**Assignment**

**Report**

**From**

**Bhupinder Singh**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| # | Topic | Pages |
| 1. | The assignment |  |
| 2. | System pre-requisites |  |
| 3. | The recipe |  |
| 4. | Application structure |  |
| 5. | Steps to run the application |  |
| 6. | Application flow and screen shoots |  |

1. The assignment

**Test Task for Frontend Developer (Angular)**

**The task**. Create a Single page web application for displaying the list of charts using the latest features of Angular.

**Libraries** that should be used:

- RxJs;

- Angular Material / Ng Bootstrap / Taiga UI / or others;

- Highcharts;

- NgRx / NgXs / Akita / or others;

- Any other library of your choice.

**Details.**

The UI must be responsive.

There will be a header with two links to two separated routes. One of them should be active connected with the current route.

"View mode" is the first route. There will be a list of charts and a date range filter (filtering by dates). Selected dates should affect charts on the page. Date range filter should be hidden if we have no charts.

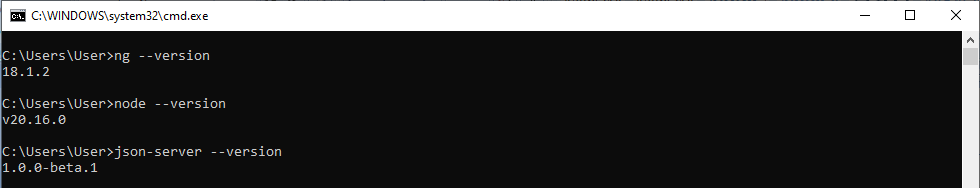
The data for the charts could be either randomly generated or fetched via any public API for one or for all charts. Each value should have a “value” and “date” fields. The default value for date range filter you can set by yourself.

"Settings" is the second route. There will be a list of charts. Users should have the possibility to add a new chart or edit an existing one. Those settings should be implemented with a modal window. It should be possible to change a name, type (line, spline, area…) and color for each of them. Also we should have the possibility to remove the chart. After saving, new settings will be applied.

1. System pre-requisites

|  |  |  |
| --- | --- | --- |
| # | Utility | Version |
| 1. | NodeJs | 20.16.0 |
| 2. | Angular CLI | 18.1.2 |
| 3. | Json-server | 1.0.0-beta.1 |

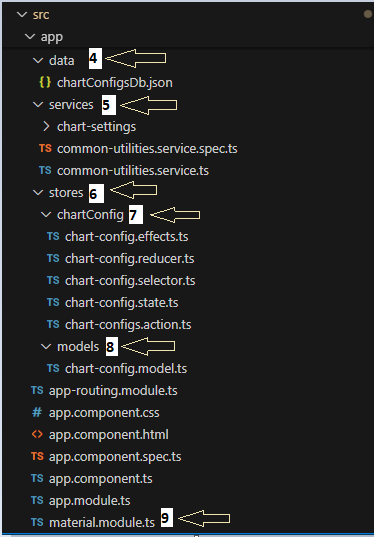
For more celerity about the above utilities, please refer the below screen-shoot.

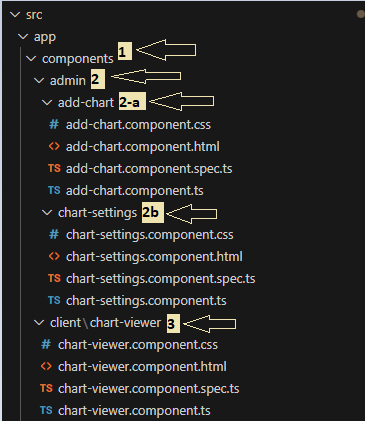


1. The Recipe

|  |  |  |
| --- | --- | --- |
| # | Ingredient | Purpose |
| 1. | Highcharts | For charts |
| 2. | Angular Material | UI Library |
| 3. | NgRx & Rxjs | State management & handling data streams |
| 4. | Json server | To mock REST API’s |

1. Application Structure

 In the following screen-shots, I have listed only those folders/files which are not included in default folder structure in an angular application.



|  |  |  |  |
| --- | --- | --- | --- |
| # | Ingredient | Path | Purpose |
| 1. | components | src\app\components | This folder consists all sub-folders/files which further contain all components (html pages). |
| 2. | admin | src\app\components\admin | This folder contains components related to the chart settings. |
| 2-a | add-chart | src\app\components\admin\add-chart | This is the folder of the component of add/edit chart configurations. |
| 2.b | chart-settings | src\app\components\admin\chart-settings | This is the folder of the component of manage & listing of added charts. |
| 3. | client | src\app\components\client | This folder contains components related to the view charts to the user. |
| 4. | data | src\app\data | This folder contains ***chartConfigsDb.json*** file which acts as database for storing charts & their settings. |
| 5. | services | src\app\services | This folder contains all services components. |
| 6. | stores | src\app\stores | This is the folder contains all folder/files related to store and state management and all CRUD operations business logic. |
| 7. | chartConfig | src\app\stores\chartConfig | This folder is has all files related to chart settings store. |
| 8. | models | src\app\stores\models | This folder contains file chart-config.model.ts, which defines the tructure of chart config. |
| 9. | material.module.ts | src\app | This is the featured module and contains all Material UI modules used in the application. |

1. Steps to run the application

Step 1: Clone the project from the following link

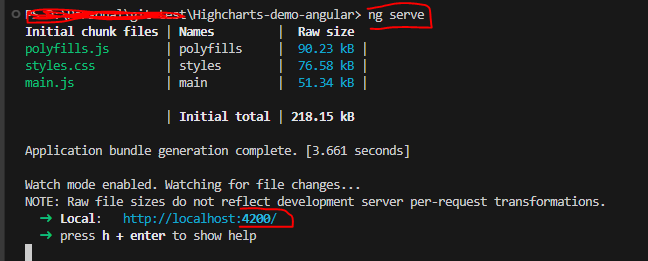
<https://github.com/officialbsingh1983/Highcharts-demo-angular>

Step 2: open the command prompt in the cloned directory (Highcharts-demo-angular)

Step 3: execute the following commands in the command prompt

1. npm install
2. ng serve

Please note ng serve expected to run the application on 4200 port, in case of differ port in your system then please use running port number in the Step 7.

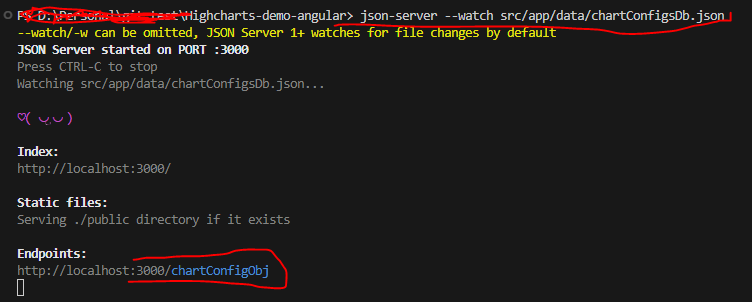


Step 4: open an another command prompt in the same directory location

Step 5: execute the following command in the command prompt

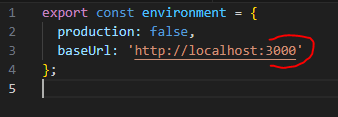
1. json-server --watch src/app/data/chartConfigsDb.json

Please note, above command is expected to run on 3000 port (<http://localhost:3000/chartConfigObj>). Please refer the following screen-shot



In case of the port number is differ from the said port the please follow the following step:

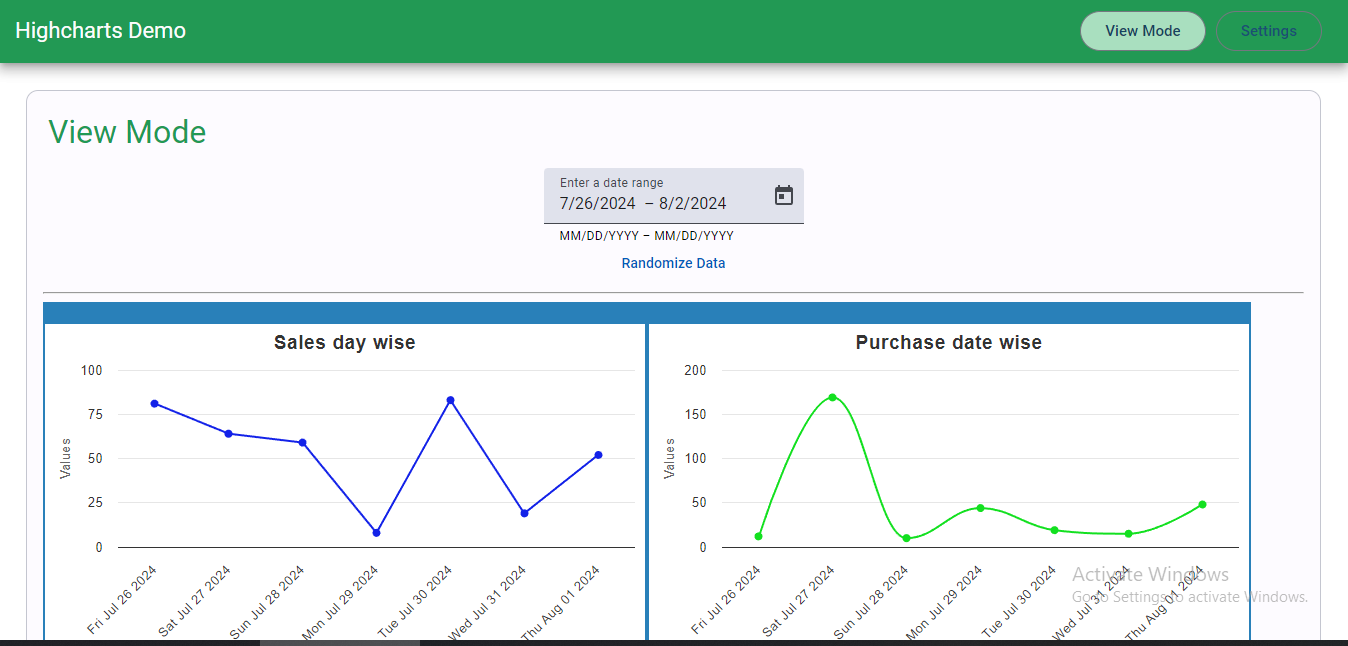
Step 6: (Please ignore this step in case your json-server is running on 3000 port) Please open ***‘\src\environments/ environment.ts ‘*** file in any text editor. Change the port number from 3000 to your system’s json-server running port number. Please refer the following screen-shot



Step 7: Open browser and go to the following address

http://localhost:4200/

1. Application flow and screen-shots
2. View Mode page (default route)



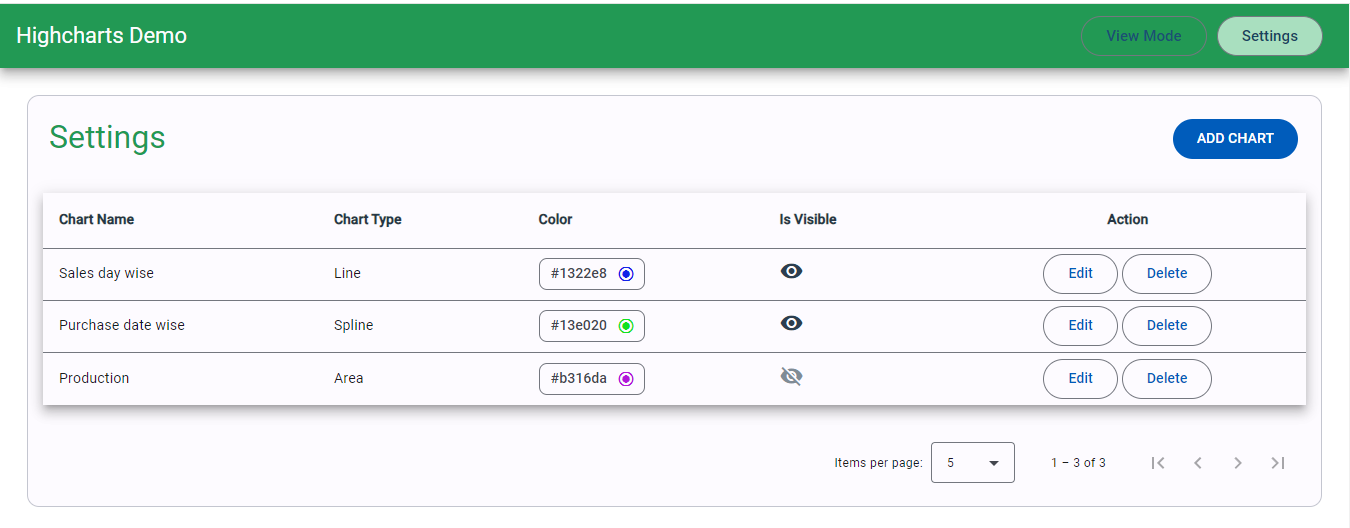
Graph Title

Button for generating randomized data

Date Range control

**Link :** View Mode ***(activated)***

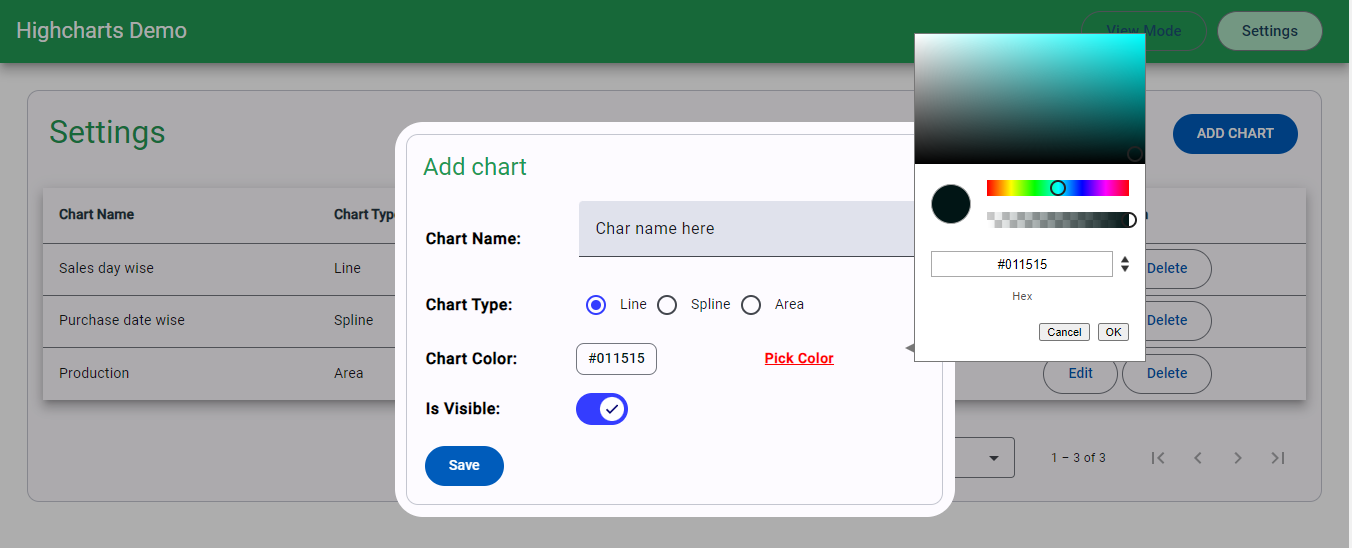
1. Settings page



**Button:** Edit & Delete buttons

**Button:** To open add chart dialog

**Link :** Settings ***(activated)***



**Button to open color picker.**

**Color picker**

**Modal Pop-up:** To Add /Edit charts data

* Run time randomized data is shown in the charts.
* To manage the charts settings, NgRx, RxJs and json-server are used.
* The default date range is last 8 days which can be change further.

Thank you