

Recommendations API reference (configurations)

Learn how to make [Recommendations API](#) requests to retrieve [recommendation data](#) generated from [recommendation configurations](#):

- [Requests](#)
- [Responses](#)

If you want to retrieve recommendation data generated from recommendation strategies, see [Recommendations API reference \(strategies\)](#).

Requests

- [URL](#)
- [Authentication](#)
- [Methods](#)
- [Syntax](#)
 - [GET](#)
 - [POST](#)
- [Parameters](#)
 - [Recommendation configuration settings object](#)
- [Request examples](#)

URL

The URL for Recommendations API requests is `https://<platform_instance>/recoApi/v2`, where `<platform_instance>` is the domain name of your Frosmo Platform instance.

To get the URL for your Frosmo Platform instance:

1. In the Frosmo Control Panel, select **Utilities > Frosmo APIs**.
2. In the **Recommendations API** section, copy the URL.



Authentication

The Recommendations API does not require authentication.

Methods

The Recommendations API supports the following HTTP methods for recommendation configurations:

- [GET](#)
- [POST](#)

The only difference between the two is that with GET you provide the recommendation details in the request URL, while with POST you provide the details in the request body. If you're retrieving data for a recommendation configuration with a GET request, and the request URL exceeds the maximum URL length, use a POST request instead.

Syntax

GET

To retrieve recommendation data for a recommendation configuration using the GET method, make the following request:

```

GET https://<platform_instance>/recoApi/v2?method=multifetch&
origin=<site_origin>&
recos=[{
  "name": "<recommendation_configuration_name>",
  "params": {
    "<parameter_1_name>": "<parameter_1_value>",
    "<parameter_n_name>": "<parameter_n_value>"
  }
}]

```

POST

To retrieve recommendation data for a recommendation configuration using the POST method, make the following request:

```

POST https://<platform_instance>/recoApi/v2?method=multifetch&
origin=<site_origin>

# REQUEST BODY

[ {
  "name": "<recommendation_configuration_name>",
  "params": {
    "<parameter_1_name>": "<parameter_1_value>",
    "<parameter_n_name>": "<parameter_n_value>"
  }
} ]

```

Parameters

Table: Query parameters for Recommendations API requests (configurations)

Parameter	Description	Type	Role	Example
method	Request type. Use the value multifetch.	String	Required	method=multifetch
origin	Site origin. To find out your site's origin, see Getting your site origin .	String	Required	origin=shop_company_com

recos	<p>Information about what recommendation data to retrieve and, if appropriate, how to filter the returned data with parameters defined in the recommendation configuration.</p> <p>You can retrieve the data for one or more recommendation configurations. You specify each configuration in its own recommendation configuration settings object using the JSON format.</p> <div data-bbox="256 247 1037 331" style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">  The array must not be empty (<code>[]</code>). </div> <div data-bbox="256 352 1037 457" style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">  If you want to guarantee a fast response, limit the number of recommendations to four. You can retrieve the data for five or more recommendations in the same request, if you want, but the platform may then take longer to respond. </div> <div data-bbox="256 478 1037 573" style="border: 1px solid #ccc; padding: 5px;">  The <code>recos</code> query parameter is only required for GET requests. POST requests pass the array of recommendation configuration settings objects in the request body. The rules for the objects nonetheless apply to both GET and POST requests. </div>	Array of recommendation configuration settings objects	Required (GET)	<p>Retrieve the data for one recommendation configuration:</p> <pre data-bbox="1312 212 1484 541">recos=[{ "name": "Most viewed products", "params": { "type": "Books" } }]</pre> <p>Retrieve the data for two recommendation configurations:</p> <pre data-bbox="1312 646 1484 1224">recos=[{ "name": "Most viewed products", "params": { "type": "Books" } },{ "name": "Products viewed together", "params": { "id": "123" } }]</pre>
variants	<p>Define whether to also return the data for product variants.</p> <p>If a product has one or more variants, the API returns the full product data for each variant in the <code>variants</code> array property of the recommended item object.</p> <p>The possible values are:</p> <ul data-bbox="277 1394 597 1436" style="list-style-type: none"> <code>false</code>: Do not return product variant data. <code>true</code>: Return product variant data. <p>The default value is <code>false</code>.</p>	Boolean	Optional	<pre data-bbox="1312 1276 1484 1329">variants=true</pre>
debug	<p>Define whether to return debugging information about the recommendation data.</p> <div data-bbox="256 1528 1037 1623" style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">  Use the debugging information only for development and testing purposes. Do not use the information in production, since Frosmo does not guarantee that the information structure and content will remain the same. </div> <p>The debugging information for a recommendation consists of:</p> <ul data-bbox="277 1682 1037 1738" style="list-style-type: none"> Selected details about the recommendation configuration that was used to generate the recommendation Raw recommendation data (<code>dataPoints</code>) that was used to build the final recommendation data For a product recommendation: Product API debugging information for the recommended products <p>The possible values are:</p> <ul data-bbox="277 1793 613 1835" style="list-style-type: none"> <code>false</code>: Do not return debugging information. <code>true</code>: Return debugging information. <p>The default value is <code>false</code>.</p>	Boolean	Optional	<pre data-bbox="1312 1499 1484 1551">debug=true</pre>

Recommendation configuration settings object

The following table describes the recommendation configuration settings object of the `recos` array. The object defines what recommendation data to retrieve and, if appropriate, how to filter the returned data.

Table: Recommendation configuration settings object properties

Property	Description	Type	Role	Example
name	Recommendation configuration name.	String	Required	<pre>"name" : "Most viewed products"</pre>

<p>params</p>	<p>Request parameters defined in the recommendation configuration. Use the parameters to filter which recommended items to return.</p> <p>You can only include parameters defined in the recommendation configuration:</p> <ul style="list-style-type: none"> • If you want to filter the results by a given parameter, assign the parameter an exact field or segment value (case insensitive). • If you do not want to filter the results by a given parameter, either omit the parameter or set the parameter value to an empty string (""). <p>If the recommendation type is <code>bundle_viewed</code>, <code>bundle_converted</code>, or <code>bundle_viewed_converted</code>, only include a single parameter: the target field against which the recommendation is generated, as defined in the recommendation configuration. For example, if you're recommending products purchased together with a product the visitor just added to their shopping cart, include the product ID parameter with a valid product ID as its value, since the recommendation is based on a specific product. Do not include any other parameters.</p> <p>If the recommendation type is <code>most_viewed</code> or <code>most_converted</code>, include zero or more of the parameters defined in the recommendation configuration. The order of the parameters does not matter in the request. In addition, if you're filtering by segment, you can only define one segment per request.</p> <p>If you do not want to use any parameters to filter the recommendation results, omit this property from the request.</p>	<p>Object</p>	<p>Optional</p>	<p>For a bundle recommendation whose configuration defines the <code>id</code> parameter:</p> <pre>"params": { "id": "123" }</pre> <p>For a most recommendation whose configuration defines the <code>type</code> parameter:</p> <pre>"params": { "type": "Books" }</pre> <p>For a most recommendation whose configuration defines the <code>type</code> and <code>language</code> parameters:</p> <pre># Filter by type and language. "params": { "type": "Books", "language": "English" }</pre> <pre># Filter by type only. "params": { "type": "Books", "language": "" }</pre> <pre># Filter by neither, return all results. "params": { "type": "", "language": "" }</pre> <pre># Omitting "params" from the request yields the same result as the empty values above.</pre>
---------------	---	---------------	-----------------	--

<pre>params. <parameter _name></pre>	<p>Request parameter defined in the recommendation configuration. Use the parameter to filter which recommended items to return.</p> <p>You can define either a single string or an array of strings as the parameter value. In the latter case, the recommendation results are filtered separately by each string value.</p> <p>If the parameter corresponds to a source data field that has a Boolean or number value, such as product price, define the parameter value nonetheless as a string. For example, define the Boolean <code>true</code> as the string <code>"true"</code>, and define the number <code>1</code> as the string <code>"1"</code>.</p> <p>If you do not want to filter the results by the parameter, set the parameter value to an empty string (<code>" "</code>).</p>	<pre>String Array of strings</pre>	<p>Required</p> <p>Filter for Books English:</p> <pre>"params": { "type": "Books", "language": "English" }</pre> <p>Filter for Books English and Books Finnish:</p> <pre>"params": { "type": "Books", "language": ["English", "Finnish"] }</pre> <p>Filter for Books English, Books Finnish, Magazines English, and Magazines Finnish:</p> <pre>"params": { "type": ["Books", "Magazines"], "language": ["English", "Finnish"] }</pre> <p>Filter by originally boolean and numeric values:</p> <pre>"params": { "isAvailable" : "true", "price": "99.99" }</pre> <p>No filtering:</p> <pre>"params": { "type": "", "language": "" }</pre>
--	--	--	---

excludeIds	 This property is only valid for product recommendations. IDs of the products to exclude from the recommendation data.	Array of strings	Optional	Exclude products 123, 456, and 789 from the recommendation data: <pre>"excludeIds": ["123", "456", "789"]</pre>
limit	Maximum number of recommended items to return. By default, the platform uses the <code>limit</code> value defined in the recommendation configuration. If you want to set a different maximum for a request, define a <code>limit</code> in the request. The minimum value is 1. The maximum value is 120.	Number	Optional	<pre>"limit": 5</pre>
mapping	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. By default, the platform uses the <code>map_ids</code> value defined in the recommendation configuration. If you want to set a different mapping for a request, define a <code>map_ids</code> in the request. The possible values are: <ul style="list-style-type: none"> <code>datapoint</code>: Do not map the item IDs to anything. Return the raw recommendation data (ID and recommendation weight) for each item. <code>product</code>: Map each item ID to a product ID. Return the full product data for each ID. 	String	Optional	<pre>"mapping": ""</pre>

Request examples

Example: Single recommendation with filtering applied (GET)

```
GET https://inpref.com/recoApi/v2?method=multifetch&
origin=shop_company_com&
recos=[{
  "name": "Most viewed products",
  "params": {
    "type": "Books"
  }
}]
```

Example: Single recommendation with filtering applied (POST)

```
POST https://inpref.com/recoApi/v2?method=multifetch&
origin=shop_company_com

# REQUEST BODY

[ {
  "name": "Most viewed products",
  "params": {
    "type": "Books"
  }
}]
```

Example: Single recommendation with filtering applied and selected results excluded (GET)

```
GET https://inpref.com/recoApi/v2?method=multifetch&
origin=shop_company_com&
recos=[{
  "name": "Most viewed products",
  "params": {
    "type": "Books"
  },
  "excludeIds": [
    "123",
    "456",
    "789"
  ]
}]
```

Example: Single recommendation with filtering applied and product variant data requested (GET)

```
GET https://inpref.com/recoApi/v2?method=multifetch&
origin=shop_company_com&
recos=[{
  "name": "Most viewed products",
  "params": {
    "type": "Books"
  }
}]&
variants=true
```

Example: Single recommendation without any filtering applied (GET)

```
GET https://inpref.com/recoApi/v2?method=multifetch&
origin=shop_company_com&
recos=[{
  "name": "Most viewed products"
}]
```

Example: Multiple recommendations with filtering applied (GET)

```
GET https://inpref.com/recoApi/v2?method=multifetch&
origin=shop_company_com&
recos=[{
  "name": "Most viewed products",
  "params": {
    "type": "Books"
  }
},{
  "name": "Most bought products",
  "params": {
    "type": "Books"
  }
}]
```

Responses

- [Success](#)
- [Error](#)

Success

On a success, the Recommendations API returns the requested recommendation data in a [JSON](#) object. The response status code is 200.

The response object contains the following root properties:

- `data`: Recommendation data as an array of arrays, where each array stores the data generated from a single recommendation configuration.
- `debug`: Object containing the debugging information. This property is included only when the `debug` parameter is set to `true` in the request.

Response structure without debugging information

```
{
  "data": [
    [<configuration 1 data>],
    [<configuration 2 data>],
    ...
    [<configuration n data>]
  ]
}
```

Response structure with debugging information

```
{
  "data": [
    [<configuration 1 data>],
    [<configuration 2 data>],
    ...
    [<configuration n data>]
  ],
  "debug": [
    <debugging information>
  ]
}
```

The data for a recommendation configuration is returned as an array of objects, where each object contains the details of a single recommended item. The items are in descending order of rank, with the most recommended item (as defined by the algorithm) ranked highest.

```

# SINGLE RECOMMENDATION CONFIGURATION

{
  "data": [
    [
      {<configuration 1 item 1 data>},
      {<configuration 1 item 2 data>},
      ...
      {<configuration 1 item n data>}
    ]
  ]
}

# MULTIPLE RECOMMENDATION CONFIGURATIONS

{
  "data": [
    [
      {<configuration 1 item 1 data>},
      {<configuration 1 item 2 data>},
      ...
      {<configuration 1 item n data>}
    ],
    [
      {<configuration 2 item 1 data>},
      {<configuration 2 item 2 data>},
      ...
      {<configuration 2 item n data>}
    ]
  ]
}

```

Depending on the `map_ids` parameter defined in the [recommendation configuration](#), each item object contains either the raw recommendation data for the item (no mapping) or the full set of data retrieved from the Frosmo back end (mapping to data). For example, for a product when the ID is mapped, the object contains the full set of product data tracked for the product on the site.

```

# PRODUCT OBJECT (MAPPED TO PRODUCT DATA)

{
  "data": [
    [
      {
        "id": "123",
        "type": "Books",
        "name": "A Developer's Guide to Everything",
        "created_at": "2018-11-08T10:45:15Z",
        "updated_at": "2018-11-08T10:51:05Z",
        "attributes": {
          "image": "https://shop.company.com/products/123/images/123.png",
          "price": "29.99",
          "url": "https://shop.company.com/products/123"
        }
      }
    ]
  ]
}

# PRODUCT OBJECT (RAW RECOMMENDATION DATA)

{
  "data": [
    [
      {
        "id": "123",
        "weight": 1056
      }
    ]
  ]
}

```

If an item is a parent product with one or more product variants, and if the request is set to also return variants, the recommended item object includes the `variants` array, which contains the full product data for each variant.

Response structure with product variant data

```
{
  "data": [
    [
      {
        "id": "1001",
        "type": "Fashion/Shoes",
        "name": "Casual Engineer Boots",
        "created_at": "2022-01-14T17:36:37+02:00",
        "updated_at": "2022-02-08T10:56:05+02:00",
        "attributes": {
          "availability": "in stock",
          "brand": "Lumberland",
          "color": "Black",
          "image": "https://shop.company.com/images/products/product-1001.png",
          "originalPrice": 350,
          "price": 290,
          "url": "https://shop.company.com/products/1001"
        },
        "variants": [
          {
            "id": "1002",
            "type": "Fashion/Shoes",
            "name": "Casual Engineer Boots",
            "created_at": "2022-01-14T17:36:37+02:00",
            "updated_at": "2022-02-08T10:56:20+02:00",
            "attributes": {
              "availability": "in stock",
              "brand": "Lumberland",
              "color": "Brown",
              "image": "https://shop.company.com/images/products/product-1002.png",
              "originalPrice": 350,
              "price": 290,
              "url": "https://shop.company.com/products/1002"
            }
          },
          {
            "id": "1003",
            "type": "Fashion/Shoes",
            "name": "Casual Engineer Boots",
            "created_at": "2022-01-14T17:36:37+02:00",
            "updated_at": "2022-02-08T10:56:34+02:00",
            "attributes": {
              "availability": "in stock",
              "brand": "Lumberland",
              "color": "White",
              "image": "https://shop.company.com/images/products/product-1003.png",
              "originalPrice": 350,
              "price": 290,
              "url": "https://shop.company.com/products/1003"
            }
          }
        ]
      }
    ]
  ]
}
```

If there's no recommendation data available for the requested configuration, the API returns an empty array.

```

# SINGLE RECOMMENDATION CONFIGURATION WITH NO DATA

{
  "data": [
    []
  ]
}

# MULTIPLE RECOMMENDATION CONFIGURATIONS, ONE WITH NO DATA

{
  "data": [
    [<configuration 1 data>],
    []
  ]
}

# MULTIPLE RECOMMENDATION CONFIGURATIONS, ALL WITH NO DATA

{
  "data": [
    [],
    []
  ]
}

```

Error

On an error, the Recommendations API returns the response status code 400 and an error message.

Table: Recommendations API error messages

Error message	Description	Solution
parameter 'method' missing	The method query parameter is missing from the request URL or its value is undefined.	Make sure that the method query parameter is properly defined.
invalid method '<parameter_value>'	The value of the method query parameter, <parameter_value>, is invalid.	Use multifetch as the value of the method query parameter.
parameter 'origin' missing	The origin query parameter is missing from the request URL or its value is undefined.	Make sure that the origin query parameter is properly defined.
invalid origin parameter '<parameter_value>'	The value of the origin query parameter, <parameter_value>, is invalid.	Use a valid site origin as the value of the origin query parameter. To find out your site's origin, see Getting your site origin .
parameter 'recos' missing	The recos query parameter is missing from the request URL or its value is undefined.	Make sure that the recos query parameter is properly defined.
parameter 'recos' contains invalid JSON: expected string for param name: '<parameter_name>', index: <parameter_index>, at recos index: <recos_index>	The <parameter_name> request parameter is an array that contains a value of the wrong type at the specified index. The API expects a string. The invalid value is located at: recos [<recos_index>].params.<parameter_name> [<parameter_index>]	Use a string as the array item value.

<pre>parameter 'recos' contains invalid JSON: expected string or array of strings for param name: '<parameter_name>', at recos index: <recos_index></pre>	<p>The value of the <parameter_name> request parameter is of the wrong type. The API expects a string or an array of strings.</p> <p>The invalid value is located at: recos [<recos_index>].params.<parameter_name></p>	<p>Use either a string or an array of strings as the value of the request parameter.</p>
<pre>parameter 'recos' contains invalid JSON: failed to parse</pre>	<p>The value of the recos query parameter is not a valid JSON array of objects.</p>	<p>Use a valid JSON array of objects as the value of the recos query parameter.</p>
<pre>parameter 'recos' contains invalid JSON: invalid model ': recommendation not found</pre> <pre>parameter 'recos' contains invalid JSON: invalid model '<name>': recommendation not found</pre>	<p>The name property is missing from a recommendation object, or the value of the property is invalid or undefined.</p>	<p>Make sure that the name property is properly defined with a valid value in each recommendation object.</p>
<pre>invalid parameters '{"<key1>":"<value1>"," <key2>":"<value2>"}' for model '<recommendation_configuration_name>': model not found</pre>	<p>There's no recommendation data available for the <recommendation_configuration_name> recommendation configuration.</p>	<p>Make sure that the data for the <recommendation_configuration_name> recommendation configuration has been successfully generated.</p>
<pre>parameter 'variants' invalid, expected either 'true', 'false', '1' or '0'</pre>	<p>The value of the variants query parameter is invalid.</p>	<p>If you want to return product variant data, set the variants query parameter to either true or 1.</p> <p>If you do not want to return product variant data, set the variants query parameter to either false or 0, or omit the parameter from the request URL.</p>