

useState & useEffect

State et Cycle de vie





useState

A quoi sert le state React ?



Pourquoi et comment useState

Jetons un oeil à la doc:

<https://react.dev/learn/state-a-components-memory>



Pourquoi et comment useState

- On utilise un **state** lorsqu'on souhaite que l'interface **réagisse** au changement d'une valeur
- Pour cela on utilise le **hook useState**
 - On ne modifie **jamais** le state directement mais on le remplace par un nouveau



Un premier state dans notre projet

- Créons un bouton sous chaque annonce permettant d'ajouter le prix de l'annonce à un compteur global
- Le compteur global doit se mettre à jour au clic



Ajout du bouton

Dans le composant **RecentAds.tsx**, ajoute un bouton sous chaque composant et englobe l'**annonce** et le **bouton** dans une **div**



```
return (  
  <>  
    <h2>Annonces récentes</h2>  
    <section className="recent-ads">  
      {ads.map((ad) => (  
        <div key={ad.title}>  
          <AdCard  
            imgUrl={ad.imgUrl}  
            link={ad.link}  
            price={ad.price}  
            title={ad.title}  
          />  
          <button className="button">Add price to total</button>  
        </div>  
      )  
    )}  
    </section>  
  </>  
)
```



Création du state

Initialise un nouveau state à 0

TS RecentAds.tsx U X

src > components > TS RecentAds.tsx > [🔗] RecentAds

```
1  import { useState } from "react";
2  import AdCard, { AdCardProps } from "../AdCard";
3
4  const RecentAds = () => {
5    const [total, setTotal] = useState(0);
6  }
```



Affichage du state

Affiche le state dans le composant :

Annonces récentes

Prix total: 0 €



Table

120 €

Add price to total



Dame-jeanne

75 €

Add price to total



Bougie

4 €

Add price to total

```
return (  
  <>  
    <h2>Annonces récentes</h2>  
    <p>Prix total: {total} €</p>  
    <section className="recent-ads">  
      {ads.map((ad) => (  
        <div key={ad.title}>  
          <AdCard
```




Modification du state

Modifie le bouton pour qu'il ajoute le prix de l'annonce au total:

```
{ads.map((ad) => (  
  <div key={ad.title}>  
    <AdCard  
      imgUrl={ad.imgUrl}  
      link={ad.link}  
      price={ad.price}  
      title={ad.title}  
    />  
    <button  
      className="button"  
      onClick={() => {  
        setTotal(total + ad.price);  
      }}  
    >  
      Add price to total  
    </button>  
  </div>  
))}
```



A toi de jouer !



- Crée le state permettant d'avoir un compteur de prix mis à jour au clic



useEffect

Intégrer des effets de bord à notre application



Effet de bord ?

En informatique, une fonction est dite à effet de bord (traduction mot à mot de l'anglais side effect) si elle modifie un état en dehors de son environnement local, c'est-à-dire “A une interaction observable avec le monde extérieur”



Fetching data with useEffect

<https://react.dev/reference/react/useEffect#fetching-data-with-effects>



Pourquoi a t'on besoin de **useEffect** ?

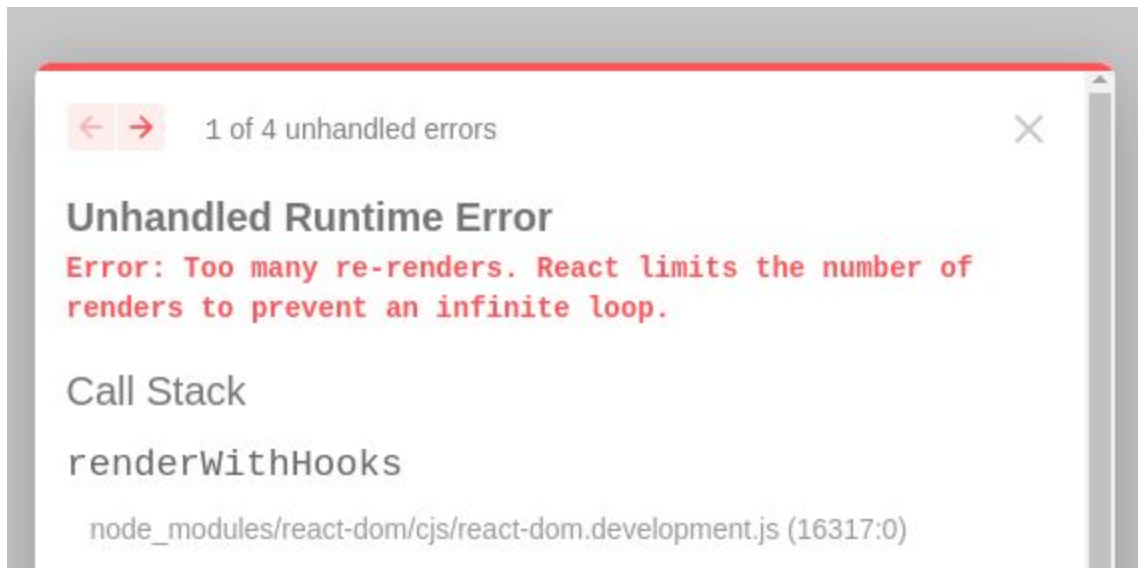
Pour mettre à jour l'interface, on utilise **useState**.

useState provoque un nouveau rendu et donc une nouvelle exécution des fonctions de notre composant.

Que se passerait t'il si on modifiait le state à l'exécution du composant sans passer par un **useEffect** ?



Pourquoi a t'on besoin de useEffect ?





Vérifions

TS RecentAds.tsx U X

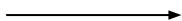


src > components > TS RecentAds.tsx > [🔍] RecentAds > [🔍] ads

```
1  import { useState, useEffect } from "react";
2  import AdCard, { AdCardProps } from "../AdCard";
3
4  const RecentAds = () => {
5    const [total, setTotal] = useState(0);
6    const everyRender = () => {
7      console.log("This will be executed after every render");
8    };
9    everyRender();
10
11    useEffect(() => {
12      const firstRenderOnly = () => {
13        setTotal(1);
14        console.log(
15          "This will be executed after the first render only. Even if I change the state"
16        );
17      };
18      firstRenderOnly();
19    }, []);
```




A retenir



- **useEffect** permet de synchroniser un composant react avec le monde extérieur
- Le premier paramètre est la fonction à exécuter
- Le second paramètre est un tableau de dépendances qui liste les valeurs à surveiller et qui déclencheront une nouvelle exécution de la fonction
- Pour aller plus loin:
<https://2019.wattenberger.com/blog/react-hooks>