React router et selector pour les donne sans rafraichissement des pages

selector

```
// Utiliser cette fonction pour donner les bons posts à afficher
// au composant <Posts />

const filterPosts = (posts, category) => {
  // Si la catégorie que je reçoi est "accueil"
  // JE veux renvoyer tous les posts
  if (category === 'Accueil') {
    return posts;
  }
  // Sinon, je veux renvoyer uniquement les posts
  // qui font parti de la catégorie
  return posts.filter((post) => post.category === category);
};
```

création d'un hook pour les requête ajax (axios)

```
import { useEffect, useState } from 'react';
import axios from 'axios';

// Je veux obtenir toute la logique nécessaire à la requête (de posts,
de catégories)
//  dans un seul hook.

export default (url) => {
  // url -> l'adresse qui me renverra ce que je veux
  // Il me faut un state pour contenir mes résultats
  const [items, setItems] = useState([]);
  const [loading, setLoading] = useState(true);
  useEffect(() => {
    // Je veux lancer ma requête avec axios
    axios.get(url)
      .then((response) => {
        setItems([...response.data]);
      })
      .catch((error) => {
        console.log(error);
      })
      .finally(() => setLoading(false));
  }, []);

  return [loading, items];
};
```

le système de route

```
const App = () => {
  const [articlesLoading, articles] =
useAjax('https://oclock-open-apis.now.sh/api/blog/posts');
  const [categoriesLoading, categories] =
useAjax('https://oclock-open-apis.now.sh/api/blog/categories');
  return (
    <div className="app">
      <Header categories={categories} />
      {
        (articlesLoading || categoriesLoading) && <Spinner />
      }

      {
        (!articlesLoading && !categoriesLoading) && (
          <Switch>
            {
              categories.map((category) => (

                <Route
                  exact
                  key={category.label}
                  path={category.route} // Réagi à "/"
                  /*
                  component={NomDuComposant} -> Pratique mais ne permet
pas de passer des props
                  */
                  render={() => (
                    <Posts
                      posts={filterPosts(articles, category.label)}
                      category={category.label}
                    />
                  )}
                /*
                  Render prend une fonction qui doit return ce qu'on
veut afficher, on peut
                  donc passer des props
                */
                />
              ))
            }
```

Route pour afficher les article

```
      <Route
              path="/articles/:slug"
              render={(object) => {
                // Je récupère mon paramètre dans objet.match.params
                const { slug } = object.match.params;
                // Etape 1, une variable pour contenir le bon article
                // Etpae 2, destrcturer la variable sur SinglePost pour
qu'il reçoive les bons props
                // Etape 3, Si jamais pas d'article trouvé -> 404
                const singleArticle = articles.find((article) =>
article.slug === slug);

                if (!singleArticle) {
                  return <PAGE404 />;
                }
                return <SinglePost {...singleArticle} />;
              }}
            />
            <Redirect from="/jquery" to="/react" />
            <Route component={PAGE404} />
          </Switch>
        )
      }
      <Footer />
    </div>
  );
};
```

exemple système de lien

```
const Post = ({
  title, category, excerpt, slug,
}) => (
    <article className="post">
      <Link to={`/articles/${slug}`}>
        <h2 className="post__title">{title}</h2>
      </Link>
      <span className="post__category">{category}</span>
      <p className="post__excerpt" dangerouslySetInnerHTML={{ __html:
DOMpurify.sanitize(excerpt) }} />
    </article>
  );
```