Noteb.com API documentation

The API can be access at this address:
https://noteb.com/api/webservice.php

General access information

In order to access the API it is mandatory to provide:

- a valid API key
- a valid request
- not have exhausted the number of daily hits allocated

For testing purposes, we provide a public API key: 112233aabbcc
The public API key only allows a maximum of 500 requests each day.

To receive an individual API key, please contact us at office@starchaser.ro and we will provide an API key free of charge with 500 requests per day. Depending on your use case scenario, we can increase the number of requests per day free of charge or for a small fee.

Current valid requests:

- get_model_info
- get_model_info_all
- list_models
- get_conf_info
- get_exact_conf_info

How to use the API

The provided web service will respond only to POST requests. The POST requests has to provide individual parameters for each element and must follow the guidelines of the programming language from which the POST requests originates. **The web service does NOT accept string requests structured in JSON format. The requests must be in the standard FORM-DATA format.**

A simple POST request will look like this:

```
{
        ["apikey"]=> "112233aabbcc",
        ["method"]=> "get_model_info",
        ["param"]=> {  }
}
```

And in code example:

Curl

```
curl --data "apikey=112233aabbcc&method=get_model_info&param[model_id]=1175"
https://noteb.com/api/webservice.php
```

Python (you will need the “requests” modules)

```
>>> import requests
>>> r = requests.post('https://noteb.com/api/webservice.php', data={'apikey': "112233aabbcc",
'method': "get_model_info", 'param[model_id]': 1175})
>>> r.text
```

The service will reply like this:

```
{
  "code":26,
  "message":"Valid method.",
  "result":{
  },
  "daily_hits_left":"818"
}
```

All requests have some required parameters and some optional parameters. The parameters define what information is retrieved. The parameters for each requests are detailed in the documentation of each request.

The reply from the server is in **JSON** format and can be parsed as an object. The reply will always include:

- a code – which is for debugging purposes.
- a message – which offers information regarding the request and possible issues with it
- a result – which contains the information that was requested only if the web service was successful in retrieving it.
- A daily hits left element – which indicates how many more requests are available for the provided API key

If the method receives the correct parameters, it will reply with the appropriate information in the „result" parameter. If the parameters are incorrect, the reply from the web service will usually include information regarding the incorrect parameters in the „message" part of the response. The web service will still try to complete the request even with incorrect parameters, but cannot guarantee that it will always be successful.

Functions

1. get_model_info

Parameters

Required parameters

model_id OR model_name

If the id given by noteb.com to a particular laptop model is known, then it can be used directly to request information on that model. Multiple ids can also be used in the request, delimited by the "," character, Ex. "54,838,1123", the function will still return information for just one of the ids, but it will search for the first matching model within the id list with the specified optional parameters.

In case the id of the model is not known, the function will try to search for the model by its name. The name of the model does not need to be exact, but cannot be very generic either. If the provided name is too generic and more than 7 laptop models match that name, then no model will be retrieved from the database.

If the number of laptops that match that name is less than 7, then the function will retrieve information on the model that best matches the rest of the parameters. If no other parameters are provided, it will retrieve the model with the best value (performance per dollar) of the ones it found.

If a valid model_id is provided the function will ignore the model_name parameter.

We advise for the model name parameter to include the laptop's brand/producer and at least part of the model's name.

Examples:

(Good)
    "Dell 7577" – will have no problem getting the model
    "Clevo P950" – will have no problem getting the model

(Bad)
    "Dell Inspiron" – will not retrieve anything, too many models with that name
     "Lenovo Thinkpad T" – will not retrieve anything, too may models with that name

Example code request

```
{
        ["apikey"]=> "112233aabbcc",
        ["method"]=> "get_model_info",
        ["param"]=>
        {
                ["model_id"]=> "",
                ["model_name"]=> "Dell 7577"
        }
}
```

Reply

```
{
  "code":26,
  "message":"Valid method.",
  "result":{
    "0":{
      "config_id":"5647892694503430848",
      "model_info":{
        "id":1175,
        "noteb_name":"Dell Inspiron 15 7577 Gaming ",
        "name":"Dell Inspiron 15 7577 Gaming",
      },
      "model_resources":{
        "image_1":"http:\/\/86.123.134.36\/notebro\/res\/img\/models\/970_1.jpg",
          .................
        "official_link2":null,
        "launch_date":"2017-09-27"
      },
      "cpu":{
        "prod":"Intel",
        "model":"i5-7300HQ",

          .................
        "rating":"53.5",
        "integrated_video":"Intel HD Graphics 630"
      },        "other_info":"Bluetooth 4.x,dual band,2 x antennas"
      .................
  },
  "daily_hits_left":"805"
}
```

Optional parameters (get_model_info)

Even if a model is identified exactly, either by name or by id, without any further information, the function will retrieve the configuration, which offers the best value for money. However, in certain cases, we may want to retrieve a model's information with specific components.

For example, the Dell 7577 comes with the processor Intel 7300HQ and the processor Intel 7700HQ, but we may only want information for the Dell 7577 with the Intel 7700HQ. This is where the optional parameters come into play.

Possible optional parameters:

- **cpu_name** - string type parameter, can contain the full name or partial name of a processor. If the function cannot identify the component, it will ignore the parameter.
- **gpu_name** - string type parameter, can contain the full name or partial name of a video card. If the function cannot identify the component, it will ignore the parameter.
- **display_res** – string type parameter that should look like this „1920x1080”, the first part before the „x” is the minim desired horizontal resolution and the second part after the „x” is the desired minim vertical resolution.
- **display_type** – string type parameter that can force select a display based on its technology, ex. “IPS” or “TN” or “OLED”.
- **display_size_min** – string type parameter with a valued between 0 and 30, defines the minimum desired screen size in inches.
- **display_size_max** – string type parameter with a valued between 0 and 30, defines the maximum desired screen size in inches.
- **display_srgb** – string type parameter that tries to select only displays with sRGB higher or equal to the specified value. Note that sRGB values are not available for all models. Models without a specified sRGB value will be excluded from the search.
- **min_mem** - string type parameter that specifies the minim RAM memory size in gigabytes. The function will only select configurations with memory higher than the specified value.

- **storage_cap** - string type parameter that specifies the minim storage size in gigabytes. The function will only select configurations with total storage capacity higher than the specified value.
- **first_hdd_min_size** – string type parameter that force the minim size of the primary hard drive to be of a certain value in gigabytes.
- **first_hdd_type** – string type parameter that force the type of the primary hard drive to be of a certain kind, ex: “SSD”, “HDD”, “SSHD”, “EMMC”.
- **second_hdd** – if left blank will not influence the retrieved model information, if anything is specified here then it will only search for configurations with two storage devices.
- **mdb_wan** – accepts values “yes” and “no”, any other value will not influence the retrieved model information. “yes” – retrieves the model information for the versions with WWAN (4G/LTE) support, “no” – retrieves the model information for the versions without WWAN support.
- **mdb_gsync** - accepts only “yes”, any other value will not influence the retrieved model information. “yes” – retrieves the model information for the versions with G-Sync/FreeSync support.
- **mdb_optimus** – accepts only “yes”, any other value will not influence the retrieved model information. “yes” – retrieves the model information for the versions with Optimus/Enduro support.
- **wireless_name** - string type parameter, can contain the full name or partial name of a wireless card. If the function cannot identify the component, it will ignore the parameter.
- **odd_type** - string type parameter, can contain the full name or partial name of the type of optical drive. Ex: “DVD”,”DVD-RW”,”BD-ROM” If the function cannot identify the component, it will ignore the parameter.
- **battery_size** - string type parameter that force the minim size of the total battery capacity to be of a certain value, the value should represent Wh (watts per hour).
- **warranty_years** - string type parameter that force the minim number of warranty years. Value ca be between “1” and “10”
- **warranty_type** - accepts values “standard” and “premium”, any other value will not influence the retrieved model information. “standard” – retrieves the model information only for standard warranties, which are mail-in or back-to-depot/store types of warranty . “premium” retrieves the model information for warranties which are with “on-site” repair.
- **opsist** - string type parameter, can contain the full name or partial name of the desired operating system. Ex. “Windows 10 Pro”, “MacOS 13.13”. If the function cannot identify the operating system, it will ignore the parameter.

Note: If the parameters are way out of range, the function will ignore the parameters.

Example of POST request with optional parameters:

```
{
    ["apikey"]=> "112233aabbcc",
    ["method"]=> "get_model_info",
    ["param"]=>
    {
        ["model_name"]=> "Apple MacBook Pro"
        ["display_res"]=> "2000x1000"
        ["display_type"]=> "IPS"
        ["display_size_max"]=> "14"
```

```
            ["display_srgb"]=> "60"
            ["min_mem"]=> "8"
            ["first_hdd_min_size"]=> "300"
            ["first_hdd_type"]=> "SSD"
            ["mdb_wan"]=> "no"
            ["warranty_years"]=> "1"
        }
    }
```

Understanding the information from the results (get_model_info)

A typical reply from the get_model_info function will look like the below JSON. Most of the information in the reply is self-explanatory. The comments below should help clear any quests.

```
{
   "code": 26,
   "message": "Valid method.",
   "result": {
      "0": {
         "config_id": "9435743435583600640",  -> noteb database id of the configuration
         "model_info": {
            "id": 415,  -> noteb database id of the model
            "noteb_name": "Apple MacBook (2016) ",  -> name as it is on noteb.com
            "name": "Apple MacBook (2016)",  -> a more common laptop name
            "submodel_info": ["Classi."]  -> array of strings that contains the name elements
used by noteb.com to differentiate sub-models by name. Omitted if there are no sub-model identifiers.
         },
         "model_resources": {
            "thumbnail": "https://noteb.com/res/img/models/thumb/t_405_1.jpg",
            "image_1": "https://noteb.com/res/img/models/405_1.jpg",
            "image_2": "https://noteb.com/res/img/models/405_2.jpg",  -> laptop images
            "image_3": "https://noteb.com/res/img/models/405_3.jpg",
            "image_4": "https://noteb.com/res/img/models/405_4.jpg",
            "official_link": "http://www.apple.com/macbook/",  -> official link
            "official_link2": null,
            "launch_date": "2017-01-16"  -> laptop launch date
         },
         "cpu": {
            "prod": "Intel",
            "model": "i5-7Y54",
            "lithography": "14",  -> value is in nm (nanometres)
            "cache": "4",  -> value is in MB (megabytes)
            "base_speed": "1.20",  -> value is in MHz
            "boost_speed": "3.20",  -> value is in MHz
            "cores": "2",  -> number of processor cores
            "tdp": "5",  -> thermal design power, the amount of heat generated by the processor
            "other_info": "SSE4.2,AVX2.0,64-bit,HT,VT-x,VT-d,TBT 2.0",
            "rating": "36.7",  -> performance score , it’s the performance percentage of the most
powerful processor currently available in a laptop
            "integrated_video_id": null, -> not applicable for this function
            "integrated_video": "Intel HD Graphics 615"  -> name of the integrated graphics
         },
         "display": {
            "size": "12.0",  -> value is in inch
            "horizontal_resolution": "2304",  -> value is in pixels
            "vertical_resolution": "1440",  -> value is in pixels
            "type": "LED IPS",
            "sRGB": "80",  -> value is a percentage
            "touch": "no", -> value can “yes” or “no”
            "other_info": "16.7M colors,Brightness: 350nit"
         },
         "memory": {
            "size": "8", -> value is in GB (gigabytes)
            "speed": "1866",-> values is in MHz
            "type": "DDR3"
         },

         "primary_storage": {
            "model": "SSD M.2 GEN3", -> defines the type of storage, “SSD M.2 GEN3” usually
means PCI-e , and GEN2 means SATA.
            "cap": "512", -> value is in GB (gigabytes)
            "rpm": null, -> rotation per minute, only applicable to hard drivers and hybrid drives.
            "read_speed": "1800" -> value is in MB/s , it’s an estimated speed value given the type
of SSD used
         },
         "secondary_storage": { -> see primary storage
            "model": "N/A",
            "cap": "0",
            "rpm": null,
            "read_speed": null
         },
         "gpu": {
            "prod": "Intel",
            "model": "HD Graphics 615",
            "architecture": "Kaby Lake",
            "lithography": "14", -> value is in nm (nanometres)
            "shaders": "24",
            "base_speed": "300", -> value is in MHz
            "boost_speed": "950", -> value is in MHz
            "shader_speed": "300", -> value is in MHz
            "memory_speed": "933", -> value is in MHz
            "memory_bandwidth": "128", -> value is in bit
            "memory_size": "1700", -> value is in MB (megabytes)
            "memory_type": "DDR3",
            "tdp": "0", -> thermal design power, the amount of heat generated by the graphics
card
            "other_info": "HDMI 1.4 (4096x2160 @ 24Hz)",
            "rating": "3.7" -> performance score , it’s the performance percentage of the most
powerful single video card currently available in a laptop

         },
         "wireless_card": {
            "model": "BCM15700A2",
            "speed": "867", -> values is in Mbps (megabits per second)
            "other_info": "Bluetooth 4.0,dual band,2 x antennas"
         },
         "optical_drive": {
            "type": "none",
            "other_info": ""
         },
         "motherboard": {
            "ram_slots": "soldered", -> number of ram slots that can be replaced
            "lan_card": "none",
            "storage_slots": "soldered", -> number of storage connectors available on the
motherboard
            "other_info": "Intel HD audio"
         },
         "chassis": {
            "height_cm": "1.31",
            "height_inch": "0.52",
            "depth_cm": "19.65",
            "depth_inch": "7.74",
            "width_cm": "28.05",
            "width_inch": "11.04",
            "colors": "Rose Gold,Space Gray,Gold,Silver",
            "build_materials": "Aluminium",
            "peripheral_interfaces": "1 X USB 3.1 (Type-C),Thunderbolt",
            "video_interfaces": "",
            "webcam_mp": "0.300", -> value is in megapixels
            "keyboard_type": "Backlit Chiclet keyboard",
            "charger": "29W (2.0A - 14.5VDC) / (2.4A - 5.2V)",
            "other_info": "2 x speakers,Microphone array"
         },
         "battery": {
            "capacity": "41.4", -> values is in Wh (watts per hour)
            "cell_type": "Li-Pol",
            "other_info": ""
         },
         "warranty": {
            "years": "1",
            "type_short": "Standard",
            "type_long": "Standard Pick-up & Return"
         },
         "operating_system": "macOS 10.13",
         "config_score": "37.74", -> it's a laptop score based on the best specifications of other
laptops on the market, it takes into consideration processor and video card performance, battery life,
weight, size, number of ports, almost all laptops characteristics are evaluated and compared to each
other. Maximum theoretical value is 100.
         "config_price": "689", -> average price for this laptop configuration
         "config_price_min": "652", -> predicated minimum price for this configuration
         "config_price_max": "727", -> predicated maximum price for this configuration
         "battery_life_raw": "8.8", -> float value, predicated battery life for continuous video
playback and medium brightness
         "battery_life_hours": "8:48", -> formatted value (hh:mm), predicated battery life for
continuous video playback and medium brightness
         "total_storage_capacity": "512" -> total storage capacity (primary + secondary) in GB
(gigabytes)
      }
   },
   "daily_hits_left": "986"
}
```

Hint: The function returns only one thumbnail address but all uploaded images have a thumbnail. Ex. `"image_3": https://noteb.com/res/img/models/405_3.jpg` will have a thumbnail at the address `https://noteb.com/res/img/models/`**`thumb/t_`**`405_3.jpg",`

2. get_model_info_all

It is very similar to the function get_model_info, but unlike get_model_info, get_model_info_all returns all available components for a requested laptop model and indicates which of the returned components are active in the returned configuration.

Parameters

Exactly the same ones as for the **get_model_info** function. The same mandatory parameters and optional parameters apply to both **get_model_info** and **get_model_info_all**.

Information from the results (get_model_info_all)

The returned information of **get_model_info_all** is similar to that of **get_model_info**, but contains more components and is structured slightly different. Now instead of getting information just for one component, there is a list with all available components for that model and their subsequent information, plus indication as to which of these components are selected in the retrieved configuration.

A typical reply from the get_model_info_all function will look like the below JSON. Here we detail only the main differences compared to get_model_info. To understand it completely please refer to get_model_info for detailed information on each component.

```
{
   "code": 26,
   "message": "Valid method.",
   "result": {
      "0": {
         "model_info": [
            {
               "id": 838, -> noteb model id for this model
               "noteb_name": "Apple MacBook Pro 13 (H12017) Classi.",
            ... -> same as get_model_info, refer to it for more information
            }
         ],
         "config_id": "12244054780192000000",
         "model_resources": {
            "image_1": "http://86.123.134.36/notebro/res/img/models/499_1.jpg",
            ... -> same as get_model_info, refer to it for more information
         },
         "cpu": {
            "374": {  -> noteb id for this cpu
               "prod": "Intel",
               "model": "i5-7360U",
               "lithography": "14",
            ... -> same as cpu section from get_model_info, refer to it for more information
               "integrated_video_id": 0, -> indicates the element number in the gpu object for
the integrated video solution associated with this processor
               "integrated_video": "Intel Iris Plus Graphics 640"
            },
            "375": {  -> noteb id for this cpu
               "prod": "Intel",
               "model": "i7-7660U",
            ... -> same as cpu section from get_model_info, refer to it for more information
               "integrated_video_id": 1, -> indicates the element number in the gpu object for
the integrated video solution associated with this processor
               "integrated_video": "Intel Iris Plus Graphics 640"
            },
            "selected": 374  -> indicates the element in the returned cpu object that is selected in
the retrieved configuration
         },
         "display": {
            "27": {  -> noteb id for this display
               "size": "13.3",
               "horizontal_resolution": "2560",
               "vertical_resolution": "1600",
            ... -> same as display section from get_model_info, refer to it for more information
            },
            "selected": 27  -> indicates the element in the returned display object that is selected
in the retrieved configuration
         },
         "memory": {
            "18": {-> noteb id for this memory configuration
            ... -> same as memory section from get_model_info, refer to it for more information
            },
            "20": {-> noteb id for this memory configuration
          ... -> same as memory section from get_model_info, refer to it for more information              },
            "selected": 18  -> indicates the element in the returned memory object that is selected
in the retrieved configuration
         },
         "primary_storage": {
            "28": {  -> noteb id for this storage
               "model": "SSD M.2 GEN3",
               "cap": "128",
            ... -> same as primary_storage section from get_model_info, refer to it for more information
            },
            "29": {-> noteb id for this storage
               "model": "SSD M.2 GEN3",
            ... -> same as primary_storage section from get_model_info, refer to it for more information
            },
            "30": {-> noteb id for this storage
               "model": "SSD M.2 GEN3",
            ... -> same as primary_storage section from get_model_info, refer to it for more information
            },
            "40": {-> noteb id for this storage
               "model": "SSD M.2 GEN3",
            ... -> same as primary_storage section from get_model_info, refer to it for more information
            },
            "selected": 29  -> indicates the element in the returned primary_storage object that is
selected in the retrieved configuration
         },
         "secondary_storage": {
            "0": {-> noteb id for this storage
               "model": "N/A",
            ... -> same as secondary_storage section from get_model_info, refer to it for more information
            },
            "selected": 0  -> indicates the element in the returned secondary_storage object that is
selected in the retrieved configuration
         },
         "gpu": {
            "499": {  -> noteb id for this gpu
               "prod": "Intel",
               "model": "Iris Plus Graphics 640",
               "architecture": "Kaby Lake",
            ... -> same as gpu section from get_model_info, refer to it for more information
               "other_info": "QuickSync",
               "rating": "5.8"
            },
            "500": {  -> noteb id for this gpu
               "prod": "Intel",
               "model": "Iris Plus Graphics 640",
               "architecture": "Kaby Lake",
            ... -> same as gpu section from get_model_info, refer to it for more information
               "other_info": "QuickSync",
               "rating": "6.4"
            },
            "selected": 499  -> indicates the element in the returned gpu object that is selected in
the retrieved configuration
         },
         "wireless": {
            "54": {-> noteb id for this wireless card
               "model": "339S025",
            ... -> same as wireless section from get_model_info, refer to it for more information
            },
            "selected": 54  -> indicates the element in the returned wireless object that is selected
in the retrieved configuration
         },
         "optical_drive": {
            "0": {-> noteb id for this optical unit
               "type": "none",
            ... -> same as optical_drive section from get_model_info, refer to it for more information
            },
            "selected": 0  -> indicates the element in the returned optical_drive object that is
selected in the retrieved configuration
         },
         "motherboard": {
            "491": {-> noteb id for this motherboard
               "ram_slots": "soldered",
            ... -> same as motherboard section from get_model_info, refer to it for more information
               "other_info": "Cirrus Logic CS42L83A audio"
            },
            "selected": 491 -> indicates the element in the returned motherboard object that is
selected in the retrieved configuration
         },
         "chassis": {
            "497": {-> noteb id for this laptop chassis
               "height_cm": "1.49",
            ... -> same as chassis_section from get_model_info, refer to it for more information
               "other_info": "4 x speakers,Microphone array"
            },
            "selected": 497 -> indicates the element in the returned chassis object that is selected
in the retrieved configuration
         },
         "battery": {
            "219": {-> noteb id for this battery
               "capacity": "54.5",
            ... -> same as battery section from get_model_info, refer to it for more information
            },
            "selected": 219 -> indicates the element in the returned battery object that is selected
in the retrieved configuration
         },
          "warranty": {
            "1": {-> noteb id for this warranty type
               "years": "1",
               "type_short": "Standard",
            ... -> same as warranty section from get_model_info, refer to it for more information
            },
            "2": {-> noteb id for this warranty type
               "years": "2",
            ... -> same as warranty section from get_model_info, refer to it for more information
            },
            "15": {-> noteb id for this warranty type
               "years": "3",
            ... -> same as warranty section from get_model_info, refer to it for more information"
            },
            "selected": 0 -> indicates the element in the returned warranty object that is selected
in the retrieved configuration
         },
         "operating_system": {
            "3": {-> noteb id for this operating system
               "name": "macOS 10.13" -> same as operating_system section from get_model_info, refer to
it for more information
            },
            "selected": 3 -> indicates the element in the returned operating_system object that is
selected in the retrieved configuration
         },
         "config_score": "41.93",
            ... -> this part is the same as get_model_info, refer to it for more information
         "total_storage_capacity": "512"
      }
   },
   "daily_hits_left": "947"
}
```

3. **list_models**

This function is meant to return a list of available laptop models. The function can return all available models in the database, or it can return just a part of them based on a provided name or partial name.

Parameters

Required parameters

There are no required parameters. With no provided parameters, the function will return all laptop models in the database.

Optional parameters (list_models)

Main optional parameters are model_id OR model_name and have similar requirements as in the "get_model_info" function.

The main difference is that in case of model_name, there is no need to provide precise model identification. Even using one character like "a" is enough. The function will return all laptop models that contain the character "a".

Another optional parameter is *from_date*. It must have the format "yyyy-mm-dd". This parameter will make the API search only through models released after the provided from_date. This parameter can be useful in returning recently introduced laptop models.

A typical request for this function should look like this:

```
{
        ["apikey"]=> "112233aabbcc",
        ["method"]=> "list_models"
        ["param"]=>
        {
                ["model_id"]=> "",
                ["model_name"]=> "Apple"
        }
}
```

Information from the results (list_models)

The function will return a JSON similar to the other function in the API. The function will return information on all the models it identifies, which can be all models in the database or none.

A typical reply will look like this:

```
{
   "code": 26,
   "message": "Valid method.",
   "result": {
      "0": { -> information on the first returned model
         "model_info": [
            {
               "id": 415,
               "noteb_name": "Apple MacBook (2016) ",
               "name": "Apple MacBook (2016)"
            } -> same as the model_info section from the get_model_info function, refer to it for more
information
         ],
         "model_resources": {
            "thumbnail":
"http://86.123.134.36/notebro/res/img/models/thumb/t_405_1.jpg",
            "image_1": "http://86.123.134.36/notebro/res/img/models/405_1.jpg",
            "image_2": "http://86.123.134.36/notebro/res/img/models/405_2.jpg",
            "image_3": "http://86.123.134.36/notebro/res/img/models/405_3.jpg",
            "image_4": "http://86.123.134.36/notebro/res/img/models/405_4.jpg",
            "official_link": "http://www.apple.com/macbook/",
            "official_link2": null,
            "launch_date": "2017-01-16"
         } -> same as the model_resources section from the get_model_info function, refer to it for
more information
      },
      "1": { -> information on the second returned model
         "model_info": [
            {
               "id": 504,
               "noteb_name": "Apple MacBook Air (2015) ",
               "name": "Apple MacBook Air (2015)"
            } -> same as the model_info section from the get_model_info function, refer to it for more
information
         ],
         "model_resources": {
            "thumbnail":
"http://86.123.134.36/notebro/res/img/models/thumb/t_498_1.jpg",
            "image_1": "http://86.123.134.36/notebro/res/img/models/498_1.jpg",
            "image_2": "http://86.123.134.36/notebro/res/img/models/498_2.jpg",
            "image_3": "http://86.123.134.36/notebro/res/img/models/498_3.jpg",
            "image_4": "http://86.123.134.36/notebro/res/img/models/496_4.jpg",
            "official_link": "http://www.apple.com/macbook-air/",
            "official_link2": null,
            "launch_date": "2017-02-06"
         } -> same as the model_resources section from the get_model_info function, refer to it for
more information
      },
[..........................]
      "5": { -> information on the fifth returned model
         "model_info": [
            {
               "id": 850,
               "noteb_name": "Apple MacBook Pro 15 (H12017) V2",
               "name": "Apple MacBook Pro 15 (H12017)",
               "submodel_info": [
                  "V2"
               ]
            } -> same as the model_resources section from the get_model_info function, refer to it for
more information
         ],
         "model_resources": {
            "thumbnail":
"http://86.123.134.36/notebro/res/img/models/thumb/t_497_1.jpg",
            "image_1": "http://86.123.134.36/notebro/res/img/models/497_1.jpg",
            "image_2": "http://86.123.134.36/notebro/res/img/models/497_2.jpg",
            "image_3": "http://86.123.134.36/notebro/res/img/models/497_3.jpg",
            "image_4": null,
            "official_link": "https://www.apple.com/shop/buy-mac/macbook-
pro?product=MPTT2LL/A&step=config",
                "official_link2": null,
                "launch_date": "2017-06-09"
            }  -> same as the model_resources section from the get_model_info function, refer to it for
more information
        }
    },
    "daily_hits_left": "993"
}
```

4. get_conf_info

This function is meant to retrieve configuration information when the components of the model are changed to a new configuration, subsequently needing a recalculation of price, battery life and rating.

For example, the function “get_model_info_all” retrieves all available configuration options for a model, alongside the information of the optimal configuration (performance/price) for that model within the provided parameters.

Take the example already used when describing the “get_model_info_all”. It retrieves the price, battery life and storage capacity for the configuration 12244054780192000000. This configuration is equivalent for the following combination of elements:

model_id: 838
cpu_id: 374
display_id: 27
memory_id: 18
primary_storage_id: 29
secondary_storage_id: 0
gpu_id: 499
wireless_id: 54
optical_drive_id: 0
motherboard_id: 491
chassis_id: 497
battery_id: 219
warranty_id: 0
operating_system_id: 3

However, maybe now that you see what processors are available you would like to change the processor from 374 - `"model": "i5-7360U` to 375: `"model": "i7-7660U"`. However, changing the processor will change the current component configuration of the model, and subsequently may mean a different price, battery life and rating. Thus, it is necessary to retrieve the information for the new configuration.

This is where the function “get_conf_info” comes into play. The function, if provided with a valid configuration id or a valid combination of components, will retrieve from the database the information for that configuration id or combination of components.

Note: Be very careful, changing the processor for a laptop, which comes with an integrated video solution, will require to change the id of the gpu as well. The web service will try to work even if the proper integrated video id is not provided, but it may not always be successful.

Required parameters

conf_id OR a valid combination of {mode_id, cpu_id, display_id, mem_id, hdd_id, shdd_id, gpu_id, wireless_id, optical_drive_id, motherboard_id, chassis_id , battery_id, warranty_id, operating_system_id }

If the id given by noteb.com to a particular laptop configuration is known, then it can be used directly to request information on that configuration.

If the id of the model is not known or needs to be retrieved, a valid combination of component ids needs to be provided. All component ids are mandatory in case this method is used.

Optional parameters
There are no optional parameters.

A typical request for this function with direct configuration id:

```
{
        ["apikey"]=> "112233aabbcc",
        ["method"]=> "get_conf_info"
        ["param"]=>
        {
                ["conf_id"]=> "12244054780192000000",
        }
}
```

A typical request for this function for a combination of model components:

```
{
        ["apikey"]=> "112233aabbcc",
        ["method"]=> "get_conf_info"
        ["param"]=>
        {
                ["model_id"]=> "838",
                ["cpu_id"]=> "375",
                ["display_id"]=> "27",
                ["memory_id"]=> "18",
                ["primary_storage_id"]=> "29",
                ["secondary_storage_id"]=> "0",
                ["gpu_id"]=> "500",
                ["wireless_id"]=> "54",
                ["optical_drive_id"]=> "0",
                ["motherboard_id"]=> "491",
                ["chassis_id"]=> "497",
                ["battery_id"]=> "219",
                ["warranty_id"]=> "0",
                ["operating_sytem_id"]=> "3"
        }
}
```

Information from the results (get_conf_info)

The function will return a JSON similar to the other function in the API. The function will return information only if it finds a valid configuration for the provided information.

This is how a typical reply will look like:

```
{
   "code": 29,
   "message": "Valid method. No valid configuration id provided, falling back to component search.",
      "result": {  ... -> this part is the same as the configuration part from the get_model_info, refer to it for more information
      "model_id": "1046",
      "config_score": "45.76",
      "config_price": "923",
      "config_price_min": "894",
      "config_price_max": "953",
      "battery_life_raw": "3",
      "battery_life_hours": "3:00",
      "total_storage_capacity": "128"
   },
   "daily_hits_left": "470"
}
```

Special notes:

- if the returned price is “0” – zero, it means that configuration is not available for purchase on the market, but it does exist in noteb.com database
- if the returned price is “-1” – minus one , it means the configuration is in the noteb.com database, but it has no information regarding pricing or availability

5. get_exact_conf_info

This function is meant to retrieve all configuration information when the exact configuration id is known. The function will retrieve all model and component information available for **the given configuration**, exactly like the “get_model_info” function.

Unlike the “get_model_info” function, which retrieves the information for the optimal price/performance configuration for the provided parameters, the “get_exact_conf_info” function will return the same data but for the configuration id that has been provided.

Required parameters

The **conf_id** parameter is mandatory. The function will fail without this parameter.

Optional parameters

There are no optional parameters.

A typical request for this function with direct configuration id:

```
{
        ["apikey"]=> "112233aabbcc",
        ["method"]=> "get_exact_conf_info"
        ["param"]=>
        {
                ["conf_id"]=> "13079402429583998976",
        }
}
```

Information from the results (get_exact_conf_info)

The function will return a JSON similar to the other function in the API.
The returned information will have exactly the same structure as the one from “get_model_info”

This is how a typical reply will look like:

```
{
   "code": 26,
   "message": "Valid method.",
   "result": {
      "model_info": [
         {
            "id": 1056,
            "noteb_name": "Lenovo Ideapad Legion Y520 AMD GP.",
            "name": "Lenovo Ideapad Legion Y520",
            "submodel_info": [
               "AMD GP."
            ]
         }
      ],
      "config_id": "13079402429583998976",
      "model_resources": {
         "thumbnail":
"http://86.123.134.36/notebro/res/img/models/thumb/t_886_1.jpg",
         "image_1": "http://86.123.134.36/notebro/res/img/models/886_1.jpg",
         "image_2": "http://86.123.134.36/notebro/res/img/models/886_2.jpg",
… -> see get_model_info for more information
         "launch_date": "2017-08-04"
      },
      "cpu": {
         "prod": "Intel",
         "model": "i5-7300HQ",
         "lithography": "14",
         "cache": "6",
         "base_speed": "2.50",
         "boost_speed": "3.50",
         "cores": "4",
         "tdp": "45",
         "other_info": "SSE4.2,AVX2.0,64-bit,VT-x,VT-d,TBT 2.0",
         "rating": "53.3",
         "integrated_video_id": "448",
         "integrated_video": "Intel HD Graphics 630"
      },
      "display": {
… -> see get_model_info for more information
       },
      "warranty": {
         "years": "1",
         "type_short": "Standard",
         "type_long": "Standard Pick-up & Return"
      },
      "operating_system": "Windows Home 10.00",
      "config_score": "38.13",
      "config_price": "586",
      "config_price_min": "559",
      "config_price_max": "614",
      "battery_life_raw": "3.2",
      "battery_life_hours": "3:10",
      "total_storage_capacity": "1000"
   },
   "daily_hits_left": "482"
}
```

Usage ideas

The functions included in this documentation should be used in conjunctions with one another. For example, if you want to display a configurable model on a mobile app, you should use the function list_models to show the user what models are available. If the user selects a laptop model, use the function get_model_info_all / get_model_info to retrieve and display all information on that model. If the user wants to configure the laptop model, any change in components should be followed by a call to the function get_conf_info to update the price, rating, battery life etc.

Final words

Please be advised that the laptop models, ids, configurations and components used in this documentation may no longer be available on the market. They are used purely for demonstrative purposes.

Other requests

If you have any specific requests that you believe should be implemented and useful for this API, please contact us.