

DOP: / / 2023

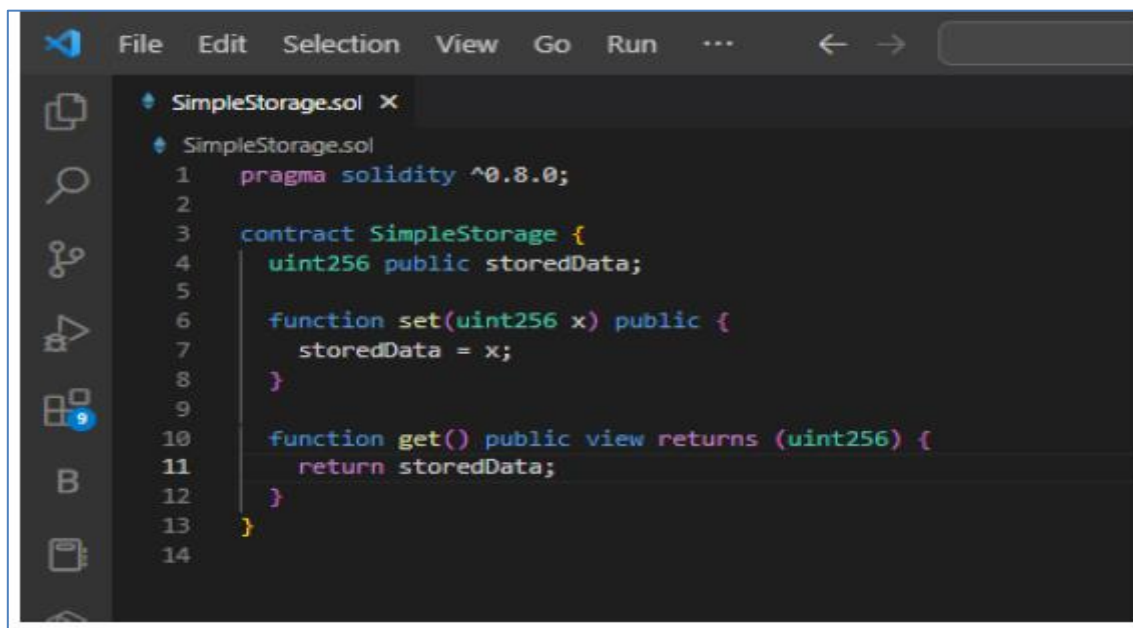
DOS: / / 2023

Experiment No: 01

Aim: - To develop and deploy smart contracts on local Blockchain using solidity programming language.

Theory:

1. Create a new directory for your project and navigate to it in your terminal.
2. Create a new Solidity file called SimpleStorage.sol with the following code:



```
File Edit Selection View Go Run ... < >
SimpleStorage.sol x
SimpleStorage.sol
1  pragma solidity ^0.8.0;
2
3  contract SimpleStorage {
4      uint256 public storedData;
5
6      function set(uint256 x) public {
7          storedData = x;
8      }
9
10     function get() public view returns (uint256) {
11         return storedData;
12     }
13 }
14
```

This contract defines a simple storage system that stores and retrieves a single integer value.

3. Install the required dependencies by running the following command in your terminal:

npm install truffle ganache-cli

```
PS C:\Users\DELL\OneDrive\Desktop\Studies\Online clg\SEM 8\Blockchain\Codes\Exp 2> npm install truffle ganache-cli
npm WARN deprecated ganache-cli@6.12.2: ganache-cli is now ganache; visit https://trfl.io/g7 for details
npm WARN deprecated mkdirp-promise@5.0.1: This package is broken and no longer maintained. 'mkdirp' itself supports promises now, please switch to that.
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated apollo-datasource@3.3.2: The 'apollo-datasource' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-server-errors@3.3.1: The 'apollo-server-errors' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm WARN deprecated apollo-server-types@3.7.1: The 'apollo-server-types' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.

added 667 packages, and audited 1026 packages in 5m

103 packages are looking for funding
  run `npm fund` for details

4 vulnerabilities (1 moderate, 3 high)

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 9.3.1 -> 9.5.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.5.1
npm notice Run npm install -g npm@9.5.1 to update!
npm notice
PS C:\Users\DELL\OneDrive\Desktop\Studies\Online clg\SEM 8\Blockchain\Codes\Exp 2>
```

This will install Truffle and Ganache CLI, which we'll use to deploy and test our smart contract.

4. Create a new Truffle project by running the following command:

truffle init

```
PS C:\Users\DELL\OneDrive\Desktop\Studies\Online clg\SEM 8\Blockchain\Codes\Exp 2> truffle init
Starting init...
=====
> Copying project files to C:\Users\DELL\OneDrive\Desktop\Studies\Online clg\SEM 8\Blockchain\Codes\Exp 2
Init successful, sweet!

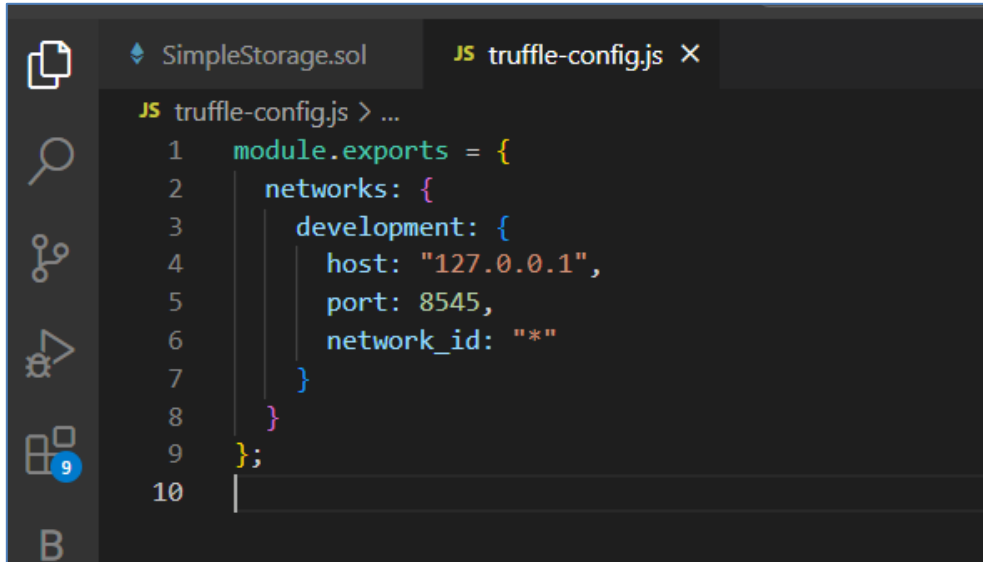
Try our scaffold commands to get started:
$ truffle create contract YourContractName # scaffold a contract
$ truffle create test YourTestName # scaffold a test

http://trufflesuite.com/docs

PS C:\Users\DELL\OneDrive\Desktop\Studies\Online clg\SEM 8\Blockchain\Codes\Exp 2>
```

This will create a basic project structure with some boilerplate code.

5. Configure your blockchain network by editing the truffle-config.js file in the root directory of your project. Add the following code to configure the Ganache CLI network:

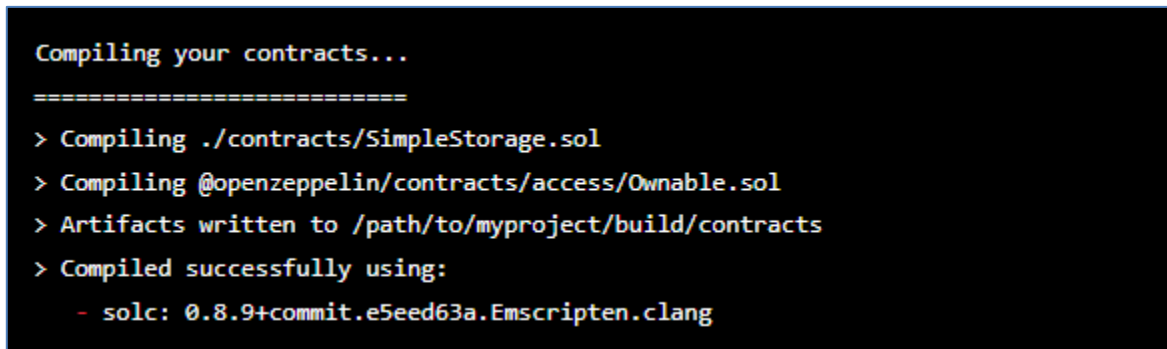


```
JS truffle-config.js > ...
1  module.exports = {
2    networks: {
3      development: {
4        host: "127.0.0.1",
5        port: 8545,
6        network_id: "*"
7      }
8    }
9  };
10
```

This code tells Truffle to use the Ganache CLI network for development, which will run on <http://127.0.0.1:8545>.

6. Compile your smart contract by running the following command:

truffle compile



```
Compiling your contracts...
=====
> Compiling ./contracts/SimpleStorage.sol
> Compiling @openzeppelin/contracts/access/Ownable.sol
> Artifacts written to /path/to/myproject/build/contracts
> Compiled successfully using:
   - solc: 0.8.9+commit.e5eed63a.Emscripten.clang
```

This will compile your Solidity code and generate the corresponding bytecode and ABI files in the build directory.

7. Migrate your smart contract by running the following command:

truffle migrate

```

Compiling your contracts...
=====
> Everything is up to date, there is nothing to compile.

Starting migrations...
=====
> Network name: 'development'
> Network id: 5777
> Block gas limit: 6721975

1_initial_migration.js
=====

Deploying 'Migrations'
-----
> transaction hash: 0x0670db2bbfe9b9a8eb2dfe7d356dbf00a878b71c4d4aa4b4aaf8382
> Blocks: 0 Seconds: 0
> contract address: 0x81993c28535d16819bc6c24861d7e2a2ca93e39d
> block number: 1
> block timestamp: 1646059215
> account: 0x5d2ea8a699aC097bC37d9Df6c8C6BfC2c07B6f29
> balance: 1000.002972 ETH
> gas used: 191162 (0x2e47a)
> gas price: 20 gwei
> value sent: 0 ETH
> total cost: 0.00382324 ETH

> Saving migration to chain.
-----
> Total cost: 0.00382324 ETH

2_deploy_simple_storage.js
=====

```

This will deploy your smart contract to the Ganache CLI network. You can interact with your smart contract by running the following commands:

truffle console

8. This will open the Truffle console, which provides a JavaScript environment for interacting with your smart contract.

let instance = await SimpleStorage.deployed()

```

PS C:\Users\DELL\OneDrive\Desktop\Studies\Online clg\SEM 8\Blockchain\Codes\Exp
2> truffle console
truffle(development)> let instance = await SimpleStorage.deployed()

```

undefined

9. This will create an instance of your deployed smart contract.

await instance.set(42)

```
await simpleStorage.set(42)
```

```
let value = await instance.get()
```

```
let result = await simpleStorage.get()
```

PRIYUSH B. KHOBRAGADE -52