

[Marquer comme terminé](#)

TP 5

Les tâches de ce TP sont à traiter dans l'ordre qui vous convient. En fin de séance, vous devez avoir un écran de listant les TodoList, permettant de les détruire individuellement et d'en ajouter une nouvelle.

- reprenez votre application du TP précédent
- sur l'écran de la liste des TodoList, faites une requête sur l'API pour obtenir la liste (utilisez `useEffect()` pour déclencher ce chargement au moment où le composant est créé). La requête GraphQL est\ :

```
query TodoLists($where: TodoListWhere) {  
  todoLists(where: $where) {  
    id  
    title  
  }  
}
```

- ajoutez un composant générique `Input`, que vous stockerez dans `components/UI/`, qui contient un `TextInput` et un bouton de soumission. Inspirez-vous du code utilisé dans les TP précédents. Déterminer quels arguments ce composant doit recevoir par ses *props*. La soumission de cette entrée déclenchera l'enregistrement en base de données d'une nouvelle TodoList, à l'aide de la requête GraphQL suivante\ :

```
mutation createTodoLists($input: [TodoListCreateInput!]) {  
  createTodoLists(input: $input) {  
    todoLists {  
      id  
      owner {  
        username  
      }  
      title  
    }  
  }  
}
```

- faites en sorte qu'un icône de poubelle soit placée à droite du titre d'une liste et que le click sur la poubelle détruise l'item. Vous pouvez réaliser un composant générique en vous inspirant du code des TP précédents, qui contienne une liste de texte suivie de l'icône de poubelle.

Modifié le: lundi 9 septembre 2024, 11:32

[◀ CM 5](#)

Choisir un élément

Aller à...

[hierarchie.png ▶](#)

[mentions légales](#) . [vie privée](#) . [charte utilisation](#) . [unicaen](#) . [cemu](#) . [moodle](#)

[f](#) [t](#) [v](#) [@](#) [in](#)

?

