React Query (TanStack Query)

React Query (TanStack Query)

La meilleure solution pour gérer l'état serveur : cache automatique, refetch, états de chargement/erreur, pagination, optimistic updates...

```
npm install @tanstack/react-query
```

Configuration

```
import { QueryClient, QueryClientProvider } from '@tanstack/react-query';

const queryClient = new QueryClient({
  defaultOptions: {
    queries: { staleTime: 1000 * 60 * 5 }, // 5 minutes
  },
});

<QueryClientProvider client={queryClient}>
  <App />
</QueryClientProvider>
```

useQuery — lecture de données

```
import { useQuery } from '@tanstack/react-query';

function Users() {
  const { data, isLoading, isError, error, refetch } = useQuery({
    queryKey: ['users'],          // Clé de cache unique
    queryFn: () => api.get('/users'),
    staleTime: 1000 * 60,         // 1 minute avant refetch
  });
```

```
  if (isLoading) return <Spinner />;
  if (isError) return <p>Erreur : {error.message}</p>;

  return <ul>{data.map(u => <li key={u.id}>{u.nom}</li>)}</ul>;
}

// Avec paramètre (refetch automatique quand userId change)
const { data: user } = useQuery({
  queryKey: ['user', userId],
  queryFn: () => api.get(`/users/${userId}`),
  enabled: !!userId, // N'exécute pas si userId est null
});
```

useMutation — écriture de données

```
import { useMutation, useQueryClient } from '@tanstack/react-query';

function CreateUser() {
  const queryClient = useQueryClient();

  const mutation = useMutation({
    mutationFn: (newUser) => api.post('/users', newUser),
    onSuccess: () => {
      // Invalider le cache pour forcer un refetch
      queryClient.invalidateQueries({ queryKey: ['users'] });
    },
    onError: (error) => {
      console.error("Erreur création :", error);
    },
  });

  return (
    <button
      onClick={() => mutation.mutate({ nom: "Hugo" })}
      disabled={mutation.isPending}
    >
      {mutation.isPending ? "Création..." : "Créer"}
    </button>
  );
}
```

Revision #1
Created 29 March 2026 12:57:39 by Hugo
Updated 29 March 2026 12:57:39 by Hugo