

Recommendations



Recommendations require the Frosmo Recommendations add-on for the Frosmo Control Panel. To get the add-on, if you do not already have it, contact your Frosmo representative or sales@frosmo.com.

This guide shows you how to create and manage recommendations using the Graniitti API. This guide also describes the recommendation resource data model.

- [Introduction to the recommendation resource](#)
- [Working with recommendations](#)
 - [Retrieving recommendations](#)
 - [Creating recommendations](#)
 - [Updating recommendations](#)
 - [Deleting recommendations](#)
- [Recommendation resource](#)
 - [Recommendation object](#)
 - [Options object](#)
 - [Model object](#)
 - [Data object](#)
 - [Data point set](#)
 - [Data point](#)
 - [Example: Recommendation object](#)

For instructions on how to develop recommendations with the Frosmo Platform, see the [Frosmo Recommendations guide](#).

Introduction to the recommendation resource

The **recommendation** resource of the Graniitti API combines both a **recommendation configuration** and the **recommendation data** generated for that configuration:

- When you retrieve a recommendation, the API returns both the configuration and the data.
- When you create or update a recommendation, the API only creates or updates the configuration. The Frosmo Platform then separately generates the data and stores it for the recommendation.
- When you delete a recommendation, the API deletes both the configuration and the data.

For actual recommendations displayed on a site, you retrieve the recommendation data with the [Recommendations API](#).



You can also create, update, and delete recommendations in the Frosmo Control Panel. For more information, see [Creating recommendations](#).

Working with recommendations

You can use the Graniitti API to:

- [Retrieve all recommendations or a single recommendation for a site.](#)
- [Create a new recommendation for a site.](#)
- [Update an existing recommendation for a site.](#)
- [Delete a recommendation for a site.](#)

The API provides the `/sites/<site_id>/recommendations` endpoint for working with recommendations. For complete information about the endpoint, see the [Graniitti API reference](#).

Retrieving recommendations



Retrieving recommendations can take a long time if the recommendations contain huge amounts of recommendation data.

To retrieve all recommendations for a site, send a GET request to `/sites/<site_id>/recommendations`.

```
curl -X GET \  
--header 'Authorization: Bearer <access_token>' \  
'https://<graniitti_api_domain>/v0/sites/<site_id>/recommendations'
```

To retrieve a single recommendation for a site, send a GET request to `/sites/<site_id>/recommendations/<recommendation_id>`, where `<recommendation_id>` is the ID of the recommendation.

```
curl -X GET \  
--header 'Authorization: Bearer <access_token>' \  
'https://<graniitti_api_domain>/v0/sites/<site_id>/recommendations/<recommendation_id>'
```

If the request is successful, the API returns a JSON array of objects for GET /sites/<site_id>/recommendations and a JSON object for GET /sites/<site_id>/recommendations/<recommendation_id>. An object contains the configuration and data for a single recommendation, that is, the full data of a [recommendation resource](#).

Creating recommendations

To create a new recommendation for a site, send a POST request to /sites/<site_id>/recommendations. Include the new recommendation configuration as a JSON object in the request body.

```
curl -X POST \  
--header 'Authorization: Bearer <access_token>' \  
--header 'Content-Type: application/json' \  
-d '{<recommendation_configuration>}' \  
'https://<graniitti_api_domain>/v0/sites/<site_id>/recommendations'
```

If the request is successful, the Frosmo Platform queues the recommendation for processing.

To track the generation progress, either [retrieve the recommendation](#) (optionally with the `fields=status` query parameter to minimize the response size) and check the `status` property in the returned JSON object, or check the recommendation status in the Frosmo Control Panel. For more information about the status, see [Creating recommendations](#).

Here's an example of a recommendation configuration to pass in the POST request:

```
{  
  "name": "Most viewed products",  
  "description": "Top 10 most viewed products (24 h)",  
  "type": "most_viewed",  
  "status": "new",  
  "frequency": "1h",  
  "hours": 24,  
  "limit": 10,  
  "map_ids": "product",  
  "params": [  
    "type"  
  ],  
  "options": {  
    "product_attrs": [  
      "type"  
    ]  
  }  
}
```

Updating recommendations

To update an existing recommendation for a site, send a PUT request to /sites/<site_id>/recommendations/<recommendation_id>, where <recommendation_id> is the ID of the recommendation. Include the updated recommendation configuration as a JSON object in the request body.

```
curl -X PUT \  
--header 'Authorization: Bearer <access_token>' \  
--header 'Content-Type: application/json' \  
-d '{<recommendation_configuration>}' \  
'https://<graniitti_api_domain>/v0/sites/<site_id>/recommendations/<recommendation_id>'
```

If the request is successful, the Frosmo Platform queues the recommendation for processing.

To track the generation progress, either [retrieve the recommendation](#) (optionally with the `fields=status` query parameter to minimize the response size) and check the `status` property in the returned JSON object, or check the recommendation status in the Frosmo Control Panel. For more information about the status, see [Creating recommendations](#).

Here's an example of a recommendation configuration to pass in the PUT request:

```

{
  "name": "Most viewed products",
  "description": "Top 5 most viewed products (24 h)",
  "type": "most_viewed",
  "status": "new",
  "frequency": "lh",
  "hours": 24,
  "limit": 5,
  "map_ids": "product",
  "params": [
    "type"
  ],
  "options": {
    "product_attrs": [
      "type"
    ]
  }
}

```

Deleting recommendations



Deleting a recommendation deletes both the recommendation configuration and all associated recommendation data from the Frosmo backend. Any modification that uses the data will no longer have access to it.

To delete a recommendation for a site, send a DELETE request to `/sites/<site_id>/recommendations/<recommendation_id>`, where `<recommendation_id>` is the ID of the recommendation.

```

curl -X DELETE \
--header 'Authorization: Bearer <access_token>' \
'https://<graniitti_api_domain>/v0/sites/<site_id>/recommendations/<recommendation_id>'

```

If the request is successful, the Frosmo Platform deletes the recommendation.

Recommendation resource

This section describes the JSON object data model for a recommendation resource.

Recommendation object

The following table describes the main recommendation object.

Table: Recommendation object properties

Property	Description	Type	Role	Example
id	Recommendation ID.  The Frosmo Platform sets this property. You cannot change it.	Number	Required	<code>"id": 1212</code>
site_id	Site ID.  The Frosmo Platform sets this property. You cannot change it.	Number	Required	<code>"site_id": 2080</code>
name	Recommendation name. For users to easily distinguish between recommendations in the Frosmo Control Panel, it is recommended that you define a name that's unique within a site. The Frosmo Platform does not check the name for uniqueness.	String	Required	<code>"name": "Most viewed products"</code>

description	Description of what the recommendation is about.	String	Optional	<pre>"description" : "Top 10 most viewed products (24 h)"</pre>								
type	<p>Recommendation type.</p> <p>The type defines the algorithm or filter that the Frosmo Platform uses to generate the recommendation data.</p> <p>The possible values for the built-in types are:</p> <ul style="list-style-type: none"> • bundle_converted • bundle_viewed • bundle_viewed_converted • most_converted • most_viewed <p>The default value is most_viewed.</p> <p>For more information about the types, see Introduction to recommendations.</p>	String	Required	<pre>"type" : "most_viewed"</pre>								
status	<p>Status of the recommendation data generation process.</p> <p>The possible values are (in process order):</p> <ul style="list-style-type: none"> • new: Recommendation generation has yet to be scheduled. • scheduled: Recommendation generation has been scheduled. Depending on how many other recommendations are queued for generation, it might take anywhere from a fraction of a second to several minutes before the generation starts. • started: Recommendation is being generated. • success: Recommendation was successfully generated. • failure: Recommendation generation failed. <p>The default value is new.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  The Frosmo Platform sets this property. You cannot change it. </div>	String	Required	<pre>"status" : "success"</pre>								
duration	<p>Total number of seconds it took to generate the recommendation data.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  The Frosmo Platform sets this property. You cannot change it. </div>	Number	Optional	<pre>"duration" : 2</pre>								
frequency	<p>How often the recommendation data should be regenerated.</p> <p>The possible values are:</p> <ul style="list-style-type: none"> • 15m: 15 minutes • 1h: 1 hour • 1d: 1 day (24 hours) <p>The default value is 15m.</p> <p>If you enter a value not listed above and save the configuration, the platform ignores the value and retains the existing one (default value, unless previously saved to another valid value).</p> <p>The generation time is calculated based on the last time the data was generated (either automatically or manually). For example, if the frequency is 15 minutes, and the data is generated at 10:15, the data is next generated at 10:30. If the frequency was 1 day, the data would be next generated at 10:30 the following day.</p> <p>It is recommended that you set <code>frequency</code> based on <code>hours</code> as follows:</p> <table border="1" style="margin-top: 10px;"> <thead> <tr> <th>Hours</th> <th>Frequency</th> </tr> </thead> <tbody> <tr> <td>Less than or equal to 24 (1 day)</td> <td>15m</td> </tr> <tr> <td>25 – 168 (1 – 7 days)</td> <td>1h</td> </tr> <tr> <td>Greater than 168 (7 days)</td> <td>1d</td> </tr> </tbody> </table> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  If generating the recommendation data takes longer than the defined frequency, increase the frequency, as there's no point in having a frequency that's shorter than the actual generation time. You can check the time after the recommendation data has been generated at least once. </div>	Hours	Frequency	Less than or equal to 24 (1 day)	15m	25 – 168 (1 – 7 days)	1h	Greater than 168 (7 days)	1d	String	Required	<pre>"frequency" : "15m"</pre>
Hours	Frequency											
Less than or equal to 24 (1 day)	15m											
25 – 168 (1 – 7 days)	1h											
Greater than 168 (7 days)	1d											

hours	<p>Number of past hours of source data on which to base the recommendation.</p> <p>For example, if you set this to 10, the recommendation will always be generated based on the last 10 hours of data collected from the site.</p> <p>The maximum value is 336 (14 days).</p> <p>The default value is 1.</p>	Number	Required	<pre>"hours": 24</pre>
limit	<p>Maximum number of items, such as products, to include in the recommendation.</p> <p>The default value is 100.</p> <p>The value must be between 1 and 120.</p>	Number	Required	<pre>"limit": 10</pre>
map_ids	<p>Logic for mapping the IDs of the recommended items to source data in the Frosmo back end. This determines what data the recommendation actually returns for each item.</p> <p>The possible values are:</p> <ul style="list-style-type: none"> product: Map each item ID to a product ID. Return the full product data for each ID. Empty string: Do not map the item IDs to anything. Return the raw recommendation data (product ID, recommendation weight, and optional custom data) for each item. For more information, see Data point. <p>The default value is product.</p>	String	Optional	<p>Map each item ID to a product ID:</p> <pre>"map_ids": "product"</pre> <p>Do not map the item IDs:</p> <pre>"map_ids": ""</pre>
params	<p>Names of the request parameters for filtering the recommendation data. If you also want to filter the recommendation data by segment, define a parameter for the segment ID as the last string in the array.</p> <p>The parameters must be included in each Recommendations API request that retrieves this recommendation data.</p> <p>You can filter recommendation data by any field or segment. How exactly you want to filter the data, and therefore what parameters to define for the request, is up to you. You can also freely name the parameters.</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p> If you're using Recommendations API v2, you must define at least one parameter, as the API expects a request to include one or more parameters for filtering the recommendation data.</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p> If you're using Recommendations API v1, do not use "name" as a parameter name, as the name property is reserved for defining the recommendation name in the API request.</p> </div> <p>The parameters together define a subset of recommended items whose data to return. In the request, you must define a value for each parameter. If you do not want to filter the returned data by a specific parameter, set its value to an empty string.</p> <p>For example, if you're creating a product recommendation and want to return the recommendation data by product type, define a parameter name for type, such as type.</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p> If the recommendation type is <code>bundle_converted</code>, <code>bundle_viewed</code>, or <code>bundle_viewed_converted</code>, define only the parameter for providing the ID of the product for which to generate the recommendation. Do not define any other parameters.</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p> For product recommendations, the <code>options.product_attrs</code> array defines how the parameters map to fields in the product data from which the recommendation is generated.</p> </div> <p>The default value is ["type"].</p>	Array of strings	Optional	<p>Filter items by type:</p> <pre>"params": ["type"]</pre> <p>Filter items by name, type, and area:</p> <pre>"params": ["name", "type", "area"]</pre> <p>Filter items by type and visitor segment:</p> <pre>"params": ["type", "segment"]</pre>
options	<p>Configuration options specific to recommendation type.</p> <p>The default value is {}.</p> <p>For more information, see Options object.</p>	Object	Required	For examples, see Options object .
model	<p>Latest generated recommendation data and configuration details related to generating the data.</p> <p>For more information, see Model object.</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p> The value of this property describes an object, but the Graniitti API returns the object in stringified form.</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p> The Frosmo Platform sets this property. You cannot change it.</p> </div>	String (stringified object)	Optional	For examples, see Model object .

error	<p>Error trace of what happened when the recommendation data generation failed.</p> <p>If the data generation was successful, the value is an empty string.</p>	String	Optional	<pre>"error": ""</pre>
created_at	<p>Date and time when the recommendation was created.</p> <p>For more information about the data and time format, see Data formats.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  The Frosmo Platform sets this property. You cannot change it. </div>	String	Required	<pre>"created_at": "2018-11- 08T10:37: 25+00:00"</pre>
updated_at	<p>Date and time when the recommendation was last updated.</p> <p>For more information about the data and time format, see Data formats.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  The Frosmo Platform sets this property. You cannot change it. </div>	String	Required	<pre>"updated_at": "2018-11- 20T12:07: 08+00:00"</pre>

Options object

The following table describes the `options` child object of the main recommendation object. `options` contains the configuration options specific to recommendation type. If you have no options to define, set the value of `options` to an empty object (`{}`).

Table: Options object properties

Property	Description	Type	Role	Example
basket_size	<div style="border: 1px solid #ffc107; padding: 5px; margin-bottom: 10px;">  This property only applies when the recommendation type is <code>bundle_converted</code>, <code>bundle_viewed</code>, or <code>bundle_viewed_converted</code>. </div> <p>Maximum number of viewed or purchased products to consider per visitor, starting with the first view or purchase within the time frame over which the recommendation is generated.</p> <p>For example, if you set this to 10, and a visitor purchases 20 products within the time frame over which the recommendation is generated, the generation will only consider and potentially return the first 10 purchases when determining which products were purchased together.</p> <p>The default value is 70.</p>	Number	Optional	<pre>"basket_size": 20</pre>
category_delimiter	<div style="border: 1px solid #ffc107; padding: 5px; margin-bottom: 10px;">  This property only applies when the recommendation type is <code>most_converted</code> or <code>most_viewed</code>. </div> <p>Delimiter string by which to split the product type. If the string is empty, the product type is not split.</p> <p>The default value is an empty string.</p>	String	Optional	<pre>"category_delimiter": ""</pre>
product_attrs	<div style="border: 1px solid #ffc107; padding: 5px; margin-bottom: 10px;">  This property only applies when the recommendation type is <code>most_converted</code> or <code>most_viewed</code>. </div> <p>Product data fields for filtering the generated recommendation data.</p> <p>Define the field names that map to the request parameters defined in the <code>params</code> array. The mapping is done in order, so make sure you provide the field names in the same order as the matching parameter names in the <code>params</code> array.</p> <p>To find out the available product data fields, use the Product API.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  The field names are based on the product object structure of a Product API response. Use the appropriate JavaScript accessor notation to reference nested object properties. <ul style="list-style-type: none"> To reference a direct property of the product object, simply use its name (<code>propertyName</code>). To reference a property of the <code>attributes</code> object, itself a direct property of the product object, use either dot notation (<code>attributes.propertyName</code>) or bracket notation (<code>attributes['propertyName']</code>). To reference within an array or object property of the <code>attributes</code> object, use dot or bracket notation (<code>attributes.propertyName.nestedName</code>) or index notation (<code>attributes.propertyName[i]</code>) as appropriate, depending on the nesting structure leading up to the target element or property. </div> <p>The <code>product_attrs</code> array tells the Frosmo Platform how to map the parameters in the Recommendations API request to product data fields.</p> <p>For example, if you've defined a request parameter for product type, define <code>type</code> as the product data field name to which the parameter maps.</p> <p>If you define no request parameters, meaning you do not want to filter the recommendation data, leave this array empty.</p> <p>The default value is <code>[]</code>.</p>	Array of strings	Optional	<p>Map a lone request parameter to product type:</p> <pre>"product_attrs": ["type"]</pre> <p>Map the first request parameter to product name, the second request parameter to product type, and the third request parameter to product area:</p> <pre>"product_attrs": ["name", "type", "attributes.area"]</pre> <p>No request parameters to map to product data:</p> <pre>"product_attrs": []</pre>

segments	<p> This property only applies when the recommendation type is <code>most_converted</code> or <code>most_viewed</code>.</p> <p>IDs of the segments by which you want to allow the returned recommendation data to be filtered. You can use one of the IDs in a Recommendations API request (as the value of the <code>segment</code> parameter) to retrieve the recommendations only for that segment. If you do not define the ID of a given segment, you cannot filter the results by that segment.</p> <p>If you define one or more segment IDs, the <code>params</code> array must include one parameter for the segment ID as its last item.</p> <p>For example, if you define <code>segment</code> as the last item in the <code>params</code> array, and you define the segment IDs 123 and 456 in this property, you can use the parameter <code>"segment": "123"</code> or <code>"segment": "456"</code> in the Recommendations API request to filter the returned data by the corresponding segment.</p> <p>The segment IDs are numerical strings, such as <code>"123"</code>, without the <code>sgmt_</code> prefix, which you see as part of segment IDs in the Frosmo Control Panel.</p> <p>The default value is <code>[]</code>.</p>	Array of strings	Optional	<p>Allow filtering by two segments:</p> <pre>"segments": ["123", "456"]</pre> <p>Do not allow filtering by segment:</p> <pre>"segments": []</pre>
----------	---	------------------	----------	---

Model object

The following table describes the `model` child object of the main recommendation object. `model` contains the latest generated recommendation data and configuration details related to generating the data.

 The Graniitti API returns this object in stringified form.

 The Frosmo Platform sets and updates this object. You cannot directly change it.

Table: Model object properties

Property	Description	Type	Role	Example
<code>map_ids</code>	Copy of the <code>map_ids</code> property of the main recommendation object .	String	Required	For examples, see Recommendation object .
<code>params</code>	Copy of the <code>params</code> property of the main recommendation object .	Array of strings	Required	For examples, see Recommendation object .
<code>version</code>	Version of the data schema used by the Frosmo Platform to generate and structure the recommendation data. Currently, the only possible value is 0.	Number (integer)	Optional	<pre>"version": 0</pre>
<code>data</code>	Latest generated recommendation data. For more information, see Data object .	Object	Required	For examples, see Data object .

Data object

The `data` object contains the latest generated recommendation data. The data is organized as follows:

- `data` consists of one or more data point sets.
- Data point set** is an array of objects (data points) identified by a unique key. For more information, see [Data point set](#).
- Data point** is an object that represents a single recommended item. For more information, see [Data point](#).

The following example shows the high-level structure of recommendation data.

```

"data": {
  "<data point set 1 key>": [
    {<data point 1>},
    {<data point 2>},
    ...
    {<data point n>}
  ],
  "<data point set 2 key>": [
    {<data point 1>},
    {<data point 2>},
    ...
    {<data point n>}
  ],
  ...
  "<data point set n key>": [
    {<data point 1>},
    {<data point 2>},
    ...
    {<data point n>}
  ]
}

```

Data point set

A data point set (array of objects) is a prefiltered subset of all recommended items (data points). The data points in a data point set are sorted in descending order by `weight` (see [Data point](#)). The purpose of data point sets is to provide fast and ready access to specific subsets of recommendation results. Since the results are prefiltered, the platform doesn't need to separately perform filtering for each request. Instead, the platform simply returns the subset or subsets matching the request.

Each data point set is based on a unique combination of available values for the request parameters defined in the `params` property of the [main recommendation object](#). A data point set key is built by concatenating the specific parameter values for that set, in the order in which the parameters are defined in the `params` property.

For recommendation types that allow empty parameter values:

- The first data point set has an empty string as its key ("") and contains all the recommended items
- The data point sets also include combinations for cases where one or more parameters are set to empty values. The keys for these sets are built with the non-empty values.

For example, if a `most_converted` product recommendation defines the request parameters `type` and `color`, the recommendation data will contain data point sets for every combination of type and color values found among the product data stored in the Frosmo database, including data point sets where the other parameter value is empty. The following example shows the data point sets when the database contains products of type `Shoe` in colors `Black` and `White` and products of type `Suit` in colors `Black`, `Purple`, and `Red`.

```

"data": {
  "": [<data points for all recommended products, no filtering by any parameter>],
  "Shoe": [<data points for recommended shoes, no filtering by color>],
  "ShoeBlack": [<data points for recommended black shoes>],
  "ShoeWhite": [<data points for recommended white shoes>],
  "Suit": [<data points for recommended suits, no filtering by color>],
  "SuitBlack": [<data points for recommended black suits>],
  "SuitPurple": [<data points for recommended purple suits>],
  "SuitRed": [<data points for recommended red suits>],
  "Black": [<data points for recommended black products, no filtering by type>],
  "Purple": [<data points for recommended purple products, no filtering by type>],
  "Red": [<data points for recommended red products, no filtering by type>],
  "White": [<data points for recommended white products, no filtering by type>]
}

```

When a client calls the Recommendations API for the above recommendation, the platform constructs the returned data from the data points matching the parameters defined in the client's request. For example, in the request:

- If `"type": ""` and `"color": ""`, the platform uses the "" data point set.
- If `"type": ""` and `"color": "Purple"`, the platform uses the `Purple` data point set.
- If `"type": "Shoe"` and `"color": ""`, the platform uses the `Shoe` data point set.
- If `"type": "Suit"` and `"color": "Red"`, the platform uses the `SuitRed` data point set.
- If `"type": "Shoe"` and `"color": "Orange"`, the platform returns an empty data object, as there's no `ShoeOrange` data point.

The platform determines what data to return based on the value of the `map_ids` property of the [main recommendation object](#):

- **product**: For each data point in the selected data point sets, the platform returns the corresponding product data. The platform calls the [Product API](#) with the data point IDs, which map to product IDs.
- **Empty string**: The platform returns the data points in the selected data point sets as is. For more information, see [Data point](#).

Data point

The following table describes the data point object. A data point represents a single recommended item. Data points are grouped into [data point sets](#) in the generated recommendation data.

Table: Data point object

Property	Description	Type	Role	Example
id	<p>Data point ID.</p> <p>In a product recommendation, this is the ID of the product that the data point represents.</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p> If the recommendation type is <code>bundle_viewed</code> or <code>most_viewed</code>, the ID maps to a product ID in product data. In the data layer, this ID corresponds to the <code>frosmoProductId</code> property of a product object.</p> <p>If the recommendation type is <code>bundle_converted</code>, <code>bundle_viewed_converted</code>, or <code>most_converted</code>, the ID maps to either a product ID in transaction data or a conversion ID in conversion data, depending on how the Frosmo Platform tracks conversions for products. In the data layer, this ID corresponds to either the <code>id</code> property of a transaction product object of a transaction object or the <code>conversionId</code> property of a conversion object, respectively.</p> </div> <div style="border: 1px solid #ffc107; padding: 5px; margin: 5px 0;"> <p> If you track conversions for products using the data layer conversion object, the conversion ID must match a valid product ID. Otherwise, the data point ID does not map to a tracked product, and the Frosmo Platform cannot return product data for that ID.</p> </div>	String	Required	<pre>"id": "123"</pre>
weight	<p>Measure of recommendation primacy or strength. The higher the weight, the more popular or recommendable the item. The value range and what the value means depends on the recommendation algorithm. For example, the value could be a count (positive integer) or a percentage (decimal value between 0 and 1).</p> <p>Data points are organized in descending order by weight.</p> <p>In a product recommendation, this is the total number of views or purchases for the product that the data point represents.</p>	Number	Required	<pre>"weight": 1056</pre>

value	<p>Custom data for the data point in the JSON format.</p> <p>This property is not used for product recommendations, since the relevant product data is retrieved through the Product API.</p>	Object	Optional	<pre>"value": { "images": ["/products/123/images/1.png", "/products/123/images/2.png", "/products/123/images/3.png", "/products/123/images/4.png", "/products/123/images/5.png"] }</pre>
-------	---	--------	----------	--

Example: Recommendation object

The following is a full example of a recommendation object returned by the Graniitti API. You can view the `model` data in unstringified form below the example.

```

{
  "id": 16,
  "site_id": 2094,
  "name": "Products purchased together",
  "description": "Top 5 products purchased together in the past 24 hours",
  "type": "bundle_converted",
  "status": "success",
  "duration": 3,
  "frequency": "15m",
  "hours": 24,
  "limit": 5,
  "params": [
    "productId"
  ],
  "map_ids": "product",
  "options": {
    "basket_size": 20
  },
  "model": "{\\"params\\":[\\\"productId\\\"],\\"map_ids\\":\\\"product\\\",\\"version\\":0,\\"data\\":{\\\"1\\":[\\\"id\\":\\\"4\\\",\\\"weight\\":2],\\\"id\\":\\\"3\\\",\\\"weight\\":2},\\\"id\\":\\\"2\\\",\\\"weight\\":1},\\\"id\\":\\\"7\\\",\\\"weight\\":1},\\\"id\\":\\\"5\\\",\\\"weight\\":1}],\\\"2\\":[\\\"id\\":\\\"4\\\",\\\"weight\\":2},\\\"id\\":\\\"3\\\",\\\"weight\\":2},\\\"id\\":\\\"7\\\",\\\"weight\\":1},\\\"id\\":\\\"5\\\",\\\"weight\\":1},\\\"id\\":\\\"1\\\",\\\"weight\\":1}],\\\"3\\":[\\\"id\\":\\\"4\\\",\\\"weight\\":3},\\\"id\\":\\\"2\\\",\\\"weight\\":2},\\\"id\\":\\\"1\\\",\\\"weight\\":2},\\\"id\\":\\\"7\\\",\\\"weight\\":1},\\\"id\\":\\\"5\\\",\\\"weight\\":1}],\\\"4\\":[\\\"id\\":\\\"3\\\",\\\"weight\\":3},\\\"id\\":\\\"2\\\",\\\"weight\\":2},\\\"id\\":\\\"1\\\",\\\"weight\\":2},\\\"id\\":\\\"7\\\",\\\"weight\\":1},\\\"id\\":\\\"5\\\",\\\"weight\\":1}],\\\"5\\":[\\\"id\\":\\\"7\\\",\\\"weight\\":2},\\\"id\\":\\\"6\\\",\\\"weight\\":2},\\\"id\\":\\\"2\\\",\\\"weight\\":1},\\\"id\\":\\\"4\\\",\\\"weight\\":1},\\\"id\\":\\\"3\\\",\\\"weight\\":1]}}",
  "error": "",
  "created_at": "2018-12-10T15:00:55+00:00",
  "updated_at": "2019-01-25T09:10:47+00:00"
}

```

Example model data unstringified and prettified as a JSON object

```

{
  "params": [
    "productId"
  ],
  "map_ids": "product",
  "version": 0,
  "data": {
    "1": [
      {
        "id": "4",
        "weight": 2
      },
      {
        "id": "3",
        "weight": 2
      },
      {
        "id": "2",
        "weight": 1
      },
      {
        "id": "7",
        "weight": 1
      },
      {
        "id": "5",
        "weight": 1
      }
    ],
    "2": [
      {
        "id": "4",
        "weight": 2
      },
    ],

```

```
{
  "id": "3",
  "weight": 2
},
{
  "id": "7",
  "weight": 1
},
{
  "id": "5",
  "weight": 1
},
{
  "id": "1",
  "weight": 1
}
],
"3": [
  {
    "id": "4",
    "weight": 3
  },
  {
    "id": "2",
    "weight": 2
  },
  {
    "id": "1",
    "weight": 2
  },
  {
    "id": "7",
    "weight": 1
  },
  {
    "id": "5",
    "weight": 1
  }
],
"4": [
  {
    "id": "3",
    "weight": 3
  },
  {
    "id": "2",
    "weight": 2
  },
  {
    "id": "1",
    "weight": 2
  },
  {
    "id": "7",
    "weight": 1
  },
  {
    "id": "5",
    "weight": 1
  }
],
"5": [
  {
    "id": "7",
    "weight": 2
  },
  {
    "id": "6",
    "weight": 2
  },
  {

```

```
    "id": "2",  
    "weight": 1  
  },  
  {  
    "id": "4",  
    "weight": 1  
  },  
  {  
    "id": "3",  
    "weight": 1  
  }  
] }  
}
```