Testing Guide

Contents

- Testnet Markets
- Fork Testing
-
  - Connect to Aave UI
- *
-

Testnet Markets

The simplest way to interact with Aave V3 in a test environment is to connect your wallet and use the app with fauceted funds on a test network. All test markets can be accessed from theapp or from cloning/forking theAave frontend .

To interact with test markets make sure you toggle theTestnet mode on . The testnet option is available onapp in top right ⚙ drop-down-menu. First, you'll need to add the test network to your wallet. You can directly add networks to your browser wallet withChainlist .

Next, you'll need to faucet some base currency for the test network to pay gas for transactions.

Each testnet market has a custom set of assets which can be fauceted from the Aave faucet. To access the faucet interface: switch to the market which you want to test, be sure your wallet is connected to the correct network andtestnet mode is on. The faucet link is available at bottom of the supply column inDashboard or you can manually update url tohttps://app.aave.com/faucet/ .

All Aave V3 testnet markets use a faucet implementation which is open for any address with one limitation: there is a limit of 10000 token per mint tx to prevent testnet tokens from reach maximum totalSupply. Attempting to mint more than 10000 tokens in a mint transaction will cause a revert.

Fork Testing

There are a variety of tools to create a fork of a production network and allow the simulated execution of new transactions. Some prominent tools are Tenderly, Foundry, Hardhat, and Ganache.

Tenderly is a tool for interacting and debugging smart contracts in a browser interface. It can be used to run simulations against live contracts or deploy/test contracts on a network fork with no coding experience required.

Simulation

To run simulations you will need two things, a contract address and contract abi.

The contract addresses for each official and testnet market of the Aave Protocol can be foundhere .

The abi for each contract can obtained from a block explorer or compiling contract code directly.

From a block explorer (etherscan, polygonscan, etc.), if the contract source code is verified you can find the abi by clicking Contract → Read As Proxy → Click the implementation address → Toward the bottom of the page will be an abi that can be copied. Most Aave contracts are deployed as a governance upgradeable proxy which is why the Read As Proxy step is required.

To compile the abi from contract code, from the deployed contracts page, click the link to source code on GitHub and take note of the repo and branch name. You will need to clone the protocol contracts repo and checkout the branch corresponding to the market deployment. Then runnpm i andnpm run compile to compile the contracts and generate abis in theartifacts folder.

Forks

A fork can be created from the Tenderly website which will allow you to deploy your own contracts to a fork of almost any EVM network.

Connect To Aave UI

A Tenderly fork can also be created with simplified steps for connecting to the Aave frontend by cloning thisrepo . Once cloned, you can input your Tenderly API credentials and wallet address into the .env file, then run the commands from theREADME to start a fork on a specified network.

With the output from running the start command you can add fork network to your browser wallet with the chainId and rpc URL provided, then paste the three commands into the browser console of an open tab of the Aave Frontend. Once you reload the page, a new market will be created with an f indicator.

Previous Liquidations Next Governance Guide Last updated6 months ago On this page *Testnet Markets * Fork Testing * Simulation * Forks * Connect To Aave UI

Was this helpful?