Full stack Angular / Java challenge - 30 minutes presentation

The advanced full stack challenge contains two mini projects:
- Angular mini project
- Console Java mini project

The goal of this challenge is to give us insight in your coding level and style.

For an advanced profile, the challenge can be done in one week for 1 or 2 hours per day. A presentation will be planned in order to expose and explain your solutions.

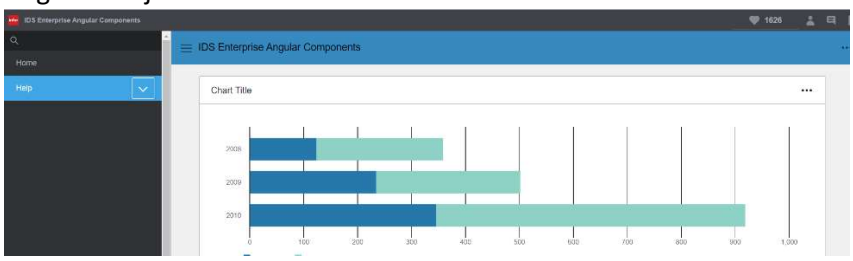
We remain at your disposal for any further information ☺

I– Mini project Angular

1- Input

"Angular challenge Starter.zip" file that contains :

- Angular Project Starter:



The project is based on :

- infor design system : https://design.infor.com/
- enterprise-ng components : https://github.com/infor-design/enterprise-ng

- Sample_Data.xlsx

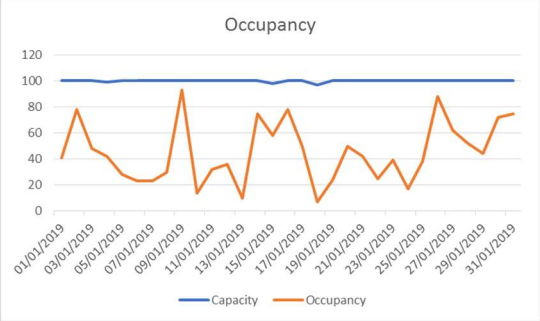
2- The challenge

The challenge consists to **implements** the scenario below using the entreprise-ng components:

- We need to add a "statistics" module that contains a component named "Occupancy"
- The "Occupancy" component will display occupancy data in two ways (a sample data is provided in Sample_Data.xlsx file):
  - **As a data grid**

Date	Capacity	Occupancy
01/01/2019	100	41
02/01/2019	100	78
03/01/2019	100	48
04/01/2019	99	42
05/01/2019	100	28
06/01/2019	100	23
07/01/2019	100	23
08/01/2019	100	30
09/01/2019	100	93
10/01/2019	100	14
11/01/2019	100	32
12/01/2019	100	36
13/01/2019	100	10
14/01/2019	100	75
15/01/2019	98	58

- **As a Line chart**



II- Mini project JAVA

1- Programming Exercise

Try to make your code well structured. Pay attention to:
- SOLID design principles
- Software design patterns
- Clean, readable code
- Unit tests for the critical sections of code

Be able to explain and defend design decisions.

2- Use case

We regularly receive documents from an external system: weekly status reports, monthly overviews, detailed reports. Sometimes, these documents are plain text files, and sometimes they are xml files.
We want to replace certain text phrases in the documents.
We only want to search for specific text phrases; so, no regular expressions, no wildcards.
In the xml files, we want to be able to replace (parts of) attribute values.

3- Exercise

Design and code a program that will allow us to search and replace text phrases in regular text documents and xml documents.
- The program should read from standard input, and write to standard output.
- The program accepts three command line parameters: 1. Type: the type of data sent on standard input (xml or text)
2. Search string
3. Replace string

- The input xml is always well formed. The output xml should also always be well formed and structurally identical to the input xml.
- In case of xml, the search and replace should be applied to attribute values.
- The program should be easy to maintain and extend, for instance to add a more flexible search method, or other file types.
- The program should be able to handle large files – think several gigabytes of data.
- The program should use a predictable, limited amount of memory – think a few hundred megabytes of memory.

- It is okay to handle errors (e.g. in user input) by throwing exceptions.
- You may assume the data is plain ASCII.

Example 1:

Input file: manifesto.txt

```
Our highest priority is to satisfy the customer
through early and continuous delivery
of valuable software.

Welcome changing requirements, even late in
development. Agile processes harness change for
the customer's competitive advantage.
```

Command:

```
searchreplace.exe txt “customer” “client” < manifesto.txt > result.txt
```

Output file: result.txt

```
Our highest priority is to satisfy the client
through early and continuous delivery
of valuable software.

Welcome changing requirements, even late in
development. Agile processes harness change for
the client's competitive advantage.
```

Example 2:

Input file: configuration.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<configuration>
  <properties>
    <!-- Uncomment the following to enable the profiler -->
    <!-- <profiler mode=”trace”/> -->
    <log level=”trace”><file name=”trace-20180101.log”/></log>
    <comment>Level can be either “trace”, “info” or “error”.</comment>
  </properties>
</configuration>
```

Command:

```
searchreplace.exe xml “trace” “error” < configuration.xml > result.xml
```

Output file: result.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<configuration>
  <properties>
    <!-- Uncomment the following to enable the profiler -->
    <!-- <profiler mode=”trace”/> -->
    <log level=”error”><file name=”error-20180101.log”/></log>
    <comment>Level can be either “trace”, “info” or “error”.</comment>
  </properties>
</configuration>
```

III- 30 minutes Presentation

After the implementation, you are invited to present your solutions including your first experience with the infor design system and entreprise-ng framework.