Protocol Standard for GetStock

Fall 2015 Class Project for CNT 4004

Version 1.5
November 24, 2015

Editors:

Amanda Stephan (astephan@mail.usf.edu)
Ryan Ingram (ryan44@mail.usf.edu)

DOCUMENT HISTORY

1.0 (October 22, 2015) – First version

1.0.1 (October 24, 2015) - Added additional specifications from proposal presentation - sections 1, 2, 4, 5, 6

1.1 October 26, 2015) - Updated message syntax with proposed changes from group discussion - sections 4, 5

1.2 (October 27, 2015) - Changed message syntax to comma separated values terminating with semicolon and added return value for nonexistent stock name – section 5

1.3 (November 3, 2015) - Added additional clarification for the conditions in which return codes would be received, as well as timeout behavior, to section 6.

1.3.1 (November 4, 2015) - Added timing diagrams to section 6.

1.3.2 (November 6, 2015) – Clarified sections 5, 6.

1.3.3 (November 11, 2015) – Editorial changes and minor clean-up by KJC.

1.4 (November 16, 2015) - Appendix added with examples, design criteria added to section 5, syntax and editorial errors corrected.

1.5 (November 18, 2015) – Corrections from comments.

1 - Reordered commands in section 4.2.
2 - Added item 5 to section 5 in Design Criteria.
3 – Corrected ranges for valid user names in section 6.
4 - Reordered commands in section 6 and added note at the end.
5 - Added item 5 to section 6 for multiple error clarification.
6 – Added item 4 to section 6 for port number
7 – Corrected invalid parameters examples in section 8.

PARTICIPANTS

Presenters:

Aziz Batihk
John Casey
Christopher Collazo
Sean Crocker
Kasey Kolyno

Contributors:

Wesam Ali
Jeffrey Barnhart
Aziz Batihk
Ryan Binder
Alexander Boesenberg
Dustin Brinegar
John Casey
Gianpaolo Coletto
Wilson Collado
Christopher Collazo
Christopher Crist
Sean Crocker
David Cruz
John Culp
Rashad Cummings
Mark Farrell
Nicholas Ferraro
Andrew Francis
Edwin Gendron
Jake Hanna
Zachary Houfek
Stephen Hull
Long Huynh
Ryan Ingram
Jeffrey Jerez
Seth Kenney
Kyla Kolb
Kasey Kolyno
Raymond Konopka
Ryan Kurtz
Matthew Lightner
Constance Luong
Zachery Magrini
Bryce Maldonado
Jacob Manfre
Chase Mayhew
Michael Mei
Corbin Moline
Jun Moon
Jon Morgan
Andres Pico Chavez
Charles Zachary Powell
Yashman Rangole
Kristen Rehak
Anthony Rodriguez
Christian Rodriquez
Alejandro Salazar
Vaddanak Seng
Bradley Shiver
Gabriel Sotolongo-Vega
Cody Stanish
Amanda Stephan
Justin Taing
Scottie Thurwanger
Kenneth Trufant
Giovanni Velasquez-Ospina
Brandon Vicinus
Yuriy Zaynulin

TABLE OF CONTENTS

1. SERVICE

The service to be provided is an application layer protocol to get stock values from a server using UDP and the least possible overhead. [1] The protocol also provides registration and unregistration capabilities for users.

2. ASSUMPTIONS

The original assumptions [1] for GetStock are:

1) May only use UDP (TCP and HTTP may not be used).
2) The server and client are reachable from each other.
3) The GetStock server contains a table of stock names and values.

Additional assumptions for GetStock are [2]:

1) Round Trip Time for communications is no greater than 5 seconds.

3. REQUIREMENTS

The original requirements [1] that GetStock must meet are:

1) A user must be able to register with the server using a user name.
   a) If a user attempts to register with a user name already registered, then a suitable error message must be returned.
2) A user must be able to unregister with the server using their user name.
3) A registered user must be able to reliably get a single or multiple stock values from the server.
4) If a registered user requests a stock value for a stock name that does not exist in the server stock table then a suitable error message must be returned.
5) If an unregistered user requests a stock value then a suitable error message must be returned.
6) The GetStock protocol must use the least possible bandwidth within reasonable means.

4. LIST OF MESSAGES

1. Messages from client to server:
   REG for registering a user name
   UNR for unregistering a user name
   QUO to request a list of stock quotes from the server

2) Messages from server to client:
   ROK for request okay
   INC for invalid command
   INP for invalid parameters
   UAE for user already exists
   UNR for user not registered
   INU for invalid user name

5. SYNTAX OF MESSAGES

Design Criteria [2]:

1) Commands shall be 3 characters long from section 4 part 1.
2) Status codes shall be 3 characters long from section 4 part 2.
3) User names will be case insensitive alphanumeric characters with ASCII decimal values in the ranges 48-57, 65-90 and 97-122 and be no longer than 32 characters. Examples in section 8 part 3.
4) All fields shall be comma delimited, terminating in a semicolon.
5) The Server shall return a value of -1 for an invalid quote id request from Client for that stock id only. See example in section 8 part 2.

Client to Server:

Definitions:

command: Message from client to server (list in section 4).

user_name: User name used to register, authenticate request, or unregister.

stock_list: Comma separated list of alphanumeric stock names to get quotes for. Stock names can be from 2-5 characters.

Register User Name (REG):

```
+---------+-----------+
| command | user_name |
+---------+-----------+
  command = REG
```

Unregister User (UNR):

```
+---------+-----------+
| command | user_name |
+---------+-----------+
   command = UNR
```

Request List of Stock Quotes (QUO):

```
+---------+-----------+------------+
| command | user_name | stock_list |
+---------+-----------+------------+
   Command = QUO
```

Server to Client:

Definitions:

status_code: Message from server to client (list in section 4).

quote_list: Comma separated list of ASCII encoded stock quotes for each requested symbol.

Response message for QUO Message:

```
+-------------+------------+
| status_code | quote_list |
+-------------+------------+
```

Response message for REG and UNR Messages:

```
+-------------+
| status_code |
+-------------+
```

6. MESSAGE RULES

Design Criteria:

1) Users (client) remember the order and quantity of requested stocks.
2) Usernames and stock names are case-insensitive.
3) Port 1050 will be used for all messages.
4) In the case of multiple errors in a message, it is undefined which error code (of the applicable error codes for the errors in the message) should be returned. A single valid error code must be returned by the server.

Response code conditions:

ROK for request OK. Delivered when client message syntax is OK for all command types.
INC for invalid command. Delivered in response to any client-to-server command other than the ones defined above
INP for invalid parameters. Delivered when parameters are not sent in comma-separated format, when additional parameters beyond a username are given in REG or UNR commands, or when no stocks are provided in a QUO command.
UAE for user already exists. Delivered in response to a REG request in which the user name provided has already been registered.
UNR for user not registered. Delivered in response to a UNR or QUO command when the provided user_name parameter does not exist on the server.
INU for invalid user name. Delivered in response to a REG command where the user_name parameter is longer than 32 characters or includes non-ASCII characters.

Note: All message must be checked for invalid command (INC) first. If this test passes, check for invalid parameter (INP).

Timing diagrams for message flows:

Successful Send and Replies Between Client and Server:

Register User Name (REG):

```
   Client               Server
     |                    |
     |     <request>      |
     |------------------->|
     |                    |
     |     <response>     |
     |<-------------------|
     |                    |
```

The <request> is a REG command with a valid user name.
The <reponse> can be ROK, UAE or INU based on the conditions.
Examples in section 8 part 1.

Unregister User (UNR):

```
   Client               Server
     |                    |
     |     <request>      |
     |------------------->|
     |                    |
     |     <response>     |
     |<-------------------|
     |                    |
```

The <reponse> can be ROK or UNR based on the conditions.

Request List of Stock Quotes (QUO):

```
   Client              Server
     |                    |
     |     <request>      |
     |------------------->|
     |                    |
     |     <response>     |
     |<-------------------|
     |                    |
```

The <request> is a QUO command followed by a comma separated list of valid stock identifiers.

The <reponse> can be ROK,<quote_list>, INP, or UNR based on the conditions.

Examples in section 8 part 2.

Unsuccessful Send and Replies between Client and Server:

Invalid Command:

```
   Client              Server
     |                    |
     |      <command>     |
     |------------------->|
     |                    |
     |         INC;       |
     |<-------------------|
     |                    |
```

The <command> is anything other that the valid commands listed in section 4.

Timeouts:

```
                  Client              Server
                    |                    |
                    |    <command1>      |
                +-- |---------------x    |
                |   |                    |
        time out|   |                    |
                |   |                    |
                +-- |    <command1>      |
                +-- |------------------->|
                |   |    <response>      |
        time out|   |     x--------------|
                |   |                    |
                +-- |                    |
                    |    <command1>      |
                    |------------------->|
                    |    <response>      |
                    |<-------------------|
                    |                    |
```

The <command1> is any valid commands listed in section 4.
The <response> is any response from the server listed in section 4.
Client will send its request, and wait 5 seconds for a response. If it fails to receive a response within that time frame, it will attempt to resend this request two more times, waiting for 5 seconds in each case.

7. LIST OF REFERENCES

[1] K. Christensen, Project for CNT 4004, Fall 2015. URL: http://www.cse.usf.edu/~christen/class2/project2.html

[2] Aziz Batihk and John Casey, Power Point Presentation, October 2015, URL: http://www.csee.usf.edu/~christen/class2/hidden/pitch2.pdf

8. APPENDIX

Examples

```
   1. Registering User:
       Valid User Name:
         Client                Server
            |                     |
            |    REG,USERNAME     |
            |-------------------->|
            |                     |
            |        ROK;         |
            |<--------------------|
            |                     |

       Invalid User Name:
         Client                Server
            |                     |
            |    REG,USER_NAME    |
            |-------------------->|
            |                     |
            |        INU;         |
            |<--------------------|
            |                     |


2.     Request List of Stock Quotes:
       Valid Request:
         Client                Server
            |                     |
            | QUO,USERNAME,APPL;  |
            |-------------------->|
            |                     |
            |     ROK,121.30;     |
            |<--------------------|
            |                     |


         Client                   Server
            |                        |
            | QUO,USERNAME,APPL,FB;  |
            |----------------------->|
            |                        |
            |   ROK,121.30,94.01;    |
            |<-----------------------|
            |                        |
      Valid & Invalid Request:
         Client               Server
            |                         |
            | QUO,USERNAME,APPL,IBMX; | # IBMX is an invalid stock id
            |------------------------>|
            |                         |
            |      ROK,121.30,-1;     |
            |<------------------------|
            |                         |
```

```
 Invalid User Name:
   Client               Server
      |                    |
      | QUO,USERNAME,APPL; |
      |------------------->|
      |                    |
      |        INU;        |
      |<-------------------|
      |                    |
```

```
 Invalid Command:
   Client               Server
      |                    |
      | QUO USERNAME,APPL; |
      |------------------->|
      |                    |
      |        INC;        |
      |<-------------------|
      |                    |
```

```
   Client               Server
      |                    |
      | REG,USERNAME,APPL; |
      |------------------->|
      |                    |
      |        INP;        |
      |<-------------------|
      |                    |
```

```
 Invalid Command:
   Client                 Server
      |                      |
      | QUOTE,USERNAME,APPL; |
      |--------------------->|
      |                      |
      |        INC;          |
      |<---------------------|
      |                      |
```

3. User Names:

Valid	Invalid
JonhDoe	John Doe
DoeJohn	Doe,John
JDALE	JD@LE
3Pigs	

The following are equivalent

janeDoe = JaneDoe = JANEDOE = janedoe