[BETA] Results Object

Last Modified on 10/25/2019 10:59 am EDT |

The GET is the only call available for the results object.

GET RESULTS - Get the aggregated results for a specified Question.

https://restapi.surveygizmo.com/v5/survey/[id]/results?options[type]=[string]

Parameters	Example	Required
Authentication Credentials	api_token=abcd12345&api_token_secret=abcd12345	True
options[type]	responsemetrics, question, rscript	True
options[question]	[id]	if options[type]=(
options[script_id]	[id]	if options[type]
options[filter_rules]	json file see example and references	False
options[filter_response_states]	array('Complete','Partial','Disqualified')	False
options[filter_include_tests]	false	False
options[filter_include_responses]	1-10	False
options[filter_exclude_responses]	11-20	False
options[filter_date_min]	'2016-12-15'	False
options[filter_date_max]	'2017-2-9'	False
options[filter_survey_links]	array(1,2,3,4)	False

Filter Rules Examples and Reference

If you wish to filter by question values or system values you will need to send a JSON file* with your filter rules. Each rule has the following parameters:

*Depending on your implementation, you many need to convert the JSON to string format and URL encode it.

input_value

This is the type of field you are referencing.*Most often, this will be the Question ID.*

If you wish to filter by something other than the a question, field types can be found by referencing the below list.

Possible Values for input_value

- Survey Question: {Question ID}
- Time Taken On Survey: 15_timetaken
- Current Date: 6_date
- Is Mobile (true/false): 21_mobile

- IP Address: 5_ip
- Country: 8_country
- Region: 9_region
- Postal Code: 19_postal
- City: 10_city
- Contact ID: 13_id
- Email Address: 13_emailaddress
- First Name: 13_firstname
- Last Name: 13_lastname
- Organization: 13_organization
- Department: 13_department
- Division: 13_division

- Team: 13_team
- Group: 13_group
- Role: 13_role
- Is Manager: 13_isManager
- Home Phone: 13_homephone
- Fax Number: 13_fax
- Business Phone: 13_workphone
- Mailing Address: 13_street
- State/Region: 13_region
- Postal Code: 13_postal
- Country: 13_country
- Title: 13_jobtitle
- Custom 1-10: 13_custom1-10
- Mailing Address 2: 13_suite
- City: 13_city

operator

This is the operator used in the logic rule.

Possible Values for operator

- is exactly equal to: 4
- is not exactly equal to: 22
- is one of the following answers: 12
- us not one of the following answers: 5
- greater than: 6
- greater than or equal to: 7
- less than: 9
- less than or equal to: 8
- Date after or equal to: 23
- Date before or equal to: 24
- is answered: 20
- is not answered: 21
- matches regex pattern: 14
- does not match regex pattern: 15
- is true: 26
- is false: 27
- contains: 10
- is always true: 25

answers_type

This is the type of data that is contained in the field specified in input_type.

Possible Values for answers_type

- Open-text Field: 2
- Input Field (Radio, Checkboxes, Drop-down Menus, etc.): 17

answers_values

These are the actual values you wish to filter by.

Possible Values for answers_values

- QID-OPTIONID - i.e. 27-10021 for question with QID 27 and answer option with OptionID 10021
- String - open-text questions and system data such as city, region, country, IP, etc.

next_operator

This is the next operator you wish to use to group your conditions.

Possible Values for next_operator (distributed=false)

- "and"
- "or"

Example json file for the filtering:

This will filter the results such that only responses with answer option 10019 from question 26 and answer option 10021 from question 27 are displayed.

```
[
  {
    "rules": [
      {
        "input_value": "26",
        "operator": "12",
        "answers_type": "17",
        "answers_values": [
          "26-10019"
        ],
        "next_operator": "and"
      },
      {
        "input_value": "27",
        "operator": "12",
        "answers_type": "17",
        "answers_values": [
          "27-10021"
        ],
        "next_operator": ""
      }
    ],
    "next_operator": "or"
  }
]
```

Example Responses

Get question results response (.json format)

At times, the `data_ready` parameter will return as `false`. This indicates that SurveyGizmo is still processing responses and as such all submitted data is not yet available. If this is the case, you will need to retry the call at a later time.

```
{
  "result_ok":true,
  "data_ready":true,
  "data_progress":100,
  "result":{
  "total":27,
  "values":[
  {
  "sku":"10001",
  "count":10,
  "percent":37,
  "option_title":"Option 1",
  "rep_val":"Option 1",
  "special":false
  },
  {
  "sku":"10002",
  "count":17,
  "percent":63,
  "option_title":"Option 2",
  "rep_val":"Option 2",
  "special":false
  }
  ],
  "stats":{
  "skipped":0
  }
  }
}
```

Get rscript results response (.json format)

At times, the `data_ready` *parameter will return as* `false` *. This indicates that SurveyGizmo is still processing responses and as such all submitted data is not yet available. If this is the case, you will need to retry the call at a later time.*

```
{
  "result_ok":true,
  "data_ready":true,
  "data_progress":100,
  "result":{
  "string_out":...your rscript output here,
  "charts":null,
  "node_out":"nothing neat",
  "node_err":""
  }
}
```

Get responsemetrics results response (.json format)

At times, the `data_ready` *parameter will return as* `false` *. This indicates that SurveyGizmo is still processing responses and as such all submitted data is not yet available. If this is the case, you will need to retry the call at a later time.*

```
{
  "result_ok":true,
  "data_ready":true,
  "data_progress":100,
  "result":{
  "values":{
  "complete":27,
  "total":27
  }
  }
}
```