useInfusedData

Overview

The `useInfusedData` hook is used to enrich a list of entities with external data that is not stored in Replyke's system. By providing a custom `infuseData` function, this hook fetches additional details for each entity using its `foreignId`. It also caches the fetched data to minimize redundant network requests and optimizes performance by only refetching data when the entities change.

Usage Example

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

function InfusedEntityList({ entities }: { entities: Entity[] }) {
  const infusedEntities = useInfusedData({
    entities,
    infuseData: async (foreignId) => {
      // Example API call to fetch external data
      const response = await fetch(`https://api.example.com/details/${foreignId}`);
      if (!response.ok) throw new Error("Failed to fetch data");
      return response.json();
    },
  });

  return (
    <ul>
      {infusedEntities.map((entity) => (
        <li key={entity.id}>
          {entity.title} - Additional Data: {JSON.stringify(entity.infusion)}
        </li>
      ))}
    </ul>
  );
}
```

Parameters & Returns

Parameters

The hook accepts an object with the following fields:

Parameter	Type	Required	Description
`entities`	`Entity[]`	Yes	The list of entities to be infused with external data.
`infuseData`	`(foreignId: string) => Promise<Record<string, any>	null>`	No	A function to fetch external data for an entity by its `foreignId`.

Returns

The hook returns a list of infused entities:

Return Value	Type	Description
`infusedEntities`	`(Entity & { infusion: Record<string, any> })[]`	The enriched list of entities with external data included.

Last updated on May 7, 2025