

UserManagement

This project was generated with [Angular CLI](#) version 13.2.3.

Development server

Run `ng serve` for a dev server. Navigate to `http://localhost:4200/`. The app will automatically reload if you change any of the source files.

Code scaffolding

Run `ng generate component component-name` to generate a new component. You can also use `ng generate directive|pipe|service|class|guard|interface|enum|module`.

Build

Run `ng build` to build the project. The build artifacts will be stored in the `dist/` directory.

Running unit tests

Run `ng test` to execute the unit tests via [Karma](#).

Running end-to-end tests

Run `ng e2e` to execute the end-to-end tests via a platform of your choice. To use this command, you need to first add a package that implements end-to-end testing capabilities.

Further help

To get more help on the Angular CLI use `ng help` or go check out the [Angular CLI Overview and Command Reference](#) page.

Instructions

NB : Le but de cet exercice est de mesurer les capacités du candidat par rapport à la compréhension du projet et des notions de base d'angular.

L'importance de certaines notions lui seront apprises à la fin de cette activité.

Le candidat doit suivre autant que possible les recommandations de cette activité et répondre aux différentes questions au fur et en mesure de son évolution dans un fichier docx fait de préférence avec google doc.

Il devra suivre la structure des composants à la lettre pour rejoindre le résultat escompté.

Des précisions lui ont été ramenées sur le résultat attendu en terme de ui.

S'il a un problème d'incompréhension, il pourra chercher à contacter le plus rapidement la personne agréée.

- créer un composant user grâce à angular cli dans un répertoire `components/` dans ce composant la structure sera constitué d'une barre de recherche avec 2 ordres de disposition un pour les listes et un

autre en carte avec un bouton de création d'utilisateur, un filtre par les rôles et de tout l'affichage de l'utilisateur (en liste ou carte), ce composant contiendra d'autres composants qu'il créera et importera au fur et en mesure de l'évolution de cet activité. N'oubliez pas de l'importer dans `app.component.html` pour avoir le résultat escompté.

1. Quelle est la commande cli utilisée ?

Résultat attendu: même interface qu'au départ

Faire un commit avec pour message `composant user créé`

- Dans le composant `user`, créer un composant `user-list` grâce à angular cli dans ce composant la structure sera d'afficher la liste d'utilisateurs en liste/tableau et importer `user-list` dans `user` selon les recommandations.

2. Quelle est la commande cli utilisée ?

Résultat attendu: même interface qu'au départ

Faire un commit avec pour message `composant user-list créé`

- Dans le composant `user`, créer un composant `user-list-card` grâce à angular cli dans ce composant la structure sera d'afficher la liste d'utilisateurs en carte avec au moins une carte et importer `user-list-card` dans `user` selon les recommandations.

3. Quelle est la commande cli utilisée ?

Résultat attendu: même interface qu'au départ

Faire un commit avec pour message `composant user-list-card créé`

- créer un modèle `user.models.ts` comme étant une interface suivant les recommandations utilisateurs et faire un commit avec pour message `modèle user créé`

4. A quoi sert de créer un modèle ?

- créer des données fictives pour la liste des utilisateurs (avec au moins 5 utilisateurs) dans le composant `user` et propager ses données à travers tous les composants, en utilisant :
 - l'interface créé dans `user.models.ts`
 - la directive structurelle approprié
 - le property binding

Résultat attendu: même interface qu'au départ avec des lignes ajoutées dans le tableau et plusieurs cartes qui s'affichent pour les différents utilisateurs

5. Quelle directive structurelle avez vous utilisé ?

6. C'est quoi le property binding ?

Faire un commit avec pour message `affichage de la liste des utilisateurs en carte et en liste sans condition préalable`

- Permettre l'affichage conditionnée des différents dispositions (en carte ou liste), en utilisant :

- la directive structurelle approprié
- property binding

Résultat attendu: le tableau d'utilisateur qui s'affiche lorsqu'on clique sur le bouton qui permet l'affichage en liste (par défaut c'est ce qui s'affiche en premier même quand on n'a pas encore effectué de clic), faire de même pour la disposition en carte.

7. Quelle est la directive structurelle utilisée ?

Faire un commit avec pour message `affichage conditionnée en liste ou en carte`