20. Handling errors during register() call

#formik #graphql-codegen #mutation #frontend

- We have the onSubmit() function as follows:

pages/register.tsx

```
<Formik // initalValues, onsubmit, setErros provded by Formik, values is inferred
from initialValues
  initialValues={{ username: "", email: "", password: "" }}
  onSubmit={async (values, { setErrors }) => {
    const response = await register({ options: values });
  }}
```

- Formik provides a setErrors() function that we will use. However it expects a parameter in the following shape:

```
setErrors({
  userName: "hey I'm an error",
})
```

- userName can be any of the initialValues that was defined in <Formik>
- However, if we go deeper into the response objest, the `response.data?.register.errors` has the type FieldError which has the foollowing shape:

```
[{field: "username", message: "hey I'm an error"}]
```

- So we write a utility function to transform this FieldError type into the shape that setErrors() expects
- FieldError type is already defined in the generated GraphQL typescript code so we can easily import it from there

utils/toErrorMap.ts

```
import { FieldError } from "../generated/graphql";

export const toErrorMap = (errors: FieldError[]) => {
  const errorMap: Record<string, string> = {};
  errors.forEach(({ field, message }) => {
    errorMap[field] = message;
  });

  return errorMap;
};
```

- And we update onSubmit() as follows

pages/register.tsx

```
<Formik // initalValues, onSubmit, setErrors provded by Formik, values is
inferred from initialValues
  initialValues={{ username: "", email: "", password: "" }}
  onSubmit={async (values, { setErrors }) => {
    const response = await register({ options: values });
    if (response.data?.register.errors) {
      setErrors(toErrorMap(response.data.register.errors));
    } else if (response.data?.register.user) {
      // register success
      console.log("registration succesful")
    }
  }}
>
```

- Note that we also check `response.data?.register.user` is not undefined to see if registration was successful