

TEST REACT JS

Afin de vérifier les compétences en React, une série d'exercices sont à effectuer, dans n'importe quel ordre, pour une meilleure organisation vous devrez créer un dossier, installer React à l'intérieur de ce dossier, et créer au moins 1 fichier par exercice (possibilité de découper davantage si beaucoup de composants).

I. Exercice 1

Installer React ainsi que Redux, et créer un composant comprenant un **formulaire**, contenant simplement un **bouton**, ainsi qu'un champ de **texte**. Les deux seront directement reliés à redux, et chaque modification sur le clavier ou la souris devront immédiatement être modifiés dans le state de redux.

PS : on peut appeler le champ du bouton "toggleButton" (qui peut prendre les valeurs true et false, et le champ texte "textField" qui lui sera une chaîne de caractère).

II. Exercice 2

Même exercice que précédemment, sauf que cette fois, au lieu de modifier directement le state global de l'application, il va falloir utiliser **useState** pour conserver l'état du formulaire en cours, et seulement lorsque l'on clique sur un bouton "envoyer", cela envoie les modifications au sein de l'état redux.

PS : <https://fr.reactjs.org/docs/hooks-state.html>

III. Exercice 3

Soit une variable nommée list, une liste d'objets (x , y), x et y deux nombres flottants, l'objectif est de créer une fonction qui prend en entrées la liste en question, et Z un nombre entier. L'objectif est de retourner l'objet de la liste donc la valeur x est le plus proche de Z.

Exemple : la liste est [{x:0, y:1},{x:1.2, y:5},{x:2.2, y:8}] et Z = 1, alors il faut retourner le deuxième objet

IV. Exercice 4

Créer une todo list, avec à gauche les éléments à **effectuer**, ainsi que le **temps** que cela prendrait, et à droite les éléments **effectués**, le **temps pris** pour chaque activité, ainsi que le **temps total** de toutes les activités effectuées

V. Exercice 5

Coder une calculatrice en React, avec tous les éléments principaux (somme, soustraction, addition, etc) et si possible des opérations plus complexes (carré, racine, cosinus, etc)