

Table of Contents

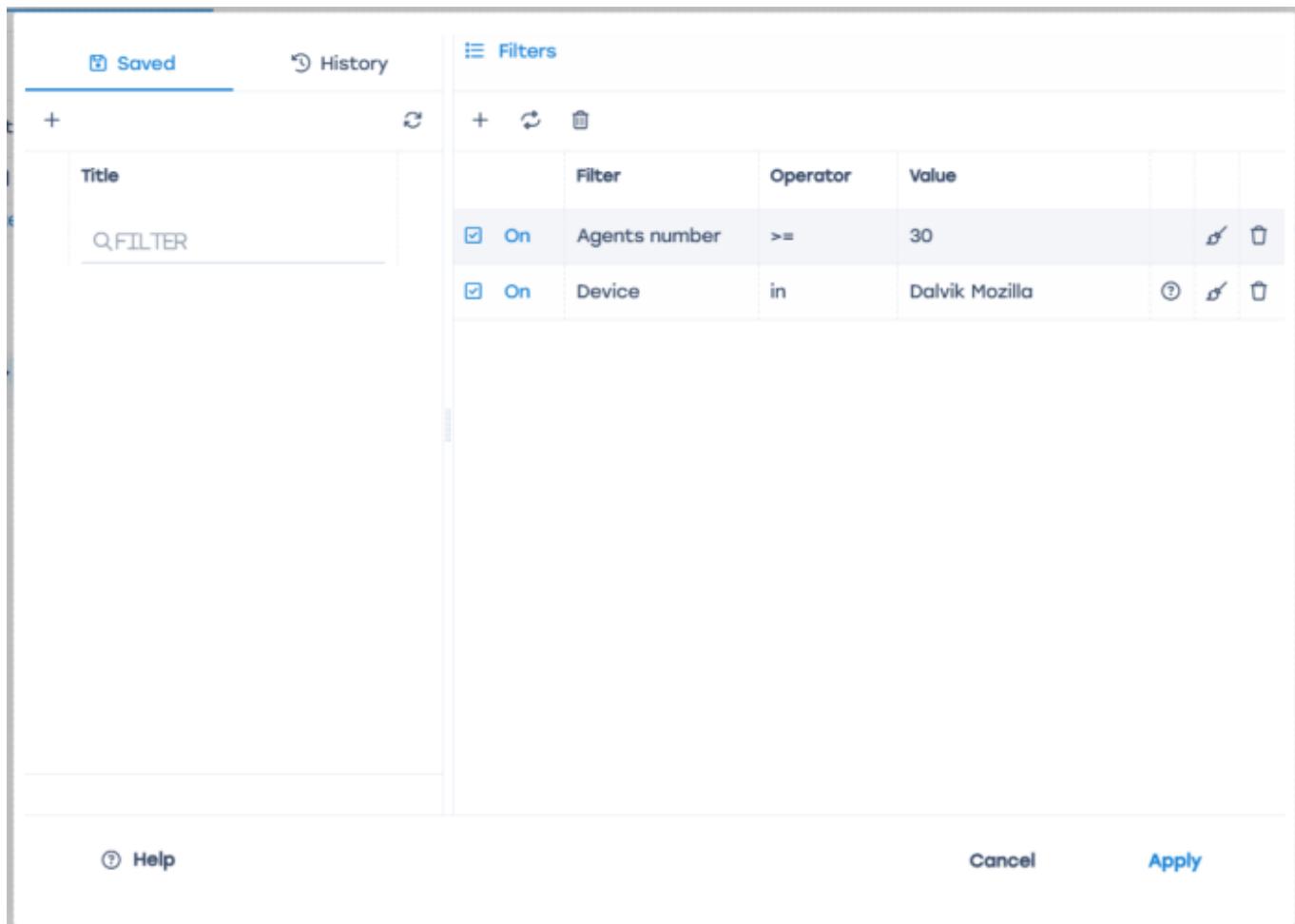
Monetization of user web request statistics	3
<i>Detecting internet service resale by residential subscribers</i>	3
<i>Smart TV detection</i>	3
<i>Video surveillance detection</i>	4
<i>Partnership with local businesses (pizza, stores, car services)</i>	5

Monetization of user web request statistics

Detecting internet service resale by residential subscribers

Calculated based on a large number of UserAgents/devices for a single subscriber. It is recommended to set more than 30 agents in the filter. Actions:

1. Go to QoE Analytics → Subscribers → Clickstream
2. Create a filter (use Shift+Enter to add multiple entries), where:
 1. Mozilla — PC identifier
 2. Dalvik — phone identifier



	Filter	Operator	Value	
<input checked="" type="checkbox"/> On	Agents number	>=	30	<input type="checkbox"/> <input type="button"/> <input type="button"/>
<input checked="" type="checkbox"/> On	Device	in	Dalvik Mozilla	<input type="checkbox"/> <input type="button"/> <input type="button"/>

There is usually no strong need to export the data into your own IS. To obtain more valid results, legal entities should be excluded from the sample — this can be done at the final stage.

Database query method:
Will be added later — create a query.

Smart TV detection

Selling OTT TV services together with the operator's core services is one of the most popular value-

added offerings. It can be sold via a digital channel or via ITM. For effective sales, it is important to understand whether a subscriber has a Smart TV and whether they already use OTT services. This is calculated based on the subscriber's UserAgent and clickstream.

Example query for Smart TV:

```
User agent match (?i)(\W|^)(smart|LG|samsung)(\W|$)
```

Example query for OTT TV:

```
host match (?i)(\W|^)(smotreshka|ivi|okko|netflix)(\W|$)
```

Database query method:

Will be added later.

Video surveillance detection

Similar to Smart TV: we look for a base for selling cloud video surveillance services. This case is a bit more complex, because there are many vendors and cloud platforms, so you need to build libraries. Vendors (cameras) can be detected via UserAgent, and services via Host.

For example, let's find subscribers using the most popular services:

<https://camera.rt.ru/>

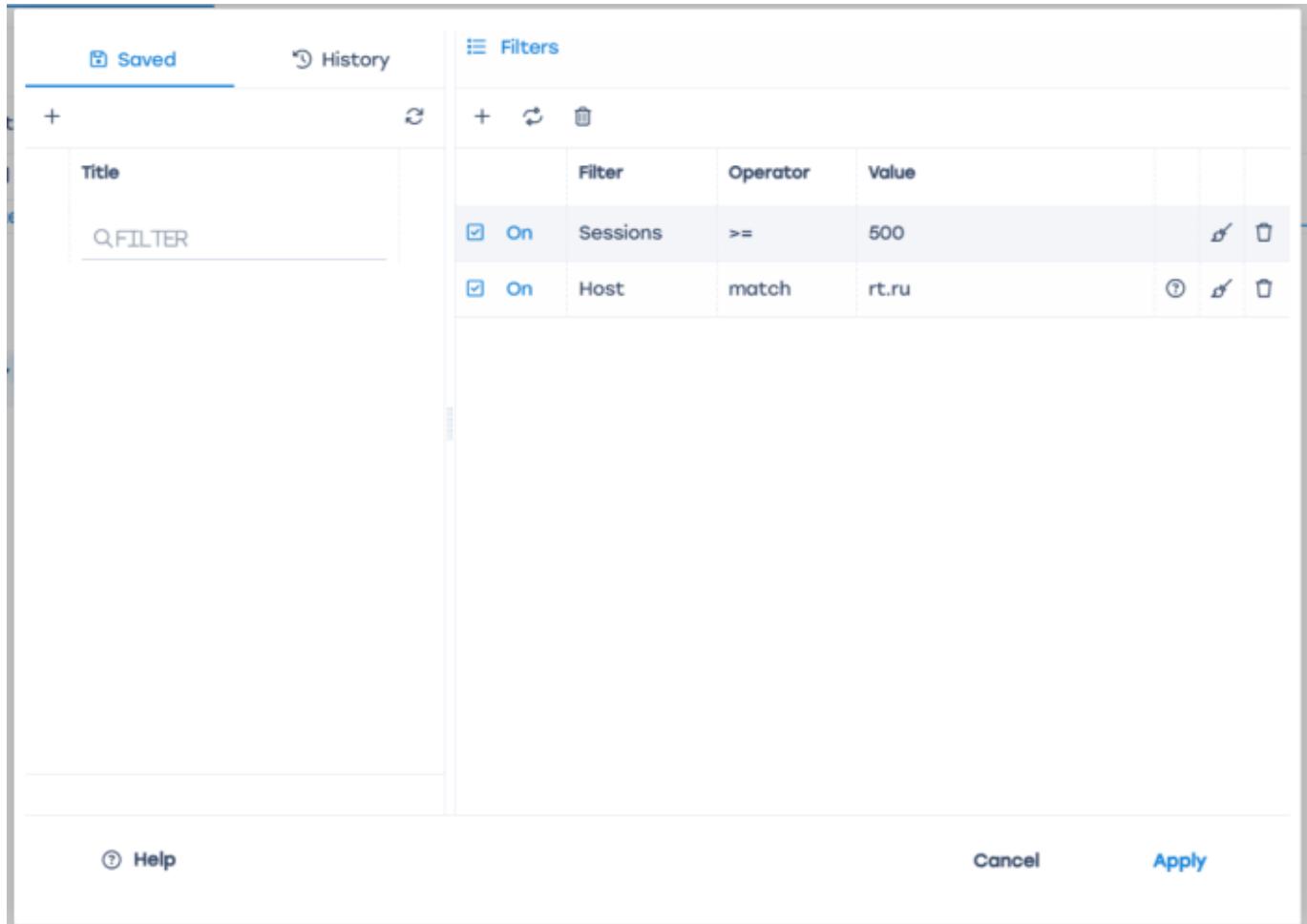
<https://spb.rt.ru/videocontrol>

<https://ru.ivideon.com/>

Ivideon is easy to find by host, example query:

```
host like
ivideon.com
```

Rostelecom is more difficult, because many services of this large operator share the same host. Let's slightly complicate the query, since cameras are known to work continuously and generate hundreds of sessions per day:



The screenshot shows a filtering interface with the following details:

- Saved:** A tab labeled "Saved" is selected.
- History:** A tab labeled "History" is visible.
- Title:** The title field contains "Q FILTER".
- Filters:** The "Filters" section contains two rows:
 - Filter: "Sessions", Operator: ">=", Value: "500".
 - Filter: "Host", Operator: "match", Value: "rt.ru".
- Buttons:** At the bottom are "Help", "Cancel", and "Apply" buttons.

Partnership with local businesses (pizza, stores, car services)

Building audience segments for local stores or businesses.

For example, to build a “warm” subscriber base, you can select those who visit a competitor car service website, order auto parts on exist.ru, or regularly use Ozon or AliExpress.

You need to configure simple host-based filters for clickstream, or use the built-in clickstream category-based segmentation.