REST API Application using NodeJS and Express

Inital Project Setup

- Create a new Project
- Create a new folder taskApp `mkdir taskApp`
- Create a new npm project using `npm init`
- Create a new npm project

```
npm init

# select defaults for app name and version
# entry file select as app.js
```

- Create a new node project using defaults

```
npm init -y
```

Setup REST API

- Add Express

```
npm install express
```

- create a entry point for our project. Add a new file app.js
- add npm package express

```
    npm i express
```

- add code to the app.js file for a default get route

```
    var express = require("express");

    var app = express();

    var port = process.env.PORT || 3000;

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

```
res.send('REST API and Mongo DB');
})

app.listen(port,(req,res)=>{
 console.log('Server running on '+port);
})
```

- open the `package.json` and add start script

```
"start":"node app.js"
```

- Add nodemon package and set the default configuration

```
npm install --save-dev nodemon
# or use the shortcut
npm i -D nodemon
```

- Update the `package.json` file, add start:dev script

```
"start:dev": "nodemon app.js"
```

- Update the `package.json` file, add default configuration for nodemon

```
"nodemonConfig": {
        "restartable": "rs",
        "ignore": ["node_modules/**/node_modules"],
        "delay": "2500",
        "env": {
            "NODE_ENV": "development",
            "PORT": 5000
        }
    }
```

- Update the `app.js` to see if nodemon watches and restarts after a delay

```
app.get('/quotes',(req,res)=>{
    res.send('You will get all the quotes here!! Wait for it');
})
```

Linting your code.

Linting is very important, so that all developers use a same style guide for development. So code follows a common standards

Installing the packages

- install the package as dev dependencies

```
    npm i eslint -D
```

- update the package script

```
    "lint":"eslint"
```

- configure eslint

```
npm run lint -- -- init
```

Select the appropriate options based on your project requirements

- Modify the script for lint to start validation "lint": "eslint .",
- use the airbnb styleGuide and select json as the file format

Liniting your code

- run the below command to check if there are any linting errors

```
    npm run lint
```

- fix the linting errors
- Add custom rules in `.eslintrc.json` file under **rules section**.

```
{
    "env": {
        "browser": true,
        "commonjs": true,
        "es2021": true
    },
    "extends": [
        "airbnb-base"
    ],
    "parserOptions": {
        "ecmaVersion": 12
    },
    "rules": {
        "semi": ["error", "never"],
        "quotes": ["error", "single"]
    }
}
```

Adding ESLint Extension

- Installing a ESLint extension will give a visual clue to the errors

Adding Essential npm packages

Adding Morgan ,Winston & DotEnv

```
npm i winston morgan dotenv
```