

PAC 7

Frameworks: Routing a Angular

UOC

Universitat Oberta
de Catalunya

Informació rellevant:

- Data límit de lliurament: 30 de juny.
- Pes a la nota de FC: 15%.



Contingut

Informació docent	3
Presentació	3
Objectius	3
Enunciat	4
Exercici 1 – Preguntes teòriques (3 punts)	4
Exercici 2 – Pràctica – Routing (5.5 punts)	6
Exercici 3 – Pràctica sobre Lazy-Loading (1.5 punts)	11
Format i data de lliurament	13

Informació docent

Presentació

Aquesta pràctica se centra a conèixer la creació de rutes a Angular, per això es farà ús del mòdul `Router` del `core` d'Angular. A més, es protegiran rutes en funció de si l'usuari està loguejat o no al sistema. Finalment, es mantindrà la sessió de l'usuari basat en tokens d'autenticació.

Objectius

Els objectius que es volen assolir amb el desenvolupament d'aquesta PAC són:

- Comprendre el funcionament del **router a Angular**.
- Implementar un **sistema de rutes**.
- Construir una aplicació composta per **diversos components i serveis repassant conceptes** de pràctiques anteriors.
- **Protegir rutes** utilitzant **guardes**.
- Conèixer i fer ús de **Lazy-Loading i estructuració de mòduls**.
- **Mantingueu la sessió** d'un usuari basat en **tokens d'autenticació**.

Enunciat

Aquesta PAC **conté 3 exercicis avaluables**. Heu de lliurar la vostra solució dels 3 exercicis avaluables (veure l'últim apartat).



Com que les activitats estan encadenades (i.e. per fer-ne una s'ha d'haver comprès l'anterior), **és altament recomanable fer les tasques i els exercicis en l'ordre en què apareixen en aquest enunciat**.

Abans de continuar has de:

Haver llegit els recursos teòrics disponibles a l'apartat "Continguts i recursos" de l'aula d'aquesta PEC

Crea una carpeta PAC7 NO ÉS NECESSARI INICIALITZA git en aquesta ruta, els commits es poden fer únicament al projecte ANGULAR. Igual que a les anteriors PAC, has de crear un fitxer README.md a l'arrel de la carpeta que contingui que contindrà almenys aquesta informació:

- Login UOC.
- Nom i cognoms de l'alumne.
- Breu descripció del que s'ha realitzat en aquesta PAC, dificultats, millores realitzades, si cal tenir alguna cosa en compte a l'hora de corregir/executar la pràctica o qualsevol aspecte que vulgueu destacar.

Exercici 1 – Preguntes teòriques (3 punts)

Crea una carpeta PEC7_Ej_Teor, dins d'aquesta carpeta crea un fitxer Markdown que tingui com a nom PEC7_Ej1_respuestas_teoría.md i **respon** a cadascuna de les preguntes següents:

- Què és i com funciona l'element `<RouterLink>` en Angular?
- Explica la diferència entre `routerLink` i `routerLinkActive`. Quines altres directives es poden utilitzar amb el router en Angular?

- c) Descriu el servei `ActivatedRouteSnapshot`. Com s'utilitza i en quins casos és útil?
- d) Què és la càrrega Lazy dels mòduls d'Angular? Com es configura en Angular la càrrega Lazy? (<https://angular.io/guide/lazy-loading-ngmodules>)
- e) Compara les diferències entre `CanDeactivate` i `CanActivate` guards en Angular. Proporciona exemples de quan s'utilitzaria cadascuna.
- f) Què és/per a què són útils els middlewares en el context d'Angular? On estàs utilitzant middlewares en la nostra aplicació?

Exercici 2 – Pràctica – Routing (5.5 punts)

Crea una carpeta `PEC7_Ej_Prac` partint de l'exercici pràctic ecommerce realitzat a la PAC anterior (o de la solució publicada) i realitza les tasques següents:

Instal·la i executa el nou servidor que se t'ha proporcionat per a aquesta PAC7 (server-articles) executant el següent:

- `npm i`
- `npm start`

Això farà que arrenqueu un servidor local de node.js, el qual estarà treballant amb articles (similar al que heu vist en els apunts sobre estocs). Aquest servidor exposa les següents APIs, al port 3000 (`http://localhost:3000/api/articles`):

- `GET a api/articles` per obtenir una llista d'articles. Aquesta pot tenir un paràmetre opcional de consulta `q`, el qual és el nom de l'article a cercar.
- `POST a api/articles` amb la informació d'un article al cos per crear un vi al servidor (en el nostre cas serà en memòria, reiniciant el servidor es perdran totes les dades creades).
- `PATCH a api/articles/:id` amb l'ID de l'article a la URL i un camp `changeInQuantity` al cos canviarà la quantitat al carret d'articles per la quantitat passada com a paràmetre.

A més tindràs dos nous ENDPOINT que corresponen a la gestió d'usuaris (falsa) per poder realitzar el registre i l'autenticació d'usuaris al nostre sistema:

- `POST a /auth/login` rep l'username i password de l'usuari i comprova si l'usuari és al sistema i la contrasenya és la correcta.

- POST **a** /auth/register rep l'username i comprova que no s'ha repetit, s'assigna la contrasenya "SECRET" a tots els usuaris.

Podeu comprovar el correcte funcionament de l'API fent ús de l'eina Postman o Insomnia.

NOTA: Si necessites ajuda sobre el funcionament de POSTMAN pots consultar el següent recurs:

https://learning.oreilly.com/videos/postman-tutorial-getting/9781803243351/9781803243351-video2_3/

Un cop instal·lat i comprovat el funcionament del servidor local de node.js, realitza els apartats següents:

a) **Repasant tot allò après (1.75 punts)**

Sobre el projecte angular has de realitzar les tasques següents utilitzant els coneixements adquirits al llarg de l'assignatura:

1. Crear els components següents:

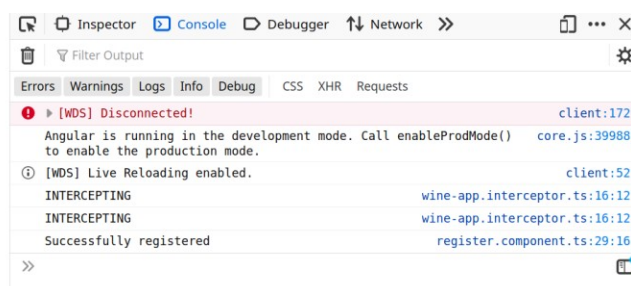
- **login:** (Formulari reactiu). Aquest formulari està compost per dos inputs (username i password) i realitzarà l'autenticació dels usuaris al sistema.
- **register;** (Formulari reactiu). Aquest formulari està compost per dos inputs (username i password) i realitza el registre dels usuaris al sistema.
- **article-detail:** Aquest component mostrarà els detalls de cada article, s'accedirà a aquest detall fent clic sobre la imatge de cada article al component article-item. A la ruta s'accedirà a través de la variable "id" de cada article (valor que ha de ser únic)

2. Crear el servei `auth` el qual fa trucades HTTP per realitzar l'autenticació (login) i registre (register) dels usuaris.
3. Crear el servei `auth-store` el qual emmagatzema si l'usuari està autenticat a l'aplicació o no, fent ús d'un token d'autenticació (serà fals i no cal fer res al backend).
4. Crear un interceptor `auth-interceptor` que enviarà el token d'autenticació si aquest existeix a cadascuna de les peticions HTTP.
5. Registrar tots els elements anteriorment citats al mòdul AppModule.

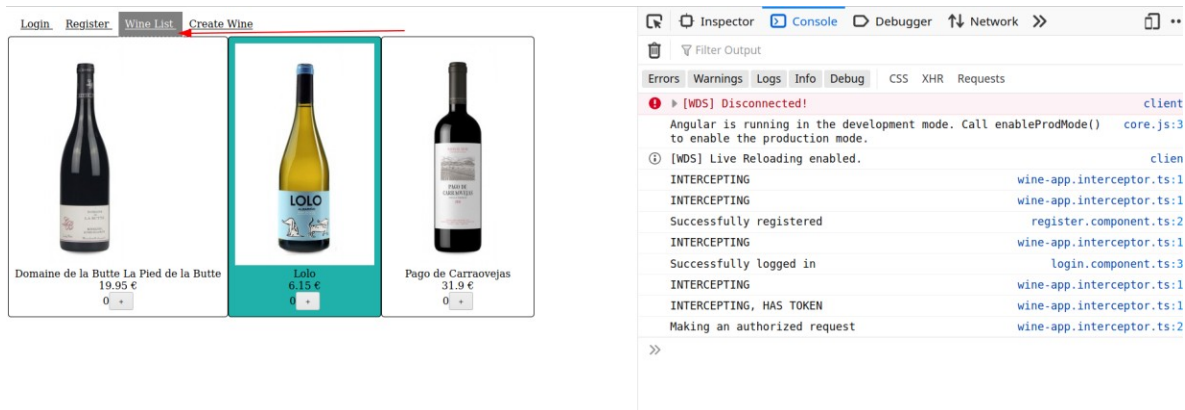
b) Configurar Routing i esquema de rutes a l'aplicació (1 punt)

Modifica el component navbar per tenir les següents opcions (Login, Register, Articles list, Create Article), també configura el mòdul de routing de manera adequada a la nostra aplicació perquè les diferents opcions de menú vagin a les següents rutes:

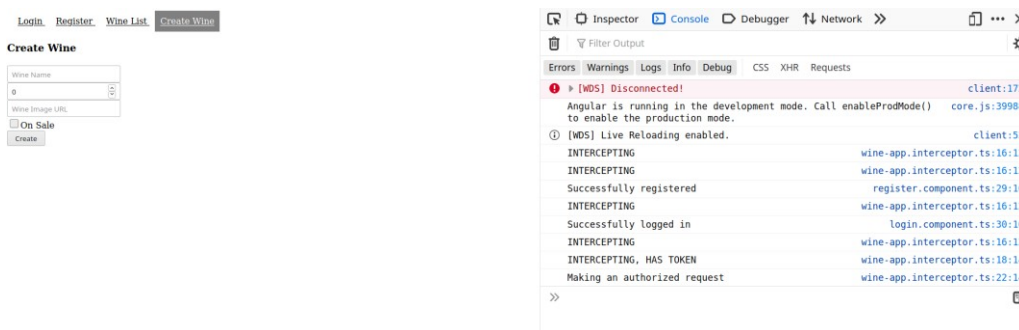
- La ruta per defecte serà login.
- Opció de menú login, path: 'login', component: login
- Opció de menú Register , path: 'register', component: register



- Opció de menú Article List path: article/list, component: article-list



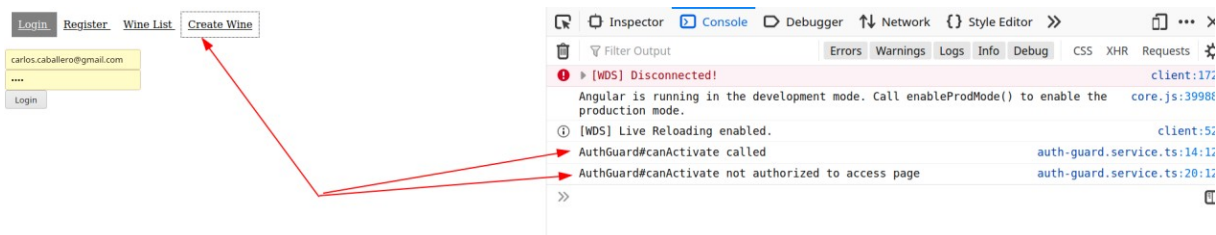
- Opció de menú Create Article path: article/create, component: article-new-reactive



- Detall de l'article: path: article/:id, component: article-detail

c) Protegint rutes (1.5 punts)

- La ruta per crear un article haurà d'estar protegida i accessible després d'autenticar-se (login).

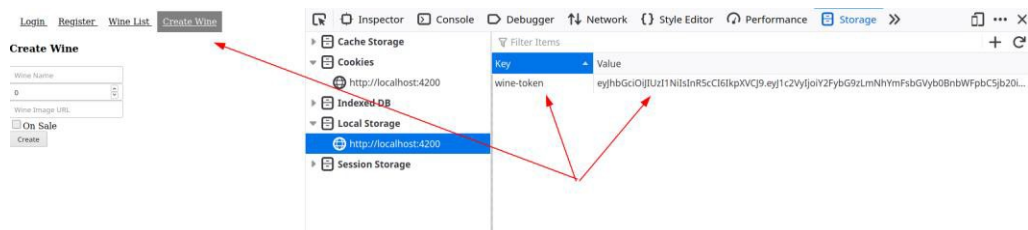


d) Remember login (1.25 punt)

Adapta el codi de manera que el flux d'autenticació de l'usuari recordi si l'usuari està autenticat al sistema quan refresquem la pàgina.

És a dir, si l'usuari està autenticat s'haurà d'emmagatzemar el token d'autenticació al sistema de manera que puguem recuperar-lo en refrescar la pàgina. En cas que l'usuari estigui autenticat no es mostrarà la pàgina de login, però en cas que no estigui autenticat, es mostrarà la pàgina d'autenticació (login).

Per exemple si ja ens hem arribat, en accedir a Create Article no se'ns mostrarà de nou el formulari de login.



Un cop haguem conclòs tots els apartats, l'estructura de directoris del nostre projecte hauria de ser una cosa semblant a això:



Durant el desenvolupament de l'exercici executa les ordres `git` necessàries per afegir aquesta nova carpeta, fitxers i els canvis que hi facis al repositori local `git` de la PAC.

La carpeta **node_modules** i la carpeta **.angular**, no ha d'aparèixer al repositori `git` local, ni tampoc lliurar-la al zip final.

Exercici 3 – Pràctica sobre Lazy-Loading (1.5 punts)

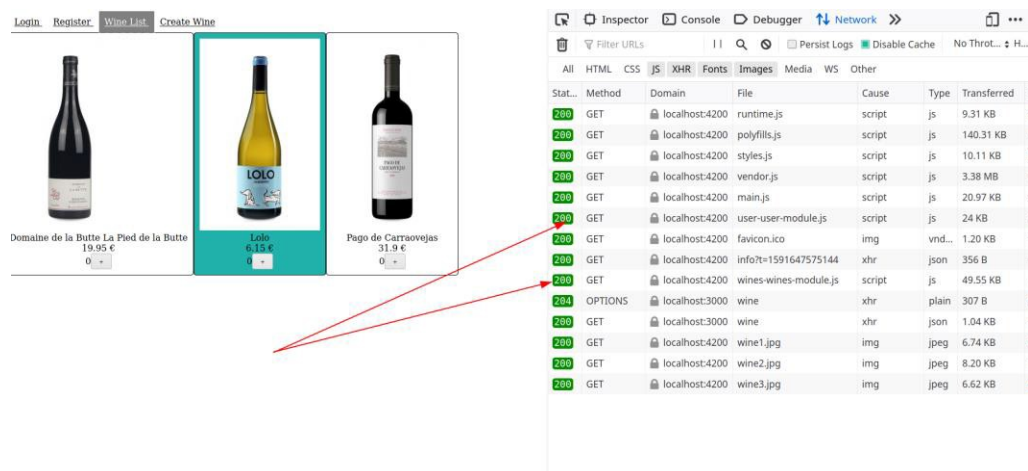
Continuem amb el nostre exercici pràctic d'**ecommerce**, realitza les modificacions necessàries per disposar de dos mòduls clarament diferenciats:

- Auth
- Article

Els quals es carreguen de manera lazy en funció de si es requereix un mòdul o un altre. La nova estructura de directoris s'ha de transformar de manera que ara tinguem un espai de models, serveis, guardes que es puguin considerar shared i d'altres que siguin específics per a cadascun dels mòduls concrets.

Realitza els canvis al codi per disposar de Lazy-Loading.

A la carpeta **ecommerce** incloure una captura de pantalla de l'inspector d'elements on es visualitzi clarament que els mòduls s'estan carregant de manera Lazy.



Durant el desenvolupament de l'exercici executa les ordres `git` necessàries per afegir aquesta nova carpeta, fitxers i els canvis que hi facis al repositori local `git` de la PAC.

La carpeta `node_modules` i la carpeta `.angular`, no ha d'aparèixer al repositori `git` local, ni tampoc lliurar-la al zip final.

Format i data de lliurament

Has de lliurar un fitxer *.zip, el nom del qual ha de seguir aquest patró: loginUOC_PEC7.zip. Per exemple: dgarciaso_PEC7.zip. Aquest fitxer comprimit ha d'incloure els elements següents:

Una carpeta amb nom PEC7, i al seu interior:

- Un fitxer **README.md** a l'arrel de la carpeta amb la informació indicada.
- Una carpeta PEC7 Ej Teor amb els fitxers markdown sol·licitat per als exercicis amb explicacions teòriques:
 - Fitxer markdown **PEC7_Ej1_respuestas_teoría.md** per a les respostes de l'exercici 1.
- Una carpeta PEC7 Ej Prac amb el repositori git i els fitxers amb el resultat d'haver realitzat les tasques dels exercicis 2 i 3 (No oblidis eliminar les carpetes node_modules i .angular)

Penalitzacions

- Lliurament en un altre format que no sigui l'especificat (ex. zip): **-0.75 punts**
- Comprimir fitxers dins del zip: **-1 punts**.
- Per a cada exercici/apartat on no es respecti la nomenclatura exacta de les carpetes o fitxers indicats (símbols, minúscules, majúscules, etc.): **-0.75 punts**.
- La no entrega del repositori local git **-3 punts**
- Per cada carpeta node_modules o .angular lliurada **-0.75 punts**

El darrer dia per lliurar aquesta PAC és el **30 de juny 2024** fins a les **23:59**. **En ser aquesta la darrera, qualsevol PAC fora de termini, no serà corregida.**