



*“Thank you for choosing TMSearch.ai as your trademark search engine and IP data provider. In this detailed guide, we’ve aimed to make it easy for you to share the instructions with other AI services, so you can integrate with us more quickly and effectively.”*

**TMSearch.ai team**

## TMSearch.ai API Manual

(version 3.02 Date 15.01.2026)

### Before starting

If you're using AWS, Azure, DigitalOcean, or any other shared cloud infrastructure, please note that access may be blocked on our side for security reasons. Please provide your subnet(s) / IP ranges so we can whitelist your specific IP address ranges.

Urgent technical support you can find in our Telegram messenger group: @tmsearchaisupport  
<https://t.me/+11ROFHHizuU5Nmly>

Official Telegram channel:  
<https://t.me/tmsearchai>

## 0. API key and subscriptions.

To obtain your personal API key and set up an individual subscription, please contact our sales team on WhatsApp +972-54-937-3974 or <https://tmsearch.ai/contacts.html>.

**All data returned by our trademark API is provided in UTF-8 encoding and in valid JSON format.**

Your subscription details will be included in each API response and may contain the following information:

===== Subscription information =====

```
..., "plan": {                                     #information about your #subscriptin – limits, expiration
  "credits": 9999911,                               #TOTAL count of credits
  "search_credits": 12345,                           #count of SEARCH credits if limited
  "info_credits": 54342,                             #count of INFO credits if limited
  "ETA_DHMS": "27013:13:50:47",                      #days to expiration of your    #subscriprion
  "date_finish": "2099-12-31 23:59:59"               #expiry date
},...
```

===== Subscription information =====

**Important:** Parameters “mid”, “owner\_id”, “attorney\_id” can be used only together with country/office char2 codename value.

**Note:** Throughout this manual, we use “TESTAPIKEY” as the “api\_key” value to demonstrate the API’s capabilities. Please be aware that results returned with this test key differ slightly from the live database — for example, the first 10 most relevant search results may be removed, and dates, classes, and numbers may be modified. We also reserve the right to make additional changes to the test-key output without prior notice. Access to live database is provided on a payment basis or as a one-week trial. Please contact the sales team for details.

URL: <https://tmsearch.ai/api/plan/>

Test URL: [https://tmsearch.ai/api/plan/?api\\_key=TESTAPIKEY](https://tmsearch.ai/api/plan/?api_key=TESTAPIKEY)

===== Only subscription plan information request =====

curl -X POST \

'https://tmsearch.ai/api/plan/' \

--data-urlencode 'api\_key=TESTAPIKEY'

===== Only subscription plan information request =====

You can use this function at any time. It does not deduct from your credits and remains available even after your subscription expires.

## 1. Search request (SEARCH)

URL: <https://tmsearch.ai/api/search/>

Test-URL: [https://tmsearch.ai/api/search/?keyword=ddd&api\\_key=TESTAPIKEY](https://tmsearch.ai/api/search/?keyword=ddd&api_key=TESTAPIKEY)

===== Example search request using CURL =====

```
curl -X POST \
```

```
'https://tmsearch.ai/api/search/' \
```

```
--data-urlencode 'keyword=ddd' \
```

```
--data-urlencode 'api key=TESTAPIKEY'
```

===== Sample search request using CURL =====

API TM search requests provide the same functionality as the search on [tmsearch.ai](https://tmsearch.ai) (all filters are powered on). For details, please refer to <https://tmsearch.ai/trademark/search.html> .

NOTE: Every request to the live database described in this section downscales your “credits” and/or “search credits” subscription values.

The requests are sent in GET-mode. To set up the system for the needs of your company individually please provide the IP addresses that will be used from your side. We will tune the system accordingly so there will be no need to provide the API key in every request.

The main request parameter is "keyword" – the combination of letters and/or numbers that will be searched in the database. If you don't fill it, there will be zero results. In the following example - the keyword is "ddd".

**Format:** letters (Latin, Cyrillic, Hebrew, and other alphabets) and numbers are allowed.

**Maximum length:** 256 characters.

The request returns a response in **text/JSON** format.

===== Example out of search request =====

{	
"timestamp": "2026-01-15 10:09:12",	#current timestamp (UTC)
"plan": {	#information about your #subscription – limits, expiration
"credits": 99999911,	#count of credits
"ETA_DHMS": "27013:13:50:47",	#days to expiration of your      #subscription
"date_finish": "2099-12-31 23:59:59"	#day of expiration
},	
"total": 390,	#count of search results
"result": [	#search results array
{	
"mid": "3815369",	#internal id number, integer
"class": [	#array of NICE classes, sorted                  #
"09",	ascending
"13",	#class number, char(2), 01..45
"45"	#class number, char(2) , 01..45
],	
"status": "DEAD",	#global status of application #LIVE, DEAD, UNKN – if we not #recognize a status correctly or #we can't do it

```

"submission": "US",           #office of application submission, #char(2) codename
"app": "76470459",           #application number in office,
                               #type – text, because not all #offices using
                               #only digits numbers

"accuracy": "99",             #level of accuracy, integer
"img": "US/TM/APP/3815369.jpg", #link for image file (prototype), #text
"date": {                     #different dates of application
  "expiration": "20040512",    #date of application expiration, #YYYYMMDD format
  "applied": "20021127"        #date of apply to submission office
                               #YYYYMMDD format

  "granted": "20030321"        #when application was granted,
                               #YYYYMMDD format
},
"verbal": "DDD",              #verbal element of application
"protection": [               #array of protection countries
  "AR","IT","US"              #country code names, char(2) code
],
"reg": "512823",              #registration number, text
},
...
]
}

```

===== Sample output of the response to a search request =====

On the upper level of the JSON-structure we have the following elements: "total" and "result".

**"total"** - shows the quantity of the results received in the response. This parameter helps to control the accuracy of the download results. Field format: integer, greater than zero.

Section **"result"** is an array of the trademark application items (structures).

Every item has the following keys:

**"mid"** - is the ancillary parameter of the trademark application. Currently this parameter is used in the image URL, and it helps to receive all the TM information promptly. In the next versions of API, it will have more powerful functionality. Field format: int8

**"verbal"** - is a verbal element of the trademark. If the trademark does not have an image - this text is used for representation. Field format: text.

**"img"** - provides an URL of a trademark's JPG image. E.g.: <https://img.tmsearch.ai/img/210/> + img

Where 210 - is the width or height (what is bigger) in pixels of the image requested. Now 3 image sizes are available: 210, 500, 700. You can use any – but 210 is the fastest. Thus, it is the most relevant for multiple image requests.

**"status"** - this field is limited to 3 values: LIVE, DEAD, UNKN.

- LIVE - trademark is protected or submitted.
- DEAD - protection has expired or has been terminated for another reason.
- UNKN – the information received from TM registering offices cannot be interpreted by our system.

**"class"** - this parameter contains an array of NICE classification. ASC sorted by default. Field format: array of int2 elements.

**"submission"** – provides the information about the registrar in which the trademark has been submitted (e.g.: WO, EU, UK, ES, IT, RU, etc.) Field format: array of char2 elements.

For complete list of offices/countries codes please see:

<https://tmsearch.ai/FAQ/countries.txt>

**“protection”** – provides the information about the region in which the trademark is/has been protected. If the TM is registered by a national registrar, for example TR (Turkey) - there will be a single protection area - TR. But if the TM has been submitted to an international registrar, e.g.: WO (WIPO), it will be protected in multiple areas – as shown in the example below: Field format: Array of char2 elements.

===== Important: protection and submission =====

```
...,
"protection": [                                #array of protection countries
    "AT","BX","CN","RU","US"                  #country code names, char(2) code
],
"submission":"WO",                             #office of application, char(2) codename
...
```

===== Important: protection and submission =====

Important note:

WIPO can also protect a trademark in regions like EU (European Union), BX (Benelux), EMEA (Europe, Middle East and Africa) etc.

The JSON file provided never bears these abbreviations' decoding. Your system should "understand" these abbreviations itself.

Benelux (BX) consists next offices/countries: LU, BE, NL

Here is the list of EU members: AT, BG, CY, CZ, DE, DK, EE, ES, FI, FR, GR, HR, HU, IE, IT, LT, LV, MT, NL, PL, PT, RO, SE, ST, SK.

UK is not EU member after Brexit.

When we add other multinational offices, information about their members will be added to this document.

**Additional note:** the countries that have ceased to exist. YU (Yugoslavia), SU (Soviet Union), etc.

They can be visible at WIPO from time to time. But for most of them (about 99%) the trademarks' status is DEAD

**“app”** - provides the application's number. Field format: text field. Max length - 256

**“reg”** - provides the trademark registration's number. Field format: text field. Max length - 256

Section **“date”** - contains a hash consisting of 3 values - YYYYMMDD. Values can be absent.

**“applied”** – provides the information about the date when the trademark application was added to TM registering office.

**“granted”** - provides the information about the date when the trademark application was registered in the TM registering office. This field can be empty. Even though many TM registering offices provide this information, some of them don't.

**“expiration”** – expiration date of a TM registration. If the expiration date is in future – the trademark has the status “LIVE”. If it is in the past – the trademark status is “DEAD”.

**“accuracy”** – the similarity level of the trademark's verbal element to the keyword entered in the GET request. The maximum value is 99, which means full coincidence. The current range of values available is set by default at 80%-99%. The option to downgrade this parameter is available on request. Field format: numeric. 0..99/100.

All the trademarks shown in the “result” array are sorted in DESC order by the “accuracy” pointer.

Important: all the information about single trademark is shown on the second level.

## 2. Get single trademark data (INFO)

URL: <https://tmsearch.ai/api/info/>

Test-URL samples:

[https://tmsearch.ai/api/info/?number=985954&type=REG&office=WO&api\\_key=TESTAPIKEY](https://tmsearch.ai/api/info/?number=985954&type=REG&office=WO&api_key=TESTAPIKEY)  
[https://tmsearch.ai/api/info/?number=83079139&type=APP&office=TR&api\\_key=TESTAPIKEY](https://tmsearch.ai/api/info/?number=83079139&type=APP&office=TR&api_key=TESTAPIKEY)

===== Sample of INFO request using CURL =====

```
curl -X POST \
  'https://tmsearch.ai/api/info/' \
  --data-urlencode 'number=985954' \
  --data-urlencode 'type=REG' \
  --data-urlencode 'office=WO' \
  --data-urlencode 'api_key=TESTAPIKEY'
```

===== Sample of INFO request using CURL =====

NOTE: Every request to the live database described in this section downscales your “credits” and/or “info\_credits” subscription values.

The following request parameters are available:

**“number”** (text) – the id of application or registration. If you are not sure how to fill it correctly, please use the search on the main page of our website [tmsearch.ai](https://tmsearch.ai). This field may contain both: letters and digits. Sometimes it may start with zeros like in the UK office. In any case it is important to use all the characters.

**“Type”** - here are only two possible values – APP or REG. APP – means that application number has been used, REG – registration number has been used. APP is the default value of this parameter. *Please note that for different countries these parameters vary. For example, for Turkey only APP can be applied, while for the World Office REG is applicable.*

**“Office”** (\*) – mandatory field. Need to insert two letter country code of office like in the upper level. Field format: Latin letters. Char2

**“mid”** (numeric) is an alternative parameter for “number”. This is an internal number used by our system. You can see it in the results of the upper-level requests.

(\*) – must have

Every request must have one of the following parameters: either parameter **“mid”** or **“number”**.

Response consists of a JSON structure similar to the upper-level response. Additionally, it includes:

- class information (including subclasses),
- owner information,
- attorney information,
- information renewal date, etc.

===== Sample output of INFO request =====

```
{
```

```

"office": "WO",           #office of application (like submission), char2
"submission": "WO",      #submission office of application, char2
"status": "LIVE",        #Status of application LIVE, DEAD, UNKN
"owner": "GOLD & GOLD S.r.l., Via Righi, --, I-50019 SESTO FIORENTINO (Firenze), (IT)",
                        #Owner of application, text field
"owner_id": 775978,      #owner ID in our system, used for 4th section of this manual
"attorney": "International institute of IP Right, Berlin",      #Owner of application, text field
                        #attorney of application, text field
"attorney_id": 17598,    #attorney ID in our system, used for 4th section of this manual
                        "timestamp": "2026-01-15 14:38:11",    #current timestamp UTC
"app": 834139,           #application number, text
"item_class": [          #section of addition class information, array
{
    "name": "18",         #NICE class number, int2, length - 2 symbols
    "more": {             #additional information about class
        "lang": "EN",     #language of subclasses
        "name_more": [    #array of subclasses
            "Leather and imitations thereof, animal hides and leather articles"
                        #subclass – text element
        ]
    }
},...
],
"class": [               #array of simple NICE classes, int2, element length is 2
    "18",
    "24",
    "25"
],
"number": 770627,
"type": "REG",
"verbal": "gold&gold",
"reg": 9974,
"plan": {               #information about yours subscription
    "ETA_DHMS": "27013:09:21:48",
    "credits": 9999900,
    "date_finish": "2099-12-31 23:59:59"
},
"date": {               #Dates section
    "applied": "20080924", #When application was applied YYYYMMDD
    "granted": "20080924", #When application was granted YYYYMMDD
    "expiration": "20180924" #Expiration of application YYYYMMDD
},
"url":
"https://www3.wipo.int/madrid/monitor/en/showData.jsp?ID=ROM.985954",
                        #url is URL links for external original data viewing
"protection": [         #offices/countries where application is protecting
    "CN","IL","US"      #array of char2, codenames
],
"img": "WO/TM/APP/1288788.jpg", #link to application image
"mid": "1288788"        #our system uncial ID in office
}
===== Example output of INFOrequest =====

```

### 3. Search by owners and attorneys (PERSON)

URL: <https://tmsearch.ai/api/person/>

Test URL sample:

[https://tmsearch.ai/api/person/?keyword=ddd&office=US&api\\_key=TESTAPIKEY](https://tmsearch.ai/api/person/?keyword=ddd&office=US&api_key=TESTAPIKEY)

===== Sample of PERSON request using CURL =====

```
curl -X POST \
  'https://tmsearch.ai/api/person/' \
  --data-urlencode 'keyword=ddd' \
  --data-urlencode 'office=US' \
  --data-urlencode 'api_key=TESTAPIKEY'
```

===== Sample of PERSON request using CURL =====

This section introduces to you information on how to find a company by a keyword in its name, and about trademarks belonging to this company if it is an owner or about trademarks registered by this company if it is an attorney.

NOTE: Every request to the live database described in this section downscales your “credits” or “search\_credits” subscription values.

===== Sample output of PERSON request =====

[..., #it is array of next structures

```
{
  "owner": 12,    #count of applications that on this company ownership
  "attorney": 5,  #count of applications that on this company management like IP attorney
  "sum": 17,      #sum of all applications owner + attorney
  "mid": "253031", #our unique ID in this office
  "name": "DDD DESARROLLISTAS S.A.", #original name of company or person
  "country": "AR", #in which office there are working
  "i": 13,        #position number in this result listing
  "merged": null  #if we merge this company with another “mid” – we will inform you about
  this in this field
},...
]
```

===== Sample output of PERSON request =====

## 4. Viewing of all owner's or attorney's applications.

URL: <https://tmsearch.ai/api/firm/>

Test URL sample:

[https://tmsearch.ai/api/firm/?firm\\_id=978&office=RU&api\\_key=TESTAPIKEY&attorney=1&page=3](https://tmsearch.ai/api/firm/?firm_id=978&office=RU&api_key=TESTAPIKEY&attorney=1&page=3)

===== Sample of CURL for getting all application from company =====

```
curl -X POST \
  'https://tmsearch.ai/api/firm/ ' \
  --data-urlencode 'firm_id=978' \
  --data-urlencode 'office=RU' \
  --data-urlencode 'api_key=TESTAPIKEY' \
  --data-urlencode 'attorney=1' \
  --data-urlencode 'page=3'
```

===== Sample of CURL for getting all application from company =====

NOTE: Every request to the live database described in this section downscales your “credits” or “search\_credits” subscription values multiplied by the number of pages in the response. Each page of the response counts. Thus, we recommend making more detailed requests with additional parameters to have reasonable number of results in the response.

To get data about application that managed by a company (attorney role) you need to add 'attorney=1', if you need only applications that in a company's ownership – no need to specify this parameter or the value must be equal zero, 'attorney=0'.

Mandatory parameters are:

1. “**firm\_id**” – you can get it from the following fields:
  - a. “**mid**” (3<sup>rd</sup> section) or
  - b. “**owner\_id**” (2<sup>nd</sup> section) or
  - c. “**attorney\_id**” (2<sup>nd</sup> section).
2. Parameter “**office**” – must have, char2 codename.

Parameter “**page**” – first page is “0”, it shows first 500 results or less if not have 500.

If in your request “**page=1**” – it means that you will receive results from 501 to 1000 etc.

Response layout is equal to response structure of 1<sup>st</sup> section of this manual. “**count**” shows total number of results in the response. To get them all you need to use pagination. Trademarks in these listings sorted by “**mid**” ascending.

===== Sample of Parameter 'page' in response of FIRM =====

```
{
  "page": "3", #number of page in this response (500 results on response)
  "count": 23462, #totally counter of results
  "results": [ ... ], #in results in elements like in 1st section of manual
  ...
}
```

===== Sample of Parameter 'page' in response of FIRM =====

**IMPORTANT!** For security reasons, please use POST requests to hide api\_key value!