

DG technical qualification test (Angular)

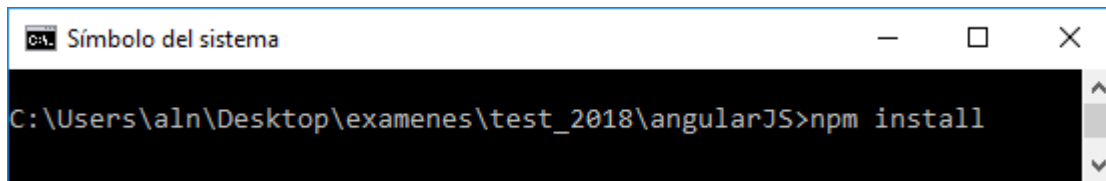
Complete the provided application as followed:

Prerequisites and installation:

Before starting it is necessary to have installed nodeJS, (preferably at least the latest LTS version). It can be obtained here: <https://nodejs.org/en/>

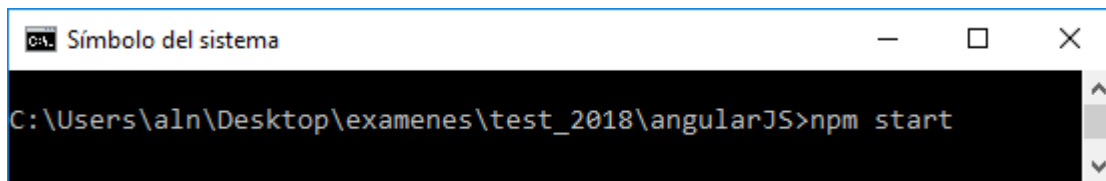
Having node installed and from a terminal, we move to the root directory of the project and execute the following commands

npm install



```
C:\Users\aln\Desktop\exámenes\test_2018\angularJS>npm install
```

npm start



```
C:\Users\aln\Desktop\exámenes\test_2018\angularJS>npm start
```

If everything went well, the browser will open with the project page. With this done you will be ready to start the test.

Complete the component called stockComponent that will show you a list of our products and allow you to add some more.

The files that you will modify during the test are:

- /src/app/sstock/**stock.component.js**
- /src/app/stock /**stock.component.html**
- /src/app/stock/**stock.component.css**

Exercise 1

Starting the component, it must make a call to the server to collect the data in order to be displayed and painted as a table in HTML. The URL where we will collect this data is in **/stock** applying the **GET** method. The table should appear centered using CSS.

Tasks to perform:

1. Obtain the data as described above from the server (**JS**)
2. Paint the table data (**HTML**)
3. Add style to visualize the table in a usable way (**CSS**)
(Note: Although it is solely required that the table is displayed centered on the page, extra style that you provide (colors, fonts, etc.) will also be positively valued.)

STOCK LIST

Product	Price	Price with taxes	Amount in stock	Total price
Cookies	1	1.21	10	12.1
bread	15	18.15	20	363
Cofee	2	2.42	32	77.44
Teapot	46	55.66	98	5454.68
Water	18	28.08	48	1347.84

Exercise 2

Insert a button under the table that when pressing it must show a form with three entry fields:

- Name of product (default: string empty)
- Price of product (default: number 0)
- Quantity of product (default: number 0)

Clicking on the button that opens the form, should lead to it closing the forms and clearing out the data entered by the user in the entry fields.

Tasks to perform:

1. Open the form to add products (**JS**)
2. Paint the form onto the page (**HTML**)
3. Closing the form and Clearing entered data (**JS**)
4. Form style (**CSS**)

(Note: Although it is solely required that the table is displayed centered on the page, extra style that you provide (colors, fonts, etc ...) will also be positively valued.)

STOCK LIST

Product	Price	Price with taxes	Amount in stock	Total price
Cookies	1	1.21	10	12.1
bread	15	18.15	20	363
Cofee	2	2.42	32	77.44
Teapot	46	55.66	98	5454.68
Water	18	28.08	48	1347.84

Add more products

Product name:

Product price:

Amount

Produc name

0

0

Save

Exercise 3

Save the data entered in the form on the server by clicking on save and update the stock table with the data obtained from the server. Once the table has been updated, it is necessary to clear out the data entered by the user in the entry fields.

Create a request to the server using the url **/stock** applying the **POST** method

The body of the request should consist of the following fields in **JSON** format:

- name: String
- price: Number
- amount: Number

Tasks to perform:

1. Make a HTTP request to the server with the new data (**JS**)
2. Update the stock table with the new data obtained from the server (**JS**)

STOCK LIST

Product	Price	Price with taxes	Amount in stock	Total price
Cookies	1	1.21	10	12.1
bread	15	18.15	20	363
Cofee	2	2.42	32	77.44
Teapot	46	55.66	98	5454.68
Water	18	28.08	48	1347.84
new product	5	6.05	24	145.2

Add more products

Product name:

Product price:

Amount

Save

Produc name

0

0|