

## 38. Page - change-password

[#formik](#) [#reactjs](#) [#nextjs](#) [#chakraui](#) [#frontend](#) [#page](#)

- We need a **variable** (token) in the **URL** of this page, so that only a user with the token can come here to change password  
i.e. `localhost:3000/change-password/05329837b4-deuh6-rben34-293874yt`
- In **NextJS** the convention for such pages is to create a folder with the page name and place the file inside this folder
- The file name should be `[variableName].tsx` - in our case it will be `[token].tsx`
- **Note that** `setErrors()` of **Formik** works automatically and displays the error to user for the **newPassword** field, but when we have a **token** error we have to display it to user manually

`/change-password/[token].tsx`

```
import { Box, Button, Flex, Link } from "@chakra-ui/react";
import { Formik, Form } from "formik";
import { NextPage } from "next";
import { useRouter } from "next/router";
import { InputField } from "../../components/InputField";
import { Wrapper } from "../../components/Wrapper";
import { toErrorMap } from "../../utils/toErrorMap";
import { useChangePasswordMutation } from "../../generated/graphql";
import { useState } from "react";
import { withUrqlClient } from "next-urql";
import { createUrqlClient } from "../../utils/createUrqlClient";
import NextLink from "next/link";

export const ChangePassword: NextPage<{ token: string }> = ({ token }) => {
  const router = useRouter();
  const [, changePassword] = useChangePasswordMutation();
  const [tokenError, setTokenError] = useState("");

  return (
    <Wrapper variant="small">
      <Formik
        initialValues={{ newPassword: "" }}
        onSubmit={async (values, { setErrors }) => {
          const response = await changePassword({
```

```

        newPassword: values.newPassword,
        token,
    });
    if (response.data?.changePassword.errors) {
        const errorMap = toErrorMap(response.data.changePassword.errors);
        if ("token" in errorMap) {
            setTokenError(errorMap.token);
        }
        setErrors(errorMap);
    } else if (response.data?.changePassword.user) {
        router.push("/");
    }
    }}
>
(({ isSubmitting }) => (
    <Form>
        <InputField
            name="newPassword"
            label="New Password"
            placeholder="Enter your new password"
            type="password"
        />
        {tokenError && (
            <Flex>
                <Box mr={2} color="red">
                    {tokenError}
                </Box>
                <Link as={NextLink} href="/forgot-password">
                    click here to get new token
                </Link>
            </Flex>
        )}
        <Button mt={4} type="submit" isLoading={isSubmitting} color="teal">
            Change Password
        </Button>
    </Form>
)}
</Formik>
</Wrapper>
);

```

```

};

ChangePassword.getInitialProps = ({ query }) => {
  return {
    token: query.token as string, // take the token from the query string section
    of URL and pass it to ResetPassword page as props
  };
};

export default withUrqlClient(createUrqlClient)(ChangePassword);

```

- `NextPage.getInitialProps()` allows us to retrieve props from the URL ( as well as other props) before rendering the page
- Here we retrieve the token from the query string section of the URL:  
localhost:3000/change-password/05329837b4-deuh6-rben34-293874yt
- ⚠ **Note that** for `#optimization` purpose, it's better to retrieve the `token` using `router.query.token` instead of `getInitialProps()` since pages without `getInitialProps()` are **optimized** as **static** pages by **NextJs**