

Secteur Tertiaire Informatique
Filière « Etude et développement »

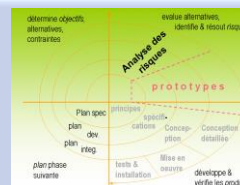
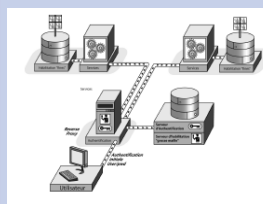
Développer en ReactJS

Création de composants

Apprentissage

Mise en situation

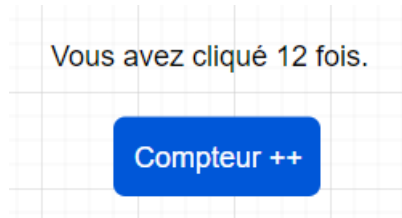
Evaluation



1. COMPOSANTS A IMPLEMENTER

1.1 COMPTEUR

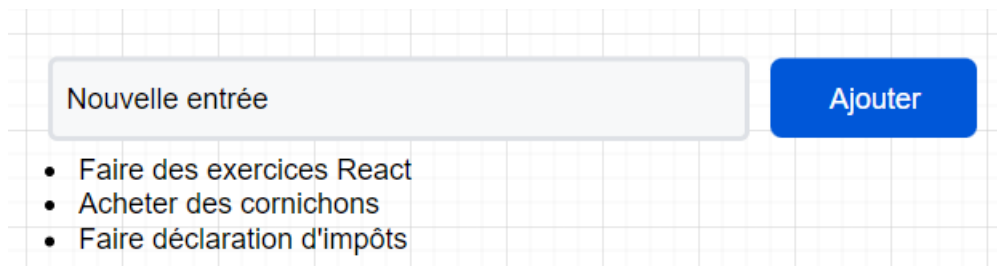
Implémentez un composant React permettant de compter le nombre de cliques sur un bouton. Il vous faudra utiliser un état pour stocker le nombre en question.



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

1.2 LISTE DE MOTS

Implémentez un formulaire pour ajouter des tâches à une liste :



La liste des tâches devra être déclarée en variable d'état.

1.3 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.

Utilisez le code suivant pour créer un nouveau composant :

```
const RandomWord = (props) => {
  console.log("Rendering component...");
  const [firstWord, setFirstWord] = useState("");
  const [secondWord, setSecondWord] = useState("");

  useEffect(() => {
    console.log("useEffect 4 ...")
  }, [firstWord]);

  useEffect(() => {
    console.log("useEffect 3 ...")
  }, [secondWord]);

  useEffect(() => {
    console.log("useEffect 2 ...")
  }, []) // tableau de dépendances à des variables d'état vide !

  useEffect(() => {
    console.log("useEffect 1 ...")
  });

  const handleClick1 = () => {
    setFirstWord(Math.random().toString(36).replace(/^[a-z]+/g, ''));
  };

  const handleClick2 = () => {
    setSecondWord(Math.random().toString(36).replace(/^[a-z]+/g, ''));
  };

  return (
    <>
      <div>
        {firstWord} - {secondWord}
      </div>
      <button onClick={handleClick1}>Changer premier mot</button>
      <button onClick={handleClick2}>Changer second mot</button>
    </>
  );
}
```

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.** = MOUNT

1.4 AFFICHAGE DE LA DATE ET DE L’HEURE

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

Voici un exemple du résultat à obtenir :

Date : 23/08/2022
Heure : 14:11:52

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.

1.5 LISTE D'UTILISATEURS

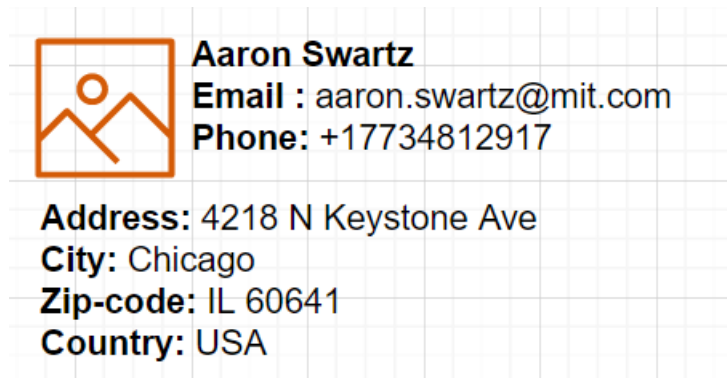
Implémentez un composant représentant des utilisateurs sous forme de « card ». Les informations utilisateur proviendront du requêtage de votre API Web développée précédemment.

1.5.1 Composant « card »

Par exemple, pour le JSon récupéré suivant :

```
{
  "id": 123,
  "firstName": "Aaron",
  "lastName": "Swartz",
  "email": "aaron.swartz@mit.com",
  "address": "4218 N Keystone Ave",
  "additionalAddress": "",
  "city": "Chicago",
  "zipCode": "IL 60641",
  "country": "États-Unis",
  "latitude": 41.95795013478338,
  "longitude": -87.72953277238744,
  "phone": "+17734812917" ?
  "picPath": null
}
```

Voici le résultat attendu pour une « card » (maquette) :



1.5.2 Composant « UserDirectory »

Il vous est maintenant demandé d'afficher tous les utilisateurs de provenant de votre webapi en utilisant un composant « UserDirectory » que vous implémenterez.

Ce composant devra avoir en « état » la liste des utilisateurs.

Maquette pour 3 utilisateurs :



Aaron Swartz

Email : aaron.swartz@mit.com

Tél. : +17734812917

Adresse : 4218 N Keystone Ave

Ville : Chicago

Code postal : IL 60641

Pays : États-Unis



Ada Lovelace

Email : aaron.swartz@mit.com

Tél. : +442031050125

Address: 61 Vallance Rd

City: London

Zip-code: E1 5AB

Pays : Angleterre



Jérémie Zimmermann

Email : jerem.zimmermann@freedom.fr

Phone: +17734812917

Adresse : 3 rue du Temple

Ville : Paris

Code postal : 75008

Pays : France

Exemple de contenu de la requête :

```
{  
  "firstName": "Aaron",  
  "lastName": "Swartz",  
  "email": "aaron.swartz@mit.com",  
  "address": "4218 N Keystone Ave",  
  "additionalAddress": "",  
  "city": "Chicago",  
  "zipCode": "IL 60641",  
  "country": "États-Unis",  
  "latitude": 41.95795013478338,  
  "longitude": -87.72953277238744,  
  "phone": "+17734812917"  
}
```

Pour effectuer des requêtes vers votre Rest API vous pourrez utiliser « fetch ». Pour une utilisation de fetch dans un composant React vous pouvez vous référer à la ressource suivante :

<https://www.freecodecamp.org/news/how-to-consume-rest-apis-in-react/>

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 : 26/08/2022

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. »