

Attachments Filters

The **Attachments Filters** in Replyke allows developers to filter entities based on the presence or absence of attachments content. This functionality provides a straightforward way to include or exclude entities that contain attachments, ensuring the entity list aligns with your application's needs.

Overview of Attachments Filters

The `attachmentsFilters` property can be passed into the `EntityListProvider` as an object with the following structure:

```
export interface AttachmentsFilters {  
  hasAttachments?: boolean;  
}
```

- `hasAttachments`: A boolean flag to include or exclude entities based on whether they contain attachments (`true`) or not (`false`).

Dynamically Updating Attachments Filters

Developers can dynamically update the attachments filters using the `setAttachmentsFilters` function provided by the `useEntityList` hook. This function expects a new `attachmentsFilters` object or `null` to clear the filters.

```
const { setAttachmentsFilters } = useEntityList();  
  
// Set a filter to include entities with attachment  
setAttachmentsFilters({ hasAttachments: true });  
  
// Reset attachment filters  
setAttachmentsFilters(null);
```

How Attachments Filters Work

Attachments Filters are applied on the backend when querying entities. The `hasAttachments` flag works as follows:

1. `true` : Includes only entities that have attachments (e.g., non-empty attachment arrays).
2. `false` : Includes only entities that do not have attachments (e.g., empty attachments arrays).
3. `null` : Clears the attachments filter, meaning no filtering is applied based on attachments presence.

Example Use Cases

1. Passing Static Filters to the EntityListProvider

To set static filters for attachments presence at the initialization of the entity list, pass them directly to the `EntityListProvider`:

```
<EntityListProvider attachmentsFilters={{ hasAttachments: true }}>
  <MyFeedComponent />
</EntityListProvider>
```

This setup ensures that only entities with attachments are included in the entity list.

2. Dynamically Updating Filters Based on User Interaction

You can also dynamically adjust the attachments filters based on user actions, such as clicking a toggle button:

```
const { setAttachmentsFilters, entities, loadMore } = useEntityList();

const toggleAttachmentsFilter = (hasAttachments: boolean) => {
  setAttachmentsFilters({ hasAttachments });
};

return (
  <div>
    <h1>Filtered by Attachments</h1>
    <div>
      <button onClick={() => toggleAttachmentsFilter(true)}>Show Only With Attachments</button>
      <button onClick={() => toggleAttachmentsFilter(false)}>Show Only Without Attachment</button>
      <button onClick={() => setAttachmentsFilters(null)}>Clear Attachments Filter</button>
    </div>
    <ul>
      {entities.map((entity) => (
        <li key={entity.d}>
          <h2>{entity.title}</h2>
          {entity.attachments.length > 0 ? <p>Attachments included</p> : <p>No attachment</p>}
        </li>
      ))}
    </ul>
  </div>
)
```

```
    </ul>
    <button onClick={loadMore}>Load More</button>
  </div>
);
```

In this example, clicking the buttons dynamically updates the attachments filters to include or exclude entities based on attachments presence.

Important Notes

- Attachments Filters operate on the presence of attachments content in entities, typically represented by non-empty or empty attachments arrays.
- Filters are applied immediately, and the entity list resets to reflect the updated conditions.
- Passing `null` to `setAttachmentsFilters` clears all attachments-based filtering.

Conclusion

Attachments Filters provide a simple yet powerful mechanism to fine-tune entity list's results based on the presence of attachments. Whether you want to include only entities with attachments, exclude them, or dynamically adjust filters based on user input, Attachments Filters help you achieve this with ease. Use them to create more tailored and engaging experiences in your application.

Last updated on May 7, 2025