

Secteur Tertiaire Informatique
Filière « Etude et développement »

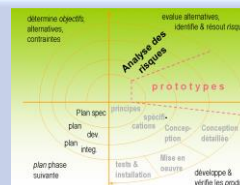
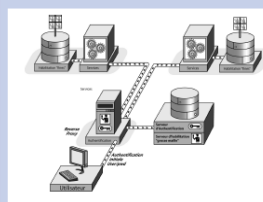
Développer en ReactJS

Création de composants

Apprentissage

Mise en situation

Evaluation



1. INTRODUCTION

La série d'exercices qui suit va vous permettre de mettre en place des composants React. Vous pourrez récupérer le code disponible à l'adresse suivante et compléter le projet : <https://github.com/ludovic-esperce/react-typescript-learning>

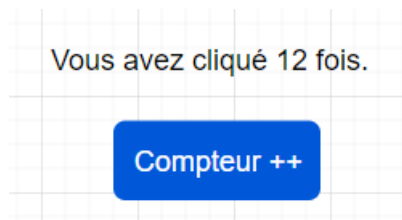
2. PAGES A DEVELOPPER

Vous allez pouvoir suivre les indications intégrées aux code (tous les « TODO ») pour mener à bien vos développements.

Ci-dessous vous trouverez des captures d'une proposition de résultat graphiques, sentez-vous libre de modifier le rendu.

2.1 COMPTEUR

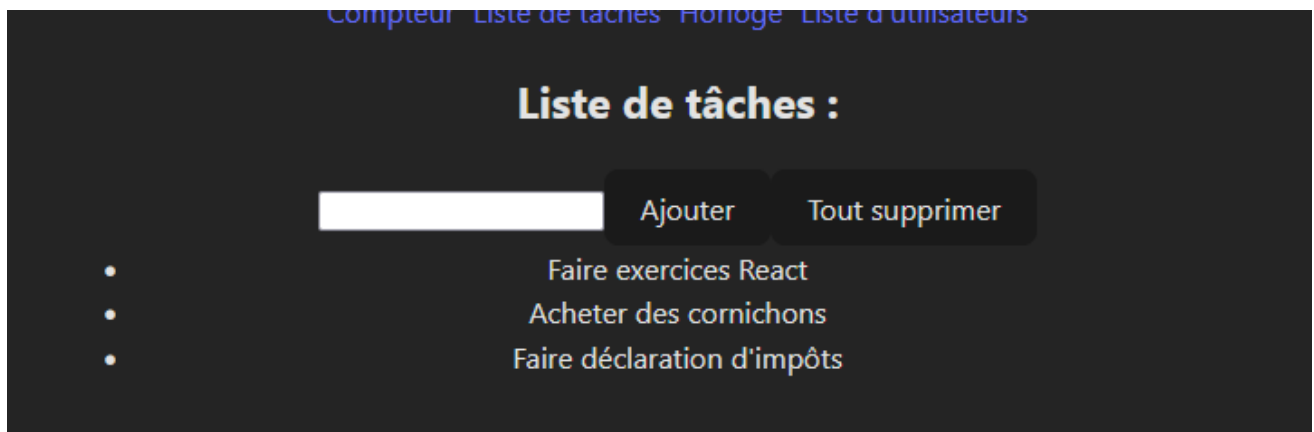
Le code à compléter se trouve dans le fichier « CounterPage.tsx ». Ci-dessous une maquette du résultat (le style intégré par défaut dans le projet ne permet, bien entendu, pas d'avoir le même rendu).



La valeur du compteur devra être déclarée en variable d'état : <https://blog.logrocket.com/a-guide-to-usestate-in-react-ecb9952e406c/>

2.2 LISTE DE TACHES

Implémentez un formulaire pour ajouter des tâches à une liste, voici un exemple de rendu :



La liste des tâches devra être déclarée en variable d'état (cf. « TODO » dans le code).

2.3 GENERATEUR DE MOTS ALEATOIRES

Cet exercice a pour objectif de mieux cerner le cycle de vie d'un composant en utilisant les hooks « `useEffect` » (<https://fr.reactjs.org/docs/hooks-effect.html>).

Ce type de hook peut être considéré comme un déclencheur appelant une fonction donnée suivant l'état d'un composant.

Le cycle de vie d'un composant est composé de trois étapes :

- **Mount** : le montage.

Il intervient quand une instance du composant est créée dans le DOM.

- **Update** : la mise à jour.

Cette phase est déclenchée par un changement d'état du composant.

- **Unmount** : le démontage.

C'est étape a lieu quand un composant est retiré du DOM.

Etudiez le code suivant et ajoutez des message utilisateurs en « `alert()` » à plusieurs niveaux (et uniquement en modifiant le code des fonctions anonymes passées aux « `useEffect` »).

Ajoutez l'affichage des messages suivants aux bon endroits du code :

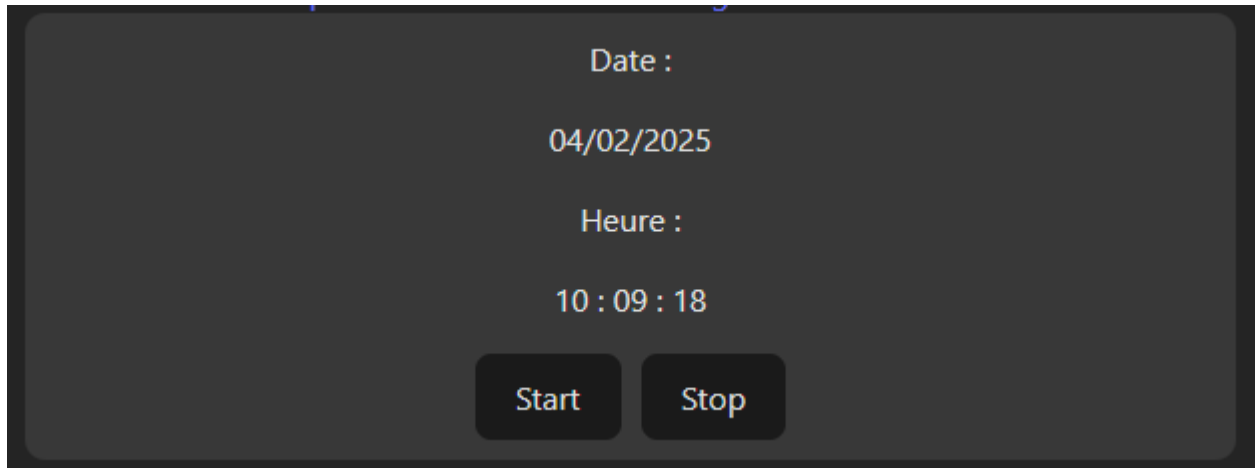
- « Clic sur le bouton de génération de mot 1 »
- « Clic sur le bouton de génération de mot 2 »
- « Clic sur un des boutons de génération de mot »

Veillez à ne pas afficher les alertes au chargement de la page.

2.4 HORLOGE

Développez un composant permettant d'afficher la date du jour ainsi que l'heure actuelle. Ces informations devront se mettre à jour grâce à un timer

Voici un exemple du résultat graphique :



Actions utilisateur à gérer :

- Un clic sur « Stop » arrête le déroulé du temps ;
- Un clic sur « Start » démarre l'horloge, les informations devront automatiquement se mettre à jour en fonction de l'heure au moment du démarrage.

Au chargement de la page l'horloge doit être en fonctionnement.

2.5 LISTE D'UTILISATEURS

Implémentez un composant représentant des utilisateurs sous forme de « card ». Les informations des utilisateurs proviendront :

- Dans un premier temps d'un fichier contenant un tableau d'utilisateur
- Dans un second temps d'une API en ligne (<https://jsonplaceholder.typicode.com/>)

Ci-dessous un exemple de résultat attendu :



CREDITS

ŒUVRE COLLECTIVE DE l'AFPA

Sous le pilotage de la DIIP et du centre d'ingénierie sectoriel Tertiaire-Services

Equipe de conception (IF, formateur, mediatiseur)

Michel Coulard – Formateur Evry

Chantal Perrachon – IF Neuilly sur Marne

Date de mise à jour : 03/02/2025

Reproduction interdite

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droits ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconque. »