**Redis with nodeJs**

var express = require('express');

var path = require('path');

var app = express();

var router = express.Router();

var axios = require('axios');

var redis = require("redis");

var client = redis.createClient();

client.on('connect', () => {

    console.log('Connected to redis store')

})

client.on('error', (err) => {

    console.log('The redis error is ', err);

})

/\*

render() method works with templating engine like handlebars , pug etc.It first parse the engine file into plain html

but sendFile() do not need any of templating engine. We can directly send the plain html directly to the client

http://data.fixer.io/api/latest?access\_key=58fb9c9f34fe8519dff4cfeb25eca1fc

http://data.fixer.io/api/2019-08-11?access\_key=58fb9c9f34fe8519dff4cfeb25eca1fc

\*/

router.get('/', (req, res) => {

    res.sendFile('index.html', {

        root: path.join(\_\_dirname, 'views')

    })

})

router.get('/rate/:date', (req, res) => {

    const date = req.params.date;

    const url = `http://data.fixer.io/api/${date}?access\_key=58fb9c9f34fe8519dff4cfeb25eca1fc`;

    const newDate = date.split('-').join('\_');

    const ratesKey = `USD\_${newDate}`;

    console.log('The key is ==>', ratesKey);

    //     client.hgetall('frameworks', (err, frms) => {

    // if(err){

    //     console.log('The error is ',err);

    //     return;

    // }

    // console.log('Redis data is ',frms);

    //     })

    client.hget('todayRates', ratesKey, (err, ratesAll) => {

        console.log('----> ', ratesAll)

        if (ratesAll) {

            console.log('Inside redis server..');

            return res.json({

                rates: JSON.parse(ratesAll)

            })

        }

        else {

            console.log('Not Inside redis server..');

            axios.get(url).then((response) => {

                console.log("\*\*\*\*\*\* ", JSON.stringify(response.data.rates))

                setTimeout(()=>{

                client.hmset('todayRates', ratesKey, JSON.stringify(response.data.rates), (err, result) => {

                    if (err) {

                        console.log('The mega error is ', err);

                        return;

                    }

                });

                return res.json({

                    rates: response.data.rates

                })

            },4000)

        })

                .catch((err) => {

                    console.log('The error is ', err);

                })

        }

    })

})

app.use('/', router);

app.listen(3001, () => {

    console.log('Server starts')

})

/\*

app.get('/rate/:date', (req, res) => {

5  const date = req.params.date;

6  const url = `${API\_URL}/${date}?base=USD`;

7

8  const countKey = `USD:${date}:count`;

9  const ratesKey = `USD:${date}:rates`;

10

11  client.incr(countKey, (err, count) => {

12    client.hgetall(ratesKey, function(err, rates) {

13      if (rates) {

14        return res.json({ rates, count });

15      }

16

17      axios.get(url).then(response => {

18        // save the rates to the redis store

19        client.hmset(

20          ratesKey, response.data.rates, function(err, result) {

21          if (err) console.log(err);

22        });

23

24        return res.json({

25          count,

26          rates: response.data.rates

27        });

28      })

29      .catch(error => {

30        return res.json(error.response.data)

31      }));

32

33    });

34  });

35});

\*/