**eMAG & FD Marketplace**

**API Documentation v4.4.9**

**19.09.2025**

| **Version** | **Date modified** | **Changes** |
| --- | --- | --- |
| 4.4.0 | 26.04.2021 | Added “source\_language” key on product\_offer/save for publishing product content in a different language than the platforms default |
| 4.4.1 | 28.04.2021 | Added voucher discounts distribution on order/read |
| 4.4.2 | 16.06.2021 | Added “detailed\_payment\_method” key on order/read  Added Draft publishing API details  Changed product ownership logic |
| 4.4.3 | 08.12.2021 | Added new API for reading commission at offer level  Added “return\_reason” and “observations” keys at product level on rma/read and rma/save  Changed reading\_rma and saving\_rma examples (7.4) |
| 4.4.4 | 09.12.2021 | Added “language” parameter on category/read |
| 4.4.5 | 11.03.2022 | Added new filters on locality/read API  Removed “Matching products” section and API |
| 4.4.6 | 03.10.2023 | Additional info on product\_offer/read and save  Added new “source\_language” values on product\_offer/save  Added “emag\_club” key on product\_offer/save  Added new “validation\_status” values on product\_offer/read: “Documentation update validation pending” and “Update rejected”  Added “genius\_eligibility” key on product\_offer/read  Added “dropoff\_locker” key on awb/save  Added “locker\_delivery\_eligible” key on order/read  Removed the following keys from rma/read and rma/save: “customer\_email”, “id” and ”invoice\_number”  Added "recycle\_warranties" key on order/read  Added new status for offers on product\_offer/read and product\_offer/save: 2 – End of life  Added additional explanations for offers available for sale  Added “tags” key on category/read  Added “tag” key on product\_offer/save (on specific characteristics)  Added new callback: Approved product documentation |
| 4.4.7 | 13.11.2024 | Added MKTP FD RO route  Added “zipcode” key on locality/read  Added “unboxing” and “date” keys on awb/save  Removed “weight” key from product\_offer/read and product\_offer/save  Added "billing\_locality\_id" and "shipping\_locality\_id" keys on order/read  Additional explanation regarding awb/read\_pdf added  Updated key name for locality/read: “country” became “country\_code”  Updated key name for order/read: “cancellation\_reason” became “reason\_cancellation”  Added new order cancellation reasons  Added new keys for vouchers: “vat” and “offered\_by”  Added new keys on order/read: “courier\_external\_office\_id”, “enforced\_vendor\_courier\_accounts”, “modified”  Added new endpoint, order/{orderId}/unlock-courier  Added new “genius\_eligibility\_type” key on product\_offer/read  Added explanations for missing values for “validation\_status” and “translation\_validation\_status”  Removed value = 7 (Rejected Association) for the above mentioned keys  Removed “original\_sale\_price” and “post\_campaign\_original\_sale\_price” keys from campaign\_proposals/save  Added “voucher\_discount” and “not\_available\_post\_campaign” keys on campaign\_proposals/save  Added missing values for “courier\_account\_properties”  Added “pickup\_country\_code” key on courier\_accounts/read  Updated wrong “date” filter on rma/read => date\_start and date\_end  New API rate limits  Added information about future new keys for GPSR on product\_offer/save (read): “safety\_information”, “manufacturer”, “eu\_representative”  Added new pagination keys on category/read  Added example requests regarding AWB issuing |
| 4.4.8 | 28.08.2025 | Updated MARKETPLACE\_API\_URL for FD RO  Added MKTP FD BG route  Added new endpoint, /api-3/smart-deals-price-check  Added "images\_overwrite" key on product\_offer/save  Added “green\_tax” key on product\_offer/save. The field is available only for eMAG RO platform.  Added “part\_number” and “part\_number\_key” filters on product\_offer/read  Updated key behaviour for “manufacturer” and “eu\_representative" on product\_offer/read  Added new value for ”validation\_status” and “translation\_validation\_status” on product\_offer/read  Added “errors” key on validation\_status” and “translation\_validation\_status” keys on product\_offer/read  Added “iso2” key on locality/read  Added “voucher\_name” on order/read -> products -> product\_voucher\_split level  Added “serial\_numbers” and “ext\_part\_number” on order/read -> products  Added “genius\_computed” key on product\_offer/read  Added “order\_product\_id” as property for order attachments  Added “date\_intervals” key on campaign\_proposals/save  Added “start\_date” key on campaign\_proposals/save  Added “start\_date.date” key on campaign\_proposals/save  Added “start\_date.timezone\_type key on campaign\_proposals/save  Added “start\_date.timezone” key on campaign\_proposals/save  Added “end\_date” key on campaign\_proposals/save  Added “end\_date.date” key on campaign\_proposals/save  Added “end\_date.timezone\_type” key on campaign\_proposals/save  Added “end\_date.timezone” key on campaign\_proposals/save  Added “voucher\_discount” key on campaign\_proposals/save  Added “index” key on campaign\_proposals/save  Added “type” on api-3/awb/read  Added “type” on api-3/rma/save  Changed constraint of “type” on api-3/order/read and api-3/rma/read  Added “type” filter for return requests |
| 4.4.9 | 19.09.2025 | Added new API for search by EAN  Added new API for offer updates  Added new API for addresses  Added new field on awb/save: address\_id  Added new field on attachments/read: order\_type |

Table of Contents

[1. eMAG Marketplace API 6](#_Toc209096043)

[1.1. Conventions 6](#_Toc209096044)

[1.2. Request, resources and actions 7](#_Toc209096045)

[1.3. Pagination and filters 8](#_Toc209096046)

[1.4. Response 8](#_Toc209096047)

[1.5. Rate limiting 9](#_Toc209096048)

[1.6. Callback URLs 10](#_Toc209096049)

[2. Publishing products and offers 10](#_Toc209096050)

[2.1. Reading categories, characteristics and family\_types 11](#_Toc209096052)

[2.2. Reading VAT rates 14](#_Toc209096053)

[2.3. Reading Handling Time values 14](#_Toc209096054)

[2.4. Sending a new product 14](#_Toc209096055)

[2.4.1. Draft product 14](#_Toc209096056)

[2.4.2. Product 15](#_Toc209096057)

[2.5. Example for a new product 22](#_Toc209096058)

[2.6. Updating existing offer 23](#_Toc209096059)

[2.6.1. Updating an existing offer using product\_offer/save 23](#_Toc209096060)

[2.6.2. Updating an existing offer using new light offer API offer/save 23](#_Toc209096061)

[2.7. Saving volume measurements on products 23](#_Toc209096062)

[2.8. Reading and counting products and offers 24](#_Toc209096063)

[2.9. Product validation responses 32](#_Toc209096064)

[2.10. Matching products by EAN 32](#_Toc209096065)

[2.11. Attaching offers on existing products 33](#_Toc209096066)

[2.12. Reading commission for an offer 37](#_Toc209096067)

[3. Updating stock 37](#_Toc209096068)

[4. Proposing offers in campaigns 38](#_Toc209096069)

[4.1. Smart Deals Badge 42](#_Toc209096070)

[5. Processing orders 42](#_Toc209096071)

[5.1. Order fields 42](#_Toc209096075)

[5.1.1. Product field in order details 47](#_Toc209096076)

[5.1.2. Customer fields in order details 50](#_Toc209096077)

[5.1.3. Order invoices and warranties 52](#_Toc209096078)

[5.2. Order notification, acknowledgment and order filters 53](#_Toc209096079)

[5.3. Order status matrix 53](#_Toc209096080)

[5.4. Order filters 54](#_Toc209096081)

[5.5. Updating orders 55](#_Toc209096082)

[5.5.1. Removing products from an order 55](#_Toc209096083)

[5.5.2. Adding products to an existng order 56](#_Toc209096084)

[5.5.3. Returned products and storno route 56](#_Toc209096085)

[6. Shipping orders 59](#_Toc209096086)

[6.1. Saving AWB 59](#_Toc209096088)

[6.2. Reading AWB 61](#_Toc209096089)

[6.3. Reading AWB PDF files 62](#_Toc209096090)

[6.4. Reading AWB ZPL type 63](#_Toc209096091)

[6.5. Counting Localities 64](#_Toc209096092)

[6.6. Reading Localities 64](#_Toc209096093)

[6.7. Reading courier accounts 65](#_Toc209096094)

[6.8. Reading addresses 66](#_Toc209096095)

[7. Processing return requests 67](#_Toc209096096)

[7.1. Return requests filters 71](#_Toc209096098)

[7.2. Status change permissions 71](#_Toc209096099)

[7.3. Return request deliveries 72](#_Toc209096100)

[7.4. Examples requests and responses 72](#_Toc209096101)

[8. Invoice API 72](#_Toc209096102)

[8.1. Reading invoice categories 72](#_Toc209096103)

[8.2. Reading invoice data 73](#_Toc209096104)

[8.3. Reading customer invoice data 75](#_Toc209096105)

# eMAG Marketplace API

eMAG Marketplace API is developed by eMAG for Marketplace partners in order to allow them to use their own CRM’s / ERP’s. This document explains the methods for calling the API.

The API can be used in order to:

* send products and offers
* process orders

## Conventions

We define MARKETPLACE\_API\_URL constant of being the API URL of the platform (ex: <https://marketplace-api.emag.ro/api-3>)

We define MARKETPLACE\_URL constant of being the URL of the platform (ex: [https://marketplace.emag.ro](https://marketplace.emag.ro/))

We define DEFAULT\_CURRENCY constant of being the default currency of the platform (ex: RON).

All API parameters are key-sensitive.

|  |  |  |
| --- | --- | --- |
| **Platform** | **eMAG Romania** | **eMAG Bulgaria** |
| **MARKETPLACE\_URL** | [https://marketplace.emag.ro](https://marketplace.emag.ro/) | [https://marketplace.emag.bg](https://marketplace.emag.bg/) |
| **MARKETPLACE\_API\_URL** | <https://marketplace-api.emag.ro/api-3> | <https://marketplace-api.emag.bg/api-3> |
| **Protocol** | HTTPS | HTTPS |
| **Locale** | ro\_RO | bg\_BG |
| **Default currency** | RON | BGN |

|  |  |  |
| --- | --- | --- |
| **Platform** | **eMAG Hungary** | **FD Romania** |
| **MARKETPLACE\_URL** | [https://marketplace.emag.](https://marketplace.emag.hu/)hu | https://marketplace-ro.fashiondays.com |
| **MARKETPLACE\_API\_URL** | <https://marketplace-api.emag.hu/api-3> | https://marketplace-ro-api.fashiondays.com/api-3 |
| **Protocol** | HTTPS | HTTPS |
| **Locale** | hu\_HU | ro\_RO |
| **Default currency** | HUF | RON |

|  |  |  |
| --- | --- | --- |
| **Platform** | **FD Bulgaria** |  |
| **MARKETPLACE\_URL** | https://marketplace-bg.fashiondays.com |  |
| **MARKETPLACE\_API\_URL** | https://marketplace-bg-api.fashiondays.com/api-3 |  |
| **Protocol** | HTTPS |  |
| **Locale** | bg\_BG |  |
| **Default currency** | BGN |  |

To access the API, simply make a Basic Authorization request with your username, password and a base64 computed hash. Please note that user should be granted API rights in order to access the API.

|  |
| --- |
| $hash = base64\_encode($username . ':' . $password); |

## Request, resources and actions

A Marketplace API call is represented by sending a request to API URL of platform. Every request consists of a POST to an URL like:

MARKETPLACE\_API\_URL/**resource/action**

Ex: <https://marketplace-api.emag.ro/api-3/product_offer/save>

| **RESOURCES AND AVAILABLE ACTIONS** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Resource** | **Resource URL** | **Available actions** | | | | |
| **product\_offer** | MARKETPLACE\_API\_URL/product\_offer | read | save | count |  |  |
| **measurements** | MARKETPLACE\_API\_URL/measurements |  | save |  |  |  |
| **offer\_stock** | MARKETPLACE\_API\_URL/offer\_stock/{resourceId} |  |  |  |  |  |
| **campaign\_proposals** | MARKETPLACE\_API\_URL/campaign\_proposals |  | save |  |  |  |
| **order** | MARKETPLACE\_API\_URL/api-3/order | read | save | count | acknowledge | unlock-courier |
| **order/attachments** | MARKETPLACE\_API\_URL/order/attachments |  | save |  |  |  |
| **message** | MARKETPLACE\_API\_URL/message | read | save | count |  |  |
| **category** | MARKETPLACE\_API\_URL/category | read |  | count |  |  |
| **vat** | MARKETPLACE\_API\_URL/vat | read |  |  |  |  |
| **handling\_time** | MARKETPLACE\_API\_URL/handling\_time | read |  |  |  |  |
| **locality** | MARKETPLACE\_API\_URL/locality | read |  | count |  |  |
| **courier\_accounts** | MARKETPLACE\_API\_URL/courier\_accounts | read |  |  |  |  |
| **addresses** | MARKETPLACE\_API\_URL/addresses | read |  |  |  |  |
| **awb** | MARKETPLACE\_API\_URL/awb | read | save |  |  |  |
| **rma** | MARKETPLACE\_API\_URL/rma | read | save | count |  |  |
| **invoice/categories** | MARKETPLACE\_API\_URL/api-3/invoice/categories | read |  |  |  |  |
| **invoice** | MARKETPLACE\_API\_URL/api-3/invoice | read |  |  |  |  |
| **customer-invoice** | MARKETPLACE\_API\_URL/api-3/customer-invoice | read |  |  |  |  |
| **smart-deals-price-check** | MARKETPLACE\_API\_URL/api-3/smart-deals-price-check | read |  |  |  |  |

Below a code example using the resource "category" and the action "read"

|  |  |  |
| --- | --- | --- |
| **Resource** | **Example** | **Context** |
| **category/read** |  | http method: POST |

The API needs authorization and has an IP level filtering. Before testing, sellers should provide a list of whitelisted IP’s. eMAG Marketplace will only allow API calls only from those IP’s.

The POST data consist of 1 mandatory key:

|  |  |
| --- | --- |
| **REQUEST** | |
| **Key** | Description |
| **data** | Data to be passed to the API. The following document will describe keys. |

## Pagination and filters

In order to limit the number of items returned, read actions accept pagination by passing to POST data following parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| **PAGINATION** | | | |
| **Key** | **Description** | **Default value** | **Example** |
| **currentPage** | Set current page displayed | 1 | currentPage =3 |
| **itemsPerPage** | Set number of items to be displayed in one page. Maximum is set to 100. | 100 | itemsPerPage=50 |

Also, filters can be included in POST to refine result set. Filters vary depending on the resource called and are exampled on every resource section.

## Response

When an API call is made, the server MUST reply with a response. The response will ALWAYS be JSON formatted and the header 'Content-type: application/json' will always be passed.

|  |  |
| --- | --- |
| **RESPONSE** | |
| **Key** | **Description** |
| **isError** | Boolean value representing response status. |
| **messages** | Messages included in the response, like error messages, etc. |
| **results** | Results included in the response, mostly when reading resources. |

**IMPORTANT:** Every API request must have a response and the response must contain the key “isError” and its value must be “false”. For each call that does not have the key and “false” value, we recommend setting up alerts, as the call most likely was not interpreted. We also recommend logging all calls and the corresponding API response for a 30 days period.  
Every request must have at most 4000 elements. If the call surpasses this limit the call will have a response with key “isError:true” and “message: Maximum input vars of 4000 exceeded”.

In the event of an documentation error when saving a product, the API will return and “isError:true” message, but the new offer will be saved and processed.

## Rate limiting

All resources described at the table from paragraph 1.2 have the following limits:

* Maximum 12 requests per second or 720 requests per minute for all orders routes
* All the other resources different from orders are limited to max. 3 requests per second or 180 requests per minute
  + Please note the limit for all other resources is cumulative, meaning that, for example, a stock update, an AWB creation and a price update will be counted together and compared to the rate limit threshold
* For optimal performance we recommend not scheduling requests at fixed hours. For example use 12:04:42 as a starting point instead of 12:00:00. The following responses are possible:

|  |  |
| --- | --- |
| **Time throttling limit is exceeded** | **Time throttling limit was not reached** |
| HTTP/1.1 429  Date: Wed, 21 Mar 2018 08:22:44 GMT  Content-Type: application/json; charset=utf-8  Transfer-Encoding: chunked  X-RateLimit-Limit-3second: 1  X-RateLimit-Remaining-3second: 0  Server: kong/0.12.1  {"message":"API rate limit exceeded"} | HTTP/1.1 200  Date: Wed, 21 Mar 2018 08:23:52 GMT  Content-Type: application/json  Transfer-Encoding: chunked  X-RateLimit-Limit-3second: 1  X-RateLimit-Remaining-3second: 0  Server: nginx  {"isError":false,"messages":[],"results":[]} |

Invalid requests will also be limited to the same maximum number of requests pe minute, as detailed above.

* For API resources that accept bulk save, the limit is 50 entities per request. For optimal performance we recommend using between 10 and 50 entities per request.

## Callback URLs

Thefollowing callback URLs can be activated from the Marketplace interface:

|  |  |
| --- | --- |
| **Context** | **Description** |
| **New order** | You will be notified for each new order |
| **Order cancellation** | You will be notified whenever an order in canceled |
| **New return & status change** | You will be notified for each new return request and select status changes (New, Acknowledged & Received) |
| **AWB status change** | You will be notified for each AWB status change |
| **Approved documentation** | You will be notified when the product documentation is validated and received a PNK |

# Publishing products and offers

We define a **draft** product as a minimum set of details required to publish a product. The elements are:

* Name
* Brand
* Part number
* Category (optional)
* EAN (optional)
* Source language (optional)

We define a **product** as a list of elements displayed for a product page. The elements are:

* Name
* Brand
* Part number
* Description
* Images
* Product characteristics (and product families)
* Category
* Barcodes (optional)
* Other attachments (optional)
* EAN (required depending on Category)
* Source language (optional)
* Safety information (optional)

We define an **offer** as a list of elements required for an offer to be available for a product. These elements are:

* Price
* Status
* VAT rate
* Warranty
* Numerical stock
* Handling\_time
* Manufacturer (optional)
* EU representative (optional)

eMAG Marketplace API allows a seller to:

* Send new products and offers
* Send new offers for existing eMAG products (sold by eMAG or any other seller)
* Update existing own offers and/or products



## Reading categories, characteristics and family\_types

Every eMAG product has to be included in a certain category. Sellers cannot create new categories or change existing ones. Also, a seller can only post products and offers in its allowed categories list.

Reading categories without parameters will generate a response containing the first 100 categories. The read can be paginated thus obtaining a full list of categories. Only active categories will be returned.

When passing a category id, the API will return the category name and the list of available characteristics and their corresponding IDs, as well as the available product family\_types and their corresponding IDs. Reading categories one by one is important as it is the only way to identify the restrictive characteristics and corresponding allowed values.

You can read the categories and their characteristics through the API.

Additional pagination filters for characteristic values:

* valuesCurrentPage
* ValuesPerPage

Eg: { "id": 15,

"valuesCurrentPage":1,

"valuesPerPage":10

}

The resource is **category** and the available actions are **read** and **count**.

|  | | **CATEGORY – read** | | | |
| --- | --- | --- | --- | --- | --- |
| **Key – level 1** | **Key – level 2** | **Key – level 3** | **Description** | **Type** | **Example** |
| id |  |  | Category eMAG id | Integer | id=604 |
| name |  |  | Category name | String | name=”Music” |
| is\_allowed |  |  | Indicates if the seller can send products and offers in the category. In order to request access to a specific category, you can use the Marketplace interface.  0 = No 1 = Yes. | Integer | is\_allowed=0 |
| parent\_id |  |  | Id of the parent category | Integer | parent = 12 |
| is\_ean\_mandatory |  |  | Indicates if the sending an EAN is mandatory when saving products   * 1 = mandatory * 0 = optional | Integer | is\_ean\_mandatory =1 |
| is\_warranty\_mandatory |  |  | Indicates if adding warranty is mandatory when saving products   * 1 = mandatory * 0 = optional | Integer | is\_warranty\_mandatory =1 |
| characteristics |  |  | All characteristics available in category | List of arrays |  |
|  | id |  | Characteristic eMAG id | Integer | id=38 |
|  | name |  | Characteristic name | String | name=”Audio” |
|  | type\_id |  | Characteristic type. Indicates the type of values that a characteristic can accept.  **Single value characteristics**:   * 1 = Numeric (ex: 20, 1, 30, 40, etc) * 60 = Size (ex. 36 EU, XL INTL, etc) * 20 = Boolean ( Yes, No, N/A)   **Multiple values characteristics**:   * 2 = Numeric + unit of measure ( ex. 30 cm, 45 m, 20 GB, 30 inch, etc) * 11 = Text Fixed (max length 255 chars) (ex. Blue, Green, Laptop, Notebook, Copil, Man, Woman) * 30 = Resolution ( Width x Height ) (ex. 100 x 200, 640 x 480) * 40 = Volume (Width x Height x Depth - Depth 2) (ex. 30 x 40 x 50 - 10) | Integer | type\_id=11 |
|  | display\_order |  | Characteristic display order | Integer | display\_order=6 |
|  | is\_mandatory |  | Indicates if the characteristic is mandatory when sending a product. Possible values are 0 (the characteristic is not mandatory) and 1 (the characteristic mandatory). | Integer | is\_mandatory=0 |
|  | is\_filter |  | Indicates if the characteristic represents a filter in the website. Possible values are 0 (the characteristic is not filter) and 1 (the characteristic is a filter). | Integer | is\_filter=1 |
|  | allow\_new\_value |  | Indicates if the current characteristic allows you to submit new values that are automatically validated. | Integer | allow\_new\_value=0 |
|  | values |  | List the first 256 existing values of the current characteristic. Important: This key is available only when reading a single category. | Array | 0 => 'Value 1'  1 => 'Value 2' |
|  | tags |  | List of all characteristic tags  Important: In the following months, characteristic tags will be mandatory.  Characteristic "Converted Size" (id 10770) will be removed and its value should be sent on characteristic "Size" (id 6506) - "converted" tag:   1. {"id":6506,    * "tag":"original",    * "value":"36 EU"}, 2. {"id":6506,    * "tag":"converted",    * "value":"39 intl"} | Array | "tags": [    "original",    "converted" ] |
| family\_types |  |  | List of all family types available in category | List of arrays |  |
|  | id |  | Family type id | Integer | Id=95 |
|  | name |  | Family name | String | name=”Quantity” |
|  | characteristics |  | All characteristics of current family type | List of arrays |  |
|  |  | characteristic\_id | Characteristic Id | Integer | characteristic\_id=44 |
|  |  | characteristic\_family\_type\_id | Can only have 3 values, each corresponding to a different display method: “1" ="Thumbnails";  "2"="Combobox";  "3"="Graphic Selection" | Integer | characteristic\_family\_type\_id=2 |
|  |  | is\_foldable | A foldable characteristic wraps all family members (with different characteristic values) as one item in the eMAG category listing | Integer | Is\_foldable=1 |
|  |  | display\_order | Characteristic display order | Integer |  |

By default responses will contain category names in the platform language but you can also send the language as a parameter, in case you want to receive category names in a specifc language. Available languages are: EN, RO, HU, BG, PL, GR and DE

E.g. <https://marketplace-api.emag.ro/api-3/category/read?language=en>

## Reading VAT rates

When sending an offer, you have to send the VAT rate id by sending us a valid VAT id.

The resource is ***vat*** and the action is ***read***. The API will return the list of available VAT rates and their corresponding id’s.

## Reading Handling Time values

When sending an offer, you have to send the Handling Time by sending us a valid value.

The resource is **handling\_time** and the action is **read**. The API will return the list of available handling\_time values.

## Sending a new product

### Draft product

Sending a draft product requires you to send a smaller set of data for publishing a new product. The information is saved in eMAG platform and the necessary details for publishing a product can be added at any time.

*Please note:*

Draft products won’t be sent to eMAG Catalogue team for validation unless you send the other details necessary for publishing a new product (described below).

If an EAN published on a draft product is found in the eMAG catalogue you can skip the product publishing process and attach the offer directly to existing product.

Details for sending a new draft can be found [here](https://marketplace.emag.ro/documentation/api/external).

### Product

Sending a product for the first time requires you to send the entire product documentation and all the offer data. Please note that creating new products implies human validation, so a new product will not be displayed in eMAG platform immediately.

The products that are not compliant with eMAG Documentation Standard will not pass the human validation; in this case you will be notified by our support team. The eMAG Documentation Standard that is available upon request for each category, and it contains the best practices for documenting a product.

Important: Content updates will be allowed only on products on which you are eligible (ownership = 1). For products on which you are not eligible (ownership = 2), any content updates will be rejected.

In order to send a new product, the resource is product\_offer and the available action is save.

| **PRODUCT OFFER – save and create/update product** | | | | |
| --- | --- | --- | --- | --- |
| **Key – level 1** | **Key – level 2** | **Description** | **Constraints** | **Example** |
| id |  | Seller internal product id. This is the primary key for identifying a product offer. | Required. Integer value between 1 and 16777215. | id=243409 |
| category\_id |  | Product category eMAG id. | Required. Integer between 1 and 65535. | category \_id=506 |
| vendor\_category\_id |  | Seller internal category id. | Integer. Optional. | vendor\_category\_id=506 |
| part\_number\_key |  | eMAG part\_number\_key. Used for attaching a product offer to an existing product in eMAG platform. If you want to create new product, don’t set this key. | Optional. String. Will be validated. | part\_number\_key=ES0NKBBBM |
| source\_language |  | The language of the product content input.  If it differs from the platform local language, then the product will enter a translation process.  Available values for this key are: en\_GB, ro\_RO, pl\_PL, bg\_BG, hu\_HU, de\_DE, it\_IT, fr\_FR, es\_ES, nl\_NL, cs\_CZ, ru\_RU, el\_GR, lt\_LT, sk\_SK, uk\_UA | Optional. String.  Default value:  on Marketplace RO: “ro\_RO”  on Marketplace BG: “bg\_BG”  on Marketplace HU: “hu\_HU” | source\_language=”de\_DE” |
| name |  | Product name. Should be consistent with eMAG Product Documentation Standard. | Required. String between 1 and 255 characters long. | name=”Test product” |
| part\_number |  | Manufacturer unique identifier of the product. | Required. String between 1 and 25 characters.  Important:  The following characters will be automatically stripped:   * spaces [ ] * comma [ , ] * semicolon [ ; ]   **Example**: “part number;” will be saved “partnumber” | part\_number=”md788hc/a” |
| description |  | Product description. Should be consistent with eMAG Product Documentation Standard. | Optional. String between 1 and 16777215 characters. Can contain basic HTML tags. | description=”test” |
| brand |  | Brand name. Should be consistent with eMAG Product Documentation Standard. | Required. String between 1 and 255 characters. | brand=”Brand test” |
| force\_images\_download |  | Product images redownload flag. Use it only for updating product images without image link change.  0 - images will be downloaded only if the image link has changed  1 - force image download | Optional. Integer value, 1 or 0. Default = 0 | force\_images\_download=1 |
| images |  | Product images data array. | Optional. List of arrays. |  |
|  | display\_type | Image display type.  1 – main image  2 – secondary image  0 – other images | Optional. Default value 0. Integer value between 0 and 2. | display\_type=1 |
|  | url | Seller image URL. Should be consistent with eMAG Product Documentation Standard. Max 6000px x 6000px and 8 Mb in size. | Required. String between 1 and 1024 characters. Valid URL. JPG, JPEG or PNG file type. | url=”http://valid-url.jpg” |
| images\_overwrite |  | Used to append or overwrite the new images to the existing ones.  If the key is not provided, by default will:   * Append the images – if the seller is not documentation owner * Overwrite the images – if the seller is documentation owner | Optional. Integer value with values: 0, 1.  1 = overwrite  0 = append | images\_overwrite:1 |
| characteristics |  | Characteristic data. Note that characteristics have to be category valid (be part of category template). Should be consistent with eMAG Product Documentation Standard. | Optional. List of arrays. |  |
|  | id | Characteristic eMAG id. | Required. Integer value between 1 and 65535 | id=24 |
|  | value | Characteristic value. Should be consistent with eMAG Product Documentation Standard. | Required. String between 1 and 255 characters | value=”test” |
|  | tag | Characteristic tag value. Should be consistent with eMAG Product Documentation Standard. | Required only for characteristics which have tags on them. String between 1 and 255 characters.  If a characteristic has tags on the category read route, you need to send the same characteristic multiple times with different value for each tag.   1. {"id":6506,    * "tag":"original",    * "value":"36 EU"}, 2. {"id":6506,    * "tag":"converted",    * "value":"39 intl"} | tag=”converted” |
| family |  | Family array. Used to create a new family, add a product to an existing family, or removing a product from a family. | Optional. Array. |  |
|  | id | The unique integer identifier of the family in your platform. If set to 0 (id=0), the product will be removed from its current family. | Required. Integer | Id=0 |
|  | name | Required. Seller Family name. | Required if family id is not equal to 0; | name="Test product" |
|  | family\_type\_id | Required. eMAG Family type id that can be acquired by API (the resource is **category** and the action is **read**). | Required if family id is not equal to 0. Integer. | family\_type\_id=95 |
| url |  | Product URL on the seller website. | Optional. String between 1 and 1024 characters. | url=”http://valid-url.html” |
| warranty |  | The warranty offered in months. | Required/Optional based on category.  Default value:  - 0 (no warranty) if optional  - No default if required.  Integer between 0 and 255. | warranty=24 |
| ean |  | Product barcode identifier (EAN -8, EAN-13, UPC-A, UPC-E, JAN, ISBN-10, ISBN-13, ISSN, ISMN-10, ISMN-13, GTIN-14). Please use the supplier barcode, not your internal barcodes. | Required/Optional based on category. No default value. Array of strings between 6 and 14 characters long. Only numeric figures allowed. | ean=Array('ean1', 'ean2') |
| attachments |  | Product attachments data. Max 10 Mb in size. | Optional. List of arrays. |  |
|  | id | Seller attachment internal id. | Optional. Integer value between 1 and 4294967295. | id=123 |
|  | url | Seller attachment URL. | Required. String between 1 and 1024 characters. Valid URL to document. | url=”http://valid-url” |
| status |  | Seller offer status.  2 – End of life  1 – status active  0 – status inactive | Required. Integer value, 0, 1, 2.  Obs. status=2 should be used only for update | status=1 |
| sale\_price |  | Seller offer sale price without VAT | Required. Decimal value greater than 0. Up to four decimals. | sale\_price=51.6477 |
| recommended\_price |  | Seller offer recommended retail price before discount, without VAT. If set, the offer will be displayed as promo. | Optional. Decimal value greater than 0. Up to four decimals. Must be greater than sale\_price. | recommended\_price=51.6477 |
| min\_sale\_price |  | Seller’s min offer sale price without VAT | Required on first product save. Decimal value greater than 0. Up to four decimals. | min\_sale\_price=40.6477 |
| max\_sale\_price |  | Seller’s max offer sale price without VAT | Required on first product save. Decimal value greater than 0. Up to four decimals. Must be greater than min\_sale\_price. | max\_sale\_price=60.6477 |
| currency\_type |  | Offer currency. Only send the key if it is different from the local Marketplace currency. Available options: EUR or PLN | Optional. 3 characters string. | currency\_type=”EUR” |
| stock |  | Offer available quantity array. | Required. List of arrays. | {  0=>{  warehouse\_id=1,  value=20}} |
|  | warehouse\_id | The id of the warehouse. Use warehouse\_id=1 for only one warehouse. | Required inside stock array. Integer. | warehouse\_id=1 |
|  | value | Offer available quantity. | Required inside stock array. Integer between 0 and 65535. | value=20 |
| handling\_time |  | Handling time array. If no array is sent, the products are shipped the same day they are received. | Optional. List of arrays. | {  0=>{  warehouse\_id=1,  value=1}} |
|  | warehouse\_id | The id of the warehouse. Use warehouse\_id=1 for only one warehouse. | Required inside handling\_time array. Integer. | warehouse\_id=1 |
|  | value | Handling time, in number of days counted from the day the order was received. If handling\_time = 0 the order will be shipped the same day it is received. | Required inside handling\_time array. Integer value between 0 and 255. Default value = 0. | value=0 |
| supply\_lead\_time |  | The number of days needed to restock the product, from order placement to the supplier or production order, until product reception in the warehouse.  Available values for this key are: 2, 3, 5, 7, 14, 30, 60, 90, 120 | Optional. Integer.  Default value = 14 | supply\_lead\_time=5 |
| start\_date |  | If it's a new offer, it represents the date your offer will be available from. For offer updates, it schedules value updates for the following data:   * sale\_price * recommended\_price * stock * handling\_time * vat\_id * warranty * status   All other data will be updated on the fly. Using start\_date, for example, you can schedule the inactivation of an offer, a price update, etc. | Optional. Text in YYYY-MM-DD format. Date can be as far as 60 days in the future (cannot be earlier than tomorrow). Cannot be null. | start\_date=”2014-12-31” |
| vat\_id |  | Seller offer VAT rate id. Use /vat/read to display possible values. | Required. Integer. | Ex: vat \_id=1 |
| emag\_club |  | Indicates if you want your offer to be available or not in the eMAG Genius program. Only eligible offers will display Genius benefits to customers eMAG website even if you have chosen them to be available. | Optional. Integer value, 1 or 0. Default = 1 | emag\_club=1 |
| safety\_information |  | GPSR safety information. This will include any warnings contained in the product or any existing warnings that accompany the product. | Optional. String between 1 and 16777215 characters. | safety\_information = “Keep out of reach of children. Danger of suffocation.” |
| manufacturer |  | GPSR manufacturer information. Manufacturer information contains name, address and email address.  We allow a maximum of 10 sets of information. The update on these fields will overwrite the existing information. | Optional. List of arrays.  This set of information is optional, but if it is provided, all corresponding fields (name, address, email) must be completed. | "manufacturer": [{  "name": "Company name Ltd.",  "address": "Company, city, street, number, country",  "email": "company@company.com"  }] |
|  | name | GPSR manufacturer information. The name of the manufacturer. | Required. String between 1 and 200 characters. |  |
|  | address | GPSR manufacturer information. The address of the manufacturer. | Required. String between 1 and 500 characters. |  |
|  | email | GPSR manufacturer information. The email address of the manufacturer. | Required. String between 1 and 100 characters. |  |
| eu\_representative |  | GPSR EU representative information. Represents the EU based economic operator ensuring the product complies with the required regulations, in case the manufacturer is not based in the EU.  If the manufacturer is based in EU, you can submit the manufacturer details.  The information provided will include the name, address and e-mail address.  We allow a maximum of 10 sets of information. The update on these fields will overwrite the existing information. | Optional. List of arrays.  This set of information is optional, but if it is provided, all corresponding fields (name, address, email) must be completed. | "eu\_representative": [{  "name": "Company name Ltd.",  "address": "Company, city, street, number, country",  "email": “company@company.com”  }] |
|  | name | GPSR EU representative information. The name of the EU based economic operator. | Required. String between 1 and 200 characters. |  |
|  | address | GPSR EU representative information. The address of the EU based economic operator. | Required. String between 1 and 500 characters. |  |
|  | email | GPSR EU representative information. The email address of the EU based economic operator. | Required. String between 1 and 100 characters. |  |
| green\_tax |  | The value for Green tax that you want to set on offer and that will be displayed in site. This value includes VAT.  The value can be set only for eMAG RO platform. | Optional. Decimal value >= 0. Up to four decimals. | green\_tax=10 |

**IMPORTANT:**

* During campaings with stock in site we will not allow the following:
  + regular offer updates;
  + updates sent during campaign time that have a start\_date in the future;
  + previously scheduled updates with start\_date during campaign time.
* You can save an offer update with currency\_type different from the local Marketplace currency and start\_date before the end of the current month. Any update attempt with a start\_date greater than the last day of the current month will be rejected.
* Be aware that prices published directly in the local Marketplace currencies will be overwritten by prices published in other currencies when these are recalculated and published automatically on FX change at the beginning of every month. The automated recalculation can be disabled on request.
* Min / Max sale price keys are used for price check purposes and are mandatory for all calls used to create product/offers for the first time. As a best practice we recommend sending these keys only when you want to change their values.
* Sale price will be validated against min\_sale\_price and max\_sale\_price. Any offer that is not within the specified range will be rejected.
* In order to change a previously sent product image or attachment the url should be different from the one already sent. We reload the images only if the URL differs.
* We recommend sending the product data only upon product create/update, as there is no need to resend product unless it changed. Also we recommend sending the offer data upon changing (no matter the frequency) and at least weekly (even if the offer is the same) rather using periodical sending (crons, agents). You should program marketing campaigns using “start\_date” campaign. Also please offer the possibility for an offer to be attached to an existing eMAG product (using part\_number\_key).
* The product part number should be assigned to a single product.

If a product part number is re-used (set on a second product) an error will be generated and the product will NOT be saved.

* Multiple EAN codes can be set on an offer, but one EAN product code can NOT be used on two or more products.

If one EAN code is added on a second new product, an error will be generated and the product will NOT be saved.

If one EAN code used on a new product is already linked to a product in the eMAG catalogue, the offer will be automatically associated to that existing product.

* In the event of an documentation error when saving a product, the API will return and “isError:true” message, but the new offer will be saved and processed.
* When adding a product to a family
  + The category id of a product and the category id of its family type (family\_type\_id) must be the same.
  + All characteristics that define a family must be present and must have a valid value
  + All characteristics that define a family must have a single value
  + If a family is not valid, you will receive a warning response, but the product will be saved/updated
  + When moving a product from one family to another you only have to send the product with its new family type, id and name and make sure you follow the same rules as above

## Example for a new product

|  |  |  |
| --- | --- | --- |
| **Resource** | **Example** | **Context** |
| **product\_offer/save** |  | http method: POST |

## Updating existing offer

### Updating an existing offer using product\_offer/save

When updating an existing offer for a product, you should send only the offer, without the documentation. Mandatory when updating a product offer are the following keys:

* id
* status
* sale\_price
* vat\_id
* handling\_time
* stock

Please note that although the API permits sending the entire documentation on each offer update (price change, out-of-stock change, etc.) we do not recommend or encourage such a practice.

If you need to deactivate a valid offer on the website, you should send the offer with the “status = 0”.

### Updating an existing offer using new light offer API offer/save

This endpoint allows you to update existing offers on the eMAG platform by changing only offer-specific data and cannot be used to create new offers or modify product information.

**API Details:**

**Mandatory key**: id

**Optional keys**:

sale\_price

recommended\_price

min\_sale\_price

max\_sale\_price

currency\_type

stock

handling\_time

vat\_id

status

Note that only the fields you want to update need to be included in the request.

|  |  |  |
| --- | --- | --- |
| **Resource** | **Example** | **Context** |
| **offer/save** |  | http method: POST |

## Saving volume measurements on products

In order to save volume measurements on existing products, the resource is ***measurements*** and the available action is ***save***. The measurement units for volume are millimeters and grams.

| **MEASUREMENTS – save** | | | |
| --- | --- | --- | --- |
| **Key – level 1** | **Description** | **Constraints** | **Example** |
| id | Seller internal product id. This is the primary key for identifying a product offer. | Required. Integer value between 1 and 16777215. | id=243409 |
| length | The length of the product in millimeters (mm) | Required. Decimal value between 0 and 999999. Up to two decimals. | length=100 |
| width | The width of the product in millimeters (mm) | Required. Decimal value between 0 and 999999. Up to two decimals. | width=122.50 |
| height | The height of the product in millimeters (mm) | Required. Decimal value between 0 and 999999. Up to two decimals. | height=250 |
| weight | The weight of the product in grams (g) | Required. Decimal value between 0 and 999999. Up to two decimals. | weight=1254.50 |

|  |  |  |
| --- | --- | --- |
| **Resource** | **Example** | **Context** |
| ***measurements*/save** |  | http method: POST |

## Reading and counting products and offers

In order to check the existing products (offers) and their status, the resource is ***product\_offer*** and the available action are ***read*** and ***count***.

Important: When reading products, only your own content input on the product will be returned.

|  | | **PRODUCT\_OFFER – read** | | | |
| --- | --- | --- | --- | --- | --- |
| **Key – level 1** | **Key – level 2** | **Description** | **Type** | **Example** | |
| part\_number\_key |  | eMAG part\_number\_key. | String | part\_number\_key=ES0NKBBBM | |
| number\_of\_offers |  | How many sellers have active offers on this product | Integer | number\_of\_offers=3 | |
| buy\_button\_rank |  | The rank of the offer in its race to win the <Add to cart> button | Integer | buy\_button\_rank=1 | |
| best\_offer\_sale\_price |  | Best selling price available in eMAG for the same Product | Decimal | best\_offer\_sale\_price=51.6477 | |
| best\_offer\_recommended\_price |  | The corresponding recommended price for the offer holding the best selling price | Decimal | best\_offer\_recommended\_price=54.6477 | |
| ownership |  | Indicates who has ownership on the product’s documentation. Posible values:  1 – Eligible for content updates  2 – Not eligible for content updates | Integer | ownership=1 | |
| category\_id |  | Product category eMAG id. | Integer | category \_id=506 | |
| vendor\_category\_id |  | Seller internal category id. | Integer | vendor\_category\_id=506 | |
| id |  | Seller internal product id. This is the primary key for identifying a product offer. | Integer | id=243409 | |
| brand |  | Product brand name. | String | brand=”Brand test” | |
| name |  | Product name. | String | name=”Test product” | |
| part\_number |  | Product part number. | String | part\_number=”md788hc/a” | |
| sale\_price |  | Seller offer sale price without VAT | Decimal | sale\_price=51.6477 | |
| recommended\_price |  | Seller offer recommended retail price before discount, without VAT. | Decimal | recommended\_price=54.6477 | |
| currency |  | Product price currency. | String | currency='RON' | |
| description |  | Product description. | String | description=”test” | |
| url |  | Product URL on the seller website. | String | url=”http://valid-url.html” | |
| warranty |  | The warranty offered in months. | Integer | warranty=24 | |
| ean |  | Product barcode identifier (EAN -8, EAN-13, UPC-A, UPC-E, JAN, ISBN-10, ISBN-13, ISSN, ISMN-10, ISMN-13, GTIN-14). | Array | ean=Array('ean1', 'ean2') |
| general\_stock |  | The sum of the stock on all seller warehouses. Is decremented and incremented when orders are processed. | Integer | general\_stock=20 | |
| estimated\_stock |  | This key takes into account the reserved stock on unacknowledged orders. | Integer | estimated\_stock=20 | |
| status |  | Seller offer status.  2 – end of life  1 – status active  0 – status inactive | Integer | status=1 | |
| images |  |  | List of arrays |  | |
|  | url | Seller image URL. | String | url=”http://valid-url.jpg” | |
|  | display\_type | Image display type.  1 – main image  2 – secondary image  0 – other images | Integer | display\_type=1 | |
| characteristics |  | All characteristics available in category | List of arrays |  | |
|  | id | Characteristic eMAG id | Integer | id=38 | |
|  | value | Characteristic value. | String | value=”test” | |
| vat\_id |  | Seller offer VAT rate id. | Integer | vat \_id=1 | |
| family |  | Product family. | Array |  | |
|  | id | Family id. | Integer | id=295 | |
|  | name | Family name. | String | name=”Test family” | |
| handling\_time |  |  | List of arrays |  | |
|  | warehouse\_id | The id of the warehouse. | Integer | warehouse\_id=1 | |
|  | value | Handling time, in number of days counted from the day the order was received. | Integer | value=0 | |
|  |  |  |  |  | |
| validation\_status |  | Product validation status | List of arrays |  | |
|  | value | Product validation status value | Integer | value=4 | |
|  | Description | Product validation status description | String | Description=”Rejected documentation” | |
|  | errors | Feedback received on product documentation. | List of arrays |  | |
| translation\_validation\_status |  | Product validation status with details included for products sent with different source language | List of arrays |  | |
|  | value | Product validation status value | Integer | value=4 | |
|  | Description | Product validation status description | String | Description=”Rejected documentation” | |
|  | errors | Feedback received on product documentation. | List of arrays |  | |
| offer\_validation\_status |  | Offer validation status | List of arrays |  | |
|  | value | Offer validation status value | Integer | value=2 | |
|  | Description | Offer validation status description | String | Description=”Invalid price” | |
| genius\_eligibility |  | Indicates if the offer is eligible for Genius program. | Integer | genius\_eligibility = 1 | |
| genius\_eligibility\_type |  | Indicates the type of Genius for which the offer is eligible:  1 – Genius Full  2 – Genius EasyBox  3 – Genius HD | Integer | genius\_eligibility\_type = 2 | |
| genius\_computed |  | Indicates offer's Genius type for offers **active** in Genius  0 – offer not active in Genius  1 – offer active in Genius Full  2 – offer active in Genius EasyBox  3 – offer active in Genius HD | Integer | genius\_computed = 2 | |
| manufacturer |  | GPSR manufacturer completion flag which indicates if the GPSR info is present on product. This flag is returned when reading products without filtering. If you need to read manufacturer info, you need to filter for a specific product. | Boolean. | manufacturer = true | |
| manufacturer |  | GPSR manufacturer information. Manufacturer information contains name, address and email address. | List of arrays. | "manufacturer": [{  "name": "Company name Ltd.",  "address": "Company, city, street, number, country",  "email": "company@company.com"  }] | |
|  | name | GPSR manufacturer information. The name of the manufacturer. | String. |  | |
|  | address | GPSR manufacturer information. The address of the manufacturer. | String. |  | |
|  | email | GPSR manufacturer information. The email address of the manufacturer. | String. |  | |
| eu\_representative |  | GPSR EU representative completion flag which indicates if the GPSR info is present on product. This flag is returned when reading products without filtering. If you need to read EU representative info, you need to filter for a specific product. | Boolean. | eu\_representative = true | |
| eu\_representative |  | GPSR EU representative information. Represents the EU based economic operator ensuring the product complies with the required regulations, in case the manufacturer is not based in the EU. The information provided will include the name, address and e-mail address. | List of arrays. | "eu\_representative": [{  "name": "Company name Ltd.",  "address": "Company, city, street, number, country",  "email": “company@company.com” }] | |
|  | name | GPSR EU representative information. The name of the EU based economic operator. | String. |  | |
|  | address | GPSR EU representative information. The address of the EU based economic operator. | String. |  | |
|  | email | GPSR EU representative information. The email address of the EU based economic operator. | String. |  | |
| safety\_information |  | GPSR safety information. This will include any warnings contained in the product or any existing warnings that accompany the product. | String. | safety\_information = “Keep out of reach of children. Danger of suffocation.” | |

The following filters are available when counting and reading products and offers:

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| id | Displays the details for the corresponding ext\_id. | Optional. Integer value between 1 and 4294967295. |
| currentPage | Set current page displayed | Optional, integer. Ex: currentPage =3 |
| itemsPerPage | Set number of items to be displayed in one page. Maximum is set to 100. | Optional, integer. itemsPerPage=50 |
| status | Returns only the offers with this status. | Optional. Seller offer status. 1 – status active  0 – status inactive |
| part\_number | Displays the details for the corresponding part\_number. | Optional.  Ex: part\_number = 12334 |
| part\_number\_key | Displays the details for the corresponding part\_number\_key. | Optional.  Ex: part\_number\_key = D9T0KBMBM |
| general\_stock | Returns only offers with numerical general\_stock that have a value between 0 and the input. | Optional general\_stock = 3 |
| estimated\_stock | Returns only offers with numerical estimated\_stock that have a value between 0 and the input. | Optional reserved\_stock = 3 |
| offer\_validation\_status | Returns only the results with this validation status. | Optional.  1 = Saleable  2 = Invalid price |
| validation\_status | Returns only the results with this validation status. | Optional.  1 = Awaiting MKTP validation 2 = Awaiting Brand validation 3 = Awaiting EAN validation 4 = Awaiting Documentation Validation 5 = Rejected Brand 6 = Rejected EAN 8 = Rejected Documentation  9 = Approved Documentation  10 = Blocked  11 = Documentation update validation pending  12 = Update rejected |
| translation\_validation\_status | Returns only the results with this validation status. | Optional.  1 = Awaiting MKTP validation 2 = Awaiting Brand validation 3 = Awaiting EAN validation 4 = Awaiting Documentation Validation 5 = Rejected Brand 6 = Rejected EAN 8 = Rejected Documentation  9 = Approved Documentation  10 = Blocked  11 = Documentation update validation pending  12 = Update rejected  13 = Waiting for salable offer  14 = Unsuccessful translation  15 = Translation in progress  16 = Translation pending  17 = Partial translation |

Reading products will always return the values sent by sellers.

In order to be available for selling, an offer has to meet the following conditions:

* Stock > 0
* Seller status should be active
* All 3 keys (“status”,”offer\_validation\_status”, “validation\_status”) should have values that allows availability, see details below:

| **Key** | **Status** | **Availability** | **Description** |
| --- | --- | --- | --- |
| “status”:0 | Inactive/Auto-inactivated | Not allowed | The status can be set by the seller on the offers temporary unavailable. The status can be also set based on losing access in the category, losing an EAN conflict, etc |
| “status”:1 | Active | Allowed | The status is set by the seller when they want to sell the offer. |
| “status”:2 | EOL/ Auto EOL | Not allowed | The status that can be set by the seller on the offers that does not want to sell them anymore. The status can be also set based on the Auto EOL flow. |
| “offer\_validation\_status”-> ”value”:1 | Valid | Allowed | Feedback based on price check. If the value =1, means that price is valid and the offer can be sell. |
| “offer\_validation\_status”:-> “value”:2 | Invalid price | Not allowed | Feedback based on price check. If the value =2, means that price is not valid and the offer cannot be sell until the price is manually validated. |
| “validation\_status”-> ”value”:0 | Draft | Not allowed | Newly documented product that does not contain all the necessary details to be sent to validation. Check the feedback received on errors key. |
| “validation\_status”:-> “value”:1 | In MKTP validation | Not allowed | This value is applied on new products that needs MKTP validation in order to be sent to eMAG systems. |
| “validation\_status”:-> “value”:2 | Awaiting Brand validation | Not allowed | The product is in validation and is pending for the  brand to be validated by eMAG system. |
| “validation\_status”:-> “value”:3 | Waiting for EAN approval | Allowed | This value signals the presence of an EAN conflict, but does not influence the availability of an offer until it is resolved. |
| “validation\_status”:-> “value”:4 | New documentation validation pending | Not allowed | Newly documented product that has not been finalized/validated. |
| “validation\_status”:-> “value”:5 | Rejected Brand | Not allowed | The product documentation is rejected because of the provided brand. Update the product with correct brand. |
| “validation\_status”:-> “value”:6 | Invalid product – EAN rejected | Not allowed | The product has lost an EAN conflict and the offer has been inactivated (“status”:0) |
| “validation\_status”:-> “value”:8 | Documentation rejected | Not allowed | Newly documented product that does not meet the criteria and requires correction to the documentation. |
| “validation\_status”:-> “value”:9 | Approved documentation | Allowed | The product is documented, approved and can be sold if it has a saleable offer attached. For this status, there may be exceptions in the case of automatically translated products, as they are not published on the site even though they have this status. Please check “translation\_validation\_status” key which allows a greater granularity to identify the translated products. |
| “validation\_status”:-> “value”:10 | Blocked | Not allowed | The offer for these products has been blocked from sale by an admin or an automatic flow, such as Product health or OPC. In order to be unblocked, It requires the validation by an admin. |
| “validation\_status”:-> “value”:11 | Update awaiting approval | Allowed | Documented and approved product that can be sold, but for which the validation of some updates sent by the seller is pending. For this status, there may be exceptions in the case of automatically translated products, as they are not published on the site even though they have this status. Please check “translation\_validation\_status” key which allows a greater granularity to identify the translated products. |
| “validation\_status”:-> “value”:12 | Update rejected | Allowed | Documented and approved product that can be sold, but for which the validation of some updates sent by the seller were rejected. |
| “translation\_validation\_status”-> ”value”:0 | Draft | Not allowed | Newly documented product that does not contain all the necessary details to be sent to validation. Check the feedback received on errors key. |
| “translation\_validation\_status”:-> “value”:1 | In MKTP validation | Not allowed | This value is applied on new products that needs MKTP validation in order to be sent to eMAG systems. |
| “translation\_validation\_status”:-> “value”:2 | Awaiting Brand validation | Not allowed | The product is in validation and is pending for the  brand to be validated by eMAG system. |
| “translation\_validation\_status”:-> “value”:3 | Waiting for EAN approval | Allowed | This value signals the presence of an EAN conflict, but does not influence the availability of an offer until it is resolved. |
| “translation\_validation\_status”:-> “value”:4 | New documentation validation pending | Not allowed | Newly documented product that has not been finalized/validated. |
| “translation\_validation\_status”:-> “value”:5 | Rejected Brand | Not allowed | The product documentation is rejected because of the provided brand. Update the product with correct brand. |
| “translation\_validation\_status”:-> “value”:6 | Invalid product – EAN rejected | Not allowed | The product has lost an EAN conflict and the offer has been inactivated (“status”:0) |
| “translation\_validation\_status”:-> “value”:8 | Documentation rejected | Not allowed | Newly documented product that does not meet the criteria and requires correction to the documentation. |
| “translation\_validation\_status”:-> “value”:9 | Approved documentation | Allowed | The product is documented, approved and can be sold if it has a saleable offer attached. |
| “translation\_validation\_status”:-> “value”:10 | Blocked | Not allowed | The offer for these products has been blocked from sale by an admin or an automatic flow, such as Product health or OPC. In order to be unblocked, It requires the validation by an admin. |
| “translation\_validation\_status”:-> “value”:11 | Update awaiting approval | Allowed | Documented and approved product that can be sold, but for which the validation of some updates sent by the seller is pending. |
| “translation\_validation\_status”:-> “value”:12 | Update rejected | Allowed | Documented and approved product that can be sold, but for which the validation of some updates sent by the seller were rejected. |
| “translation\_validation\_status”:-> “value”:13 | Waiting for salable offer | Not allowed | In order for the translation to be prioritized, make sure that the offer has stock and is active |
| “translation\_validation\_status”:-> “value”:14 | Unsuccessful translation | Not allowed | Translation request for new product that does not meet the criteria and requires correction to the documentation. |
| “translation\_validation\_status”:-> “value”:15 | Translation in progress | Not allowed | Automatic translated product, with PNK, not yet displayed on customer website, which is now in the process of manual translation |
| “translation\_validation\_status”:-> “value”:16 | Translation pending | Not allowed | New product translation request which waits processing |
| “translation\_validation\_status”:-> “value”:17 | Partial translation | Allowed | Automatic translated product with PNK, already published to cutomer website |

## Product validation responses

After reading a product, all the elements previously sent are returned, along with the key *doc\_errors*. The key is not null for products that were rejected due to improper documentation. Below the list of possible errors, when they occur and the possible actions you need to take.

|  |
| --- |
|  |

## Matching products by EAN

To simplify and speed up the offer association process, we’ve launched a new API endpoint. With this enhancement, you can now search directly by EAN codes to quickly check if your products already exist on eMAG. This means faster, more accurate, and easier offer associations.

|  |  |  |  |
| --- | --- | --- | --- |
| **EAN reading - request** | | | |
| **Key – level 1** | **Description** | **Type** | **Example** |
| eans | EANs | Array of strings | eans[]=7086812930967&eans[]=5904862975146 |

|  |  |  |  |
| --- | --- | --- | --- |
| **EAN reading - response** | | | |
| **Key – level 1** | **Description** | **Type** | **Example** |
| eans | European Article Numbers | String | 5904862975146 |
| part\_number\_key | eMAG part\_number\_key. | String | DY74FJYBM |
| product\_name | Product name | String | Tenisi barbati, Sprandi, 590486297490344, Sintetic, 44 EU, Negru |
| brand\_name | Brand name | String | Sprandi |
| category\_name | Category name | String | Men Trainers |
| doc\_category\_id | Category id (doc template id) | Integer | 2735 |
| site\_url**\*** | eMAG product site URL | String | <http://emag.ro/product_details/pd/DY74FJYBM> |
| allow\_to\_add\_offer | Seller has access in category | Boolean | true |
| vendor\_has\_offer | Seller has an offer on this product | Boolean | false |
| hotness | Product performance | String | SUPER COLD |
| product\_image | The main image 150x150 | String | <https://s13emagst.akamaized.net/products/15566/15565152/images/res_50a7b8ffc4ea179f6b63e6d50810e8de.png?width=150&height=150&hash=6A5AE673361E590A60A27E86CED60A5B> |

**\*** *Not included in Fashion Days responses*

**Usage details:**

**EAN limit:** up to 100 codes per request. If more are submitted, only the first 100 will be processed, and a notification message will be returned.

**Rate details:**

5 requests/second

200 requests/minute

5.000 requests/day

Examples of requests and responses

|  |  |  |
| --- | --- | --- |
| **Resource** | **Example** | **Context** |
| **documentation/find\_by\_eans** |  | http method: GET |

## Attaching offers on existing products

The attaching action can be done via product\_offer/save route.

You can choose between using PNK (part\_number\_key) or EAN for attaching offers on existing eMAG products.

If the product already exists in eMAG catalog, just add the key “part\_number\_key” with product’s part\_number\_key or the “ean” key with a single EAN.

**IMPORTANT:**

Only one offer can be attached to an existing product (identified by a “part\_number\_key”) in eMAG catalogue.

In case you try to attach a second offer to a “part\_number\_key” that already has one of your offers attached, an error will be generated and the offer will NOT be saved.

If you already have an offer attached to a “part\_number\_key” please update it instead of trying to attach a new one.

|  |
| --- |
| The part\_number\_key is the last key found in the URL of an eMAG product. It will ALWAYS have both numbers and characters. Ex: for product http://www.emag.ro/telefon-mobil-nokia-105-black-105-black/pd/D5DD9BBBM/ the part\_number\_key is D5DD9BBBM. |

| **PRODUCT OFFER – save and create/update product** | | | | |
| --- | --- | --- | --- | --- |
| **Key – level 1** | **Key – level 2** | **Description** | **Constraints** | **Example** |
| id |  | Seller internal product id. This is the primary key for identifying a product offer. | Required. Integer value between 1 and 16777215. | id=243409 |
| name |  | Product name. Should be consistent with eMAG Product Documentation Standard. | Required. String between 1 and 255 characters long. | name=”Test product” |
| ean |  | Product barcode identifier (EAN -8, EAN-13, UPC-A, UPC-E, JAN, ISBN-10, ISBN-13, ISSN, ISMN-10, ISMN-13, GTIN-14). Please use the supplier barcode, not your internal barcodes. | Required if part\_number\_key is not present. No default value. Array of strings between 6 and 14 characters long. Only numeric figures allowed.  **OBS**: “part\_number\_key” and “ean” keys are mutually exclusive, you should use one or the other. | ean=Array('ean1') |
| part\_number\_key |  | eMAG part\_number\_key. Used for attaching a product offer to an existing product in eMAG platform. If you want to create new product, don’t set this key. | Required if ean is not present. String. Will be validated  **OBS**: “part\_number\_key” and “ean” keys are mutually exclusive, you should use one or the other. | part\_number\_key=ES0NKBBBM |
| status |  | Seller offer status.  1 – status active  0 – status inactive | Required. Integer value, 1 or 0. | status=1 |
| sale\_price |  | Seller offer sale price without VAT | Required. Decimal value greater than 0. Up to four decimals. | sale\_price=51.6477 |
| recommended\_price |  | Seller offer recommended retail price before discount, without VAT. If set, the offer will be displayed as promo. | Optional. Decimal value greater than 0. Up to four decimals. Must be greater than sale\_price. | recommended\_price=51.6477 |
| min\_sale\_price |  | Seller’s min offer sale price without VAT | Required on first product save. Decimal value greater than 0. Up to four decimals. | min\_sale\_price=40.6477 |
| max\_sale\_price |  | Seller’s max offer sale price without VAT | Required on first product save. Decimal value greater than 0. Up to four decimals. Must be greater than min\_sale\_price. | max\_sale\_price=60.6477 |
| currency\_type |  | Offer currency. Only send the key if it is different from the local Marketplace currency. Available options: EUR or PLN | Optional. 3 characters string. | currency\_type=”EUR” |
| vat\_id |  | Seller offer VAT rate id. Use /vat/read to display possible values. | Required. Integer. | Ex: vat \_id=1 |
| stock |  | Offer available quantity array. | Required. List of arrays. | {  0=>{  warehouse\_id=1,  value=20}} |
| stock | warehouse\_id | The id of the warehouse. Use warehouse\_id=1 for only one warehouse. | Required inside stock array. Integer. | warehouse\_id=1 |
| stock | value | Offer available quantity. | Required inside stock array. Integer between 0 and 65535. | value=20 |
| handling\_time |  | Handling time array. If no array is sent, the products are shipped the same day they are received. | Optional. List of arrays. | {  0=>{  warehouse\_id=1,  value=1}} |
| handling\_time | warehouse\_id | The id of the warehouse. Use warehouse\_id=1 for only one warehouse. | Required inside handling\_time array. Integer. | warehouse\_id=1 |
| handling\_time | value | Handling time, in number of days counted from the day the order was received. If handling\_time = 0 the order will be shipped the same day it is received. | Required inside handling\_time array. Integer value between 0 and 255. Default value = 0. | value=0 |
| start\_date |  | If it's a new offer, it represents the date your offer will be available from. For offer updates, it schedules value updates for the following data:   * sale\_price * recommended\_price * stock * handling\_time * vat\_id * warranty * status   All other data will be updated on the fly. Using start\_date, for example, you can schedule the inactivation of an offer, a price update, etc. | Optional. Text in YYYY-MM-DD format. Date can be as far as 60 days in the future (cannot be earlier than tomorrow). Cannot be null. | start\_date=”2014-12-31” |
| warranty |  | The warranty offered in months. | Required/Optional based on category.  Default value:  - 0 (no warranty) if optional  - No default if required.  Integer between 0 and 255. | warranty=24 |
| manufacturer |  | GPSR manufacturer information. Manufacturer information contains name, address and email address.  The information provided will include the name, address and e-mail address.  We allow a maximum of 10 sets of information. The update on these fields will overwrite the existing information. | Optional. List of arrays.  This set of information is optional, but if it is provided, all corresponding fields (name, address, email) must be completed. | "manufacturer": [{  "name": "Company name Ltd.",  "address": "Company, city, street, number, country",  "email": "company@company.com"  }] |
|  | name | GPSR manufacturer information. The name of the manufacturer. | Required. String between 1 and 200 characters. |  |
|  | address | GPSR manufacturer information. The address of the manufacturer. | Required. String between 1 and 500 characters. |  |
|  | email | GPSR manufacturer information. The email address of the manufacturer. | Required. String between 1 and 100 characters. |  |
| eu\_representative |  | GPSR EU representative information. Represents the EU based economic operator ensuring the product complies with the required regulations, in case the manufacturer is not based in the EU. The information provided will include the name, address and e-mail address.  If the manufacturer is based in EU, you can submit the manufacturer details.  The information provided will include the name, address and e-mail address.  We allow a maximum of 10 sets of information. The update on these fields will overwrite the existing information. | Optional. List of arrays.  This set of information is optional, but if it is provided, all corresponding fields (name, address, email) must be completed. | "eu\_representative": [{  "name": "Company name Ltd.",  "address": "Company, city, street, number, country",  "email": “company@company.com”  }] |
|  | name | GPSR EU representative information. The name of the EU based economic operator. | Required. String between 1 and 200 characters. |  |
|  | address | GPSR EU representative information. The address of the EU based economic operator. | Required. String between 1 and 500 characters. |  |
|  | email | GPSR EU representative information. The email address of the EU based economic operator. | Required. String between 1 and 100 characters. |  |

|  |  |  |
| --- | --- | --- |
| **Resource** | **Example** | **Context** |
| **product\_offer/save** |  | http method: POST |

## Reading commission for an offer

In order to read the estimated commission for an offer, a new REST API is available. Please consult the [API Swagger](https://marketplace.emag.ro/documentation/api/external).

# Updating stock

In order to update only the stock of an offer a REST resource is available

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resource** | **Resource URL** | **Method** | **Authorization** | **Headers** | **Parameters** |
| **offer\_stock** | MARKETPLACE\_API\_URL/offer\_stock/{resourceId} | PATCH | Basic (Base64) | Content-Type application/json | resourceId |

The resourceId parameter represents the Seller internal product id. This is the primary key for identifying a product offer.

|  |  |  |
| --- | --- | --- |
| **Resource** | **Example** | **Context** |
| **offer\_stock** |  | http method: PATCH |

# Proposing offers in campaigns

To propose a valid offer in a campaign the resource is **campaign\_proposals** and the available action is **save**.

| **Proposing offers in campaigns** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Key – level 1** | **Key – level 2** | **Key – level 3** | **Description** | **Constraints** | **Example** |
| id |  |  | Seller internal product id. This is the primary key for identifying a product offer. | Required. Integer value between 1 and 16777215. | id=243409 |
| sale\_price |  |  | Seller offer sale price without VAT available in the campaign | Required. Decimal value greater than 0. Up to four decimals. | sale\_price=51.6477 |
| stock |  |  | Available stock for the campaign. Once the stock has finished, the product can no longer be ordered. | Required. Integer value between 0 and 255. | stock=1 |
| max\_qty\_per\_order |  |  | The maximum quantity that a customer can order for a product from the campaign. This column is mandatory for stock in site campaigns. | Required/Optional depending on campaign type. List of arrays. | max\_qty\_per\_order=4 |
| post\_campaign\_sale\_price |  |  | Product price after campaign end. The automatically filled price is the sale price of the product from the moment when offers are downloaded. | Optional. Decimal. Up to four decimals. | post\_campaign\_sale\_price=55.6477 |
| campaign\_id |  |  | eMAG internal campaign ID in which the proposal will be uploaded | Required. Integer value between 1 and 16777215. | campaign\_id=344 |
| not\_available\_post\_campaign |  |  | If it`s not completed, the offer will remain valid after campaign end and if it`s filled with "1", the offer will no longer be active and valid after campaign end. | Optional. Integer value. 0 or 1. | not\_available\_post\_campaign=1 |
| not\_available\_post\_campaign |  |  | If it`s not completed, the offer will remain valid after campaign end and if it`s filled with "1", the offer will no longer be active and valid after campaign end. | Optional. Integer value. 0 or 1. | not\_available\_post\_campaign=1 |
| voucher\_discount |  |  | Value of the voucher discount proposed in campaign. It should be minimum 10 (%).  You are free to decide indepedently the value of the voucher. | Optional/Required depending on campaign type. Integer value.  Maximum allowed value is 100. | voucher\_discount=10 |
| date\_intervals |  |  | A list of date intervals for which different discounts apply during the campaign. Each interval is an object with specific properties. | Optional/Required depending on campaign type. List of arrays.  For MultiDeals campaigns is required | date\_intervals": [  {  "start\_date": {  "date": "2025-04-21 00:00:00.000000",  "timezone\_type": 3,  "timezone": "Europe\\/Bucharest"  },  "end\_date": {  "date": "2025-04-22 23:59:59.000000",  "timezone\_type": 3,  "timezone": "Europe\\/Bucharest"  },  "voucher\_discount": 10,  "index": 1  } |
|  | start\_date |  | The start of the discount interval. Contains full timestamp and timezone metadata. | Optional/Required depending on campaign type.  For MultiDeals campaigns is required  Object | "start\_date": {  "date": "2025-04-21 00:00:00.000000",  "timezone\_type": 3,  "timezone": "Europe\\/Bucharest"  } |
|  |  | date | The start timestamp of the interval, in YYYY-MM-DD HH:MM:SS.ssssss format. | Optional/Required depending on campaign type.  For MultiDeals campaigns is required  String | "date": "2025-04-22 23:59:59.000000", |
|  |  | timezone\_type | Timezone type identifier (e.g., 3 for Europe/Bucharest). | Optional. Integer | "timezone\_type": 3 |
|  |  | timezone | Timezone name string (e.g., "Europe/Bucharest"). | Optional. String | "timezone": "Europe\\/Bucharest" |
|  | end\_date |  | The end of the discount interval. Structure identical to start\_date. | Optional/Required depending on campaign type.  For MultiDeals campaigns is required  Object | "end\_date": {  "date": "2025-04-22 23:59:59.000000",  "timezone\_type": 3,  "timezone": "Europe\\/Bucharest"  } |
|  |  | date | The end timestamp of the interval, in YYYY-MM-DD HH:MM:SS.ssssss format. | Optional/Required depending on campaign type.  For MultiDeals campaigns is required  String | "date": "2025-04-23 00:00:00.000000", |
|  |  | timezone\_type | Timezone type identifier (e.g., 3 for Europe/Bucharest). | Optional. Integer | "timezone\_type": 3 |
|  |  | timezone | Timezone name string (e.g., "Europe/Bucharest"). | Optional. String | "timezone": "Europe\\/Bucharest" |
|  | voucher\_discount |  | The discount percentage to be applied during this interval (e.g., 10 for 10%). | Optional/Required depending on campaign type.  For MultiDeals campaigns is required  Integer | "voucher\_discount": 12, |
|  | index |  | Order of the interval. Must be unique and incremented (e.g., 1, 2, 3, etc.). The limit is 30. | Optional/Required depending on campaign type.  For MultiDeals campaigns is required  Integer | "index": 1  "index": 2 |

|  |  |  |
| --- | --- | --- |
| **Resource** | **Example** | **Context** |
| ***campaign\_proposals*/save** |  | http method: POST |

## Smart Deals Badge

The new /api-3/smart-deals-price-check endpoint helps to retrieve the price required to qualify for the Smart Deals badge.

By providing the productId in the request, the system returns the target price needed for eligibility.

|  |  |  |
| --- | --- | --- |
| **Resource** | **Example** | **Context** |
| ***smart-deals-price-check*** |  | http method: GET |

# Processing orders

An order consists of customer details, products and discounts from vouchers. It also has information about payment method, shipping tax. Also, each order always has a status attached. The available statuses are:

0 – canceled

1 – new

2 – in progress

3 – prepared

4 – finalized

5 – returned

The resource is ***order*** and the available actions are ***read***, ***save***, ***count*** and ***acknowledge***.



## Order fields

An order has the following properties:

| **Key – level 1** | **Key – level 2** | **Description** | **Constraints** | **Example** |
| --- | --- | --- | --- | --- |
| id |  | The number that uniquely identifies an order. | Required. Integer value between 1 and 4294967295. | id=939393 |
| status |  | The order processing status. Possible values:  0 - cancelled  1 - new  2 - in progress  3 – prepared  4 - finalized  5 - returned | Required. Integer value between 0 and 5. | status=1 |
| is\_complete |  | A flag indicating if the order is complete (has all details necessary for processing) or not. Possible values:  0 - incomplete;  1 - complete. | Optional. Integer value. | is\_complete=1 |
| type |  | A flag indicating if the order contains products fulfilled by eMAG or by seller. Possible values: 2 – fulfilled by eMAG  3 – fulfilled by seller  You can edit only orders having type 3. The only action that can be done on orders having type 2 is attachment upload. | Required\*. Integer value.  \*Starting with August 2025, the type of the order is mandatory. | type=3 |
| payment\_mode\_id |  | The order payment method. Possible values:  1 - COD (cash on delivery)  2 - bank transfer  3 - online card payment | Required. Integer. | payment\_mode\_id=1 |
| detailed\_payment\_method |  | The detailed order payment method. | Optional. String. | detailed\_payment\_method = “eCREDIT” |
| delivery\_mode |  | The order delivery method. Possible values:  “courier” – home delivery  “pickup” – locker delivery | Optional. String. | delivery\_mode=”courier” |
| details |  | Order extra details. | Optional. Array. |  |
| details | locker\_id | The pickup point id, if the locker delivery option was selected by a customer. | Optional. String. | locker\_id=”dce0b7cf-dc38-11e8-a7d8-001a4a160153” |
|  | locker\_name | The pickup point name, if the locker delivery option was selected by a customer. | Optional. String. | locker\_name=”easybox eMAG Showroom” |
|  | locker\_delivery\_eligible | Indicates if the order, regardless of the delivery method selected by the customer, would fit into a locker.  0 – not eligible for locker delivery  1 – eligible for locker delivery | Optional. Integer value. | locker\_delivery\_eligible=1 |
|  | courier\_external\_office\_id | The ID represents the identification of the delivery point chosen by the customer in the courier’s internal system (the courier which makes the last mile in Romania). | Optional. String.  Should be used only by sellers included in the Door2Door China project | courier\_external\_office\_id= "183" |
| date |  | The cart submission timestamp. | Optional. Text in YYYY-mm-dd HH:ii:ss format. | date=”1970-01-01 23:59:59” |
| modified |  | The timestamp of the last update on the order. | Optional. Text in YYYY-mm-dd HH:ii:ss format. | date=”1970-01-01 23:59:59” |
| payment\_status |  | The payment status. Possible values:  0 - not paid  1 - paid | Optional. It is highly recommended to also interpret the payment status when reading orders with Card Online payment method. The key is available for both Card Online and Cash on delivery orders. For COD, payment\_status will be 0 until the cashed amount is received by eMAG. | payment\_status=0 |
| cashed\_co |  | The cashed amount from Card online payment | Optional. Integer. |  |
| cashed\_cod |  | The cashed amount from cash on delivery payment | Optional. Integer. |  |
| shipping\_tax |  | The shipment tax value. | Optional. Decimal. | shipping\_tax=”19.99” |
| shipping\_tax\_voucher\_split |  | A list of arrays describing the voucher discounts distribution on shipping tax level. | List. |  |
|  | voucher\_id | The ID of the voucher discount | Optional. Integer value between 1 and 9999999. | voucher\_id=123 |
|  | value | The value of the discount, whitout VAT | Optional. Negative value. | value=-200 |
|  | vat\_value | The value of the VAT | Optional. Negative value. | Vat\_value=-38 |
| customer |  | A list with the details about the customer, the shipping and the billing addresses. | Optional. List. | The field list is detailed below. |
| products |  | A list of arrays describing the products in the order. | List. | The field list is detailed below. |
| attachments |  | A list of arrays describing the attachments in the order. | List. | The field list is detailed below (chapter 3.1.3). |
| vouchers |  | A list describing the voucher discounts. | List. |  |
|  | voucher\_id | The ID of the voucher discount | Optional. Integer value between 1 and 9999999. | voucher\_id=123 |
|  | modified | The modified date of the voucher discount | Optional. Text in YYYY-mm-dd HH:ii:ss format. | modified="2015-04-23 11:30:09" |
|  | created | The date of the voucher discount | Optional. Text in YYYY-mm-dd HH:ii:ss format. | created="2015-04-23 11:30:09" |
|  | status | The status of the voucher discount | Optional. Integer. | status=1 |
|  | sale\_price\_vat | The value of the VAT | Optional. Negative value. | sale\_price\_vat=”-1.9355" |
|  | sale\_price | The value of the discount, without VAT | Optional. Negative value. | sale\_price="-8.0645" |
|  | voucher\_name | The name of the voucher  PAY ATTENTION! You should check from now on the name of the voucher at voucher split level, since now a voucher may be supported in a mixed way between eMAG/FD and you and can have different names. | Optional. String. | voucher\_name="eMAG giftcard" |
|  | vat | The VAT rate | Optional. Decimal. | vat="0.24" |
|  | issue\_date | The date when the voucher was issued | Optional. Text in YYYY-mm-dd format. | issue\_date="2020-06-09" |
| is\_storno |  | Mandatory key when products are returned for a finalized order. Further details [here](#_Returned_products_and). | Optional. Boolean.  True indicates partial storno. | is\_storno=true |
| reason\_cancellation |  | The order cancellation reason. Possible values:  1 - Out of stock  2 - Cancelled by the client  3 - The client can not be contacted  15 - Courier delivery term is too large  16 - Transport tax, is too large  17 - Large delivery term, until the product will arrive in our warehouse  18 - Better offer in another store  19 - Payment order has not been paid  20 - Undelivered order, courier reasons  21 - Others  22 - Order Incomplete - automatic cancellation  23 - The customer changed his mind  24 - By customer request  25 - Failed delivery  26 - Late shipment  27 - Irrelevant Order  28 - Canceled by SuperAdmin on seller request  29 - Blacklisted customer  30 - No VAT invoice  31 - The eMAG Marketplace partner requested the order cancellation  32 - The delivery estimate is too long  33 - The product is no longer available in the stock of the eMAG Marketplace partner  34 - Other reasons  35 - The delivery is too expensive  36 - The customer found a better price elsewhere  37 - The customer registered another eMAG order with a similar product  38 - The customer changed his mind, does not need the product  39 - The customer can purchase the product only by installments  40 - Other reasons EIS  41 - Out of stock - instalments  42 - The product is no longer available in the stock of the eMAG Marketplace partner - instalments  43 - The delivery estimate is too long | Optional. Integer value between 1 and 5. | cancellation\_reason=1 |
| enforced\_vendor\_courier\_accounts |  | List of the mandatory courier accounts that must be used in AWB issuing.  If null, you may use any courier from your available ones, if <> null you must use one of the ones from the array and if it’s an empty array, you won’t be able to issue AWB from MKTP. | Optional. Array. |  |

### Product field in order details

| **Key – level 1** | **Key – level 2** | **Key – level 3** | **Description** | **Constraints** | **Example** |
| --- | --- | --- | --- | --- | --- |
| id |  |  | eMAG internal order product line id. Any update on order product lines must use this id. | Required. Integer value between 1 and 9999999. id=243409 | id=123 |
| product\_id |  |  | Seller internal product id. This is the primary key for identifying a product offer. | Optional. Integer.  Important:  You cannot change product\_id on api-3/order/save. | product\_id=3331 |
| product\_voucher\_split |  |  | A list of arrays describing the voucher discounts distribution on product level. | List. |  |
|  | voucher\_id |  | The ID of the voucher discount | Optional. Integer value between 1 and 9999999. | voucher\_id=123 |
|  | value |  | The value of the discount, whitout VAT | Optional. Negative value. | value=-200 |
|  | vat\_value |  | The value of the VAT | Optional. Negative value. | Vat\_value=-38 |
|  | vat |  | The value of the VAT rate |  |  |
|  | offered\_by |  | Who supports the voucher: eMAG or seller |  |  |
|  | voucher\_name |  | The name of the voucher | Optional. String. | voucher\_name="eMAG giftcard" |
| status |  |  | The status of product of the order. Possible values: 0 - cancelled 1 - active | Required. Integer. | status=1 |
| part\_number |  |  | Manufacturer unique identifier for the product.  It corresponds to the PNK. | Optional. String between 1 and 25 characters.  Important:  The following characters will be automatically stripped:   * spaces [ ] * comma [ , ] * semicolon [ ; ]   **Example**: “part number;” will be saved “partnumber”  You cannot change part\_number on api-3/order/save. | part\_number='682133frs' |
| ext\_part\_number |  |  | Your unique identifier for the product.  Coincides with part\_number if you are the owner of the product’s documentation. | Optional. String between 1 and 25 characters.  Important:  The following characters will be automatically stripped:   * spaces [ ] * comma [ , ] * semicolon [ ; ]   **Example**: “ext part number;” will be saved “extpartnumber”  You cannot change ext\_part\_number on api-3/order/save. | ext\_part\_number='682133frs' |
| created |  |  | The date when the order product line was created. | Optional. Text in YYYY-mm-dd HH:ii:ss format. | created='2014-07-24 12:16:50' |
| modified |  |  | The date when the order product line was last modified. | Optional. Text in YYYY-mm-dd HH:ii:ss format. | modified='2014-07-24 12:18:53' |
| currency |  |  | Product price currency. | Optional. String. | currency='RON' |
| quantity |  |  | Product quantity. | Required. Integer. Positive, different than 0. | quantity=2 |
| sale\_price |  |  | The sale price without VAT. | Required. Integer. | sale\_price=12.1234 |
| details |  |  | Additional product notes. | Optional. Text. | details=”text” |
| status |  |  | The status of product of the order. Possible values: 0 - cancelled 1 - active | Required. Integer. | status=1 |
| sale\_price |  |  | The sale price without VAT. | Required. Integer. | sale\_price=12.1234 |
| details |  |  | Additional product notes. | Optional. Text. | details=”text” |
| recycle\_warranties |  |  |  |  |  |
|  | quantity |  | Warranty quantity. | Optional. Integer. | quantity=6 |
|  | sale\_price |  | The sale price without VAT. | Optional. Integer. | sale\_price=0.5 |
|  | vat\_rate |  | The rate of the VAT | Optional. Integer. | vat\_rate=0 |
|  | product\_name |  | The name of the recycle warranty: plastic, metal, glass | Optional. String. | product\_name=”Taxa SGR metal” |
|  | recycle\_warranty\_voucher\_split |  | A list of arrays describing the voucher discounts distribution on recyle warranty level. | List. |  |
|  |  | voucher\_id | The ID of the voucher discount | Optional. Integer value between 1 and 9999999. | voucher\_id=875 |
|  |  | value | The value of the discount, whitout VAT | Optional. Negative value. | value=-3 |
|  |  | vat\_value | The value of the VAT | Optional. Negative value. | vat\_value=0 |
| serial\_numbers |  |  | It’s the serial number (IMEI) of the product. If there are more pieces of a product with IMEI on your order, you will find here as many codes. | Optional. Text. |  |

**IMPORTANT**

Please note that multiple vouchers, possibly having different VAT rates, can be applied on a product. You should always read all voucher parameters.

### Customer fields in order details

The customer field has the following properties:

| **Key** | **Description** | **Constraints** | **Example** |
| --- | --- | --- | --- |
| id | The number that uniquely identifies a customer. | Optional. Integer value between 1 and2147483647. | id=1 |
| name | The customer's name. | Optional. Text. | name=”Surname Name” |
| email | This is a hash that uniquely identifies the customer’s email. | Optional. Text. | email=”1243536@emag.ro” |
| company | The name of the company. For physical person it has the same value as name. | Optional. Text. | company=”Company name ltd.” |
| gender | The customer gender. Possible values:  M - male  F – female | Optional. Text. | gender=”M” |
| code | The company registration code. | Optional. Text. | code=”14399840” |
| registration\_number | The company registration number. | Optional. Text | registration\_number=” 40/372/2002” |
| bank | The bank name. | Optional. Text. | bank=”Bank name” |
| iban | The bank account. | Optional. Text. | iban=” RO24BACX0000000031430310” |
| fax | The customer's fax number. | Optional. Text. | fax=”4021123123” |
| legal\_entity | A flag indicating if the customer is physical or juridical entity. Possible values:  0 - private entity;  1 - legal entity. | Optional. Integer value. | legal\_entity=1 |
| is\_vat\_payer | A flag indicating it the customer is vat payer. Possible values:  0 - the customer is not vat payer;  1 - the customer is vat payer. | Optional. Integer value. | is\_vat\_payer=0 |
| phone\_1 | The customer's first phone number. | Optional. Text. | phone\_1=”4021123123” |
| phone\_2 | The customer's second phone number. | Optional. Text. |  |
| phone\_3 | The customer's third phone number. | Optional. Text. |  |
| billing\_name | The customer's invoice name. | Optional. Text. | billing\_name=”Surname Name” |
| billing\_phone | The customer's invoice phone. | Optional. Text. | billing\_phone=”4021123123” |
| billing\_country | The customer's invoice country. | Optional. Text. | billing\_country=”Romania” |
| billing\_suburb | The customer's invoice county. | Optional. Text. | billing\_suburb=”Suburb” |
| billing\_city | The customer's invoice city. | Optional. Text. | billing\_city=”City” |
| billing\_locality\_id | The customer's invoice locality id. | Optional. Text. | billing\_locality\_id=”7” |
| billing\_street | The customer's invoice address. | Optional. Text. | billing\_street=”Street Name” |
| billing\_postal\_code | The customer's invoice postal code. | Optional. Text. | billing\_postal\_code=”23125” |
| shipping\_locality\_id | The shipping locality id. | Optional. Text. | shipping\_locality\_id=”7” |
| shipping\_contact | The name of the contact person that will pick up the parcel. Should be printed on the AWB. | Optional. Text. | shipping\_contact=”Name Surname” |
| shipping\_phone | The phone used by the courrier to contact the shipping person. Should be printed on the AWB. | Optional. Text. | shipping\_phone=”23125” |

### Order invoices and warranties

When pushing orders into finalized status, you should also send the invoice PDF file location for the specific order.

The resource is **order/attachments** and the available action is **save**.

The following keys should be sent in attachments array in order to display an invoice in the customer’s order details: name, url, type.

If your order contains product(s) that have a warranty, you can upload it as well, so that the customer can have access to it in his account. All you have to do on the request is to use type = 3 + order\_product\_id (the value from api-3/order/read -> products -> id), in order to indicate the exact product for that warranty.

An attachment has the following properties:

| **Key** | **Description** | **Constraints** | **Example** |
| --- | --- | --- | --- |
| order\_id | The number that uniquely identifies an order. | Required for invoices. Integer value between 1 and 4294967295. | id=939393 |
| order\_type | Indicates the type of the order. Possible values: 2 – fulfilled by eMAG  3 – fulfilled by seller  You can attach files on both order types, so you must provide the order type to attach the file correctly. | Integer. Required. | type=3 |
| order\_product\_id | The number that uniquely identifies a product. You can attach only one warranty per product. | Required for warranties. Integer value between 1 and 4294967295.  Same value as products.id from api-3/order/read. | id= 171958974 |
| name | The name of the attachment displayed to the customer (in order history or in email) | Optional. String between 1 and 60 characters | name='Invoice title' |
| url | Attachment URL. | Required. String between 1 and 1024 characters. Valid URL to document. | url=”<http://valid-url>/invoice.pdf” |
| type | The type of document attached to the order. Possible values are:   * 1 - invoice * 3 - warranty * 4 - user manual * 8 - user guide * 10 - AWB * 11 - proforma | Integer. Optional. Default 1=”Invoice”. Only .pdf files are accepted | type=1 |
| force\_download | Flag used in order to force attachment download restrictions. If value is 0 and the attachment URL has not changed, the attachment will not be downloaded again. | Integer. Optional. Default value 0. Possible values: 0,1. | force\_download=0 |

## Order notification, acknowledgment and order filters

When a new order is placed in eMAG Marketplace for the first time, it’s status is 1 (new) and a GET request with the order id is automatically made to an URL you provide (call-back URL). Ex: <http://valid_url/path?order_id>=123

In the next step, you should read the order passing the id previously mentioned and after successfully saving the order in your database you should notify us by calling back the API using the route MARKETPLACE \_URL/api-3/order/acknowledge/[orderId]. This stops the order notification system for the mentioned order.

**IMPORTANT**:

* Order acknowledge is the only method of marking the order status as “in progress”. It is available only for 3P orders.
* Clients may ask for an order to be canceled, this will be done by eMAG only if the order was not acknowledged by the seller, thus some of the orders may be read directly with status 0 (canceled).

## Order status matrix

The following matrix defines the order flow in eMAG Marketplace:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **New status** | | | | | |
| **Actual status** | 1 - New | 2 - In progress | 3 – Prepared | 4 - Finalized | 0 - Canceled | 5 - Returned |
| 1 - New | No | Yes by ACK only | No | No | No | No |
| 2 - In progress | No | Yes | Yes | Yes | Yes | No |
| 3 - Prepared | No | No | Yes | Yes | Yes | No |
| 4 - Finalized | No | No | Yes in 48h max | Yes | Yes in 48h max | Yes in RT\* + 5 days max |
| 0 - Canceled | No | Yes in 48h max | Yes in 48h max | Yes in 48h max | Yes | No |
| 5 - Returned | No | No | No | No | No | No |

*\*RT = return time allowed to customers*

**IMPORTANT**:

* We recommend setting up a periodical /order/read (cron, agent) that should identify orders that were not acknowledged. By default on /order/read we expose the last 100 orders, but you can request up to 1000 or use pagination. Do not forget to test the order status matrix against your internal order workflow. As a best practice you should either acknowledge the order prior the read or re-read the order after acknowledging it; an order can be modified by eMAG employees upon the client’s request as long as it is not acknowledged;
* You can only edit the order (add/remove products modify quantity or price) when in status 2 (in progress) or 3 (prepared);
* Once an order is finalized, you can change its status back to status 3 (prepared) or 0 (canceled) only in the first 48 hours since finalization;
* Order status “finalized” will be set automatically when issuing the first AWB for that order. See chapter [Saving AWBs](#_Saving_AWB's);
* The order status “returned” is set automatically when all the products from the initial invoice are marked as returned. The change is permitted only within the maximum return timeframe allowed to the customer.

## Order filters

You can read all your orders though the API, using filters. The following are available when counting orders:

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| id | Only the order with this value. | Optional. Integer value between 1 and4294967295. |
| createdBefore | Only the orders created before the specified date. Can only be set if “createdAfter” is present. Maximum 1 month difference. | Optional. Text in YYYY-mm-dd HH:ii:ss format. |
| createdAfter | Only the orders created after the specified date. Can only be set if “createdBefore” is present. Maximum 1 month difference. | Optional. Text in YYYY-mm-dd HH:ii:ss format. |
| modifiedBefore | Only the orders modified before the specified date. Can only be set if “modifiedAfter” is present. Maximum 1 month difference. | Optional. Text in YYYY-mm-dd HH:ii:ss format. |
| modifiedAfter | Only the orders after before the specified date. Can only be set if “modifiedBefore” is present. Maximum 1 month difference. | Optional. Text in YYYY-mm-dd HH:ii:ss format. |
| status | Only the orders with the specified status. It is a single value or a list of values. | Optional. Integer or list. |
| payment\_mode\_ id | Only the orders with the specified payment method id. It is a single value or a list of values. | Optional. Integer or list. |
| is\_complete | Only the orders with the specified completion status. | Optional. Order completion status. 1 – complete orders  0 – incomplete orders |
| type | Only the orders with the specified type | Optional.  2 – fulfilled by eMAG  3 – fulfilled by seller |

The following filters are available when reading orders:

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| itemsPerPage | The maximum number of orders to return. | Optional. Integer value between 1 and 100. |
| currentPage | The page offset. | Optional. Integer value between 1 and 65535. |
| id | Only the order with this value. | Optional. Integer value between 1 and4294967295. |
| createdBefore | Only the orders created before the specified date. Can only be set if “createdAfter” is present. Maximum 1 month difference. | Optional. Text in YYYY-mm-dd HH:ii:ss format. |
| createdAfter | Only the orders created after the specified date. | Optional. Text in YYYY-mm-dd HH:ii:ss format. |
| modifiedBefore | Only the orders modified before the specified date. Can only be set if “modifiedAfter” is present. Maximum 1 month difference. | Optional. Text in YYYY-mm-dd HH:ii:ss format. |
| modifiedAfter | Only the orders after the specified date. | Optional. Text in YYYY-mm-dd HH:ii:ss format. |
| status | Only the orders with the specified status. It is a single value or a list of values. | Optional. Integer or list. |
| payment\_mode\_id | Only the orders with the specified payment method id. It is a single value or a list of values. | Optional. Integer or list. |
| is\_complete | Only the orders with the specified completion status. | Optional. Order completion status. 1 – complete orders  0 – incomplete orders |
| type | Only the orders with the specified type | Optional. Default value = 3.  2 – fulfilled by eMAG  3 – fulfilled by seller |

## Updating orders

You cannot create new orders through the API, you can only read and update them. When updating an order, the seller should send ALL the fields initially read.

**IMPORTANT**:

* Updating products by reducing their quantities for orders with Online Card payment method is no longer possible.
* Updating product prices is no longer available
* A canceled order can no longer be reactivated if more than 48 hours have passed since cancelation

|  |  |  |
| --- | --- | --- |
| **Resource** | **Example** | **Context** |
| **order/save** |  | http method: POST |

### Removing products from an order

To remove a product from the order send the status=0 for the product or do not send it at all. Products can be removed from an order only while in status 2 or 3 (in progress or prepared) for orders with payment methods different than Online card. For returned products (the order is in status 4, finalized), please use the storno route.

**IMPORTANT**: Removing products for orders with Online Card payment method is no longer possible.

### Adding products to an existng order

To add a new product to an existing order, add it to the order by sending the product id (mandatory), name, status and sale price.

**IMPORTANT:** virtual products such as internal discounts can be inserted in an order, even if they were not previously sent to eMAG. Adding these products to an order will not make them available for purchase in the eMAG Marketplace platform.

### Returned products and storno route

A finalized order cannot be modified, it can be fully reversed by changing the order status from finalized (4) to returned (5) or have only some of the products reversed using a call with is\_storno key true.

The following conditions must be met in order for a partial storno to occur:

* Order must be in status 4
* At least one product quantity was reduced

The following scenarios can be used as a guideline for returning products (partial storno) from a finalized order:

| **Current order status** | **Request** | **isError** | **Order read** |
| --- | --- | --- | --- |
| status' => 4,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  1 =>  array (  'id' => 2,  'product\_id' => '2',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  ), | status' => 4,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 1,  'sale\_price' => '123.4567',  'status' => 1,  ),  1 =>  array (  'id' => 2,  'product\_id' => '2',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  ),  'is\_storno'=true | FALSE | status' => 4,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 1,  'sale\_price' => '123.4567',  'status' => 1,  ),  1 =>  array (  'id' => 2,  'product\_id' => '2',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  ), |
| status' => 4,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  1 =>  array (  'id' => 2,  'product\_id' => '2',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  ), | status' => 4,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  1 =>  array (  'id' => 2,  'product\_id' => '2',  'quantity' => 0,  'sale\_price' => '123.4567',  'status' => 1,  ),  ),  'is\_storno'=true | FALSE | status' => 4,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 0,  'sale\_price' => '123.4567',  'status' => 1,  ),  ), |
| status' => 4,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  1 =>  array (  'id' => 2,  'product\_id' => '2',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  ), | status' => 4,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  1 =>  array (  'id' => 2,  'product\_id' => '2',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 0,  ),  ),  'is\_storno'=true | FALSE | status' => 4,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 0,  'sale\_price' => '123.4567',  'status' => 1,  ),  ), |
| status' => 4,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  1 =>  array (  'id' => 2,  'product\_id' => '2',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  ), | status' => 4,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  1 =>  array (  'id' => 2,  'product\_id' => '2',  'quantity' => 1,  'sale\_price' => '123.4567',  'status' => 1,  ),  ), | TRUE | The request will be discarded, as you are trying to modify a finalized order without is\_storno key. |
| status' => 4,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  1 =>  array (  'id' => 2,  'product\_id' => '2',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  ), | status' => 4,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  1 =>  array (  'id' => 2,  'product\_id' => '2',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  ),  'is\_storno'=true | TRUE | The request will be discarded, as you are sending is\_storno key without any change to an order line |
| status' => 3,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  1 =>  array (  'id' => 2,  'product\_id' => '2',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  ), | status' => 3,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  1 =>  array (  'id' => 2,  'product\_id' => '2',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  ),  'is\_storno'=true | TRUE | The request will be discarded, as you are sending is\_storno key for an order with a status different than 4 |
| status' => 4,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  1 =>  array (  'id' => 2,  'product\_id' => '2',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  ), | status' => 4,  'products' =>  array (  0 =>  array (  'id' => 1,  'product\_id' => '1',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  1 =>  array (  'id' => 2,  'product\_id' => '2',  'quantity' => 2,  'sale\_price' => '123.4567',  'status' => 1,  ),  2 =>  array (  'id' => 2,  'product\_id' => '2',  'quantity' => -1,  'sale\_price' => '123.4567',  'status' => 1,  ),  ),  'is\_storno'=true | TRUE | The request will be discarded, as you are trying to send a negative quantity for a product |

# Shipping orders

For electronic deliveries and downloadable goods, please skip this section. Shipping an eMAG Marketplace order requires the seller to issue an AWB using eMAG Marketplace API.

The resource is ***AWB*** and the available actions are ***read*** and ***save.***



## Saving AWB

To save an AWB just call the API with the following parameters:

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| order\_id | Identifies the order | Required. Integer value between 1 and 4294967295.  Must be a valid order in the eMAG database, and must be owned by the seller. |
| rma\_id | Identifies the return request | Optional. Integer value between 1 and 4294967295.  Must be a valid return request in the eMAG database, and must be owned by the seller. |
| sender | \*Array explained [below](#_AWB_sender/receiver_fields) |  |
| receiver | \*Array explained [below](#_AWB_sender/receiver_fields) |  |
| locker\_id | The pickup point id. Should be filled in with the pickup point id received on the order. If filled in, the courier will deliver the parcel in the designated locker. | Optional. String value between 3 and 255 characters |
| is\_oversize | If set to 1, marks the delivery as containing oversized products | Required. Value can only be 0 or 1. |
| insured\_value | The insured value | Optional. Double value between 0 and 999999999 |
| weight | The weight of delivery | Optional. Double value between 0 and 99999 |
| envelope\_number | Number of envelopes to be delivered | Required. Integer value between 0 and 9999. If parcel\_number is 0, this parameter cannot be 0 |
| parcel\_number | Number of parcels to be delivered | Required. Integer value between 0 and 999. If envelope\_number is 0, this parameter cannot be 0 |
| observation | Observation text | Optional. String value between 0 and 255 |
| cod | Cash on delivery | Required. Double value between 0 and 999999999 |
| courier\_account\_id | Unique identifier for seller’s courier account. If not provided, a default account will be used. | Optional. Integer. |
| pickup\_and\_return | If set to 1, sender expects something in return to this expedition (documents, buy-back products, etc).  For an AWB belonging to a return must be 0 or not sent. | Optional. Value can only be 0 or 1. |
| saturday\_delivery | If set to 1, sender requests the package to be delivered on Saturday. | Optional. Value can only be 0 or 1. |
| sameday\_delivery | If set to 1, sender requests the package to be delivered the same day. | Optional. Value can only be 0 or 1. |
| dropoff\_locker | If set to 1, you will issue the AWB without a request sent to courier to pickup your order. If set to 0 or not provided, the courier will pickup up the order from your warehouse. | Optional. Value can only be 0 or 1. Used only for orders. |
| unboxing | If set to 1, you will issue the AWB with the “Open on delivery” option for the customer. If set to 0 or not provided, the AWB will be issued without the “Open on delivery” option for the customer. | Optional. Value can only be 0 or 1. It will be taken into consideration only for sellers which have control on this field. In order to be allowed to control this field, you need to contact Marketplace team. |
| date |  | Required for AWBs belonging to returns. Must be at least the next day of the AWB issuing. |

An AWB S/R (sender/receiver) has the following properties:

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| name | S/R's name | Required. String value between 3 and 255 |
| contact | Receiver's contact person name | Required. String value between 1 and 255 |
| phone1 | S/R first phone number | Required. String value between 8 and 11 digits (only '+' character is allowed at the beginning of the string) |
| phone2 | S/R second phone number | Optional. String value between 8 and 11 digits (only '+' character is allowed at the beginning of the string) |
| legal\_entity | If Receiver is legal entity (applicable only to receiver) | LEGAL\_ENTITY\_NO = 0 LEGAL\_ENTITY\_YES = 1 |
| address\_id | The id of your address (seller address). | Optional.  String value between 0 and 21  If sent on S (for orders) or R (for returns), we will use the address saved in your account, regardless of the other address fields sent in the request. |
| locality\_id | S/R's locality\_id | Required. Integer value between 1 and 4294967295.  Must be a valid locality in the eMAG database. |
| street | S/R's street | Required. String value between 3 and 255 |
| zipcode | S/R's zipcode | Optional. String value between 1 and 255 |

**Examples of requests**

|  |  |  |
| --- | --- | --- |
| **Resource** | **Example** | **Context** |
| **awb/save** | s | http method: POST  Seller wants to issue AWB for orders |
| **awb/save** |  | http method: POST  Seller wants to issue AWB for returns |

**IMPORTANT:**

* For orders with “pickup” as a delivery method if you do not change the locker id that is already included in the “shipping\_street” field the AWB will be issued as a locker delivery using the proper courier account regardless of the actual courier account you specified when issuing the AWB.

## Reading AWB

The following filters are available when reading AWBs:

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| emag\_id | The eMAG internal barcode id | Integer value between 1 and 4294967295.  Must be a valid AWB in eMAG database. |
| reservation\_id | The eMAG internal AWB reservation id | Integer value between 1 and 4294967295.  Must be a valid AWB in eMAG database. |

An AWB has the following properties:

| **AWB – read** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Key – level 1** | **Key – level 2** | **Description** | **Constraints** | **Example** |
| emag\_id |  | The eMAG internal AWB id | Integer. | emag\_id=243409 |
| order\_id |  | The id of the order on which the AWB was issued | Integer. | order\_id=243409 |
| rma\_id |  | The id of the return request on which the AWB was issued | Integer. | rma\_id=243409 |
| type |  | The type of the order/return request.  2 – fulfilled by eMAG  3 – fulfilled by seller | Integer. | type=3 |
| weight |  | The weight of delivery | Integer. | weight=1 |
| awb\_type |  | The type of delivery. Possible values:  1 – delivery to customer  2 – pickup from customer  The key is returned only for AWBs belonging to return requests. | Integer. | awb\_type=1 |
| awb |  | The AWB | List of arrays. |  |
|  | emag\_id | The eMAG internal AWB barcode id | Integer | emag\_id=243409 |
|  | awb\_number | The AWB number | String | awb\_number = “2EMG00011012” |
|  | awb\_barcode | The AWB barcode | String | awb\_barcode = “2EMG00011012001” |
| status |  | The status of the delivery | List of arrays. |  |
|  | code | The code status of the delivery | String. | code=”DLV” |
|  | name | The name of the status of the delivery | String. | name=”Delivered” |
|  | description | The description of the status of the delivery | String. | description=”AWB delivered” |
| courier |  | The courier used for issuing the AWB | List of arrays. |  |
|  | courier\_account\_id | The eMAG internal courier account id used for issuing the AWB | Integer. | courier\_account\_id=5186 |
|  | courier\_name | The eMAG internal courier name used for issuing the AWB | String. | courier\_name="SAMEDAY" |

## Reading AWB PDF files

To download an AWB PDF file call the MARKETPLACE\_API\_URL/awb URL as in the example below

|  |
| --- |
| <html>  Running...<br>  <?  $username = 'user';  $password = 'pass';  $hash = base64\_encode($username . ':' . $password);  $headers = array(  'Authorization: Basic ' . $hash  );  $ch = curl\_init();  curl\_setopt($ch, CURLOPT\_URL,'**https://marketplace-api.emag.ro/awb/read\_pdf?emag\_id=9755945&awb\_format=A4**';  curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);  curl\_setopt($ch, CURLOPT\_FOLLOWLOCATION, 1);  curl\_setopt($ch, CURLOPT\_HEADER, 0);  curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);  curl\_setopt($ch, CURLOPT\_CUSTOMREQUEST, 'GET');  curl\_setopt($ch, CURLOPT\_HTTPHEADER, $headers);  $result = curl\_exec($ch);  echo $result . "\n";  ?>  </html> |

**Optional:** You can set the paper format on PDF download link by using the parameter "awb\_format=A4" in the link. The possible values are A4, A5, A6.

To download an AWB just call the API with the following parameters:

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| emag\_id | The AWB's eMAG id. | Integer value between 1 and 4294967295.  Must be a valid AWB in eMAG database. |
| awb\_format | The paper format on PDF download | The possible values are A4, A5, A6 and ZPL |

**You can read only AWBs issued via API, only for them you receive the “emag\_id” key, which is mandatory.**

## Reading AWB ZPL type

To read an AWB in ZPL type call the MARKETPLACE\_API\_URL/awb URL as in the example below

|  |
| --- |
| <html>  Running...<br>  <?  $username = 'user';  $password = 'pass';  $hash = base64\_encode($username . ':' . $password);  $headers = array(  'Authorization: Basic ' . $hash  );  $ch = curl\_init();  curl\_setopt($ch, CURLOPT\_URL,'**https://marketplace-api.emag.ro/awb/read\_zpl?emag\_id=9755945**';  curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);  curl\_setopt($ch, CURLOPT\_FOLLOWLOCATION, 1);  curl\_setopt($ch, CURLOPT\_HEADER, 0);  curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);  curl\_setopt($ch, CURLOPT\_CUSTOMREQUEST, 'GET');  curl\_setopt($ch, CURLOPT\_HTTPHEADER, $headers);  $result = curl\_exec($ch);  echo $result . "\n";  ?>  </html> |

Using this request will return a base64 encoded content of the ZPL format as in the example below

|  |  |  |
| --- | --- | --- |
| **Resource** | **Example** | **Context** |
| **awb/read\_zpl** |  | http method: POST |

## Counting Localities

In order to issue an AWB you need to submit the correct locality id. You can also use the id from the order.

The resource is ***locality*** and the available actions are ***read*** and ***count***.

The following filters are available when counting localities:

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| emag\_id | The locality with this id | Integer |
| name | All localities with this name | String of length between 0 and 60 |
| modified | All localities modified after this date | Date with the 'Y-m-d H:i:s' format |

## Reading Localities

The following filters are available when reading localities:

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| emag\_id | The locality with this id | Integer |
| name | All localities with this name | String of length between 0 and 60 |
| region2 | The name of the county for which you want localities | String of length between 0 and 60 |
| country\_code | Country Alpha-2 code. Available values:  RO – Romania  BG – Bulgaria  HU – Hungary  PL – Poland  EL – Greece  DE – Deutchland | String of length 0 and 60 |
| modified | All localities modified after this date | Date with the 'Y-m-d H:i:s' format |
| itemsPerPage | The maximum number of localities to return. | Optional. Integer value between 1 and 100. |
| currentPage | The page offset. | Optional. Integer value between 1 and 65535. |

When not using the country filter the API will respond only with localities from the platform country.

A locality has the following properties:

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| emag\_id | The id of the locality | Integer |
| name | The name of the locality | String of length between 0 and 60 |
| name\_latin | The latin name of the locality | String of length between 0 and 60 |
| region([1-4]+) | Region name | String of length between 0 and 60 |
| region([1-4]+)\_latin | Region name latin version | String of length between 0 and 60 |
| geoid | Geographic id of the location | Integer |
| modified | Last modification date | Date with the 'Y-m-d H:i:s' format |
| zipcode | The zipcode of the locality | String of length between 0 and 10 |
| country\_code | Country Alpha-2 code | String of length between 0 and 60 |
| iso2 | Combination between country’s code and county’s code | String of length between 0 and 60 |

## Reading courier accounts

In order to issue an AWB you need to submit the correct courier account id.

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| account\_id | The id of the account | Integer |
| account\_display\_name | The name of the account | String of length between 0 and 60 |
| courier\_account\_type | The type of the account | Integer. Possible values: 1 - RMA; 2 - Order; 3 - RMA & Order; 4 - Non Marketplace |
| courier\_name | The name of the courier | String of length between 0 and 60 |
| courier\_account\_properties | The courier account properties | Array. Possible values:  0 – Regular (HD)  2 – Lockers  6 – Offices  8 – Crossborder  9 – BPO |
| created | The account creation date | Date with the 'Y-m-d H:i:s' format |
| status | Account status | Integer. Possible values: 1 - Active; 0 - Inactive |
| pickup\_country\_code | Country Alpha-2 code.  The country from where the courier picks up the orders | String of length between 0 and 60 |

## Reading addresses

In order to issue an AWB you need to provide the pickup/return address. Your addresses are defined by the following fields:

|  |  |  |
| --- | --- | --- |
| **Key** | **Description** | **Constraints** |
| address\_id | The id of the address | String of length between 0 and 21 |
| country\_id | The id of the country | Integer |
| country\_code | Country Alpha-2 code | String of length between 0 and 45 |
| address\_type\_id | The type of the address. Possible values are:   * 1 - return * 2 - pickup * 3 - invoice (head quarter) * 4 - delivery estimates | Integer |
| locality\_id | The id of the locality | Integer |
| suburb | The county of the address | String of length between 0 and 200 |
| city | The city of the address | String of length between 0 and 300 |
| address | The address: containing street, street number, etc. | String of length between 0 and 65535 |
| zipcode | The postal code of the address | String of length between 0 and 100 |
| quarter | The quarter of the address | String of length between 0 and 200 |
| floor | The quarter of the address | String of length between 0 and 100 |
| is\_default | Indicates if the address is set as default for pickup/return. Possible values are:   * false - is not default * true - is default | Boolean |

# Processing return requests

A return request consists of customer details, products and reason for returning products. Each return request always has a status attached. The available statuses are:

1 - incomplete

2 - new

3 - acknowledged

4 - refused

5 - canceled

6 - received

7 - finalized

The resource is ***RMA*** and the available actions are ***read***, ***save***.

Return requests fields

The message structure for both read and save actions is detailed below:

| **RMA – read and save/update** | | | | |
| --- | --- | --- | --- | --- |
| **Key – level 1** | **Key – level 2** | **Description** | **Constraints** | **Example** |
| emag\_id |  | Return request eMAG system ID | Required.  Type: bigint |  |
| id |  | Seller internal return request ID | Type: bigint |  |
| order\_id |  | The id of the order in which the product to be returned was included | Required.  Type: bigint |  |
| type |  | A flag indicating if the return request contains products fulfilled by eMAG or by seller. Possible values: 2 – fulfilled by eMAG  3 – fulfilled by seller | Required.  Type: Integer. | type=3 |
| invoice\_number |  | Invoice of the order in which the product to be returned was included | Optional.  Type: varchar  Default value: NULL |  |
| customer\_name |  | Customer name | Required.  Type: varchar |  |
| customer\_company |  | Customer company info | Optional.  Type: varchar  Default value NULL |  |
| customer\_phone |  | Customer phone no. | Required.  Type: varchar |  |
| products |  | Product to be returned info | Array |  |
|  | id | eMAG internal return product line id. Any update on return product lines must use this id. | Required. Integer value between 1 and 9999999. | id=123 |
|  | product\_emag\_id | Product to be returned eMAG ID |  |  |
|  | product\_id | Product to be returned seller internal ID | Required.  Type: int |  |
|  | quantity | Product quantity | Required  Type: int |  |
|  | product\_name | Product name | Required.  Type: varchar |  |
|  | return\_reason | It holds the return reason selected by the customer. In the attached files you will find the possible IDs for the return reasons and their hierarchy.      Note: If you opt for a return reason that has other higher level reason(s), you will only need to fill in the last level ID from the hierarchy. | Required  Type: int |  |
|  | observations | Free text notes field.    Note: According to the attached files above, the observations key could be optional (value 0 or value 1) or mandatory (value 2). | Optional/ Required  Type: text  Default value: NULL |  |
|  | diagnostic | Diagnostic after product analysis | Optional.  Type: int  Default value NULL |  |
|  | reject\_reason | Reason of return rejection | Optional.  Type: int  Default value NULL |  |
|  | refund\_value | Refund value | Optional.  Type: varchar |  |
| awbs |  | Issued AWBs | List of arrays. |  |
|  | reservation\_id | eMAG internal AWB reservation id. Use this id to read an AWB | Optional.  Type: int | reservation\_id= 4528511 |
| pickup\_country |  |  | Required.  Type: Varchar |  |
| pickup\_suburb |  |  | Required.  Type: Varchar |  |
| pickup\_city |  |  | Required.  Type: Varchar |  |
| pickup\_address |  |  | Required.  Type: Varchar |  |
| pickup\_address\_id |  | Id of address already saved on the customers' account | Optional.  Type: Int  Default value: NULL |  |
| pickup\_zipcode |  |  | Optional.  Type: Varchar  Default value: NULL |  |
| pickup\_date |  | Returned product pickup date (in case of vendor pickup from customer) | Optional.  Type: datetime  Default value NULL |  |
| pickup\_locality\_id |  | The internal eMAG ID of the pickup address city/locality | Required.  Type: int |  |
| pickup\_method |  | The product pickup method selected by the customer when inserting the return request. Possible values:   * 1 - eMAG courier * 2 - Seller’s own courier * 3 - Sent by client | Required.  Type: int |  |
| return\_reason |  | It holds the return reason selected by the customer. In the attached files you will find the possible IDs for the return reasons and their hierarchy.      Note: If you opt for a return reason that has other higher level reason(s), you will only need to fill in the last level ID from the hierarchy. | Required  Type: int |  |
| observations |  | Free text notes field.  Note: According to the attached files above, the observations key could be optional (value 0 or value 1) or mandatory (value 2). | Optional/ Required  Type: text  Default value: NULL |  |
| return\_type |  | It holds the return type selected by the customer. Possible values:  1 - Replacement with same product  2 - Replacement with a different product  3 - Refund  4 - Cancel payment contract for this product  5 - Voucher | Required.  Type: Int |  |
| return\_address\_id |  | It will hold the id of the return address selected by the vendor in RMA UI | Optional.  Type: Int  Default value: id of the address set as default in vendor profile address page. |  |
| return\_tax\_value |  | Shipping tax for refused returned products (VAT included)  The currency used will be the platforms' default | Optional.  Type: Float Default value: NULL |  |
| customer\_account\_iban |  |  | Type: Varchar  Default value: NULL |  |
| customer\_account\_bank |  |  | Type: Varchar  Default value: NULL |  |
| customer\_account\_beneficiary |  |  | Type: Varchar  Default value: NULL |  |
| replacement\_product\_emag\_id |  | The eMAG ID of the replacement product | Type: Int  Default value: NULL |  |
| replacement\_product\_id |  | The seller internal ID of the replacement product | Type: Int  Default value: NULL |  |
| replacement\_product\_name |  |  | Type: Varchar  Default value: NULL |  |
| replacement\_product\_quantity |  |  | Type: Int  Default value: NULL |  |
| date |  | RMA request insertion date | Required.  Type: datetime |  |
| request\_status |  | RMA request status. Possible values:  1 - Incomplete  2 - New  3 - Approved  4 - Refused  5 - Canceled  6 - Received  7 - Finalized | Optional.  Type: Int |  |



## Return requests filters

When reading return requests the following filters are available:

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| id | Seller internal return request ID |  |
| emag\_id | eMAG return request ID |  |
| order\_id | Order on which the product to be returned was included |  |
| product\_id | Seller internal returned product ID |  |
| product\_emag\_id | eMAG returned product ID |  |
| request\_status | Return request status ID |  |
| date\_start | Only the returns placed starting with the mentioned date | Text in YYYY-mm-dd HH:ii:ss format. |
| date\_end | Only the returns placed before the mentioned date | Text in YYYY-mm-dd HH:ii:ss format. |
| type | The type of the order/return request.  2 – fulfilled by eMAG  3 – fulfilled by seller | Integer |

## Status change permissions

The following matrix defines the return request processing flow in eMAG Marketplace:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **New status** | | | | | | |
| **Actual status** | 2 - New | 3 - Acknowledged | 4 - Rejected | 5 - Canceled | 6 - Received | 7 - Finalized |
| 2 - New | Yes | Yes | No | Yes | No | No |
| 3 - Acknowledged | No | Yes | No | Yes | Yes | No |
| 4 - Rejected | No | No | Yes | No | No | No |
| 5 - Canceled | No | No | No | Yes | No | No |
| 6 - Received | No | No | Yes | No | Yes | Yes |
| 7 - Finalized | No | No | No | No | No | Yes |

*\*Some of the statuses were left out by design; these should not be used in any seller implementation*

## Return request deliveries

There are two types of possible deliveries for the return requests:

* pick-up requests - courier picks up the returned product(s) from the customer and delivers them to the seller
* regular deliveries - courier delivers the returned/replaced product back to the customer

 The delivery requests will be generated using the ***AWB save*** resource.

## Examples requests and responses

|  |  |  |
| --- | --- | --- |
| **Resource** | **Example** | **Context** |
| **RMA/read** |  | http method: POST  Seller has return requests |
| **RMA/read** |  | http method: POST  Seller does not have return requests |
| **RMA/save** |  | http method: POST |

# Invoice API

The following documentation covers the available APIs used to:

* Read invoice categories
* Read invoice data
* Read customer invoice data

## Reading invoice categories

Every MKTP invoice is included in a category. In order to read all invoice data of a specific MKTP invoice type, first call the following API without any parameter **api-3/invoice/categories**.

|  |  |  |
| --- | --- | --- |
| **Resource** | **Example** | **Context** |
| **invoice/categories** |  | http method: POST |

As a result, we will return a collection of categories and the invoice name as follows:

| **Key – level 1** | **Key – level 2** | **Key – level 3** | **Description** | **Constraints** | **Example** |
| --- | --- | --- | --- | --- | --- |
| category |  |  | Invoice type | Type: string | category='FC' |
| name |  |  | Invoice type name | Type: string | name='Commission' |

|  |  |  |
| --- | --- | --- |
| **Resource** | **Example** | **Context** |
| **invoice/categories** |  | http method: POST |

## Reading invoice data

URL: **api-3/invoice/read**

The resource is ***invoice*** and for the moment the only available action is ***read***.

Reading invoice data without parameters, will generate a response containing the last 100 invoices.

You can read the invoice details using the following available filters:

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| category | The invoice category from the results displayed when calling the api-3/invoice/categories API. | Optional  category='FC' |
| number | The invoice series+number. | Optional  number='C-MKTP-100001' |
| date\_start | Only invoices created after date\_start. | Optional  Text in YYYY-mm-dd format |
| date\_end | Only invoices created before date\_end. | Optional  Text in YYYY-mm-dd format |
| itemsPerPage | The maximum number of invoice data to return. | Optional. Integer value between 1 and 1000. Default value=100. |
| currentPage | The page offset. | Default value=1. |

Reponse:

| **Key – level 1** | **Key – level 2** | **Key – level 3** | **Description** | **Constraints** | **Example** |
| --- | --- | --- | --- | --- | --- |
| total\_results |  |  | Number of invoices identified | Type:interger | total\_results=1 |
| invoices | category |  | Invoice type | Type: string | category='FC' |
|  | name |  | Invoice name | Type: string | name=’Commision’ |
|  | number |  | Invoice series+number | Type: string | number='C-MKTP-100001' |
|  | date |  | The date when the invoice was created. | Text in YYYY-mm-dd | date= '2020-07-24' |
|  | is\_storno |  | The invoice represents a reversal of another invoice. | Type: integer | is\_storno=1/is\_storno=0 |
|  | supplier | name | Supplier name (legal name) | Type: string between 1 and 100 characters. | name='Dante International SA' |
|  |  | register\_number | Registration number | Type: string between 1 and 50 characters. | register\_number='J40/372/2002' |
|  |  | cif | Unique Identification Code | Type: string between 1 and 50 characters. | cif='14399840' |
|  |  | tax\_code | VAT Number | Type: string between 1 and 50 characters. | tax\_code='RO14399840' |
|  |  | social\_capital | Subscribed and paid capital | Type: string between 1 and 50 characters. | social\_capital='1.210.822 RON' |
|  |  | iban | Bank account | Type: string between 1 and 100 characters. | iban='RO73INGB0001008199078940' |
|  |  | bank | Bank name | Type: string between 1 and 100 characters. | bank='ING BANK' |
|  |  | address | Headquarters | Type: string between 1 and 255 characters. | address=' 148 Virtutii, E47, 060787, Sector 6,  Bucuresti' |
|  |  | phone\_number | Phone number | Type: string between 1 and 50 characters. | phone=number='40212005200' |
|  | customer | name | Buyer name (legal name) | Type: string between 1 and 100 characters. | name='Test SRL' |
|  |  | register\_number | Registration number | Type: string between 1 and 50 characters. | register\_number=' JXX/XXX/2002'' |
|  |  | cif | Unique Identification Code | Type: string between 1 and 50 characters. | cif='123456' |
|  |  | tax\_code | VAT Number | Type: string between 1 and 50 characters. | tax\_code='RO123456' |
|  |  | iban | Bank account | Type: string between 1 and 100 characters. | iban=' RO00INGB0000000000070000' |
|  |  | bank | Bank name | Type: string between 1 and 100 characters. | bank='ING' |
|  |  | country | Country | Type: string between 1 and 100 characters. | country='Romania' |
|  |  | address | Headquarters | Type: string between 1 and 255 characters. | address='Strada ABC, Bucuresti' |
|  | lines | product\_name | Products/Services Description | Type: text | product\_name='Comision aferent desfasurator \_dc\_082015\_1443024267\_v1, conform contract' |
|  |  | unit\_of\_measure | Unit of measure | Type: string between 1 and 20 characters | unit\_of\_measure='Buc' |
|  |  | quantity | Quantity | Type: double | quatity=1 |
|  |  | unit\_price | Unit value | Type: double | unit\_price=100 |
|  |  | vat\_rate | VAT Rate | Type: smallint | vat\_rate=19 |
|  |  | value | Product value (without VAT) | Type: integer | value=119 |
|  |  | vat\_value | VAT value | Type: integer | vat\_value=19 |
|  | payment\_term |  | Invoice due date | Type: integer | payment\_term=0 |
|  | total\_without\_vat |  | Total invoice without VAT | Type: integer | total\_without\_vat=100 |
|  | total\_vat\_value |  | Total invoice VAT value | Type: integer | total\_vat\_value=19 |
|  | total\_with\_vat |  | Total invoice with VAT | Type: integer | total\_with\_vat=119 |
|  | currency |  | Invoice currency | Type: string | currency='RON' |

## Reading customer invoice data

URL: **api-3/customer-invoice/read**

The resource is ***customer-invoice*** and for the moment the only available action is ***read***.

Reading customer invoice data without parameters, will generate a response containing the last 100 invoices.

You can read the customer invoice details using the following available filters:

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| category | The invoice category: normal or storno invoice. | Optional  category='normal' / category='storno' |
| order\_id | The order which was invoiced. | Optional  order\_id='148717039' |
| number | The invoice series+number. | Optional  number='PRAF100010' |
| date\_start | Only invoices created after date\_start. | Optional  Text in YYYY-mm-dd format |
| date\_end | Only invoices created before date\_end. | Optional  Text in YYYY-mm-dd format |
| itemsPerPage | The maximum number of invoice data to return. | Optional. Integer value between 1 and 1000. Default value=100. |
| currentPage | The page offset. | Default value=1. |

Response:

| **Key – level 1** | **Key – level 2** | **Key – level 3** | **Description** | **Constraints** | **Example** |
| --- | --- | --- | --- | --- | --- |
| total\_results |  |  | Number of invoices identified | Type:interger | total\_results=1 |
| invoices | category |  | Invoice type | Type: string | category='storno' |
|  | order\_id |  | The order which was invoiced. | Type: integer | order\_id='148717039' |
|  | number |  | Invoice series+number | Type: string | number='PRAF101092' |
|  | date |  | The date when the invoice was created. | Text in YYYY-mm-dd | date= '2020-11-11' |
|  | is\_storno |  | The invoice represents a reversal of another invoice. | Type: integer | is\_storno=1/is\_storno=0 |
|  | reversal\_for |  | The invoice which was canceled through this invoice. | Type:string | reversal\_for='PRAF101030' |
|  | supplier | name | Supplier name (legal name) | Type: string between 1 and 100 characters. | name=Dobre BA Shop SRL' |
|  |  | register\_number | Registration number | Type: string between 1 and 50 characters. | register\_number='J40/900/2000' |
|  |  | cif | Unique Identification Code | Type: string between 1 and 50 characters. | cif='10874881' |
|  |  | tax\_code | VAT Number | Type: string between 1 and 50 characters. | tax\_code='RO10874881' |
|  |  | social\_capital | Subscribed and paid capital | Type: string between 1 and 50 characters. | social\_capital='200 RON' |
|  |  | iban | Bank account | Type: string between 1 and 100 characters. | iban='RO02BTRLRONCRT00W6717503' |
|  |  | bank | Bank name | Type: string between 1 and 100 characters. | bank='BANCA TRANSILVANIA S.A.' |
|  |  | country | Country | Type: string between 1 and 100 characters. | country='Romania' |
|  |  | address | Headquarters | Type: string between 1 and 255 characters. | address='Colentina 32, bl 2 sc 4 et 33  Bucuresti' |
|  | customer | name | Buyer name (legal name) | Type: string between 1 and 100 characters. | name='Test Test' |
|  |  | register\_number | Registration number | Type: string between 1 and 50 characters. | register\_number=' ' |
|  |  | cif | Unique Identification Code | Type: string between 1 and 50 characters. | cif='' |
|  |  | tax\_code | VAT Number | Type: string between 1 and 50 characters. | tax\_code='' |
|  |  | iban | Bank account | Type: string between 1 and 100 characters. | iban='' |
|  |  | bank | Bank name | Type: string between 1 and 100 characters. | bank='' |
|  |  | country | Country | Type: string between 1 and 100 characters. | country='RO' |
|  |  | address | Headquarters | Type: string between 1 and 255 characters. | address='Strada ABC, Bucuresti' |
|  | lines | product\_name | Products/Services Description | Type: text | product\_name='Usa de intrare, termopan , model Roxy white 110x210[61-110x210] ' |
|  |  | unit\_of\_measure | Unit of measure | Type: string between 1 and 20 characters | unit\_of\_measure='Buc' |
|  |  | quantity | Quantity | Type: double | quatity=-1 |
|  |  | unit\_price | Unit value | Type: double | unit\_price=1008.4 |
|  |  | vat\_rate | VAT Rate | Type: smallint | vat\_rate=19 |
|  |  | value | Product value (without VAT) | Type: integer | value=-1008.4 |
|  |  | vat\_value | VAT value | Type: integer | vat\_value=-191.6 |
|  | total\_without\_vat |  | Total invoice without VAT | Type: integer | total\_without\_vat=-1008.4 |
|  | total\_vat\_value |  | Total invoice VAT value | Type: integer | total\_vat\_value=-191.6 |
|  | total\_with\_vat |  | Total invoice with VAT | Type: integer | total\_with\_vat=-1200 |
|  | currency |  | Invoice currency | Type: string | currency='RON' |