# Overview of Server (Part 2)

## News

1. 2. 3.

## Impostor Syndrome

The syndrome.

## Server: Connect to Ethereum Network

### Accounts

Config Code

```
    URL_WS_8546     string            `json:"geth_ws_8546" default:"ws://127.0.0.1:9
    URL_8545        string            `json:"geth_rpc_8545" default:"http://127.0.0.
    ContractAddress map[string]string `json:"ContractAddress"`
    FromAddress     string            `json:"FromAddress"`
    KeyFilePassword string            `json:"key_file_password" default:"$ENV$Key_Fi
    KeyFile         string            `json:"key_file" default:"$ENV$Key_File"`

    Client    *ethclient.Client `json:"-"` // used in secalling contract
    ClientRPC *rpc.Client       `json:"-"`
    ClientWS  *rpc.Client       `json:"-"`

    AccountKey *keystore.Key `json:"-"`

    ASignedDataContract *SignedDataContract `json:"-"`
```

```
export Key_File=./testdata/UTC--2019-04-03T02-41-09.945205084Z--1d217e902Bc1deB2e75D1Ec
export Key_File_Password=BuPgWKoLOWhuue8p
```

```
        $ . ./:set
        $ ls ./testdata
```

```
UTC--2019-04-03T02-41-09.945205084Z--1d217e902Bc1deB2e75D1Ec44bcAE03A1227a126
```

## KeyFile

```
{
        "address": "1d217e902bc1deb2e75d1ec44bcae03a1227a126",
        "crypto": {
                "cipher": "aes-128-ctr",
                "cipherparams": {
                        "iv": "305d1eb07d717e8933668faeb7d04c43"
                },
                "ciphertext": "6a0c48361bb90c8bbcb33d7b53bef982b6620c7b1e5fd1d1c24457fc
                "kdf": "scrypt",
                "kdfparams": {
                        "dklen": 32,
                        "n": 262144,
                        "p": 1,
                        "r": 8,
                        "salt": "8d27d87b2ec6462fd577f833b72a461965769faef7fd5daf70b0c8
                },
                "mac": "6bbfb5cab3aed19070b7927fccfc62a56452fdc2a1325f70df23ea8c5179438
        },
        "id": "e9c6ccb4-b1e2-45e5-bfca-7d39004cb3f4",
        "version": 3
}
```

To Read a Keyfile in Go

```
                key, err := DecryptKeyFile(gCfg.KeyFile, gCfg.KeyFilePassword)
                ...
                gCfg.AccountKey = key
```

```
// DecryptKeyFile reads in a key file decrypt it with the password.
func DecryptKeyFile(keyFile, password string) (*keystore.Key, error) {
        data, err := ioutil.ReadFile(keyFile)
        if err != nil {
                return nil, fmt.Errorf("Faield to read KeyFile %s [%v]", keyFile, err)
        }
        key, err := keystore.DecryptKey(data, password)
        if err != nil {
                return nil, fmt.Errorf("Decryption error %s [%v]", keyFile, err)
        }
        return key, nil
}
```

```
        transactorOptions := bind.NewKeyedTransactor(
                gCfg.AccountKey.PrivateKey,
                ...
        )
```

## Need accounts you control to test

```
        modifier needMinPayment {
                require(msg.value >= minPayment, "Insufficient payment.  Must send more
                _;
        }

        function init() public {
                minPayment = 1000;
        }
```

## Funds in Account

List of Accounts in Truffle

```
$ truffle develop
Truffle Develop started at http://127.0.0.1:9545/

Accounts:
(0) 0xdc60dc086226238c47ac77b71dfded515146ac0e
(1) 0x942d325062f4597a13449010039e5a24fe1da3f3
(2) 0x1f6d2b271708d72d2fa2c4e3c4e7925ff42d44af
(3) 0x7da51fa8fa25246791e34fe546ec5ee9b8348851
(4) 0xc9eb88a2f224cd440de8edad7702c7235590789a
(5) 0x789eb9c8950f871a90b2e5dfcd21daf90c1d63fe
(6) 0x6a090c8323ec61954bdeddc7d2eef081c7d92b79
(7) 0x2b2ad1becfddd20ccdefaab5e9fd160512d29cf3
(8) 0x28e7f16cc55c6b2f553e28830e62eb693ba630a1
(9) 0xb951f5cf65dd61b9f2fb8b19db95e37afcf5eab2
...
>
```

Unlock an Account

```
> web3.eth.personal.unlockAccount("0xdc60dc086226238c47ac77b71dfded515146ac0e", "passwo
```

Transfer funds to an account that you control (see send_funds.sh)

```
$ curl -H "Content-type: application/json" -X POST --data '{"jsonrpc":"2.0","method":"e
```

Response

```
{"id":"1","jsonrpc":"2.0","result":"0xf1e96688831e4b7f1297dfc9d76c00c0ac950365c79cfa10c
```

## Migrate Contracts

In the directory with `truffle.js`

```
$ truffle migrate --reset
```

What get’s loaded ( file: migrations/2_*.js ) :

```
const SignedData = artifacts.require("./SignedData.sol");
const SignedDataVersion01 = artifacts.require("./SignedDataVersion01.sol");

module.exports = function(deployer) {
  deployer.deploy(SignedDataVersion01)
    .then(function() {
      return deployer.deploy(SignedData, 10000, SignedDataVersion01.address);
    })
        ;
};

*/
```

## Output from Migrations

```
$ truffle migrate --reset

Compiling your contracts...
===========================
> Compiling ./contracts/Migrations.sol
> Compiling ./contracts/SignedData.sol
> Compiling ./contracts/SignedDataVersion01.sol
> Compiling openzeppelin-solidity/contracts/ownership/Ownable.sol
> Artifacts written to /Users/corwin/go/src/github.com/Univ-Wyo-Education/S19-4
> Compiled successfully using:
   - solc: 0.5.0+commit.1d4f565a.Emscripten.clang


Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 0x6691b7


1_initial_migration.js
======================

   Replacing 'Migrations'
   ----------------------
   > transaction hash:    0x650cc1505c627c2eca03175c9ea6c49c6b4a9e9817a27508f14
   > Blocks: 0            Seconds: 0
   > contract address:    0xa9F1FF9d54De3217274d48dBbF1295DCFFae8b07
   > account:             0xDc60Dc086226238C47Ac77b71dFded515146AC0E
   > balance:             99.99430184
   > gas used:            284908
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00569816 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00569816 ETH


2_initial_migration.js
======================

   Replacing 'SignedDataVersion01'
   -------------------------------
   > transaction hash:    0xbf3f6f7ef78e1048b495edf7ed321b8b29414f0184faf96297a
   > Blocks: 0            Seconds: 0
   > contract address:    0x07c82a6245Df224A2D874558b554D37367E46F54
   > account:             0xDc60Dc086226238C47Ac77b71dFded515146AC0E
   > balance:             99.97361672
   > gas used:            992222
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.01984444 ETH


   Replacing 'SignedData'
   ----------------------
   > transaction hash:    0x3b8a840cfa3d84544ef0adba1ecfa38b2b6337c2c6ac0d4f270
   > Blocks: 0            Seconds: 0
   > contract address:    0x7282Fe21EB3f4df40cc0063e16F7934de384510B
   > account:             0xDc60Dc086226238C47Ac77b71dFded515146AC0E
   > balance:             99.95807326
   > gas used:            777173
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.01554346 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:           0.0353879 ETH


Summary
=======
> Total deployments:   3
> Final cost:          0.04108606 ETH
```

You need the address of the contract to call it.

Put that into the `cfg.json` file.

```
{
	"db_flag": "DbFlag,OtherFlag",
	"log_file_name": "./log/log.out",
	"auth_key": "1234",
	"ContractAddress" : {
		"SignedData": "0x72D0f0E20e38CF6f550D5879dE5428f851cD9482"
	},
	"FromAddress": "0xc0c4B94355fD676a29856008e625B51d1acD04eD"
}
```

# Go Code to Call The Contract

```
app := fmt.Sprintf("%x",HashStrings.HashStrings ( "app.signedcontract.com" ))
buf, err := ioutil.ReadFile ( file_name )
...
msgHash, signature := SignMessage ( buf, gCfg.AccountKey )
name := fmt.Sprintf("%x",msgHash)
sig := fmt.Sprintf("%x",signature)
...
tx, err := gCfg.ASignedDataContract.SetData(app, name, sig)
```