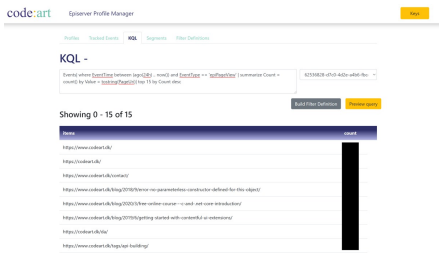
ALLAN THRAEN | 3 years ago | PDF |

Behavior Analytics Big Data Optimizely (Episerver)

PROFILE MANAGER V2 - KQL EDITION



We just launched a new version of the online tool Profile Manager - a tool that makes it easier for developers and content analysts to work with Episervers Profile Store. The new version lets you easily try out different KQL queries and build Filter Definitions with them.

Quite a while ago, we launched an online tool that seems to have gotten pretty popular - Profile Manager. It's a tool that lets developers and analysts access the data in the Episerver Profile store in a more flexible way than the built-in user interface. It's a completely browser based tool that calls the Episerver Rest API directly from your local browser with your queries or changes to the data.

But ever since Episerver launched version 2 of the profile store, introducing support for KQL (Kusto Query Language) it's been on our wishlist to make it easy to try out KQL queries and build up Filter Definitions with it. And now we have (credits to Kseniia Zar for the coding)! Now you can use the Profile Manager tool to try out KQL queries and when you are happy with them you can store them as filter definitions.

Recap - Profile Store basics

Ok - there are many new concepts and tricky constraints to understand when you start working with the profile store, so perhaps a little recap can be useful.

A website that tracks behavior to Episervers profile store typically track ***events***. The most common is probably the Page View event - but often you'd also want to track other user interactions such as form submissions, product purchases and so on.

Episerver then builds **Profiles** from those events, aggregating user data learned by the events. It's possible to directly update the profiles as well - but that is mostly used for importing or enriching profiles with data. Profiles and Events can both contain custom 'payload' data as well as the predefined structured data about the event and the visitor.

Each event has a DeviceId - unique key for the device that accessed it (typically stored in a cookie called _madid in the browser). Profiles have a list of DeviceIds (as one profile could be using multiple devices - but merging is still a tricky area) - and this is how Profiles and events are connected. Both also have a dimension called 'scope' which is essentially a string. The idea is that multiple different website for example can track in different scopes. Default behavior is to use the Site Guid identifier as the scope.

It's possible to query profiles and events using an 'odata-like' syntax (example: "EventType eq 'epiPageView' will return events of the type epiPageView) - but it is very limited to say the least. A great addition was added when Episerver released v2 of the profile store API, cause now you also get the option of using the full power of KQL (Kusto Query Language - **not** to be confused with Keyword Query Language) - but only against Events. The above example would be "Events|where EventType == 'epiPageView" - but you can really doso much more in KQL. In a future blog post I'll share some really cool episerver-related examples I've collected over the time.

A downside to the Episerver profile store, is that the UI it comes with is still pretty limited - which is why we created the Profile Manager tool - a tool that really lets you browse, query and change profiles, events and some of the other entities in the profile store.

The odata-like queries could be stored as **segments** and used to segment users. You could extract lists of profiles in a segment, or even use them in Visitor Groups and personalize content for them.

With **KQL** queries you can store your queries as **FilterDefinitions** - and these stored queries are really useful, as they can even contain parameters. Then, an editor can in the UI build new segments by putting together building blocks of FilterDefinitions with custom parameters. The best analogy I can think of here is how a **VisitorGroup** can be built from different **Criteria** (that also have custom parameters). At least, that's how I think it's intended to go. But at this point segments can only contain 1 filter definition - but still quite useful.

Confused yet? Don't worry. Further down I'll take you through a fun use case. But first, let's look at what the Profile Manager offers.

The Profile Manager tool

First of all, you find the profile manager tool here: **https://profiles.codeart.dk/**

It's worth noting that even though it's hosted there, it's a 100% pure client side page - this means that no data is sent back to our servers. For convenience you can setup multiple Url/Key pairs for different profile stores - they are stored in your browsers local storage, so be aware that you'll loose them if you change device or clear stored site settings in your browser. We did, however also for this release add the ability to export the keys to a file - and we hope in the future to support importing them as well.

Once connected to an instance, you get your choice of 5 tabs:

- **Profiles**.
This is where you can see the profiles, or query them with odata-like syntax. You can also edit profiles (in json) or delete profiles (but be aware that this automatically deletes any associated events). If you do a custom query against the profiles you can even click a shortcut button and create a segment based on the query.
- **Events**.
Similar to Profiles - except here you can use the odata like query against events and see their details. Since Events are considered immutable you cannot change them.
There is a shortcut button for each event that takes you back to the profile listing and executes a search for the parent profile for that event.
- **KQL**.
This is one of the new tabs we added. This is where you can experiment with KQL statements and preview the results. However, from Episervers side this preview is limited to 1000 rows. Since KQL is only executed within a specific scope, you have to pick the scope from a list before previewing. Typically, you would create KQL that returns events - and in that case, the events will be shown just as on the "Events" page. But since KQL is extremely powerful, you can pretty much return any kind of data - like a summarized dataset of the 10 most popular pages in the last 24h:

```
Events|where EventTime between (ago(24h) .. now()) and EventType == 'epiPageView'|summarize cnt=cou
```

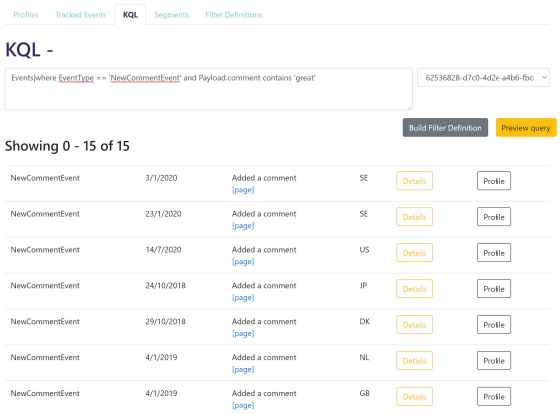
A really powerful shortcut is the ability to create a Filter Definition straight from a KQL query. You just need to tweak the query to use placeholders using the {{placeholder}} syntax in the query and list their names and types.

- **Segments**.
This tab lets you list, query and edit the segments defined in the profile store. Remember - segments was what can be used in Visitor Groups for personalization. And they either consist of odata like queries or of FilterDefinitions.
- **FilterDefinitions.**
The final tab lists the filter definitions and lets you edit or delete them. You can also see the filter definitions and use them to create segments in Episervers own UI. Again - FilterDefinitions can be a bit hard to wrap your head around, but think of them as KQL building blocks you can use to define segments. In many cases you can probably benefit from the same filter definitions as many others - and for a future version we plan to include a number of common FilterDefinitions in Profile Manager, so you can easily add them.

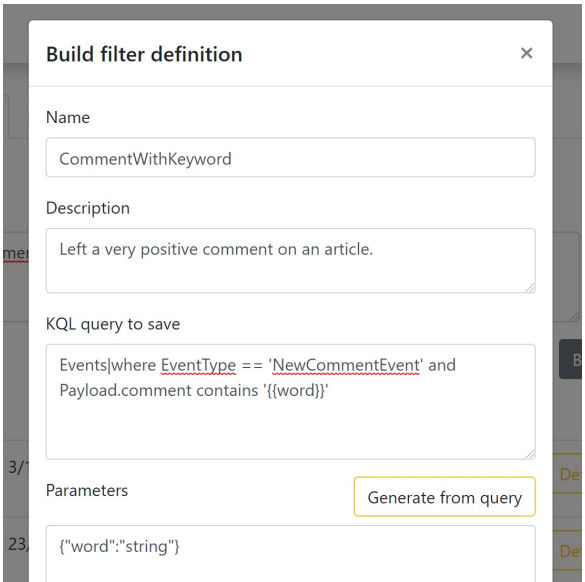
Now, the promised use case

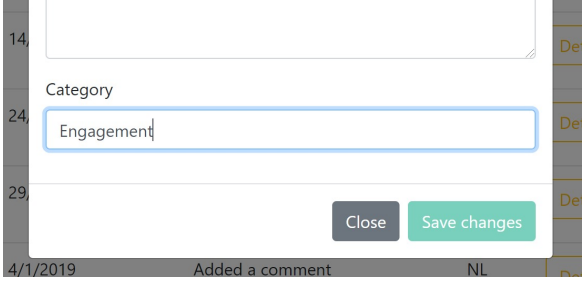
On this blog - in fact, you can see it if you scroll down - there is a commenting functionality. I love it when people leave comments - and even more if they leave positive, constructive comments. So - let's reward those die hard fans that write exceptionally positive comments with a dedicated message - just for them.

First, we experiment a bit to find just the right KQL query. This one seems to do the trick. Now, we can click the "Build Filter Definition" button to create a Filter Definition.

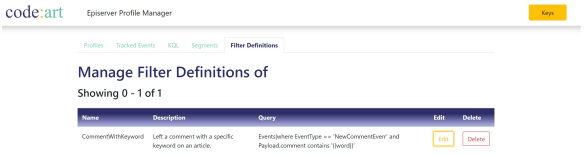


FilterDefinitions shouldn't be hardcoded, so we generalize it a bit by removing the keyword and instead inserting a placeholder. Using the "Generate from query" button we populate the "Parameters" field - and in this case the type is in fact a string. Now, we can go ahead and save changes.



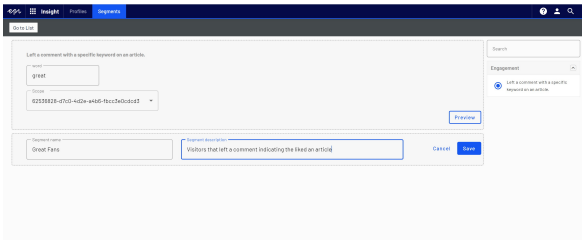


Next, we can check out the Filter Definitions tab and see that it's actually there. And edit it, if we find any mistakes.

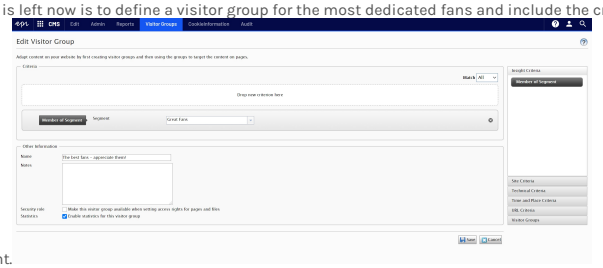


The time has now come to shift over to the "Insight" UI in Episerver - and there navigate to Segments and create a new segment. Creating a new segment takes us to a view where we can pick the FilterDefinition in the right hand side, select the scope and configure the parameter.

Once completed, we can save it - and it's now registered as a segment.



All that is left now is to define a visitor group for the most dedicated fans and include the criteria for the



segment.

And of course customize the content based on the visitor group.

Future developments

These are some of the features we plan to add to the Profile Manager in the near future. Feel free to leave a comment if you have further ideas!

- Export data and extract lists.
- Importing of key-pairs.
- Predefined filter definitions that can easily be added to your profile store.
- Add logically missing buttons to various tabs (like Create new Profile, and so on).
- Add Blacklist management
- Embed in Episervers UI.

Behavior Analytics | Big Data | Optimizely (Episerver)

CodeArt ApS
Teknikerbyen 5, 2830 Virum, Denmark
Email: info@codeart.dk
Phone: +45 26 13 66 96
CVR: 39680688