n our React Native application, we will be using the same [@apollo/client](https://www.npmjs.com/package/@apollo/client) library as in part 8. Let's get started by installing the library along with the [graphql](https://www.npmjs.com/package/graphql) library which is required as a peer dependency:

npm install @apollo/client graphql

Before we can start using Apollo Client, we will need to slightly configure the Metro bundler so that it handles the *.cjs* file extensions used by the Apollo Client. First, let's install the *@expo/metro-config* package which has the default Metro configuration:

npm install @expo/metro-config

npx expo customize metro.config.js

Then, we can add the following configuration to a *metro.config.js* in the root directory of our project:

const { getDefaultConfig } = require('@expo/metro-config');

const defaultConfig = getDefaultConfig(\_\_dirname);

defaultConfig.resolver.sourceExts.push('cjs');

module.exports = defaultConfig;