Dynamically Filter Data

When your app is heavy on data usage we suggest to go for Operation based filter creating. You will not needs tens or sometimes hundreds of filters. To avoid this use this technique to filter any data in you app.

The technique uses Mobsted's internal API and is the same for API methods of:

- Objects,
- Events,
- Lists of Tables.

The key is to send a field named "query". Below is an explanation of how this field works&

Lets consider this technique for Obejcts:

We are using the method - /api/v8/object

Lets say we need to get all Objects that match a complex criteria and data for that criteria is stored in different fields of Objects.

We need Objects which have a field "Bedrooms" between 0 - 3.9 AND field "ObjectType" is Flat AND **(** field "ObjectStatus" is Order OR Free **)** , AND "Price" is 0 - 10 000 000. Note that you can sort of place math type brackets in the logic of this search when you combine results of "ObjectStatus" search with other results.

The key is to create a correct text for query. We are working on a visual tool similar to Filters for this, but for now this is how it needs to be done:

```
query -

[

    {"field":"Bedrooms","value":" '0' and '3.9' ","operation":"between","andor":"and"},

    {"field":"ObjectType","value":"Flat","operation":"=","andor":"and"},



        {"useBrackets":"(","field":"ObjectStatus","value":"Order","operation":"=","andor":"and"},

       {"field":"ObjectStatus","value":"Free","operation":"=","a ndor":"or","useBrackets":")"},



    {"field":"Price","value":" '0' and '10000000' ","operation":"between","andor":"and"}

]
```

The function useBrackets is what allows to assemble this into one.

Where:

field - column name, which is essentially a separate filter

value - a value to be searched for in the column

operation - type of operation to be searched for (=, <>, >, <, between, like),

andor - type of data connection (and, or)

useBrackets - allows joining results of several filters at once, you need to open them and close them, like in math formulas.

Note that syntax for when you do an operation BETWEEN in "value" is a single upper quote (‘)

{"field":"Bedrooms","value":" '2' and '2.9' ","operation":"between","andor":"or"}

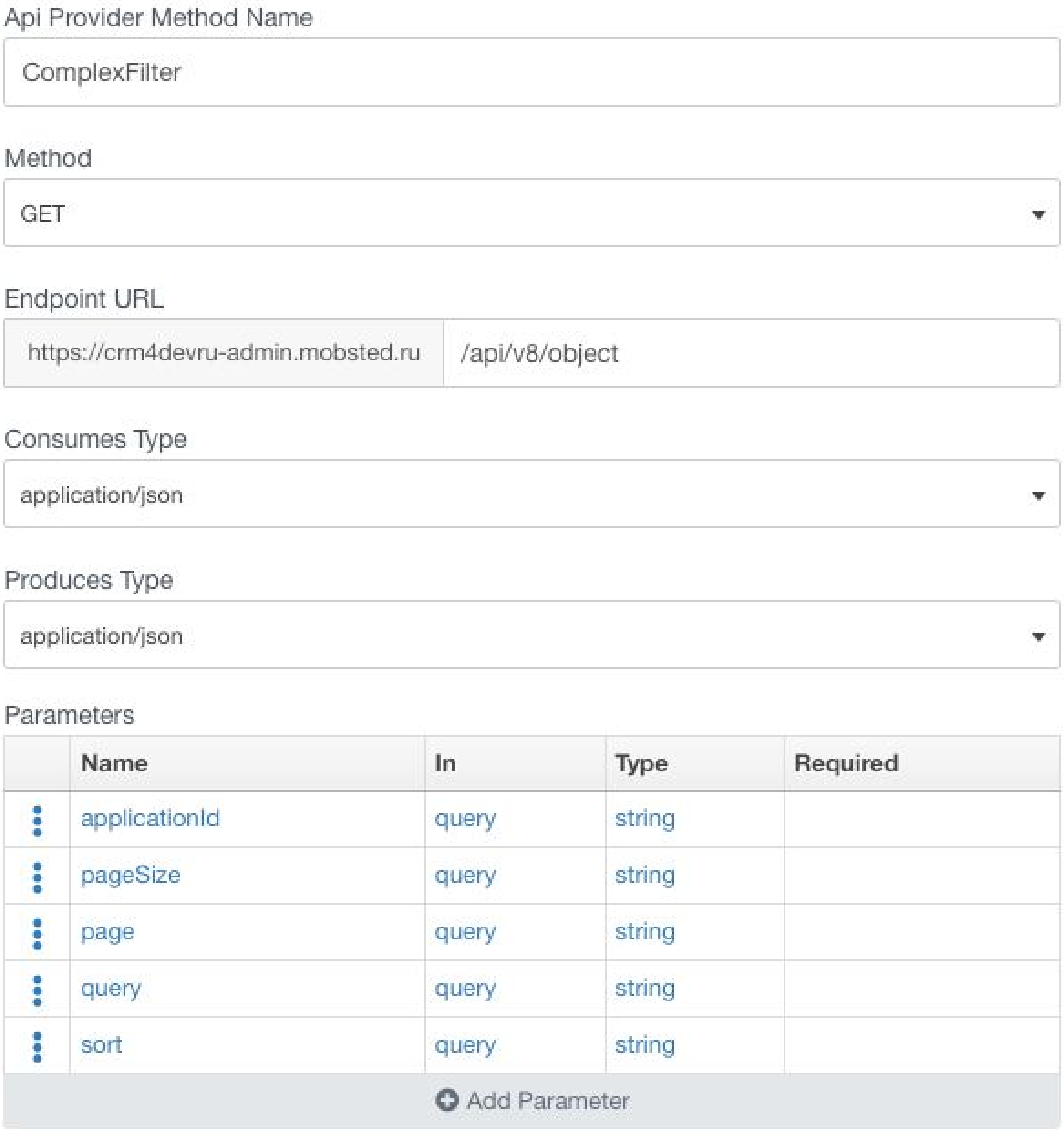
but - {"field":"Bedrooms","value":"2","operation":"=","andor":"or"} , when no between is used.

You can also sort results -

[{"field":"Price","order":"asc"}]

field -

order - type asc or desc



Api Provider Method Name

ComplexFilter

Method

GET ▾

Endpoint URL

https://crm4devru-admin.mobsted.ru /api/v8/object

Consumes Type

application/json ▾

Produces Type

application/json ▾

Parameters

	Name	In	Type	Required
⋮	applicationId	query	string	
⋮	pageSize	query	string	
⋮	page	query	string	
⋮	query	query	string	
⋮	sort	query	string	

Where request type is GET

Query Params are:

applicationId 1 - application ID - positive integer* - and ID of the app at which the filter needs to work

page 1 - page number - positive integer * - a page number to be returned

pageSize 2 - number of records on the page - positive integer * - how many records on one page to return

query - explained above

sort - explained above

If you really want to get more advance and funky you can send in data into filters dynamically as well, for example, when you need to change any parameter in the search of an apartment.

Number of Bedrooms, Size of a kitchen, total apartment size, floor, price and type of finishing is all relayed by a #variable# from the app's interface.

query:

```
[
      {"field":"Bedrooms","value":" '#Variable:f_cb_0#' and '#Variable:f_cb_0#.9' ","operation":"between","andor":"or","useBrackets":"("},
      {"field":"Bedrooms","value":" '#Variable:f_cb_1#' and '#Variable:f_cb_1#.9' ","operation":"between","andor":"or"},
      {"field":"Bedrooms","value":" '#Variable:f_cb_2#' and '#Variable:f_cb_2#.9' ","operation":"between","andor":"or"},
      {"field":"Bedrooms","value":" '#Variable:f_cb_3#' and '#Variable:f_cb_3#.9' ","operation":"between","andor":"or"},
      {"field":"Bedrooms","value":" #Variable:f_cb_4# ","operation":">=","andor":"or","useBrackets":")"},
  {"field":"Kitchen_size","value":" '#Variable:filter_square_kit_ot#' and '#Variable:filter_square_kit_do#' ","operation":"between","andor":"and"},
  {"field":"Total_Size","value":" '#Variable:filter_square_ot#' and '#Variable:filter_square_do#' ","operation":"between","andor":"and"},
  {"field":"Floor","value":" '#Variable:filter_floor_ot#' and '#Variable:filter_floor_do#' ","operation":"between","andor":"and"},
  {"field":"Floor","value":"#Variable:filter_floor_not_one#","operation":"<>","andor":"and"},
  {"field":"ID_Building","value":"#Variable:filter_building#","operation":"=","andor":"and"},
  {"field":"Type","value":"apartment","operation":"=","andor":"and"},
  {"field":"Price","value":" '#Variable:filter_price_ot#' and '#Variable:filter_price_do#' ","operation":"between","andor":"and"},
   {"field":"Finishing","value":"#Variable:filter_finishing#","operation":"=","andor":"and"}
```

Note that number of bedrooms is a joined OR type filter where variable add an INTEGER and search is set for ranges of 0 - 0.9 for studios, 1 - 1.9 for one bedrooms, and so on, but 4 and higher for larger apartments.

As always, get back to us with any questions you have at support@mobsted.com or in our online chat.