

# useEntityListData

## Overview

The `useEntityListData` hook provides a powerful and flexible solution for managing and presenting entity data in a feed. Unlike individual hooks like `useFetchManyEntities` or `useCreateEntity`, this hook combines functionalities such as fetching, creating, deleting, and managing entities while offering advanced filter customizations and state management. It is ideal for implementing dynamic feeds with tailored sorting and filtering options.

## Usage Example

```
import { useEntityListData } from "@replyke/react-js";

function Feed() {
  const {
    entities,
    infusedEntities,
    loading,
    hasMore,
    sortBy,
    setSortBy,
    loadMore,
    resetEntities,
    createEntity,
    deleteEntity,
  } = useEntityListData({ limit: 10, sortBy: "hot" });

  const handleCreateEntity = async () => {
    try {
      await createEntity({
        title: "New Post",
        content: "This is a new post content",
      });
    } catch (error) {
      console.error("Failed to create entity:", error.message);
    }
  };

  return (
    <div>
      <h1>Feed</h1>
      <ul>
        {entities.map(entity => (
          <li>{entity.title} - {entity.content}</li>
        ))}
      </ul>
      <button onClick={handleCreateEntity}>Create New Post</button>
    </div>
  );
}
```

```

<div>
  <select
    value={sortBy}
    onChange={(e) => setSortBy(e.target.value as "hot" | "new")}
  >
    <option value="hot">Hot</option>
    <option value="new">New</option>
  </select>

  <ul>
    {entities.map((entity) => (
      <li key={entity.id}>{entity.title}</li>
    )))
  </ul>

  {loading && <p>Loading...</p>}
  {hasMore && !loading && (
    <button onClick={loadMore}>Load More</button>
  )}

  <button onClick={handleCreateEntity}>Create New Entity</button>
</div>
);
}

```

## Parameters & Returns

### Parameters

The hook accepts an object with the following fields:

Parameter	Type	Required	Description
<code>limit</code>	<code>number</code>	No	The number of entities to fetch per page. Default is <code>10</code> .
<code>sortBy</code>	<code>FeedSortByOptions</code>	No	The default sorting criteria for the feed (e.g., <code>hot</code> , <code>new</code> ).
<code>timeFrame</code>	<code>TimeFrame</code>	No	The time range for fetching entities.

Parameter	Type	Required	Description
sourceId	string   null	No	Filters entities created by a source ID.
userId	string   null	No	Filters entities created by a specific user.
followedOnly	boolean	No	If true , fetches entities from followed users only. Default is false .
keywordsFilters	KeywordsFilters	No	Filters entities based on keywords.
titleFilters	TitleFilters	No	Filters entities based on titles.
contentFilters	ContentFilters	No	Filters entities based on content.
attachmentsFilters	AttachmentsFilters	No	Filters entities based on attachments presence.
locationFilters	LocationFilters	No	Filters entities based on location.
metadataFilters	MetadataFilters	No	Filters entities based on metadata.
idle	boolean	No	If true , the hook starts in an idle state and fetches entities only when activated.
onReset	() => void	No	Callback executed when the feed is reset.

Parameter	Type	Required	Description
<code>infuseData</code>	<code>(foreignId: string) =&gt; Promise&lt;Record&lt;string, any&gt;   null&gt;</code>	No	A function to fetch and add additional data to each entity.

## Returns

The hook returns an object with the following fields:

Return Value	Type	Description
<code>entities</code>	<code>Entity[]</code>	The flat list of entities fetched so far.
<code>infusedEntities</code>	<code>(Entity &amp; Record&lt;string, any&gt;)[]</code>	The list of entities infused with additional data (if <code>infuseData</code> is provided).
<code>loading</code>	<code>boolean</code>	Indicates whether entities are being loaded.
<code>hasMore</code>	<code>boolean</code>	Indicates whether there are more entities to fetch.
<code>resetting</code>	<code>boolean</code>	Indicates whether the feed is resetting.
<code>sortBy</code>	<code>FeedSortByOptions   null</code>	The current sorting criteria for the feed.
<code>setsortBy</code>	<code>(sortBy: FeedSortByOptions) =&gt; void</code>	A function to update the sorting criteria.
<code>timeFrame</code>	<code>TimeFrame   null</code>	The current time range filter for the feed.

Return Value	Type	Description
<code>setTimeFrame</code>	<code>(timeFrame: TimeFrame   null) =&gt; void</code>	A function to update the time range filter.
<code>sourceId</code>	<code>string   null</code>	The current source ID filter.
<code>setSourceId</code>	<code>(userId: string   null) =&gt; void</code>	A function to update the source ID filter.
<code>userId</code>	<code>string   null</code>	The current user ID filter.
<code>setUserId</code>	<code>(userId: string   null) =&gt; void</code>	A function to update the user ID filter.
<code>followedOnly</code>	<code>boolean</code>	Indicates whether only followed users' entities are fetched.
<code>setFollowedOnly</code>	<code>(state: boolean) =&gt; void</code>	A function to toggle the <code>followedOnly</code> filter.
<code>keywordsFilters</code>	<code>KeywordsFilters   null</code>	The current keywords filter.
<code>updateKeywordsFilters</code>	<code>(type: "add"   "remove"   "reset"   "replace", key: "includes"   "doesNotInclude"   "both", value?: string   string[]) =&gt; void</code>	A function to modify the keywords filter.
<code>titleFilters</code>	<code>TitleFilters   null</code>	The current title filters.
<code>setTitleFilters</code>	<code>(newTitleFilters: TitleFilters   null) =&gt; void</code>	A function to update the title filters.
<code>contentFilters</code>	<code>ContentFilters   null</code>	The current content filters.

Return Value	Type	Description
<code>setContentFilters</code>	<code>(newContentFilters: ContentFilters   null) =&gt; void</code>	A function to update the content filters.
<code>attachmentsFilters</code>	<code>AttachmentsFilters   null</code>	The current attachments filters.
<code>setAttachmentsFilters</code>	<code>(newAttachmentsFilters: AttachmentsFilters   null) =&gt; void</code>	A function to update the attachments filters.
<code>locationFilters</code>	<code>LocationFilters   null</code>	The current location filters.
<code>setLocationFilters</code>	<code>(location: LocationFilters   null) =&gt; void</code>	A function to update the location filters.
<code>metadataFilters</code>	<code>MetadataFilters   null</code>	The current metadata filters.
<code>setMetadataFilters</code>	<code>(newMetadataFilters: MetadataFilters   null) =&gt; void</code>	A function to update the metadata filters.
<code>kickstart</code>	<code>() =&gt; void</code>	A function to activate the feed if it starts in an idle state.
<code>loadMore</code>	<code>() =&gt; void</code>	A function to fetch the next page of entities.
<code>resetEntities</code>	<code>() =&gt; Promise&lt;void&gt;</code>	A function to reset the feed and fetch the first page of entities.
<code>createEntity</code>	<code>(props: { foreignId?: string; title?: string; content?: string; media?: any[]; keywords?: string[]; location?: { latitude: number; longitude: number }; metadata?: Record&lt;string, any&gt;; }) =&gt; void</code>	A function to create a new entity and optionally insert it into the feed.

Return Value	Type	Description
	<pre>insertPosition?: "first"   "last" }) =&gt; Promise&lt;void&gt;</pre>	
<code>deleteEntity</code>	<pre>(props: { entityId: string }) =&gt; Promise&lt;void&gt;</pre>	A function to delete an entity from the feed.

Last updated on May 7, 2025