TypeScript and GraphQL Example

One of the strengths of GraphQL is enforcing data types on runtime. Further, TypeScript and GraphQL Code Generator (graphql-codegen) make it safer by typing data statically, so you can write truly type-protected code with rich IDE assists.

This template extends Apollo Server and Client Example by rewriting in TypeScript and integrating graphql-let, which runs TypeScript React Apollo in graphql-codegen under the hood. It enhances the typed GraphQL use as below:

```
import { useNewsQuery } from './news.graphql'

const News = () => {
    // Typed already⚡
    const { data: { news } } = useNewsQuery()

    return <div>{news.map(...)}</div>
}
```

By default `**/*.graphqls` is recognized as GraphQL schema and `**/*.graphql` as GraphQL documents. If you prefer the other extensions, make sure the settings of the webpack loader in `next.config.js` and `.graphql-let.yml` are consistent.

Deploy your own

Deploy the example using Vercel or preview live with StackBlitz

▲ Deploy

How to use

Execute `create-next-app` with npm or Yarn to bootstrap the example:

```
npx create-next-app --example with-typescript-graphql with-typescript-graphql-app
# or
yarn create next-app --example with-typescript-graphql with-typescript-graphql-app
# or
pnpm create next-app -- --example with-typescript-graphql with-typescript-graphql-app
```

Deploy it to the cloud with Vercel (Documentation).