

TRAVAUX DIRIGÉS

Conception Web Avancée

Composants REACT & Formulaires

OBJECTIFS

- Gérer un formulaire avec React
- Lister des éléments avec React

I. RESULTAT ATTENDU

Nous souhaitons réaliser une application permettant de chiffrer des informations et d'en conserver le résultat.

Important

L'application développée ici est à visée pédagogique uniquement. Elle ne constitue absolument pas une bonne méthode pour stocker des éléments chiffrés.

L'application sera constituée de deux éléments principaux :

- une zone d'édition permettant de saisir le texte à chiffrer et d'afficher le résultat
- une liste recensant les éléments précédemment chiffrés

II. FORMULAIRE

- Réalisez le composant **EncryptionForm** qui contiendra :
 - un `<input type="password">` pour saisir la clé de chiffrement
 - un `<textarea>` pour saisir le texte à chiffrer
 - un `<label>` pour afficher le résultat du chiffrement
 - un `<button>` permettant d'enregistrer le résultat du chiffrement

Note

Dans un premier temps, vous ne coderez pas l'algorithme de chiffrement qui sera vu à la fin du sujet. Concentrez-vous sur React :-)

Le contenu du `<label>` reprendra donc tel quel, et en temps réel, le contenu du `<textarea>`.

Astuce

React propose un hook permettant de se lier à un élément du DOM :

```
export function TestComponent()
{
  const inputHTML = useRef<HTMLInputElement>(null);

  return (
    <input ref={inputHTML} type= "text" />
  )
}
```

- Créez un type **EncryptedData** contenant :
 - un attribut **key** de type **string**
 - un attribut **plain** de type **string**
 - un attribut **cypher** de type **string**
- Lorsque l'utilisateur cliquera sur le **<button>** du composant, ce dernier transmettra à son parent les données sous la forme d'un objet de type **EncryptedData**.

III. LISTE

- Créez un composant **EncryptedDataList** qui listera tous les **EncryptedData** enregistrés.
- Lorsque l'utilisateur cliquera sur un élément de la liste, ce dernier changera d'état pour montrer qu'il est sélectionné.
- Lorsqu'un élément de la liste est sélectionné, les données du composant **EncryptionForm** sont mises à jour.
- Lorsque les données du **EncryptionForm** sont enregistrées et qu'un élément de la liste est sélectionné, ses données sont mises à jour.

IV. LOCALSTORAGE

- Utilisez le **localStorage** pour garantir la persistance des informations de votre application.

V. CHIFFREMENT

- Implémenter le chiffrement. Pour cela vous avez le choix :
 - Coder un algorithme basique de chiffrement
 - Utiliser l'API Crypto disponible nativement en JS

Note

Le tutoriel de GrafikArt à ce sujet est pas mal : <https://grafikart.fr/tutoriels/web-crypto-javascript-1971>