

**File: client/src/lib/queryClient.ts**

**Type:** ts | **Size:** 1376 bytes

```
import { QueryClient, QueryFunction } from "@tanstack/react-query";
async function throwErrorIfResNotOk(res: Response) {
  if (!res.ok) {
    const text = (await res.text()) || res.statusText;
    throw new Error(`$${res.status}: ${text}`);
  }
}
export async function apiRequest(
  method: string,
  url: string,
  data?: unknown | undefined,
): Promise<any> {
  const res = await fetch(url, {
    method,
    headers: data ? { "Content-Type": "application/json" } : {},
    body: data ? JSON.stringify(data) : undefined,
    credentials: "include",
  });
  await throwErrorIfResNotOk(res);
  return res;
}
type UnauthorizedBehavior = "returnNull" | "throw";
export const getQueryFn: (options: {
  on401: UnauthorizedBehavior;
}) => QueryFunction =
  ({ on401: unauthorizedBehavior }) =>
  async ({ queryKey }) => {
    const res = await fetch(queryKey[0] as string, {
      credentials: "include",
    });
    if (unauthorizedBehavior === "returnNull" && res.status === 401) {
      return null;
    }
    await throwErrorIfResNotOk(res);
    return await res.json();
  };
export const queryClient = new QueryClient({
  defaultOptions: {
    queries: {
      queryFn: getQueryFn({ on401: "throw" }),
      refetchInterval: false,
      refetchOnWindowFocus: false,
      staleTime: Infinity,
      retry: false,
    },
    mutations: {
      retry: false,
    },
  },
});
```