

Test Java

Necesitaríamos que crear una aplicación utilizando el siguiente stack de tecnologías
maven + java + spring boot

La aplicación debe ser una restfull web-api que exponga dos servicios.

1)Un servicio "cotización" que sea llamado con los parámetros de la moneda ej:
<http://localhost:8080/MyResftfullApp/Cotizacion/MONEDA>, donde MONEDA podría ser
"Dolar", "Pesos", "Real"; implementando el patrón Strategy para consumir los diferentes
servicios.

El patrón strategy se puede leer desde <http://www.oodesign.com/strategy/-pattern.html>

Para los parámetros Pesos y Real, implementar un servicio que dé una respuesta que sea
error 401 no authorized de http. Para el parámetro Dólar implementar un servicio que
consuma la cotización del dólar actual desde el servicio externo:

<http://www.bancoprovincia.com.ar/Principal/Dolar>

2)Crear un servicio "usuarios" que devuelva todos los usuarios de una tabla "User" con los
campos, id, nombre, apellido, email, password. Para esto deberás crear la base de datos, la
tabla, popular con un script y empaquetar todos los datos necesarios para poder correr en
otra computadora la aplicación y los servicios.

3)Crear unit test para la capa de servicios interna.

Necesitamos el código fuente, instrucciones y script para generar la base de datos y popular
con datos de prueba.

4)Subir el proyecto a github

Test Javascript

Necesitamos crear el FrontEnd de una aplicación que muestre la cotizaciones de monedas.

1 - Monedas: Pesos, Real, Dólar.

Crear datos mock para las diferentes cotizaciones de las monedas.

(sugerencia técnica "Promises")

2 - Para simular datos reales los resultados (mock) de las cotizaciones deberán oscilar entre 4 modos configurables mediante un (config.js):

A: +/- 1

B: +/- 0.1

C: +/- 0.01

D: +/- 0.001

2 B - Se puede implementar un BackEnd que consuma la Cotización de Dólares de

<http://www.bancoprovincia.com.ar/Principal/Dolar>

La recomendación es conectarlo al proyecto anterior donde se desarrolló la web api

3 - La cotizaciones de las 3 monedas se deben ver en 3 columnas claramente tituladas donde las cotizaciones se deberán ver con un estilo "Card"

(opcion: <http://v4-alpha.getbootstrap.com/components/card/>)

4 - La cotizaciones se deberán actualizar cada 10 segundos

(número configurable en el config.js)

5 - Tras cada actualización en vez de Reemplazar cada "Card" se deberá agregar una Card nueva a la columna de manera que se vea el histórico de la misma.

6 - Cada Card deberá mostrar el tiempo al estilo Twitter:

A: Hace X segundos

B: Hace Y minutos

C: hace Z horas

(Se agradecen UnitTest con Jasmine o el framework de preferencia)

7 - El estado de la aplicación deberá mantenerse incluso después de un Refresh del browser.

(Se recomienda LocalStorage)

(Se agradecen UnitTest)

8 - Subir el Proyecto a GitHub

Tecnologías recomendadas:

- React/Angular
- Less/Sass
- ES5/ES6/Typescript
- Jasmine
- Grunt/Gulp
- Webpack
- RequiredJs