

## useEntityList Hook

The `useEntityList` hook provides access to all the context and functionality exposed by the `EntityListProvider`. It allows developers to interact with the entity list's data, manage its state, and dynamically modify filters. Below is the interface for all available values and functions:

```
export interface useEntityListDataValues {
  entities: Entity[];
  infusedEntities: (Entity & Record<string, any>)[];

  loading: boolean;
  hasMore: boolean;
  resetting: boolean;

  sortBy: EntityListSortByOptions | null;
  setSortBy: (sortBy: EntityListSortByOptions) => void;
  timeFrame: TimeFrame | null;
  setTimeFrame: (timeFrame: TimeFrame | null) => void;
  sourceId: string | null;
  setSourceId: (sourceId: string | null) => void;
  userId: string | null;
  setUserId: (userId: string | null) => void;
  followedOnly: boolean;
  setFollowedOnly: (state: boolean) => void;

  keywordsFilters: KeywordsFilters | null;
  updateKeywordsFilters: (
    type: "add" | "remove" | "reset",
    key: "includes" | "doesNotInclude" | "both",
    value?: string | string[]
  ) => void;
  metadataFilters: MetadataFilters | null;
  setMetadataFilters: (metadata: MetadataFilters | null) => void;
  titleFilters: TitleFilters | null;
  setTitleFilters: (metadata: TitleFilters | null) => void;
  contentFilters: ContentFilters | null;
  setContentFilters: (metadata: ContentFilters | null) => void;
  attachmentsFilters: AttachmentsFilters | null;
  setAttachmentsFilters: (metadata: AttachmentsFilters | null) => void;
  locationFilters: LocationFilters | null;
  setLocationFilters: (location: LocationFilters | null) => void;

  kickstart: () => void;
  loadMore: () => void;
  resetEntities: () => Promise<void>;
  createEntity: (props: {
```

```

    foreignId?: string;
    sourceId?: string;
    title?: string;
    content?: string;
    attachments?: Record<string, any>[];
    keywords?: string[];
    location?: {
        latitude: number;
        longitude: number;
    };
    metadata?: Record<string, any>;
    insertPosition?: "first" | "last";
}) => Promise<void>;
deleteEntity: (props: { entityId: string }) => Promise<void>;
}

```

## Explanation of Hook Values and Functions

### Entities and Data Management

1. `entities` : Contains an array of all entities fetched for the list.
2. `infusedEntities` : If an `infuseData` function is provided to the provider, this array contains entities with additional external data merged. The `entities` array remains “pure,” holding only the data fetched by Replyke.
3. `loading` : A boolean flag indicating whether entities are currently being fetched. Useful for showing loading indicators in the UI.
4. `hasMore` : Indicates whether there are more entities to fetch. If `false`, the entity list has reached its end and there is no more data to be fetched.
5. `resetting` : A flag indicating that the entity list is currently being reset. When `resetting` is true, `loading` will also be true, but not vice versa.

### Sorting

6. `sortBy` : Holds the current sorting criteria for the entity list.
7. `setSortBy` : Updates the sorting criteria dynamically.

### Filter States and Functions

The following values represent the current state of entity list filters, and corresponding functions allow dynamic updates. Each filter change will reset the entity list.

8. `timeFrame` : The current state of the time frame filter.
9. `setTimeFrame` : Updates the time frame filter.

10. `sourceId` : The current source ID filter.
11. `setSourceId` : Updates the source ID filter.
12. `userId` : The current user ID filter.
13. `setUserId` : Updates the user ID filter.
14. `followedOnly` : A boolean indicating whether the entity list is limited to entities from followed accounts.
15. `setFollowedOnly` : Updates the `followedOnly` filter.
16. `keywordsFilters` : The current keywords filter state.
17. `updateKeywordsFilters` : Dynamically updates keywords filters by adding, removing, or resetting values.
18. `titleFilters` : The current state of title filters.
19. `setTitleFilters` : Updates the title filters.
20. `contentFilters` : The current state of content filters.
21. `setContentFilters` : Updates the content filters.
22. `attachmentsFilters` : The current state of attachments filters.
23. `setAttachmentsFilters` : Updates the attachments filters.
24. `locationFilters` : The current state of location filters.
25. `setLocationFilters` : Updates the location filters.
26. `metadataFilters` : The current state of metadata filters.
27. `setMetadataFilters` : Updates the metadata filters.

## Actions and Mutations

26. `kickstart` : A function that triggers the first fetch of entities manually when the `idle` property in the `EntityListProvider` is set to true.
27. `loadMore` : Fetches additional entities for the list.
28. `resetEntities` : Resets the list, refreshing data without altering filter states.
29. `createEntity` : Creates a new entity within the list's context. The newly created entity is automatically added to the list.
30. `deleteEntity` : Deletes an entity within the list's context. The deleted entity is automatically removed from the list.

This hook gives you complete control over managing and customizing the list's behavior, enabling a wide range of use cases. In the following chapters, we'll explore practical examples of implementing the `EntityListProvider` with `useEntityList` using filters in our app.

