Bitcoin (Assignment 1)

Štefan Töltési

Bitcoinj

- library for communicating with Bitcoin protocol
- implemented in Java
- I did not succeed with connecting to regtest deamon

Bitcoin RPC

Full list

Required arguments are denoted inside < and > Optional arguments are inside [and].

Command	Parameters	Description	Requires unlocked wallet? (v0.4.0+)
`addmultisigaddress`	<nrequired> <'["key","key"]'> [account]	Add a nrequired-to-sign multisignature address to the wallet. Each key is a bitcoin address or hex-encoded public key. If [account] is specified, assign address to [account]. Returns a string containing the address.	N
`addnode`	<node> <add/remove/onetry>	**version 0.8** Attempts add or remove <node> from the addnode list or try a connection to <node> once.	N
`backupwallet`	<destination>	Safely copies wallet.dat to destination, which can be a directory or a path with filename.	N
`createmultisig`	<nrequired> <'["key,"key"]'>	Creates a multi-signature address and returns a json object	
`createrawtransaction`	[{"txid":txid,"vout":n},...] {address:amount,...}	**version 0.7** Creates a raw transaction spending given inputs.	N
`decoderawtransaction`	<hex string>	**version 0.7** Produces a human-readable JSON object for a raw transaction.	N
`dumpprivkey`	<bitcoinaddress>	Reveals the private key corresponding to <bitcoinaddress>	Y
`dumpwallet`	<filename>	**version 0.13.0** Exports all wallet private keys to file	Y
`encryptwallet`	<passphrase>	Encrypts the wallet with <passphrase>.	N
`getaccount`	<bitcoinaddress>	Returns the account associated with the given address.	N
`getaccountaddress`	<account>	Returns the current bitcoin address for receiving payments to this account. If <account> does not exist, it will be created along with an associated new address that will be returned.	N

Checking balance

```
public void checkBalance() throws org.json.simple.parser.ParseException, IOException {
    JSONObject json = createJSONObject();
    json.put("method", "getbalance");

    JSONObject responseJsonObj = getResponse(json);

    System.out.println("Your balance is: " + responseJsonObj.get("result"));
}
```

Sending to specific address

```
public void sendToAdress(List<String> params) throws org.json.simple.parser.ParseException, IOException {
    JSONObject json = createJSONObject();
    json.put("method", "sendtoaddress");

    JSONArray paramsArray = new JSONArray();
    paramsArray.addAll(params);
    json.put("params", paramsArray);

    JSONObject responseJsonObj = getResponse(json);

    System.out.println("Your transaction ID is : " + responseJsonObj.get("result"));
}
```

```
switch (words[0]) {
    case "balance":
        client.checkBalance();
        break;
    case "newaddress":
        client.createNewAddress();
        break;
    case "listunspent":
        client.listUnspentTransactions();
        break;
    case "sendto":
        if(words.length != 3){
            System.err.println("You need to specify a recipients address and ammont of bitcoin to be send");
        }else{
            List<String> params = new ArrayList<>();
            params.add(words[1]);
            params.add(words[2]);
            client.sendToAdress(params);
        }
        break;
    default:
        System.err.println("Unknown command");
}
```