React.js with PHP API REST

4. Composant ArticleScreen, data mapping and propTypes

Voici une proposition de correction pour le composant ArticleScreen.jsx
Nous reprenons les states et le useEffect déjà développés dans HomeScreen pour récupérer l'article.

src > screens > article > ArticleScreen.jsx > ArticleScreen

```jsx
import { useState, useEffect } from "react";

function ArticleScreen() {
    const [data, setData] = useState({});
    const [loading, setLoading] = useState(true);

    useEffect(() => {
        const fetchData = async () => {
            try {
                const url = "http://api.php-blog-project.loc/article/12";
                const response = await fetch(url);
                if (!response.ok) {
                    throw new Error("Erreur de réseau");
                }
                const result = await response.json();
                console.log(result);
                setData(result || {});
            } catch (error) {
                console.log(error);
            } finally {
                setLoading(false);
            }
        };
        fetchData();
    }, []);

    return (
```

Puis nous adaptons la vue php pour écrire le render du composant.

-> voir page suivante

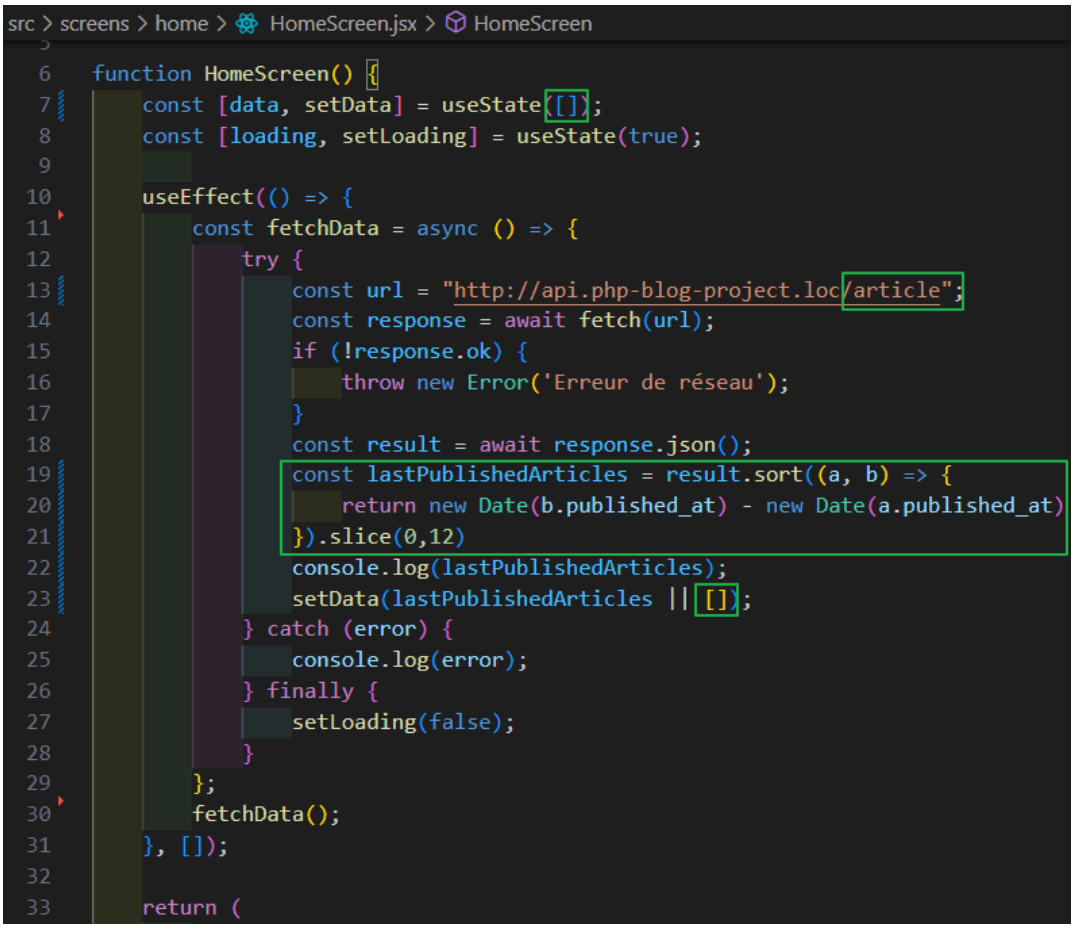
src > screens > article > ArticleScreen.jsx > ArticleScreen

```jsx
function ArticleScreen() {

    return (
        <main className="mt-5 pt-3 row">
            {loading && (<p className="col-12 text-center">
                            Chargement des données ...
                        </p>)}
            {("id_article" in data) ? (
                <div className="col-12">
                    <div className="text-center mb-3">
                        <h1>{data.title}</h1>
                        <img src={data.img_src} alt={data.title} />
                    </div>
                    <p style={{ textAlign: "justify" }}>{data.summary}</p>
                    <p>
                        Publié le{" "}
                        {new Date(data.published_at).toLocaleDateString("fr-FR" )}
                        {data.updated_at &&
                        " (Mis à jour le" +
                            new Date(data.updated_at).toLocaleDateString("fr-FR") +
                        ")"}
                    </p>
                </div>
            ) : (
                !loading && (<p className="col-12 text-center">
                                Aucun article trouvé ...
                            </p>)
            )}
        </main>
    );
}

export default ArticleScreen;
```

Résultat

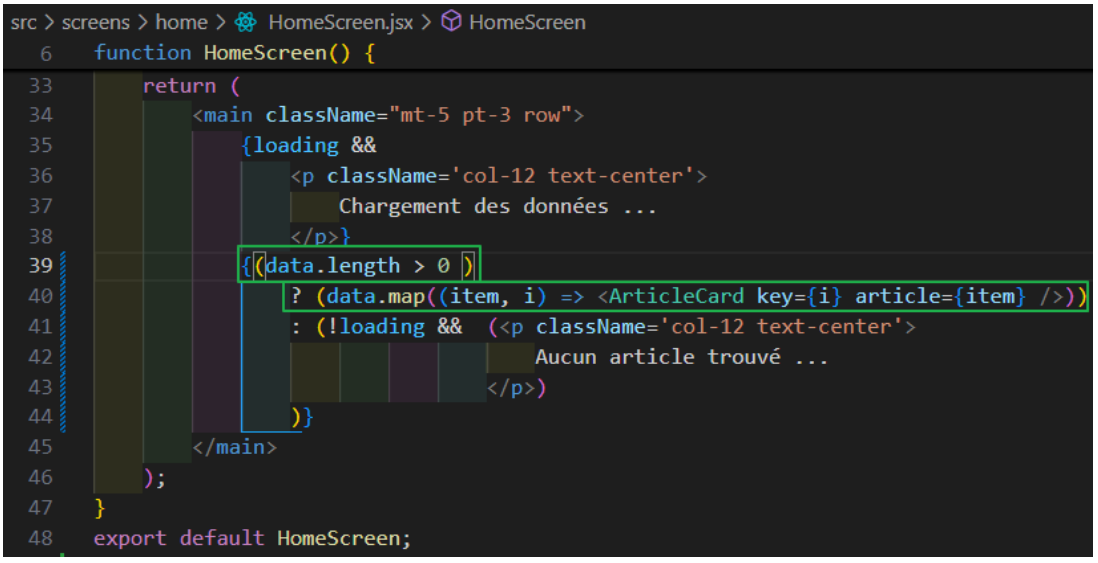
Le data mapping

Nous allons maintenant modifier le HomeScreen pour qu'il affiche la liste des 12 derniers articles publiés (comme c'est le cas sur l'appli PHP MVC)



Cette fois ci la requête vers l'api doit nous retourner un tableau, nous modifions donc l'initialisation du state (ligne 7) ainsi que la valeur alternative lors de l'utilisation du setter (ligne 23)
La requête ne comporte plus d'id pour récupérer tous les articles (ligne 13)
Les lignes 19 à 21 permettent de trier les articles par date de publication de la plus récente à la plus ancienne et de ne garder que les 12 premiers.

Pour le render :

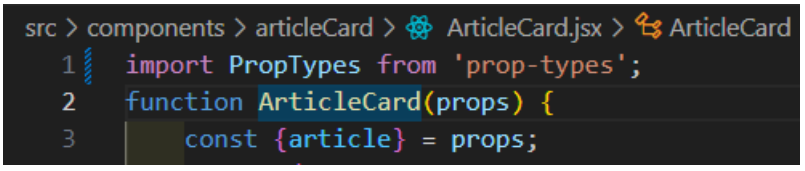


Nous modifions la condition d'affichage des données, le tableau ne doit pas être vide (ligne 39) il nous reste à parcourir le tableau à l'aide de la fonction map et insérer le composant ArticleCard pour chaque article du tableau data.

Ressources : https://react.dev/learn/rendering-lists

Les propTypes

Les propTypes, comme son nom l'indique, servent à vérifier (valider) les types des props d'un composant
Il faut commencer par importer le package



Puis après la déclaration du composant et avant l'export, définir les types des différentes props

Pour les objets comme article, il y a la possibilité de détailler le type de chaque propriété

Ressources :
https://hygraph.com/blog/react-proptypes
https://www.freecodecamp.org/news/how-to-use-proptypes-in-react/

A vous de coder les pages **SeriesScreen** et **TechsScreen** et les composants **SerieCard** et **TechCard** en vous basant sur ce que nous avons fait pour HomeScreen et ArticleCard

git : https://github.com/DWWM-23526/REACT_BLOG_PROJECT/tree/Step04