

```
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_ELECTRUM_HPP
20 #define LIBBITCOIN_NETWORK_DISTRIBUTORS_DISTRIBUTOR_ELECTRUM_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 electrum_methods
30 {
31     static constexpr std::tuple methods
32     {
33         // Blockchain methods.
34         method<"blockchain.block.header", number_t, optional<0>>{ "height", "cp_height" },
35         method<"blockchain.block.headers", number_t, number_t, optional<0>>{ "start_height", "count",
```

```
        "cp_height" },
36     method<"blockchain.estimatefee", number_t>{ "number" },
37     method<"blockchain.headers.subscribe">{},
38     method<"blockchain.relayfee">{},
39     method<"blockchain.scripthash.get_balance", string_t>{ "scripthash" },
40     method<"blockchain.scripthash.get_history", string_t>{ "scripthash" },
41     method<"blockchain.scripthash.get_mempool", string_t>{ "scripthash" },
42     method<"blockchain.scripthash.listunspent", string_t>{ "scripthash" },
43     method<"blockchain.scripthash.subscribe", string_t>{ "scripthash" },
44     method<"blockchain.scripthash.unsubscribe", string_t>{ "scripthash" },
45     method<"blockchain.transaction.broadcast", string_t>{ "raw_tx" },
46     method<"blockchain.transaction.get", string_t, optional<false>>{ "tx_hash", "verbose" },
47     method<"blockchain.transaction.get_merkle", string_t, number_t>{ "tx_hash", "height" },
48     method<"blockchain.transaction.id_from_pos", number_t, number_t, optional<false>>{ "height",
        "tx_pos", "merkle" },
49
50     // Server methods.
51     method<"server.add_peer", object_t>{ "features" },
52     method<"server.banner">{},
53     method<"server.donation_address">{},
54     method<"server.features">{},
55     method<"server.peers.subscribe">{},
56     method<"server.ping">{},
57     method<"server.version", optional<"_t>, optional<"1.4"_t>>{ "client_name", "protocol_version" }
58 };
59
60 using blockchain_block_header = at<0, decltype(methods)>;
61 using blockchain_block_headers = at<1, decltype(methods)>;
62 using blockchain_estimatefee = at<2, decltype(methods)>;
63 using blockchain_headers_subscribe = at<3, decltype(methods)>;
64 using blockchain_relayfee = at<4, decltype(methods)>;
65 using blockchain_scripthash_get_balance = at<5, decltype(methods)>;
66 using blockchain_scripthash_get_history = at<6, decltype(methods)>;
67 using blockchain_scripthash_get_mempool = at<7, decltype(methods)>;
68 using blockchain_scripthash_listunspent = at<8, decltype(methods)>;
```

```
69     using blockchain_scripthash_subscribe = at<9, decltype(methods)>;
70     using blockchain_scripthash_unsubscribe = at<10, decltype(methods)>;
71     using blockchain_transaction_broadcast = at<11, decltype(methods)>;
72     using blockchain_transaction_get = at<12, decltype(methods)>;
73     using blockchain_transaction_get_merkle = at<13, decltype(methods)>;
74     using blockchain_transaction_id_from_pos = at<14, decltype(methods)>;
75     using server_add_peer = at<15, decltype(methods)>;
76     using server_banner = at<16, decltype(methods)>;
77     using server_donation_address = at<17, decltype(methods)>;
78     using server_features = at<18, decltype(methods)>;
79     using server_peers_subscribe = at<19, decltype(methods)>;
80     using server_ping = at<20, decltype(methods)>;
81     using server_version = at<21, decltype(methods)>;
82 };
83
84 using electrum = interface<electrum_methods>;
85
86 } // namespace rpc
87 } // namespace network
88 } // namespace libbitcoin
89
90 #endif
91
```