

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.

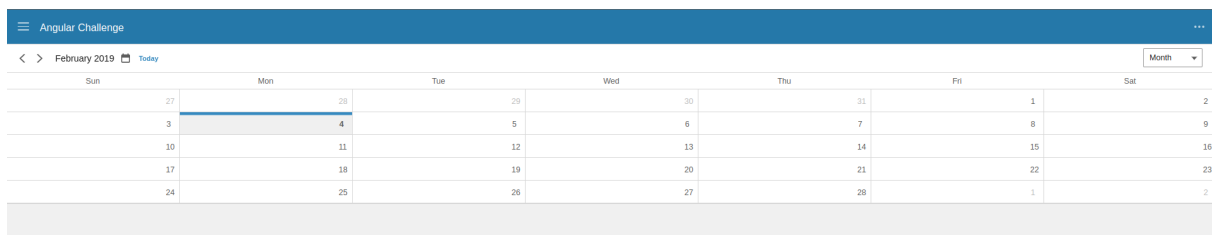
We remain at your disposal for any further information

I– Mini project Angular

1- Input

“Angular challenge Starter.zip” file that contains :

- Angular Project Starter:



Angular Challenge						
< > February 2019 Today Month						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2

The project is based on :

- o [infor design system](https://design.infor.com/) : <https://design.infor.com/>
- o [enterprise-ng components](https://github.com/infor-design/enterprise-ng) : <https://github.com/infor-design/enterprise-ng>

2- The challenge

The challenge consists to implement the scenario below using the enterprise-ng components:

- We need to add missed components:
 - Combobox with “DayLegend” as default value
 - Date range ([soho-datepicker](#))
 - Days of week ([soho-checkbox](#))

Type

DayLegend

Start

End

04/03/2022

04/21/2022

monday

tuesday

wednesday

thursday

friday

saturday

sunday

save

<

>

March 2022

Today

Mon	Tue	Wed	Thu	Fri	Sat	Sun
28	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

event1

event2

weekend

- By clicking on the button “save”, we need to fill all columns from start date to end date while respecting the Days of week conditions

Type

DayLegend

Start

End

04/03/2022

04/21/2022

monday

tuesday

wednesday

thursday

friday

saturday

sunday

save

<

>

April 2022

Today

Mon	Tue	Wed	Thu	Fri	Sat	Sun
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1

- Here we can apply a filter

Type
DayLegend ▾

Start End
04/03/2022 04/21/2022

☒ monday
 ☐ tuesday
 ☒ wednesday
 ☐ thursday
 ☒ friday
 ☒ saturday
 ☒ sunday

save

< > April 2022 Today

Mon	Tue	Wed	Thu	Fri	Sat	Sun
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1

- If you disable friday in the checkbox, it will be automatically hidden from the calendar

Type
DayLegend ▾

Start End
04/03/2022 04/21/2022

☒ monday
 ☐ tuesday
 ☒ wednesday
 ☒ thursday
 ☐ friday
 ☒ saturday
 ☒ sunday

save

< > April 2022 Today

Mon	Tue	Wed	Thu	Fri	Sat	Sun
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1

- We can make some changes on the date range

Type
DayLegend ▼

Start: 04/03/2022 End: 04/14/2022 change end date

☒ monday
 ☐ tuesday
 ☒ wednesday
 ☐ thursday
 ☐ friday
 ☐ saturday
 ☐ sunday

save

Mon	Tue	Wed	Thu	Fri	Sat	Sun
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1

- We can add a new day legend using modal: for example, applying the day legend on April 12th

> April 2022 Today

Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	
4	5	6	7	8	
11	12	13	14	15	
18	19	20	21	22	
25	26	27	28	29	

DayLegend

add new daylegend

Cancel Submit

< > April 2022 Today

Mon	Tue	Wed	Thu	Fri	Sat	Sun
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1

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>
```