Running the Wallet Locally

How to setup the Celo Wallet locally, complete basic wallet functions, and troubleshoot common issues.

Wallet Setup

The app usesReact Native and alight node running thecelo client . You will need to install Java 8, the Android SDK, Yarn, and Node.js v12.x to run the app.

To do this, follow thesetup instructions .

(Optional

) Gradle improvement

This makes Gradle faster:

export

GRADLE_OPTS

'-Dorg.gradle.daemon=true -Dorg.gradle.parallel=true -Dorg.gradle.jvmargs="-Xmx4096m -XX:+HeapDumpOnOutOfMemoryError"'

Running

1. If you haven't already, runyarn
2. from the monorepo root to install dependencies.
3. Attach your device or start an emulated one.
4. You can verify if your device is properly connecting to ADB withadb devices
5. . More information about running the app on Android devices can be found on theReact Native docs
6. .
7. Compile the project and start the bundler with
8. yarn
9. run dev
10. This will build the app in a device (physical or emulated) and open a terminal with a js server.
11. Note:
12. We've seen some issues running the metro bundler from iTerm

Debugging

In order to debug, you should run:

yarn run dev:show-menu A menu will pop-up in the app and you should hitStart Remote JS Debugging . This will open a new tab in your browser with React Native logger in the console. In order to get a full picture, the console's filter should be set toAll levels .

You will probably want to open the dev menu again and enableLive Reloading andHot Reloading to make development faster.

(Optional

) React Native debugger app

TheReact Native debugger app bundles together the Redux and Chrome dev tools nicely.

App Profiling

Start the emulator and load up the app. Then run the following to start react devtools.

yarn run react-devtools It should automatically connect to the running app, and includes a profiler (second tab). Start recorder with the profiler, using the app, and the stop recording. The flame graph provides a view of each component and sub-component. The width is proportional to how long it took to load. If it is grey, it was not re-rendered at that 'commit' or DOM change. Details on the react native profiler arehere . The biggest thing to look for are large number of renders when no state has changed. Reducing renders can be done via pure components in react or overloading the should component

update methodexample here .

Connecting to networks

By default, we have thealfajores network set up. If you have other testnets that you want to use with the app, you can download the contract artifacts as perinstructions in the protocol package . Once you have done that, you can run

yarn run update-contracts --testnets = my_testnet,my_other_testnet --testnets should be a comma-separated list of testnets whose artifacts you have under../protocol/build .update-contracts will copy them to the mobile app for use. Avoid unnecessary check-ins of contract artifacts.

Testing

Snapshot Testing

We use Jest snapshot testing to assert that no intentional changes to the component tree have been made without explicit developer intention. See an example atsrc/components/CodeRow.test.tsx . If your snapshot is expected to deviate, you can update the snapshot with the--updateSnapshot flag when running the test.

Shallow rendering testing

We use Enzyme as an additional testing tool. It allows for shallow rendering, which reduces running time and the amount of stubbing you have to do by only rendering the immediate children (and not the grandchildren etc.) Further, you can interact with the rendered tree to assert proper reactions to user interaction and input. See an example atsrc/components/CodeRow.test.tsx or read more about thedocs .

E2E testing

We useDetox for E2E testing. In order to run the tests locally, you must have the proper emulator set up. Emulator installation instructions can befound here . Please make sure to set up the emulator you would like to run the tests with is set up properly with the namePixel_API_29_AOSP_x86_64 .

Please set123456 as the pin code in the emulator, since the e2e tests rely on that.

Next, the VM snapshot settings should be modified:

1. Close all apps and lock the emulator (go to lock screen).
2. Power off the emulator
3. Power it back on and go to emulator settings (... button) -> Snapshots -> Settings
4. Set Auto-Save to No

Troubleshooting

1. Make sure to have watchman installed:brew install watchman
2. It's a knowndetox issue
3. that test will fail if a emulator is renamed.
4. Make sure the emulator is not running before runing the tests.

For information on how to run and extend the e2e tests, refer to thee2e readme .

Generating GraphQL Types

We're usingGraphQL Code Generator to properly type GraphQL queries. If you make a change to a query, runyarn run gen-types to update the typings in thetypings directory.

Celo Wallet API Keys

Running The Celo App Without API Keys

The mobile app can run without putting the Firebase keys in place by changing the env variableFIREBASE_ENABLED to false in the mobile package.

For Testnet configuration, an env variable calledMNEMONIC should be set with a seed to generate validators private keys. Remember to make this seed long enough to be secured (20 words or more).

Please keep your keys secret- avoid committing them to github and/or publishing keys for your infrastructure in any way.
Edit this page Previous Using the Mobile WalletNext Celo Wallet Testnet Funds