

Business Unit et MLD

Examen / Projet : cours Angular, EPSI juin 2019 – En cours à finir à la maison

Une entreprise de papiers fabrique et vend ses produits.

Elle souhaite disposer d'une interface graphique faite en angular pour la gestion de ses produits. Le projet étant en angular uniquement, nous accepterons que tout se passe en angular et que les données soient perdues à chaque déconnexion. Cependant, il faudra prévoir les services fournissant la donnée comme s'ils appelaient une API, de manière synchrone pour la première partie de ce travail.

L'entreprise souhaite disposer d'une page permettant de saisir et modifier l'ensemble de ses produits.

Un produit est caractérisé par les propriétés suivantes :

- Id
- Nom
- Texture,
- Grammage,
- Couleur

Dans un premier temps, ces attributs sont de type « texte libre ».

La page doit être constituée de la forme suivante :



Le design n'est pas important, mais le cadre doit bien être présent pour visualiser les différentes sections.

Questions : *(le barème est donné à titre indicatif, il pourra être ajusté lors de la correction)*

- Produire le système permettant de lister les produits (une simple liste par le nom du produit sera suffisante). La liste sera initialisée avec 2 produits (produit 1 et produit 2 par exemple). Le produit sélectionné devra être mis en évidence par un style particulier.
- Produire le système permettant de mettre à jour un produit

- Produire le système permettant de bloquer la sélection de l'utilisateur pendant la modification d'un produit (ie l'utilisateur a commencé à modifier un produit, on ne veut pas pouvoir modifier le produit en cours)
- Produire le système permettant d'ajouter un produit. L'id du produit sera égal à l'ID max du dernier produit existant auquel on ajoute 1.

Notes importantes :

- Les premiers tests qui seront faits seront « fonctionnels », les bugs seront bien évidemment pris en compte dans la note. Un soin particulier apporté au design pourra améliorer la note.
- L'organisation des développements, à savoir le bon découpage des rôles, sera également noté
- Important : cet examen sera rendu via git, vous devrez créer un projet sous github et me le partager (thibaut.montanier@gmail.com). Merci de me faire un mail afin que je puisse savoir quand vous aurez terminé. Le partiel sera rendu sur la branche « master ».

thibaut-montanier / AngularTennisPlayer Private

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Security Insights Settings

Options Collaborators 1 Branches Webhooks Notifications Integrations & services Deploy keys

Collaborators Push access to the repository

This repository doesn't have any collaborators yet. Use the form below to add a collaborator.

Search by username, full name or email address

You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

2 Add collaborator 3

0 of 3 collaborators

Business Unit et MLD – partie 2

La seconde partie du projet consiste à mettre en œuvre :

- Le routing
- Amélioration du fonctionnement en utilisant mieux les services
- La programmation réactive (en utilisant l'asynchrone et ReactiveX).

Pour cela, je vous demande de :

1. Créer une page qui s'occupera uniquement du listing
 - a. Cette page affiche uniquement la liste des produits
 - b. Au clic sur un produit, on doit être redirigé sur une page affichant uniquement le détail d'un produit
 - c. Elle contiendra un bouton (ou un lien) « créer » qui affiche le formulaire de modification vierge
2. Créer une page qui affichera le détail d'un produit (et qui permettra de sauvegarder les modifications)
 - a. Cette page permettra de modifier un produit
 - b. Elle permettra également de créer un nouveau produit (en passant 0 comme identifiant via l'url par exemple)
3. Mettre en place le routing
 - a. Créer un menu qui permet d'accéder aux pages suivantes :
 - i. Affichage de la liste des produits, avec le détail en dessous (page faite dans la première partie du projet)
 - ii. Affichage du listing des produits (nouvelle page créée dans cette seconde partie du projet)
 - b. Créer dynamiquement le menu, sans passer par l'html
 - i. Idéalement, retirer le /produit/:id
4. Service : faire passer l'ensemble des ajouts / modifications par le service
 - a. => Il faut créer une méthode update qui prend en paramètre un produit qui fera l'ajout ou la mise à jour
5. Mettre en place l'asynchrone : la liste des produits sera fournie de manière asynchrone, de la même manière que les informations d'un produit.