Area Visitors API

API Version: 1.1

Overview

Unacast processes global geo-location data, called observations, creating pseudonymized, cleansed, merged, and enriched data sets. This observation data can be used for anything from foot traffic analysis to fraud detection to identifying variances in activity levels and performing real estate area analysis.

The Area Visitors API returns Pseudonymized Registration IDs observed inside a specific area and time range, and is based on the Unacast observations data set. observations contain information about a Pseudonymized Registration ID at a point in time and have the Pseudonymized Registration ID, timestamp, and location (as longitude and latitude), among other attributes.

The Area Visitors API can also, instead of returning the list of Pseudonymized Registration IDs that match the query parameters, can aggregate these Pseudonymized Registration ID counts and return the trends of Pseudonymized Registration ID counts over days, weeks, and months covering the selected time range. In addition to an overall count, the Pseudonymized Registration ID trends can also be returned for the CBG, county, or zip code in which the Pseudonymized Registration ID was recorded.

The purpose of the Area Visitors API is to obtain Pseudonymized Registration IDs that have observations:

- within one or more of the specified spatial areas (currently polygons, multi-polygons, points, or line-strings)
- during a specified time frame
- with desired forensic flags applied

This is performed by first finding observations that meet the specific criteria (polygon(s), time range, forensic flags), and then creating a list of Pseudonymized Registration IDs with these observations.

Endpoints

The following endpoints are covered in this document.

- /areas/devices (POST)
- /areas/devices/trends (POST)

Request Content-Type

The API requires all request bodies to be JSON. As a result, the `Content-Type` header for POST requests must be supplied and set to `application/json`.

```
Content-Type: application/json
```

Response Content-Type

The API only returns JSON for all response payloads. As a result, the `Accept` header must be set to `application/json` for all requests.

```
Accept: application/json
```

If an unknown or unsupported content type is specified, then a “406 Not Acceptable” response will be returned.

Accept Encoding

By default, the API returns each response in plain JSON but also supports encoding the response if an Accept-Encoding header is supplied. This can be useful as large searches can have responses that are greater than 10MB in size as plain JSON. The only encoding currently supported is gzip.

```
Accept-Encoding: gzip
```

Authentication

In order to make requests using the API, a valid API key must be set in the 'Authorization' header of the request. Please contact your API sales rep to obtain an API key. A sample header is shown below:

```
Authorization: 21867b4e-2e47-418e-9877-debce2859f3e
```

Environments

The API is available in a sandbox and production environment. The sandbox environment is used to work with users to test improvements and modifications. As a result, the sandbox environment will sometimes not behave the same as production and can be less stable.

Environment	URL Format	Description
Production	https://api.gravyanalytics.com/v1.1/	Has the most stable version of the API.
Staging	https://api.gravyanalytics.com/v1.1-staging/	Consists of the production API plus soon-to-be-production improvements to the API.

Endpoint Versioning

The API incorporates versioning into the base URL path in order to support iterative development and the functionality of applications dependent on legacy endpoints. The current version is `v1.1` and therefore the base URL should look like the following:

```
https://api.gravyanalytics.com/v1.1/
```

Status Codes

Before attempting to parse a response from the API, the HTTP response status code should be examined. A status code of 200 indicates that the request completed successfully. All other status code values indicate there was a problem processing the request.

Status Code	Description
200	Indicates that the supplied request was processed without a failure. It is possible to have a 200 response with an empty result set if no entries meet the search criteria.
400	Indicates that the supplied request was not valid. The most likely causes are malformed JSON, a missing parameter, or an illegal value for a parameter.
401	Indicates that there was an issue verifying the API key. Likely causes are an improperly formatted 'Authorization' header, a lack of an API key in the 'Authorization' header, the API key is incorrect/unauthorized, the API key does not have permission for the specific endpoint requested, or the 'TempSecurityToken' header value is incorrect.
404	Indicates that the supplied request was for a non-existent URL. This will happen when attempting to access an endpoint that does not exist or when trying to retrieve an entry by id and no entry is found with the given id.
405	Indicates that the supplied request used an incorrect HTTP method. This occurs when the client attempts to use a method that is not allowed for the requested endpoint, such as sending a POST request to an endpoint that only accepts GET requests.
406	Indicates that the content type specified by the 'Accept' header is unknown or unaccepted.
408	Indicates that the request took too long to process and was cancelled. The max processing time for a request is 180 seconds. This is more likely to occur when searching over a long time period and with many search objects.
410	Indicates that the requested version of the API no longer exists. Please consult your latest API documentation for the currently supported endpoint versions.
429	Indicates that the request was denied because too many requests have been made by the API key in the current day. The start and end of each day is at 00:00:00 UTC. Daily request limits are per API key and may be increased as necessary. Please contact your API representative if you wish to have your rate limits modified.

Status Code	Description
500	Indicates that the supplied request resulted in an error on the server. In the event of a 500 status code response, please notify your API representative.

Forensic Indicators

In general, location data is collected with varying levels of precision and the data feeds typically also contain a substantial percentage of fraudulent data. This combination creates a need for a thorough data cleansing process not only to identify and remove the problematic data, but also to identify data precision so that you use the right subset of data for the right use case.

During the cleansing process, Unacast’s analytical and ML engine identifies the quality and validity of each observation, assigning it one or several forensic flags. This will give you insights into the collection method of Unacast observations, precision, and several quality indicators. You can use these indicators to decide which observations to include or exclude to get the data set that works best for your use case. For example, depending on your use case, you may want to include or exclude observations that were collected when Pseudonymized Registration IDs were moving, or include or exclude location signals with a lower horizontal accuracy.

The API call thus allows you to specify the observation’s forensic indicators that should be included or excluded. By default, all Pseudonymized Registration IDs associated with any observation inside the search features(s) will be returned, irrespective of any forensic flags the observations might have.

Please refer to the “Using Unacast Forensics To Build Analytics and Products” white paper for an overview of our forensic indicators and the “Unacast Forensic Flag Index Chart” for a technical description of individual forensic flags. The latter document also defines how to calculate individual forensic indicator values.

Retrieving Results Directly vs as Export

In addition to the search parameters of each search endpoint, there are also parameters that control if the results are returned as part of the JSON response, `responseType` = `DIRECT`, or as files available via AWS S3 pre-signed URLs, `responseType` = `EXPORT`. Depending on the endpoint, the search parameter and result limits may differ for direct and export requests and responses. Export search requests, once validated, run as background tasks within the API and therefore will often return in a few seconds whereas the same request as direct could take longer to return. Additionally, because the export task runs inside the API and is not tied to a request with a 300-second timeout, an export task can run for as long as it needs to in order to return. Note, that export tasks are still subject to query timeouts and therefore can still timeout if an internal query were to take too long.

When the results of a request are exported, they are stored as files in AWS S3 and made available to the user via S3 pre-signed URLs that can be used to download the files. Depending on the endpoint and the size of its largest responses, the results may be collected in one file or may be broken up into multiple files. The number of files may vary by request and should not be relied upon to be a consistent number. These files will be compressed or uncompressed PSV files with a header on each file whose column names are dependent on the endpoint and results. The file extension of the files will be `.psv` if uncompressed or `.psv.gz` if compressed. The status, and results when ready, of an export request is available via the `requestStatus/{requestID}` (GET) endpoint. See separate documentation for more on the `requestStatus/{requestID}` endpoint.

The following request and response fields pertain to `EXPORT` requests/responses. Request parameter `compressOutputFiles`, default true, controls whether the result PSV files are GZIP compressed or uncompressed. At this time no ordering of the results is guaranteed. Response field `requestid`, a UUIDv4 identifying the request, serves only a debugging role for direct requests and for export request is used by the `requestStatus/{requestID}` endpoint to check the status of the request and to get its results when ready. Additionally, there is an `message` response field acknowledging that the request is being processed as an export task.

Export functionality is an optional API enhancement that is enabled on a per-API-key basis and disabled by default. Please speak to your API sales rep if you would like to utilize export functionality.

/areas/devices

POST

Summary: Retrieve Pseudonymized Registration IDs and Counts For Search Areas

Description: This endpoint exists in order to search one or more geographic areas for the Pseudonymized Registration IDs seen to have observation data inside. The response includes a count of Pseudonymized Registration IDs found as well as an optional list of Pseudonymized Registration ID identifiers for those Pseudonymized Registration IDs. The Pseudonymized Registration ID identifies can either be Unacast registration IDs or Pseudonymized Registration ID advertiser IDs depending on the API key's settings.

Search requirements and limitations are described below:

Some search parameters can be specified only as query parameters, only in the request body, or in either location. If a search parameter is specified as a query parameter, it overrides what is specified in the request body.

The request body for this endpoint is required to be a GeoJSON FeatureCollection. Each feature in the FeatureCollection must have a geometry of type polygon, multipolygon, point, or linestring. If a point or linestring is specified, the feature is also required to have the property `radiusInMeters`. This property is used for converting points and linestrings to polygons for processing. At most 20 features can be specified in a single request. A search feature's geometry must be valid and not fold across itself. It is okay if two or more search features overlap.

By default, an API key is limited to making requests in these countries: USA, CAN, MEX. If you would like to search outside these countries, please reach out to your API sales representative.

Feature-Specific Search Parameters

Each feature can have its own search parameters as specified in the feature's `properties`. See model information below for a list of search parameters that can be specified per feature. As a reminder, any search parameters specified as query parameters will override those specified for a feature. Additionally, there are some search parameters that are not feature specific, and these can be specified either as query parameters or in the `properties` of the FeatureCollection's `crs` object. When metrics and results are returned for a `resultType` = `DIRECT` request, each metric or result will be returned either in the `properties` of a feature if the metric/result is specific to a feature, or in the `properties` of the `crs` object if not specific to a feature.

Along with the search features, the user can also specify a `startDateTimeEpochMS` and `endDateTimeEpochMS` to limit the time range of the search. Both parameters are optional. The start date-time must be before the end date-time, `startDateTimeEpochMS` must be at or after the start of the day 3 years ago, and `endDateTimeEpochMS` must be equal to or before the time of the request. If you want to use the current time as the end date, it is best to not supply an end date as the default end date value is the current time. If no start date-time is supplied, the default behavior is for start date to be the beginning of the day seven days before `endDateTimeEpochMS`. The default behavior for endDate is to be the current time. For example if the time of the request is `2022-03-10T12:34:56Z`, then startDate defaults to the equivalent of `2022-03-03T00:00:00Z` and endDate defaults to the equivalent of `2022-03-10T12:34:56Z`. `endDateTimeEpochMS` must be within 90 days from `startDateTimeEpochMS` disregarding time of day. For example, a `startDateTimeEpochMS` of 1597622400000, equal to `2020-08-17T00:00:00Z`, and `endDateTimeEpochMS` of 1605443696000, equal to `2020-11-15T12:34:56Z`, is allowed since the day components are 90 days apart. The `startDateTime` and `endDateTime` parameters allow users to specify time parameters in a human-readable format instead of epoch milliseconds. The time format is ISO 8601, which includes a date, time, and timezone. An example of this format is `2020-05-12T13:45:22.893+04:00`, this timestamp's equivalent time in epoch milliseconds is `1589276722893`. `startDateTime` is mutually exclusive with `startDateTimeEpochMS` and `endDateTime` is mutually exclusive with `endDateTimeEpochMS`.

While `startDateTimeEpochMS` and `endDateTimeEpochMS` enable a user to specify the overall time range to search, `startTimeOfDay` and `endTimeOfDay` enable a user to specify a period within each day to search. For example a user may want to search for all observations that only occurred during business hours and therefore would set `startTimeOfDay` to "08:00:00Z" and `endTimeOfDay` to "18:00:00Z". Overnight time periods are supported as well. For example a user could search from "22:00:00Z" to "02:00:00Z" if they were looking for observations that occurred in the middle of the night. Both start time of day and end time of day are supplied as strings and must be in format `HH:mm:ss'Z'`. Again, all time parameters are interpreted as being in UTC.

The following search parameters allow the user to filter the observations returned based on their flags values. These search parameters are `includeFlags`, `includeExactFlags`, `excludeFlags`, and `excludeExactFlags`. All are 64-bit integers (longs) and control whether an observation must have, or not have, specific flags in order to be counted. These four default to null with no effect applied.

Search parameter `deviceCountOnly` controls whether just the Pseudonymized Registration ID counts are returned for the feature or if the Pseudonymized Registration ID identifiers are returned as well.

For `DIRECT` requests, `returnDeviceCountByGeoHash` controls whether metrics are returned per geo hash in the `geoHashDeviceCount` response field. When a search feature is submitted to the API, the API calculates the geo-hashes needed to cover the geometry and uses those to query the database. The API uses the most precise geo hashes it can while keeping the number of geo hashes to a manageable amount. As a result, the precision and size of the geo hashes used scales with the size of the search feature. Alongside the overall Pseudonymized Registration ID count returned, the `geoHashDeviceCount` metrics, when combined with `geoHashWidth` and `geoHashHeight`, provide a quick way to create heatmap visualizations.

Cross-Feature Search Parameters

The search parameters that are not feature specific are: `inMin` and `placeIdentifier`. `inMin` controls how many search features a Pseudonymized Registration ID must be seen in for the Pseudonymized Registration ID to be counted/returned in the results. In order to keep the result format consistent across features, the `placeIdentifier` search parameter must be specified as a query parameter or in the `properties` of the `crs`.

The query parameter `placeIdentifier` controls whether additional geospatial data is returned for feature and is a comma-separated list of strings with acceptable values being `NONE`, `ZIPCODE`, `COUNTY`, and `CBG` (census block group). The additional data is available only for the USA. If the list contains `NONE`, no place identifier information will be returned. `NONE` is mutually exclusive with all other options. If the list contains `ZIPCODE`, each feature will have a list of overlapping zip codes and the Pseudonymized Registration ID count from the original Pseudonymized Registration ID search that appears in each zip code. If the list contains `COUNTY`, each feature will have a list of overlapping county FIPS (Federal Information Processing Standards) code, county name, state FIPS (Federal Information Processing Standards) code, and state name along with the Pseudonymized Registration ID count from the original Pseudonymized Registration ID search that appears in each county. If the list contains `CBG`, each feature will have a map of overlapping census block groups and the Pseudonymized Registration ID count from the original Pseudonymized Registration ID search that appears in each census block group.

If PSV export is specified and the placeIdentifier is not `NONE`, an additional file type will be available for each placeIdentifier in the `presignedUrlsByDataType` field of the request status endpoint.

Direct Response Fields For `DIRECT` responses, each geoJSON feature will have fields added to its `properties`. These include `deviceCount`, the number of Pseudonymized Registration IDs found for the search feature and other search criteria including the `inMin` parameter. `deviceCount` is distinct from the number of Pseudonymized Registration IDs returned in `Pseudonymized Registration IDs`. The `Pseudonymized Registration IDs` response field is a list of objects containing either a `registrationID` or `advertiserID` field depending on which Pseudonymized Registration ID identifier the API key is configured to return. As a reminder, if `deviceCountOnly` is true for the feature, then `Pseudonymized Registration IDs` will always be empty in the response. Response field `deviceLimitHit` is a boolean that is true when the number of Pseudonymized Registration IDs returned for the search feature equals the limit of 10k. This is useful to quickly determine if the feature's returned list of Pseudonymized Registration IDs was likely truncated or not.

Alongside the response fields described above, the API can also return per-geo-hash information when `returnDeviceCountByGeoHash` is true. In that case, `geoHashDeviceCount` will be returned. All three can be used together to create a heatmap of Pseudonymized Registration ID counts across the search area. The Pseudonymized Registration ID counts returned are impacted by the `inMIn` and other search parameters.

Direct Response Limitations The maximum number of Pseudonymized Registration IDs in the `devices` response field that the API will return is 10k per search feature. When selecting the Pseudonymized Registration IDs to return, the API selects the first 10k per feature by registration ID (even if the request is for advertiser IDs). If a direct request does hit the 10k limit and the user wants the Pseudonymized Registration IDs for all found Pseudonymized Registration IDs, it is recommended that they decrease the size of their search area or time range and/or retrieve their results via export.

Export Response Limitations The maximum number of results in an export is 100 billion, effectively unlimited. If the limit is ever reached, the results returned are not ordered and therefore which subset of results is exported would vary randomly between requests.

Parameters

Name	Located in	Description	Required	Schema

Name	Located in	Description	Required	Schema
responseType	Query or CRS Properties	Optional. Default `DIRECT`. Possible values: `DIRECT` and `EXPORT`. Case insensitive. `DIRECT` indicates to return the results of the search as part of the JSON response body. `EXPORT` indicates that the results should be exported as files made available via download URLs from the `requestStatus/{requestID}` endpoint. `DIRECT` requests wait for the results to be available before returning while `EXPORT` requests run as a background task after being validated. As a result, a `DIRECT` request could take a minute for the results to come back while the same request as an `EXPORT` would return in seconds but the user would have to wait for the results to become available via the `requestStatus/{requestID}` endpoint.	No	string
includeHeaders	Query or CRS Properties	Optional boolean, default true. Specifies whether output psv files include a header row. If true, psv output files will include a header row. If false, psv output files will not include a header row.	No	Boolean
compressOutputFiles	Query or CRS Properties	Optional boolean, default true. Specifies whether output files are gzip compressed. If true, output files are gzip compressed. If false, output files are not compressed.	No	Boolean
exportLimit	Query or CRS Properties	Optional integer, default null. Specifies max number of rows output files will have (not including optional header row). If defined, psv output files will have at most exportLimit rows. If not defined, psv output files have no row limit.	No	number (int64)
startDateTimeEpochMS	Query or Feature Properties	Optional. Default is the beginning of the day 7 days prior to `endDateTimeEpochMS`. The start of the time range to search for, expressed as epoch milliseconds.	No	number (int64)

Name	Located in	Description	Required	Schema
endDateTimeEpochMS	Query or Feature Properties	Optional. Default is time of request. The end of the time range to search for, expressed as epoch milliseconds.	No	number (int64)
startDateTime	Query or Feature Properties	Optional. Default is the beginning of the day 7 days prior to `endDateTime`. The start of the time range to search for, expressed as an ISO-8601 offset timestamp.	No	string (ISO-8601 offset timestamp)
endDateTime	Query or Feature Properties	Optional. Default is time of request. The end of the time range to search for, expressed as ISO-8601 offset timestamp.	No	string (ISO-8601 offset timestamp)
startTimeOfDay	Query or Feature Properties	Optional. Default null. The string representation of the start time of day that observations must be after in order to be counted. An overnight filter is supported where `startTimeOfDay` is after `endTimeOfDay`. A user must supply either both `startTimeOfDay` and `endTimeOfDay` or neither since both are necessary to implement the filter. Format is `HH:mm:ss'Z'`.	No	string
endTimeOfDay	Query or Feature Properties	Optional. Default null. The string representation of the end time of day that observations must be before in order to be counted. An overnight filter is supported where `startTimeOfDay` is after `endTimeOfDay`. A user must supply either both `startTimeOfDay` and `endTimeOfDay` or neither since both are necessary to implement the filter. Format is `HH:mm:ss'Z'`.	No	string

Name	Located in	Description	Required	Schema
includeFlags	Query or Feature Properties	Optional. Default null. A bit array in the form of an int64. This search parameter is used to include observations in the results based on the observation's flags value. The bit indices of `includeFlags` correspond to the bit indices of the observations flags. The bitwise AND of `includeFlags` and a observation's flags value must be non-zero for the observation to be counted in the search results.	No	number (int64)
includeExactFlags	Query or Feature Properties	Optional. Default null. A bit array in the form of an int64. This search parameter is used to include observations in the results based on the observation's flags value. The bit indices of `includeExactFlags` correspond to the bit indices of the observations flags. The `includeExactFlags` value must exactly equal a observation's flags value for the observation to be counted in the search results.	No	number (int64)
excludeFlags	Query or Feature Properties	Optional. Default null. A bit array in the form of an int64. This search parameter is used to filter out observations from the results based on the observation's flags value. The bit indices of `excludeFlags` correspond to the bit indices of the observations flags. When the result of a bitwise AND of `excludeFlags` and a observation's flags value is non-zero, then the observation will not be counted in the search results.	No	number (int64)

Name	Located in	Description	Required	Schema
excludeExactFlags	Query or Feature Properties	Optional. Default null. A bit array in the form of an int64. This search parameter is used to filter out observations from the results based on the observation's flags value. The bit indices of `excludeExactFlags` correspond to the bit indices of the observations flags. When `excludeExactFlags` exactly equals a observation's flags value, then the observation will not be counted in the search results.	No	number (int64)
deviceCountOnly	Query or Feature Properties	Default false. If false just the Pseudonymized Registration ID counts are returned for the feature. If true, the Pseudonymized Registration ID identifiers are returned alongside the Pseudonymized Registration ID counts.	No	boolean
placeIdentifier	Query or CRS Properties	Default `NONE`. A comma-separated list of geospatial regions to return aggregate data for. Acceptable string values are `NONE`, `ZIPCODE`, `COUNTY`, `CBG`. If the list contains `NONE`, no place identifier information will be counted. `NONE` is mutually exclusive with all other options.	No	[string]
returnDeviceCountByGeoHash	Query or Feature Properties	Default false. If true, the `geoHashDeviceCount` and `geoHashWidthHeights` fields are populated per feature.	No	boolean

Name	Located in	Description	Required	Schema
inMin	Query or CRS Properties	Default 0. Specifies the number of search features a Pseudonymized Registration ID must be seen in for it to be counted/returned in the result set. Must be less than or equal to the number of search features in the request. For example, if three search features are in the request, `inMin` must be less than or equal to three. For the previous example, if `inMin` is two, then a Pseudonymized Registration ID must have been seen in at least two of the three search features in order to be returned. If the Pseudonymized Registration ID was seen in all three, it would be returned as well. However, if the Pseudonymized Registration ID was seen in only one of the search features, it would not be returned. A negative or zero `inMin` value behaves the same as an `inMin` value of 1. An `inMin` value of 0 is used by default so empty requests with no specified search features are acceptable.	No	number (int32)
body	Body		Yes	AreasDevicesRequestFeatureCollection

application/json

```
{
    "type": "FeatureCollection",
    "crs": {
        "type": "name",
        "properties": {
            "inMin": 1,
            "placeIdentifier": ["ZIPCODE","COUNTY","CBG"]
        }
    },
    "features": [
        {
            "type": "Feature",
            "properties": {
                "endDateTimeEpochMS": 1706745600000,
                "startDateTimeEpochMS": 1704067200000,
                "startTimeOfDay": null,
                "endTimeOfDay": null,
                "includeRegistrationIDs": [],
                "excludeRegistrationIDs": [],
                "includeFlags": 32768,
                "includeExactFlags": null,
                "excludeFlags": null,
                "excludeExactFlags": null,
                "deviceCountOnly": true,
                "returnDeviceCountByGeoHash": true
            },
            "geometry": {
                "type": "Polygon",
                "coordinates": [
                    [
                        [
                            -77.36331939697266,
                            38.957089496994705
                        ],
                        [
                            -77.35445737838745,
                            38.957089496994705
                        ],
                        [
                            -77.35445737838745,
                            38.96044327386091
                        ],
                        [
                            -77.36331939697266,
                            38.96044327386091
                        ],
                        [
                            -77.36331939697266,
                            38.957089496994705
                        ]
                    ]
                ]
            },
            "id": "place-of-interest-1"
        },
        {
            "type": "Feature",
            "properties": {
                "endDateTimeEpochMS": null,
                "startDateTimeEpochMS": null,
                "startTimeOfDay": null,
                "endTimeOfDay": null,
                "includeRegistrationIDs": [],
                "excludeRegistrationIDs": [],
                "includeFlags": null,
                "includeExactFlags": null,
                "excludeFlags": null,
                "excludeExactFlags": null,
                "deviceCountOnly": false,
                "returnDeviceCountByGeoHash": false
            },
            "geometry": {
                "type": "Polygon",
                "coordinates": [
                    [
                        [
                            -77.35402822494507,
                            38.969431780888165
                        ],
                        [
                            -77.35480606555939,
                            38.96860179949563
                        ],
                        [
                            -77.35380828380585,
                            38.968084620242585
                        ],
                        [
                            -77.35303312540054,
                            38.968931290760246
                        ],
                        [
                            -77.35402822494507,
                            38.969431780888165
                        ]
                    ]
                ]
            },
            "id": "place-of-interest-2"
        }
    ]
}
```

Responses

Note: For the direct and/or export examples below, the results were truncated for brevity.

Code	Description	Schema
200	OK	AreasDevicesResponseFeatureCollection

application/json

```
{
   "type": "FeatureCollection",
   "crs": {
      "type": "name",
      "properties": {
         "inMin": 1,
         "placeIdentifier": [
            "ZIPCODE",
            "COUNTY",
            "CBG"
         ],
         "responseType": "DIRECT",
         "includeHeaders": true,
         "compressOutputFiles": true,
         "exportLimit": null,
         "requestID": "5e6b15e2-6e59-4c24-ad44-9044941a6711",
         "name": "urn:ogc:def:crs:OGC::CRS84",
         "uniqueDeviceCount": 23742,
         "uniqueDeviceInMinCount": 23742
      }
   },
   "features": [
      {
         "type": "Feature",
         "properties": {
            "endDateTimeEpochMS": 1706745600000,
            "startDateTimeEpochMS": 1704067200000,
            "startTimeOfDay": null,
            "endTimeOfDay": null,
            "includeRegistrationIDs": [],
            "excludeRegistrationIDs": [],
            "includeFlags": 32768,
            "includeExactFlags": null,
            "excludeFlags": null,
            "excludeExactFlags": null,
            "deviceCountOnly": true,
            "returnDeviceCountByGeoHash": true,
            "id": "place-of-interest-1",
            "startDateTime": "2024-01-01T00:00:00Z",
            "endDateTime": "2024-02-01T00:00:00Z",
            "deviceCount": 23433,
            "deviceLimitHit": false,
            "devices": [],
            "counties": [
               {
                  "county": "Fairfax County",
                  "countyFIPS": "059",
                  "state": "Virginia",
                  "stateFIPS": "51",
                  "uniqueDeviceCount": 23433,
                  "uniqueDeviceCountInMin": 23433
               }
            ],
            "zipCodes": {
               "20190": {
                  "uniqueDeviceCount": 23433,
                  "uniqueDeviceCountInMin": 23433
               }
            },
            "censusBlockGroups": {
               "510594822034": {
                  "uniqueDeviceCount": 19564,
                  "uniqueDeviceCountInMin": 19564
               },
               "510594822031": {
                  "uniqueDeviceCount": 3243,
                  "uniqueDeviceCountInMin": 3243
               },
               "510594822042": {
                  "uniqueDeviceCount": 540,
                  "uniqueDeviceCountInMin": 540
               },
               "510594822032": {
                  "uniqueDeviceCount": 3626,
                  "uniqueDeviceCountInMin": 3626
               }
            }
         },
         "geometry": {
            "type": "Polygon",
            "coordinates": [
               [
                  [
                     -77.36331939697266,
                     38.957089496994705
                  ],
                  [
                     -77.35445737838745,
                     38.957089496994705
                  ],
                  [
                     -77.35445737838745,
                     38.96044327386091
                  ],
                  [
                     -77.36331939697266,
                     38.96044327386091
                  ],
                  [
                     -77.36331939697266,
                     38.957089496994705
                  ]
               ]
            ]
         },
         "id": "place-of-interest-1"
      },
      {
         "type": "Feature",
         "properties": {
            "endDateTimeEpochMS": 1729271658247,
            "startDateTimeEpochMS": 1728604800000,
            "startTimeOfDay": null,
            "endTimeOfDay": null,
            "includeRegistrationIDs": [],
            "excludeRegistrationIDs": [],
            "includeFlags": null,
            "includeExactFlags": null,
            "excludeFlags": null,
            "excludeExactFlags": null,
            "deviceCountOnly": false,
            "returnDeviceCountByGeoHash": false,
            "id": "place-of-interest-2",
            "startDateTime": "2024-10-11T00:00:00Z",
            "endDateTime": "2024-10-18T17:14:18.247726934Z",
            "deviceCount": 335,
            "deviceLimitHit": false,
            "devices": [
               {
                  "registrationID": "5cc0cf9d-502d-3577-a2ca-f5e8ff78fb80",
                  "advertiserID": null
               },
               {
                  "registrationID": "cdbe951e-57f2-38f3-9c19-7050504ad7a7",
                  "advertiserID": null
               }
            ],
            "counties": [
               {
                  "county": "Fairfax County",
                  "countyFIPS": "059",
                  "state": "Virginia",
                  "stateFIPS": "51",
                  "uniqueDeviceCount": 335,
                  "uniqueDeviceCountInMin": 335
               }
            ],
            "zipCodes": {
               "20190": {
                  "uniqueDeviceCount": 335,
                  "uniqueDeviceCountInMin": 335
               }
            },
            "censusBlockGroups": {
               "510594805053": {
                  "uniqueDeviceCount": 335,
                  "uniqueDeviceCountInMin": 335
               }
            }
         },
         "geometry": {
            "type": "Polygon",
            "coordinates": [
               [
                  [
                     -77.35402822494507,
                     38.969431780888165
                  ],
                  [
                     -77.35480606555939,
                     38.96860179949563
                  ],
                  [
                     -77.35380828380585,
                     38.968084620242585
                  ],
                  [
                     -77.35303312540054,
                     38.968931290760246
                  ],
                  [
                     -77.35402822494507,
                     38.969431780888165
                  ]
               ]
            ]
         },
         "id": "place-of-interest-2"
      }
   ]
}
```

Devices File Export PSV Headers

Name	Type	Description
FEATUREID	string	The unique identifier of the search feature.
DEVICE_ID	string	Either the registration ID or advertiser ID depending on the API key's settings.

Devices File Example Export PSV Contents

```
FEATURE_ID|DEVICE_ID
place-of-interest-1|d82fce23-a9b5-3c07-9466-d108f8f5efa0
place-of-interest-2|53a69d78-74c6-3de6-bb87-168f445e6fe9
```

Device Count Export PSV Headers

Name	Type	Description
FEATUREID	string	The unique identifier of the search feature.
DEVICE_COUNT	number (int32)	The number of Pseudonymized Registration IDs found for the search feature and other search criteria including the `inMin` parameter.
DEVICE_LIMIT_HIT	boolean	True if the number of Pseudonymized Registration IDs exported for the search features equals the limit of 10k per feature. False otherwise.

Device Count Example Export PSV Contents

```
FEATURE_ID|DEVICE_COUNT|DEVICE_LIMIT_HIT
place-of-interest-1|763|false
place-of-interest-2|44|false
```

Zip Code Device Count Export PSV Headers

Name	Type	Description
SEARCH_OBJECT_ID	string	The unique identifier of the search feature/object.
ZIPCODE	string	The zip code that the search feature overlaps and observations and/or registration IDs were found in. Only available for known zip codes, generally those inside the USA.
UNIQUE_DEVICE_COUNT	string	The number of unique Pseudonymized Registration IDs found in the search feature, other search criteria, and zip code, regardless of the `inMin` parameter.
UNIQUE_DEVICE_COUNT_IN_MIN	string	The number of unique Pseudonymized Registration IDs found in the search feature, other search criteria, and zip code, including the `inMin` parameter.

Zipcode Device Count Example Export PSV Contents

```
SEARCH_OBJECT_ID|ZIPCODE|UNIQUE_DEVICE_COUNT|UNIQUE_DEVICE_COUNT_IN_MIN
place-of-interest-2|20190|127|20
```

```
place-of-interest-1|20190|4896|20
```

County Device Count Export PSV Headers

Name	Type	Description
SEARCH_OBJECT_ID	string	The unique identifier of the search feature/object.
COUNTY	string	The name of the county.
COUNTYFIPS	number (int32)	The Federal Information Processing Standards (FIPS) code of the county.
STATE	string	The name of the state.
STATEFIPS	number (int32)	The Federal Information Processing Standards (FIPS) code of the state.
UNIQUE_DEVICE_COUNT	string	The number of unique Pseudonymized Registration IDs found in the search feature, other search criteria, and county, regardless of the `inMin` parameter.
UNIQUE_DEVICE_COUNT_IN_MIN	string	The number of unique Pseudonymized Registration IDs found in the search feature, other search criteria, and county, including the `inMin` parameter.

County Device Count Example Export PSV Contents

```
SEARCH_OBJECT_ID|COUNTY|COUNTYFIPS|STATE|STATEFIPS|UNIQUE_DEVICE_COUNT|UNIQUE_DEVICE_COUNT_IN_MIN
place-of-interest-2|Fairfax|059|Virginia|51|127|20
place-of-interest-1|Fairfax|059|Virginia|51|4896|20
```

Census Block Group Device Count Export PSV Headers

Name	Type	Description
SEARCH_OBJECT_ID	string	The unique identifier of the search feature/object.
CENSUS_BLOCK_GROUP_ID	number (int64)	The census block group ID for the census block group that the observations and/or registration IDs were found in. Only available for known census block groups, generally those inside the USA.
UNIQUE_DEVICE_COUNT_IN_MIN	string	The number of unique Pseudonymized Registration IDs found in the search feature, other search criteria, and county, including the `inMin` parameter.

Census Block Group Device Count Example Export PSV Contents

```
SEARCH_OBJECT_ID|CENSUS_BLOCK_GROUP_ID|UNIQUE_DEVICE_COUNT|UNIQUE_DEVICE_COUNT_IN_MIN
place-of-interest-1|510594822031|394|8
place-of-interest-1|510594822034|4239|17
place-of-interest-2|510594805053|127|20
```

/areas/devices/trends

POST

Summary: Retrieve Unique Pseudonymized Registration ID Trends For Search Areas

Description: This endpoint exists in order to search one or more geographic areas for the counts of unique Pseudonymized Registration IDs seen broken out by time. The Pseudonymized Registration ID counts returned are grouped by UTC day, week, month, and overall search time range.

Search requirements and limitations are described below:

Some search parameters can be specified only as query parameters, only in the request body, or in either location. If a search parameter is specified as a query parameter, it overrides what is specified in the request body.

The request body for this endpoint is required to be a GeoJSON FeatureCollection. Each feature in the FeatureCollection must have a geometry of type polygon, multipolygon, point, or linestring. If a point or linestring is specified, the feature is also required to have the property `radiusInMeters`. This property is used for converting points and linestrings to polygons for processing. At most 20 features can be specified in a single request. The size of a feature's geometry must be less than or equal to 400 square kilometers (sq km) and the total area of all search geometries must be less than or equal to 1k sq km. A search feature's geometry must be valid and not fold across itself. It is okay if two or more search features overlap.

By default, an API key is limited to making requests in these countries: USA, CAN, MEX. If you would like to search outside these countries, please reach out to your API sales representative.

Feature-Specific Search Parameters

Each feature can have its own search parameters as specified in the feature's `properties`. See model information below for a list of search parameters that can be specified per feature. As a reminder, any search parameters specified as query parameters will override those specified for a feature. Additionally, there are some search parameters that are not feature specific, and these can be specified either as query parameters or in the `properties` of the FeatureCollection's `crs` object. When metrics and results are returned for a `resultType` = `DIRECT` request, each metric or result will be returned either in the `properties` of a feature if the metric/result is specific to a feature, or in the `properties` of the `crs` object if not specific to a feature.

Along with the search features, the user can also specify a `startDateTimeEpochMS` and `endDateTimeEpochMS` to limit the time range of the search. Both parameters are optional. The start date-time must be before the end date-time, `startDateTimeEpochMS` must be at or after the start of the day 3 years ago, and `endDateTimeEpochMS` must be equal to or before the time of the request. If you want to use the current time as the end date, it is best to not supply an end date as the default end date value is the current time. If no start date-time is supplied, the default behavior is for start date to be the beginning of the day seven days before `endDateTimeEpochMS`. The default behavior for endDate is to be the current time. For example if the time of the request is `2022-03-10T12:34:56Z`, then startDate defaults to the equivalent of `2022-03-03T00:00:00Z` and endDate defaults to the equivalent of `2022-03-10T12:34:56Z`. `endDateTimeEpochMS` must be within 90 days from `startDateTimeEpochMS` disregarding time of day. For example, a `startDateTimeEpochMS` of 1597622400000, equal to `2020-08-17T00:00:00Z`, and `endDateTimeEpochMS` of 1605443696000, equal to `2020-11-15T12:34:56Z`, is allowed since the day components are 90 days apart. The `startDateTime` and `endDateTime` parameters allow users to specify time parameters in a human-readable format instead of epoch milliseconds. The time format is ISO 8601, which includes a date, time, and timezone. An example of this format is `2020-05-12T13:45:22.893+04:00`, this timestamp's equivalent time in epoch milliseconds is `1589276722893`. `startDateTime` is mutually exclusive with `startDateTimeEpochMS` and `endDateTime` is mutually exclusive with `endDateTimeEpochMS`.

While `startDateTimeEpochMS` and `endDateTimeEpochMS` enable a user to specify the overall time range to search, `startTimeOfDay` and `endTimeOfDay` enable a user to specify a period within each day to search. For example a user may want to search for all observations that only occurred during business hours and therefore would set `startTimeOfDay` to "08:00:00Z" and `endTimeOfDay` to "18:00:00Z". Overnight time periods are supported as well. For example a user could search from "22:00:00Z" to "02:00:00Z" if they were looking for observations that occurred in the middle of the night. Both start time of day and end time of day are supplied as strings and must be in format `HH:mm:ss'Z'`. Again, all time parameters are interpreted as being in UTC.

The following search parameters allow the user to filter the observations returned based on their flags values. These search parameters are `includeFlags`, `includeExactFlags`, `excludeFlags`, and `excludeExactFlags`. All are 64-bit integers (longs) and control whether an observation must have, or not have, specific flags in order to be counted. These four default to null with no effect applied.

For `DIRECT` requests, `returnDeviceCountByGeoHash` controls whether metrics are returned per geo hash in the `geoHashDeviceCount` response field. When a search feature is submitted to the API, the API calculates the geo-hashes needed to cover the geometry and uses those to query the database. The API uses the most precise geo hashes it can while keeping the number of geo hashes to a manageable amount. As a result, the precision and size of the geo hashes used scales with the size of the search feature. Alongside the overall Pseudonymized Registration ID count returned, the `geoHashDeviceCount` metrics, when combined with `geoHashWidthHeights` entries, provide a quick way to create heatmap visualizations.

Cross-Feature Search Parameters

The search parameters that are not feature specific are: `inMin` and `placeIdentifier`. `inMin` controls how many search features a Pseudonymized Registration ID must be seen in for the Pseudonymized Registration ID to be counted/returned in the results. In order to keep the result format consistent across features, the `placeIdentifier` search parameter must be specified as a query parameter or in the `properties` of the `crs`.

The query parameter `placeIdentifier` controls whether additional geospatial data is returned for feature and is a comma-separated list of strings with acceptable values being `NONE`, `ZIPCODE`, `COUNTY`, and `CBG` (census block group). The additional data is available only for the USA. If the list contains `NONE`, no place identifier information will be returned. `NONE` is mutually exclusive with all other options. If the list contains `ZIPCODE`, each feature will have a list of overlapping zip codes and the Pseudonymized Registration ID count trends for each. Similarly, if the list contains `COUNTY`, each feature will have a list of overlapping county FIPS (Federal Information Processing Standards) code, county name, state FIPS (Federal Information Processing Standards) code, and state name along with the Pseudonymized Registration ID count trends. If the list contains `CBG`, each feature will have a map of overlapping census block groups and the associated Pseudonymized Registration ID count trends.

Direct Response Fields For `DIRECT` responses, each geoJSON feature will have fields added to its `properties`. If `placeIdentifier` is not specified or `NONE`, then just `overallDeviceCounts` will be added to the properties of each search feature. This will contain the daily, weekly, monthly, and overall unique Pseudonymized Registration ID counts for the search feature (and other search criteria including the `inMin` parameter). If `placeIdentifier` includes `ZIPCODE`, then `zipCodeDeviceCounts` will be returned as well. `zipCodeDeviceCounts` is similar to `overallDeviceCounts` but is first broken out by zip code. Similarly, if `placeIdentifier` includes `COUNTY`, then `countyDeviceCounts` will be returned and if `placeIdentifier` includes `CBG`, then `cbgDeviceCounts` will be returned. As a reminder, zip code, county, and CBG data is only generally available in the USA.

Alongside the response fields described above, the API can also return per-geo-hash information when `returnDeviceCountByGeoHash` is true. In that case, `geoHashDeviceCount` will be populated. This can be used to create a heatmap of Pseudonymized Registration ID counts across the search area. The Pseudonymized Registration ID counts returned are impacted by the `inMIn` and other search parameters.

Parameters

Name	Located in	Description	Required	Schema
responseType	Query or CRS Properties	Optional. Default `DIRECT`. Possible values: `DIRECT` and `EXPORT`. Case insensitive. `DIRECT` indicates to return the results of the search as part of the JSON response body. `EXPORT` indicates that the results should be exported as files made available via download URLs from the `requestStatus/{requestID}` endpoint. `DIRECT` requests wait for the results to be available before returning while `EXPORT` requests run as a background task after being validated. As a result, a `DIRECT` request could take a minute for the results to come back while the same request as an `EXPORT` would return in seconds but the user would have to wait for the results to become available via the `requestStatus/{requestID}` endpoint.	No	string
includeHeaders	Query or CRS Properties	Optional boolean, default true. Specifies whether output psv files include a header row. If true, psv output files will include a header row. If false, psv output files will not include a header row.	No	Boolean

Name	Located in	Description	Required	Schema
compressOutputFiles	Query or CRS Properties	Optional boolean, default true. Specifies whether output files are gzip compressed. If true, output files are gzip compressed. If false, output files are not compressed.	No	Boolean
exportLimit	Query or CRS Properties	Optional integer, default null. Specifies max number of rows output files will have (not including optional header row). If defined, psv output files will have at most exportLimit rows. If not defined, psv output files have no row limit.	No	number (int64)
startDateTimeEpochMS	Query or Feature Properties	Optional. Default is the beginning of the day 7 days prior to `endDateTimeEpochMS`. The start of the time range to search for, expressed as epoch milliseconds.	No	number (int64)
endDateTimeEpochMS	Query or Feature Properties	Optional. Default is time of request. The end of the time range to search for, expressed as epoch milliseconds.	No	number (int64)
startDateTime	Query or Feature Properties	Optional. Default is the beginning of the day 7 days prior to `endDateTime`. The start of the time range to search for, expressed as an ISO-8601 offset timestamp.	No	string (ISO-8601 offset timestamp)
endDateTime	Query or Feature Properties	Optional. Default is time of request. The end of the time range to search for, expressed as ISO-8601 offset timestamp.	No	string (ISO-8601 offset timestamp)
startTimeOfDay	Query or Feature Properties	Optional. Default null. The string representation of the start time of day that observations must be after in order to be counted. An overnight filter is supported where `startTimeOfDay` is after `endTimeOfDay`. A user must supply either both `startTimeOfDay` and `endTimeOfDay` or neither since both are necessary to implement the filter. Format is `HH:mm:ss'Z'`.	No	string

Name	Located in	Description	Required	Schema
endTimeOfDay	Query or Feature Properties	Optional. Default null. The string representation of the end time of day that observations must be before in order to be counted. An overnight filter is supported where `startTimeOfDay` is after `endTimeOfDay`. A user must supply either both `startTimeOfDay` and `endTimeOfDay` or neither since both are necessary to implement the filter. Format is `HH:mm:ss'Z'`.	No	string
includeFlags	Query or Feature Properties	Optional. Default null. A bit array in the form of an int64. This search parameter is used to include observations in the results based on the observation's flags value. The bit indices of `includeFlags` correspond to the bit indices of the observations flags. The bitwise AND of `includeFlags` and a observation's flags value must be non-zero for the observation to be counted in the search results.	No	number (int64)
includeExactFlags	Query or Feature Properties	Optional. Default null. A bit array in the form of an int64. This search parameter is used to include observations in the results based on the observation's flags value. The bit indices of `includeExactFlags` correspond to the bit indices of the observations flags. The `includeExactFlags` value must exactly equal a observation's flags value for the observation to be counted in the search results.	No	number (int64)

Name	Located in	Description	Required	Schema
excludeFlags	Query or Feature Properties	Optional. Default null. A bit array in the form of an int64. This search parameter is used to filter out observations from the results based on the observation's flags value. The bit indices of `excludeFlags` correspond to the bit indices of the observations flags. When the result of a bitwise AND of `excludeFlags` and a observation's flags value is non-zero, then the observation will not be counted in the search results.	No	number (int64)
excludeExactFlags	Query or Feature Properties	Optional. Default null. A bit array in the form of an int64. This search parameter is used to filter out observations from the results based on the observation's flags value. The bit indices of `excludeExactFlags` correspond to the bit indices of the observations flags. When `excludeExactFlags` exactly equals a observation's flags value, then the observation will not be counted in the search results.	No	number (int64)
deviceCountOnly	Query or Feature Properties	Default false. If false just the Pseudonymized Registration ID counts are returned for the feature. If true, the Pseudonymized Registration ID identifiers are returned alongside the Pseudonymized Registration ID counts.	No	boolean
placeIdentifier	Query or CRS Properties	Default `NONE`. A comma-separated list of geospatial regions to return aggregate data for. Acceptable string values are `NONE`, `ZIPCODE`, `COUNTY`, `CBG`. If the list contains `NONE`, no place identifier information will be counted. `NONE` is mutually exclusive with all other options.	No	[string]
returnDeviceCountByGeoHash	Query or Feature Properties	Default false. If true, the `geoHashDeviceCount` field is populated per feature.	No	boolean

Name	Located in	Description	Required	Schema
inMin	Query or CRS Properties	Default 0. Specifies the number of search features a Pseudonymized Registration ID must be seen in for it to be counted/returned in the result set. Must be less than or equal to the number of search features in the request. For example, if three search features are in the request, `inMin` must be less than or equal to three. For the previous example, if `inMin` is two, then a Pseudonymized Registration ID must have been seen in at least two of the three search features in order to be returned. If the Pseudonymized Registration ID was seen in all three, it would be returned as well. However, if the Pseudonymized Registration ID was seen in only one of the search features, it would not be returned. A negative or zero `inMin` value behaves the same as an `inMin` value of 1. An `inMin` value of 0 is used by default so empty requests with no specified search features are acceptable.	No	number (int32)
body	Body		Yes	DevicesTrendsRequestFeatureCollection

application/json

```
{
    "type": "FeatureCollection",
    "crs": {
        "type": "name",
        "properties": {
            "inMin": 1,
            "placeIdentifier": ["ZIPCODE","COUNTY","CBG"]
        }
    },
    "features": [
        {
            "type": "Feature",
            "properties": {
                "endDateTimeEpochMS": 1706745600000,
                "startDateTimeEpochMS": 1704067200000,
                "startTimeOfDay": null,
                "endTimeOfDay": null,
                "includeRegistrationIDs": [],
                "excludeRegistrationIDs": [],
                "includeFlags": 32768,
                "includeExactFlags": null,
                "excludeFlags": null,
                "excludeExactFlags": null,
                "deviceCountOnly": true,
                "returnDeviceCountByGeoHash": true
            },
            "geometry": {
                "type": "Polygon",
                "coordinates": [
                    [
                        [
                            -77.36331939697266,
                            38.957089496994705
                        ],
                        [
                            -77.35445737838745,
                            38.957089496994705
                        ],
                        [
                            -77.35445737838745,
                            38.96044327386091
                        ],
                        [
                            -77.36331939697266,
                            38.96044327386091
                        ],
                        [
                            -77.36331939697266,
                            38.957089496994705
                        ]
                    ]
                ]
            },
            "id": "place-of-interest-1"
        },
        {
            "type": "Feature",
            "properties": {
                "endDateTimeEpochMS": null,
                "startDateTimeEpochMS": null,
                "startTimeOfDay": null,
                "endTimeOfDay": null,
                "includeRegistrationIDs": [],
                "excludeRegistrationIDs": [],
                "includeFlags": null,
                "includeExactFlags": null,
                "excludeFlags": null,
                "excludeExactFlags": null,
                "deviceCountOnly": false,
                "returnDeviceCountByGeoHash": false
            },
            "geometry": {
                "type": "Polygon",
                "coordinates": [
                    [
                        [
                            -77.35402822494507,
                            38.969431780888165
                        ],
                        [
                            -77.35480606555939,
                            38.96860179949563
                        ],
                        [
                            -77.35380828380585,
                            38.968084620242585
                        ],
                        [
                            -77.35303312540054,
                            38.968931290760246
                        ],
                        [
                            -77.35402822494507,
                            38.969431780888165
                        ]
                    ]
                ]
            },
            "id": "place-of-interest-2"
        }
    ]
}
```

Responses

Note: For the direct and/or export examples below, the results were truncated for brevity.

Code	Description	Schema
200	OK	DevicesTrendsResponseFeatureCollection

application/json

```
{
   "type": "FeatureCollection",
   "crs": {
      "type": "name",
      "properties": {
         "inMin": 1,
         "placeIdentifier": [
            "ZIPCODE",
            "COUNTY",
            "CBG"
         ],
         "responseType": "DIRECT",
         "includeHeaders": true,
         "compressOutputFiles": true,
         "exportLimit": null,
         "requestID": "05f99cda-bfec-4cc2-abd9-3ae79b0b7bd9",
         "name": "urn:ogc:def:crs:OGC::CRS84",
         "uniqueDeviceCount": 23742,
         "uniqueDeviceInMinCount": 23742
      }
   },
   "features": [
      {
         "type": "Feature",
         "properties": {
            "endDateTimeEpochMS": 1706745600000,
            "startDateTimeEpochMS": 1704067200000,
            "startTimeOfDay": null,
            "endTimeOfDay": null,
            "includeRegistrationIDs": [],
            "excludeRegistrationIDs": [],
            "includeFlags": 32768,
            "includeExactFlags": null,
            "excludeFlags": null,
            "excludeExactFlags": null,
            "returnDeviceCountByGeoHash": true,
            "id": "place-of-interest-1",
            "startDateTime": "2024-01-01T00:00:00Z",
            "endDateTime": "2024-02-01T00:00:00Z",
            "overallDeviceCounts": {
               "countsPerDay": {
                  "2024-01-01": 1259,
                  "2024-01-02": 1295,
                  "2024-01-03": 1104,
                  "2024-01-04": 1311,
                  "2024-01-05": 1212,
                  "2024-01-06": 1215,
                  "2024-01-07": 1038,
                  "2024-01-08": 1057,
                  "2024-01-09": 1066,
                  "2024-01-10": 1172,
                  "2024-01-11": 1038,
                  "2024-01-12": 1158,
                  "2024-01-13": 1384,
                  "2024-01-14": 1017,
                  "2024-01-15": 953,
                  "2024-01-16": 1159,
                  "2024-01-17": 1214,
                  "2024-01-18": 1459,
                  "2024-01-19": 1502,
                  "2024-01-20": 1560,
                  "2024-01-21": 1097,
                  "2024-01-22": 1093,
                  "2024-01-23": 1107,
                  "2024-01-24": 971,
                  "2024-01-25": 1044,
                  "2024-01-26": 1144,
                  "2024-01-27": 1188,
                  "2024-01-28": 1052,
                  "2024-01-29": 942,
                  "2024-01-30": 783,
                  "2024-01-31": 1150
               },
               "countsPerWeek": {
                  "2024-01-01": 6427,
                  "2024-01-08": 6186,
                  "2024-01-15": 6680,
                  "2024-01-22": 5823,
                  "2024-01-29": 2535
               },
               "countsPerMonth": {
                  "2024-01-01": 23433
               },
               "totalCount": 23433
            },
            "zipCodeDeviceCounts": {
               "20190": {
                  "countsPerDay": {
                     "2024-01-01": 1259,
                     "2024-01-02": 1295,
                     "2024-01-03": 1104,
                     "2024-01-04": 1311,
                     "2024-01-05": 1212,
                     "2024-01-06": 1215,
                     "2024-01-07": 1038,
                     "2024-01-08": 1057,
                     "2024-01-09": 1066,
                     "2024-01-10": 1172,
                     "2024-01-11": 1038,
                     "2024-01-12": 1158,
                     "2024-01-13": 1384,
                     "2024-01-14": 1017,
                     "2024-01-15": 953,
                     "2024-01-16": 1159,
                     "2024-01-17": 1214,
                     "2024-01-18": 1459,
                     "2024-01-19": 1502,
                     "2024-01-20": 1560,
                     "2024-01-21": 1097,
                     "2024-01-22": 1093,
                     "2024-01-23": 1107,
                     "2024-01-24": 971,
                     "2024-01-25": 1044,
                     "2024-01-26": 1144,
                     "2024-01-27": 1188,
                     "2024-01-28": 1052,
                     "2024-01-29": 942,
                     "2024-01-30": 783,
                     "2024-01-31": 1150
                  },
                  "countsPerWeek": {
                     "2024-01-01": 6427,
                     "2024-01-08": 6186,
                     "2024-01-15": 6680,
                     "2024-01-22": 5823,
                     "2024-01-29": 2535
                  },
                  "countsPerMonth": {
                     "2024-01-01": 23433
                  },
                  "totalCount": 23433
               }
            },
            "countyDeviceCounts": {
               "Fairfax County": {
                  "countsPerDay": {
                     "2024-01-01": 1259,
                     "2024-01-02": 1295,
                     "2024-01-03": 1104,
                     "2024-01-04": 1311,
                     "2024-01-05": 1212,
                     "2024-01-06": 1215,
                     "2024-01-07": 1038,
                     "2024-01-08": 1057,
                     "2024-01-09": 1066,
                     "2024-01-10": 1172,
                     "2024-01-11": 1038,
                     "2024-01-12": 1158,
                     "2024-01-13": 1384,
                     "2024-01-14": 1017,
                     "2024-01-15": 953,
                     "2024-01-16": 1159,
                     "2024-01-17": 1214,
                     "2024-01-18": 1459,
                     "2024-01-19": 1502,
                     "2024-01-20": 1560,
                     "2024-01-21": 1097,
                     "2024-01-22": 1093,
                     "2024-01-23": 1107,
                     "2024-01-24": 971,
                     "2024-01-25": 1044,
                     "2024-01-26": 1144,
                     "2024-01-27": 1188,
                     "2024-01-28": 1052,
                     "2024-01-29": 942,
                     "2024-01-30": 783,
                     "2024-01-31": 1150
                  },
                  "countsPerWeek": {
                     "2024-01-01": 6427,
                     "2024-01-08": 6186,
                     "2024-01-15": 6680,
                     "2024-01-22": 5823,
                     "2024-01-29": 2535
                  },
                  "countsPerMonth": {
                     "2024-01-01": 23433
                  },
                  "totalCount": 23433
               }
            },
               "cbgDeviceCounts": {
                  "510594822034": {
                     "countsPerDay": {
                        "2024-01-01": 1055,
                        "2024-01-02": 1073,
                        "2024-01-03": 914,
                        "2024-01-04": 1073,
                        "2024-01-05": 979,
                        "2024-01-06": 974,
                        "2024-01-07": 853,
                        "2024-01-08": 823,
                        "2024-01-09": 850,
                        "2024-01-10": 996,
                        "2024-01-11": 883,
                        "2024-01-12": 961,
                        "2024-01-13": 1086,
                        "2024-01-14": 844,
                        "2024-01-15": 778,
                        "2024-01-16": 949,
                        "2024-01-17": 959,
                        "2024-01-18": 1230,
                        "2024-01-19": 1287,
                        "2024-01-20": 1267,
                        "2024-01-21": 927,
                        "2024-01-22": 904,
                        "2024-01-23": 899,
                        "2024-01-24": 784,
                        "2024-01-25": 824,
                        "2024-01-26": 932,
                        "2024-01-27": 965,
                        "2024-01-28": 872,
                        "2024-01-29": 791,
                        "2024-01-30": 662,
                        "2024-01-31": 961
                     },
                     "countsPerWeek": {
                        "2024-01-01": 5325,
                        "2024-01-08": 5063,
                        "2024-01-15": 5565,
                        "2024-01-22": 4784,
                        "2024-01-29": 2136
                     },
                     "countsPerMonth": {
                        "2024-01-01": 19564
                     },
                     "totalCount": 19564
                  },
                  "510594822032": {
                     "countsPerDay": {
                        "2024-01-01": 171,
                        "2024-01-02": 144,
                        "2024-01-03": 130,
                        "2024-01-04": 129,
                        "2024-01-05": 153,
                        "2024-01-06": 141,
                        "2024-01-07": 131,
                        "2024-01-08": 169,
                        "2024-01-09": 132,
                        "2024-01-10": 123,
                        "2024-01-11": 121,
                        "2024-01-12": 136,
                        "2024-01-13": 121,
                        "2024-01-14": 139,
                        "2024-01-15": 115,
                        "2024-01-16": 129,
                        "2024-01-17": 134,
                        "2024-01-18": 164,
                        "2024-01-19": 170,
                        "2024-01-20": 174,
                        "2024-01-21": 125,
                        "2024-01-22": 104,
                        "2024-01-23": 116,
                        "2024-01-24": 131,
                        "2024-01-25": 155,
                        "2024-01-26": 165,
                        "2024-01-27": 188,
                        "2024-01-28": 144,
                        "2024-01-29": 118,
                        "2024-01-30": 98,
                        "2024-01-31": 172
                     },
                     "countsPerWeek": {
                        "2024-01-01": 876,
                        "2024-01-08": 846,
                        "2024-01-15": 892,
                        "2024-01-22": 900,
                        "2024-01-29": 372
                     },
                     "countsPerMonth": {
                        "2024-01-01": 3626
                     },
                     "totalCount": 3626
                  },
                  "510594822042": {
                     "countsPerDay": {
                        "2024-01-01": 26,
                        "2024-01-02": 25,
                        "2024-01-03": 16,
                        "2024-01-04": 23,
                        "2024-01-05": 21,
                        "2024-01-06": 20,
                        "2024-01-07": 19,
                        "2024-01-08": 12,
                        "2024-01-09": 15,
                        "2024-01-10": 17,
                        "2024-01-11": 8,
                        "2024-01-12": 15,
                        "2024-01-13": 20,
                        "2024-01-14": 40,
                        "2024-01-15": 22,
                        "2024-01-16": 11,
                        "2024-01-17": 20,
                        "2024-01-18": 21,
                        "2024-01-19": 24,
                        "2024-01-20": 41,
                        "2024-01-21": 14,
                        "2024-01-22": 13,
                        "2024-01-23": 11,
                        "2024-01-24": 17,
                        "2024-01-25": 8,
                        "2024-01-26": 17,
                        "2024-01-27": 22,
                        "2024-01-28": 26,
                        "2024-01-29": 23,
                        "2024-01-30": 13,
                        "2024-01-31": 25
                     },
                     "countsPerWeek": {
                        "2024-01-01": 147,
                        "2024-01-08": 120,
                        "2024-01-15": 138,
                        "2024-01-22": 112,
                        "2024-01-29": 60
                     },
                     "countsPerMonth": {
                        "2024-01-01": 540
                  },
                  "totalCount": 540
               },
               "510594822031": {
                  "countsPerDay": {
                     "2024-01-01": 129,
                     "2024-01-02": 139,
                     "2024-01-03": 124,
                     "2024-01-04": 160,
                     "2024-01-05": 163,
                     "2024-01-06": 145,
                     "2024-01-07": 98,
                     "2024-01-08": 105,
                     "2024-01-09": 129,
                     "2024-01-10": 101,
                     "2024-01-11": 98,
                     "2024-01-12": 110,
                     "2024-01-13": 214,
                     "2024-01-14": 142,
                     "2024-01-15": 93,
                     "2024-01-16": 116,
                     "2024-01-17": 141,
                     "2024-01-18": 121,
                     "2024-01-19": 115,
                     "2024-01-20": 176,
                     "2024-01-21": 99,
                     "2024-01-22": 112,
                     "2024-01-23": 112,
                     "2024-01-24": 116,
                     "2024-01-25": 147,
                     "2024-01-26": 140,
                     "2024-01-27": 135,
                     "2024-01-28": 122,
                     "2024-01-29": 96,
                     "2024-01-30": 89,
                     "2024-01-31": 127
                  },
                  "countsPerWeek": {
                     "2024-01-01": 856,
                     "2024-01-08": 815,
                     "2024-01-15": 749,
                     "2024-01-22": 791,
                     "2024-01-29": 289
                  },
                  "countsPerMonth": {
                     "2024-01-01": 3243
                  },
                  "totalCount": 3243
               }
            },
            "deviceCount": 23433,
            "deviceLimitHit": false,
            "devices": []
         },
         "geometry": {
            "type": "Polygon",
            "coordinates": [
               [
                  [
                     -77.36331939697266,
                     38.957089496994705
                  ],
                  [
                     -77.35445737838745,
                     38.957089496994705
                  ],
                  [
                     -77.35445737838745,
                     38.96044327386091
                  ],
                  [
                     -77.36331939697266,
                     38.96044327386091
                  ],
                  [
                     -77.36331939697266,
                     38.957089496994705
                  ]
               ]
            ]
         },
         "id": "place-of-interest-1"
      },
      {
         "type": "Feature",
         "properties": {
            "endDateTimeEpochMS": 1729272170852,
            "startDateTimeEpochMS": 1728604800000,
            "startTimeOfDay": null,
            "endTimeOfDay": null,
            "includeRegistrationIDs": [],
            "excludeRegistrationIDs": [],
            "includeFlags": null,
            "includeExactFlags": null,
            "excludeFlags": null,
            "excludeExactFlags": null,
            "returnDeviceCountByGeoHash": false,
            "id": "place-of-interest-2",
            "startDateTime": "2024-10-11T00:00:00Z",
            "endDateTime": "2024-10-18T17:22:50.852334817Z",
            "overallDeviceCounts": {
               "countsPerDay": {
                  "2024-10-11": 121,
                  "2024-10-12": 134,
                  "2024-10-13": 89,
                  "2024-10-14": 13,
                  "2024-10-15": 1
               },
               "countsPerWeek": {
                  "2024-10-07": 325,
                  "2024-10-14": 13
               },
               "countsPerMonth": {
                  "2024-10-01": 335
               },
               "totalCount": 335
            },
            "zipCodeDeviceCounts": {
               "20190": {
                  "countsPerDay": {
                     "2024-10-11": 121,
                     "2024-10-12": 134,
                     "2024-10-13": 89,
                     "2024-10-14": 13,
                     "2024-10-15": 1
                  },
                  "countsPerWeek": {
                     "2024-10-07": 325,
                     "2024-10-14": 13
                  },
                  "countsPerMonth": {
                     "2024-10-01": 335
                  },
                  "totalCount": 335
               }
            },
               "countyDeviceCounts": {
                  "Fairfax County": {
                     "countsPerDay": {
                        "2024-10-11": 121,
                        "2024-10-12": 134,
                        "2024-10-13": 89,
                        "2024-10-14": 13,
                        "2024-10-15": 1
                     },
                     "countsPerWeek": {
                        "2024-10-07": 325,
                        "2024-10-14": 13
                     },
                     "countsPerMonth": {
                        "2024-10-01": 335
                     },
                     "totalCount": 335
                  }
               },
               "cbgDeviceCounts": {
                  "510594805053": {
                     "countsPerDay": {
                        "2024-10-11": 121,
                        "2024-10-12": 134,
                        "2024-10-13": 89,
                        "2024-10-14": 13,
                        "2024-10-15": 1
                     },
                     "countsPerWeek": {
                        "2024-10-07": 325,
                        "2024-10-14": 13
                     },
                     "countsPerMonth": {
                        "2024-10-01": 335
                     },
                     "totalCount": 335
                  }
               },
               "deviceCount": 335,
               "deviceLimitHit": false,
               "devices": [
                  {
                     "registrationID": "5cc0cf9d-502d-3577-a2ca-f5e8ff78fb80",
                     "advertiserID": null
                  },
                  {
                     "registrationID": "cdbe951e-57f2-38f3-9c19-7050504ad7a7",
                     "advertiserID": null
                  }]
            },
            "geometry": {
               "type": "Polygon",
               "coordinates": [
                  [
                     [
                        -77.35402822494507,
                        38.969431780888165
                     ],
                     [
                        -77.35480606555939,
                        38.96860179949563
                     ],
                     [
                        -77.35380828380585,
                        38.968084620242585
                     ],
                     [
                        -77.35303312540054,
                     38.968931290760246
                  ],
                  [
                     -77.35402822494507,
                     38.969431780888165
                  ]
               ]
            ]
         },
         "id": "place-of-interest-2"
      }
   ]
}
```

Export PSV Headers

Name	Type	Description
SEARCHOBJECTID	string	The unique identifier of the search feature/object.
PLACE_IDENTIFIER_TYPE	string	Either `TOTAL`, `ZIPCODE`, `COUNTY`, or `CENSUS_BLOCK_GROUP_ID`. The spatial aggregate used for that row. Only included in the results if `placeIdentifier` is not `NONE`. When `placeIdentifier` is `NONE`, only the `TOTAL` Pseudonymized Registration ID counts are returned.
PLACE_IDENTIFIER_VALUE	string	The unique ID for the place identifier type. Null for `TOTAL` counts. Only included in the results if `placeIdentifier` is not `NONE`. When `placeIdentifier` is `NONE`, only the `TOTAL` Pseudonymized Registration ID counts are returned.
DATE_TYPE	string	Either `DAY`, `START_OF_WEEK`, or `START_OF_MONTH`. The temporal aggregate used for that row.
DATE_VALUE	string	The date string corresponding to the temporal range. For `START_OF_WEEK` and `START_OF_MONTH`, the date will be the first date of the week or month.
DEVICE_COUNT	number (int32)	The number of Pseudonymized Registration IDs found for matching the search area and criteria for the specified spatial and temporal aggregation.

Example Export PSV Contents

```
SEARCHOBJECTID|PLACE_IDENTIFIER_TYPE|PLACE_IDENTIFIER_VALUE|DATE_TYPE|DATE_VALUE|DEVICE_COUNT
place-of-interest-1|CENSUS_BLOCK_GROUP_ID|510594822031|DAY|2022-03-18|77
place-of-interest-1|CENSUS_BLOCK_GROUP_ID|510594822031|START_OF_MONTH|2022-03-01|394
place-of-interest-1|CENSUS_BLOCK_GROUP_ID|510594822031|START_OF_WEEK|2022-03-14|179
place-of-interest-1|COUNTY|Fairfax|DAY|2022-03-18|1764
place-of-interest-1|COUNTY|Fairfax|START_OF_MONTH|2022-03-01|4892
place-of-interest-1|COUNTY|Fairfax|START_OF_WEEK|2022-03-14|3254
place-of-interest-1|TOTAL|NULL|DAY|2022-03-18|1764
place-of-interest-1|TOTAL|NULL|START_OF_MONTH|2022-03-01|4892
place-of-interest-1|TOTAL|NULL|START_OF_WEEK|2022-03-14|3254
place-of-interest-1|ZIPCODE|20190|DAY|2022-03-18|1764
place-of-interest-1|ZIPCODE|20190|START_OF_MONTH|2022-03-01|4892
place-of-interest-1|ZIPCODE|20190|START_OF_WEEK|2022-03-14|3254
```

Models

AreasDevicesRequestFeatureCollection

Name	Type	Description	Required
type	string	The GeoJSON object type. Must be "FeatureCollection".	Yes

Name	Type	Description	Required
crs	AreasDevicesRequestCRS	The GeoJSON FeatureCollection CRS. Because the GeoJSON specification does not support properties in a FeatureCollection, in order to have non-feature-specific request or response properties, the `properties` of the CRS object are used instead. The rest of the CRS information is ignored by the API.	No
features	[AreasDevicesRequestFeature]	A list of GeoJSON Features. The `geometry` of the feature is used to identify the user's desired search area and the `properties` of the feature are used to send and receive feature-specific search parameters and results.	Yes

AreasDevicesRequestFeature

Name	Type	Description	Required
type	string	The GeoJSON object type. This field must be "Feature".	Yes
id	string	The GeoJSON specification allows for an `id` string field to provided per Feature to identify it. A user may provide an `id` for the feature here or as an `id` field in the properties of the feature. If a value is provided here, it will override the `id` field in the properties. If no identifier is provided for the feature in either location, the API will assign it a random UUIDv4 as its `id`. When the results of a request are exported as PSV, the feature ID will be included as a column to identify which search feature the results are associated with. If a user wants to export their results, it is recommended that the user specify the `id` per feature since otherwise a random identifier will be used and the user will be unaware of which identifier goes with which feature.	No
properties	AreasDevicesRequestFeatureProperties	The properties of the feature. Used to provide per-feature request parameters and to return per-feature results.	No
geometry	GeoJSON Polygon, MultiPolygon, Point, or LineString	The GeoJSON geometry of the feature. Required by the API for this endpoint to be a GeoJSON Polygon, MultiPolygon, Point, or LineString. See GeoJSON specification for details about GeoJSON objects.	Yes

AreasDevicesRequestFeatureProperties

Name	Type	Description	Required
id	string	Optional. Default random UUIDv4. The unique identifier of the feature. Overridden by feature's `id` field.	No
startDateTimeEpochMS	number (int64)	Optional. Default is the beginning of the day 7 days prior to `endDateTimeEpochMS`. The start of the time range to search for, expressed as epoch milliseconds.	No
endDateTimeEpochMS	number (int64)	Optional. Default is time of request. The end of the time range to search for, expressed as epoch milliseconds.	No
startDateTime	string	Optional. Default is the beginning of the day 7 days prior to `endDateTime`. The start of the time range to search for, expressed as an ISO-8601 offset timestamp.	No
endDateTime	string	Optional. Default is time of request. The end of the time range to search for, expressed as ISO-8601 offset timestamp.	No

Name	Type	Description	Required
startTimeOfDay	string	Optional. Default null. The string representation of the start time of day that observations must be after in order to be counted. An overnight filter is supported where `startTimeOfDay` is after `endTimeOfDay`. A user must supply either both `startTimeOfDay` and `endTimeOfDay` or neither since both are necessary to implement the filter. Format is `HH:mm:ss'Z'`.	No
endTimeOfDay	string	Optional. Default null. The string representation of the end time of day that observations must be before in order to be counted. An overnight filter is supported where `startTimeOfDay` is after `endTimeOfDay`. A user must supply either both `startTimeOfDay` and `endTimeOfDay` or neither since both are necessary to implement the filter. Format is `HH:mm:ss'Z'`.	No
includeRegistrationIDs	[string]	Default empty array. A set of registration IDs. A registration ID must be in this set to be counted and/or returned in results. A maximum of 10k entries can be provided.	No
excludeRegistrationIDs	[string]	Default empty array. A set of registration IDs. A registration ID must not be in this set to be counted and/or returned in the results. A maximum of 10k entries can be provided.	No
includeAdvertiserIDs	[string]	Default empty array. A set of advertiser IDs. An advertiser ID must be in this set to be counted and/or returned in results. This field replaces `includeRegistrationIDs` if the API key is configured to return `maid`. A maximum of 10k entries can be provided.	No
excludeAdvertiserIDs	[string]	Default empty array. A set of advertiser IDs. An advertiser ID must not be in this set to be counted and/or returned in the results. This field replaces `excludeRegistrationIDs` if the API key is configured to return `maid`. A maximum of 10k entries can be provided.	No
includeFlags	number (int64)	Optional. Default null. A bit array in the form of an int64. This search parameter is used to include observations in the results based on the observation's flags value. The bit indices of `includeFlags` correspond to the bit indices of the observations flags. The bitwise AND of `includeFlags` and a observation's flags value must be non-zero for the observation to be counted in the search results.	No
includeExactFlags	number (int64)	Optional. Default null. A bit array in the form of an int64. This search parameter is used to include observations in the results based on the observation's flags value. The bit indices of `includeExactFlags` correspond to the bit indices of the observations flags. The `includeExactFlags` value must exactly equal a observation's flags value for the observation to be counted in the search results.	No
excludeFlags	number (int64)	Optional. Default null. A bit array in the form of an int64. This search parameter is used to filter out observations from the results based on the observation's flags value. The bit indices of `excludeFlags` correspond to the bit indices of the observations flags. When the result of a bitwise AND of `excludeFlags` and a observation's flags value is non-zero, then the observation will not be counted in the search results.	No
excludeExactFlags	number (int64)	Optional. Default null. A bit array in the form of an int64. This search parameter is used to filter out observations from the results based on the observation's flags value. The bit indices of `excludeExactFlags` correspond to the bit indices of the observations flags. When `excludeExactFlags` exactly equals a observation's flags value, then the observation will not be counted in the search results.	No
deviceCountOnly	boolean	Default false. Indicates, when true, to return just the Pseudonymized Registration ID count and not the found Pseudonymized Registration IDs.	No
returnDeviceCountByGeoHash	boolean	Default false. If true, the `geoHashDeviceCount` field is populated.	No

AreasDevicesRequestCRS

Name	Type	Description	Required
type	string	Required by GeoJSON specification if `crs` is included in feature collection. Should be "name".	Yes
properties	AreasDevicesRequestCRSProperties	The properties of the GeoJSON CRS object. Used by the API for non-feature-specific search parameters and results since the GeoJSON specification does not support a `properties` field in a FeatureCollection.	No

AreasDevicesRequestCRSProperties

Name	Type	Description	Required
inMin	number (int32)	Default 0. Specifies the number of search features a Pseudonymized Registration ID must be seen in for it to be counted/returned in the result set. Must be less than or equal to the number of search features in the request. For example, if three search features are in the request, `inMin` must be less than or equal to three. For the previous example, if `inMin` is two, then a Pseudonymized Registration ID must have been seen in at least two of the three search features in order to be returned. If the Pseudonymized Registration ID was seen in all three, it would be returned as well. However, if the Pseudonymized Registration ID was seen in only one of the search features, it would not be returned. A negative or zero `inMin` value behaves the same as an `inMin` value of 1. An `inMin` value of 0 is used by default so empty requests with no specified search features are acceptable.	No
responseType	string	Optional. Default `DIRECT`. Possible values: `DIRECT` and `EXPORT`. Case insensitive. `DIRECT` indicates to return the results of the search as part of the JSON response body. `EXPORT` indicates that the results should be exported as files made available via download URLs from the `requestStatus/{requestID}` endpoint. `DIRECT` requests wait for the results to be available before returning while `EXPORT` requests run as a background task after being validated. As a result, a `DIRECT` request could take a minute for the results to come back while the same request as an `EXPORT` would return in seconds but the user would have to wait for the results to become available via the `requestStatus/{requestID}` endpoint.	No
includeHeaders	Boolean	Optional boolean, default true. Specifies whether output psv files include a header row. If true, psv output files will include a header row. If false, psv output files will not include a header row.	No
compressOutputFiles	Boolean	Optional boolean, default true. Specifies whether output files are gzip compressed. If true, output files are gzip compressed. If false, output files are not compressed.	No
exportLimit	number (int64)	Optional integer, default null. Specifies max number of rows output files will have (not including optional header row). If defined, psv output files will have at most exportLimit rows. If not defined, psv output files have no row limit.	No
placeIdentifier	[string]	Default `NONE`. A comma-separated list of geospatial regions to return aggregate data for. Acceptable string values are `NONE`, `ZIPCODE`, `COUNTY`, `CBG`. If the list contains `NONE`, no place identifier information will be counted. `NONE` is mutually exclusive with all other options.	No

AreasDevicesResponseFeatureCollection

Name	Type	Description	Required
type	string	The GeoJSON object type. The `/observations/geo/search` endpoint requires that the request body is a FeatureCollection. As a result this field must be "FeatureCollection".	Yes

Name	Type	Description	Required
crs	AreasDevicesResponseCRS	The GeoJSON FeatureCollection CRS. Because the GeoJSON specification does not support properties in a FeatureCollection, in order to have non-feature-specific request or response properties, the `properties` of the CRS object are used instead. The rest of the CRS information is ignored by the API.	No
features	[AreasDevicesResponseFeature]	A list of GeoJSON Features. The `geometry` of the feature is used to identify the user's desired search area and the `properties` of the feature are used to send and receive feature-specific search parameters and results.	Yes

AreasDevicesResponseFeature

Name	Type	Description	Required
type	string	The GeoJSON object type. This field must be "Feature".	Yes
id	string	The GeoJSON specification allows for an `id` string field to provided per Feature to identify it. A user may provide an `id` for the feature here or as an `id` field in the properties of the feature. If a value is provided here, it will override the `id` field in the properties. If no identifier is provided for the feature in either location, the API will assign it a random UUIDv4 as its `id`. When the results of a request are exported as PSV, the feature ID will be included as a column to identify which search feature the results are associated with. If a user wants to export their results, it is recommended that the user specify the `id` per feature since otherwise a random identifier will be used and the user will be unaware of which identifier goes with which feature.	No
properties	AreasDevicesResponseFeatureProperties	The properties of the feature. Used to provide per-feature request parameters and to return per-feature results.	No
geometry	GeoJSON Polygon, MultiPolygon, Point, or LineString	The GeoJSON geometry of the feature. Required by the API for this endpoint to be a GeoJSON Polygon, MultiPolygon, Point or LineString. See GeoJSON specification for details about GeoJSON objects.	Yes

AreasDevicesResponseFeatureProperties

Name	Type	Description	Required
id	string	Optional. Default random UUIDv4. The unique identifier of the feature. Overridden by feature's `id` field.	No
startDateTimeEpochMS	number (int64)	Optional. Default is the beginning of the day 7 days prior to `endDateTimeEpochMS`. The start of the time range to search for, expressed as epoch milliseconds.	No
endDateTimeEpochMS	number (int64)	Optional. Default is time of request. The end of the time range to search for, expressed as epoch milliseconds.	No
startDateTime	string	Optional. Default is the beginning of the day 7 days prior to `endDateTime`. The start of the time range to search for, expressed as an ISO-8601 offset timestamp.	No
endDateTime	string	Optional. Default is time of request. The end of the time range to search for, expressed as ISO-8601 offset timestamp.	No

Name	Type	Description	Required
startTimeOfDay	string	Optional. Default null. The string representation of the start time of day that observations must be after in order to be counted. An overnight filter is supported where `startTimeOfDay` is after `endTimeOfDay`. A user must supply either both `startTimeOfDay` and `endTimeOfDay` or neither since both are necessary to implement the filter. Format is `HH:mm:ss'Z'`.	No
endTimeOfDay	string	Optional. Default null. The string representation of the end time of day that observations must be before in order to be counted. An overnight filter is supported where `startTimeOfDay` is after `endTimeOfDay`. A user must supply either both `startTimeOfDay` and `endTimeOfDay` or neither since both are necessary to implement the filter. Format is `HH:mm:ss'Z'`.	No
includeRegistrationIDs	[string]	Default empty array. A set of registration IDs. A registration ID must be in this set to be counted and/or returned in results. A maximum of 10k entries can be provided.	No
excludeRegistrationIDs	[string]	Default empty array. A set of registration IDs. A registration ID must not be in this set to be counted and/or returned in the results. A maximum of 10k entries can be provided.	No
includeAdvertiserIDs	[string]	Default empty array. A set of advertiser IDs. An advertiser ID must be in this set to be counted and/or returned in results. This field replaces `includeRegistrationIDs` if the API key is configured to return `advertiserID`. A maximum of 10k entries can be provided.	No
excludeAdvertiserIDs	[string]	Default empty array. A set of advertiser IDs. An advertiser ID must not be in this set to be counted and/or returned in the results. This field replaces `excludeRegistrationIDs` if the API key is configured to return `advertiserID`. A maximum of 10k entries can be provided.	No
includeFlags	number (int64)	Optional. Default null. A bit array in the form of an int64. This search parameter is used to include observations in the results based on the observation's flags value. The bit indices of `includeFlags` correspond to the bit indices of the observations flags. The bitwise AND of `includeFlags` and a observation's flags value must be non-zero for the observation to be counted in the search results.	No
includeExactFlags	number (int64)	Optional. Default null. A bit array in the form of an int64. This search parameter is used to include observations in the results based on the observation's flags value. The bit indices of `includeExactFlags` correspond to the bit indices of the observations flags. The `includeExactFlags` value must exactly equal a observation's flags value for the observation to be counted in the search results.	No
excludeFlags	number (int64)	Optional. Default null. A bit array in the form of an int64. This search parameter is used to filter out observations from the results based on the observation's flags value. The bit indices of `excludeFlags` correspond to the bit indices of the observations flags. When the result of a bitwise AND of `excludeFlags` and a observation's flags value is non-zero, then the observation will not be counted in the search results.	No

Name	Type	Description	Required
excludeExactFlags	number (int64)	Optional. Default null. A bit array in the form of an int64. This search parameter is used to filter out observations from the results based on the observation's flags value. The bit indices of `excludeExactFlags` correspond to the bit indices of the observations flags. When `excludeExactFlags` exactly equals a observation's flags value, then the observation will not be counted in the search results.	No
deviceCountOnly	boolean	Default false. Indicates, when true, to return just the Pseudonymized Registration ID count and not the found Pseudonymized Registration IDs.	No
returnDeviceCountByGeoHash	boolean	Default false. If true, the `geoHashDeviceCount` field is populated.	No
geoHashWidthHeights	[GeoHashWidthHeight]	List of geo hash lengths along with their width and heights used to query the database for this search feature.	No
geoHashDeviceCount	[GeoHashDeviceCount]	Populated when `returnPerGeoHashInfo` is true, otherwise empty. Each entry contains the geo hash string, its center-point latitude and longitude, and result metrics.	No
deviceCount	number (int32)	The number of unique Pseudonymized Registration IDs found to have observation data inside the search area matching the search criteria, including the `inMin` parameter. The `deviceCount` returned is independent of the number of entries in `devices` list. If `deviceCountOnly` is true, then `deviceCount` will be populated but `devices` will be empty.	Yes
devices	DeviceIDRecord	The number of unique Pseudonymized Registration IDs found to have observation data inside the search area matching the search criteria, including the `inMin` parameter. The `deviceCount` returned is independent of the number of entries in `devices` list. If `deviceCountOnly` is true, then `deviceCount` will be populated but `devices` will be empty.	Yes
deviceLimitHit	boolean	True if the number of Pseudonymized Registration IDs returned in `devices` is equal to the limit of 10k.	Yes
zipCodes	Object: key -> ZipCodeInfo	An object where each key is a zip code that the search area overlaps. For each zip code key, the value is an object populated with the unique Pseudonymized Registration ID count (with and without `inMin` applied).	Yes
counties	[CountyInfo]	A list of objects where each is for a county overlapped by the search area. For each county, identifier info about the county is returned along with the unique Pseudonymized Registration ID count (with and without `inMin` applied).	Yes
censusBlockGroups	Object: key -> CBGInfo	An object where each key is a census block group identifier that the search area overlaps. For each CBG key, the value is an object populated with the unique Pseudonymized Registration ID count (with and without `inMin` applied).	Yes

GeoHashWidthHeight

Name	Type	Description	Required
length	number (int32)	The length of the geo hashes that the width and height correspond to.	No
width	double	The width in decimal degrees of the geo hashes used to query the database for this search feature. Useful to know when trying to plot on a map the geo hashes of `perGeoHashInfo`.	No

Name	Type	Description	Required
height	double	The height in decimal degrees of the geo hashes used to query the database for this search feature. Useful to know when trying to plot on a map the geo hashes of `perGeoHashInfo`.	No

GeoHashDeviceCount

Name	Type	Description	Required
geohash	string	The geo hash string.	Yes
latitude	double	The latitude of the center-point of the geo-hash in decimal degrees.	Yes
longitude	double	The longitude of the center-point of the geo-hash in decimal degrees.	Yes
deviceCount	number (int64)	The number of unique Pseudonymized Registration IDs matching the search criteria found in the geo hash including the `inMin` requirement.	Yes

DeviceIDRecord

Name	Type	Description	Required
registrationID	string	The UUIDv4 anonymous Unacast registration ID of the Pseudonymized Registration ID. Only one of `registrationID` or `advertiserID` will be populated based on API key setting. `registrationID` is the default.	No
advertiserID	string	The Pseudonymized Registration ID advertiser ID. Only one of `registrationID` or `advertiserID` will be populated based on API key setting. `registrationID` is the default.	No

ZipCodeInfo

Name	Type	Description	Required
uniqueDeviceCount	number (int64)	The number of unique Pseudonymized Registration IDs for the search feature in the zip code that met the search criteria regardless of the `inMin` requirement.	Yes
uniqueDeviceInMinCount	number (int64)	The number of unique Pseudonymized Registration IDs for the search feature in the zip code that met the search criteria including the `inMin` requirement.	Yes

CountyInfo

Name	Type	Description	Required
uniqueDeviceCount	number (int64)	The number of unique Pseudonymized Registration IDs for the search feature in the county that met the search criteria regardless of the `inMin` requirement.	Yes
uniqueDeviceInMinCount	number (int64)	The number of unique Pseudonymized Registration IDs for the search feature in the county that met the search criteria including the `inMin` requirement.	Yes
county	String	The name of the county.	Yes
countyFips	String	The Federal Information Processing Standard unique identifier of the county within the state.	Yes
county	String	The name of the state.	Yes
countyFips	String	The Federal Information Processing Standard unique identifier of the state.	Yes

CBGInfo

Name	Type	Description	Required
uniqueDeviceCount	number (int64)	The number of unique Pseudonymized Registration IDs for the search feature in the census block group, CBG, that met the search criteria regardless of the `inMin` requirement.	Yes

Name	Type	Description	Required
uniqueDeviceInMinCount	number (int64)	The number of unique Pseudonymized Registration IDs for the search feature in the census block group, CBG, that met the search criteria including the `inMin` requirement.	Yes

AreasDevicesResponseCRS

Name	Type	Description	Required
type	string	Required by GeoJSON specification if `crs` is included in feature collection. Should be "name".	Yes
properties	AreasDevicesResponseCRSProperties	The properties of the GeoJSON CRS object. Used by the API for non-feature-specific search parameters and results since the GeoJSON specification does not support a `properties` field in a FeatureCollection.	No

AreasDevicesResponseCRSProperties

Name	Type	Description	Required
inMin	number (int32)	Default 0. Specifies the number of search features a Pseudonymized Registration ID must be seen in for it to be counted/returned in the result set. Must be less than or equal to the number of search features in the request. For example, if three search features are in the request, `inMin` must be less than or equal to three. For the previous example, if `inMin` is two, then a Pseudonymized Registration ID must have been seen in at least two of the three search features in order to be returned. If the Pseudonymized Registration ID was seen in all three, it would be returned as well. However, if the Pseudonymized Registration ID was seen in only one of the search features, it would not be returned. A negative or zero `inMin` value behaves the same as an `inMin` value of 1. An `inMin` value of 0 is used by default so empty requests with no specified search features are acceptable.	Yes
name	String	Always "urn:ogc:def:crs:OGC::CRS84".	Yes
uniqueDeviceCount	number (int64)	The number of unique Pseudonymized Registration IDs across all search features that met the search criteria regardless of the `inMin` requirement.	Yes
uniqueDeviceInMinCount	number (int64)	The number of unique Pseudonymized Registration IDs across all search features that met the search criteria including the `inMin` requirement.	Yes
totalSignalCount	number (int64)	The number of unique observations across all search features that met the search criteria regardless of the `inMin` requirement.	Yes
totalSignalInMinCount	number (int64)	The number of unique observations across all search features that met the search criteria including the `inMin` requirement.	Yes
placeIdentifier	[string]	Default `NONE`. A comma-separated list of geospatial regions to return aggregate data for. Acceptable string values are `NONE`, `ZIPCODE`, `COUNTY`, `CBG`. If the list contains `NONE`, no place identifier information will be counted. `NONE` is mutually exclusive with all other options.	No
responseType	string	Optional. Default `DIRECT`. Possible values: `DIRECT` and `EXPORT`. Case insensitive. `DIRECT` indicates to return the results of the search as part of the JSON response body. `EXPORT` indicates that the results should be exported as files made available via download URLs from the `requestStatus/{requestID}` endpoint. `DIRECT` requests wait for the results to be available before returning while `EXPORT` requests run as a background task after being validated. As a result, a `DIRECT` request could take a minute for the results to come back while the same request as an `EXPORT` would return in seconds but the user would have to wait for the results to become available via the `requestStatus/{requestID}` endpoint.	No

Name	Type	Description	Required
includeHeaders	boolean	Whether or not the exported files have a header row or not. `true` indicated the output files have a header row. `false` indicates that the files do not have a header row.	No
compressOutputFiles	boolean	Whether or not the exported files are compressed or uncompressed. `true` indicates the output files are gzip compressed. `false` indicates that the files are not compressed.	No
exportLimit	integer	Specifies max number of rows output files have (not including optional header row). If defined, psv output files has at most `exportLimit` rows. If not defined, psv output files have no row limit.	No
requestID	string	Used by the `requestStatus/{requestID}` endpoint to check the status of the request and to get its results when ready.	Yes

DevicesTrendsRequestFeatureCollection

Name	Type	Description	Required
type	string	The GeoJSON object type. Must be "FeatureCollection".	Yes
crs	DevicesTrendsRequestCRS	The GeoJSON FeatureCollection CRS. Because the GeoJSON specification does not support properties in a FeatureCollection, in order to have non-feature-specific request or response properties, the `properties` of the CRS object are used instead. The rest of the CRS information is ignored by the API.	No
features	[DevicesTrendsRequestFeature]	A list of GeoJSON Features. The `geometry` of the feature is used to identify the user's desired search area and the `properties` of the feature are used to send and receive feature-specific search parameters and results.	Yes

DevicesTrendsRequestFeature

Name	Type	Description	Required
type	string	The GeoJSON object type. This field must be "Feature".	Yes
id	string	The GeoJSON specification allows for an `id` string field to provided per Feature to identify it. A user may provide an `id` for the feature here or as an `id` field in the properties of the feature. If a value is provided here, it will override the `id` field in the properties. If no identifier is provided for the feature in either location, the API will assign it a random UUIDv4 as its `id`. When the results of a request are exported as PSV, the feature ID will be included as a column to identify which search feature the results are associated with. If a user wants to export their results, it is recommended that the user specify the `id` per feature since otherwise a random identifier will be used and the user will be unaware of which identifier goes with which feature.	No
properties	DevicesTrendsRequestFeatureProperties	The properties of the feature. Used to provide per-feature request parameters and to return per-feature results.	No
geometry	GeoJSON Polygon, MultiPolygon, Point, LineString	The GeoJSON geometry of the feature. Required by the API for this endpoint to be a GeoJSON Polygon, MultiPolygon, Point, LineString. See GeoJSON specification for details about GeoJSON objects.	Yes

DevicesTrendsRequestFeatureProperties

Name	Type	Description	Required
id	string	Optional. Default random UUIDv4. The unique identifier of the feature. Overridden by feature's `id` field.	No
startDateTimeEpochMS	number (int64)	Optional. Default is the beginning of the day 7 days prior to `endDateTimeEpochMS`. The start of the time range to search for, expressed as epoch milliseconds.	No
endDateTimeEpochMS	number (int64)	Optional. Default is time of request. The end of the time range to search for, expressed as epoch milliseconds.	No
startDateTime	string	Optional. Default is the beginning of the day 7 days prior to `endDateTime`. The start of the time range to search for, expressed as an ISO-8601 offset timestamp.	No
endDateTime	string	Optional. Default is time of request. The end of the time range to search for, expressed as ISO-8601 offset timestamp.	No
startTimeOfDay	string	Optional. Default null. The string representation of the start time of day that observations must be after in order to be counted. An overnight filter is supported where `startTimeOfDay` is after `endTimeOfDay`. A user must supply either both `startTimeOfDay` and `endTimeOfDay` or neither since both are necessary to implement the filter. Format is `HH:mm:ss'Z'`.	No
endTimeOfDay	string	Optional. Default null. The string representation of the end time of day that observations must be before in order to be counted. An overnight filter is supported where `startTimeOfDay` is after `endTimeOfDay`. A user must supply either both `startTimeOfDay` and `endTimeOfDay` or neither since both are necessary to implement the filter. Format is `HH:mm:ss'Z'`.	No
includeRegistrationIDs	[string]	Default empty array. A set of registration IDs. A registration ID must be in this set to be counted and/or returned in results. A maximum of 10k entries can be provided.	No
excludeRegistrationIDs	[string]	Default empty array. A set of registration IDs. A registration ID must not be in this set to be counted and/or returned in the results. A maximum of 10k entries can be provided.	No
includeAdvertiserIDs	[string]	Default empty array. A set of advertiser IDs. An advertiser ID must be in this set to be counted and/or returned in results. This field replaces `includeRegistrationIDs` if the API key is configured to return `maid`. A maximum of 10k entries can be provided.	No
excludeAdvertiserIDs	[string]	Default empty array. A set of advertiser IDs. An advertiser ID must not be in this set to be counted and/or returned in the results. This field replaces `excludeRegistrationIDs` if the API key is configured to return `maid`. A maximum of 10k entries can be provided.	No
includeFlags	number (int64)	Optional. Default null. A bit array in the form of an int64. This search parameter is used to include observations in the results based on the observation's flags value. The bit indices of `includeFlags` correspond to the bit indices of the observations flags. The bitwise AND of `includeFlags` and a observation's flags value must be non-zero for the observation to be counted in the search results.	No
includeExactFlags	number (int64)	Optional. Default null. A bit array in the form of an int64. This search parameter is used to include observations in the results based on the observation's flags value. The bit indices of `includeExactFlags` correspond to the bit indices of the observations flags. The `includeExactFlags` value must exactly equal a observation's flags value for the observation to be counted in the search results.	No

Name	Type	Description	Required
excludeFlags	number (int64)	Optional. Default null. A bit array in the form of an int64. This search parameter is used to filter out observations from the results based on the observation's flags value. The bit indices of `excludeFlags` correspond to the bit indices of the observations flags. When the result of a bitwise AND of `excludeFlags` and a observation's flags value is non-zero, then the observation will not be counted in the search results.	No
excludeExactFlags	number (int64)	Optional. Default null. A bit array in the form of an int64. This search parameter is used to filter out observations from the results based on the observation's flags value. The bit indices of `excludeExactFlags` correspond to the bit indices of the observations flags. When `excludeExactFlags` exactly equals a observation's flags value, then the observation will not be counted in the search results.	No
returnDeviceCountByGeoHash	boolean	Default false. If true, the `geoHashDeviceCount` field is populated.	No

DevicesTrendsRequestCRS

Name	Type	Description	Required
type	string	Required by GeoJSON specification if `crs` is included in feature collection. Should be "name".	Yes
properties	DevicesTrendsRequestCRSProperties	The properties of the GeoJSON CRS object. Used by the API for non-feature-specific search parameters and results since the GeoJSON specification does not support a `properties` field in a FeatureCollection.	No

DevicesTrendsRequestCRSProperties

Name	Type	Description	Required
inMin	number (int32)	Default 0. Specifies the number of search features a Pseudonymized Registration ID must be seen in for it to be counted/returned in the result set. Must be less than or equal to the number of search features in the request. For example, if three search features are in the request, `inMin` must be less than or equal to three. For the previous example, if `inMin` is two, then a Pseudonymized Registration ID must have been seen in at least two of the three search features in order to be returned. If the Pseudonymized Registration ID was seen in all three, it would be returned as well. However, if the Pseudonymized Registration ID was seen in only one of the search features, it would not be returned. A negative or zero `inMin` value behaves the same as an `inMin` value of 1. An `inMin` value of 0 is used by default so empty requests with no specified search features are acceptable.	No
responseType	string	Optional. Default `DIRECT`. Possible values: `DIRECT` and `EXPORT`. Case insensitive. `DIRECT` indicates to return the results of the search as part of the JSON response body. `EXPORT` indicates that the results should be exported as files made available via download URLs from the `requestStatus/{requestID}` endpoint. `DIRECT` requests wait for the results to be available before returning while `EXPORT` requests run as a background task after being validated. As a result, a `DIRECT` request could take a minute for the results to come back while the same request as an `EXPORT` would return in seconds but the user would have to wait for the results to become available via the `requestStatus/{requestID}` endpoint.	No
includeHeaders	Boolean	Optional boolean, default true. Specifies whether output psv files include a header row. If true, psv output files will include a header row. If false, psv output files will not include a header row.	No
compressOutputFiles	Boolean	Optional boolean, default true. Specifies whether output files are gzip compressed. If true, output files are gzip compressed. If false, output files are not compressed.	No

Name	Type	Description	Required
exportLimit	number (int64)	Optional integer, default null. Specifies max number of rows output files will have (not including optional header row). If defined, psv output files will have at most exportLimit rows. If not defined, psv output files have no row limit.	No
placeIdentifier	[string]	Default `NONE`. A comma-separated list of geospatial regions to return aggregate data for. Acceptable string values are `NONE`, `ZIPCODE`, `COUNTY`, `CBG`. If the list contains `NONE`, no place identifier information will be counted. `NONE` is mutually exclusive with all other options.	No

DevicesTrendsResponseFeatureCollection

Name	Type	Description	Required
type	string	The GeoJSON object type. The `/observations/geo/search` endpoint requires that the request body is a FeatureCollection. As a result this field must be "FeatureCollection".	Yes
crs	DevicesTrendsResponseCRS	The GeoJSON FeatureCollection CRS. Because the GeoJSON specification does not support properties in a FeatureCollection, in order to have non-feature-specific request or response properties, the `properties` of the CRS object are used instead. The rest of the CRS information is ignored by the API.	No
features	[DevicesTrendsResponseFeature]	A list of GeoJSON Features. The `geometry` of the feature is used to identify the user's desired search area and the `properties` of the feature are used to send and receive feature-specific search parameters and results.	Yes

DevicesTrendsResponseFeature

Name	Type	Description	Required
type	string	The GeoJSON object type. This field must be "Feature".	Yes
id	string	The GeoJSON specification allows for an `id` string field to provided per Feature to identify it. A user may provide an `id` for the feature here or as an `id` field in the properties of the feature. If a value is provided here, it will override the `id` field in the properties. If no identifier is provided for the feature in either location, the API will assign it a random UUIDv4 as its `id`. When the results of a request are exported as PSV, the feature ID will be included as a column to identify which search feature the results are associated with. If a user wants to export their results, it is recommended that the user specify the `id` per feature since otherwise a random identifier will be used and the user will be unaware of which identifier goes with which feature.	No
properties	DevicesTrendsResponseFeatureProperties	The properties of the feature. Used to provide per-feature request parameters and to return per-feature results.	No
geometry	GeoJSON Polygon, MultiPolygon, Point, or LineString	The GeoJSON geometry of the feature. Required by the API for this endpoint to be a GeoJSON Polygon, MultiPolygon, Point, or LineString. See GeoJSON specification for details about GeoJSON objects.	Yes

DevicesTrendsResponseFeatureProperties

Name	Type	Description	Required

Name	Type	Description	Required
id	string	Optional. Default random UUIDv4. The unique identifier of the feature. Overridden by feature's `id` field.	No
startDateTimeEpochMS	number (int64)	Optional. Default is the beginning of the day 7 days prior to `endDateTimeEpochMS`. The start of the time range to search for, expressed as epoch milliseconds.	No
endDateTimeEpochMS	number (int64)	Optional. Default is time of request. The end of the time range to search for, expressed as epoch milliseconds.	No
startDateTime	string	Optional. Default is the beginning of the day 7 days prior to `endDateTime`. The start of the time range to search for, expressed as an ISO-8601 offset timestamp.	No
endDateTime	string	Optional. Default is time of request. The end of the time range to search for, expressed as ISO-8601 offset timestamp.	No
startTimeOfDay	string	Optional. Default null. The string representation of the start time of day that observations must be after in order to be counted. An overnight filter is supported where `startTimeOfDay` is after `endTimeOfDay`. A user must supply either both `startTimeOfDay` and `endTimeOfDay` or neither since both are necessary to implement the filter. Format is `HH:mm:ss'Z'`.	No
endTimeOfDay	string	Optional. Default null. The string representation of the end time of day that observations must be before in order to be counted. An overnight filter is supported where `startTimeOfDay` is after `endTimeOfDay`. A user must supply either both `startTimeOfDay` and `endTimeOfDay` or neither since both are necessary to implement the filter. Format is `HH:mm:ss'Z'`.	No
includeRegistrationIDs	[string]	Default empty array. A set of registration IDs. A registration ID must be in this set to be counted and/or returned in results. A maximum of 10k entries can be provided.	No
excludeRegistrationIDs	[string]	Default empty array. A set of registration IDs. A registration ID must not be in this set to be counted and/or returned in the results. A maximum of 10k entries can be provided.	No
includeAdvertiserIDs	[string]	Default empty array. A set of advertiser IDs. An advertiser ID must be in this set to be counted and/or returned in results. This field replaces `includeRegistrationIDs` if the API key is configured to return `maid`. A maximum of 10k entries can be provided.	No
excludeAdvertiserIDs	[string]	Default empty array. A set of advertiser IDs. An advertiser ID must not be in this set to be counted and/or returned in the results. This field replaces `excludeRegistrationIDs` if the API key is configured to return `maid`. A maximum of 10k entries can be provided.	No
includeFlags	number (int64)	Optional. Default null. A bit array in the form of an int64. This search parameter is used to include observations in the results based on the observation's flags value. The bit indices of `includeFlags` correspond to the bit indices of the observations flags. The bitwise AND of `includeFlags` and a observation's flags value must be non-zero for the observation to be counted in the search results.	No

Name	Type	Description	Required
includeExactFlags	number (int64)	Optional. Default null. A bit array in the form of an int64. This search parameter is used to include observations in the results based on the observation's flags value. The bit indices of `includeExactFlags` correspond to the bit indices of the observations flags. The `includeExactFlags` value must exactly equal a observation's flags value for the observation to be counted in the search results.	No
excludeFlags	number (int64)	Optional. Default null. A bit array in the form of an int64. This search parameter is used to filter out observations from the results based on the observation's flags value. The bit indices of `excludeFlags` correspond to the bit indices of the observations flags. When the result of a bitwise AND of `excludeFlags` and a observation's flags value is non-zero, then the observation will not be counted in the search results.	No
excludeExactFlags	number (int64)	Optional. Default null. A bit array in the form of an int64. This search parameter is used to filter out observations from the results based on the observation's flags value. The bit indices of `excludeExactFlags` correspond to the bit indices of the observations flags. When `excludeExactFlags` exactly equals a observation's flags value, then the observation will not be counted in the search results.	No
returnDeviceCountByGeoHash	boolean	Default false. If true, the `geoHashDeviceCount` field is populated.	No
geoHashWidthHeights	[GeoHashWidthHeight]	List of geo hash lengths along with their width and heights used to query the database for this search feature.	No
geoHashDeviceCount	[GeoHashDeviceCount]	Populated when `returnPerGeoHashInfo` is true, otherwise empty. Each entry contains the geo hash string, its center-point latitude and longitude, and result metrics.	No
overallDeviceCounts	DevicesTrends	A breakdown of the number of unique Pseudonymized Registration IDs found for the search feature, search criteria (including the `inMin` parameter). The breakdown includes the daily, weekly, monthly, and overall unique Pseudonymized Registration ID counts.	No
zipCodeDeviceCounts	Object: key -> DevicesTrends	An object where each key is a zipcode and the value is a breakdown of the number of unique Pseudonymized Registration IDs for that zipcode found for the search feature, search criteria (including the `inMin` parameter). The breakdown includes the daily, weekly, monthly, and overall unique Pseudonymized Registration ID counts. Only returned if `placeIdentifier` includes `ZIPCODE`.	No
countyDeviceCounts	Object: key -> DevicesTrends	An object where each key is a county and the value is a breakdown of the number of unique Pseudonymized Registration IDs for that county found for the search feature, search criteria (including the `inMin` parameter). The breakdown includes the daily, weekly, monthly, and overall unique Pseudonymized Registration ID counts. Only returned if `placeIdentifier` includes `COUNTY`.	No

Name	Type	Description	Required
cbgDeviceCounts	Object: key -> DevicesTrends	An object where each key is a census block group (CBG) identifier and the value is a breakdown of the number of unique Pseudonymized Registration IDs for that CBG found for the search feature, search criteria (including the `inMin` parameter). The breakdown includes the daily, weekly, monthly, and overall unique Pseudonymized Registration ID counts. Only returned if `placeIdentifier` includes `CBG`.	No

DevicesTrends

Name	Type	Description	Required
countsPerDay	Object: key -> number (int32)	An object where each key is a UTC date string representing that UTC date. The value is the number of Pseudonymized Registration IDs for that day found for the search feature, search criteria (including the `inMin` parameter), and place identifier if applicable. The days included in the response are determined by the time range of the request. Because the user can control the time range down to the second, it is possible, and likely, that the first and last day of the time range are only partially covered with the Pseudonymized Registration ID counts returned for those only representing that partial coverage.	Yes
countsPerWeek	Object: key -> number (int32)	An object where each key is a UTC date string representing the beginning of the week. The value is the number of Pseudonymized Registration IDs for that week found for the search feature, search criteria (including the `inMin` parameter), and place identifier if applicable. The weeks included in the response are determined by the time range of the request. Because the user can control the time range down to the second, it is possible, and likely, that the first and last weeks of the time range are only partially covered with the Pseudonymized Registration ID counts returned for those only representing that partial coverage.	Yes
countsPerMonth	Object: key -> number (int32)	An object where each key is a UTC date string representing the beginning of the month. The value is the number of Pseudonymized Registration IDs for that month found for the search feature, search criteria (including the `inMin` parameter), and place identifier if applicable. The months included in the response are determined by the time range of the request. Because the user can control the time range down to the second, it is possible, and likely, that the first and last months of the time range are only partially covered with the Pseudonymized Registration ID counts returned for those only representing that partial coverage.	Yes
totalCount	Object: key -> number (int32)	The overall number of unique Pseudonymized Registration IDs found for the search feature, search criteria (including the `inMin` parameter), and place identifier if applicable.	Yes

DevicesTrendsResponseCRS

Name	Type	Description	Required
type	string	Required by GeoJSON specification if `crs` is included in feature collection. Should be "name".	Yes
properties	DevicesTrendsResponseCRSProperties	The properties of the GeoJSON CRS object. Used by the API for non-feature-specific search parameters and results since the GeoJSON specification does not support a `properties` field in a FeatureCollection.	No

DevicesTrendsResponseCRSProperties

Name	Type	Description	Required

Name	Type	Description	Required
inMin	number (int32)	Default 0. Specifies the number of search features a Pseudonymized Registration ID must be seen in for it to be counted/returned in the result set. Must be less than or equal to the number of search features in the request. For example, if three search features are in the request, `inMin` must be less than or equal to three. For the previous example, if `inMin` is two, then a Pseudonymized Registration ID must have been seen in at least two of the three search features in order to be returned. If the Pseudonymized Registration ID was seen in all three, it would be returned as well. However, if the Pseudonymized Registration ID was seen in only one of the search features, it would not be returned. A negative or zero `inMin` value behaves the same as an `inMin` value of 1. An `inMin` value of 0 is used by default so empty requests with no specified search features are acceptable.	Yes
name	String	Always "urn:ogc:def:crs:OGC::CRS84".	Yes
uniqueDeviceCount	number (int64)	The number of unique Pseudonymized Registration IDs across all search features that met the search criteria regardless of the `inMin` requirement.	Yes
uniqueDeviceInMinCount	number (int64)	The number of unique Pseudonymized Registration IDs across all search features that met the search criteria including the `inMin` requirement.	Yes
totalSignalCount	number (int64)	The number of unique observations across all search features that met the search criteria regardless of the `inMin` requirement.	Yes
totalSignalInMinCount	number (int64)	The number of unique observations across all search features that met the search criteria including the `inMin` requirement.	Yes
placeIdentifier	[string]	Default `NONE`. A comma-separated list of geospatial regions to return aggregate data for. Acceptable string values are `NONE`, `ZIPCODE`, `COUNTY`, `CBG`. If the list contains `NONE`, no place identifier information will be counted. `NONE` is mutually exclusive with all other options.	No
responseType	string	Optional. Default `DIRECT`. Possible values: `DIRECT` and `EXPORT`. Case insensitive. `DIRECT` indicates to return the results of the search as part of the JSON response body. `EXPORT` indicates that the results should be exported as files made available via download URLs from the `requestStatus/{requestID}` endpoint. `DIRECT` requests wait for the results to be available before returning while `EXPORT` requests run as a background task after being validated. As a result, a `DIRECT` request could take a minute for the results to come back while the same request as an `EXPORT` would return in seconds but the user would have to wait for the results to become available via the `requestStatus/{requestID}` endpoint.	No
includeHeaders	boolean	Whether or not the exported files have a header row or not. `true` indicated the output files have a header row. `false` indicates that the files do not have a header row.	No
compressOutputFiles	boolean	Whether or not the exported files are compressed or uncompressed. `true` indicates the output files are gzip compressed. `false` indicates that the files are not compressed.	No
exportLimit	integer	Specifies max number of rows output files have (not including optional header row). If defined, psv output files has at most `exportLimit` rows. If not defined, psv output files have no row limit.	No
requestID	string	Used by the `requestStatus/{requestID}` endpoint to check the status of the request and to get its results when ready.	Yes