

```
1  /**
2   * Copyright (c) 2011-2025 libbitcoin developers (see AUTHORS)
3   *
4   * This file is part of libbitcoin.
5   *
6   * This program is free software: you can redistribute it and/or modify
7   * it under the terms of the GNU Affero General Public License as published by
8   * the Free Software Foundation, either version 3 of the License, or
9   * (at your option) any later version.
10  *
11  * This program is distributed in the hope that it will be useful,
12  * but WITHOUT ANY WARRANTY; without even the implied warranty of
13  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
14  * GNU Affero General Public License for more details.
15  *
16  * You should have received a copy of the GNU Affero General Public License
17  * along with this program. If not, see <http://www.gnu.org/licenses/>.
18  */
19 #ifndef LIBBITCOIN_NETWORK_DISTRIBUTORS_DISTRIBUTOR_BITCOIND_HPP
20 #define LIBBITCOIN_NETWORK_DISTRIBUTORS_DISTRIBUTOR_BITCOIND_HPP
21
22 #include <bitcoin/network/define.hpp>
23 #include <bitcoin/network/messages/rpc/rpc.hpp>
24
25 namespace libbitcoin {
26 namespace network {
27 namespace rpc {
28
29 struct bitcoind_methods
30 {
31     static constexpr std::tuple methods
32     {
33         // Blockchain methods.
34         method<"getbestblockhash">{},
35         method<"getblock", string_t, optional<0>>{ "blockhash", "verbosity" },
```

```
36     method<"getblockchaininfo">{},
37     method<"getblockcount">{},
38     method<"getblockfilter", string_t, optional<"basic"_t>>{ "blockhash", "filtertype" },
39     method<"getblockhash", number_t>{ "height" },
40     method<"getblockheader", string_t, optional<true>>{ "blockhash", "verbose" },
41     method<"getblockstats", string_t, optional<empty::array>>{ "hash_or_height", "stats" },
42     method<"getchaintxstats", optional<-1>, optional<""_t>>{ "nblocks", "blockhash" },
43     method<"getchainwork">{},
44     method<"gettxout", string_t, number_t, optional<true>>{ "txid", "n", "include_mempool" },
45     method<"gettxoutsetinfo">{},
46     method<"pruneblockchain", number_t>{ "height" },
47     method<"savemempool">{},
48     method<"scantxoutset", string_t, optional<empty::array>>{ "action", "scanobjects" },
49     method<"verifychain", optional<4>, optional<288>>{ "checklevel", "nblocks" },
50     method<"verifytxoutset", string_t>{ "input_verify_flag" },
51
52     // Control methods.
53     method<"getmemoryinfo", optional<"stats"_t>>{ "mode" },
54     method<"getrpcinfo">{},
55     method<"help", optional<""_t>>{ "command" },
56     method<"logging", optional<""_t>>{ "include" },
57     method<"stop">{},
58     method<"uptime">{},
59
60     // Mining methods.
61     method<"getblocktemplate", optional<empty::object>>{ "template_request" },
62     method<"getmininginfo">{},
63     method<"getnetworkhashps", optional<120>, optional<-1>>{ "nblocks", "height" },
64     method<"prioritisetransaction", string_t, number_t, number_t>{ "txid", "dummy",
        "priority_delta" },
65     method<"submitblock", string_t, optional<""_t>>{ "block", "parameters" },
66
67     // Network methods.
68     method<"addnode", string_t, string_t>{ "node", "command" },
69     method<"clearbanned">{},
```

```
70     method<"disconnectnode", string_t, optional<-1>>{ "address", "nodeid" },
71     method<"getaddednodeinfo", optional<false>, optional<true>, optional<""_t>>{ "include_chain_ ↗
    info", "dns", "addnode" },
72     method<"getconnectioncount">{},
73     method<"getnetworkinfo">{},
74     method<"getpeerinfo">{},
75     method<"listbanned">{},
76     method<"ping">{},
77     method<"setban", string_t, string_t, optional<86400>, optional<false>, optional<""_t>>{ "addr", ↗
    "command", "bantime", "absolute", "reason" },
78     method<"setnetworkactive", boolean_t>{ "state" },
79
80     // Rawtransactions methods.
81     method<"combinerawtransaction", array_t>{ "txs" },
82     method<"createrawtransaction", array_t, object_t, optional<0>, optional<false>>{ "inputs", ↗
    "outputs", "locktime", "replaceable" },
83     method<"decoderawtransaction", string_t>{ "hexstring" },
84     method<"fundrawtransaction", string_t, optional<empty::object>>{ "rawtx", "options" },
85     method<"getrawtransaction", string_t, optional<0>, optional<""_t>>{ "txid", "verbose", ↗
    "blockhash" },
86     method<"sendrawtransaction", string_t, optional<0>>{ "hexstring", "maxfeerate" },
87     method<"signrawtransactionwithkey", string_t, optional<empty::array>, optional<empty::array>, ↗
    optional<"ALL|FORKID"_t>>{ "hexstring", "privkeys", "prevtxs", "sighashtype" },
88     method<"testmempoolaccept", array_t, optional<0>>{ "rawtxs", "maxfeerate" },
89     method<"testrawtransaction", string_t>{ "rawtx" },
90
91     // Util methods (node-related).
92     method<"createmultisig", number_t, array_t>{ "nrequired", "keys" },
93     method<"decodepsbt", string_t>{ "psbt" },
94     method<"decodescript", string_t>{ "hex" },
95     method<"estimatesize", number_t, optional<"unset"_t>>{ "conf_target", "estimate_mode" },
96     method<"getdescriptorinfo", string_t>{ "descriptor" },
97     method<"validateaddress", string_t>{ "address" },
98
99     // Wallet methods (unsupported).
```

```

...tcoin-network\include\bitcoin\network\distributors\distributor_bitcoind.hpp 4
100     method<"abandontransaction", string_t>{ "txid" },
101     method<"addmultisigaddress", number_t, array_t, optional<""_t>, optional<"legacy"_t>> 7
        { "nrequired", "keys", "label", "address_type" },
102     method<"backupwallet", string_t>{ "destination" },
103     method<"bumpfee", string_t, optional<empty::object>>{ "txid", "options" },
104     method<"createwallet", string_t, optional<false>, optional<false>, optional<false>, 7
        optional<""_t>, optional<false>, optional<false>, optional<false>, optional<"set"_t>> 7
        { "wallet_name", "disable_private_keys", "blank", "passphrase", "avoid_reuse", "descriptors", 7
        "load_on_startup", "external_signer", "change_type" },
105     method<"dumpprivkey", string_t>{ "address" },
106     method<"dumpwallet", string_t>{ "filename" },
107     method<"encryptwallet", string_t>{ "passphrase" },
108     method<"getaddressinfo", string_t>{ "address" },
109     method<"getbalances">{},
110     method<"getnewaddress", optional<""_t>, optional<"legacy"_t>>{ "label", "address_type" },
111     method<"getreceivedbyaddress", string_t, optional<0>>{ "address", "minconf" },
112     method<"getreceivedbylabel", string_t, optional<0>>{ "label", "minconf" },
113     method<"gettransaction", string_t, optional<false>, optional<false>>{ "txid", "include_ 7
        watchonly", "verbose" },
114     method<"getunconfirmedbalance">{},
115     method<"getwalletinfo">{},
116     method<"importaddress", string_t, optional<""_t>, optional<false>, optional<true>>{ "address", 7
        "label", "rescan", "p2sh" },
117     method<"importmulti", array_t, optional<empty::object>>{ "requests", "options" },
118     method<"importprivkey", string_t, optional<""_t>, optional<false>>{ "privkey", "label", 7
        "rescan" },
119     method<"importprunedfunds", string_t, string_t>{ "rawtransaction", "txoutproof" },
120     method<"importpubkey", string_t, optional<""_t>, optional<false>>{ "pubkey", "label", "rescan" },
121     method<"importwallet", string_t>{ "filename" },
122     method<"keypoolrefill", optional<100>>{ "newsize" },
123     method<"listaddressgroupings", optional<1>, optional<false>>{ "minconf", "include_watchonly" },
124     method<"listlabels", optional<"receive"_t>>{ "purpose" },
125     method<"listlockunspent">{},
126     method<"listreceivedbyaddress", optional<1>, optional<false>, optional<false>, optional<""_t>> 7
        { "minconf", "include_empty", "include_watchonly", "address_filter" },

```

```

...tcoin-network\include\bitcoin\network\distributors\distributor_bitcoind.hpp 5
127     method<"listreceivedbylabel", optional<1>, optional<false>, optional<false>>{ "minconf", ↗
        "include_empty", "include_watchonly" },
128     method<"listtransactions", optional<""_t>, optional<10>, optional<0>, optional<false>>{ "label", ↗
        "count", "skip", "include_watchonly" },
129     method<"listunspent", optional<1>, optional<empty::array>, optional<true>, optional<false>> ↗
        { "minconf", "addresses", "include_unsafe", "query_options" },
130     method<"loadwallet", string_t, optional<false>>{ "filename", "load_on_startup" },
131     method<"lockunspent", boolean_t, optional<empty::array>>{ "unlock", "transactions" },
132     method<"removeprunedfunds", string_t>{ "txid" },
133     method<"rescanblockchain", optional<0>>{ "start_height" },
134     method<"send", object_t, optional<empty::object>>{ "outputs", "options" },
135     method<"sendmany", string_t, object_t, optional<1>, optional<""_t>, optional<""_t>, ↗
        optional<false>, optional<false>, optional<25>, optional<"unset"_t>, optional<false>, ↗
        optional<0>>{ "dummy", "outputs", "minconf", "comment", "comment_to", "subtractfeefrom", ↗
        "replaceable", "conf_target", "estimate_mode", "avoid_reuse", "fee_rate" },
136     method<"sendtoaddress", string_t, number_t, optional<""_t>, optional<""_t>, optional<false>, ↗
        optional<25>, optional<"unset"_t>, optional<false>, optional<0>, optional<false>>{ "address", ↗
        "amount", "comment", "comment_to", "subtractfeefromamount", "conf_target", "estimate_mode", ↗
        "avoid_reuse", "fee_rate", "verbose" },
137     method<"setlabel", string_t, string_t>{ "address", "label" },
138     method<"settxfee", number_t>{ "amount" },
139     method<"signmessage", string_t, string_t>{ "address", "message" },
140     method<"signmessagewithprivkey", string_t, string_t>{ "privkey", "message" },
141     method<"syncwithvalidationinterfacequeue">{},
142     method<"unloadwallet", optional<""_t>, optional<false>>{ "wallet_name", "load_on_startup" },
143     method<"walletcreatefundedpsbt", optional<empty::array>, optional<empty::object>, optional<0>, ↗
        optional<empty::object>>{ "inputs", "outputs", "locktime", "options" },
144     method<"walletlock">{},
145     method<"walletpassphrase", string_t, number_t>{ "passphrase", "timeout" },
146     method<"walletprocesspsbt", string_t, optional<false>, optional<false>, optional<false>>{ "psbt", ↗
        "sign", "bip32derivs", "complete" }
147 };
148
149 // Derive this from above in c++26 using reflection.
150 // node (supported)

```

```
151     using getbestblockhash = at<0, decltype(methods)>;
152     using getblock = at<1, decltype(methods)>;
153     using getblockchaininfo = at<2, decltype(methods)>;
154     using getblockcount = at<3, decltype(methods)>;
155     using getblockfilter = at<4, decltype(methods)>;
156     using getblockhash = at<5, decltype(methods)>;
157     using getblockheader = at<6, decltype(methods)>;
158     using getblockstats = at<7, decltype(methods)>;
159     using getchaintxstats = at<8, decltype(methods)>;
160     using getchainwork = at<9, decltype(methods)>;
161     using gettxout = at<10, decltype(methods)>;
162     using gettxoutsetinfo = at<11, decltype(methods)>;
163     using pruneblockchain = at<12, decltype(methods)>;
164     using savemempool = at<13, decltype(methods)>;
165     using scantxoutset = at<14, decltype(methods)>;
166     using verifychain = at<15, decltype(methods)>;
167     using verifytxoutset = at<16, decltype(methods)>;
168     using getmemoryinfo = at<17, decltype(methods)>;
169     using getrpcinfo = at<18, decltype(methods)>;
170     using help = at<19, decltype(methods)>;
171     using logging = at<20, decltype(methods)>;
172     using stop = at<21, decltype(methods)>;
173     using uptime = at<22, decltype(methods)>;
174     using getblocktemplate = at<23, decltype(methods)>;
175     using getmininginfo = at<24, decltype(methods)>;
176     using getnetworkhashps = at<25, decltype(methods)>;
177     using prioritisetransaction = at<26, decltype(methods)>;
178     using submitblock = at<27, decltype(methods)>;
179     using addnode = at<28, decltype(methods)>;
180     using clearbanned = at<29, decltype(methods)>;
181     using disconnectnode = at<30, decltype(methods)>;
182     using getaddednodeinfo = at<31, decltype(methods)>;
183     using getconnectioncount = at<32, decltype(methods)>;
184     using getnetworkinfo = at<33, decltype(methods)>;
185     using getpeerinfo = at<34, decltype(methods)>;
```

```
186     using listbanned = at<35, decltype(methods)>;
187     using ping = at<36, decltype(methods)>;
188     using setban = at<37, decltype(methods)>;
189     using setnetworkactive = at<38, decltype(methods)>;
190     using combinerawtransaction = at<39, decltype(methods)>;
191     using createrawtransaction = at<40, decltype(methods)>;
192     using decoderawtransaction = at<41, decltype(methods)>;
193     using fundrawtransaction = at<42, decltype(methods)>;
194     using getrawtransaction = at<43, decltype(methods)>;
195     using sendrawtransaction = at<44, decltype(methods)>;
196     using signrawtransactionwithkey = at<45, decltype(methods)>;
197     using testmempoolaccept = at<46, decltype(methods)>;
198     using testrawtransaction = at<47, decltype(methods)>;
199     using createmultisig = at<48, decltype(methods)>;
200     using decodepsbt = at<49, decltype(methods)>;
201     using decodescript = at<50, decltype(methods)>;
202     using estimatorawfee = at<51, decltype(methods)>;
203     using getdescriptorinfo = at<52, decltype(methods)>;
204     using validateaddress = at<53, decltype(methods)>;
205
206     // wallet (unsupported)
207     using abandontransaction = at<54, decltype(methods)>;
208     using addmultisigaddress = at<55, decltype(methods)>;
209     using backupwallet = at<56, decltype(methods)>;
210     using bumpfee = at<57, decltype(methods)>;
211     using createwallet = at<58, decltype(methods)>;
212     using dumpprivkey = at<59, decltype(methods)>;
213     using dumpwallet = at<60, decltype(methods)>;
214     using encryptwallet = at<61, decltype(methods)>;
215     using getaddressinfo = at<62, decltype(methods)>;
216     using getbalances = at<63, decltype(methods)>;
217     using getnewaddress = at<64, decltype(methods)>;
218     using getreceivedbyaddress = at<65, decltype(methods)>;
219     using getreceivedbylabel = at<66, decltype(methods)>;
220     using gettransaction = at<67, decltype(methods)>;
```

```
221     using getunconfirmedbalance = at<68, decltype(methods)>;
222     using getwalletinfo = at<69, decltype(methods)>;
223     using importaddress = at<70, decltype(methods)>;
224     using importmulti = at<71, decltype(methods)>;
225     using importprivkey = at<72, decltype(methods)>;
226     using importprunedfunds = at<73, decltype(methods)>;
227     using importpubkey = at<74, decltype(methods)>;
228     using importwallet = at<75, decltype(methods)>;
229     using keypoolrefill = at<76, decltype(methods)>;
230     using listaddressgroupings = at<77, decltype(methods)>;
231     using listlabels = at<78, decltype(methods)>;
232     using listlockunspent = at<79, decltype(methods)>;
233     using listreceivedbyaddress = at<80, decltype(methods)>;
234     using listreceivedbylabel = at<81, decltype(methods)>;
235     using listtransactions = at<82, decltype(methods)>;
236     using listunspent = at<83, decltype(methods)>;
237     using loadwallet = at<84, decltype(methods)>;
238     using lockunspent = at<85, decltype(methods)>;
239     using removeprunedfunds = at<86, decltype(methods)>;
240     using rescanblockchain = at<87, decltype(methods)>;
241     using send = at<88, decltype(methods)>;
242     using sendmany = at<89, decltype(methods)>;
243     using sendtoaddress = at<90, decltype(methods)>;
244     using setlabel = at<91, decltype(methods)>;
245     using settxfee = at<92, decltype(methods)>;
246     using signmessage = at<93, decltype(methods)>;
247     using signmessagewithprivkey = at<94, decltype(methods)>;
248     using syncwithvalidationinterfacequeue = at<95, decltype(methods)>;
249     using unloadwallet = at<96, decltype(methods)>;
250     using walletcreatefundedpsbt = at<97, decltype(methods)>;
251     using walletlock = at<98, decltype(methods)>;
252     using walletpassphrase = at<99, decltype(methods)>;
253     using walletprocesspsbt = at<100, decltype(methods)>;
254 };
255
```

```
256 using bitcoind = interface<bitcoind_methods>;
257
258 } // namespace rpc
259 } // namespace network
260 } // namespace libbitcoin
261
262 #endif
263
```