**Numbers game HTTP server with REST API:**

**Approach:** Client (a Web browser or Curl) will send a data to the HTTP server. The backend of the server is created with python.

HTTP Server

HTTP client (Web browser

Or cURL)

1.POST /range (min, max, correct)

response JSON min, max and correct

2. GET /range

response JSON (min, max) values

3. POST /guess (guess value)

till we get the correct value

Response JSON {of posted values}

4. GET /guess (guess history)

Return JSON array with all the

guesses

The Diagram above describes the sequence of instructions and design of the software

Run the python script using

*$ python numbers\_httpserver.py*

**Design of the application is in the following way:**

**STEP 1:** Person A chooses a range with Minimum, maximum values and also chooses correct value to be guessed in that particular range. We use POST call to define this range and correct values. The values set are returned in JSON format.

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | *$ curl -i -H "Content-Type: application/json" -X POST -d '{"minimum":<yourchoice(num)>,"maximum":<yourchoice(num)>,"correct":<yourchoice(num)>}'* [*http://localhost:5000/range*](http://localhost:5000/range)  Example: /curl -i -H "Content-Type: application/json" -X POST -d '{"minimum":23,"maximum":27,"correct":24}' <http://localhost:5000/range> |

**STEP 2:** Person B gets the range by calling /range using GET and sees the maximum and minimum values.

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | *$ curl* ***-i*** [***http://localhost:5000/range***](http://localhost:5000/range) |

**STEP 3:** Person B tries to guess the value assigned by A using POST (guess) and gets responses according in JSON format.

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | *$ curl -i -H "Content-Type: application/json" -X POST -d '{"guess": <your guess(number)>}'* [*http://localhost:5000/guesses*](http://localhost:5000/guesses)  example: *$ curl -i -H "Content-Type: application/json" -X POST -d '{"guess": 28}'* [*http://localhost:5000/guesses*](http://localhost:5000/guesses) |

STEP 4: After a few guesses or after guessing the correct value Person B can see the guess history of JSON output. By using GET on /guess. As we use /guess as the end point.

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | *curl -i http://localhost:5000/guesses* |

STEP 5: Re run the above steps as many times starting from selecting a new range and selecting value to be guessed. ☺

**Testing:**

The Test script is run using:

*$ pytest testing.py*

It runs automated test cases against the numbers\_httpserver.py