Parspack Coding Challenge

Description:

We have a caching system for websites where users can upload their website’s content to be cached. There are two subsystems in this system:

- ➔ **Minification**

  This part is based on NodeJS. The following functionalities should be provided via the Restful API.

  1. Sign up
  2. Authenticate via token
  3. Upload CSS/JS files and save them to a preconfigured path. The endpoint must include the following conditions:
     - The CSS/JS files of received from the user should be placed in the following sample path: "/opt/username”, the path should be created if it’s not there.
     - If a file is saved previously it should be overwritten by new version.
     - Minification of CSS & JS. When receiving the file (CSS/JS) from the user, he/she should be able to ask the system to minify the CSS/JS file.
     - The system should be able to measure memory and time spend for minifcation of each file and log them in database.
  4. List of files per users with file details (size, creation date, type, duration of minifying each file and consuming RAM)

- ➔ **Web Server**

  This part is based on Openresty[1]. This webserver is a reverse proxy in front of minifcation subsystem, every API call to that subsystem is being passed via Openresty which is based on Nginx and it’s behavior can be modified via the LUA language[2]. (The installation guide link can be found in the footer). We expect you to be able to do the following using LUA at this stage:

  1. To all http responses, add this header: Header[parspack] = "HR_Code_Challenge"
  2. In the minification log, you should save all requests sent to the Minify endpoint, the request body size, the name of minified file and the response size.

[1] https://openresty.org

[2] https://github.com/openresty/lua-nginx-module

3. Add an OpenResty health check endpoint with LUA. The output should have the following conditions:
    - content-type: application/json
    - Body response : {
      ```
              "status": true
        }
      ```

Requirements

- Use NodeJS and popular framework
- Authenticate via JWT token
    - Add this user with factory: (user: parspack, pass: Twu5hKXXKZEQaJ)
- Please use one Creational and Structural design pattern

Delivery

- Upload the code on a private git server of your choice and also, deploy the project on the cloud server provided in the email.
- Please send to us postman collection

Plus Points

- Having test for endpoints.
- Converting a received image via API to WEBP format.