COSMATT App Setup

1. Pre-requisites

a. Node.js v8.0 + NPM
b. Angular CLI
c. Python 2.X (with environment variable)
d. Git
e. nodemon (package installed globally)
f. http-server (package installed globally)

2. Repositories

a. Client - https://github.com/comprodls/COSMATT-client

i. Take checkout of repo from above link
ii. Go into the directory & in a terminal, type - ***npm install***

b. Middleware - https://github.com/comprodls/COSMATT-backend

i. Take checkout of repo from above link
ii. Go into the directory & in a terminal, type - ***npm install***
iii. To start a localhost Middleware app, type - ***npm start***

c. Libraries - https://github.com/comprodls/libs-frontend-cosmatt

i. Take checkout of repo from above link

d. Product - https://github.com/comprodls/product-cosmatt

i. Take checkout of repo from above link
ii. Go into *assets\public* directory & then in a terminal, type - ***npm install***

3. App Development

a. Running localhost Ng-app

i. Go to *COSMATT-Client* folder & in a terminal type -
    1. For dev instance -
        a. ***ng serve --environment dev***
    2. For staging instance -
        a. ***ng serve --environment stg***
    3. For production instance -

a. ***ng serve --environment prod***

b. Running localhost Ng-app that uses local Middleware Instance

i. Go to COSMTATT middleware directory & run ***npm start***

ii. Now, according to the environment being used in Client app (dev/stg/prod), change the value of ***API_URL*** key in ***src\environments\environment.xxx.ts*** file to the URL of local middleware instance. Example - *http://localhost:3000/api/*

c. Running localhost Ng-app That Uses local Product Repo

i. Go to product directory - ***product-cosmatt\assets\public*** & in a terminal, run - ***http-server --cors="*"***

ii. Copy the localhost URL that is given in terminal. Example - *http://127.0.0.1:8080*

iii. Now, in the COSMATT-Client directory, open ***src\app\servo-system-course\services\product.service.ts*** file & in ***getPublicAssetsPath()*** function, change function return statement to return the URL of HTTP server created in the first step. Example, ***return `http://127.0.0.1:8080`;***