

# DEX Project with Hardhat

---

## Overview

This project demonstrates a decentralized exchange (DEX) smart contract written in Solidity using the Hardhat development environment. It allows users to buy and sell tokens securely. The project includes the following components:

1. **DEX.sol**: The main decentralized exchange contract.
2. **Token.sol**: A simple ERC20 token contract.
3. **DEXTESTT.js**: A test script for the DEX functionality.

## Prerequisites

Before starting, ensure you have the following installed:

- [Node.js](#) (v14.x or higher)
- [Hardhat](#)
- A package manager like [npm](#) or [yarn](#)

## Setup

### Step 1: Clone the Repository

```
git clone <repository-url>  
cd <repository-directory>
```

### Step 2: Install Dependencies

```
npm install
```

This will install the required dependencies, including OpenZeppelin contracts.

### Step 3: Run Tests

To test the DEX functionality, run:

```
npx hardhat test
```

Ensure **DEXTESTT.js** is properly configured to test all scenarios.

## Files

## Contracts

### contracts/DEX.sol

- A DEX smart contract allowing users to:
  - Buy and sell tokens.
  - Withdraw tokens and funds (restricted to the owner).
  - Retrieve the price and token balance.

### contracts/Token.sol

- A simple ERC20 token contract named "Kibo" (`symbol: MRAF`) with an initial supply.

## Tests

### test/DEXTESTT.js

- A JavaScript file for testing the DEX contract using Hardhat's testing framework and Mocha/Chai assertions.

## Scripts

Run the deployment script:

```
npx hardhat run scripts/deploy.js --network <network-name>
```

## Testing

Ensure `test/DEX.js` tests the following:

```
npx hardhat test --verbose
```

1. Token creation and minting.
2. Buying and selling tokens.
3. Withdraw functionality for tokens and funds.
4. Price calculations.