# Network Protocol

Both the server and client sides operate on separate threads. When a request is made on the client side, through the Client class, a new socket connection is established to the Server. The Server waits until a socket connection is established, then invokes a request handler on a separate thread. The Client then sends a payload request object, containing the payload object to be sent to the server, a request payload type as an enumerator, an object type stored as a string and a string checksum. The payload request object is then sent to the request handler by an object output stream.

The request handler on the server side receives the stream, reassembles the payload request object and uses the object type string to determine what the object type of the payload object is. The request handler then interfaces with the database through a Hibernate session, and its actions are determined by the request payload type. A “Buy” or “Sell” type request will cause the request handler to cast the request payload object as a Trade class object and create a new entry in the database’s trade table. A “Get” type request will cause the request handler to pass a select query to the Hibernate session to return an object matching the request payload object or null, if there is no matching entry in the database. A “List” type request will cause the request handler to pass a select query to the Hibernate session to return a list of objects matching some property of the payload object, or null if no such entry exists in the database. The “Create”, “Update” and “Delete” type requests will create, update or delete entries, respectively, from the table matching the object type.

Once the request handler has processed the request object, a response object is created and passed back to the client by an object output stream. The request handler socket is then closed and the request handler process ends. The Client reads in the response object, then closes the socket and returns the response.