Place Search

The Google Places API Web Service is also available with the Java Client, Python Client, Go Client and Node.js Client for Google Maps Services (https://developers.google.com/places/web-service/client-library). The Places API Web Service and the client libraries are for use in server applications.

If you're building a client-side application, take a look at the Google Places API for Android (https://developers.google.com/places/android-api/), the Google Places API for iOS (https://developers.google.com/places/ios-api/), and the Places Library in the Google Maps JavaScript API (https://developers.google.com/maps/documentation/javascript/places).

Notice: Radar Search (https://developers.google.com/places/web-service/search#RadarSearchRequests) is deprecated as of June 30, 2017. This feature will be turned down on June 30, 2018, and will no longer be available after that date.

The Google Places API Web Service allows you to query for place information on a variety of categories, such as: establishments, prominent points of interest, geographic locations, and more. You can search for places either by proximity or a text string. A Place Search returns a list of places along with summary information about each place; additional information is available via a Place Details (https://developers.google.com/places/web-service/details) query.

Nearby Search Requests

Earlier versions of the Places API referred to Nearby Search as Place Search.

A Nearby Search lets you search for places within a specified area. You can refine your search request by supplying keywords or specifying the type of place you are searching for.

A Nearby Search request is an HTTP URL of the following form:

```
https://maps.googleapis.com/maps/api/place/nearbysearch/output?parameters
```

where `output` may be either of the following values:

- `json` (recommended) indicates output in JavaScript Object Notation (JSON)
- `xml` indicates output as XML

Certain parameters are required to initiate a Nearby Search request. As is standard in URLs, all parameters are separated using the ampersand (&) character.

Required parameters

- `key` — Your application's API key (https://support.google.com/googleapi/answer/6158862). This key identifies your application for purposes of quota management and so that places added (https://developers.google.com/places/web-service/add-place) from your application are made immediately available to your app. See Get a key (https://developers.google.com/places/web-service/get-api-key) for more information.
- `location` — The latitude/longitude around which to retrieve place information. This must be specified as *latitude,longitude*.
- `radius` — Defines the distance (in meters) within which to return place results. The maximum allowed radius is 50 000 meters. Note that `radius` must not be included if `rankby=distance` (described under **Optional parameters** below) is specified.
- If `rankby=distance` (described under **Optional parameters** below) is specified, then one or more of `keyword`, `name`, or `type` is required.

Optional parameters

- `keyword` — A term to be matched against all content that Google has indexed for this place, including but not limited to name, type, and address, as well as customer reviews and other third-party content.

- `language` — The language code, indicating in which language the results should be returned, if possible. See the <u>list of supported languages</u> (https://developers.google.com/maps/faq#languagesupport) and their codes. Note that we often update supported languages so this list may not be exhaustive.

- `minprice` and `maxprice` (*optional*) — Restricts results to only those places within the specified range. Valid values range between 0 (most affordable) to 4 (most expensive), inclusive. The exact amount indicated by a specific value will vary from region to region.

- `name` — A term to be matched against all content that Google has indexed for this place. Equivalent to `keyword`. The `name` field is no longer restricted to place names. Values in this field are combined with values in the `keyword` field and passed as part of the same search string. We recommend using only the `keyword` parameter for all search terms.

- `opennow` — Returns only those places that are open for business at the time the query is sent. Places that do not specify opening hours in the Google Places database will not be returned if you include this parameter in your query.

- `rankby` — Specifies the order in which results are listed. Note that `rankby` must not be included if `radius` (described under **Required parameters** above) is specified. Possible values are:

  - `prominence` (default). This option sorts results based on their importance. Ranking will favor prominent places within the specified area. Prominence can be affected by a place's ranking in Google's index, global popularity, and other factors.

  - `distance`. This option biases search results in ascending order by their distance from the specified `location`. When `distance` is specified, one or more of `keyword`, `name`, or `type` is required.

- `type` — Restricts the results to places matching the specified type. Only one type may be specified (if more than one type is provided, all types following the first entry are ignored). See the <u>list of supported types</u> (https://developers.google.com/places/web-service/supported_types).

- `pagetoken` — Returns the next 20 results from a previously run search. Setting a `pagetoken` parameter will execute a search with the same parameters used previously — all parameters other than `pagetoken` will be ignored.

Note for Google Maps APIs Premium Plan customers: You must include an API key in your requests. You should **not** include a `client` or `signature` parameter with your requests.

Nearby search example

The following example is a search request for places of type 'restaurant' within a 500m radius of a point in Sydney, Australia, containing the word 'cruise':

```
https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=-33.8670522,151.1957362&radius=500&typ
```

Note: In this example, you need to replace the `key` with your own API key in order for the request to work in your application.

Text Search Requests

The Google Places API Text Search Service is a web service that returns information about a set of places based on a string — for example "pizza in New York" or "shoe stores near Ottawa" or "123 Main Street". The service responds with a list of places matching the text string and any location bias that has been set.

The service is especially useful for making <u>ambiguous address queries</u> (https://developers.google.com/maps/documentation/geocoding/best-practices) in an automated system, and non-address components of the string may match businesses as well as addresses. Examples of ambiguous address queries are incomplete addresses, poorly formatted addresses, or a request that includes non-address components such as business names.

The search response will include a list of places. You can send a Place Details request for more information about any of the places in the response.

The Google Places search services share the same <u>usage limits</u> (https://developers.google.com/places/web-service/usage). However, the Text Search service is subject to a 10-times multiplier. That is, each Text Search request that you make will count as 10 requests against your quota. If you've purchased the Google Places API as part of your Google Maps APIs Premium Plan contract, the multiplier may be different. Please refer to the <u>Google Maps APIs Premium Plan documentation</u> (https://developers.google.com/maps/premium/usage-limits#places-api-usage) for details.

A Text Search request is an HTTP URL of the following form:

```
https://maps.googleapis.com/maps/api/place/textsearch/output?parameters
```

where `output` may be either of the following values:

- `json` (recommended) indicates output in JavaScript Object Notation (JSON)
- `xml` indicates output as XML

Certain parameters are required to initiate a search request. As is standard in URLs, all parameters are separated using the ampersand (`&`) character.

Required parameters

- `query` — The text string on which to search, for example: "restaurant" or "123 Main Street". The Google Places service will return candidate matches based on this string and order the results based on their perceived relevance. This parameter becomes optional if the `type` parameter is also used in the search request.
- `key` — Your application's API key (https://support.google.com/googleapi/answer/6158862). This key identifies your application for purposes of quota management and so that places added (https://developers.google.com/places/web-service/add-place) from your application are made immediately available to your app. See Get a key for Google Places API Web Service (https://developers.google.com/places/web-service/get-api-key) to see how to create an API Project and obtain your key.

Optional parameters

- `region` — The region code, specified as a ccTLD (https://en.wikipedia.org/wiki/CcTLD) (country code top-level domain) two-character value. Most ccTLD codes are identical to ISO 3166-1 codes, with some exceptions. This parameter will only influence, not fully restrict, search results. If more relevant results exist outside of the specified region, they may be included. When this parameter is used, the country name is omitted from the resulting `formatted_address` for results in the specified region.
- `location` — The latitude/longitude around which to retrieve place information. This must be specified as *latitude,longitude*. If you specify a `location` parameter, you must also specify a `radius` parameter.
- `radius` — Defines the distance (in meters) within which to bias place results. The maximum allowed radius is 50 000 meters. Results inside of this region will be ranked higher than results outside of the search circle; however, prominent results from outside of the search radius may be included.
- `language` — The language code, indicating in which language the results should be returned, if possible. See the list of supported languages (https://developers.google.com/maps/faq#languagesupport) and their codes. Note that we often update supported languages so this list may not be exhaustive.
- `minprice` and `maxprice` (*optional*) — Restricts results to only those places within the specified price level. Valid values are in the range from 0 (most affordable) to 4 (most expensive), inclusive. The exact amount indicated by a specific value will vary from region to region.
- `opennow` — Returns only those places that are open for business at the time the query is sent. Places that do not specify opening hours in the Google Places database will not be returned if you include this parameter in your query.
- `pagetoken` — Returns the next 20 results from a previously run search. Setting a `pagetoken` parameter will execute a search with the same parameters used previously — all parameters other than `pagetoken` will be ignored.
- `type` — Restricts the results to places matching the specified type. Only one type may be specified (if more than one type is provided, all types following the first entry are ignored). See the list of supported types (https://developers.google.com/places/web-service/supported_types).

You may bias results to a specified circle by passing a `location` and a `radius` parameter. This will instruct the Google Places service to prefer showing results within that circle. Results outside the defined area may still be displayed. Biasing results to a region or circle is recommended to improve relevance of results for otherwise ambiguous queries.

Note for Google Maps APIs Premium Plan customers: You must include an API key in your requests. You should **not** include a `client` or `signature` parameter with your requests.

Text search examples

Note: In these examples, you need to replace the `key` with your own API key in order for the request to work in your application.

Example 1: The following example shows a search for restaurants near Sydney.

```
https://maps.googleapis.com/maps/api/place/textsearch/xml?query=restaurants+in+Sydney&key=YOUR_API_KEY
```

Example 2: The following example shows a search for an incomplete address, in this case, a street address that does not include a city or state or country.

```
https://maps.googleapis.com/maps/api/place/textsearch/json?query=123+main+street&key=YOUR_API_KEY
```

Example 3: The following example shows a search for the same incomplete address in sample 2, and includes `location` and `radius` parameters to bias the results to a region of interest. Compare the results of sample 2 to sample 3.

```
https://maps.googleapis.com/maps/api/place/textsearch/json?query=123+main+street&location=42.3675294,-71.186
```

Radar Search Requests (deprecated)

Notice: Radar search is deprecated as of June 30, 2018. After that time, this feature will no longer be available.

The Google Places API Radar Search Service allows you to search for up to 200 places at once, but with less detail than is typically returned from a Text Search or Nearby Search request. With Radar Search, you can create applications that help users identify specific areas of interest within a geographic area.

The search response will include up to 200 places, and will include only the following information about each place:

- The `geometry` field containing geographic coordinates.
- The `place_id`, which you can use in a Place Details request to get more information about the place. For more information about place IDs, see the place ID overview (https://developers.google.com/places/web-service/place-id).

A Radar Search request is an HTTP URL of the following form:

```
https://maps.googleapis.com/maps/api/place/radarsearch/output?parameters
```

where `output` may be either of the following values:

- `json` (recommended) indicates output in JavaScript Object Notation (JSON)
- `xml` indicates output as XML

Certain parameters are required to initiate a search request. As is standard in URLs, all parameters are separated using the ampersand (`&`) character.

Required parameters

- `key` — Your application's API key (https://support.google.com/googleapi/answer/6158862). This key identifies your application for purposes of quota management and so that places added (https://developers.google.com/places/web-service/add-place) from your application are made immediately available to your app. See Get a key for Google Places API Web Service (https://developers.google.com/places/web-service/get-api-key) to see how to create an API Project and obtain your key.
- `location` — The latitude/longitude around which to retrieve place information. This must be specified as *latitude,longitude*.
- `radius` — Defines the distance (in meters) within which to return place results. The maximum allowed radius is 50 000 meters.
- A Radar Search request must include at least one of `keyword`, `name`, or `type`.

Optional parameters

- `keyword` — A term to be matched against all content that Google has indexed for this place, including but not limited to name, type, and address, as well as customer reviews and other third-party content.
- `language` — The language code, indicating in which language the results should be returned, if possible. Searches are also biased to the selected language; results in the selected language may be given a higher ranking. See the list of supported languages (https://developers.google.com/maps/faq#languagesupport) and their codes. Note that we often update supported languages so this list may not be exhaustive.
- `minprice` and `maxprice` (*optional*) — Restricts results to only those places within the specified price level. Valid values are in the range from 0 (most affordable) to 4 (most expensive), inclusive. The exact amount indicated by a specific value will vary from

region to region.

- **name** — A term to be matched against all content that Google has indexed for this place. Equivalent to `keyword`. The `name` field is no longer restricted to place names. Values in this field are combined with values in the `keyword` field and passed as part of the same search string. We recommend using only the `keyword` parameter for all search terms.

- **opennow** — Returns only those places that are open for business at the time the query is sent. Places that do not specify opening hours in the Google Places database will not be returned if you include this parameter in your query.

- **type** — Restricts the results to places matching the specified type. Only one type may be specified (if more than one type is provided, all types following the first entry are ignored). See the <u>list of supported types</u> (https://developers.google.com/places/web-service/supported_types).

Note for Google Maps APIs Premium Plan customers: You must include an API key in your requests. You should **not** include a `client` or `signature` parameter with your requests.

Radar search examples

Note: In these examples, you need to replace the `key` with your own API key in order for the request to work in your application.

Example 1: The following example returns a list of museums near London, England.

```
https://maps.googleapis.com/maps/api/place/radarsearch/json?location=51.503186,-0.126446&radius=5000&type=mu
```

Example 2: Using a combination of the `keyword` and `type` parameters, you can perform more precise queries. The following example shows restaurants and cafes in Paris that users have described as vegetarian.

```
https://maps.googleapis.com/maps/api/place/radarsearch/json?location=48.859294,2.347589&radius=5000&type=caf
```

Search Responses

Search responses are returned in the format indicated by the `output` flag within the URL request's path.

The following example shows a **Nearby Search** response. A Text Search response is similar, except that it returns a `formatted_address` instead of a `vicinity` property. A Radar Search includes only limited fields, as described <u>above</u> (#RadarSearchRequests).

JSON	XML

```
{
   "html_attributions" : [],
   "results" : [
      {
         "geometry" : {
            "location" : {
               "lat" : -33.870775,
               "lng" : 151.199025
            }
         },
         "icon" : "http://maps.gstatic.com/mapfiles/place_api/icons/travel_agent-71.png",
         "id" : "21a0b251c9b8392186142c798263e289fe45b4aa",
         "name" : "Rhythmboat Cruises",
         "opening_hours" : {
            "open_now" : true
         },
         "photos" : [
            {
               "height" : 270,
               "html_attributions" : [],
               "photo_reference" : "CnRnAAAAF-LjFR1ZV93eawe1cU_3QNMCNmaGkowY7CnOf-kcNmPhNnPEG9W979jOuJJ1sGr75rhD
               "width" : 519
            }
         ],
         "place_id" : "ChIJyWEHuEmuEmsRm9hTkapTCrk",
```

```
        "scope" : "GOOGLE",
        "alt_ids" : [
           {
               "place_id" : "D9iJyWEHuEmuEmsRm9hTkapTCrk",
               "scope" : "APP"
           }
        ],
        "reference" : "CoQBdQAAAFSiijw5-cAV68xdf2018pKIZ0seJh03u9h9wk_lEdG-cP1dWvp_QGS4SNCBMk_fB06YRsfMrNkINtPe
        "types" : [ "travel_agency", "restaurant", "food", "establishment" ],
        "vicinity" : "Pyrmont Bay Wharf Darling Dr, Sydney"
     },
     {
        "geometry" : {
           "location" : {
              "lat" : -33.866891,
              "lng" : 151.200814
           }
        },
        "icon" : "http://maps.gstatic.com/mapfiles/place_api/icons/restaurant-71.png",
        "id" : "45a27fd8d56c56dc62afc9b49e1d850440d5c403",
        "name" : "Private Charter Sydney Habour Cruise",
        "photos" : [
           {
               "height" : 426,
               "html_attributions" : [],
               "photo_reference" : "CnRnAAAAL3n0Zu3U6fseyPl8URGKD49aGB2Wka7CKDZfamoGX2ZTLMBYgTUshjr-MXc0_O2BbvlU
               "width" : 640
           }
        ],
        "place_id" : "ChIJqwS6fjiuEmsRJAMiOY9MSms",
        "scope" : "GOOGLE",
        "reference" : "CpQBhgAAAFN27qR_t5oSDKPUzjQIeQa3lrRpFTm5alW3ZYbMFm8k10ETbISfK9S1nwcJVfrP-bjra7NSPuhaRulx
        "types" : [ "restaurant", "food", "establishment" ],
        "vicinity" : "Australia"
     },
     {
        "geometry" : {
           "location" : {
              "lat" : -33.870943,
              "lng" : 151.190311
           }
        },
        "icon" : "http://maps.gstatic.com/mapfiles/place_api/icons/restaurant-71.png",
        "id" : "30bee58f819b6c47bd24151802f25ecf11df8943",
        "name" : "Bucks Party Cruise",
        "opening_hours" : {
           "open_now" : true
        },
        "photos" : [
           {
               "height" : 600,
               "html_attributions" : [],
               "photo_reference" : "CnRnAAAA48AX5MsHIMiuipON_Lgh97hPiYDFkxx_vnaZQMOcvcQwYN92o33t5RwjRpOue5R47Ajf
               "width" : 800
           }
        ],
        "place_id" : "ChIJLfySpTOuEmsRsc_JfJtljdc",
        "scope" : "GOOGLE",
        "reference" : "CoQBdQAAANQSThnTekt-UokiTiX3oUFT6YDfdQJIG0ljlQnkLfWefcKmjxax0xmUpWjmpWdOsScl9zSyBNImmrTC
        "types" : [ "restaurant", "food", "establishment" ],
        "vicinity" : "37 Bank St, Pyrmont"
     },
     {
        "geometry" : {
           "location" : {
              "lat" : -33.867591,
              "lng" : 151.201196
           }
        },
        "icon" : "http://maps.gstatic.com/mapfiles/place_api/icons/travel_agent-71.png",
        "id" : "a97f9fb468bcd26b68a23072a55af82d4b325e0d",
```

```
            "name" : "Australian Cruise Group",
            "opening_hours" : {
               "open_now" : true
            },
            "photos" : [
               {
                  "height" : 242,
                  "html_attributions" : [],
                  "photo_reference" : "CnRnAAAABjeoPQ7NUU3pDitV4Vs0BgP1FLhf_iCgStUZUr4ZuNqQnc5k43jbvjKC2hTGM8SrmdJY
                  "width" : 200
               }
            ],
            "place_id" : "ChIJrTLr-GyuEmsRBfy61i59si0",
            "scope" : "GOOGLE",
            "reference" : "CoQBeQAAAFvf12y8veSQMdIMmAXQmus1zqkgKQ-O2KEX0Kr47rIRTy6HNsyosVl0CjvEBulIu_cujrSOgICdcxNi
            "types" : [ "travel_agency", "restaurant", "food", "establishment" ],
            "vicinity" : "32 The Promenade, King Street Wharf 5, Sydney"
         }
      ],
      "status" : "OK"
}
```

A JSON response contains up to four root elements:

- "`status`" contains metadata on the request. See Status Codes (#PlaceSearchStatusCodes) below.
- "`results`" contains an array of places, with information about each. See Search Results (#PlaceSearchResults) for information about these results. The Places API returns up to 20 `establishment` results per query. Additionally, `political` results may be returned which serve to identify the area of the request.
- `html_attributions` contain a set of attributions about this listing which must be displayed to the user.
- `next_page_token` contains a token that can be used to return up to 20 additional results. A `next_page_token` will not be returned if there are no additional results to display. The maximum number of results that can be returned is 60. There is a short delay between when a `next_page_token` is issued, and when it will become valid.

See Processing JSON with Javascript (https://developers.google.com/maps/web-services/overview#ParsingJSON) for help parsing JSON responses.

Status Codes

The "`status`" field within the search response object contains the status of the request, and may contain debugging information to help you track down why the request failed. The "`status`" field may contain the following values:

- `OK` indicates that no errors occurred; the place was successfully detected and at least one result was returned.
- `ZERO_RESULTS` indicates that the search was successful but returned no results. This may occur if the search was passed a `latlng` in a remote location.
- `OVER_QUERY_LIMIT` indicates that you are over your quota.
- `REQUEST_DENIED` indicates that your request was denied, generally because of lack of an invalid `key` parameter.
- `INVALID_REQUEST` generally indicates that a required query parameter (`location` or `radius`) is missing.
- `UNKNOWN_ERROR` indicates a server-side error; trying again may be successful.

Error Messages

When the Google Places service returns a status code other than `OK`, there may be an additional `error_message` field within the search response object. This field contains more detailed information about the reasons behind the given status code.

Note: This field is not guaranteed to be always present, and its content is subject to change.

Search Results

When the Google Places service returns JSON results from a search, it places them within a `results` array. Even if the service returns no results (such as if the `location` is remote) it still returns an empty `results` array. XML responses consist of zero or more `<result>` elements.

Each element of the `results` array contains a single result from the specified area (`location` and `radius`), ordered by prominence.

The result may also contain attribution information which must be displayed to the user. This is an example of an attribution in JSON format:

```
"html_attributions" : [
      "Listings by \u003ca href=\"http://www.example.com/\"\u003eExample Company\u003c/a\u003e"
],
```

This is an attribution in XML format:

```
<html_attribution>Listings by <a href="http://www.example.com/">Example Company</a></html_attribution>
```

Each result within the `results` array may contain the following fields:

- `icon` contains the URL of a recommended icon which may be displayed to the user when indicating this result.
- `geometry` contains geometry information about the result, generally including the `location` (geocode) of the place and (optionally) the `viewport` identifying its general area of coverage.
- `name` contains the human-readable name for the returned result. For `establishment` results, this is usually the business name.
- `opening_hours` may contain the following information:
  - `open_now` is a boolean value indicating if the place is open at the current time.
- `photos[]` — an array of `photo` objects, each containing a reference to an image. A Place Search will return at most one `photo` object. Performing a Place Details request on the place may return up to ten photos. More information about Place Photos and how you can use the images in your application can be found in the Place Photos (https://developers.google.com/places/web-service/photos) documentation. A `photo` object is described as:
  - `photo_reference` — a string used to identify the photo when you perform a Photo request.
  - `height` — the maximum height of the image.
  - `width` — the maximum width of the image.
  - `html_attributions[]` — contains any required attributions. This field will always be present, but may be empty.
- `place_id` — a textual identifier that uniquely identifies a place. To retrieve information about the place, pass this identifier in the `placeId` field of a Places API request. For more information about place IDs, see the place ID overview (https://developers.google.com/places/web-service/place-id).
- `scope` — Indicates the scope of the `place_id`. The possible values are:
  - `APP`: The place ID is recognised by your application only. This is because your application added the place, and the place has not yet passed the moderation process.
  - `GOOGLE`: The place ID is available to other applications and on Google Maps.

  Note: The `scope` field is included only in Nearby Search results and Place Details results. You can only retrieve app-scoped places via the Nearby Search and the Place Details requests. If the `scope` field is not present in a response, it is safe to assume the scope is `GOOGLE`.
- `alt_ids` — An array of zero, one or more alternative place IDs for the place, with a scope related to each alternative ID. Note: This array may be empty or not present. If present, it contains the following fields:
  - `place_id` — The most likely reason for a place to have an alternative place ID is if your application adds a place and receives an application-scoped place ID, then later receives a Google-scoped place ID after passing the moderation process.
  - `scope` — The scope of an alternative place ID will always be `APP`, indicating that the alternative place ID is recognised by your application only.

  For example, let's assume your application adds a place and receives a `place_id` of `AAA` for the new place. Later, the place passes the moderation process and receives a Google-scoped `place_id` of `BBB`. From this point on, the information for this place will contain:

```
"results" : [
  {
    "place_id" : "BBB",
    "scope" : "GOOGLE",
    "alt_ids" : [
      {
```

```
            "place_id" : "AAA",
            "scope" : "APP",
        }
    ],
  }
]
```

- `price_level` — The price level of the place, on a scale of 0 to 4. The exact amount indicated by a specific value will vary from region to region. Price levels are interpreted as follows:
  - `0` — Free
  - `1` — Inexpensive
  - `2` — Moderate
  - `3` — Expensive
  - `4` — Very Expensive
- `rating` contains the place's rating, from 1.0 to 5.0, based on aggregated user reviews.
- `types[]` contains an array of feature types describing the given result. See the list of supported types (https://developers.google.com/places/web-service/supported_types#table2). XML responses include multiple `<type>` elements if more than one type is assigned to the result.
- `vicinity` contains a feature name of a nearby location. Often this feature refers to a street or neighborhood within the given results. The `vicinity` property is only returned for a Nearby Search (#PlaceSearchRequests).
- `formatted_address` is a string containing the human-readable address of this place. Often this address is equivalent to the "postal address". The `formatted_address` property is only returned for a Text Search (#TextSearchRequests).
- `permanently_closed` is a boolean flag indicating whether the place has permanently shut down (value `true`). If the place is not permanently closed, the flag is absent from the response.

From our Terms of Service

If you change Google data, let your users know

Show Google data as it is provided to you — don't change or modify it secretly. For example, no mixing data from Google and other providers unless you make it clearly visible to the user.

EARN MORE (HTTPS://DEVELOPERS.GOOGLE.COM/MAPS/TERMS#SECTION_10_3)

Accessing Additional Results

By default, each Nearby Search or Text Search returns up to 20 `establishment` results per query; however, each search can return as many as 60 results, split across three pages. If your search will return more than 20, then the search response will include an additional value — `next_page_token`. Pass the value of the `next_page_token` to the `pagetoken` parameter of a new search to see the next set of results. If the `next_page_token` is null, or is not returned, then there are no further results. There is a short delay between when a `next_page_token` is issued, and when it will become valid. Requesting the next page before it is available will return an `INVALID_REQUEST` response. Retrying the request with the same `next_page_token` will return the next page of results.

For example, in the query below, we search for restaurants near Darling Harbour, in Sydney Australia, and rank the results by distance. You can see that the response contains a `next_page_token` property.

```
https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=-33.8670522,151.1957362&rankby=distanc
```

```
{
   "html_attributions" : [],
   "next_page_token" : "CpQCAgEAAFxg8o-eU7_uKn7Yqjana-HQIx1hr5BrT4zBaEko29ANsXtp9mrqN0yrKWhf-y2PUpHRLQb1GT-mtxNcXc
   "results" : [
      {
         "geometry" : {
            "location" : {
               "lat" : -33.867217,
               "lng" : 151.195939
            }
         },
         "icon" : "http://maps.gstatic.com/mapfiles/place_api/icons/cafe-71.png",
         "id" : "7eaf747a3f6dc078868cd65efc8d3bc62fff77d7",
         "name" : "Biaggio Cafe - Pyrmont",
         "opening_hours" : {
            "open_now" : true
         },
         "photos" : [
            {
               "height" : 600,
               "html_attributions" : [],
               "photo_reference" : "CnRnAAAAmWmj0BqA0Jorm1_vjAvx1n6c7ZNBxyY-U9x99-oNyOxvMjDlo2npJzyIq7c3EK1YyoNXdN
               "width" : 900
            }
      }
```

```
      ],
      "place_id" : "ChIJIfBAsjeuEmsRdgu9Pl1Ps48",
      "scope" : "GOOGLE",
      "price_level" : 1,
      "rating" : 3.4,
      "reference" : "CoQBeAAAAGu0wNJjuZ40DMrRe3mpn7fhlfIK1mf_ce5hgkhfM79u-lqy0G2mnmcueTq2JGWu9wsgS1ctZDHTY_pcqF
      "types" : [ "cafe", "bar", "restaurant", "food", "establishment" ],
      "vicinity" : "48 Pirrama Rd, Pyrmont"
   },
   {
      "geometry" : {
         "location" : {
            "lat" : -33.866786,
            "lng" : 151.195633
         }
      },
      "icon" : "http://maps.gstatic.com/mapfiles/place_api/icons/generic_business-71.png",
      "id" : "3ef986cd56bb3408bc1cf394f3dad9657c1d30f6",
      "name" : "Doltone House",
      "photos" : [
         {
            "height" : 1260,
            "html_attributions" : [ "From a Google User" ],
            "photo_reference" : "CnRwAAAAeM-aLqAm573T44qnNe8bGMkr_BOh1MOVQaA9CCggqtTwuGD1rjsviMyueX_G4-mabgH41\
            "width" : 1890
         }
      ],
      "place_id" : "ChIJ5xQ7szeuEmsRs6Kj7YFZE9k",
      "scope" : "GOOGLE",
      "reference" : "CnRvAAAA22k1PAGyDxAgHZk6ErHh_h_mLUK_8XNFLvixPJHXRbCzg-gw1ZxdqUwA_8EseDuEZKolBs82orIQH4m6-a
      "types" : [ "food", "establishment" ],
      "vicinity" : "48 Pirrama Rd, Pyrmont"
   },
   {
      "aspects" : [
         {
            "rating" : 23,
            "type" : "overall"
         }
      ],
      ...
   ],
   "status" : "OK"
}
```

To see the next set of results you can submit a new query, passing the result of the `next_page_token` to the `pagetoken` parameter. For example:

```
https://maps.googleapis.com/maps/api/place/nearbysearch/json?pagetoken=CpQCAgEAAFxg8o-eU7_uKn7Yqjana-HQIx1hr
```

Setting `pagetoken` will cause any other parameters to be ignored. The query will execute the same search as before, but will return a new set of results. You can request a new page up to two times following the original query. Each page of results must be displayed in turn. Two or more pages of search results should not be displayed as the result of a single query. Note that each search counts as a single request against your usage limits.

The **sensor** Parameter

The Google Places API previously required that you include the `sensor` parameter to indicate whether your application used a sensor to determine the user's location. This parameter is no longer required.