**\libraries\app\application_impl.hxx**

```
#pragma once

#include <fc/network/http/websocket.hpp>
#include <graphene/app/application.hpp>
#include <graphene/app/api_access.hpp>
#include <graphene/chain/genesis_state.hpp>
#include <graphene/chain/protocol/types.hpp>
#include <graphene/net/message.hpp>

namespace graphene { namespace app { namespace detail {
   class application_impl : public net::node_delegate
   {
   public:
      fc::optional<fc::temp_file> _lock_file;
      bool _is_block_producer = false;
      bool _force_validate = false;
      application_options _app_options;

      void reset_p2p_node(const fc::path& data_dir);
      std::vector<fc::ip::endpoint> resolve_string_to_ip_endpoints(const std::string& endpoint_string);
      void new_connection( const fc::http::websocket_connection_ptr& c );
      void reset_websocket_server();
      void reset_websocket_tls_server();

      explicit application_impl(application* self)
          : _self(self),
            _chain_db(std::make_shared<chain::database>())
      ~application_impl()

      void set_dbg_init_key( graphene::chain::genesis_state_type& genesis, const std::string& init_key );
      void startup();
      fc::optional< api_access_info > get_api_access_info(const string& username)const;
      void set_api_access_info(const string& username, api_access_info&& permissions);
      virtual bool has_item(const net::item_id& id) override;
      virtual bool handle_block(const graphene::net::block_message& blk_msg, bool sync_mode,
                                std::vector<fc::uint160_t>& contained_transaction_message_ids) override;
      virtual void handle_transaction(const graphene::net::trx_message& transaction_message) override;
      void handle_message(const graphene::net::message& message_to_process);
      bool is_included_block(const graphene::chain::block_id_type& block_id);
      virtual std::vector<graphene::net::item_hash_t> get_block_ids(const std::vector<graphene::net::item_hash_t>& blockchain_synopsis,
                                                                    uint32_t& remaining_item_count,
                                                                    uint32_t limit) override;
      virtual graphene::net::message get_item(const graphene::net::item_id& id) override;
      virtual graphene::chain::chain_id_type get_chain_id()const override;
      virtual std::vector<graphene::net::item_hash_t> get_blockchain_synopsis(const graphene::net::item_hash_t& reference_point,
                                                                              uint32_t number_of_blocks_after_reference_point) override;
      virtual void sync_status(uint32_t item_type, uint32_t item_count) override;
      virtual void connection_count_changed(uint32_t c) override;
      virtual uint32_t get_block_number(const graphene::net::item_hash_t& block_id) override;
      virtual fc::time_point_sec get_block_time(const graphene::net::item_hash_t& block_id) override;
      virtual graphene::net::item_hash_t get_head_block_id() const override;
      virtual uint32_t estimate_last_known_fork_from_git_revision_timestamp(uint32_t unix_timestamp) const override;
      virtual void error_encountered(const std::string& message, const fc::oexception& error) override;
      uint8_t get_current_block_interval_in_seconds() const override;
      application* _self;
      fc::path _data_dir;
      const boost::program_options::variables_map* _options = nullptr;
      api_access _apiaccess;
      std::shared_ptr<graphene::chain::database>        _chain_db;
```

```
      std::shared_ptr<graphene::net::node>                 _p2p_network;
      std::shared_ptr<fc::http::websocket_server>      _websocket_server;
      std::shared_ptr<fc::http::websocket_tls_server>  _websocket_tls_server;

      std::map<string, std::shared_ptr<abstract_plugin>> _active_plugins;
      std::map<string, std::shared_ptr<abstract_plugin>> _available_plugins;

      bool _is_finished_syncing = false;
   };
}}} // namespace graphene namespace app namespace detail
```

7/9/2018

BitShares Core Release 2.0.180612