# **Fetching and Updating Data with React Query**

**React Query** is a powerful data-fetching, cashing and state management library for React applications. It simplifies API calls by **handling caching**, **background** updates, **synchronization**, **pagination**, and **retries automatically**.

### **React query Vs. Axios:**

* **Use Axios** if you only need to make simple API requests (GET, POST, etc.), handle responses, and manage errors manually. It provides more control but requires additional logic for caching, retries, and state management.
* **Use react-query** if you need advanced features like automatic caching, background fetching, data synchronization, pagination, and stale data management. It simplifies API state management and improves performance.

### **React query features:**

1. **Automatic Caching & Background Fetching**
2. **Refetching & Stale Data Handling** – Keeps your UI updated with the latest data.
3. **Optimistic Updates** – Instantly update UI while the request is still in progress.
4. **Automatic Retries** – Handles network errors with retry logic.
5. **Pagination & Infinite Scroll** – Built-in support for paginated API responses.
6. **Mutation Handling** – Easily handle POST, PUT, DELETE requests with rollback support.

### **What is Cashing?**

**Caching** is the process of storing frequently used data in memory (or a temporary storage location) to **improve performance** and **reduce redundant API calls**. Instead of fetching the same data from a server every time, cached data can be quickly retrieved from memory, making apps **faster and more efficient**.