

Background

An airport requires a grid view of flight schedules. The users expect to update or clear registration against individual flight movements.

Technical Requirements

- Extend a Single Page Application using Angular, CSS, HTML and Typescript. Attached is a zipped skeleton project.
- To run the skeleton project, unzip the project. Ensure that you install npm and node (<https://nodejs.org/en/download/>) - Verify that you are running the latest LTS version of node and npm.
- Open terminal window – change folder to unzipped project folder flight-schedules. First time, you will need to install all npm packages. Execute the following command:

```
npm install
```

This installs all packages and needs to be done once only.

To run the project, execute the following command:

```
npm start
```

```
PS E:\tmp\tst\new\flight-schedules> npm start
> flight-schedules@0.0.0 start E:\tmp\tst\new\flight-schedules
> ng serve
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
```

This will start the app on port localhost:4200 – open your browser on <http://localhost:4200/>

The code you write should be tested using latest Chrome browser. There are no other browser compatibility requirements.

The skeleton project contains a grid only and should look like the below screenshot in a browser:

FLIGHT	SCH. DATE	SCH.TIME	O/D	REGISTRATION

- If you are not familiar with Angular framework, consult online documentation <https://angular.io/>
- You may choose whatever editor rocks your boat. Internally we use Visual Studio Code for all front-end development which is a free download: <https://code.visualstudio.com/>
- The aim of the test project is to test your front-end development skills. There is no requirement for backend/webserver/database integration. All data should be mocked (up to you how this is done).
- Your first task is to plug in mock flight data to the skeleton project. All mock data is located in "src\assets\mock-data\".
- The entry component which is rendered on start-up is src\app\app.component

- Once you are successful and plug in mock data, the grid will look similar to the below screenshot:

FLIGHT	SCH. DATE	SCH.TIME	O/D	REGISTRATION
PC123	23/11/17	11:39	Wroclaw
PC145	23/11/17	11:52	Warsaw
FEI760	27/11/17	12:30	Vestmannaeyjar
N27VP	27/11/17	12:30	Keflavik	N27VP
MYA77	27/11/17	12:45	Reykjavik	TFMYV
FEI760	27/11/17	12:47	Reykjavik
FEI720	27/11/17	01:00	Bildudalur
FEI761	27/11/17	01:15	Reykjavik
MYA77	27/11/17	01:23	Akureyri	TFMYV
FEI720	27/11/17	01:31	Reykjavik
FEI761	27/11/17	01:31	Vestmannaeyjar
FEI721	27/11/17	02:00	Reykjavik
MYA77	27/11/17	02:00	Hornafjörður	TFMYV

- Your next task is to implement the features described below. None of the below has been implemented in the skeleton project.
- Clicking registration cell directs a user to a modal window. Modal window displays on top of grid, centred. Use your creativity to focus user's attention on modal content – there is no specific requirement around this.
- Header of the modal contains the following elements:
 - Row 1 is flight number for which the modal is open and word "Registration"
 - Row 2 consists of scheduled date time of the flight in format DD/MM/YY – HH:MM
 - "X" icon in the right corner. Clicking the icon closes the modal without saving any changes.
- Registration input with label. This contains currently assigned registration to the flight which is being edited. If there is no registration assigned to the flight, the input is empty.
- When user starts typing into registration input box, a suggestion box appears if user entered at least two characters.
- Footer contains two buttons:
 - Cancel – clicking this button discards all changes and closes the modal.
 - Save – clicking this button saves current value of input as the registration of the flight. There is no need for persistent storage, this must be saved in memory only – all changes are lost on browser refresh.
- Visual design – no registration stored against the flight:

LH857 Registration

03/12/17 - 14:45

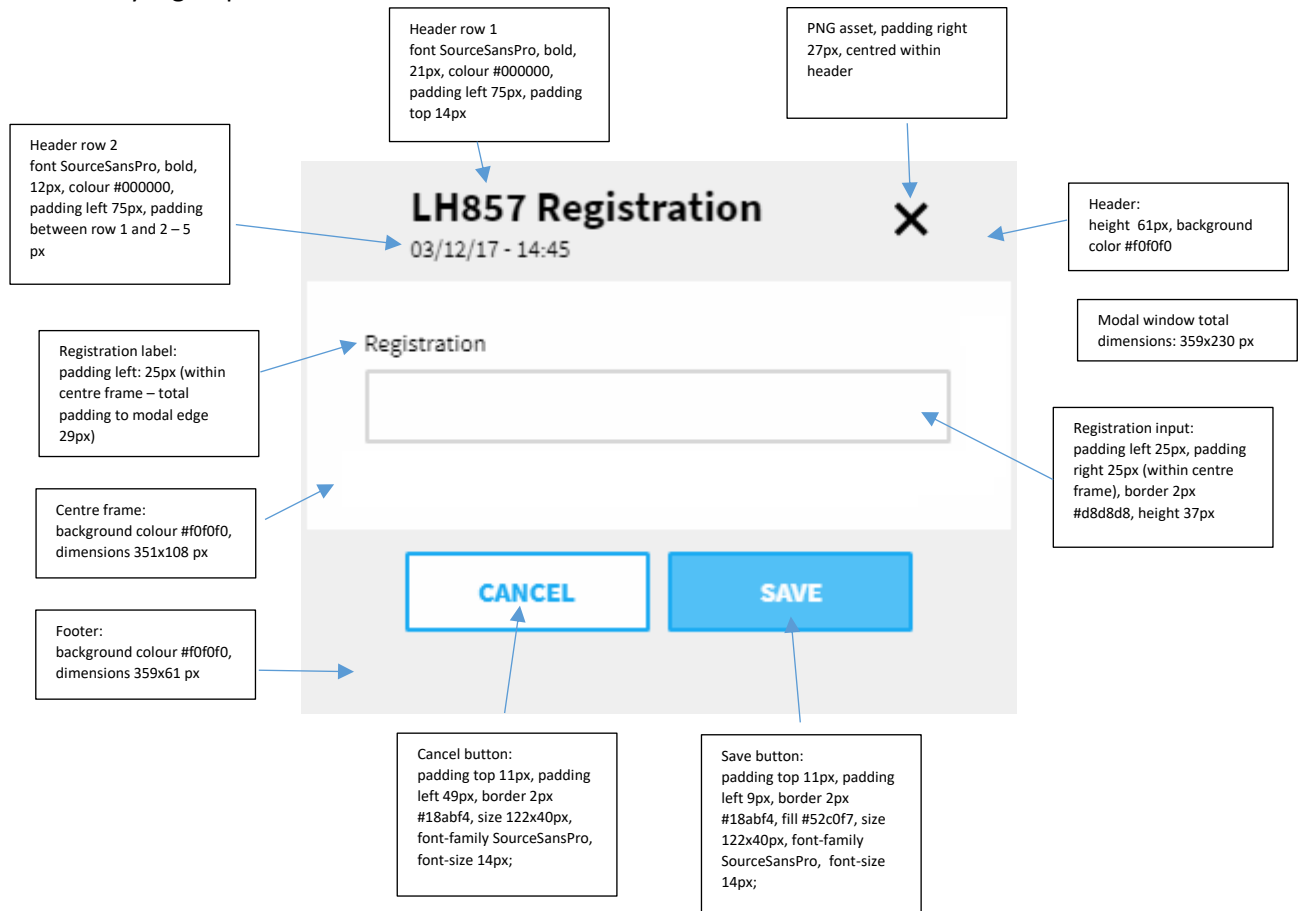


Registration

CANCEL

SAVE

Styling requirements:



Dropdown expanded:

LH857 Registration X
03/12/17 - 14:45

Registration

TFF
REGISTRATION (27)
TFFAA
TFFAE
TFFET
TFFIA
TFFIC
TFFIG
TFFIH

Font size 14px, bold

Fixed header
REGISTRATION (bold),
number of matched
records in brackets, size
12px, header height 30px

Row height: 37px, font size
14px, matched part of
registration in bold

Border colour changes to
#18abf4

No custom scrollbar – use
default browser scrollbar

Row separator: height 1px,
colour #d8d8d8

Result box height: fit 7
rows, if more than 7 rows
show scrollbar

Modal to be displayed on
top of grid, use your
creativity on how to focus
your user's attention to the
modal instead of the grid.

Summary

Please follow all explicit requirements closely. If specific areas are not described in the specification – feel free to use your judgement to finish the project.

Zip finished project and upload to google drive / dropbox or a similar service. Do not include npm packages as part of the upload (clear node_modules folder before zipping and uploading). Please send us the link to your solution.