

## API DASHBOARD

- [Your Account](#)
- [Your API Metrics](#)
- [Your Searches](#)
- [Your Playground](#)

## API DOCUMENTATION

- [Google Search API](#)
- [Google Light Search API](#)
- [Google Ads Transparency API](#)
- [Google AI Mode API](#)
- [Google AI Overview API](#)
- [Google Autocomplete API](#)
- [Google Events API](#)
- [Google Forums API](#)
- [Google Finance API](#)
- [Google Flights API](#)
- [Google Hotels API](#)
- [Google Images API](#)
- [Google Images Light API](#)
- [Google Immersive Product API](#)
- [Google Jobs API](#)
- [Google Lens API](#)
- [Google Local API](#)
- [Google Local Services API](#)
- [Google Maps API](#)
- [Google Maps Reviews API](#)
- [Google News API](#)
- [Google News Light API](#)
- [Google Patents API](#)
- [Google Play Store API](#)
- [Google Related Questions API](#)
- [Google Reverse Image API](#)
- [Google Scholar API](#)
- [Google Shopping API](#)
- [AI Overview](#)
- [Categorized Shopping Results](#)
- [Filters Results](#)
- [Inline Results](#)
- [Shopping Results](#)
- [Google Shopping Light API](#)
- [Google Light Fast API](#)
- [Google Short Videos API](#)
- [Google Travel Explore API](#)
- [Google Trends API](#)
- [Google Videos API](#)
- [Google Videos Light API](#)
- [Amazon Search API](#)
- [Apple App Store API](#)
- [Baidu Search API](#)
- [Bing Images API](#)
- [Bing Copilot API](#)
- [Bing Search API](#)
- [DuckDuckGo Search API](#)
- [DuckDuckGo Light API](#)
- [eBay Search API](#)
- [Facebook Profile API](#)
- [Naver Search API](#)
- [The Home Depot Search API](#)
- [Tripadvisor Search API](#)
- [Walmart Search API](#)
- [Yahoo! Search API](#)
- [Yandex Search API](#)
- [Yelp Search API](#)
- [YouTube Search API](#)
- [Extra APIs](#)
- [Status and Error Codes](#)

# Google Shopping API

API uptime 99.750%

The Google Shopping Results API allows a user to scrape the results of a Google Shopping search.

The API is accessed through the following endpoint: [/search?](#)  
`engine=google_shopping`.

You can query: [https://serpapi.com/search?engine=google\\_shopping&q=coffee](https://serpapi.com/search?engine=google_shopping&q=coffee) utilizing a `GET` request. Head to the [playground](#) for a live and interactive demo.

The screenshot shows a search interface with 'Search Query' set to 'Iphone' and 'location' set to 'Austin, Texas, United States'. The results page displays a grid of iPhone models with their prices and availability from various sellers like Verizon, AT&T, and T-Mobile. A JSON response on the right side details the search metadata, parameters, information, and results, including the search URL and raw HTML file.

```

{
  "search_metadata": {
    "id": "68ac0607fc8a2b0f3a52f842",
    "status": "Success",
    "json_endpoint": "https://serpapi.com/searches/493df56ac431ca6e/68ac0607fc8a2b0f3a52f842.json",
    "created_at": "2025-08-25 06:43:19 UTC",
    "processed_at": "2025-08-25 06:43:19 UTC",
    "google_shopping_url": "https://www.google.com/search?utm=28&q=Iphone&hl=en&gl=us",
    "raw_html_file": "https://serpapi.com/searches/493df56ac431ca6e/68ac0607fc8a2b0f3a52f842.html",
    "total_time_taken": 1.31
  },
  "search_parameters": {
    "engine": "google_shopping",
    "q": "Iphone",
    "google_domain": "google.com",
    "hl": "en",
    "gl": "us",
    "device": "desktop"
  },
  "search_information": {
    "query_displayed": "Iphone",
    "shopping_results_state": "Results for exact spelling"
  },
  "shopping_results": [
    {
      "position": 1,
      "title": "Apple iPhone 16 2GB 8GB Smartphone",
      "product_link": "https://www.google.com/shopping/product/4697624384995825361?gl=us",
      "product_id": "4697624384995825361",
      "serpapi_product_api": "https://serpapi.com/search.json?engine=google_product&gl=us&google_impressive_product_page_token=eyJlaSI6IkNBYXNhTkNXT0t1SnhjOFBoZFNIMLF3IwiHJvZHvjdGK",
      "serpapi_impressive_product_api": "https://serpapi.com/search.json?engine=google_impressive",
      "source": "eBay",
      "source_icon": "https://serpapi.com/images/url/xZqwXnicDcnhDkJAAADgJwojk7bWGhFKYyrX",
      "price": "$430.00",
      "extracted_price": 430,
      "second_hand_condition": "pre-owned",
      "rating": 3.4,
      "reviews": 97,
      "thumbnail": "https://encrypted-tbn2.gstatic.com/shopping?q=tbn:ANd9GcRaVC3KFmjYI0kYoKs3s",
      "serpapi_thumbnail": "https://serpapi.com/images/url/xZqwXnicDcnhDkJAAADgJwojk7bWGhFKYyrX",
      "delivery": "Get it by Sep 4 (Free)"
    },
    {
      "position": 2,
      "title": "Apple iPhone 14",
      "product_link": "https://www.google.com/shopping/product/5641263725124944166?gl=us",
      "product_id": "5641263725124944166"
    }
  ]
}
  
```

## ↔ GET

```
1 | https://serpapi.com/search.json?
  engine=google_shopping&q=Iphone&location=Austin,+Texas,+United+States&google_domain=google.com&hl=en&gl=us
```

## ↔ Code to integrate

```
1 | require "serpapi"
2 |
3 | client = SerpApi::Client.new(
4 |   engine: "google_shopping",
5 |   q: "Iphone",
6 |   location: "Austin, Texas, United States",
7 |   google_domain: "google.com",
8 |   hl: "en",
9 |   gl: "us",
10 |   api_key: "a85853e3401522bd0dbb7b0d6b4901dc8a5849baae4354be6cea1dd229296958"
11 |
12 |
13 | results = client.search
```

CURL Ruby Python JavaScript Go PHP Java Rust .NET Google Sheets

## API Parameters

### Search Query



Required

Parameter defines the query you want to search. You can use anything that you would use in a regular Google Shopping search.

## API DASHBOARD

- [Your Account](#)
  - [Your API Metrics](#)
  - [Your Searches](#)
  - [Your Playground](#)
- API DOCUMENTATION
- [Google Search API](#)
  - [Google Light Search API](#)
  - [Google Ads Transparency API](#)
  - [Google AI Mode API](#)
  - [Google AI Overview API](#)
  - [Google Autocomplete API](#)
  - [Google Events API](#)
  - [Google Forums API](#)
  - [Google Finance API](#)
  - [Google Flights API](#)
  - [Google Hotels API](#)
  - [Google Images API](#)
  - [Google Images Light API](#)
  - [Google Immersive Product API](#)

[location](#)

Optional

Parameter defines from where you want the search to originate. If several locations match the location requested, we'll pick the most popular one. Head to the [/locations.json API](#) if you need more precise control. The [location](#) and [uule](#) parameters can't be used together. It is recommended to specify [location](#) at the city level in order to simulate a real user's search. If [location](#) is omitted, the search may take on the location of the proxy.

[uule](#)

Optional

Parameter is the Google encoded location you want to use for the search. [uule](#) and [location](#) parameters can't be used together.

## Localization

[google\\_domain](#)

Optional

Parameter defines the Google domain to use. It defaults to [google.com](#). Head to the [Google domains](#) for a full list of supported Google domains.

[gl](#)

Optional

Parameter defines the country to use for the Google search. It's a two-letter country code. (e.g., [us](#) for the United States, [uk](#) for United Kingdom, or [fr](#) for France) Head to the [Google Shopping countries](#) for a full list of supported Google Shopping countries.

[hl](#)

Optional

Parameter defines the language to use for the Google Shopping search. It's a two-letter language code. (e.g., [en](#) for English, [es](#) for Spanish, or [fr](#) for French) Head to the [Google languages](#) for a full list of supported Google languages.

## Advanced Filters

[shops](#)

Optional

The parameter defines the token that includes metadata about query and search filter(s). Providing [q](#) parameter alongside the [shops](#) is **not** required.

To apply multiple filters join them with `||` separator, e.g.  
[shops\\_1||shops\\_2||shops\\_3](#).

[min\\_price](#)

Optional

Lower bound of price range query.

This parameter overrides corresponding filter embedded into [shops](#) parameter.

[max\\_price](#)

Optional

Upper bound of price range query.

This parameter overrides corresponding filter embedded into [shops](#) parameter.

[sort\\_by](#)

Optional

Parameter defines the sorting order of the results.

Available options:

[1](#) - Price: low to high

[2](#) - Price: high to low

This parameter overrides corresponding filter embedded into [shops](#) parameter.

[free\\_shipping](#)

Optional

Show only products with free shipping.

This parameter overrides corresponding filter embedded into [shops](#) parameter.

[on\\_sale](#)

Optional

Show only products that are on sale.

This parameter overrides corresponding filter embedded into [shops](#) parameter.

[small\\_business](#)

Optional

Show only products from small business.

This parameter overrides corresponding filter embedded into [shops](#) parameter.

## Pagination

[start](#)

Optional

Parameter defines the result offset. It skips the given number of results. It's used for pagination. (e.g., [0](#) (default) is the first page of results, [60](#) is the 2nd page of results, [120](#) is the 3rd page of results, etc.).

For the new layout, the parameter is not recommended. To easily retrieve paginated results accurately, it is advisable to follow the link provided in [serpapi\\_pagination.next](#).

[num](#)

Optional

Parameter defines the maximum number of results to return. (e.g., [60](#) (default) returns 60 results, [40](#) returns 40 results, and [100](#) (maximum) returns 100 results).

Any number greater than maximum number ([100](#)) will default to [100](#).

Any number lesser than minimum number ([1](#)) will default to [60](#).

For the new layout, the parameter is no longer supported.

## Serpapi Parameters

[engine](#)

Required

Set parameter to [google\\_shopping](#) to use the Google Shopping API engine.

[device](#)

Optional

Parameter defines the device to use to get the results. It can be set to [desktop](#) (default) to use a regular browser, [tablet](#) to use a tablet browser (currently using iPads), or [mobile](#) to use a mobile browser.

[no\\_cache](#)

Optional

Parameter will force SerpApi to fetch the Google Shopping results even if a cached version is already present. A cache is served only if the query and all parameters are exactly the same. Cache expires after 1h. Cached searches are free, and are

## API DASHBOARD

- [Your Account](#)
- [Your API Metrics](#)
- [Your Searches](#)
- [Your Playground](#)

## API DOCUMENTATION

- [Google Search API](#)
- [Google Light Search API](#)
- [Google Ads Transparency API](#)
- [Google AI Mode API](#)
- [Google AI Overview API](#)
- [Google Autocomplete API](#)
- [Google Events API](#)
- [Google Forums API](#)
- [Google Finance API](#)
- [Google Flights API](#)
- [Google Hotels API](#)
- [Google Images API](#)
- [Google Images Light API](#)
- [Google Immersive Product API](#)
- [Google Jobs API](#)
- [Google Lens API](#)
- [Google Local API](#)
- [Google Local Services API](#)
- [Google Maps API](#)
- [Google Maps Reviews API](#)
- [Google News API](#)
- [Google News Light API](#)
- [Google Patents API](#)
- [Google Play Store API](#)
- [Google Related Questions API](#)
- [Google Reverse Image API](#)
- [Google Scholar API](#)
- [Google Shopping API](#)
- [AI Overview](#)
- [Categorized Shopping Results](#)
- [Filters Results](#)
- [Inline Results](#)
- [Shopping Results](#)
- [Google Shopping Light API](#)
- [Google Light Fast API](#)
- [Google Short Videos API](#)
- [Google Travel Explore API](#)
- [Google Trends API](#)
- [Google Videos API](#)
- [Google Videos Light API](#)
- [Amazon Search API](#)
- [Apple App Store API](#)
- [Baidu Search API](#)
- [Bing Images API](#)
- [Bing Copilot API](#)
- [Bing Search API](#)
- [DuckDuckGo Search API](#)
- [DuckDuckGo Light API](#)
- [eBay Search API](#)
- [Facebook Profile API](#)
- [Naver Search API](#)
- [The Home Depot Search API](#)
- [Tripadvisor Search API](#)
- [Walmart Search API](#)
- [Yahoo! Search API](#)
- [Yandex Search API](#)
- [Yelp Search API](#)
- [YouTube Search API](#)
- [Extra APIs](#)
- [Status and Error Codes](#)

[async](#)

Optional

Parameter defines the way you want to submit your search to SerpApi. It can be set to `false` (default) to open an HTTP connection and keep it open until you got your search results, or `true` to just submit your search to SerpApi and retrieve them later. In this case, you'll need to use our [Searches Archive API](#) to retrieve your results. `async` and `no_cache` parameters should not be used together. `async` should not be used on accounts with [Ludicrous Speed](#) enabled.

[zero\\_trace](#)

Optional

Enterprise only. Parameter enables [ZeroTrace mode](#). It can be set to `false` (default) or `true`. Enable this mode to skip storing search parameters, search files, and search metadata on our servers. This may make debugging more difficult.

[api\\_key](#)

Required

Parameter defines the SerpApi private key to use.

[output](#)

Optional

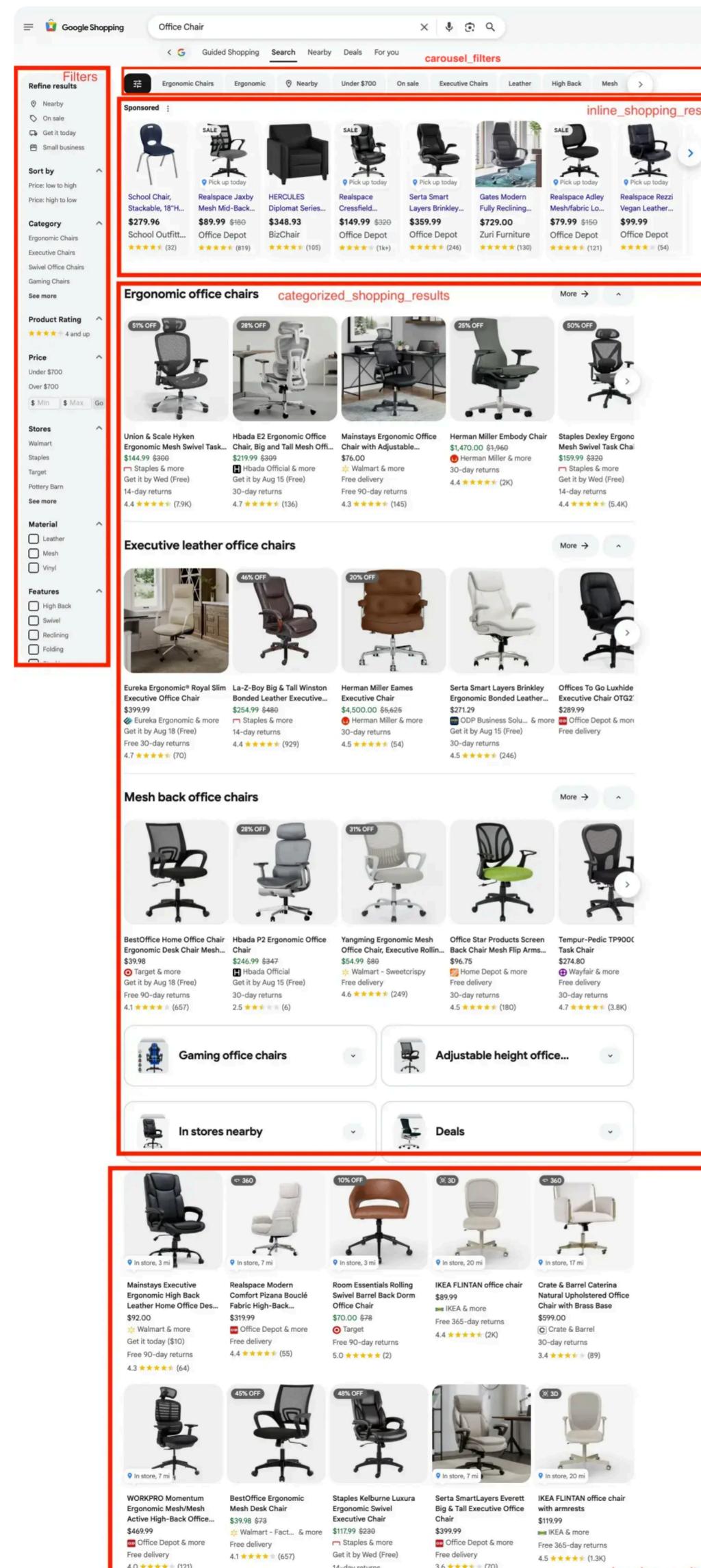
Parameter defines the final output you want. It can be set to `json` (default) to get a structured [JSON](#) of the results, or `html` to get the raw html retrieved.

[json\\_restrictror](#)

Optional

Parameter defines the fields you want to restrict in the outputs for smaller, faster responses. See [JSON Restrictor](#) for more details.

## API Results

Full results for [Office Chair](#)[GET](#)

```
</> GET
1 | https://serpapi.com/search.json?
engine=google_shopping&q=Office+Chair
```

[Code to integrate](#)

```
1 | require "serpapi"
2 |
3 | client = SerpApi::Client.new(
4 |   engine: "google_shopping",
5 |   q: "Office Chair",
6 |   api_key:
7 |     "a85853e3401522bd0db7b0d6b4901dc8a5849baae4354be6cea1dd229296958"
8 |
9 |   results = client.search
10 | shopping_results = results[:shopping_results]
```

[JSON Example](#)[↗](#)[↑](#)[↓](#)



