



API v1.6

Octubre/2020

precios

CONTENIDO

Introducción

[HTTP 1.1](#)

[Host URL](#)

[Clave de acceso \(API Key\)](#)

[CORS](#)

[Límites de peticiones](#)

Pricing API

[Paginación](#)

[Filtros](#)

[SKUs](#)

[Status](#)

[Marca](#)

[Categoría](#)

[Cambiados recientemente](#)

[Variantes](#)

[Estatus del Competidor \(Match Directo\)](#)

[Download de toda la data](#)

Retailers API

[Obteniendo el árbol de categorías de mi tienda](#)

[Obteniendo subcategorías](#)

Introducción

Con el objetivo de facilitar que nuestros clientes puedan integrar la data provista por RetailCompass con sus plataformas internas, desarrollamos un conjunto de APIs que permitan el acceso a la base de datos de RetailCompass. A través de esta API, nuestros clientes van a poder integrar la información de precios y disponibilidad con su data y así generar aún más inteligencia competitiva para sus negocios.

Este documento tiene como objetivo facilitar el consumo de este servicio a su equipo técnico.

HTTP 1.1

Todas las transferencias de datos están en base al protocolo HTTP / 1.1, y todas las URLs de acceso al API requieren **HTTPS**.

Host URL

Todos los requests a la API deben usar el host **https://services.retailcompass.com/api**

El intento de acceder a otro endpoint/url no definido en este documento será tratado como amenaza y el usuario será bloqueado permanentemente.

Clave de acceso (API Key)

Permiten que su aplicación acceda a RetailCompass API. Todas las peticiones a la API deben tener el API Key especificado en el *header*.

```
curl -X GET \  
  https://services.retailcompass.com/api/pricing/v1/products \  
  -H 'apikey: <API_KEY>'
```

Para obtener una clave de acceso a la API debes contactar nuestro equipo de soporte: soporte@retailcompass.com.

La clave de acceso cumple con dos funciones:

- Nos permite identificar su aplicación y que retailer representa
- Controlar los límites de requests y acceso a la data contratada



ATENCIÓN

Las claves de acceso a la API son de responsabilidad de los **clientes** de RetailCompass. No deben ser compartidas fuera de la organización y/o emplearse para uso personal.

CORS

La API RetailCompass permite la ejecución de peticiones desde cualquier origen. Si se necesita configurar Cross-origin resource sharing (CORS), debes contactar el equipo de soporte de RetailCompass soporte@retailcompass.com.

Límites de peticiones

El número de solicitudes que pueden realizarse por minuto y mes son limitados, según lo definido en la siguiente tabla:

Requests por Minuto	10
Requests por Mes	5,000
Download de toda la data	2x por día

Tabla 1 - Límite de Solicitudes /Requests por Usuario

OBS: Esos límites pueden ser ajustados de acuerdo con la necesidad de los clientes.

Una vez alcanzado el límite de requests permitido, la API va retornar un error **429 Too Many Requests**, con el mensaje:

```
{
  "message": "API rate limit exceeded"
}
```

En la respuesta de cada petición, es posible ver la cantidad de requests disponibles en el minuto y mes.

Ejemplo del *Response Headers*:

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=UTF-8
Vary: Accept-Encoding
X-RateLimit-Limit-minute: 10
X-RateLimit-Limit-month: 100000
X-RateLimit-Remaining-minute: 9
X-RateLimit-Remaining-month: 99965
Content-Length: 7565
```

Connection: keep-alive

Pricing API

La API de precios permite nuestros clientes acceder a la información de sus propios productos y de productos competidores. De esta manera es posible obtener la información de **precios** y **disponibilidad** de sus productos y los respectivos competidores contratados en RetailCompass.

Endpoints

Endpoint	Descripción	HTTP Methods
/pricing/v1/products	Permite buscar información de tus productos y respectivos competidores.	GET, POST

Conceptos

Product	Representa un producto vendido por un retailer.
Competitors	Representa un <u>producto competidor</u> , vendido por un retailer.
Retailer	Representa un eCommerce que vende un producto
Out of Stock	Productos que no están disponibles para venta en el Retailer. La API solo almacena productos sin stock hasta 30 días. Después de 30 días se entiende que el producto está discontinuado y no estará accesible en la API.

Precios

La API de RetailCompass soporta 3 tipos de precios para cada producto además de un campo que informa del precio más barato del producto.

Normal Price**	Precio normal del producto
Offer Price**	Precio Oferta o Precio Internet, es el precio de venta del producto por internet, sin considerar cualquier promoción o descuento con tarjeta.
Card Price**	Representa el precio del producto cuando la compra es realizada con

	alguna tarjeta de fidelización (e.g CMR, Tarjeta Cencosud, etc)
Lowest Price	Representa el precio más barato de venta de un producto. Puede ser cualquier uno de los precios definidos anteriormente.
Instalment**	Representa con cuantas cuotas sin interés el producto está siendo ofertado

** Esta información está disponible solamente para algunos retailers. Consultar la disponibilidad de precios con nuestro equipo de soporte.

Estructura de la respuesta

La respuesta de la API de Pricing tiene dos elementos principales ***paging*** y ***products***.

Ejemplo:

```
{
  "paging": {
    "total": 87120,
    "pages": 4356,
    "size": 20,
    "currentPage": 0
  },
  "products": [
    {
      "productId": 97346323,
      "storeId": 4,
      "storeName": "Falabella",
      "name": "Cargador Huawei Carga Rapida Micro Usb",
      "sku": "6697840",
      "brand": "Huawei",
      "model": "microusb",
      "url":
        "http://www.falabella.com/falabella-cl/product/6697840/Cargador-Huawei-Carga-Rapida-Micro-Usb/6697840",
      "status": "AVAILABLE",
      "created": "2018-10-01T09:53:38Z",
      "updated": "2019-03-16T15:52:08Z",
      "extracted": "2019-04-23T15:52:08Z",
      "prices": {
        "lowest": 1049,
        "cardPrice": 1049,
        "offerPrice": 1149,
        "normalPrice": 1299
      },
      "paymentMethod": {
        "instalments": 12
      }
    },
    {
      "categories": {
        "web": [
          {
            "id": 887442,
            "categoryIdPath": "73112>67267",

```



```

        "fullPath": "Tecnología > Accesorios"
      }
    ],
    "internal": [
      {
        "id": 688471,
        "categoryIdPath": "54312334>1523",
        "fullPath": "<Path de las categorías internas>"
      }
    ]
  },
  "competitors": [...]
}
]
}

```

Información de productos

Nombre Atributo	Descripción
productId	ID interno del producto.
storeId	ID interno del Retailer que vende el producto
storeName	Nombre del Retailer que vende el producto
name	Nombre del producto conforme descrito en el retailer
sku	SKU del producto en el Retailer
brand (opcional)	Marca del producto conforme descrita en la página del Retailer. OBS: algunos retailers no especifican la marca del producto. Siendo así, este campo no estará presente en la respuesta.
model (opcional)	Modelo del producto conforme descrito en la página del Retailer. OBS: algunos retailers no especifican la modelo del producto. Siendo así, este campo no estará presente en la respuesta.
url	Url del producto en el Retailer
status	Status del producto. Valores posibles: AVAILABLE o OUT_OF_STOCK

created	<p>Fecha en la cual el producto fue creado en la base de datos de RetailCompass. Timezone: UTC</p>
updated	<p>Fecha en la cual el precio o disponibilidad del producto ha cambiado. Timezone: UTC</p>
extracted	<p>Fecha en la cual la información del producto fue actualizada (ultimo scraping). Timezone: UTC</p>
prices	<p>Conforme descrito anteriormente, este campo contiene información de los precios del producto.</p> <p>Este objeto contiene los siguientes atributos:</p> <p>lowest: precio más bajo del producto. Obligatorio cardPrice: precio por tarjeta. Opcional offerPrice: precio internet/Oferta. Opcional</p>
categories	<p>Información de las categorías de los productos. Ese objeto contiene dos atributos:</p> <p>web: lista de categorías la cual el producto esta presenta en la web</p> <p>internal: lista de categorías internas la cual el producto está presenta. La información de categorías internas solo aparecerá para clietes que tienen el servicio contratado.</p>

Paginación

Para navegar el listado de productos, se utilizan los parámetros **page** y **size**. En la primera petición a Pricing API, la respuesta contendrá el total de páginas disponibles.

OBS: La numeración de páginas inicial es cero.

Request para obtener el número de páginas:

```
curl -X GET \
  https://services.retailcompass.com/api/pricing/v1/products \
  -H 'apikey: <API_KEY>'
```

Respuesta:

```
{
  "paging": {
    "total": 88383,
    "pages": 4419,
    "size": 20
  },
  "products": [...]
}
```

En este ejemplo, el resultado tiene 4419 páginas de 20 productos. Para obtener la página siguiente, basta enviar el número de la próxima página usando el parámetro **page**:

Request para obtener la segunda página del listado de productos:

```
curl -X GET \
  'https://services.retailcompass.com/api/pricing/v1/products?page=1' \
  -H 'apikey: <API_KEY>'
```

Por defecto, el tamaño de las páginas es de 20 productos. Para cambiar el valor por defecto, basta ejecutar el request informando el **size** deseado (el máximo es 50 productos por página).

Request que cambia el total de productos a 50 por página:

```
curl -X GET \  
'https://services.retailcompass.com/api/pricing/v1/products?size=50&page=0'  
\  
-H 'apikey: <API_KEY>'
```

La paginación también puede ser definida en el cuerpo del request POST. Según el siguiente ejemplo:

```
curl -X POST \  
https://services.retailcompass.com/api/pricing/v1/products \  
-H 'Content-Type: application/json' \  
-H 'apikey: <API_KEY>' \  
-d '{  
  "paging": {  
    "page": 0,  
    "size": 50  
  },  
  "filters": {  
    "status": [  
      "AVAILABLE"  
    ]  
  }  
'
```

Filtros

La versión actual de la API RetailCompass permite filtros de productos, con el objetivo de facilitar el acceso a la data.

Los filtros pueden ser enviados usando **GET** o **POST**

OBS¹: Para utilizar POST, se debe enviar el encabezado HTTP: Content-Type: application/json

OBS²: Los filtros funcionan con el operador lógico **AND**. Es decir, si el request contiene el filtro de marca y sku, la API buscará productos que cumplan con las dos condiciones.

Ejemplo filtrando marca y status por **GET**:

```
curl -X GET \
'https://services.retailcompass.com/api/pricing/v1/products?brand=LG&status=AVAILABLE,OUT_OF_STOCK' \
-H 'apikey: <API_KEY>'
```

Ejemplo filtrando marca y status por **POST**:

```
curl -X POST \
https://services.retailcompass.com/api/pricing/v1/products \
-H 'Content-Type: application/json' \
-H 'apikey: <API_KEY>' \
-d '{
  "filters": {
    "brand": "LG",
    "status": [
      "AVAILABLE", "OUT_OF_STOCK"
    ]
  }
}'
```

SKUs

Permite filtrar productos por SKUs de **mi tienda**

OBS: Este campo acepta como máximo **mil SKUs**. Retornando un error *400 Bad Request*, caso el número de SKUs sea inválido.

Ejemplo 1:

```
curl -X GET \
'https://services.retailcompass.com/api/pricing/v1/products?skus=6454689,6469819,6469820' \
-H 'apikey: <API_KEY>'
```

Ejemplo 2:

```
curl -X POST \
https://services.retailcompass.com/api/pricing/v1/products \
-H 'Content-Type: application/json' \
-H 'apikey: <API_KEY>' \
-d '{
  "filters": {
    "skus": [
      "6454689",
      "6469819",
      "6469820"
    ]
  }
}'
```

Status

Los productos en los retailers tiene dos status principales:

AVAILABLE	Producto está disponible en el retailer. OBS: este es el valor por defecto del filtro de productos.
OUT_OF_STOCK	Define que un producto no está disponible en el Retailer o la URL ya no existe.

Ejemplo 1

```
curl -X GET \
'https://services.retailcompass.com/api/pricing/v1/products?status=AVAILABLE,OUT_OF_STOCK' \
-H 'apikey: <API_KEY>'
```

Exemplo 2

```
curl -X POST \
  https://services.retailcompass.com/api/pricing/v1/products \
  -H 'Content-Type: application/json' \
  -H 'apikey: <API_KEY>' \
  -d '{
    "filters": {
      "status": [
        "AVAILABLE",
        "OUT_OF_STOCK"
      ]
    }
  }'
```



ATENCIÓN

La API RetailCompass disponibiliza productos sin stock **hasta 30 días**.
Un producto con más de 30 días sin stock, no estará disponible para consulta en la API.

Marca

El filtro de marca de la API usa una búsqueda textual y retorna productos que coinciden con el texto especificado. Es texto no es “case sensitive”, el uso de mayúsculas y minúsculas no cambia el resultado de la consulta.

OBS: La marca buscada no puede tener más que 50 caracteres. Retornando un error *400 Bad Request*, si la marca no es válida.

Ejemplo 1:

```
curl -X GET \
  'https://services.retailcompass.com/api/pricing/v1/products?brand=Sony' \
  -H 'apikey: <API_KEY>'
```

Ejemplo 2:

```
curl -X POST \
  https://services.retailcompass.com/api/pricing/v1/products \
  -H 'Content-Type: application/json' \
  -H 'apikey: <API_KEY>' \
  -d '{
```

```
"paging": {
  "size": 5
},
"filters": {
  "brand": "Sony"
}
}'
```

Categoría

Para filtrar productos por una categoría, es necesario que el cliente del API conozca el “ID Path” de la categoría. Para obtener esta información, por favor revisar **Retailer API**. Una vez con la información de categorías se puede filtrar productos solamente de esta categoría, según se muestra en los ejemplos:

Ejemplo 1:

```
curl -X GET \
'https://services.retailcompass.com/api/pricing/v1/products?categoryIdPath=73112%3E293006%3E293007' \
-H 'apikey: <API_KEY>'
```

Ejemplo 2:

```
curl -X POST \
https://services.retailcompass.com/api/pricing/v1/products \
-H 'Content-Type: application/json' \
-H 'apikey: <API_KEY>' \
-d '{
  "filters": {
    "categoryIdPath": "73112>293006>293033"
  }
}'
```

Cambiados recientemente

El filtro “*updatedAfter*” busca productos cuya disponibilidad o el precio del producto o de sus competidores ha cambiado después de la fecha definida.

OBS: La fecha buscada debe estar dentro de las últimas 24 horas. Retornando un error **400 Bad Request**, en el caso de que la fecha sea inválida.


```
{
  "filters": {
    "updatedAfter": "2019-03-17T08:48:00-03:00"
  }
}
```

En este ejemplo, la API devuelve productos cuya información se actualizó después del **17/03/2019 08:48 AM** hora de Chile.

El string pasado como parámetro usa el estándar **ISO 8601** de fechas.

Ejemplo 1

```
curl -X GET \
'https://services.retailcompass.com/api/pricing/v1/products?updatedAfter=2019-03-25T17:40:13-00:00' \
-H 'apikey: <API_KEY>'
```

Ejemplo 2:

```
curl -X POST \
https://services.retailcompass.com/api/pricing/v1/products \
-H 'Content-Type: application/json' \
-H 'apikey: <API_KEY>' \
-d '{
  "filters": {
    "updatedAfter": "2019-03-25T17:40:13-00:00"
  }
}'
```

Variantes

El filtro “*variants*” posibilita filtrar productos que son variantes (por ejemplo: tallas de un producto) de un producto. Por defecto, la API no retorna las variantes de los productos.

Valores Posibles:

ALL	Retorna todos los productos, siendo una variante o no.
NO_VARIANTS	Retorna solamente productos “padres”, abrir por variante.
VARIANTS	Retorna solamente productos que son variantes

Ejemplo 1

```
curl -X GET \
'https://services.retailcompass.com/api/pricing/v1/products?variants=ALL' \
-H 'apikey: <API_KEY>'
```

Ejemplo 2:

```
curl -X POST \
https://services.retailcompass.com/api/pricing/v1/products \
-H 'Content-Type: application/json' \
-H 'apikey: <API_KEY>' \
-d '{
  "filters": {
    "variants": "NO_VARIANTS"
  }
}'
```

Estatus del Competidor (Match Directo)

El filtro “*matchStatus*” posibilita filtrar productos por el estatus del competidor.

Valores Posibles:

ALL	Retorna todos los productos y competidores.
IDENTICAL	Retorna solamente productos con competidor directo.
COMPARABLE	Retorna solamente productos con competidor comparable.

Ejemplo 1

```
curl -X GET \
'https://services.retailcompass.com/api/pricing/v1/products?matchStatus=ALL' \
-H 'apikey: <API_KEY>'
```

Obs: Cuando se filtra por Identical o Comparable no se retornan productos que no tengan competidores.

Download de toda la data

Para escenarios donde es necesario bajar toda la información de sus productos, se creó un endpoint especial, lo cual permite con un request bajar toda la información disponible de su retail.

Ejemplo 1

```
curl -X GET \
'https://services.retailcompass.com/api/pricing/v1/products/stream' \
-H 'apikey: <API_KEY>' \
-H 'Accept: application/stream+json' \
```

Este endpoint tiene un comportamiento distinto de los demás. Retornará un “stream” de objetos json, donde cada línea representa un producto. La estructura del objeto producto es la misma definida anteriormente.

Para más información sobre ese tipo de respuesta, consulte [Line-delimited JSON](#).

Ejemplo de la respuesta:

```
{"productId":97346323,"storeId":3,"prices":{}}\n{"productId":86814458,"storeId":3,"prices":{}}\n{"productId":87020719,"storeId":3,"prices":{}}\n{"productId":85743814,"storeId":3,"prices":{}}\n
```

El endpoint “*pricing/v1/products/stream*”, soporta los mismos filtros *brand*, *skus*, *status*, *updatedAfter*, definidos anteriormente.

Ejemplo 2 - Agregando filtros:

```
curl -X GET \
'https://services.retailcompass.com/api/pricing/v1/products/stream?status=A\nAVAILABLE,OUT_OF_STOCK&brand=Sony' \
-H 'apikey: <API_KEY>' \
-H 'Accept: application/stream+json' \
```

Ejemplos de librerías que soportan respuestas del tipo *application/stream+json*:
[JacksonStreaming API \(Java\)](#),
[Spring Webflux \(Java\)](#), [Hyperquest \(node.js\)](#), [Requests \(Python\)](#)

Cómo probar sin gastar cuotas de full dump

Básicamente se deberá probar con la siguiente url

Ejemplo 1

```
curl -X GET \
'https://services.retailcompass.com/api/pricing/v1/products/stream/test' \
-H 'apikey: <API_KEY>' \
-H 'Accept: application/stream+json' \
```

Se comporta de la misma forma que el stream anterior descrito, pero con la salvedad que solo devolverá los primeros 20 productos del stream. Las respuestas son idénticas al stream, solamente tendrá la restricción de 20 primeros productos y no descontará las descargas masivas.

Retailers API

La API de Retailers, permite a los clientes acceder a la información de tiendas. Sirve principalmente como soporte a las demás APIs de RetailCompass.

Endpoints

Endpoint	Descripción	HTTP Methods
retailers/v1/competitors	Permite buscar información de tus productos y respectivo competidores.	GET
retailers/v1/categories	Permite navegar en el árbol de tus categorías.	GET

Obteniendo el árbol de categorías de mi tienda

Esta información es necesaria para filtrar productos por categoría en la API de Pricing.

OBS: El campo “idsPath” retornado puede ser utilizado para filtrar productos de una categoría en la API de Pricing.

Ejemplo 1 - Obteniendo las categorías de primer nivel:

```
curl -X GET \  
  https://services.retailcompass.com/api/retailers/v1/categories \  
  -H 'apikey: <API_KEY>'
```

Respuesta:

```
{  
  "total": 3,  
  "categories": [  
    {  
      "id": 97251,  
      "name": "Belleza",  
      "fullPath": "Belleza",  
      "idsPath": "93112>97251"  
    },  
    {  
      "id": 97332,  
      "name": "Decohogar",  
      "fullPath": "Decohogar",  
      "idsPath": "93112>97332"  
    },  
    {  
      "id": 97419,  
      "name": "Dormitorio",  
      "fullPath": "Dormitorio",  
      "idsPath": "93112>97419"  
    }  
  ]  
}
```

Obteniendo subcategorías

Para navegar en las subcategorías, basta informar el id de la categoría de la que quiere buscar las subcategorías, según el siguiente ejemplo:

```
curl -X GET \  
  'https://services.retailcompass.com/api/retailers/v1/categories?parentId=97332' \  
  -H 'apikey: <API_KEY>'
```

Este ejemplo busca las subcategorías de la categoría ID 97332 Decohogar, obtenida en el ejemplo anterior.

OBS: Los ids son solo ejemplos, deben ser reemplazados por los IDs de sus categorías.