

CountryList.ts X

```
85  /**
86   * __useCountryListQuery__
87   *
88   * To run a query within a React component, call `useCountryListQuery` and pass it any options that fit your needs.
89   * When your component renders, `useCountryListQuery` returns an object from Apollo Client that contains loading,
90   * error, and data properties
91   *
92   * you can use to render your UI.
93   *
94   * @param baseOptions options that will be passed into the query, supported options are listed on: https://www.apollographql.com/docs/react/api/react-hooks/#options;
95   *
96   * @example
97   * const { data, loading, error } = useCountryListQuery({
98   *   variables: {
99   *   },
100  * });
101  */
102 export function useCountryListQuery(
103   baseOptions?: Apollo.QueryHookOptions<
104     CountryListQuery,
105     CountryListQueryVariables
106   >,
107 ) {
108   const options = { ...defaultOptions, ...baseOptions };
109   return Apollo.useQuery<CountryListQuery, CountryListQueryVariables>(
110     CountryListDocument,
111     options,
112   );
113 }
114
115 export function useCountryListLazyQuery(
116   baseOptions?: Apollo.LazyQueryHookOptions<
117     CountryListQuery,
118     CountryListQueryVariables
119   >,
120 ) {
121   const options = { ...defaultOptions, ...baseOptions };
122   return Apollo.useLazyQuery<CountryListQuery, CountryListQueryVariables>(
123     CountryListDocument,
124     options,
125   );
126 }
```

CountryList.tsx X

```
1  import { Loader, ErrorScreen, CountryCard } from '../components';
2  import { useCountryListQuery } from './__generated__/CountryList.query';
3
4  const CountriesPage: React.FC = () => {
5    const { data, loading, error } = useCountryListQuery();
6
7    if (loading) {
8      return <Loader />;
9    }
10
11    if (error) {
12      return <ErrorScreen message={error.message} />;
13    }
14
15    return (
16      <div>
17        {data?.countries?.map(
18          country =>
19            country && (
20              <CountryCard
21                key={country._id}
22                name={country.name}
23                region={country.subregion?.name}
24              />
25            ),
26        )}
27      </div>
28    );
29  };
30
31  export default CountriesPage;
```