1. 下载钱包

<https://bitcoin.org/en/download>

1. bitcoin.conf配置文件
   1. 在安装根目录下新建此文件
   2. 个人当前配置

# 告知 Bitcoin-QT 接受 JSON-RPC 命令

server=1

# JSON-RPC 连接使用的用户名和密码

rpcuser=root

rpcpassword=root

rpcport=8332

rpctimeout=30

# 需要远程访问钱包的时候这个需要配置

rpcbind=0.0.0.0

# rpc访问白名单

rpcallowip=0.0.0.0/0

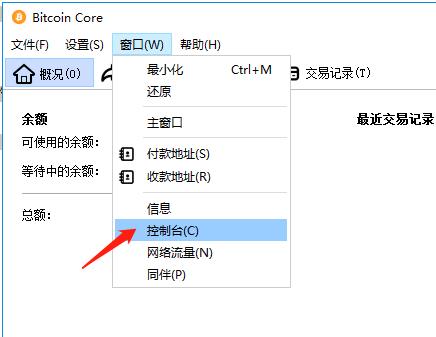
* 1. 其他配置

<https://blog.csdn.net/mongo_node/article/details/89633834>

1. cmd进入安装根目录，以下命令运行bitcoin钱包

bitcoin-qt -conf=E:\Bitcoin\bitcoin.conf

1. 开始下载区块
2. 可以在控制台中输入指令



* 1. help：获取指令

1. 所有指令及其解释

<http://cw.hubwiz.com/card/c/bitcoin-json-rpc-api/>

1. Java调用
   1. 导入json-rpc Jar

<dependency>  
 <groupId>com.github.briandilley.jsonrpc4j</groupId>  
 <artifactId>jsonrpc4j</artifactId>  
 <version>1.1</version>  
</dependency>

* 1. 小demo

package cn.mb.bitcoincore.btc;  
  
import com.alibaba.fastjson.JSONObject;  
import com.googlecode.jsonrpc4j.Base64;  
import com.googlecode.jsonrpc4j.JsonRpcHttpClient;  
  
import java.net.URL;  
import java.util.HashMap;  
import java.util.Map;  
  
*/\*\*  
 \* <p>  
 \* bitcoin-core钱包相关操作  
 \* </p>  
 \*  
 \** ***@author*** *郭海斌  
 \** ***@since*** *2019/9/9  
 \*/*public class BTC {  
  
 public static void main(String[] args) throws Throwable {  
 String username = "root";  
 String password = "root";  
 String url = "http://159.138.39.181:8332/wallet/";  
 String address = "385rbVnXBfcT3oCZvwRjGGf4NqHjQQ5qwE";  
 JsonRpcHttpClient client = *connect*(username, password, url);  
 boolean validateAddress = *validateAddress*(client, address);  
 System.*out*.println("钱包是否有效：" + validateAddress);  
 String addr = *getAddress*(client);  
 System.*out*.println("钱包地址：" + addr);  
 Map<String, Object> map = new HashMap<>();  
 map.put("toAddress", "3P78ES3b59dHK9foacgYhbT3CSRSrJPqnb");  
 map.put("count", 10.5);  
 *tranferMoney*(client, map);  
 }  
  
 */\*\*  
 \* 转账  
 \** ***@param*** *client  
 \** ***@param*** *map 收款地址、金额  
 \** ***@return*** *\** ***@throws*** *Throwable  
 \*/* public static Map<String, Object> tranferMoney(JsonRpcHttpClient client, Map<String, Object> map) throws Throwable {  
 HashMap<String, Object> resultMap = new HashMap<>();  
 String toAddress = (String) map.get("toAddress");  
 // 验证收款地址  
 if(!*validateAddress*(client, toAddress)) {  
 throw new Exception("地址无效，转账失败！");  
 }  
 // 获取转账数量  
 double count = Double.*parseDouble*(map.get("count").toString());  
 try {  
 // 转账金额不能为0  
 if (count <= 0) {  
 throw new Exception("转账金额不能小于或等于0，转账失败！");  
 }  
 // 金额不够或者转0都会报异常，0可以判断，那么异常就都是余额不足的情况  
 String result = client.invoke("sendtoaddress", new Object[]{toAddress, count}, Object.class).toString();  
 System.*out*.println("转账成功，转账结果：" + result);  
 } catch (Throwable e) {  
 throw new Exception("您的余额不足，转账失败！");  
 }  
 return resultMap;  
 }  
  
 */\*\*  
 \* 生成btc钱包地址  
 \** ***@param*** *client  
 \** ***@return*** *\** ***@throws*** *Throwable  
 \*/* public static String getAddress(JsonRpcHttpClient client) throws Throwable {  
 // 直接调用rpc接口即可  
 String result = client.invoke("getnewaddress", new Object[]{}, Object.class).toString();  
 return result;  
 }  
  
 */\*\*  
 \* 验证钱包地址是否有效  
 \** ***@param*** *client rpc客户端  
 \** ***@param*** *address 钱包地址  
 \** ***@return*** *boolean，钱包地址是否有效  
 \** ***@throws*** *Throwable  
 \*/* public static boolean validateAddress(JsonRpcHttpClient client, String address) throws Throwable {  
 // 调用验证钱包地址方法  
 String result = client.invoke("validateaddress", new Object[]{address}, JSONObject.class).toString();  
 System.*out*.println("返回结果：" + result);  
 // 无论传进来的地址是否有效，都会返回一个JSON对象，都包含isvalid字段，可通过此字段判断是否地址有效  
 JSONObject object = JSONObject.*parseObject*(result);  
 return object.getBoolean("isvalid");  
 }  
  
 */\*\*  
 \* 连接bitcoin-core钱包rpc客户端  
 \** ***@param*** *username 账号  
 \** ***@param*** *password 密码  
 \** ***@param*** *url 连接地址  
 \** ***@return*** *rpc客户端  
 \** ***@throws*** *Throwable  
 \*/* public static JsonRpcHttpClient connect(String username, String password, String url) throws Throwable {  
 // 设置账号密码  
 String cred = Base64.*encodeBytes*((username + ":" + password).getBytes());  
 HashMap<String, String> headers = new HashMap<>();  
 // 将账号密码放入json-rpc的头  
 headers.put("Authorization", "Basic " + cred);  
 try {  
 // 指定钱包并连接 - 默认钱包：/wallet/  
 JsonRpcHttpClient client = new JsonRpcHttpClient(new URL(url), headers);  
 // 调用获取区块链信息方法  
 Object result = client.invoke("getblockchaininfo", new Object[]{}, Object.class);  
 System.*out*.println("返回结果：" + result);  
 return client;  
 } catch (Throwable e) {  
 // 若报异常则说明钱包连接失败  
 throw new Exception("btc钱包连接失败...");  
 }  
 }  
  
}

1. Json-rpc协议（包含错误相应码）

<https://www.cnblogs.com/cielosun/p/6762550.html>

1. 总结：
   1. 调用很简单，需求很难搞懂（mmp）
   2. 找的海枯石烂，结果却很easy（mmp\*2）
   3. 接下来的以太坊，搞快点