

Insights API Deep Dive

Search CTRL-K

Insights API Deep Dive >

https://staging.api.qloo.com/v2/insights

Returns taste-based insights based on the input parameters you provide.

Qloo's Insights API helps uncover the underlying factors that shape human preferences, offering cultural intelligence about how people relate to different entities like brands, artists, destinations, and more. It draws from billions of signals to deliver nuanced, taste-based insights that reflect real-world behavior and affinities.

This Deep Dive is designed to help users explore the full capabilities of the Insights API, emphasizing how different parameters shape results. It allows you to view and test the complete set of supported parameters directly on this page, making it easier to explore functionality in context and understand how different inputs impact responses.

Helpful Resources

- Need to understand how current parameters map to legacy fields? Visit the Parameter Reference.
- Want to know which parameters are valid for each entity type (like Actor, Destination, or Brand)? See the Entity Type Parameter Guide.

Participating in the Qloo LLM Hackathon?

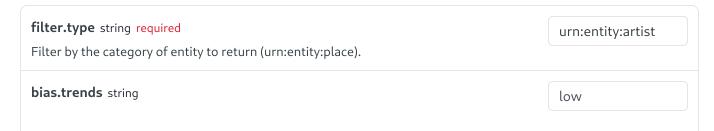
Use the dedicated hackathon API URL for all requests:

https://hackathon.api.gloo.com

Hackathon API keys won't work on staging or production URLs.

Not signed up yet? Join the hackathon <u>here</u>.

QUERY PARAMS



The level of impact a trending entity has on the results. Supported by select categories only.

diversify.by string

Limits results to a set number of high-affinity entities per city. Set this to "properties.geocode.city" to enable city-based diversification. Cities are ranked based on the highest-affinity entity within them, and entities within each city are ordered by their individual affinities.

properties.geocod€

diversify.take integer ≥ 1

Sets the maximum number of results to return per city when using "diversify.by": "properties.geocode.city". For example, if set to 5, the response will include up to 5 entities with the highest affinities in each city.

5

true

feature.explainability boolean

When set to true, the response includes explainability metadata for each recommendation and for the overall result set. Default is set to false.

Per-recommendation: Each result includes a query.explainability section showing which input entities (e.g. signal.interests.entities) contributed to the recommendation and by how much. Scores are normalized between 0–1. Entities with scores ≥ 0.1 are always included; those below may be omitted to reduce response size.

Aggregate impact: The top-level query.explainability object shows average influence of each input entity across top-N result subsets (e.g. top 3, 5, 10, all).

Note: If explainability cannot be computed for the request, a warning is included under query.explainability.warning, but results still return normally.

filter.address string

Filter by address using a partial string query.

123 Main St

filter.content_rating string

Filter by a comma-separated list of content ratings based on the MPAA film rating system, which determines suitability for various audiences.

PG

filter.date_of_birth.max date

Filter by the most recent date of birth desired for the queried person.

2004-01-01

filter.date_of_birth.min date

Filter by the earliest date of birth desired for the queried person.

1996-01-01

filter.date_of_death.max date

Filter by the most recent date of death desired for the gueried person.

2004-01-01

filter.date_of_death.min date

Filter by the earliest date of death desired for the gueried person.

1985-01-01

filter.exclude.tags string Exclude entities associated with a comma-separated list of tags.	urn:tag:genre:medi
operator.exclude.tags string Specifies how multiple filter.exclude.tags values are combined in the query. Use "union" (equivalent to a logical "or") to exclude results that contain at least one of the specified tags, or "intersection" (equivalent to a logical "and") to exclude only results that contain all specified tags. The default is "union".	union
filter.exists string Filter results to include only entities that have one or more specified properties. Use properties image to return only entities that include an image URL.	properties.image
filter.external.exists string Filter by a comma-separated list of external keys. (resy michelin tablet).	resy,michelin
operator.filter.external.exists string Specifies how multiple filter.external.exists values are combined in the query. Use "union" (equivalent to a logical "or") to return results that match at least one of the specified external keys (e.g., resy, michelin, or tablet), or "intersection" (equivalent to a logical "and") to return only results that match all specified external keys. The default is "union".	union
<pre>filter.external.resy.count.max integer ≥ 0 Filter places to include only those with a Resy rating count less than or equal to the specified maximum. Applies only to entities with filter.type of urn:entity:place.</pre>	200
<pre>filter.external.resy.count.min integer ≥ 0 Filter places to include only those with a Resy rating count greater than or equal to the specified minimum. Applies only to entities with filter.type of urn:entity:place.</pre>	20
filter.external.resy.party_size.max integer ≥ 1 Filter by the maximum supported party size required for a Point of Interest.	10
filter.external.resy.party_size.min integer ≥ 1 Filter by the minimum supported party size required for a Point of Interest.	2
filter.external.resy.rating.max number 0 to 5 Filter places to include only those with a Resy rating less than or equal to the specified maximum (1–5 scale). Applies only to entities with filter.type of urn:entity:place.	4.5
filter.external.resy.rating.min number 0 to 5 Filter places to include only those with a Resy rating greater than or equal to the specified minimum (1–5 scale). Applies only to entities with filter.type of	3.5

or the specified minimum. This filter only applies to entities with filter.type of urn:entity:place. ilter.external.tripadvisor.rating.count.max integer ≥ 0 ilter places to include only those with a Tripadvisor review count less than or equal to the specified maximum. This filter only applies to entities with filter.type of urn:entity:place. ilter.external.tripadvisor.rating.max number 0 to 5 ilter places to include only those with a Tripadvisor rating less than or equal to the specified maximum. This filter only applies to entities with filter.type of urn:entity:place. ilter.external.tripadvisor.rating.min number 0 to 5 ilter places to include only those with a Tripadvisor rating greater than or equal to the specified minimum. This filter only applies to entities with filter.type of urn:entity:place. ilter.finale_year.max integer ilter by the latest desired year for the final season of a TV show. ilter.finale_year.min integer ilter by the earliest desired year for the final season of a TV show. ilter.gender string ilter place to align with a specific gender identity. Used to personalize output based on known or inferred gender preferences. ilter.geocode.admin1_region string ilter by properties.geocode.admin1_region . Exact match (usually state). ilter.geocode.admin2_region string ilter by properties.geocode.admin2_region . Exact match (often county or borough). ilter.geocode.country_code string New York County ilter.geocode.country_code string	ilter.external.tripadvisor.rating.count.min integer ≥ 0	50
Filter places to include only those with a Tripadvisor review count less than or equal to the specified maximum. This filter only applies to entities with filter, type of urn:entity:place. Filter.external.tripadvisor.rating.max number 0 to 5 Filter places to include only those with a Tripadvisor rating less than or equal to the specified maximum. This filter only applies to entities with filter.type of urn:entity:place. Filter.external.tripadvisor.rating.min number 0 to 5 Filter places to include only those with a Tripadvisor rating greater than or equal to the specified minimum. This filter only applies to entities with filter.type of urn:entity:place. Filter places to include only those with a Tripadvisor rating greater than or equal to the specified minimum. This filter only applies to entities with filter.type of urn:entity:place. Filter finale_year.max integer Filter by the latest desired year for the final season of a TV show. Filter.finale_year.min integer Filter by the earliest desired year for the final season of a TV show. Filter.gender string Filter results to align with a specific gender identity. Used to personalize output based on known or inferred gender preferences. Filter.geocode.admin1_region string Filter.geocode.admin1_region string Filter.geocode.admin2_region string Filter by properties.geocode.admin2_region . Exact match (often county or porough). Filter.geocode.country_code string US	Filter places to include only those with a Tripadvisor review count greater than or equal to the specified minimum. This filter only applies to entities with filter.type of urn:entity:place.	
Filter places to include only those with a Tripadvisor rating less than or equal to the specified maximum. This filter only applies to entities with filter.type of urn:entity:place. Filter.external.tripadvisor.rating.min number 0 to 5 3 3	filter.external.tripadvisor.rating.count.max integer ≥ 0	500
Filter.external.tripadvisor.rating.min number 0 to 5 Filter.external.tripadvisor.rating.min number 0 to 5 Filter.external.tripadvisor.rating.min number 0 to 5 Filter places to include only those with a Tripadvisor rating greater than or equal to the specified minimum. This filter only applies to entities with filter.type of urn:entity:place. Filter.finale_year.max integer Filter by the latest desired year for the final season of a TV show. Filter.finale_year.min integer Filter by the earliest desired year for the final season of a TV show. Filter.gender string Filter results to align with a specific gender identity. Used to personalize output based on known or inferred gender preferences. Filter.geocode.admin1_region string Filter.geocode.admin1_region string Filter.peocode.admin2_region string Filter.peocode.admin2_region string Filter.geocode.admin2_region string New York County Filter.geocode.country_code string US	the specified maximum. This filter only applies to entities with filter.type of	
filter.external.tripadvisor.rating.min number 0 to 5 Filter places to include only those with a Tripadvisor rating greater than or equal to the specified minimum. This filter only applies to entities with filter.type of urn:entity:place. filter.finale_year.max integer Filter by the latest desired year for the final season of a TV show. filter.finale_year.min integer Filter by the earliest desired year for the final season of a TV show. filter.gender string Filter results to align with a specific gender identity. Used to personalize output based on known or inferred gender preferences. filter.geocode.admin1_region string Filter by properties.geocode.admin1_region . Exact match (usually state). NY New York County Filter by properties.geocode.admin2_region . Exact match (often county or borough).	filter.external.tripadvisor.rating.max number 0 to 5	4
Filter places to include only those with a Tripadvisor rating greater than or equal to the specified minimum. This filter only applies to entities with filter.type of urn:entity:place. filter.finale_year.max integer Filter by the latest desired year for the final season of a TV show. filter.finale_year.min integer Filter by the earliest desired year for the final season of a TV show. filter.gender string Filter results to align with a specific gender identity. Used to personalize output based on known or inferred gender preferences. filter.geocode.admin1_region string Filter by properties.geocode.admin1_region . Exact match (usually state). filter.geocode.admin2_region string Filter by properties.geocode.admin2_region . Exact match (often county or borough).	specified maximum. This filter only applies to entities with filter.type of	
specified minimum. This filter only applies to entities with filter.type of urn:entity:place. filter.finale_year.max integer Filter by the latest desired year for the final season of a TV show. filter.finale_year.min integer Filter by the earliest desired year for the final season of a TV show. filter.gender string Filter results to align with a specific gender identity. Used to personalize output based on known or inferred gender preferences. filter.geocode.admin1_region string Filter by properties.geocode.admin1_region . Exact match (usually state). filter.geocode.admin2_region string Filter by properties.geocode.admin2_region . Exact match (often county or borough).	filter.external.tripadvisor.rating.min number 0 to 5	3
Filter by the latest desired year for the final season of a TV show. filter.finale_year.min integer Filter by the earliest desired year for the final season of a TV show. filter.gender string Filter results to align with a specific gender identity. Used to personalize output based on known or inferred gender preferences. filter.geocode.admin1_region string Filter by properties.geocode.admin1_region . Exact match (usually state). NY Filter.geocode.admin2_region string Filter.geocode.admin2_region . Exact match (often county or borough). US	specified minimum. This filter only applies to entities with filter.type of	
filter.finale_year.min integer Filter by the earliest desired year for the final season of a TV show. filter.gender string Filter results to align with a specific gender identity. Used to personalize output based on known or inferred gender preferences. filter.geocode.admin1_region string Filter by properties.geocode.admin1_region . Exact match (usually state). filter.geocode.admin2_region string Filter by properties.geocode.admin2_region . Exact match (often county or borough). filter.geocode.country_code string US	filter.finale_year.max integer	2021
Filter by the earliest desired year for the final season of a TV show. filter.gender string Filter results to align with a specific gender identity. Used to personalize output based on known or inferred gender preferences. filter.geocode.admin1_region string Filter by properties.geocode.admin1_region . Exact match (usually state). filter.geocode.admin2_region string Filter by properties.geocode.admin2_region . Exact match (often county or borough). filter.geocode.country_code string US	Filter by the latest desired year for the final season of a TV show.	
filter.gender string Filter results to align with a specific gender identity. Used to personalize output based on known or inferred gender preferences. filter.geocode.admin1_region string Filter by properties.geocode.admin1_region . Exact match (usually state). filter.geocode.admin2_region string Filter by properties.geocode.admin2_region . Exact match (often county or borough). filter.geocode.country_code string US	filter.finale_year.min integer	2014
Filter results to align with a specific gender identity. Used to personalize output based on known or inferred gender preferences. filter.geocode.admin1_region string Filter by properties.geocode.admin1_region . Exact match (usually state). NY New York County Filter by properties.geocode.admin2_region . Exact match (often county or borough).	Filter by the earliest desired year for the final season of a TV show.	
on known or inferred gender preferences. filter.geocode.admin1_region string Filter by properties.geocode.admin1_region . Exact match (usually state). filter.geocode.admin2_region string Filter by properties.geocode.admin2_region . Exact match (often county or borough). filter.geocode.country_code string US	filter.gender string	female
Filter by properties.geocode.admin1_region . Exact match (usually state). filter.geocode.admin2_region string Filter by properties.geocode.admin2_region . Exact match (often county or borough). filter.geocode.country_code string US	, , , , , , , , , , , , , , , , , , , ,	
filter.geocode.admin2_region string Filter by properties.geocode.admin2_region . Exact match (often county or borough). filter.geocode.country_code string US	filter.geocode.admin1_region string	NY
Filter by properties.geocode.admin2_region . Exact match (often county or borough). filter.geocode.country_code string US	Filter by properties.geocode.admin1_region . Exact match (usually state).	
filter.geocode.country_code string US	filter.geocode.admin2_region string	New York County
Filter by properties.geocode.country_code . Exact match (two-letter country code).	filter.geocode.country_code string	US
	Filter by properties.geocode.country_code .Exact match (two-letter country code).	
filter.geocode.name string New York	filter.geocode.name string	New York

filter.hotel_class.max integer 1 to 5 Filter by the maximum desired hotel class (1-5, inclusive).	5
filter.hotel_class.min integer 1 to 5 Filter by the minimum desired hotel class (1-5, inclusive).	3
filter.hours string Filter by the day of the week the Point of Interest must be open (Monday, Tuesday, etc.).	monday
filter.latest_known_year.max integer Filter by a certain maximum year that shows were released or updated.	2023
filter.latest_known_year.min integer Filter by a certain minimum year that shows were released or updated.	2014
filter.location string Used to filter by a WKT POINT, POLYGON, MULTIPOLYGON or a single Qloo ID for a named urn:entity:locality.WKT is formatted as X then Y, therefore longitude is first(POINT(-73.99823 40.722668)). If a Qloo ID or WKT POLYGON is passed, filter.location.radius will create a fuzzy boundary when set to a value > 0.	POINT(-73.99823 4
filter.exclude.location string Exclude results that fall within a specific location, defined by either a WKT POINT, POLYGON, MULTIPOLYGON, or a Qloo ID for a named urn:entity:locality. WKT is formatted with longitude first (e.g., POINT(-73.99823 40.722668)). When using a locality ID or a WKT POLYGON, setting filter.location.radius to a value > 0 creates a fuzzy exclusion boundary.	POINT(-73.99823 4

filter.location.query

A query used to search for one or more named urn:entity:locality Qloo IDs for filtering requests, equivalent to passing the same Locality Qloo ID(s) into filter.location.

- For **GET requests**: Provide a single locality query as a string.
- For **POST requests**:
 - You can still send a single locality as a string.
 - Or you can send an array of locality names to query multiple localities at once. When multiple localities are provided, their geographic shapes are merged, and the system returns results with the highest affinities across the combined area.

Locality queries are fuzzy-matched and case-insensitive. Examples include New York City, Garden City, New York, Los Angeles, Lower East Side, and AKAs like The Big Apple. When a single locality is supplied, the response JSON includes query.locality.signal with the matched Qloo entity. If multiple are supplied, this field is omitted. By default, the API includes a tuning that also captures nearby entities just outside the official boundaries of the locality. To turn this off and limit results strictly to within the locality, set filter.location.radius=0. If no localities are found, the API returns a 400 error.

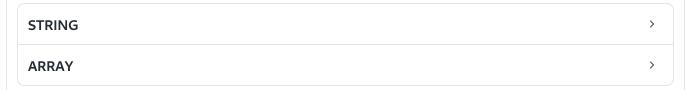


filter.exclude.location.query

A query used to exclude results based on one or more named urn:entity:locality Qloo IDs, resolved from fuzzy-matched locality names. This is equivalent to passing the resolved Locality Qloo ID(s) into filter.exclude.location.

- For **GET requests**: Provide a single locality query as a string. For **POST requests**:
 - You can still send a single locality as a string.
 - Or send an array of locality names to exclude multiple areas at once. When multiple localities are provided, their geographic shapes are merged, and the system excludes results from across the combined area.

Locality queries are case-insensitive and support common AKAs (e.g., The Big Apple for New York). When a single locality is supplied, the response includes query.locality.exclude.signal with the matched Qloo entity. If multiple are supplied, this field is omitted. If no localities are matched, the API returns a 400 error.



filter.location.geohash string

Filter by a geohash. Geohashes are generated using the Python package pygeohash with a precision of 12 characters. This parameter returns all POIs that start with the specified geohash. For example, supplying dr5rs would allow returning the geohash dr5rsjk4sr2w.

dr5rsjk4sr2w

filter.exclude.location.geohash string

Exclude all entities whose geohash starts with the specified prefix. Geohashes are generated using the Python package pygeohash with a precision of 12 characters. For example, supplying dr5rs would exclude any result whose geohash begins with dr5rs, such as dr5rsjk4sr2w.

dr5rs

filter.location.radius integer

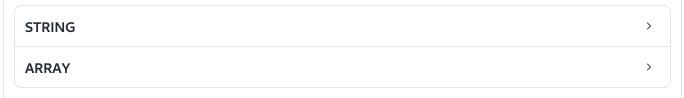
Filter by the radius (in meters) when also supplying filter.location or filter.location.query. When this parameter is **not provided**, the API applies a default tuning that slightly expands the locality boundary to include nearby entities outside its official shape. To **disable** this behavior and strictly limit results to entities inside the defined locality boundary, set filter.location.radius=0.

10000

filter.parents.types array of strings Filter by a comma-separated list of parental entity types (urn; audience; communities). Each type must match exactly. **ADD STRING** + filter.popularity.max number 0 to 1 0.98 Filter by the maximum popularity percentile a Point of Interest must have (float, between 0 and 1; closer to 1 indicates higher popularity, e.g., 0.98 for the 98th percentile). filter.popularity.min number 0 to 1 0.5 Filter by the minimum popularity percentile required for a Point of Interest (float, between 0 and 1; closer to 1 indicates higher popularity, e.g., 0.98 for the 98th percentile). filter.price_level.max integer 1 to 4 3 Filter by the maximum price level a Point of Interest can have (1/2/3/4, similar to dollar signs). filter.price_level.min integer 1 to 4 1 Filter by the minimum price level a Point of Interest can have (1|2|3|4, similar to dollar signs). filter.price_range.from integer 0 to 1000000 200 Filter places by a minimum price level, representing the lowest price in the desired range. Accepts an integer value between 0 and 1,000,000. filter.price_range.to integer 0 to 1000000 200 Filter places by a maximum price level, representing the highest price in the desired range. Accepts an integer value between 0 and 1,000,000. Only applies to places. filter.price.max float 200 maximum price filter.price.min float 1 minimum price filter.properties.business_rating.max float 3 Filter by the highest desired business rating. filter.properties.business_rating.min float 3 Filter by the lowest desired business rating. **filter.publication_year.max** number 2021

Filter by the latest desired year of initial publication for the work.	
filter.publication_year.min number	1985
Filter by the earliest desired year of initial publication for the work.	
filter.rating.max number 0 to 5	4.5
Filter by the maximum Qloo rating a Point of Interest must have (float, between 0 and 5).	
filter.rating.min number 0 to 5	3.5
Filter by the minimum Qloo rating a Point of Interest must have (float, between 0 and 5).	
filter.references_brand array of strings	
Filter by a comma-separated list of brand entity IDs. Use this to narrow down place recorbrands. For example, to include only Walmart stores, pass the Walmart brand ID. Each ID	•
ADD STRING	+
filter.release_country array of strings	
Filter by a list of countries where a movie or TV show was originally released.	
ADD STRING	+
operator.filter.release_country string	union
Specifies how multiple `filter.release_country`` values are combined in the query. Use "union" (equivalent to a logical "or") to return results that match at least one of the specified countries, or "intersection" (equivalent to a logical "and") to return only results that match all specified countries. The default is "union".	
filter.release_date.max date	2024-01-01
Filter by the latest desired release date.	
filter.release_date.min date	2021-01-01
Filter by the earliest desired release date.	
filter.release_year.max integer	2024
Filter by the latest desired release year.	
filter.release_year.min integer	1996
Filter by the earliest desired release year.	
filter.results.entities string	636E1B95-6232-43
Filter by a comma-separated list of entity IDs. Often used to assess the affinity of an entity towards input.	
filter.results.entities.query	

Search for one or more entities by name to use as filters. - For **GET requests**: Provide a single entity name as a string. - For **POST requests**: You can provide a single name or an array of names.



filter.exclude.entities string

39458DC0-F91D-4

A comma-separated list of entity IDs to remove from the results.

filter.exclude.entities.query array

This parameter can only be supplied when using POST HTTP method, since it requires JSON encoded body. The value for filter.exclude.entities.query is a JSON array with objects containing the name and address properties. For a fuzzier search, just include an array of strings. When supplied, it overwrites the filter.exclude.entities object with resolved entity IDs. The response will contain a path query.entities.exclude, with partial Qloo entities that were matched by the query. If no entities are found, the API will throw a 400 error.

ADD +

filter.results.tags array of strings

Filter by a comma-separated list of tag IDs. Often used to assess the affinity of a tag towards input.

ADD STRING +

filter.tags string

urn:tag:genre:resta

Filter by a comma-separated list of tag IDs (urn:tag:genre:restaurant:Italian).

operator.filter.tags string

union

Specifies how multiple filter.tags values are combined in the query. Use "union" (equivalent to a logical "or") to return results that match at least one of the specified tags, or "intersection" (equivalent to a logical "and") to return only results that match all specified tags. The default is "union".

offset integer

5

The number of results to skip, starting from 0. Allows arbitrary offsets but is less commonly used than page .

output.heatmap.boundary string

urn:entity:locality

Indicates the type of heatmap output desired: The default is geohashes. The other options are a city or a neighborhood.

page integer ≥1

1

The page number of results to return. This is equivalent to take + offset and is the recommended approach for most use cases.

36_to_55

male

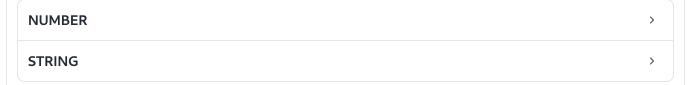
signal.demographics.age string

A comma-separated list of age ranges that influence the affinity score.

(35_and_younger|36_to_55|55_and_older).

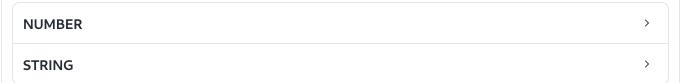
signal.demographics.age.weight

Specifies the extent to which results should be influenced by age-based demographic signals. Higher values increase the influence of age data; lower values reduce its impact.



signal.demographics.audiences.weight

Specifies the extent to which results should be influenced by the preferences of the selected audience. Higher values increase the influence of audience preferences; lower values reduce their impact.



signal.demographics.audiences array of strings

A comma-separated list of audiences that influence the affinity score. Audience IDs can be retrieved via the v2/audiences search route.



signal.demographics.gender string

Influence the affinity score based on gender (male|female).

signal.demographics.gender.weight

Specifies the extent to which results should be influenced by gender-based demographic signals. Higher values increase the influence of gender data; lower values reduce its impact.



signal.interests.entities array of strings

Allows you to supply a list of entities to influence affinity scores. You can also include a weight property that will indicate the strength of influence for each entity in your list.

- For GET requests: Provide a comma-separated list of entity IDs.
- For POST requests, you can either:
 - Send the same string of comma-separated values.

• Send an array of objects with "entity" and "weight" properties, such as: [{ "entity": "urn:entity:movie:inception", "weight": 10 }, { "entity": "urn:entity:movie:interstellar", "weight": 25 }] Weights must be greater than 0 and are relative. So, a weight of 25 means that entity will more heavily influence affinity scores than a weight of 10.

ADD STRING +

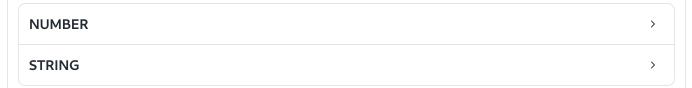
signal.interests.entities.query array

This parameter can only be supplied when using POST HTTP method, which requires a JSON-encoded body. The value should be a JSON array of objects with 'name' and 'address' properties; supports 'resolve_to' for specifying resolution to place, brand, or both.

ADD +

signal.interests.entities.weight

Specifies the extent to which results should be influenced by the relevance of entities (in-domain or cross-domain). Higher values increase the influence of entities; lower values reduce their impact.



signal.interests.tags

Allows you to supply a list of tags to influence affinity scores. You can also include a weight property that will indicate the strength of influence for each tag in your list.

- For GET requests: Provide a comma-separated list of tag IDs.
- For POST requests, you can either:
 - Send the same string of comma-separated values.
 - Send an array of objects with "tag" and "weight" properties, such as: [{ "tag": "urn:tag:genre:media:horror", "weight": 7 }, { "tag": "urn:tag:genre:media:thriller", "weight": 20 }] Weights must be greater than 0 and are relative. So, a weight of 20 means that tag will more heavily influence affinity scores than a weight of 7.



signal.interests.tags.weight

Specifies the extent to which results should be influenced by the presence of tags (taste analysis). Higher values increase the influence of tags; lower values reduce their impact.



signal.location string

The geolocation to use for geospatial results. The value will be a WKT POINT, POLYGON or a single Qloo ID for a named urn:entity:locality to filter by. WKT is formatted as X then Y, therefore longitude is first (POINT(-73.99823 40.722668)). Unlike

POINT(-73.99823 4

filter.location.radius, signal.location.radius is ignored if a Qloo ID or WKT POLYGON is passed.

signal.location.radius integer

5000

The optional radius (in meters), used when providing a WKT POINT. We generally recommend avoiding this parameter, as it overrides dynamic density discovery.

signal.location.query string

New York City

A string query used to search for a named urn:entity:locality Qloo ID for geospatial results, effectively equivalent to passing the same Locality Qloo ID into signal.location. Examples of locality queries include New York City, Garden City, New York, Los Angeles, Lower East Side, and AKAs like The Big Apple. These queries are fuzzy-matched and case-insensitive. When filter.location.query is supplied, the response JSON will include query.locality.signal, which contains the partially matched Qloo entity. If no locality is found, the API will return a 400 error.

signal.location.weight

Specifies the extent to which results should be influenced by location-based signals (geospatial). Higher values increase the influence of location; lower values reduce its impact.

NUMBER > STRING >

sort_by string

affinity

This parameter modifies the results sorting algorithm (affinity|distance). The distance option can only be used when `filter.location`` is supplied.

take integer ≥1

10

The number of results to return.

RESPONSES

200
Successful Operation
400
Bad Request
500

Internal Server Error

```
LANGUAGE
    ://
                (S)
                                        php
   Shell
               Node
                           Ruby
                                        PHP
                                                   Python
CREDENTIALS
                                                                                            HEADER ①
 Header
           X-Api-Key
  REQUEST
   $ python -m pip install requests
      import requests
                                 1
      url = "https://staging
                               3 |loo.com/v2/insights?filter.type=urn%3Aentity%3Aartist"
      headers = {"accept": "
                                 5 cation/json"}
      response = requests.ge
                                 7  headers=headers)
      print(response.text)
                                                                                         Try It!
 RESPONSE
                                                                                     EXAMPLES ~
             Click Try It! to start a request and see the response here! Or choose an example:
                                          application/json
                                               200
```

Updated 3 days ago

 ← Qloo LLM Hackathon Developer Guide

Parameter Reference →