Web3 Js and IPFS

Hypermine
TECHNOLOGIES

Overview

- DApps needs to connect to Eth nodes over RPC
- Libraries available
  - Web3 js : Javascript library
  - Web3J : java library
  - **n**ethereum : .Net library
  - Web3.py : Python library

Hypermine
TECHNOLOGIES

Web3 Js Overview

Exposes multiple APIs for connecting and using Ethereum network over RPC

- **eth :** Ethereum blockchain related methods
- **net :** Node's network related
- **personal :** Account related
- **db** : Level DB related methods
- **shh** : P2P messaging using Wishper

Steps to interact with contracts

1. Install Web3 js
   - ***npm install web3 --save***

2. Check if web3 object is already there, if not create one using HttpProvider

```
if (typeof web3 !== 'undefined') {
    web3 = new Web3(web3.currentProvider);
} else {
    // set the provider you want from Web3.providers
    web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));
}
```

Hypermine
TECHNOLOGIES

Steps to interact with contracts

3. Specify the default ethereum account to use to

   - ***web3.eth.defaultAccount = web.eth.accounts[0];***

4. Create contract object using ABI

   - ***var contractObj= web3.eth.contract(<paste ABI here>);***

5. Create contract instance using contract address

   - ***var contractInstance = contractObj.at(<Contract address>);***

Hypermine
TECHNOLOGIES

Steps to interact with contracts

6. Now you can call contract methods

```
contractInstance.setterMethod(param1, param2, {
  from : '<address of account>',
  gas : 40000
}).then(function(result){
  //callback
  console.log('Transaction Id = '+ result)
})
```

Hypermine
TECHNOLOGIES

IPFS

- Inter Planetary File System
- Peer to Peer distributed file system
- Protocol to create permanent and decentralize method of storing files
- Like any one run Ethereum, any one can run IPFS node and join the IPFS network to form global file system
- Files are replicated across many node
- No chances of losing them

Hypermine
TECHNOLOGIES

Steps to use IPFS

1. Installation
- ***npm install --save ipfs-api***

2. Run the IPFS daemon (see the supported doc for details)

3. Upload files using ipfs object (see the next slide)

Hypermine
TECHNOLOGIES

Steps to use IPFS

```
//import ipfs
import ipfsAPI from "ipfs-api";
//create ipfs object
const ipfs = ipfsAPI({host: 'localhost', port: '5001', protocol: 'http'})
//creating files object to store in ipfs
const files = [
  {
    path: '<file path>',
    content: '<Buffer or Readable stream>'
  }
];
//call ipfs add function to add files to ipfs
ipfs.files.add(files, function(err, files) {
  //print the hash
  console.log('File hash : ' + files[0].hash)
});
```

Hypermine
TECHNOLOGIES

References

Web3 Js API

Web3.net

Web3.eth

Web3.shh

Web3.db

Web3 Js Doc

IPFS Doc

IPFS API

Hypermine
TECHNOLOGIES