

80. Custom Hook - useGetPostFromUrl()

[#reactjs](#) [#nextjs](#) [#hook](#) [#fronent](#)

- In the **edit** page we will use the same method we used in **post** page to retrieve the **id** of the **post** from the **url**, and pass it into **usePostQuery()** to retrieve the **post** data :

/pages/post/[id].tsx

```
const router = useRouter();
const postId =
  typeof router.query.id === "string" ? parseInt(router.query.id) : -1;
const [{ data, error, fetching }] = usePostQuery({
  pause: postId === -1,
  variables: {
    id: postId,
  },
});
```

- So based on the DIY principle, we extract this code into a **custom hook**:

/utils/useGetPostFromUrl.ts

```
import { useRouter } from "next/router";
import { usePostQuery } from "../generated/graphql";

export const useGetPostFromUrl = () => {
  const router = useRouter();
  const postId =
    typeof router.query.id === "string" ? parseInt(router.query.id) : -1;
  const [{ data, error, fetching }] = usePostQuery({
    pause: postId === -1,
    variables: {
      id: postId,
    },
  });
  return [{ data, error, fetching, postId }];
};
```

- And we use this custom hook both in **post** page and **edit** page

- We also use `postId` in the `edit` page

`/pages/post/[id].tsx`

```
const [{ data, fetching, error }] = useGetPostFromUrl();
```

`/pages/post/edit/[id].tsx`

```
const [{ data, fetching, error, postId }] = useGetPostFromUrl();
```