

# Advanced Sample Hardhat Project

---

This project demonstrates an advanced Hardhat use case, integrating other tools commonly used alongside Hardhat in the ecosystem.

The project comes with a sample contract, a test for that contract, a sample script that deploys that contract, and an example of a task implementation, which simply lists the available accounts. It also comes with a variety of other tools, preconfigured to work with the project code.

Try running some of the following tasks:

```
npx hardhat accounts
npx hardhat compile
npx hardhat clean
npx hardhat test
npx hardhat node
npx hardhat help
REPORT_GAS=true npx hardhat test
npx hardhat coverage
npx hardhat run scripts/deploy.ts
TS_NODE_FILES=true npx ts-node scripts/deploy.ts
npx eslint '**/*.js,ts'
npx eslint '**/*.js,ts' --fix
npx prettier '**/*.json,sol,md' --check
npx prettier '**/*.json,sol,md' --write
npx solhint 'contracts/**/*.sol'
npx solhint 'contracts/**/*.sol' --fix
```

## Etherscan verification

---

To try out Etherscan verification, you first need to deploy a contract to an Ethereum network that's supported by Etherscan, such as Ropsten.

In this project, copy the .env.example file to a file named .env, and then edit it to fill in the details. Enter your Etherscan API key, your Ropsten node URL (eg from Alchemy), and the private key of the account which will send the deployment transaction. With a valid .env file in place, first deploy your contract:

```
hardhat run --network ropsten scripts/deploy.ts
```

Then, copy the deployment address and paste it in to replace **DEPLOYED\_CONTRACT\_ADDRESS** in this command:

```
npx hardhat verify --network ropsten DEPLOYED_CONTRACT_ADDRESS "Hello, Hardhat!"
```

## Performance optimizations

---

For faster runs of your tests and scripts, consider skipping ts-node's type checking by setting the environment variable `TS_NODE_TRANSPILE_ONLY` to `1` in hardhat's environment. For more details see [the documentation](#).