Node.JS

Practice – 4

Overview

*In this hands-on session you will continue learning **Node.js**. Upon completion of this hands-on session, you should be able to:*

- *Create HTTP Web Server*
  - *Status Code, Methods*
  - *Routing*
- *Create and observe the processing on large file with streams*
  - *In the Web Browser*
- *Create and use Custom Modules*
  - *Understand how to expose functionality*
- *Use third-party modules – chalk and request*
- *Using RESTFUL APIs*
  - *For getting weather data*
  - *Sports events data*

Hands-on

1. Create a simple **HTTP** server, using the '**http**' core module.

   It should display the text '**Hello HTTP Server**' with '**h1**' element

   You are at liberty to decide the port for your server.

2. Your challenge is to create an HTTP server which will have the routes for the following URLs

   [a] / The root URL should display a home webpage
   It should have a simple UI with the logo of your organization.

   [b] **/about** URL should display the **mission**, **vision** statements of your organization

   [c] **/contact** URL should display
   - The address,
   - The URL for the company
   - The email address for info
   - The phone number(s)
   - Optionally even the Google Map location

[d] If the URL is other than the one given above, appropriate message with the appropriate Status Code should be displayed.

Execute the application and check the behaviour.

3. The next challenge is to design a HTTP Server which should enable us to read the contents of a **big/large** file.

   NOTE: Write a simple Node.js program which will help you in creating a big file

   HINT: You should be using **'stream'** for this purpose.

   Inspect the same in your web-browser and understand the **use-case** why we have to use **'stream'**

4. Create a custom module which will have functionality to compute the area and perimeter/circumference of a circle.

   - Expose the functionality
   - You are at liberty to decide about the name of the custom module

   Develop an app which uses the custom module and test the functionality.

5. Using the **'chalk'** third-party module, display the messages of an application as follows:

   [a] Application messages in Green
   [b] Error messages in Red

   Your task is to check if the 'chalk' module is able to display the required texts in different colours.

6. Google out – **Rapid API**

   Your challenge is to write a application which display the weather details of a city.

   It should display the name of the city.
   The current weather condition and temperature in Celsius.

   NOTE: You can use the **'request'** third-party module and get the information from the REST API from the web-site 'rapidapi'

   You need to create your account, subscribe and use the API.

7. The REST API for current **'sports'** events can be got from 'rapidapi'

   Using the API generate a list of all the upcoming Cricket events:

   - The tournament name
   - Match details
   - Start date
   - Stadium being played.

8. The REST API for current **'astronomy'** from 'rapidapi' will give the details on Sun rise and Sunset besides Moon rise and Moon set.

   Write an CLI application to display the same.

   Record your observation!

9. Using the **'readline'** core module, develop an application which will help us in accepting Employee details like - first name, last name, gender and age

   You should write a function which will help to display the employee details which were accepted. Moreover, the function should display – the full name of the employee, gender, current age and the age next year.

10. Your next challenge is to customize the earlier written application (Ref. Q6) and enhance it such that it accept a city name and gives the weather report for the given city.