

**CSED490U Blockchain & Cryptocurrency**

**Assignment 9**



**Submitted by- Sajan Maharjan**

**POVIS id- [thesajan@postech.ac.kr](mailto:thesajan@postech.ac.kr)**

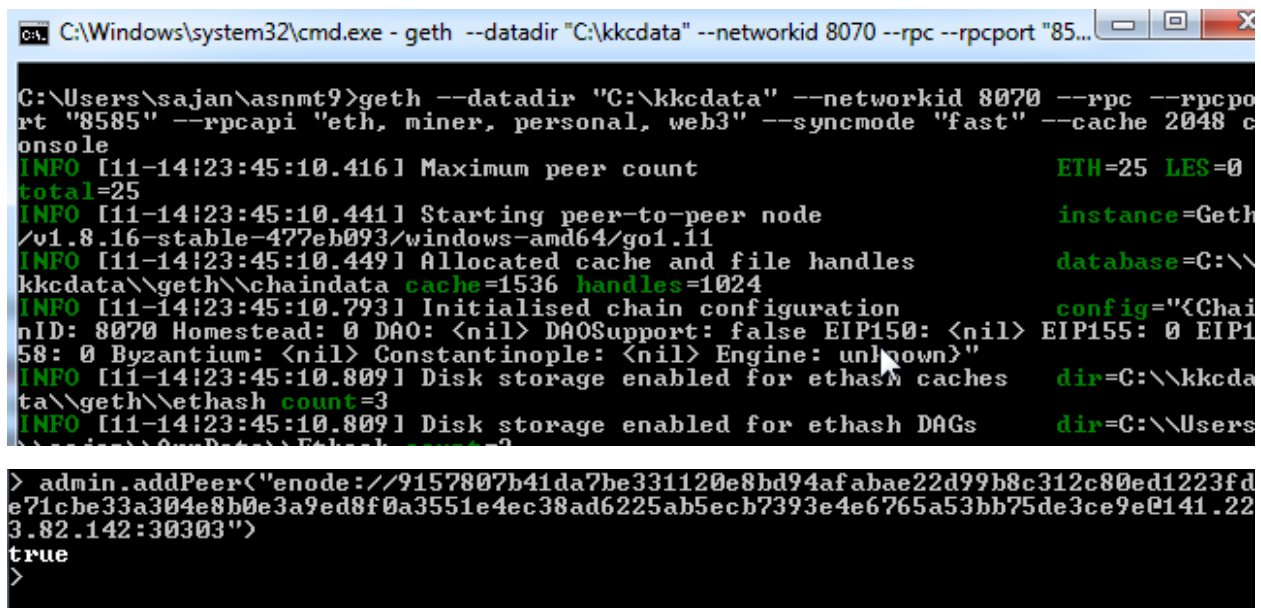
**Registration Number- 20182095**

## Assignment 9: Simple Ethereum Wallet

### <Step 0> Connect to private network

> The blockchain node operating in my computer was connected to the given private network using geth command along with the same genesis file, network id/ chain id and peer was added using admin.addPeers() command.

```
geth --datadir "C:\kkcdata" --networkid 8070 --rpc --rpcport "8585" --rpcapi "eth, miner, personal, web3" console
```



```
C:\Windows\system32\cmd.exe - geth --datadir "C:\kkcdata" --networkid 8070 --rpc --rpcport "8585" --rpcapi "eth, miner, personal, web3" --syncmode "fast" --cache 2048 console
C:\Users\sajan\asnmt9>geth --datadir "C:\kkcdata" --networkid 8070 --rpc --rpcport "8585" --rpcapi "eth, miner, personal, web3" --syncmode "fast" --cache 2048 console
INFO [11-14:23:45:10.416] Maximum peer count ETH=25 LES=0 total=25
INFO [11-14:23:45:10.441] Starting peer-to-peer node instance=Geth/v1.8.16-stable-477eb093/windows-amd64/go1.11
INFO [11-14:23:45:10.449] Allocated cache and file handles database=C:\Users\sajan\AppData\Local\Geth\cache
INFO [11-14:23:45:10.793] Initialised chain configuration config="{ChainID: 8070 Homestead: 0 DAO: <nil> DAOSupport: false EIP150: <nil> EIP155: 0 EIP158: 0 Byzantium: <nil> Constantinople: <nil> Engine: unknown}"
INFO [11-14:23:45:10.809] Disk storage enabled for ethash caches dir=C:\Users\sajan\AppData\Local\Geth\cache count=3
INFO [11-14:23:45:10.809] Disk storage enabled for ethash DAGs dir=C:\Users\sajan\AppData\Local\Geth\cache
> admin.addPeer("enode://9157807b41da7be331120e8bd94afabae22d99b8c312c80ed1223fde71cbe33a304e8b0e3a9ed8f0a3551e4ec38ad6225ab5ech7393e4e6765a53bb75de3ce9e0141.223.82.142:30303")
true
>
```

### <Step 1> Create a simple ethereum wallet

1) Show the code of your simple ethereum wallet and briefly describe it

```
//server.js file

//import necessary libraries
const express = require('express');
const app = express();
const fs = require('fs');
var bodyParser = require('body-parser');
var Web3 = require('web3');
var w3 = new Web3();

//logger for writing transaction hash to a file
```

```

var logger = fs.createWriteStream('txnHashList.txt', {
  flags: 'a' // 'a' means appending (old data will be preserved)
})

//array containing txn hash, which is read from a file and used to listing transactions of wallet
var txnList = [];

//setting web3 object
w3.setProvider(new Web3.providers.HttpProvider('http://localhost:8585'));

//setting for body-parser and ejs engine and adding public folder to our path
app.use(bodyParser.urlencoded({ extended: true }));
app.use(express.static('public'));
app.set('view engine', 'ejs')

//handling GET requests
app.get('/', function (req, res) {
  var accounts = w3.personal.listAccounts;
  console.log(accounts)
  var balArray = [];
  for(var i=0; i < accounts.length; i++) {
    balArray.push(w3.fromWei(w3.eth.getBalance(accounts[i]),'ether'));
  }
  console.log(balArray.toString());
  var addressBalanceMap = {};
  accounts.forEach((key, i) => addressBalanceMap[key] = balArray[i].toString());
  console.log(addressBalanceMap);
  res.render('index', { accounts: accounts, balArray: balArray });
})

//handling POST requests
app.post('/', function (req, res) {
  var accounts = w3.personal.listAccounts;
  console.log(accounts)
  var balArray = [];
  for(var i=0; i < accounts.length; i++) {
    balArray.push(w3.fromWei(w3.eth.getBalance(accounts[i]),'ether'));
  }
  console.log(balArray.toString());
  var addressBalanceMap = {};
  accounts.forEach((key, i) => addressBalanceMap[key] = balArray[i].toString());
  console.log(addressBalanceMap);
  res.render('index', { accounts: accounts, balArray: balArray });
})

//custom defined function to unlock account in the wallet with password
function unlockAccountIfNeeded(account, password) {
  console.log("Account " + account + " is locked. Unlocking ...")
}

```

```

        w3.personal.unlockAccount(account, password, 300);
    }

    //function used for sending ether to another address
    function sendAmt(sender, receiver, amt) {
        var txHash = w3.eth.sendTransaction({
            from: sender,
            to: receiver,
            value: w3.toWei(amt,'ether')
        });
        console.log("Current Transaction Hash: " + txHash);
        logger.write(txHash);
        logger.write("\n");
    }

    //function call made to unlockAccount and senderTransaction after getting form data
    app.post('/processRequest', function(req, res) {
        //print the POST variables in console
        var obj = req.body.transmitter + " " + req.body.recipient + " " + req.body.sendAmount + " " +
        req.body.pwd;
        console.log(obj.toString());

        //store POST variable for send operation
        var sender = req.body.transmitter;
        var receiver = req.body.recipient;
        var amt = parseFloat(req.body.sendAmount);
        var pwd = req.body.pwd;

        unlockAccountIfNeeded(sender, pwd, function(err, res) {
            if(err) {
                console.log("Password Error! Try Again");
                res.send("Password Error! Try Again");
            }
        });
        sendAmt(sender, receiver, amt);
        res.send("Transaction Sent. Wait for block to be mined");
    })

    //handler for 'View Transactions' button
    app.post('/viewTxn', function(req, res) {
        txnList = fs.readFileSync('txnHashList.txt').toString().split("\n");
        console.log(txnList);
        res.render('viewtxn', { txnList : txnList });
    })

    //handler for 'View Transaction Details' button
    app.post('/viewTxnDetails', function(req, res) {
        var selectedTxHash = req.body.selectedTxn
    })

```

```

        var txnDetails = w3.eth.getTransaction(selectedTxHash);
        res.send(txnDetails);
    })

//handler for 'Search blocks' button
app.post('/searchBlock', function(req, res) {
    res.render('block')
})

//handling GET with blockNumber requests
app.get('/block', callBlk);
function callBlk(req, res){
    let number = req.query.blockNumber;
    w3.eth.getBlock(number, function(err, Blk){
        console.log('Block: ' + number + ' is sent ...');
        console.log(Blk);
        res.send(Blk);
    });
}

//running application at PORT 3000
app.listen(3000, function () {
    console.log('Example app listening on port 3000!')
})

```

## 2) Show your simple ethereum wallet's web page

←

→

↺

🏠

localhost:3000

⚙️ Most Visited

🌐 Getting Started

📁 crypto

📁 bigdata

📁 web

### Sajan's Ethereum Wallet

Your List of Accounts

Accounts	Balance
0x66162b35e599fbb0b0d66837c55be33304385d7e	1844.009781044 ETH
0xcd717c241d4c810fa450af9f774933a8084573ff	50.996523864 ETH
0x387ced4c77faad4ce7ff9a68090ca3e75d292d20	96.993695092 ETH
0x75f542c8036aee648df1f0cde919b109c97cd556	23 ETH
<b>Total</b>	<b>2015 ETH</b>

Send Ether?

Transmitter

0x66162b35e599fbb0b0d66837c55be33304385d7e

Recipient

Address to Send

Value

Value to send

ETH

Password

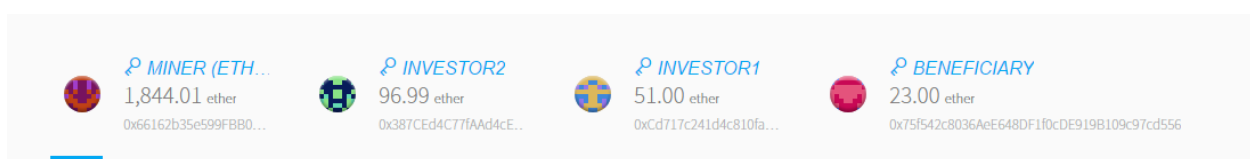
Submit

View Your Transactions

Search By Blocks

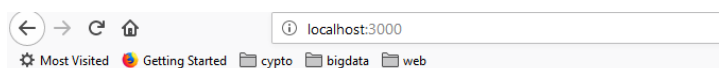
This detail matches with the response from geth command line interface and ethereum wallet as-

```
> eth.accounts
["0x66162b35e599fbb0b0d66837c55be33304385d7e", "0xcd717c241d4c810fa450af9f774933a8084573ff", "0x387ced4c77faad4ce7ff9a68090ca3e75d292d20", "0x75f542c8036aee648df1f0cde919b109c97cd556"]
> eth.getBalance(eth.accounts[0])
1.844009781044e+21
> eth.getBalance(eth.accounts[1])
50996523864000000000
> eth.getBalance(eth.accounts[2])
96993695092000000000
> eth.getBalance(eth.accounts[3])
23000000000000000000
>
```



### 3) Transfer Ether between accounts

> Let us transfer 10 units of ether from the address **0xcd717c241d4c810fa450af9f774933a8084573ff** (currently holding 40.99ETH) to the address **0x75f542c8036aee648df1f0cde919b109c97cd556** (currently holding 43ETH) using our ethereum wallet web application. *The units of ether shown above differs from here due to some test mining and transfer being done before taking snapshots.*



### Sajan's Ethereum Wallet

#### Your List of Accounts

Accounts	Balance
0x66162b35e599fbb0b0d66837c55be33304385d7e	1914.009823044 ETH
0xcd717c241d4c810fa450af9f774933a8084573ff	40.996481864 ETH
0x387ced4c77faad4ce7ff9a68090ca3e75d292d20	96.993695092 ETH
0x75f542c8036aee648df1f0cde919b109c97cd556	43 ETH
Total	2095 ETH

#### Send Ether?

Transmitter

0x66162b35e599fbb0b0d66837c55be33304385d7e

Recipient

0xcd717c241d4c810fa450af9f774933a8084573ff

Value

10

ETH

Password

Submit

View Your Transactions

Search By Blocks

**Sajan's Ethereum Wallet**

**Your List of Accounts**

Accounts	Balance
0x66162b35e599fbb0b0d66837c55be33304385d7e	1914.009823044 ETH
0xcd717c241d4c810fa450af9f774933a8084573ff	40.996481864 ETH
0x387ced4c77faad4ce7ff9a68090ca3e75d292d20	96.993695092 ETH
0x75f542c8036aee648df1f0cde919b109c97cd556	43 ETH
<b>Total</b>	<b>2095 ETH</b>

---

**Send Ether?**

Transmitter: 0xcd717c241d4c810fa450af9f774933a8084573ff

Recipient: 0x75f542c8036aee648df1f0cde919b109c97cd556

Value: 10 ETH

Password: [masked]

---

On successfully entering password, the transaction is submitted to the blockchain and waits for miners to commit, which is shown by the wallet application as-

**Transaction Sent. Wait for block to be mined**

This can be confirmed from the geth terminal which shows new submitted transaction as-

```
> INFO [11-15!00:40:01.845] Submitted transaction fullhash=0x2afc64d4163d00704a26698fcc2844d3bbb0ec11fa143f37432b63ea5a11c6db recipient=0x75f542c8036AeE648DF1f0cDE919B109c97cd556
```

Also, the node console also prints the submitted transaction hash as-

```
'0x75f542c8036aee648df1f0cde919b109c97cd556': '43' }
0xcd717c241d4c810fa450af9f774933a8084573ff 0x75f542c8036aee648df1f0cde919b109c97
cd556 10 123456789
Account 0xcd717c241d4c810fa450af9f774933a8084573ff is locked. Unlocking ...
Current Transaction Hash: 0x2afc64d4163d00704a26698fcc2844d3bbb0ec11fa143f37432b
63ea5a11c6db
```

Now, we start mining and confirm the transfer of ether as-

```
> miner.start()
INFO [11-15!00:31:17.266] Updated mining threads threads=4
INFO [11-15!00:31:17.282] Transaction pool price threshold updated price=1000000
000
nullINF
0 > [11-15!00:31:17.301] Commit new mining work number=406 sea
lhash=f90246.05f786 uncles=0 txs=0 gas=0 fees=0 elapsed=1.000ms
INFO [11-15!00:31:17.316] Commit new mining work number=406 se
alhash=8b233e.6ed244 uncles=0 txs=1 gas=21000 fees=4.2e-05 elapsed=16.001ms
```

⚙ Most Visited 🌐 Getting Started 📁 cypto 📁 bigdata 📁 web





## Sajan's Ethereum Wallet

### Your List of Accounts

Accounts	Balance
0x66162b35e599fb0b0d66837c55be33304385d7e	1949.009865044 ETH
0xcd717c241d4c810fa450af9f774933a8084573ff	30.996439864 ETH
0x387ced4c77faad4ce7ff9a68090ca3e75d292d20	96.993695092 ETH
0x75f542c8036aee648df1f0cde919b109c97cd556	53 ETH
Total	2130 ETH

10 units of ether was deducted from **0xcd717c241d4c810fa450af9f774933a8084573ff** (now holding 30.99ETH) account while corresponding 10 units were added to **0x75f542c8036aee648df1f0cde919b109c97cd556** (now holding 53ETH) through the above transaction.

Accounts are password protected keys that can hold Ether and Ethereum-based tokens. They can control contracts, but can't display incoming transactions.

 <b>MINER (ETH...</b> 1,949.01 ether 0x66162b35e599fb0b0d66837c55be33304385d7e...	 <b>INVESTOR2</b> 96.99 ether 0x387ced4c77faad4ce7ff9a68090ca3e75d292d20...	 <b>BENEFICIARY</b> 53.00 ether 0x75f542c8036aee648df1f0cde919b109c97cd556...	 <b>INVESTOR1</b> 31.00 ether 0xcd717c241d4c810fa450af9f774933a8084573ff...
+ ADD ACCOUNT			



4) Check the transaction that send 10 units of ether

> Our wallet application GUI facilitates checking all the transaction details corresponding to the wallet as-

## Sajan's Ethereum Wallet

**Your List of Accounts**

Accounts	Balance
0x66162b35e599fbb0b0d66837c55be33304385d7e	1949.009865044 ETH
0xcd717c241d4c810fa450af9f774933a8084573ff	30.996439864 ETH
0x387ced4c77faad4ce7ff9a68090ca3e75d292d20	96.993695092 ETH
0x75f542c8036aee648df1f0cde919b109c97cd556	53 ETH
Total	2130 ETH

**Send Ether?**

Transmitter

Recipient

Value  ETH

Password

## Sajan's Ethereum Wallet

**Your List of Transaction**

Transaction Hash
0xf98711298bc2221757f653f65008431816aadcc821131ed2b24bb854754e34aa
0x26fde3dd9bce87635b04c19b94b5c350cc1acbcc975584e5caa3fabfe3bfc645
0x2afc64d4163d00704a26698fcc2844d3bbb0ec11fa143f37432b63ea5a11c6db

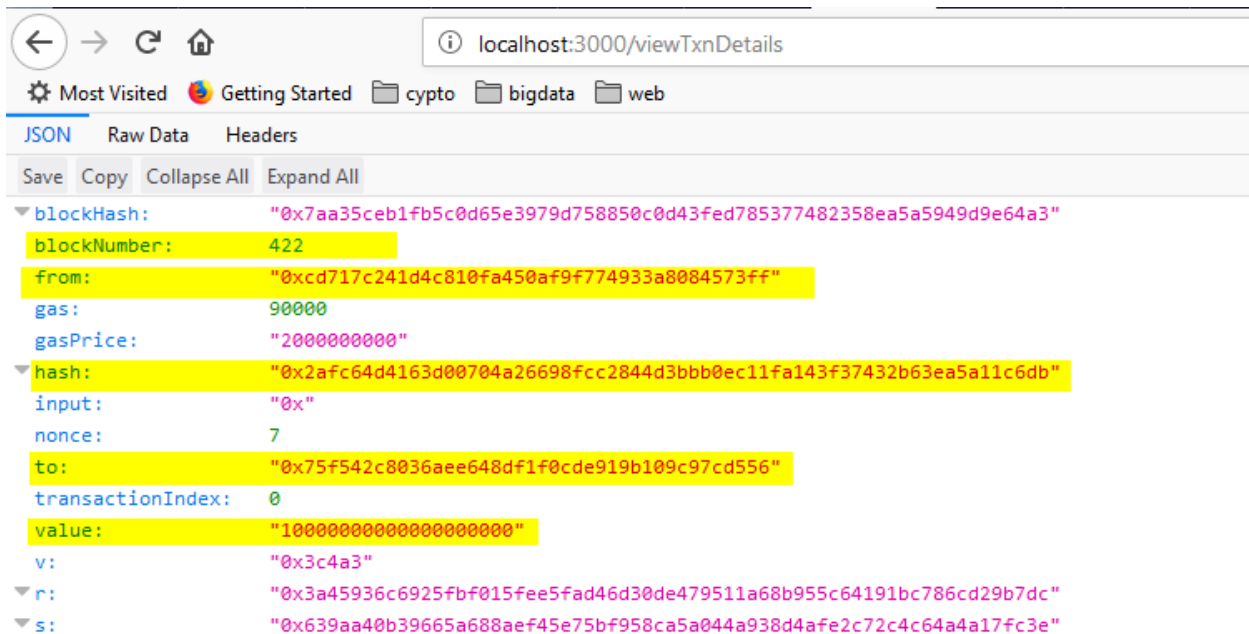
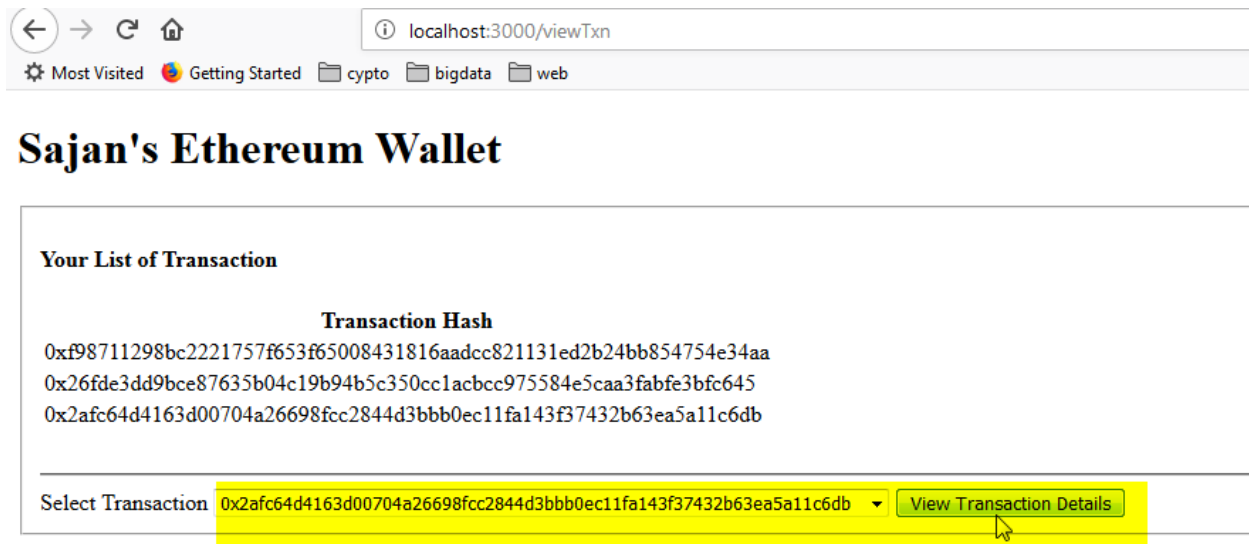
Select Transaction

0xf98711298bc2221757f653f65008431816aadcc821131ed2b24bb854754e34aa

0x26fde3dd9bce87635b04c19b94b5c350cc1acbcc975584e5caa3fabfe3bfc645

0x2afc64d4163d00704a26698fcc2844d3bbb0ec11fa143f37432b63ea5a11c6db

The transaction hash of the last transaction was- *0x2afc64d4163d00704a26698fcc2844d3bbb0ec11fa143f37432b63ea5a11c6db*. We can view the details of this transactions by clicking on View Transaction Details button as-



We can confirm the details of the transactions such as from, to, value, blockNumber containing the transaction and more.

## 5) Check a block that includes the transaction

> We can search for a particular block in our ethereum wallet application by using block number. This is shown as-

localhost:3000

⚙ Most Visited

🚀 Getting Started

📁 crypto

📁 bigdata

📁 web

# Sajan's Ethereum Wallet

### Your List of Accounts

Accounts	Balance
0x66162b35e599fbb0b0d66837c55be33304385d7e	1949.009865044 ETH
0xcd717c241d4c810fa450af9f774933a8084573ff	30.996439864 ETH
0x387ced4c77faad4ce7ff9a68090ca3e75d292d20	96.993695092 ETH
0x75f542c8036aee648df1f0cde919b109c97cd556	53 ETH
Total	2130 ETH

### Send Ether?

Transmitter

0x66162b35e599fbb0b0d66837c55be33304385d7e ▾

Recipient

Address to Send

Value

Value to send

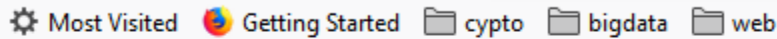
ETH

Password

Submit

View Your Transactions

Search By Blocks



Block Number



This will return the block details such as- transaction merkle root, list of transactions (our previous transaction is also included as shown above), parent block hash and more.