

# Getting Started Developing for OP Mainnet

This guide explains the basics of OP Mainnet development. OP Mainnet is [EVM equivalent \(opens in a new tab\)](#), meaning we run a slightly modified version of the same thing you run on mainnet. Therefore, the differences between OP Mainnet development and Ethereum development are minor. But a few differences [do exist](#).

## OP Mainnet and OP Sepolia Endpoint URLs

To access any Ethereum type network you need an endpoint. [These providers](#) support our networks.

### Network Choice

For development purposes we recommend you use either a local development node or [OP Sepolia \(opens in a new tab\)](#). That way you don't need to spend real money. If you need ETH on OP Sepolia for testing purposes, [you can use this faucet \(opens in a new tab\)](#).

## Interacting with Contracts on OP Mainnet or OP Sepolia

We have Hardhat's Greeter contract on OP Sepolia at address [0x9d334aFBa83865E67a9219830ADA57aaA9406681 \(opens in a new tab\)](#). You can verify your development stack configuration by interacting with it.

## Development Stacks

As you can see in the different development stacks below, the way you deploy contracts and interact with them on OP Mainnet or OP Sepolia is almost identical to the way you do it with L1 Ethereum. The most visible difference is that you have to specify a different endpoint (of course). The list of other differences is [here](#).

- [Apeworx \(opens in a new tab\)](#)
- [Brownie \(opens in a new tab\)](#)
- [Foundry \(opens in a new tab\)](#)
- [Hardhat \(opens in a new tab\)](#)
- [Remix \(opens in a new tab\)](#)
- [Truffle \(opens in a new tab\)](#)
- [Waffle \(opens in a new tab\)](#)

## Best Practices

### Use Provided EVM

It is best to start development with the EVM provided by the development stack. Not only is it faster, but such EVMs often have extra features, such as the [ability to log messages from Solidity \(opens in a new tab\)](#) or a [graphical user interface \(opens in a new tab\)](#).

### Debug Before Deploying

After you are done with that development, debug your decentralized application using either [a development node](#) or the [Sepolia test network](#). This lets you debug parts that are OP Mainnet specific such as calls to bridges to transfer ETH or tokens between layers.

Only when you have a version that works well on a test network should you deploy to the production network, where every transaction has a cost.

### Contract Source Verification

You don't have to upload your source code to [block explorers](#), but it is a good idea. On the test network, it lets you issue queries and transactions from the explorer's user interface. On the production network, it lets users know exactly what your contract does, which is conducive to trust.

Just remember, if you use [the Etherscan API \(opens in a new tab\)](#), you need one API key for OP Mainnet and a separate one for OP Sepolia.

[Superchain Dapp Console Differences Between Ethereum and OP Mainnet](#)