

# 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.qloo.com`

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

Not signed up yet? Join the hackathon [here](#).

#### QUERY PARAMS

**filter.type** string **required**

Filter by the category of entity to return (urn:entity:place).

urn:entity:artist

**bias.trends** string

low

The level of impact a trending entity has on the results. Supported by select categories only.		
<b>diversify.by</b> string		<div>properties.geocode</div>
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.		
<b>diversify.take</b> integer $\geq 1$		<div>5</div>
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.		
<b>feature.explainability</b> boolean		<div>true</div>
When set to <code>true</code> , the response includes explainability metadata for each recommendation and for the overall result set. Default is set to <code>false</code> .		
<b>Per-recommendation:</b> Each result includes a <code>query.explainability</code> section showing which input entities (e.g. <code>signal.interests.entities</code> ) contributed to the recommendation and by how much. Scores are normalized between 0–1. Entities with scores $\geq 0.1$ are always included; those below may be omitted to reduce response size.		
<b>Aggregate impact:</b> The top-level <code>query.explainability</code> object shows average influence of each input entity across top-N result subsets (e.g. top 3, 5, 10, all).		
<b>Note:</b> If explainability cannot be computed for the request, a warning is included under <code>query.explainability.warning</code> , but results still return normally.		
<b>filter.address</b> string		<div>123 Main St</div>
Filter by address using a partial string query.		
<b>filter.content_rating</b> string		<div>PG</div>
Filter by a comma-separated list of content ratings based on the MPAA film rating system, which determines suitability for various audiences.		
<b>filter.date_of_birth.max</b> date		<div>2004-01-01</div>
Filter by the most recent date of birth desired for the queried person.		
<b>filter.date_of_birth.min</b> date		<div>1996-01-01</div>
Filter by the earliest date of birth desired for the queried person.		
<b>filter.date_of_death.max</b> date		<div>2004-01-01</div>
Filter by the most recent date of death desired for the queried person.		
<b>filter.date_of_death.min</b> date		<div>1985-01-01</div>
Filter by the earliest date of death desired for the queried person.		

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

urn:entity:place .		
<b>filter.external.tripadvisor.rating.count.min</b> integer $\geq 0$	<input type="text" value="50"/>	
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 <code>filter.type</code> of <code>urn:entity:place</code> .		
<b>filter.external.tripadvisor.rating.count.max</b> integer $\geq 0$	<input type="text" value="500"/>	
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 <code>filter.type</code> of <code>urn:entity:place</code> .		
<b>filter.external.tripadvisor.rating.max</b> number 0 to 5	<input type="text" value="4"/>	
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 <code>filter.type</code> of <code>urn:entity:place</code> .		
<b>filter.external.tripadvisor.rating.min</b> number 0 to 5	<input type="text" value="3"/>	
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 <code>filter.type</code> of <code>urn:entity:place</code> .		
<b>filter.finale_year.max</b> integer	<input type="text" value="2021"/>	
Filter by the latest desired year for the final season of a TV show.		
<b>filter.finale_year.min</b> integer	<input type="text" value="2014"/>	
Filter by the earliest desired year for the final season of a TV show.		
<b>filter.gender</b> string	<input type="text" value="female"/>	
Filter results to align with a specific gender identity. Used to personalize output based on known or inferred gender preferences.		
<b>filter.geocode.admin1_region</b> string	<input type="text" value="NY"/>	
Filter by <code>properties.geocode.admin1_region</code> . Exact match (usually state).		
<b>filter.geocode.admin2_region</b> string	<input type="text" value="New York County"/>	
Filter by <code>properties.geocode.admin2_region</code> . Exact match (often county or borough).		
<b>filter.geocode.country_code</b> string	<input type="text" value="US"/>	
Filter by <code>properties.geocode.country_code</code> . Exact match (two-letter country code).		
<b>filter.geocode.name</b> string	<input type="text" value="New York"/>	
Filter by <code>properties.geocode.name</code> . Exact match (usually city or town name).		

<div><b>filter.hotel_class.max</b> integer 1 to 5</div> <div>Filter by the maximum desired hotel class (1-5, inclusive).</div>	<div>5</div>
<div><b>filter.hotel_class.min</b> integer 1 to 5</div> <div>Filter by the minimum desired hotel class (1-5, inclusive).</div>	<div>3</div>
<div><b>filter.hours</b> string</div> <div>Filter by the day of the week the Point of Interest must be open (Monday, Tuesday, etc.).</div>	<div>monday</div>
<div><b>filter.latest_known_year.max</b> integer</div> <div>Filter by a certain maximum year that shows were released or updated.</div>	<div>2023</div>
<div><b>filter.latest_known_year.min</b> integer</div> <div>Filter by a certain minimum year that shows were released or updated.</div>	<div>2014</div>
<div><b>filter.location</b> string</div> <div>Used to filter by a WKT <code>POINT</code> , <code>POLYGON</code> , <code>MULTIPOLYGON</code> or a single Qloo ID for a named <code>urn:entity:locality</code> . WKT is formatted as X then Y, therefore longitude is first ( <code>POINT(-73.99823 40.722668)</code> ). If a Qloo ID or WKT <code>POLYGON</code> is passed, <code>filter.location.radius</code> will create a fuzzy boundary when set to a value <code>&gt; 0</code>.</div>	<div>POINT(-73.99823 4</div>
<div><b>filter.exclude.location</b> string</div> <div>Exclude results that fall within a specific location, defined by either a WKT <code>POINT</code> , <code>POLYGON</code> , <code>MULTIPOLYGON</code> , or a Qloo ID for a named <code>urn:entity:locality</code> . WKT is formatted with longitude first (e.g., <code>POINT(-73.99823 40.722668)</code> ). When using a locality ID or a WKT <code>POLYGON</code> , setting <code>filter.location.radius</code> to a value <code>&gt; 0</code> creates a fuzzy exclusion boundary.</div>	<div>POINT(-73.99823 4</div>
<div><b>filter.location.query</b></div> <div>A query used to search for one or more named <code>urn:entity:locality</code> Qloo IDs for filtering requests, equivalent to passing the same Locality Qloo ID(s) into <code>filter.location</code> .</div> <div><ul style="list-style-type: none"><li>For <b>GET requests</b>: Provide a single locality query as a string.</li><li>For <b>POST requests</b>:<ul style="list-style-type: none"><li>You can still send a single locality as a string.</li><li>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.</li></ul></li></ul></div> <div>Locality queries are fuzzy-matched and case-insensitive. Examples include <code>New York City</code> , <code>Garden City</code> , <code>New York</code> , <code>Los Angeles</code> , <code>Lower East Side</code> , and AKAs like <code>The Big Apple</code> . When a single locality is supplied, the response JSON includes <code>query.locality.signal</code> 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 <code>filter.location.radius=0</code> . If no localities are found, the API returns a 400 error.</div>	

STRING >

ARRAY >

**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.

STRING >

ARRAY >

**filter.location.geohash** string

dr5rsjk4sr2w

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`.

**filter.exclude.location.geohash** string

dr5rs

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`.

**filter.location.radius** integer

10000

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`.

<p><b>filter.parents.types</b> array of strings</p> <p>Filter by a comma-separated list of parental entity types ( <code>urn:audience:communities</code> ). Each type must match exactly.</p> <div><div>ADD STRING</div><div>+</div></div>	
<p><b>filter.popularity.max</b> number 0 to 1</p> <p>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).</p>	<div>0.98</div>
<p><b>filter.popularity.min</b> number 0 to 1</p> <p>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).</p>	<div>0.5</div>
<p><b>filter.price_level.max</b> integer 1 to 4</p> <p>Filter by the maximum price level a Point of Interest can have (1 2 3 4, similar to dollar signs).</p>	<div>3</div>
<p><b>filter.price_level.min</b> integer 1 to 4</p> <p>Filter by the minimum price level a Point of Interest can have (1 2 3 4, similar to dollar signs).</p>	<div>1</div>
<p><b>filter.price_range.from</b> integer 0 to 1000000</p> <p>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.</p>	<div>200</div>
<p><b>filter.price_range.to</b> integer 0 to 1000000</p> <p>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.</p>	<div>200</div>
<p><b>filter.price.max</b> float</p> <p>maximum price</p>	<div>200</div>
<p><b>filter.price.min</b> float</p> <p>minimum price</p>	<div>1</div>
<p><b>filter.properties.business_rating.max</b> float</p> <p>Filter by the highest desired business rating.</p>	<div>3</div>
<p><b>filter.properties.business_rating.min</b> float</p> <p>Filter by the lowest desired business rating.</p>	<div>3</div>
<p><b>filter.publication_year.max</b> number</p>	<div>2021</div>

Filter by the latest desired year of initial publication for the work.	
<b>filter.publication_year.min</b> number	<input type="text" value="1985"/>
Filter by the earliest desired year of initial publication for the work.	
<b>filter.rating.max</b> number 0 to 5	<input type="text" value="4.5"/>
Filter by the maximum Qloo rating a Point of Interest must have (float, between 0 and 5).	
<b>filter.rating.min</b> number 0 to 5	<input type="text" value="3.5"/>
Filter by the minimum Qloo rating a Point of Interest must have (float, between 0 and 5).	
<b>filter.references_brand</b> array of strings	
Filter by a comma-separated list of brand entity IDs. Use this to narrow down place recommendations to specific brands. For example, to include only Walmart stores, pass the Walmart brand ID. Each ID must match exactly.	
<div>ADD STRING</div> <div>+</div>	
<b>filter.release_country</b> array of strings	
Filter by a list of countries where a movie or TV show was originally released.	
<div>ADD STRING</div> <div>+</div>	
<b>operator.filter.release_country</b> string	<input type="text" value="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".	
<b>filter.release_date.max</b> date	<input type="text" value="2024-01-01"/>
Filter by the latest desired release date.	
<b>filter.release_date.min</b> date	<input type="text" value="2021-01-01"/>
Filter by the earliest desired release date.	
<b>filter.release_year.max</b> integer	<input type="text" value="2024"/>
Filter by the latest desired release year.	
<b>filter.release_year.min</b> integer	<input type="text" value="1996"/>
Filter by the earliest desired release year.	
<b>filter.results.entities</b> string	<input type="text" value="636E1B95-6232-43"/>
Filter by a comma-separated list of entity IDs. Often used to assess the affinity of an entity towards input.	
<b>filter.results.entities.query</b>	



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.

STRING >

ARRAY >

**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  $\geq 1$

1

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

**signal.demographics.age** string

36\_to\_55

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.

NUMBER	>
STRING	>

**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.

NUMBER	>
STRING	>

**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.

ADD STRING +

**signal.demographics.gender** string

male

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.

NUMBER	>
STRING	>

**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.

- NUMBER

>
- STRING

>

**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.

- ARRAY

>
- ARRAY

>

**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.

- NUMBER

>
- STRING

>


<div><b>signal.location</b> string</div> <div>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 filter.location.radius, signal.location.radius is ignored if a Qloo ID or WKT POLYGON is passed.</div>	POINT(-73.99823 40.722668)
<div><b>signal.location.radius</b> integer</div> <div>The optional radius (in meters), used when providing a WKT POINT. We generally recommend avoiding this parameter, as it overrides dynamic density discovery.</div>	5000
<div><b>signal.location.query</b> string</div> <div>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.</div>	New York City
<div><b>signal.location.weight</b></div> <div>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.</div> <div><div>NUMBER</div><div>STRING</div></div>	
<div><b>sort_by</b> string</div> <div>This parameter modifies the results sorting algorithm (affinity distance). The distance option can only be used when `filter.location` is supplied.</div>	affinity
<div><b>take</b> integer ≥ 1</div> <div>The number of results to return.</div>	10


RESPONSES


<div><div>200</div><div>Successful Operation</div></div>	>
<div><div>400</div><div>Bad Request</div></div>	
<div><div>500</div></div>	


Internal Server Error


LANGUAGE


  
Shell

  
Node

  
Ruby


  
PHP

  
Python



CREDENTIALS

HEADER ⓘ

Header	X-Api-Key	
--------	-----------	---

REQUEST

```
$ python -m pip install requests

import requests          1
                           2
url = "https://staging    3 qloo.com/v2/insights?filter.type=urn%3Aentity%3Aartist"
                           4
headers = {"accept": "    5 cation/json"}
                           6
response = requests.ge    7 , headers=headers)
                           8
print(response.text)     9
```



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

Did this page help you?  **Yes**  **No**

