TD Todo List - Partie 03

React

Fonctionnalités avancées

Les fonctionnalités avancées ci-après, sont communiquées à titre indicatif, leur réalisation est facultative mais recommandée pour vous permettre de vous exercer en vue de la réalisation de votre projet collectif à venir.

Réalisez ces fonctionnalités uniquement si vous avez déjà réalisé les fonctionnalités des parties 01 et 02.

11 : Composition

- A ce stade, l'application repose intégralement dans le fichier ***App.tsx*** ce qui n'est pas une bonne pratique pour la lisibilité, la maintenabilité et la réutilisabilité de l'application.
  Décomposez l'application en composants indépendants et organisez la collaboration entre composants, ex :
  - La liste dynamique peut être affichée par un composant ***TasksMaster***,
  - Chaque item affiché dans ***TasksMaster*** peut être géré par un composant ***TaskPreview***.
  - Le formulaire d'ajout peut être géré dans un composant ***TaskForm***.
- Dans le fichier ***App.tsx***, procédez à l'import de ces nouveaux composants afin de reconstituer l'application. Employez les attributs ***props*** de chaque composant pour faire circuler les données entre composants. Employez un ***type*** ou une ***interface*** TypeScript pour typer les ***props*** de chaque composant concerné.
- Mettez en place des ***props*** de type ***fonction***, pour organiser la collaboration entre composants sur un mode événementiel :
  - un composant *A* passe en paramètre ***props*** d'un composant *B* une référence vers une fonction définie dans le composant *A*.
  - Lorsqu'un événement se produit au sein du composant *B*, celui-ci est en mesure d'appeler la fonction du composant *A*, et potentiellement de lui transmettre des données. La fonction du composant *A* joue le rôle de ***fonction callback***.

Etape 12 page suivante

12 : Suppression

- Permettez à l'utilisateur de supprimer les items de type ***Task*** uniquement lorsque leur statut est égal à ***Done***.
- En option, affichez un message de demande de confirmation à l'utilisateur avant de supprimer l'item concerné.

Etape 13 page suivante

13 : Routage

- Permettez à l'utilisateur de consulter la liste des items de type ***Task*** en version ***TaskPreview*** dans un écran ***TasksMaster*** (toutes les tasks) et en version détaillée (ex: date de création, mise à jour…) dans un écran ***TaskDetails*** (1 item ***Task*** sélectionné selon son ***id***).
- L'utilisateur doit pouvoir naviguer de l'*URI* "/" à "/tasks/{taskId}" (et inversement).
- Pour ce faire, importez et utilisez le module *NPM* ***React Router*** en ***mode déclaratif*** (https://reactrouter.com/start/declarative/installation).

Etape 14 page suivante

14 : Formulaire

- Importez et utilisez le module *NPM* ***React Hook Form*** (https://react-hook-form.com/) afin de mettre en place un formulaire d'ajout d'item de type ***Task*** en remplacement de la solution avec le prompt *JavaScript* mise en place précédemment (cf. *étape 8, Partie 2*).

Fin du projet

Fonctionnalités bonus facultatives
dans un document distinct (à demander)