

.graphqlrc X

```
1  overwrite: true
2  schema: schema.graphql
3  documents: ./src/**/*.tsx
4  extensions:
5    codegen:
6      generates:
7        src/:
8          config:
9            dedupeOperationSuffix: true
10           documentMode: graphQLTag
11           nonOptionalTypename: true
12           omitOperationSuffix: true
13           preResolveTypes: true?
14           withHooks: true
15         plugins:
16           - typescript-operations
17           - typescript-react-apollo
18         preset: near-operation-file
19         presetConfig:
20           baseTypesPath: __generated__/types.ts
21           extension: .ts
22           folder: __generated__
23         src/__generated__/types.ts:
24           config:
25             namingConvention: keep
26           plugins:
27             - typescript
```

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 }
```