


47. Page - create-post

#formik #reactjs #nextjs #chakraui #frontend #page

- Here we use the `Layout` component instead of the `Wrapper`, so that the `NavBar` is also displayed
- We *could* handle the "not authenticated" error, here with something like

```
if (error?.message.includes('not authenticated')) {  
  router.push("/login");  
}
```

But then we would have to do it like this for *every* global error (i.e. from the `middleware`) that is returned from `GraphQL`. There's a bit more complicated but much better way to do it which we implement through an `errorExchange` at [48. Global Error Handling with urql](#) 

/pages/create-post.tsx

```
import { Box, Button } from "@chakra-ui/react";  
import { Form, Formik } from "formik";  
import React, { useEffect } from "react";  
import { InputField } from "../components/InputField";  
import { TextAreaField } from "../components/TextAreaField";  
import { useCreatePostMutation } from "../generated/graphql";  
import { useRouter } from "next/router";  
import { withUrqlClient } from "next-urql";  
import { createUrqlClient } from "../utils/createUrqlClient";  
import { Layout } from "../components/Layout";  
  
const CreatePost: React.FC<{}> = ({}) => {  
  const router = useRouter();  
  const [, createPost] = useCreatePostMutation();  
  return (  
    <Layout variant="small">  
      <Formik  
        initialValues={{ title: "", text: "" }}  
        onSubmit={async (values) => {  
          const { error } = await createPost({ input: values });  
          if (!error) {  
            router.push("/");  
          }  
        }}  
      />  
    </Layout>  
  );  
}
```

```

    }
  }}
>
  ({ { isSubmitting }) => (
    <Form>
      <InputField
        name="title"
        label="Title"
        placeholder="Title of the post"
      />
      <Box mt={4}>
        <TextAreaField
          name="text"
          label="Body"
          placeholder="Enter your text"
        />
      </Box>
      <Button type="submit" isLoading={isSubmitting} mt={4} color="teal">
        Create Post
      </Button>
    </Form>
  )}
</Formik>
</Layout>
);
};

export default withUrqlClient(createUrqlClient)(CreatePost);

```