



Python Developer – Software challenge

State machine structure

Description:

In this challenge, you are asked to implement a client that queries and visualises the structure of a state machine hosted on an MEQ server (see the server details section). The state machine consists of 26 states named from A to Z, where A is always the initial state, and Z is always the terminal state. Each state, except Z, has exactly 3 transitions, described by actions “1”, “2”, and “3”.

Upon establishing a connection to the TCP server, the server randomises the state machine. The structure is kept the same throughout the session.

After the connection is established, the server responds with the initial state “A”. Then, with the request of either “1”, “2”, “3”, the server will transition to a new state, and responds with the name of the new state. Upon reaching the terminal state “Z”, the server moves back to the initial state “A” and respond to the client with both “Z” and “A” (see example). Every request and response must be padded with <LF> (the Line Feed character) at the end.

Example:

<Connection established>

Server: A<LF>

Client: 1<LF>

Server: B<LF>

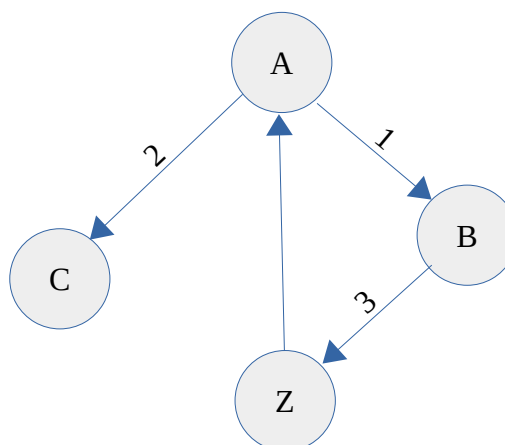
Client: 3<LF>

Server: Z<LF>

Server: A<LF>

Client: 2<LF>

Server: C<LF>



...

Server details:

Server IP: 20.28.230.252

Port: 65432

Solution delivery:

Please provide a link to the repository with your solution, as well as the image of the state machine structure.

Useful links:

1. <https://graphviz.org/>