

Getrealprice Snapshots API guideline

Introduction

You are welcome to get our database access using GRP Snapshots API. For example, you might get information about any SKU price for specific date or easily build it's price history for few years. GRP Snapshots API is accessed via <API KEY>, we provide this parameter after requesting it on mail@getrealprice.com.

How it works in general

We generate database extraction containing actual data with fixed frequency. This data is saved to **.JSON** files that are put to **.ZIP** archive. Regular **.ZIP** file include next .json files:

cities.json,
prices.json,
items.json,
properties.json,
attributes.json.

The .JSON file structure description comes next. Please note that the content of snapshots might vary and depends on clients business, for example:

Fashion clients: does not receive **properties.json**;
DIY clients: does not receive **attributes.json**;
Beauty clients: **attributes.json** and **properties.json** might vary;

API Methods

Account information

GET <https://getrealprice.com/master/api/snapshots/?api-key=<API-KEY>>

List of available snapshots

GET <https://getrealprice.com/master/api/snapshots/list/?api-key=<API-KEY>>

Download the latest snapshot without accessing to list of available snapshots

GET <https://getrealprice.com/master/api/snapshots/last/?api-key=<API-KEY>>

How to get API KEY

Request your API KEY at mail@getrealprice.com.

UPD: Since June 2018 new parameters are available:

Optional .JSON files:

Generation of **cities.json** and price recording by city is optional and regulated via parameter **dump_region_prices** (values: true/false)

Generation of **properties.json** is optional, and regulated via parameter **dump_properties** (values: true/false)

Generation of **attributes.json** is optional, and regulated via parameter **dump_attributes** (values: true/false)

Partition:

We propose to use partition for large snapshot files in order to keep RAM in good condition and to speedup .JSON internal parsing.

This is regulated by volume partitioning with listing filenames of a snapshot and amount of records per file.

Partitioning values:

false - partitioning is off;

int - amount of records per file.

If you activate partitioning, we begin to generate files named:

items-part-**1**.json,
items-part-**2**.json,
items-part-**3**.json,
items-part-**4**.json,
items-part-**n**...json,

```

"snapshots_api": {
  "email_notification": {
    "cc": [],
    "to": [],
    "bcc": []
  },
  "partitioning": {
    "prices.json": false,
    "items.json": 30000,
    "properties.json": 30000,
    "attributes.json": 30000
  },
  "snapshot_name": "snapshot",
  "dump_region_prices": true,
  "dump_properties": true,
  "dump_attributes": false
}

```

Data files structure (about .JSON)

sources.json - list of competitors, example:

```

{
  "216": "toom baumarkt",
  "386": "Globus Baumarkt",
  "75": "Hornbach",
  "309": "Bauhaus",
  "45": "OBI"
}

```

categories.json - list of categories of input(clients) catalogue

```

"6107": { // basic level category ID
  "subitems": { // дочерние категории / sub-nested categories
    "9329": {
      "name": "Куртки и пуховики" / "jackets"
    },
    "6108": {

```

```

    "name": "Пуховики и куртки" / "jackets and warm jackets"
  },
  "6125": {
    "name": "Жилеты утепленные" / "gillettes"
  },
  "6119": {
    "name": "Пальто" / "coats"
  }
},
"name": "Верхняя одежда" / "outdoor wear"
}

```

cities.json - list of cities (price clusters) in monitoring

```

{
  "1": "Moscau",
  "null": "City is not applied"
}

```

items.json - list of all active items with it's modifications (if applicable)

(Example) "147457":{ // item SKU code

"category": 6000, //category ID, might be null if not matched

"source": 12, // ID

"mods": { // item ID

"28090": {

"color": "ярко-оранжевый", //modification color

"images": [...], // list of URLs with this item images

"price": 849.0 // SKU price of this color modification

},

....

},

"name": "Свитер", // название, выводимое на сайте источника // sku
name as on webshop

"url": "...", // ссылка на товар на сайте источника // sku url on
webshop

```

"article": "SW-104/650-3416",
"price": 849.0, // цена товара // sku price
"mainImage": "...", //ссылка на основную картинку товара // sku main
image url
"description": "..." //текстовое описание с сайта источника // sku text
description from webshop source
}

```

UPD: Since June 2018 there are few minor changes:

- Active prices are recorded by city level if the city level is applied;
- filed Category now has value **null** instead of **None** for the cases if category is not set:

```

"10835286": {
  "updated": "2018-06-15",
  "description": "",
  "prices": [
    {"city": 1, "price": 22.9},
    {"city": null, "price": 22.9},
  ],
  "active": false,
  "images": [
    "http://www.hornbach.at/data/shop/
D04/001/780/491/226/945/
DV_157x152_8747340_02_4c_DE_20170908031808.jpg
"
  ],
  "article": "8747340",
  "category": null,
  "mods": {},
  "name": "Unterschrank Pelipal Lagos 72x40x33 cm
Eiche natur quer",

```

```

        "url": "http://www.hornbach.at/shop/Unterschrank-
Pelipal-Lagos-72x40x33-cm-Eiche-natur-quer/8747340/
artikel.html",
        "source": 75,
        "mainImage": "http://www.hornbach.at/data/shop/
D04/001/780/491/226/945/
DV_157x152_8747340_02_4c_DE_20170908031808.jpg
"
    },

```

Properties.json - includes product description

```

"10835286": {
    "Räume": "Badezimmer, Gäste-WC",
    "Korpusfarbe": "Eiche natur quer",
    "Frontfarbe": "Weiß glänzend",
    "Einsatzbereich": "Feuchtraum, Innen",
    "Artikeltyp": "Schrank",
    "Montage": "Zerlegt",
    "Beschichtung": "Melaminbeschichtet",
    "Farbe Griff": "Chrom glanz",
    "Material Front": "Spanplatte",
    "Serie": "Lagos",
    "Maße (HxBxT)": "72 cm x 40 cm x 33 cm",
    "Details": "1 Drehtür, 1 Auszug",
    "Form Griff": "Bügelgriff",
    "Ausführung": "Unterschrank",
    "Anwendungsbereich": "Wandhängend",
    "Material Korpus": "Spanplatte"
},

```

prices.json - price change story for each items

```

( Example ) "312996": { // SKU ID
    "2013-10-26": 349.0, // started to monitor - date of 1'st "item meet"
    "2013-11-24": null, // 24.11.2013 this date was no item on the
webshop

```

"2013-11-25": 349.0, // 25.11.2013 this date this item was on the webshop

"2013-12-06": 244.0, // 06.12.2013 this date this item was on the webshop and changed price from 349 to 244
}

UPD: Since June 2018 we store prices for each needed city:

```
"3495244": {  
  "2017-04-28": [  
    {"city": null, "price": null},  
    {"city": 1, "price": null},  
  ]  
  "2016-04-26": [  
    {"city": 1, "price": 22.9},  
    {"city": null, "price": 22.9},  
  ]  
},
```

attributes.json - optional, see API methods.

mail@getrealprice.com

© GetRealPrice 2013-2018