ENUNCIAT PEC 4

Testing en Angular

Desenvolupament front-end avançat

Máster universitari en desenvolupament de llocs i aplicacions web

Universitat Oberta de Catalunya

Contingut

Format d'entrega

S'entregarà tot el projecte comprimit en format **.zip** sense incloure el directori **"node_modules"**.

Aquesta pràctica 4 es resoldrà a partir de la solució oficial de la pràctica 2, concretament l'exercici on tenim totes les crides de la api implementades amb observables.

Enunciat

Una cop estudiats els exemples de testing del document **Teoria_PEC4_es.pdf**, podrem començar a treballar en la resolució d'aquesta PEC4.

Passos previs:

- El primer que haurem de fer es eliminar tots els fitxers de test (.spec.ts) que s'han creat automàticament al projecte.

- Posteriorment, afegir al fitxer **angular.json** a l'apartat **test.options** la línia:

```
"codeCoverage": true,
```

Exercici 1 – test pipe

Seguint l'exemple del document de teoria on s'implementa el test d'un pipe que duplica un valor, en aquest exercici tindrem que implementar el test del pipe que tenim al nostre projecte el qual dona format a la sortida de una data en funció del paràmetre que li passem.

Tindrem que crear el fitxer **format-date.pipe.spec.ts** i implementar els següents test:

- Que es crea el pipe correctament
- Donada una data i l'argument 1 retorna el format esperat
- Donada una data i l'argument 2 retorna el format esperat
- Donada una data i l'argument 3 retorna el format esperat
- Donada una data i l'argument 4 retorna el format esperat

Exercici 2 – test rutes

Seguint l'exemple del test d'una ruta del document de teoria, en aquest exercici haurem d'implementar els següents passos:

- refactoritzar el component **header.component** de la següent manera:
    - En lloc de tindre per cada punt de menú un mètode **home(), login(), register(), adminPosts(), …** i que cada un cridi a un **router.navigateByUrl** amb la url determinada, anem a implementar un únic mètode **navigationTo** en el que li passarem per argument la ruta destí.
    - Adaptarem la vista **header.component.html** de manera que tots els botons cridaran al mètode **navigationTo** però cada un li passarà per argument la ruta destí que toqui, **home**, **login**, **register**, …
- Una cop refactoritzat el component, tindrem que implementar els següents test:
    - Que el component es crea correctament
    - Testejar que la navegació es correcta
        - Per cada possible ruta (home, login, register, posts, categories i profile) de les que gestiona el **header.component** validar que es llança el **navigateByUrl** amb l'argument correcte.
            - Exemple de ruta **home**:
                - component.navigationTo('home');
                - expect(spy).toHaveBeenCalledWith('home');

Exercici 3 – test servicios

Seguint l'exemple del document de teoria on s'implementa el test d'una funció d'un servei, haurem d'implementar el test dels següents serveis:

- CategoryService
- PostService

De cada servei haurem de:

- implementar el test de que el servei es crea correctament
- implementar el test de cada funció que implementa el servei
    - per cada funció, a part de testejar que el resultat es l'esperat, haurem de testejar que el tipus de crida es l'esperat també (get, put, post o delete)

Exercici 4 – test components

Seguint l'exemple del document de teoria on s'implementa el test d'un component en el que es valida el mètode **loadPosts** del component **HomeComponent,** en aquest exercici haurem d'implementar uns test molt semblants. Concretament haurem d'implementar:

- al component **CategoriesListComponent**
    - un test que validi que es crea el component correctament
    - un test que ens asseguri dos coses:
        - que quan es crida el **loadCategories** es llanci la crida **getCategoriesByUserId** del servei
        - que la resposta de la crida sigui la esperada
    - un test que validi que es crida el **navigateByUrl** amb l'argument correcte quan creem una categoria
    - un test que validi que es crida el **navigateByUrl** amb l'argument correcte quan actualitzem una categoria
- al component **PostsListComponent**
    - un test que validi que es crea el component correctament
    - un test que ens asseguri dos coses:
        - que quan es crida el **loadPosts** es llanci la crida **getPostsByUserId** del servei

        - que la resposta de la crida sigui la esperada

    - un test que validi que es crida el **navigateByUrl** amb l'argument correcte quan creem un post

    - un test que validi que es crida el **navigateByUrl** amb l'argument correcte quan actualitzem un post

Exercici 5 – test vista

Investigar com es poden testejar elements de la vista de manera que haurem testejar que:

- quan estem autenticats que existeixin els punts de menú home, admin posts, admins categories, profile i logout

- quan no estem autenticats que existeixin els punts de menú home, login i register.

Puntuació

A continuació mostrem la puntuació de cada apartat de la pràctica:

- Exercici 1**[2 punts]**
- Exercici 2**[2 punts]**
- Exercici 3**[3 punts]**
- Exercici 4**[2 punts]**
- Exercici 5**[1 punt]**