OPEN DATA HUB DATA BROWSER 2.0 - ADD REMOVE COLUMS IN TABLE VIEW

TECHPARK SÜDTIROL/ALTO ADIGE

Product Owner: Patrick Ohnewein
Project Manager: Stefano Seppi
Software Architect: Rudi Thöni

01/01/2022

Nature of Innovation

AGENDA

TARGET

Who are the users getting the benefits from the value proposition of this strategic action?

- **Open Data Hub Data Browser users** who are interested to configure their Table View to see the only the data they are interested to and store it as bookmark in their browser.
- **Open Data Hub Data Consumers** who are interested to test the results of the API calls they are implementing in their application by using the Data Browser.

HYPOTHESIS - VALUE PROPOSITION

Describe the Value Proposition for the Target. What is the benefits the target users get?

The Open Data Hub Data Browser users should have in the tooltip at the right an easy way to set the columns they are interested to see in the table view.

The Open Data Hub Data Browser users should have the possibility to store in the bookmarks of his browser the configured Table View.

The Open Data Hub Data Browser users should have the possibility to share the Table View he configured by sending a link.

The Open Data Hub Data Consumer should have the possibility to copy and paste the API call setup in the Data Browser URL and see the results they get rendered in the Table View.

SOLUTION 1

SCREEN DESIGN

The place where the user can select the columns could be in the menu that it opens on the right as it is for the "Search and Filter" and the "Export Dataset".
The user could add and remove columns either:

- Using a checkbox and choosing from the entire list of attributes present in the JSON;
- Adding the names of the attirbute in text and having an autocomplete functionality.

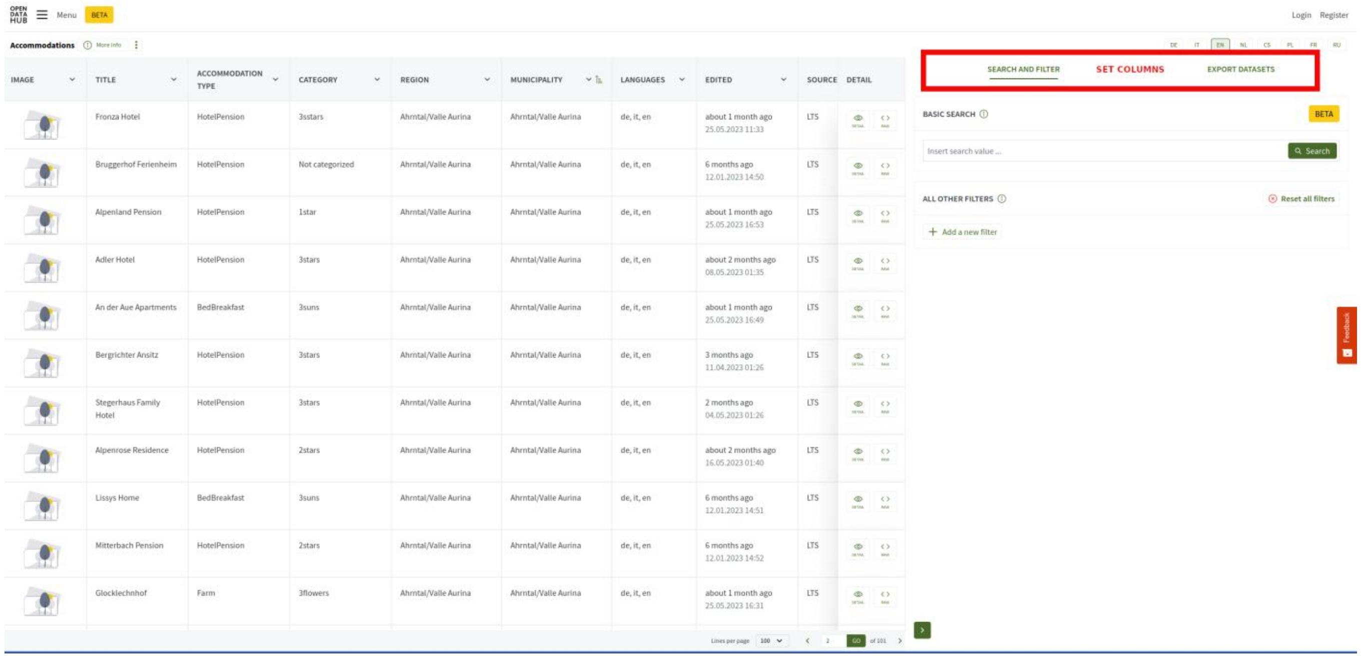

IMAGE	TITLE	ACCOMMODATION TYPE	CATEGORY	REGION	MUNICIPALITY	LANGUAGES	EDITED	SOURCE	DETAIL
	Fronza Hotel	HotelPension	3sstars	Ahrntal/Valle Aurina	Ahrntal/Valle Aurina	de, it, en	about 1 month ago 25.05.2023 11:33	LTS	
	Bruggerhof Ferienheim	HotelPension	Not categorized	Ahrntal/Valle Aurina	Ahrntal/Valle Aurina	de, it, en	6 months ago 12.01.2023 14:50	LTS	
	Alpenland Pension	HotelPension	1star	Ahrntal/Valle Aurina	Ahrntal/Valle Aurina	de, it, en	about 1 month ago 25.05.2023 16:53	LTS	
	Adler Hotel	HotelPension	3stars	Ahrntal/Valle Aurina	Ahrntal/Valle Aurina	de, it, en	about 2 months ago 08.05.2023 01:35	LTS	
	An der Aue Apartments	BedBreakfast	3suns	Ahrntal/Valle Aurina	Ahrntal/Valle Aurina	de, it, en	about 1 month ago 25.05.2023 16:49	LTS	
	Bergrichter Ansitz	HotelPension	3stars	Ahrntal/Valle Aurina	Ahrntal/Valle Aurina	de, it, en	3 months ago 11.04.2023 01:26	LTS	
	Stegerhaus Family Hotel	HotelPension	3stars	Ahrntal/Valle Aurina	Ahrntal/Valle Aurina	de, it, en	2 months ago 04.05.2023 01:26	LTS	
	Alpenrose Residence	HotelPension	2stars	Ahrntal/Valle Aurina	Ahrntal/Valle Aurina	de, it, en	about 2 months ago 16.05.2023 01:40	LTS	
	Lissys Home	BedBreakfast	3suns	Ahrntal/Valle Aurina	Ahrntal/Valle Aurina	de, it, en	6 months ago 12.01.2023 14:51	LTS	
	Mitterbach Pension	HotelPension	2stars	Ahrntal/Valle Aurina	Ahrntal/Valle Aurina	de, it, en	6 months ago 12.01.2023 14:52	LTS	
	Glocklechnhof	Farm	3flowers	Ahrntal/Valle Aurina	Ahrntal/Valle Aurina	de, it, en	about 1 month ago 25.05.2023 16:31	LTS	

TECHNICAL SOLUTION

One possible way could be to use the "fields" feature in the tourism API and the "select" feature in the mobility API to select the attributes to be queried and displayed at an API level and not at a frontend level.

The configuration files that we use will be kept, but they will provide only the information about how to render the single values (e.g. which component use, column name, column width, etc.) and not which one will be rendered.

The attribute list to be rendered in the Table View will be put in the URL as parameter of the "fields" or "select" feature and as list in the "Set Columns" section mentioned in the previous slide.

The Data Browser will render only all attributes included in the JSON received by the API and if the attribute is empty but present in the JSON will be rendered as empty.

From a design point of view the rendering should work as follows:

- If the "Embedded view" is selected and the attribute is present in the default configuration file, the attribute will be rendered as specified in the configuration;
- If the "Embedded view" is selected and the attribute isn't configured in the default configuration file, the attribute will be rendered as in the "Generated View";
- If the "Generated view" is selected, all attribute will be rendered as in the "Generated View".

OPEN QUESTIONS

- (Stefano) Where do we store the default column configuration we have?
    - (Stefano) Maybe we do not need to store this configuration, we could put the link in the metadata API as default link to table view.

- (Christian) Do we want to have that config on a global, endpoint or even user level? Global means that the visualization for a field is the same for all tables, endpoint means that the visualization for a field can be different for each endpoint, user means that a user can override the config (again: globally? per endpoint?). Note that having only a global level config could be problematic when different endpoints have same-named fields that contain semantically different data.
    - (Christian) I think the problem mentioned herein could be resolved by having some priority ordered config resolution: user level (endpoint) > user level (global) (if wanted) > endpoint level > global level
    - (Stefano) In a first iteration I would avoid to let the user change the rendering configuration, this is a feature that in case we could add in next iterations.

- If a user should be able to change the renderer for a column, then this information won't be part of the URL. Therefor, sharing URL's won't guarantee that all see the same result
    - (Stefano) In a first iteration I would avoid to let the user change the rendering configuration, this is a feature that in case we could add in next iterations.

Stefano

- Setup of the Table View easily sharable by using the URL.
- Setup of the Table View storable as bookmark in the Browser.
- Same behavior on Data Browser and API side.

Christian

- the URL contains the information what columns to show. If the URL is shared with others, those will see the same information as anybody else (with some restrictions like user defined rendering)
- if the user copies the URL from the given link-tool in the right toolbox, it will see exactly the same data (in raw) as it is shown in the table - there will be no additional fields
- we could take the effort to unify the generated and embedded TableViews (the concept can't be applied 1:1 to the Detail-/Edit-/QuickViews without affecting usability)

CONTRA

Stefano

- Each attribute can be rendered in the Table View only one time.
- URL could become very long?

Christian

- added complexity for the implementation as the cases to support increase (see next points)
  - we have to support two different config concepts: one for the TableView and one for the Detail-/Edit-/QuickViews
  - at least the TourismAPI behaves differently when "fields" are set or not: in the former case, the response data is a flattened array, in the latter case it is a tree structure. This two behaviors must be handled and implemented separately
  - we mix data retrieval with data visualization (two different dimensions) into one dimension. This prevents us to support some use cases, e.g. it won't be possible to show an image and its URL in two different columns. Of course it will always be possible to implement complex render components that provide that information, but those won't simplify things.
  - It is not clear how to support the use-case that a user can define the TableView columns to show and how to order them. It looks like we need to come back to the concept of JSON tree with adjacent checkboxes to define columns and ordering? Or use some auto-complete mechanism? Having the user to add e.g. "ImageGallery.0.ImageUrl" manually to a query param in the URL seems overly complex fo r an ordinary user
  - It is not clear how the visualization for a column can be customized (if we want to support that feature). The only way I see at the moment is to have a user-defined config, that now has to be merged with the "fields" / "select" information. This info can't be shared anymore easily, so the idea of having a link that defines visualization is not 100% working for this use-case - at least of now
  - all endpoints should provide support for "fields", "select" or similar ways to restrict the data to be shown (note that e.g. https://tourism.opendatahub.com/swagger/index.html#/Weather/get_v1_Weather_SnowReport or https://swagger.opendatahub.com/?url=https://mobility.api.opendatahub.com/v2/apispec#/Mobility%20V2/get_v2__representation_ do not provide such capabilities). This is not a hard requirement, but if is not fulfilled we need to handle each case (support for "fields" / "select" or not) for each API, which adds complexity
  - URLs could get very long (maybe only a minor / esthetical issue). The example URL for current ODHActivityPoi: https://v2-beta.databrowser.opendatahub.com/dataset/table/tourism/v1/ODHActivityPoi/?fields=ImageGallery.0.ImageUrl,ContactInfos.en.LogoUrl,Detail.en.Title,AdditionalPoiInfos.en.Categories,LocationInfo.RegionInfo.Name.en,LocationInfo.MunicipalityInfo.Name.en,ContactInfos.en.Url,HasLanguage,LastChange,Source,Active,OdhActive

TECHPARK SÜDTIROL/ALTO ADIGE

THANK YOU.

NOI TECHPARK
SÜDTIROL/ALTO ADIGE

VIA A.-VOLTA-STRASSE 13/A
I – 39100 BOZEN/BOLZANO

+39 0471/066 600
INFO@NOI.BZ.IT
WWW.NOI.BZ.IT