(Frontend) ChartJs Example project with Blazor

Overview

This project is an example implementation using .NET Blazor Web Assemble + ChartJs

Dependencies

- .NET Blazor Web Assemble: Tested with dotnet6
- ChartJs: https://www.chartjs.org/docs/latest/getting-started/usage.html

How to try

(We assume you have already set up dotnet6, if not, check the tutorial: https://dotnet.microsoft.com/en-us/learn/aspnet/blazor-tutorial/install)
Build and run project:

- With Visual Studio, try "Build Solution" -> "Debug Run"
- With CLI, use `dotnet run` command under the project folder, ./ChartJsExample

Code architecture

- `wwwroot/scripts/chartjs_app.js` : javascript application logic that manipulates ChartJs library
- `Shared/Chart/Gateways/ChartJsInterop.cs` : Interop code to use `chartjs_app.js` from C# and Javascript
- `Shared/Chart/Views/RenderChart.razor` : View implementation that shows example implementation of graph-chart using ChartJs

Assignment

Modify `Shared/RenderChart.razor`, an chart view to visualize the test data to meet the following requirements
If you run the project, you'll see test data as json string at https://localhost:7194/. This is an example data of our robot's maintenance report. You will also see an example graph-chart on the same page.

- Task 1. Modify the graph-chart to show the following;
    - The horizontal axis should show each date (`WorkDate` in the json)
    - The vertical axis should show the time (minutes)
    - Plot each date's total `WorkTime.Maintenance` (if we visited stores a few times on a day, the sum of WorkTime.Maintenance should be plotted)
    - The chart should be "Line Chart"
    - The horizontal axis should show the recent 7days from the latest date in the received data
- Task 2. Add the following functions to what you made at Task1;
    - Add a UI to select the latest date to show the line chart. If you changed that date, then line chart will render it according to that new-date-range

- Task 3. Add a test process to verify functionality what you made at Task 1 and 2;
    - You don't need to cover all the cases in this session if your time is limited. In that case, please leave a text for the further test plan

(Backend) Firebase functions project

Overview

This project is an example implementation using firebase functions in typescript

Dependencies

- Node.js 16
- Firebase CLI

How to try

(We assume you have already set up firebase CLI. If not check https://firebase.google.com/docs/cli)
Build and run project:

- Inside functions folder run `npm run build` & `firebase emulators:start` to test in emulator environment

Note: no need to deploy. Just run locally using firebase emulator is fine

Assignment

Task: Create functions which allow following behaviours

- Save robot maintenance report data
- Get maintenance data of specific store
- Update the WorkTime.Maintenance time for a specific store
- Get maintenance data for a specified data range and a specific store

Note: you can find the initial data to load in firestore under `ChartJsExample/wwwroot/sample-data/reports.json`

Notes:

- If you have any question to proceed the assignment, feel free to ask on slack
- Please submit the followings as your assignment results
    - The code itself, push to git
    - Text report to explain how long time you spent for what part through the assignment
- We assume you will use less than 10 hours. If you reached the time-limit, please submit your working-results at the time