **POST** is a request **method** supported by HTTP used by the World Wide Web. By design, the **POST** request **method** requests that a web server accepts the data enclosed in the body of the request message, most likely for storing it. It is often used when uploading a file or when submitting a completed web form.
- 5 - **GET** : The **GET method** means retrieve whatever information (in the form of an entity) is identified by the Request-URI.
- 6 - **REST API** : A **RESTful API** is an application program interface (**API**) that uses HTTP requests to GET, PUT, POST and DELETE data. ... **REST** technology is generally preferred over the more robust Simple Object Access Protocol (SOAP) technology because **REST** uses less bandwidth, making it more suitable for efficient internet usage.
- 7 - **JSON** : **JavaScript Object Notation** is an open standard file format, and data interchange format, that uses human-readable text to store and transmit data objects consisting of attribute-value pairs and array data types (or any other serializable value).
- 8 - **Staging** : A **staging environment (stage)** is a nearly exact replica of a production **environment** for software testing. **Staging environments** are made to test codes, builds, and updates to ensure quality under a production-like **environment** before application deployment.