Environment set up

info If you already went through the quick start guide this section follows the same initial steps. You can skip this and go straight to the next section if you want to implement into your existing quick start codebase. We will clone a preconfigured Node.js project with TypeScript support to get started. Follow these steps to clone the repository to your local machine using your preferred command line interface:

1. Open your command line interface, Terminal, Command Prompt, or PowerShell.
2. Navigate to the desired directory where you would like to clone the repository.
3. Execute the following command to clone the repository from the provided GitHub link

git clone git@github.com:bcnmy/quickstart.git Note that this is the ssh example, use http or GithubCli options if you prefer.

git clone https://github.com/bcnmy/quickstart.git Once you have the repository on your local machine - start by installing all dependencies using your preferred package manager. In this tutorial we will use yarn.

yarn install yarn dev After running these two commands you should see the printed statement “Hello World!” in your terminal. Any changes made to theindex.ts file in the src directory should now automatically run in your terminal upon save.

All packages you need for this guide are all configured and installed for you, check out thepackage.json file if you want to explore the dependencies.

Click to learn more about the packages * The account package will help you with creating smart contract accounts and an interface with them to create transactions. * The bundler package helps you with interacting with our bundler or alternatively another bundler of your choice. * The core types package will give us Enums for the proper ChainId we may want to use * The paymaster package works similarly to the bundler package in that you can use our paymaster or any other one of your choice. * The core types package will give us Enums for the proper ChainId we may want to use. * The common package is needed by our accounts package as another dependency. * Finally the ethers package at version 5.7.2 will help us with giving our accounts an owner which will be our own EOA. Let’s first set up a .env file in the root of our project, this will need a Private Key of any Externally Owned Account (EOA) you would like to serve as the owner of the smart account we create. This is a private key you can get from wallets like MetaMask, TrustWallet, Coinbase Wallet etc. All of these wallets will have tutorials on how to export the Private key.

PRIVATE_KEY = "" Let’s give our script the ability to access this environment variable. Delete the console log inside ofsrc/index.ts and replace it with the code below. All of our work for the remainder of the tutorial will be in this file.

import

{ config }

from

"dotenv" ;

config ( ) ; Now our code is configured to access the environment variable as needed. Let's continue in the next section with setting up our smart account Previous Introduction Next Initialize Smart Account