# API – upper level (search requests)

URL: https://tmsearch.ai/api/search/  
Test-URL: https://tmsearch.ai/api/search/?keyword=ddd&api\_key= TESTAPIKEY

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

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.

## How it works

The main request parameter is "keyword" – the letter combination that will be searched in the database. If you don't fill it, there will be 0 results.

In the following example - keyword is "ddd".

The request returns the following response in text/json

  
  
  
On the upper level of the array there are two elements - "total" and "result".

"total" - shows the quantity of the results received under the request fulfilled. This parameter allows to control if the results have been downloaded correctly or not.

The next parameter - "result" contains the array of the trademark application items.

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.

“verbal” - is the verbal element of the trademark. If the trademark does not have an image - this text is used for representation.

“img” - provides an URL of a trademark’s JPG image. E.g.: https://img.tmsearch.ai/img/210/ + img

Where 210 - is the size in MB 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.

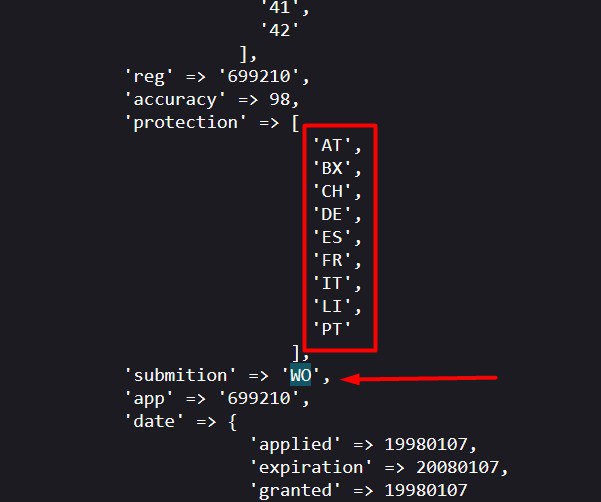
“status” - this field can receive 3 values: LIVE, DEAD, UNKN.

* 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 key contains an array of NICE classification. They are sorted ASC by default.

“submission” – provides the information about the registrar in which the trademark has been submitted (e.g.: WO, EU, UK, ES, IT, RU, etc.)

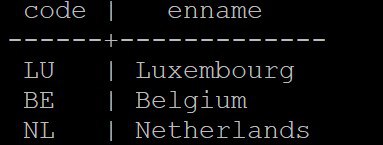
“protection” – provides the information about the region in which the trademark is/has been protected. If the TM is registered by a national registrar, for example TR (Turkey) - there will be a single protection area - TR. But if the TM has been submitted to an international registrar, e.g.: WO (WIPO),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

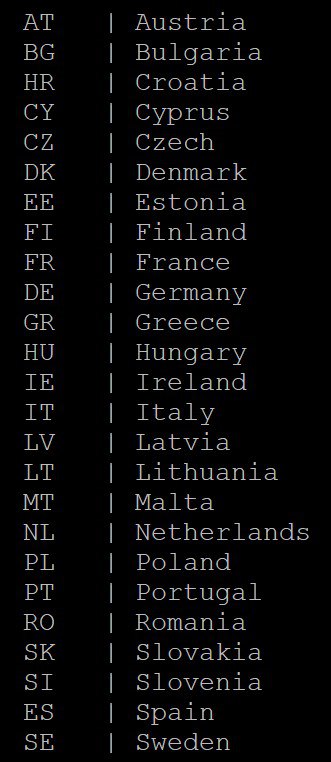


Important note:

WIPO can protect a TM in the 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.

  
  
  
Here is the table of EU members sorted ASC



To finalize the observation of TM registering offices it is important to mention 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.

“reg” - provides the trademark registration’s number.

“date” - contains a hash consisting of 3 values - day, month, year. Value 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, because a lot of TM registering offices provide this information, but some of them don't.

“expiration” – the date TM registration expiration. 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 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 pointer is available on request.

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

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

# API – second level (get single trademark data)

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

Test-URL: <https://tmsearch.ai/api/info/?number=1580418&type=REG&office=WO&api_key=TESTAPIKEY>

The following request parameters are available:  
**Number** – 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](http://www.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**. There 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.

**Office** (\*) – mandatory field. Need to insert two letter country code of office like in the upper level.

**Mid** (numeric) is an alternative parameter for “number”. This is an internal number use by our system. You can get it in response of the upper-level requests.

(\*) – must have

Every request must have one of the following parameters: either **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.

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