✕ Lessons

Programming Assignment: NodeRED application

You have not submitted. You must earn 1/1 points to pass.

Deadline Pass this assignment by August 7, 11:59 PM PDT

Instructions

My submission

Discussions

Assignment goals

This assignment will demonstrate that you have a good understanding of how to create an application using NodeRED. You will create an application that will accept:

- an http GET request to http://*<your-node-red-instance>*/time that returns a JSON document containing the current time on the NodeRED server
- an http GET request to http://*<your-node-red-instance>*/random that returns a JSON document containing a generated random number
- an http GET request to http://*<your-node-red-instance>*/page that returns an http page that shows:

1. when the last time was requested
2. when the last random number was requested
3. the value of the last random number generated

How to complete the assignment

To complete this assignment you will need to have a NodeRED environment deployed into your Bluemix account. The flow you create can use any nodes in the input, output and function sections of the standard pallet available when you deploy the Bluemix boilerplate, with the addition of the Random node highlighted in a previous presentation (node-red-node-random). You should use features of Node-RED discussed in the previous lecture to complete this assignment and not use a database or other external storage service when completing this assignment.

When returning JSON data on /time the following format must be used

```
{"time" : "hh:mm"}
```

The time must be formatted to have a 2-digit hour in the 24-hour clock. Times before 10am should start with 0. Minutes also need to have 2 digits in the string.

The following are valid times:

- 09:04
- 13:54
- 12:00

They represent 4 minutes past 9 in the morning, 6 minutes to 2 in the afternoon and midday.

The following are not valid time formats for the same 3 times:

- 9:04 or 9:4 (hours and minutes must have 2 digits)
- 1:54 or 01:54 (must use 24 hour clock)
- 12:0 (minutes needs to have 2 digits)

The response must have a Content-Type of 'application/json' to be properly parsed as a JSON document.

When returning JSON data on /random the following format must be used

```
{"random" : number}
```

The response must have a Content-Type of 'application/json' to be properly parsed as a JSON document.

When returning the HTML data on /page the following format must be used

The http document (returned by /page) must not contain any formatting; simply return the data to show:

Time last server time request received at <hh:mm>

Last random number request returned <number>, which was received at <hh:mm>

Times should be formatted using the same rules used for JSON and in the above strings you need to replace <hh:mm> and <number> with the correct values. If no previous request has been received the http output should show the following strings:

No server time requests have been received

No requests for random numbers have been received

What to submit

All parts will be marked together, so when you have the flows completed you should select all the nodes then export the flow to your clipboard. Open a text editor and copy the flow to the text editor. Save as a text file named assignment2.txt. Submit the text file containing your flow

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.