Calculator Protocol

General

The Calculator protocol used for communication between server and client over TCP using sockets.
The protocol allows the client to

- Authenticate
- Calculate arithmetic operations
- Session termination

The protocol uses length-prefixed format of 4-bytes int for indicating the message size to overcome the gap of socket send/recv reliability.

Each message is prefixed with the length and some header which indicates the server which command the client wishes to execute.

The server response is prefixed with the length and some header which indicates the client the response of the server.

The server and client handle errors in socket, and run time, server also handles calculations errors.

The communication is performed over the TCP-Socket using util.py helper methods which ensure reliable communication over the un-reliable functionality of python-socket functions.

Welcome

Once a socket is accepted the client listens for a welcome message from the server.

The server sends to the client the welcome message “Welcome! Please log in.”

Description (Server -> Client):

Value	data_length	data
Size(bytes)	4	data_length

data_length: the length of the data will be sent: 4 bytes unsigned int

data: the data to transmit, not including data_length, UTF-8

Example:

No cli usage example – server functionality

Send message example:

<length><data>

00000016Welcome! Please log in.
(length in hex)

Authentication

Request:
The client sends the server authentication request with information

Description (Client -> Server):

Value	data_length	header	data
Size(bytes)	4	4	data_length

data_length: the length of the data and header will be sent: 4 bytes

header: the string “AUTH” so the server can parse and handle.

data: the data to transmit, not including data_length and header, UTF-8

Example:

User: Bob

Password: simplepass

Send message example:

<length><header><data>

00000024AUTH User: user1 Password: password

Response:

The server sends the client authentication response

<u>Success</u>

Description (Server -> Client):

Value	data_length	header	data
Size(bytes)	4	3	data_length

data_length: the length of the data and header will be sent: 4 bytes

header: the string “SUC” so the client can parse and handle.

data: the message to print

Example:

No cli usage example – server functionality

Send message example:

<length><header><data>

000000##SUC Hi {username_of_user}, good to see you.

(# is some number calculated in runtime – for convenience)

Failure

Description (Server -> Client):

Value	data_length	header	data
Size(bytes)	4	3	int(data_length)

data_length: the length of the data and header will be sent: 4 bytes

header: the string “FLR” so the client can parse and handle.

data: the message to print

Example:

No cli usage example – server functionality

Send message example:

<length><header><data>

000000##FLR Failed to login.

Error

Description (Server -> Client):

Server tells the client an error has occurred and session is terminated.

Value	data_length	header
Size(bytes)	4	3

data_length: the length of the header will be sent: 4 bytes

header: the string “ERR” so the client can parse and handle.

Example:

No cli usage example – server functionality

Send message example:

<length><header>

00000003ERR

Calculation

The client sends the server calculation request

Description (Client -> Server):

Value	data_length	header	type	X	Y	Z
Size(bytes)	4	3	calculate:	4	1	4

data_length: the length of the data and header will be sent: 4 bytes

header: the string “CLC” so the server can parse and handle.

“calculate:”: the command from the user

X, Y, Z: operand, operator, operand

Example:

calculate: 4 + 3

Send message example:

<length><header><calculate:><operand> <operation> <operand>

00000000CLCcalculate: 4 + 3

The server sends the client calculation response

Success

Description (Server -> Client):

Value	data_length	header	R
Size(bytes)	4	3	8

data_length: the length of the data and header will be sent: 4 bytes

header: the string “RES” so the client can parse and handle.

R: flaot

Example:

No cli example

Send message example:

<length> <header> <R>

00000000RES 7

The server sends the client calculation error

Failure

Description (Server -> Client):

Value	data_length	header	message
Size(bytes)	4	3	25

data_length: the length of the data will be sent: 4 bytes

header: the string “CER” so the client can parse and handle.

Message: string saying there was an error

Example:

No cli example

Send message example:

<length> <header> <Message>

000000##CER error: result is too big

Maximum

The client sends the server maximum request

Description (Client -> Server):

Value	data_length	header	type	X1	...	Xn
Size(bytes)	4	3	max:	4	...	4

data_length: the length of the data will be sent: 4 bytes

header: the string “MAX” so the server can parse and handle.

type: max: command to execute

X1, ... , Xn: params to maximize

Example:

max: (X1 X2 ... Xn)

Send message example:

<length><header><max:> <(> <X1> ...<Xn> <)>

00000000MAXmax: (X1 X2 ... Xn)

The server sends the client max response

Description (Server -> Client):

Value	data_length	header	the maximum is	M
Size(bytes)	4	4	14	4

data_length: the length of the data will be sent: 4 bytes

header: the string “MRS” so the client can parse and handle.

M: the max

Example:

No cli example

Send message example:

<length> <header><the maximum is> <M>

000000##MRS the maximum is M

The server sends the client maximization error

Failure

Description (Server -> Client):

Value	data_length	header
Size(bytes)	4	3

data_length: the length of the data will be sent: 4 bytes

header: the string “MER” so the client can parse and handle.

Example:

No cli example

Send message example:

<length> <header> <Message>

00000003MER

Factorization

The client sends the server factorization request

Description (Client -> Server):

Value	data_length	header	X
Size(bytes)	4	3	4

data_length: the length of the data will be sent: 4 bytes

header: the string “FAC” so the server can parse and handle.

X: param to factorize

Example:

factors: X

Send message example:

<length><header> <X>

00000000FAC X

The server sends the client max response

Description (Server -> Client):

Value	data_length	header	M
Size(bytes)	4	3	4

data_length: the length of the data will be sent: 4 bytes

header: the string “FRS” so the server can parse and handle.

F: the factors

Example:

No cli example

Send message example:

<length> <header> <F>

00000000FRS F

The server sends the client factorization error

Failure

Description (Server -> Client):

Value	data_length	header	Message
Size(bytes)	4	3	48

data_length: the length of the data will be sent: 4 bytes

header: the string “FER” so the client can parse and handle.

Message: the message regarding the factorization error

Example:

No cli example

Send message example:

<length> <header> <Message>

000000##FER Can't calculate factors of a negative number

Failure (for num=1)

Description (Server -> Client):

Value	data_length	header	Message
Size(bytes)	4	3	32

data_length: the length of the data will be sent: 4 bytes

header: the string “FER” so the client can parse and handle.

Message: the message regarding the factorization error of 1

Example:

No cli example

Send message example:

<length> <header> <Message>

000000##FER Can't calculate factors of 1

Failure (for num=0)

Description (Server -> Client):

Value	data_length	header	Message
Size(bytes)	4	3	32

data_length: the length of the data will be sent: 4 bytes

header: the string “FER” so the client can parse and handle.

Message: the message regarding the factorization error of 0

Example:

No cli example

Send message example:

<length> <header> <Message>

000000##FER Can't calculate factors of 0

Quit

The client sends the server quit request or other direction

Description (Client -> Server):

Value	data_length	header
Size(bytes)	4	3

data_length: the length of the data will be sent: 4 bytes

header: the string “QUT” so the server can parse and handle.

Example:

quit

Send message example:

<length><header>

00000003QUT