**eMAG Marketplace API**

**Documentation V3.6**

**18.11.2014**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date modified** | **Changes** |
| 3.4 | 26.08.2014 | Product families  Example for sending a new product Example for reading an order |
| 3.5 | 29.10.2014 | Offer start date details |
| 3.6 | 06.11.2014 | More details in product / order keys Removed keys on AWB save |

Table of Contents

[1. eMAG Marketplace API 3](#_Toc388603036)

[1.1. Conventions 3](#_Toc388603037)

[1.2. Request, resources and actions 3](#_Toc388603038)

[1.3. Pagination and filters 5](#_Toc388603039)

[1.4. Response 5](#_Toc388603040)

[2. Sending products and offers 6](#_Toc388603041)

[2.1. Reading eMAG categories and characteristics 6](#_Toc388603042)

[2.2. Reading VAT rates 7](#_Toc388603043)

[2.3. Sending a new product 7](#_Toc388603044)

[2.4. Updating existing offer 13](#_Toc388603045)

[2.5. Attaching offers on existing eMAG products 14](#_Toc388603046)

[2.6. Warehouse read 14](#_Toc388603047)

[3. Processing orders 14](#_Toc388603048)

[3.1. Order fields 15](#_Toc388603049)

[3.1.1. Product field in order details 16](#_Toc388603050)

[3.1.2. Customer fields in order details 16](#_Toc388603051)

[3.1.3. Order invoices 17](#_Toc388603052)

[3.2. Order notification, acknowledgment and order filters 18](#_Toc388603053)

[3.3. Order status matrix 19](#_Toc388603054)

[3.4. Order filters 19](#_Toc388603055)

[3.5. Updating orders 20](#_Toc388603056)

[3.5.1. Removing products from an order 23](#_Toc388603057)

[3.5.2. Adding products to an existing order 23](#_Toc388603058)

[3.5.3. Returned products 23](#_Toc388603059)

[4. Shipping eMAG Marketplace orders 23](#_Toc388603060)

[4.1. Saving AWB's 24](#_Toc388603061)

[4.2. Reading AWB PDF files 25](#_Toc388603062)

[4.3. Counting Localities 25](#_Toc388603063)

[4.4. Reading Localities 25](#_Toc388603064)

[4.5. Reading AWB 26](#_Toc388603065)

[5. Using the messaging service 26](#_Toc388603066)

[5.1. Reading messages 26](#_Toc388603067)

[5.2. Sending messages 27](#_Toc388603068)

# 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
* use the internal messaging system

## Conventions

We define MARKETPLACE\_API\_URL constant of being the API URL of the platform (ex: [https://marketplace.emag.ro/api-3](https://marketplace.emag.ro/api-2))

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

To access the API, simply pass your vendor username, code and well computed hash. Please note that user should be granted API rights in order to access the API.

|  |
| --- |
| $hash = sha1(http\_build\_query($data).sha1('testpassword')); |

## 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\_URL/api-3/**resource/action**

Ex: [https://marketplace.emag.ro/api-3/product\_offer/save](https://marketplace.emag.ro/api-2/product_offer/save)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **RESOURCES AND AVAILABLE ACTIONS** | | | | | |
| **Resource** | **Resource URL** | **Available actions** | | | |
| **product\_offer** | MARKETPLACE\_URL/api-3/product\_offer |  | save |  |  |
| **order** | MARKETPLACE\_URL/api-3/order | read | save | count | acknowledge |
| **message** | MARKETPLACE\_URL/api-3/message | read | save | count |  |
| **category** | MARKETPLACE\_URL/api-3/category | read |  |  |  |
| **vat** | MARKETPLACE\_URL/api-3/vat | read |  |  |  |
| **locality** | MARKETPLACE\_URL/api-3/locality | read | count |  |  |
| **awb** | MARKETPLACE\_URL/api-3/awb | read | save |  |  |
| **warehouse** | MARKETPLACE\_URL/api-3/warehouse | read |  |  |  |

PHP code example using the resource "order" and the action "read":

|  |
| --- |
| $data =  array (  'currentPage' => 1,  'itemsPerPage' => 10  );  $hash = sha1(http\_build\_query($data) . sha1('vendorpassword'));  $requestData = array(  'code' => 'vendorcode',  'username' => 'vendoruser',  'data' => $data,  'hash' => $hash);  $ch = curl\_init();  curl\_setopt($ch, CURLOPT\_URL, MARKETPLACE\_URL . '/api-3/order/read');  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($ch, CURLOPT\_CUSTOMREQUEST, 'POST');  curl\_setopt($ch, CURLOPT\_POSTFIELDS, http\_build\_query($requestData));  $result = curl\_exec($ch); |

The API needs authorization and has an IP level filtering. Before testing, vendors 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 4 mandatory keys as following:

|  |  |
| --- | --- |
| **REQUEST** | |
| **Key** | Description |
| **username** | User name of vendor. Will be provided on vendor creation. |
| **code** | Code of vendor. Will be provided on vendor creation. |
| **data** | Data to be passed to the API. The following document will describe keys. |
| **hash** | SHA1 hash of URL encoded data concatenated with SHA1 hash of user password. |

## 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 1000. | 100 | itemsPerPage=1000 |

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:** We recommend setting up alerts for API calls with no response, API calls with response “isError:true” or HTTP response codes different than 200 OK. The logs should save both the request and response for debugging. As a best practice, you should keep a log with all the API responses for the last 30 days.

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”.

# Sending products and offers

We define a **product** as a list of elements displayed in in eMAG platform for a product page. These elements are:

* Name
* Brand
* Part number
* Description
* Images
* Product characteristics (and product families)
* Category
* Barcodes (optional)
* Other attachments (optional)

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

* Price
* VAT rate
* Warranty
* Numerical stock or availability info
* Handling\_time

eMAG Marketplace API allows a vendor to:

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

## Reading eMAG categories, characteristics and family\_types

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

Reading categories without parameters will generate a response containing the first 100 categories.

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.

Usually, the corresponding categories for a vendor’s products are provided by eMAG Marketplace vendor support team. But you can read the categories and their characteristics through the API also.

The resource is ***category*** and the available action is ***read***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **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” |
| characteristics |  |  | All characteristics available in category | List of arrays |  |
|  | id |  | Characteristic eMAG id | Integer | id=38 |
|  | name |  | Characteristic name | String | name=”Audio” |
|  | display\_order |  | Characteristic display order | Integer | display\_order=6 |
| 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=”cantitate” |
|  | 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 |  |

## 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.

## Sending a new 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.

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 |  | Vendor 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 |  | Vendor internal category id. | Integer. Optional. |  |
| 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 |
| 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 128 characters. | 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” |
| weight |  | The weight of the product | Optional. Decimal value between 0 and 999999. Up to six decimals. | weight=12.123456 |
| 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 | Vendor image URL. Should be consistent with eMAG Product Documentation Standard. | Required. String between 1 and 1024 characters. Valid URL. JPG, JPEG or PNG file type. | url=”http://valid-url.jpg” |
| 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” |
| 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. Vendor 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 vendor website. | Required. String between 1 and 2014 characters. | url=”http://valid-url.html” |
| warranty |  | The warranty offered in months. | Optional. Default value 0 (no warranty). Integer between 0 and 255. | warranty=24 |
| barcode |  | Product barcode identifier (EAN, UPC, ISBN, GTIN). Please use the supplier barcode, not your internal barcodes. | Optional. No default value. Array of strings between 1 and 20 characters long. | barcode=Array('barcode1', 'barcode2') |
| attachments |  | Product attachments data. | Optional. List of arrays. |  |
|  | id | Vendor attachment internal id. | Optional. Integer value between 1 and 4294967295. | id=123 |
|  | url | Vendor attachment URL. | Required. String between 1 and 1024 characters. Valid URL to document. | url=”http://valid-url” |
| status |  | Vendor offer status.  1 – status active  0 – status inactive | Required. Integer value, 1 or 0. | status=1 |
| sale\_price |  | Vendor offer sale price without VAT | Required. Decimal value between 0 and 999999999. Up to four decimals. | sale\_price=51.6477 |
| recommended\_price |  | Vendor offer recommended retail price before discount, without VAT. If set, the offer will be displayed as promo. | Optional. Decimal value between 0 and 999999999. Up to four decimals. Must be greater than sale\_price. | recommended\_price=51.6477 |
| availability |  | Offer availability array. | Required. List of arrays. | {  0=>{  warehouse\_id=1,  id=3}} |
| availability | warehouse\_id | The id of the warehouse. Use warehouse\_id=1 for only one warehouse. | Required. Integer. | warehouse\_id=1 |
| availability | id | Offer availability id.  2 = limited stock  3 = in stock  5 = out of stock | Required. Integer. | id=3 |
| stock |  | Offer available quantity array. | Optional. 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. If present, stock overwrites availability\_id. | 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 |
| commission |  | Commission array for the specified offer. | Required. Array. |  |
|  | type | The type of commission for a finalized order containing the offer. Possible values are percentage or absolute | Required. String. | type=percentage |
|  | value | The commission without VAT for a finalized order of the containing offer. | Required. Float with 4 decimals. If type= percentage the value should be between 0 and 100. | value=12,1234 |
| 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 * commission * vat\_id * warranty * status * availability   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 |  | Vendor offer VAT rate id. Use /vat/read to display possible values. | Required. Integer. | Ex: vat \_id=1 |

**IMPORTANT:**

* 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.
* 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

|  |
| --- |
| Array(  Array(  "id" => "6050",  "family" => Array(  "id" => 111,  "family\_type\_id" => 97,  "name" => "test\_family "  ),  "category\_id" => "1315",  "part\_number" => "test-part-number",  "name" => "Test name",  "description" => "Test description",  "brand" => "Test brand name",  "images" => Array(  Array(  "display\_type" => "1",  "url" => "http://www.image-url.test"  )  ),  "url" => "http://www.product-url.test",  "status" => "1",  "sale\_price" => "406.45",  "availability" => Array(  Array(  "warehouse\_id" => "1",  "id" => "3"  )  ),  "handling\_time" => Array(  Array(  "warehouse\_id" => "1",  "value" => "2"  )  ),  "stock" => Array(  Array(  "warehouse\_id" => "1",  "value" => "2"  )  ),  "commission" => Array(  "type" => "percentage",  "value" => "8"  ),  "vat\_id" => "1",  "characteristics" => Array(  Array(  "id" => "5213",  "value" => "Characteristic 5213 value"  ),  Array(  "id" => "1339",  "value" => "Characteristic 1339 1st value"  ),  Array(  "id" => "1339",  "value" => " Characteristic 1339 2nd value"  )  )  )  ) |

## Updating existing offer

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
* commission
* availability\_id (*we strongly suggest also sending numerical stock if available*)
* handling\_time

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”.

## 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. 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.

If you send the part\_number\_key the keys category\_id, brand, name, part\_number become optional.

## Warehouse read

You can have multiple warehouses the products are shipped from. For defining these warehouses, you have to contact eMAG Marketplace and we will define the warehouses for you. The resource is ***warehouse*** and the available action is ***read***.

The following fields are available for warehouse:

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| id | The warehouse with this id | Integer value between 1 and 4294967295. |
| name | The warehouse name | String of length between 0 and 60 |
| contact | The warehouse contact person name | Required. String value between 1 and 255 |
| phone\_1 | The warehouse first phone number | Required. String value between 8 and 11 digits (only '+' character is allowed at the beginning of the string) |
| phone\_2 | The warehouse second phone number | Optional. String value between 8 and 11 digits (only '+' character is allowed at the beginning of the string) |
| legal\_entity | If the warehouse is legal entity (applicable only to receiver) | LEGAL\_ENTITY\_NO = 0  LEGAL\_ENTITY\_YES = 1 |
| email | The warehouse email address | Optional. Valid email address |
| locality\_id | The warehouse's locality\_id | Required. Integer value between 1 and 4294967295.  Must be a valid locality in the eMAG database. |
| street | The warehouse's street | Required. String value between 3 and 255 |
| zipcode | The warehouse's zipcode | Optional. String value between 1 and 255 |

# 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** | **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. The possible values are:  0 - cancelled  1 - new  2 - in progress  3 – prepared  4 - finalized  5 - returned | Required. Integer value between 0 and 5. | status=1 |
| payment\_mode\_id | The order payment method. The possible values are:  1 - COD (cash on delivery)  2 - bank transfer  3 - online card payment | Required. Integer. | payment\_mode\_id=1 |
| observation | Additional customer notes. | Optional. Text. | observation=”delivery after 6 PM” |
| date | The cart submission timestamp. | Optional. Text in YYYY-mm-dd HH:ii:ss format. | date=”1970-01-01 23:59:59” |
| payment\_status | The online payment status. Only used for online payment methods. The possible values are:  0 - not paid  1 - paid | Required only for online payment methods. Integer. | payment\_status=0 |
| shipping\_tax | The shipment tax value. | Optional. Decimal. | shipping\_tax=”19.99” |
| details | A list with additional order details that are not standard. | Optional. List. |  |
| 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. |
| vouchers | A list describing the voucher discounts. | List. |  |
| invoices | A list describing vendor invoice details. Vendors may only save invoice details. | List. |  |

### Product field in order details

| **Key – level 1** | **Key – level 2** | **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 |  | Vendor internal product id. This is the primary key for identifying a product offer. | Optional. Integer. | product\_id=3331 |
| status |  | The status of product of the order. The possible values are: 0 - cancelled 1 - active | Required. Integer. | status=1 |
| part\_number |  | Manufacturer unique identifier for the product. | Optional. String between 1 and 128 characters. | 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' |
| attachments |  | Product attachments data. | Optional. List of arrays. |  |
|  | id | Attachment id. | Optional. Integer value between 1 and 4294967295. | id=123 |
|  | url | Attachment URL. | Required. String between 1 and 1024 characters. Valid URL to document. | url=”<http://valid-url>” |
| currency |  | Product price currency. | Optional. String. | currency='RON' |
| quantity |  | Product quantity. | Integer. | 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. The possible values are: 0 - cancelled 1 - active | Required. Integer. | status=1 |
| quantity |  | Product quantity | Integer. | quantity=2 |
| sale\_price |  | The sale price without VAT. | Required. Integer. | sale\_price=12.1234 |
| details |  | Additional product notes. | Optional. Text. | details=”text” |

### 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. The possible values are:  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. The possible values are:  0 - private entity;  1 - legal entity. | Optional. Integer value. | legal\_entity=1 |
| is\_vat\_payer | A flag indicating it the customer is vat payer. The possible values are>  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\_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\_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\_country | The customer's shipping country. | Optional. Text. | shipping\_country=”Romania” |
| shipping\_suburb | The customer's shipping county. | Optional. Text. | shipping\_suburb=”Suburb” |
| shipping\_city | The customer's shipping city. | Optional. Text. | shipping\_city=”City name” |
| shipping\_street | The customer's shipping suburb. | Optional. Text. | shipping\_street=”Street name” |
| shipping\_postal\_code | The customer's shipping postal code. | Optional. Text. | shipping\_postal\_code=”23125” |
| billing\_locality\_id | This field uniquely identifies a locality in the eMAG database.  It represents the billing locality. | Integer value between 1 and 4294967295. | billing\_locality\_id=”23” |
| shipping\_locality\_id | This field uniquely identifies a locality in the eMAG database.  It represents the shipping locality. | Integer value between 1 and 4294967295. | shipping\_locality\_id=”23” |

### Order invoices

When pushing orders into finalized status, you MUST always send the invoice data for the order. An order cannot be finalized if no invoice is sent.

An item in the invoice list has the following properties:

| **Key** | **Description** | **Constraints** | **Example** |
| --- | --- | --- | --- |
| id | A value that uniquely identifies an invoice per vendor. | Optional. Integer value between 1 and 16777215. | id=1234 |
| series | The invoice series. | Required. Text. | series=”EMG” |
| number | The invoice number. | Required. Integer value between 1 and 16777215. | number=123456 |
| date | The date when the invoice was created. | Required. Text in YYYY-mm-dd HH:ii:ss format. | date=”1970-01-01 23:59:59” |
| due\_date | The invoice due date. | Required. Text in YYYY-mm-dd HH:ii:ss format. | due\_date=”1970-01-01 23:59:59” |
| net\_value | The total invoice value without VAT. | Required. Decimal value between 0 and 999999999. Up to four decimals. | net\_value=123.1234 |
| gross\_value | The total invoice value including VAT. | Required. Decimal value between 0 and 999999999. Up to four decimals. | gross\_value=123.1234 |
| products | A list describing the invoice lines. | Optional. List. Described below. |  |
| details | Invoice notes. | Optional. Text. | details=”text” |
| type | Invoice type. Invoices can be normal or storno invoices for returned products. | Optional. Integer. Possible values are 0 – Normal invoice  1 – Storno/Cancellation invoice  Default is “normal”. | type=”storno” |

**IMPORTANT:** The final contents of the order (order value, product quantity and prices, vouchers, shipping tax will be generated into an invoice that will be available in the customer’s order history).

Product details in invoice field are detailed below

| **Key** |  | **Description** | **Constraints** | **Example** |
| --- | --- | --- | --- | --- |
| id |  | Vendor internal product id. This is the primary key for identifying a product offer. | Optional. Integer value between 1 and 16777215. id=243409 | id=123 |
| invoice\_position |  | The product position on the invoice. | Optional. Integer. | invoice\_position=1 |
| quantity |  | Product quantity | Required. Integer. | quantity=2 |
| sale\_price |  | The sale price without VAT per unit. | Required. Integer. | sale\_price=12.1234 |
| vat |  | VAT amount, absolute value, per unit. | Required. Decimal value. | vat=12.34 |
| details |  | Product notes in invoice. | Optional. Text. | details=”text” |

## 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/order/acknowledge/[orderId]. This stops the order notification system for the mentioned order. Unless acknowledged, we will notify the new orders for up to 48 hours.

**IMPORTANT**:

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

## Order status matrix

The following matrix defines the order flux 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 | No | Yes | Yes | Yes | No |
| 3 - prepared | No | No | No | Yes | Yes | No |
| 4 - finalized | No | No | No/24h | No | No/24h | Yes/30 days |
| 0 - canceled | No | No | Yes | No | No | No |
| 5 - returned | No | No | No | No | No | No |

**IMPORTANT**:

* You can only edit the order 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 24 hours since finalization.
* Order status “finalized” will be set automatically when issuing the first AWB for that order. See chapter [Saving AWB’s](#_Saving_AWB's).
* The order status “returned” is set automatically when all the products from the initial invoice are canceled. 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 filters 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. |

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. |

## Updating orders

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

**Example (order update with bills):**

|  |
| --- |
| Array (  0 =>  Array (  'status' => 3,  'date' => '2014-07-24 12:16:47',  'observation' => NULL,  'id' => 41200,  'payment\_mode\_id' => 1,  'payment\_status' => 0,  'vendor\_name' => 'vendor1',  'customer' =>  Array (  'id' => 12556,  'name' => 'Name Surname',  'company' => 'Company name',  'gender' => 'M',  'code' => 128312xxxx212,  'email' => '',  'created' => '2014-07-24 12:17:20',  'modified' => '2014-07-24 12:17:32',  'bank' => '',  'iban' => '',  'fax' => '',  'mkt\_id' => 312556,  'phone\_1' => '0724xxxx32',  'phone\_2' => '',  'phone\_3' => '',  'registration\_number' => '',  'billing\_country' => 'RO',  'billing\_suburb' => 'Suburb',  'billing\_city' => 'City Name',  'billing\_locality\_id' => 3,  'billing\_street' => 'Street,  'billing\_postal\_code' => '',  'shipping\_country' => 'RO',  'shipping\_suburb' => 'Suburb',  'shipping\_city' => 'City',  'shipping\_locality\_id' => 3,  'shipping\_street' => 'Street',  'shipping\_postal\_code' => '',  'is\_vat\_payer' => 1,  'legal\_entity' => 0,  ),  'details' =>  Array (  ),  'attachments' =>  Array (  ),  'products' =>  Array (  0 =>  Array (  'id' => 82407,  'product\_id' => 1264,  'part\_number' => '68133',  'quantity' => 1,  'sale\_price' => '967.6613',  'currency' => 'RON',  'created' => '2014-07-24 12:16:50',  'modified' => '2014-07-24 12:17:32',  'status' => 1,  'attachments' =>  Array (  ),  'details' =>  Array (  ),  'vat' => '0.2400',  ),  ),  'shipping\_tax' => '0.0000',  'vouchers' =>  Array (  ),  'proforms' =>  Array (  ),  'invoices' =>  Array (  0 =>  Array (  'id' => '3432',  'series' => 'SER',  'number' => '4103',  'date' => '2014-07-25 08:59:48',  'due\_date' => '2014-07-25 08:59:48',  'net\_value' => 967.6613,  'gross\_value' => '1199.9000',  'products' =>  Array (  0 =>  Array (  'id' => '1264',  'quantity' => '1.0000',  'sale\_price' => 967.6613,  'vat' => 0.24,  'details' => '',  ),  ),  'details' => '',  'type' => 0,  ),  1 =>  Array (  'id' => '3433',  'series' => 'SER2',  'number' => '36841034',  'date' => '2014-07-25 08:59:48',  'due\_date' => '2014-07-25 08:59:48',  'net\_value' => 967.6613,  'gross\_value' => '1199.9000',  'products' =>  Array (  0 =>  Array (  'id' => '1264',  'quantity' => '1.0000',  'sale\_price' => 967.6613,  'vat' => 0.24,  'details' => '',  ),  ),  'details' => '',  'type' => 0,  ),  ),  ),  ) |

### 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.

### Adding products to an existing 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

A finalized order cannot be modified; it can have some of the products returned or fully returned.

In the event an entire order is returned, a call must be made with the corresponding storno invoice that has all the products and their negative quantities in the order. The order will automatically change status from finalized (4) to returned (5).

If some of the products are returned, a call must be made with the corresponding storno invoice that has the returned products and their negative quantities in the order. You can send more than one storno invoice per order, but only within the maximum return timeframe allowed to the customer.

If storno invoices are sent for all the products and their corresponding quantities in the order the order will automatically change it’s status from finalized (4) to returned (5).

# Shipping eMAG Marketplace orders

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

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

## Saving AWB's

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 vendor. |
| sender | \*Array explained [below](#_AWB_sender/receiver_fields) |  |
| receiver | \*Array explained [below](#_AWB_sender/receiver_fields) |  |
| 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 vendor’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). | 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. |
|  |  |  |

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 |
| 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 |

## Reading AWB PDF files

To read an existing AWB PDF simply access the URL below and replace code, username with proper values and hash value with sha1 value of password:

MARKETPLACE\_URL/awb/read\_pdf?emag\_id=awb\_id&code=my\_**code**&**username**=my\_username&hash=**a28bb826824c30ffddd47e9dce41dfb6c39bd655**

## 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 |
| 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. |

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 |
| 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 |

## Reading AWB

The following filters are available when reading AWBs:

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

# Using the messaging service

eMAG Marketplace operators communicate with vendors through messages. The messages are organized in topics. The first message is also the topic while the following messages are considered replies.

## Reading messages

A message has the following properties:

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| id | A value that uniquely identifies a message in the system. | Optional. Integer value between 1 and4294967295. |
| message\_id | A value that uniquely identifies a message in the vendor system. | Optional. Integer value between 1 and4294967295. |
| order\_id | An order id. If supplied, the messages are associated with the order identified by this value. | Optional. Integer value between 1 and4294967295. |
| text | The message body. | Required. Text. |
| created | The message date. | Required. Text in YYYY-mm-dd HH:ii:ss format. |
| author | The name of the user that created the message. | Optional. Text. |
| source | The message source. The possible values are:  1 – eMAG platform;  2 - vendor | Required. |
| status | The message status. The possible values are:  0 - inactive (deleted);  1 - active;  2 - closed. | Required. |

The following filters are available when reading messages:

| **Key** | **Description** | **Constraints** |
| --- | --- | --- |
| order\_id | Filter only messages associated to this order. | Optional. Integer value between 1 and 4294967295. |
| createdBefore | Only the messages 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 messages created after the specified date. | Optional. Text in YYYY-mm-dd HH:ii:ss format. |

The result is limited to maximum 1000 messages.

Messages can be read if orders status is greater than 1 (new).

## Sending messages

To add a new message the vendor should attach the message to the end of the list and then republish the entire list.