useEntityComments

Overview

The `useEntityComments` hook provides a higher-level abstraction for fetching and managing comments related to a specific entity. It handles pagination, sorting, and comment tree structure, allowing you to easily build a threaded comment UI. It's especially useful when you want to render nested replies and dynamically manage additions or deletions in a structured way.

Usage Example

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

function EntityComments({ entityId }: { entityId: string }) {
  const {
    comments,
    newComments,
    entityCommentsTree,
    loading,
    hasMore,
    sortBy,
    setSortBy,
    loadMore,
    addCommentsToTree,
    removeCommentFromTree,
  } = useEntityComments({ entityId, limit: 10, defaultSortBy: "new" });

  return (
    <div>
      <select
        value={sortBy}
        onChange={(e) => setSortBy(e.target.value as "new" | "top")}
      >
        <option value="new">Newest</option>
        <option value="top">Top</option>
      </select>

      <ul>
        {comments.map((comment) => (
          <li key={comment.id}>{comment.content}</li>
        ))}
      </ul>
```

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

Parameters & Returns

Parameters

The hook accepts an object with the following fields:

Parameter	Type	Required	Description
`entityId`	`string | undefined | null`	Yes	The ID of the entity whose comments are being fetched.
`limit`	`number`	No	The number of comments to fetch per page. Default is `10`.
`defaultSortBy`	`CommentsSortByOptions`	No	The default sorting criteria for comments (e.g., `"new"`, `"top"`).

Returns

The hook returns an object with the following fields:

Return Value	Type	Description
`entityCommentsTree`	`EntityCommentsTree`	The structured tree of comments, keyed by comment ID.
`comments`	`Comment[]`	The list of root-level comments (non-new).
`newComments`	`Comment[]`	Comments recently added during the session.

Return Value	Type	Description
`loading`	`boolean`	Indicates whether comments are currently loading.
`hasMore`	`boolean`	Whether there are more comments available to fetch.
`sortBy`	`CommentsSortByOptions`	Current sorting method for the comments (`"new"` or `"top"`).
`setSortBy`	`(newSortBy: CommentsSortByOptions) => void`	Updates the sort order for comments.
`loadMore`	`() => void`	Loads the next page of comments, if available.
`addCommentsToTree`	`(newComments: Comment[] | undefined, newlyAdded?: boolean) => void`	Adds comments to the internal tree structure.
`removeCommentFromTree`	`(commentId: string) => void`	Removes a comment (and its descendants) from the tree structure.

Last updated on May 7, 2025