ENUNCIAT PEC 5

PWA, Angular Universal, Deploy

Desenvolupament front-end avançat

Máster universitari en desenvolupament de llocs i aplicacions web

Universitat Oberta de Catalunya

Contingut

Introducció

En aquesta pràctica implementarem una aplicació lliure que consumeixi una api pública i posteriorment la transformarem a **PWA**.

Finalment, desplegarem aquesta aplicació a **Github**.

Format i data d'entrega

S'entregarà tot el projecte comprimit en formato **.zip** sense incloure els directoris **"node_modules"** i **".angular"** juntament amb un documento de text responent a les dos/tres tasques escrites que es demanen. La data d'entrega es mostrarà a l'aula de l'assignatura.

Enunciat

Exercici 1 – Implementar una PWA

En aquest primer exercici implementarem una aplicació completa lliure i la transformarem a **PWA**.

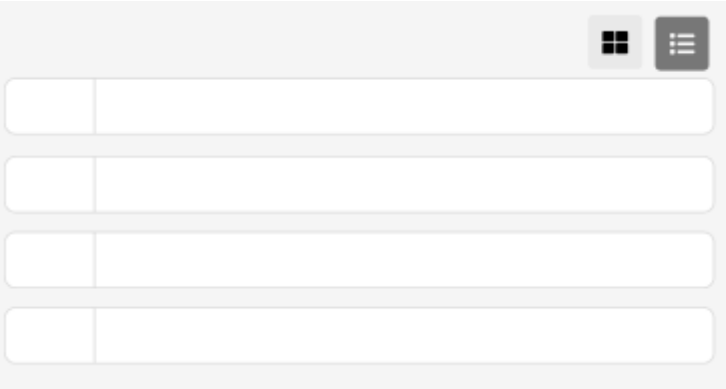
Podem seguir els passos dels materials de la teoria del tema 5. La idea seria buscar una api pública que puguem consultar, que retorni un llistat d'alguna entitat, i implementar una aplicació llista-detall tipus hem estudiat a la teoria (amb la api de imatges) i un cop feta aquesta implementació passar-la a **PWA**.

Haurem de configurar el **manifest** i tot el necessari del **service worker** per a que pugui instal·lar-se i funcionar sense internet.

Haureu de pensar en quina estratègia de cache seguir en funció dels recursos que tingueu.

La aplicació haurà de tindre almenys:

- Component-list:
    - Una **home** on aparegui el llistat de l'entitat retornada per la api escollida utilitzant una **taula** o un llistat de **cards** de **Angular Material**
        - Farem la següent implementació:



A la part superior dreta, implementarem dos botons per intercanviar la vista "**cards**" de la vista "**taula**".

    - Mentre esperem el llistat mostrarem l'**spinner** de **Angular Material**
    - L'aparició del llistat la farem amb alguna animació (elecció lliure)
    - Al fer clic a un element de la llista navegarem a la pàgina del detall

- Component-detail:
    - Pàgina on mostrarem els detalls de la entitat.

- Aquesta pantalla dependrà de cada api utilitzada. Seria important que la api retorni algun tipus de imatge per a poder comprovar com es guarda a la cache.

- A la part superior mostrarem només un camp de la entitat, el nom o el títol (una propietat que sigui representativa de la entitat).

- A la dreta mostrarem el botó de "**back**" per poder tornar a la **home**

- A sota mostrarem un botó "**show all details**", al polsar el botó:

  - Es mostraran la resta de les propietats de l'entitat

  - Per a mostrar la resta de les propietats haurem d'utilitzar almenys un dels següents elements d'**Angular Material**:
    - **Tree**
    - **Tabs**
    - **Expansion panel**
    - **Progress bar**
    - **Slider**

  - *Si veiem que a la resposta de la api ens falta informació per a poder utilitzar algun dels components anteriors ens podem inventar algunes propietats "afegint-les" al array de resposta "manualment".*

Implementació de components base

Haurem d'implementar almenys els següents components base:

- Component **card** per reaprofitar al llistat de **cards** de la **home**. Al fer clic a la *card* anirem al detall de la entitat.

- Component **grid** per reaprofitar al llistat de la **home** per quan polsem el botó "modo tabla". Al fer clic a la fila anirem al detall de la entitat.

Podeu implementar més components base en funció dels vostres requisits.

*Un cop implementada, haureu d'adjuntar un petit document amb alguns comentaris bàsics de la informació que ens mostra l'informe generat per la eina **LightHouse** de **Google Chrome** i les decisions de disseny més rellevants en quan a configuració del **service worker**.*

Exercici 2 – Desplegar una aplicació Angular en GitHub

Una cop tenim una aplicació **Angular** finalitzada, la pujarem a un servidor remot en mode producció. Per a pujar una aplicació **Angular** en mode producció disposem de diverses opcions:

- Si estiguéssim en un entorn real, hauríem de copiar tots els fitxers generats a un servidor web. Només tenim que crear un **build** en mode producció de la nostra aplicació utilitzant la següent comanda:

- **ng build --outputHashing=all**

  - el **flag outputHashing=all** es per a forçar al navegador a que netegi la cache i es quedi amb la última versió del codi, això es súper útil ja que, si nosaltres tenim una **app** amb molts usuaris i anem desplegant noves versions, no podem dir als usuaris, escolta! Neteja la cache per a tenir la última versió. Amb aquest **flag**, forcem a que netegi cache i que el navegador treballi amb la última versió del codi.

  Nota: Teniu en compte que poden aparèixer problemes que en la fase de desenvolupament no existien al fer el **ng build**.

  A continuació, haurem de copiar el directori **dist/** al nostre servidor web (**Apache**, **nginx** o el que haguéssim seleccionat per a tal fi).

- Ara que estem estudiant, podríem desplegar els nostres projectes a pàgines **GitHub**. Una aplicació **Angular** esta composada per fitxers estàtics **HTML**, **JavaScript** y **CSS** que poder ser servits per qualsevol servidor web. Per això podem desplegar aplicacions **Angular** transpilades a **Github**, ja que el contingut que es genera és estàtic. Per tant, en aquest exercici es demana que despleguem la vostra aplicació a Github. Podeu fer-ho de dues maneres:

  - Seguint aquest tutorial podeu desplegar a GitHub de la manera “tradicional”:

    Haciendo deploy de una app en angular a GitHub Pages - DEV Community

  - Una altra alternativa seria utilitzar **Netlify**, aquí un tutorial:

    Deploying Angular app with Netlify in 3 steps - DEV Community

Quan tingueu l’aplicació funcionant a **Github**, mireu d’accedir des dels vostres telèfons i valideu que funciona com **PWA**, que os demani d’instal·lar-se i que funcioni correctament.

*Afegiu un document de text amb la **url** resultant per a poder revisar l’aplicació desplegada.*

Puntuació

A continuació, mostrem quan puntuen cada un dels apartats de la pràctica per obtenir la nota final:

- Exercici 1**[8 punts]**
- Exercici 2**[2 punts]**