

17. Urql for GraphQL requests

#graphql #urql #frontend

- To make GraphQL requests we will use **urql** graphql client
- formidable.com/open-source/urql/

Install packages

```
yarn add urql graphql
```

Initial setup

- Wrap the app with the **urql client**

pages/_app.tsx

```
import { Provider, createClient } from 'urql'

function MyApp({ Component, pageProps }: AppProps) {
  const client = createClient({ url: "http://localhost:4000/graphql" })

  return (
    <Provider value={client}>
      <ChakraProvider theme={theme}>
        <Component {...pageProps} />
      </ChakraProvider>
    </Provider>
  );
}
```

- Copy the **Register mutation** from the **GraphQL playground** (<http://localhost:4000/graphql>) into **useMutation()** in the code
- Update **onSubmit()** inside the **Formik** form

pages/register.tsx

```
import React from "react";
import { Form, Formik } from "formik";
import { Wrapper } from "../components/Wrapper";
import { InputField } from "../components/InputField";
import { useMutation } from "urql";
```

```
const REGISTER_MUT = "mutation Register($username: String!, $email: String!,
$password:String!) {
  register(options: { username: $username, email: $email, password: $password })
{
  errors {
    field
    message
  }
  user {
    id
    username
  }
}
}"
```

```
const Register: React.FC<registerProps> = ({}) => {

  const [, register] = useMutation(REGISTER_MUT);

  return (
    <Wrapper variant="small">
      <Formik // initialValues, onSubmit, setErrors provided by Formik, values is
inferred from initialValues
        initialValues={{ username: '', email: '', password: '' }}
        onSubmit={(values) => {
          return register(values)
        }}
      >
        {
          { isSubmitting } // isSubmitting is provided by Formik
        } => (
          <Form>
            <InputField
              name="username"
              label="Username"
```

```

        placeholder="Username"
      />
      <Box mt={4}>
        <InputField
          name="email"
          label="Email"
          placeholder="Email"
          type="email"
        />
      </Box>
      <Box mt={4}>
        <InputField
          name="password"
          label="Password"
          placeholder="Password"
          type="password"
        />
      </Box>
      <Button mt={4} type="submit" isLoading={isSubmitting} color="teal">
        Register
      </Button>
    </Form>
  )}
</Formik>
</Wrapper>
);
};

```

- Now the register button sends a request to the [GraphQL API](#) which in turn executes the [register mutation](#)

localhost:3000/register

Username

username

Password

password

⌂

⚡

Elements Console Sources Network Performance Memory Application Security Lighthouse Redux »

⌂ 🔍 ⚙️ ⚙️

⌂ 🔍 ⚙️ ⚙️

graphql

Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other Has blocked cookies Blocked Requests

Name

graphql

× Headers Preview Response Initiator Timing

```
{data: {_,_}}
  data: {_,_}
    register: {errors: [{field: "username", message: "length must be greater than 2", __typename: "FieldEr
    errors: [{field: "username", message: "length must be greater than 2", __typename: "FieldError"}]
      0: {field: "username", message: "length must be greater than 2", __typename: "FieldError"}
      user: null
      __typename: "UserResponse"}
```