

# [개발환경] QueryClient, QueryClientProvider 어느 폴더에 설정해야 할까? (page.tsx or layout.tsx)

☰ 구현 어려움 정도	★
☑ 구현 완료	☑
📅 기간	@November 28, 2024
☑ 정리 완료	☑

Next.js에서 React Query를 사용하려면 **QueryClient**, **QueryClientProvider**를 설정해줘야 한다. 새로운 **QueryClient** 인스턴스를 생성하고, **QueryClientProvider**를 통해 앱 전체에서 생성한 **QueryClient**에 접근 가능하도록 해준다.

그렇다면 어느 파일에 설정하는 것이 좋을까?

## 1. layout.tsx에 설정하는 경우

### 장점

- 애플리케이션의 모든 페이지에서 동일한 QueryClient를 공유하여 데이터를 전역적으로 관리할 수 있다.
- 각 페이지에서 QueryClientProvider를 반복해서 설정할 필요가 없어 코드가 간결해진다.

### 단점

- 모든 페이지에서 동일한 QueryClient를 사용해야 하므로, 페이지별 다른 데이터 패칭 설정이 어려울 수 있음
- 불필요한 데이터를 모든 페이지에서 관리하게 되면 메모리 사용량이 증가하여 성능 저하 가능성이 있음

## 2. page.tsx에 설정하는 경우

### 장점

- 각 페이지에서 독립적인 QueryClient를 사용해 유연성 있게 관리 가능
- 메모리 사용량을 줄일 수 있음
- 페이지별 다른 패칭 설정 가능

### 단점

- 필요한 곳마다 반복 설정해야함

저는 모든 페이지에서 동일한 데이터를 사용, 공유할 것이기 때문에 layout.tsx에 설정해주도록 하였습니다. (일반적으로 layout.tsx에 사용된다고 합니다.)

참고로 서버사이드렌더링(SSR)을 사용하는 경우 공통으로 사용할 데이터를 미리 가져와 패칭해야 하므로 layout.tsx에 사용하는 것이 좋다고 합니다.

참고로 대규모 애플리케이션이나 데이터 보안이 중요한 경우 page.tsx에도 설정한다고 합니다!

```
// src/utils/ReactQueryProviders.tsx

'use client';
import { QueryClientProvider, QueryClient } from '@tanstack/react-query';
import { ReactQueryDevtools } from '@tanstack/react-query-devtools';
import { useState } from 'react';

export default function ReactQueryProviders({
  children,
}): React.PropsWithChildren {
  const [client] = useState(
    new QueryClient({
      defaultOptions: {
        queries: {

```

```

    refetchOnWindowFocus: false, // 윈도우가 다시 포커스되었을때 데이터를 refetch
    refetchOnMount: false, // 데이터가 stale 상태이면 컴포넌트가 마운트될 때 refetch
    retry: 1, // API 요청 실패시 재시도 하는 옵션 (설정값 만큼 재시도)
  },
},
}),
);

return (
  <QueryClientProvider client={client}>
    {children}
    <ReactQueryDevtools initialIsOpen={false} />
  </QueryClientProvider>
);
}

```

```

// src/apps/layout.tsx
...
import ReactQueryProviders from "@/utils/ReactQueryProvider";

export default function RootLayout({
  children,
}: Readonly<{
  children: React.ReactNode;
}>) {
  return (
    <html lang="en">
      <body
        className={` ${geistSans.variable} ${geistMono.variable} antialiased`}
      >
        <ReactQueryProviders>
          {children}
        </ReactQueryProviders>
      </body>
    </html>
  );
}

```

## 참고

<https://velog.io/@rgfdds98/오류-React.jsx-type-is-invalid-expected-a-string-for-built-in-components-or-a-classfunction-for-composite-components-but-got-undefined>

"Error: No QueryClient set, use QueryClientProvider to set one"

I was trying React Query and got this at the start of my React app.

import React from 'react'

🔥 <https://stackoverflow.com/questions/65590195/error-no-queryclient-set-use-queryclientprovider-to-set-one>

