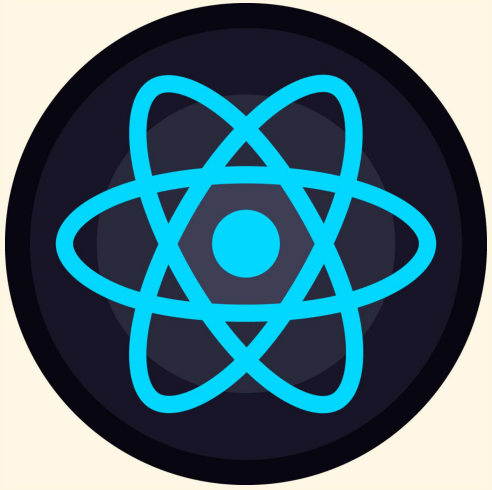
REACT : RECOIL



QU'EST-CE QUE RECOIL ?

- outil de gestion de l'état pour les applications *React*
- alternative aux options de gestion de l'état telles que *Redux* et *MobX*

INSTALLATION DE RECOIL

- Avoir au préalable installé React

```
npm install recoil
```

COMMENT FONCTIONNE RECOIL ?

- On définit des "**atoms**" qui représentent des morceaux d'état dans votre application
- Vous utilisez ces **atoms** dans vos *composants* en utilisant des "**selectors**"

COMMENT FONCTIONNE RECOIL ?

- Vous pouvez également définir des "**family selectors**" qui dépendent de plusieurs **atoms** et peuvent être utilisés pour calculer des valeurs à partir de ces **atoms**
- Vous pouvez également utiliser des "**hooks**" pour mettre à jour l'état de vos **atoms**

COMMENT FONCTIONNE RECOIL ?

RECOILROOT

- Les composants qui utilisent l'état recoil ont besoin que ***RecoilRoot*** apparaisse quelque part dans l'arborescence parente
  - Le mettre dans le composant racine

COMMENT FONCTIONNE RECOIL ?

```
import {
  RecoilRoot,
  atom,
  selector,
  useRecoilState,
  useRecoilValue,
} from 'recoil';

function App() {
  return (
    <RecoilRoot>
      <CharacterCounter />
    </RecoilRoot>
  );
}
```

COMMENT FONCTIONNE RECOIL ?

ATOME

- Représente un morceau d'état
- Peut être lu et écrit depuis n'importe quel composant
- Les composants qui lisent la valeur d'un atome sont implicitement abonnés à cet atome

Toute mise à jour d'atome entraînera un nouveau rendu de tous les composants abonnés à cet atome

ATOME

```
const textState = atom({
  key: 'textState',
  default: '',
});
```

ATOME

Les composants qui doivent lire et écrire sur un atome doivent utiliser ***useRecoilState()***

```
function CharacterCounter() {
  return (
    <div>
      <TextInput />
      <CharacterCount />
    </div>
  );
}

function TextInput() {
  const [text, setText] = useRecoilState(textState);

  const onChange = (event) => {
    setText(event.target.value);
  };
```

COMMENT FONCTIONNE RECOIL ?

SELECTEUR

- Représente un morceau d'état dérivé
  - L'état dérivé est une transformation d'état

SELECTEUR

On peut considérer l'état dérivé comme la sortie de l'état passant à une fonction pure qui modifie l'état donné d'une manière ou d'une autre

SELECTEUR

```
const charCountState = selector({
  key: 'charCountState', // unique ID (with respect to other atoms/selectors
  get: ({get}) => {
    const text = get(textState);

    return text.length;
  },
});
```

SELECTEUR

Nous pouvons utiliser le *hook* ***useRecoilValue()*** pour lire la valeur de ***charCountState***

```
function CharacterCount() {
  const count = useRecoilValue(charCountState);

  return <>Character Count: {count}</>;
}
```