Truffle

Truffle is a blockchain development framework dedicated to Ethereum blockchain platform. This open-source framework was developed by Consensus with an objective to simplify the blockchain and DApp development in Ethereum plat- form. Truffle is a well-equipped framework which makes Ethereum platform more user-friendly and interactive. Following features of truffle make it the best choice for developments in Ethereum.

Features of Truffle

1) Automated tools • Truffle provides built-in smart contract creation tools, which will perform com- pilation, linking, deployment and binary management of the smart contracts. It offers a development environment with a testing framework and digital asset pipeline. The truffle reduces the complexities of team-based development, testing, deployment, and migration activities in an Ethereum project. In total, smart contract creation and testing which are two tedious tasks in blockchain development have become easier with truffle.

2) Scriptable • Along with those automated tools available, the truffle is scriptable too. Which means the developer is allowed to add more scripts to the project during the development. For running these scripts a 'script runner' is also available in truffle, which can run both external and internal scripts.

3) Networking • As the blockchain is growing day by day extensions and migrations have become a frequent requirement in the development environment. The truffle framework is fully capable of integrating an infinite number of networks to it and will completely support migrations and extensions of networks. The configurable build pipeline in truffle supports tight integration within the network.

4) Package support • Truffle provides two package management tools, Ethpm & Npm. Ethpm is the Ethereum Package Manager and Npm is Node Package Manager These package includes several modules and set of predefined codes which can be utilized while scripting.

5) User interaction • Truffle has an interactive console for the direct contract communication. Through this console, the developer can create, compile and test the smart contracts. The console also manages the communications between the user and smart contracts. Another interaction environment available in truffle is the browser portal. The developer can use the default account in truffle to use the browser environment. It is possible to see the local transactions, pending requests etc. from there.

Development-Truffle boxes

Before installing a truffle environment owe must install an Ethereum client like 'Geth', 'WebThree', Parity' or any other. Truffle is installed as a separate project above Ethereum client and it can be deployed on it without any extra configuration.

After installing the Ethereum client, create a project directory for truffle using the command

mkdir projectfolder'

To initialize the project (i.e. truffle) navigate to the created directory

cd projectfolder

And run following command.

init truffle

If the initialization is complete then the following project structure will be created

1. Contracts
2. Migrations
3. test
4. truffle.js

By default, there will be some sample contracts and set of sample projects in truffle environment to get acquainted with the truffle environment

Truffle Box

In truffle, each individual project is called truffle boxes. The truffle box will contain required modules, front end views, solidity smart contract libraries of a project etc..

As mentioned earlier, Truffle comes with some sample Truffle Boxes available. The user can unbox and run them in the test environment.

Creating a Truffle Box

Users can create their own truffle boxes in the truffle environment. They can either create truffle from scratch or can add an existing project to the box and develop from it. If the project is starting from scratch a 'blueprint truffle box' is readily available in the truffle. The 'blueprint truffle box' will contain all the necessary configuration files and common values for a truffle box.

For developing truffle boxes the user need a

1. Github repository,

2. Configuration file,

3. Optionally small and large images for the boxes listing.

The truffle configuration file name is truffle-box.json. It contains 3 attributes ignore, commands and hooks

Ignore: An array attribute contains the list of files to be ignored while unboxing.

Commands: Object attributes of a key-value pair, contains the list of files to be compiled or migrated. After unboxing these files are visible to users.

Hooks: Object attribute which contains a list of commands need to be executed when unboxing. The blueprint contains all the basic components needed for developing a truffle box. The developer can delete the default sample files and images from the box and can create new files. The configuration file is also customizable.

Community truffle box

Truffle is already providing several official truffle boxes developed by the truffle developers. Along with that a vibrant truffle community is also actively contributing new truffle boxes. Any individual can contribute to the community by sending the developed truffle details and GitHub repository details to truffle community. The Box will undergo a screening and if it is compatible with truffle then it will be published as a community truffle box