

users-rtk-query

✓ 1. Создаём API slice через RTK Query

[src/features/users/usersApi.ts](#)

```
import { createApi, fetchBaseQuery } from "@reduxjs/toolkit/query/react";

export interface User {
  id: number;
  email: string;
  username: string;
  name: {
    firstname: string;
    lastname: string;
  };
  phone: string;
  address: {
    city: string;
    street: string;
    number: number;
    zipcode: string;
  };
}

export const usersApi = createApi({
  reducerPath: "usersApi",
  baseQuery: fetchBaseQuery({ baseUrl: "https://fakestoreapi.com/" }),
  endpoints: (builder) => ({
    getUsers: builder.query<User[], void>({
      query: () => "users",
    }),
  }),
});
```

```
export const { useGetUsersQuery } = usersApi;
```

✓ 2. Подключение API в store

`src/app/store.ts`

```
import { configureStore } from "@reduxjs/toolkit";
import { usersApi } from "../features/users/usersApi";

export const store = configureStore({
  reducer: {
    [usersApi.reducerPath]: usersApi.reducer,
  },
  middleware: (getDefaultMiddleware) =>
    getDefaultMiddleware().concat(usersApi.middleware),
});

export type RootState = ReturnType<typeof store.getState>;
export type AppDispatch = typeof store.dispatch;
```

✓ 3. Переписываем компонент UsersList на RTK Query

Было:

- useEffect
- dispatch(fetchUsers())
- useSelector

Станет:

- `const { data, isLoading, error } = useGetUsersQuery();`

src/features/users/UsersList.tsx

```
import { useGetUsersQuery } from "./usersApi";
import styles from "./UsersList.module.css";

export const UsersList = () => {
  const { data: users, isLoading, error } = useGetUsersQuery();

  if (isLoading) return <p className={styles.loading}>Загрузка...</p>;
  if (error)
    return <p className={styles.error}>Ошибка загрузки данных</p>;

  return (
    <div className={styles.grid}>
      {users?.map((user) => (
        <div key={user.id} className={styles.card}>
          <h2 className={styles.name}>
            {user.name.firstname} {user.name.lastname}
          </h2>

          <p className={styles.field}>
            <span>Email:</span> {user.email}
          </p>

          <p className={styles.field}>
            <span>Username:</span> {user.username}
          </p>

          <p className={styles.field}>
            <span>Phone:</span> {user.phone}
          </p>
      ))
    </div>
  );
}
```

```
<div className={styles.address}>
  <span>Address:</span>
  <p>
    {user.address.city}, {user.address.street} {user.address.number}
  </p>
  <p>ZIP: {user.address.zipcode}</p>
</div>
</div>
))}
</div>
);
};
```

✨ RTK Query — преимущества для твоего проекта

- ✗ Нет ручного стейта `loading`, `error`
- ✗ Не нужны thunk-и
- ✗ Нет лишнего slice
- ✗ Нет `useEffect` для загрузки
- ✓ Кэширование
- ✓ Автоматический рефетч
- ✓ Хуки как `useGetUsersQuery()`
- ✓ Меньше кода и чище структура