

CountryList.ts X

```
1  import * as Types from '../../__generated__/types';
2
3  export type CountryListVariables = Types.Exact<{ [key: string]: never }>;
4
5  export type CountryList = {
6    __typename: 'Query';
7    countries?: Types.Maybe<
8      Array<
9        Types.Maybe<{
10          __typename: 'Country';
11          _id?: Types.Maybe<string>;
12          name: string;
13          subregion?: Types.Maybe<{
14            __typename: 'Subregion';
15            _id?: Types.Maybe<string>;
16            name: string;
17          }>;
18        }>
19      >
20    >;
21  };
```

CountryList.tsx X

```
1  import { gql, useQuery } from '@apollo/client';
2  import { Loader, ErrorScreen, CountryCard } from '../components';
3  import { CountryList } from '../__generated__/CountryList';
4
5  const COUNTRY_LIST = gql`
6    query CountryList {
7      Execute Query
8      countries: Country {
9        _id
10       name
11       subregion {
12         _id
13         name
14       }
15     }
16 `;
17
18 const CountriesPage: React.FC = () => {
19   const { data, loading, error } = useQuery<CountryList>(COUNTRY_LIST);
20
21   if (loading) {
22     return <Loader />;
23   }
24
25   if (error) {
26     return <ErrorScreen message={error.message} />;
27   }
28
29   return (
30     <div>
```