

React - TD6 - communication client serveur

Exercice 1 – Interrogation serveur avec React - suite du jeu de cartes

Pour rappel, dans le TD 7, nous avons créé une interface qui permet de saisir des cartes et d'afficher la liste des cartes.

Q 1.1 On considère que la liste des cartes doit être récupérée sur un serveur REST qui est interrogé au chargement de la page et qui retourne les données au format JSON. Compléter le code ci-dessous. L'adresse du service dédié est la suivante 'localhost :8080/cardsList'.

Version fonctionnelle :

```

1  import React, { useState } from "react";
2  import axios from "axios";
3
4  const CardList = () => {
5    const [cards, setCards] = useState([]);
6
7    return (
8      <div className="Cardlist">
9        {cartes.map((card, index) => (
10          <Card symbol={card.card} affichage={card.feedback} />
11        ))}
12      </div>
13    );
14  };

```

Version classe :

```

1  import React from "react";
2  import axios from "axios";
3
4  class Cardlist extends Component {
5    constructor(props) {
6      super(props);
7      this.state = { cartes: [] }; // optionnel
8    }
9
10   render() {
11     return (
12       <div className="Cardlist">
13         {this.state.cartes.map((card, index) => (
14           <Card symbol={card.card} affichage={card.feedback} />
15         ))}
16       </div>
17     );
18   }
19 }

```

Q 1.2 Modifier le code précédent pour ne récupérer que les 4 premières cartes et ajouter aussi un attribut couleur: 'noir' à chaque carte.

Q 1.3 Écrire la requête axios qui permet de créer une carte grâce à la route d'API 'localhost/cardCreate'. Cette requête est stockée dans la méthode `postCardHandler`.

On se basera sur le formulaire de création de carte suivant :

```

1  return (
2    <div className="FormulaireCard">
3      <label>Symbol <input type="text" value={symbol} onChange={  } /></label>
4      <label>Affichage <input type="text" value={affichage} onChange={  } /></label>
5      <button onClick={postCardHandler}>Add Card</button>
6    </div>
7  )

```

Exercice 2 – Réfléchissons au projet

Q 2.1 On considère le formulaire de connexion. Écrire le code vous permettant de connecter l'utilisateur lorsqu'il clique sur le bouton. On souhaite stocker le statut de l'exécution dans l'état du composant :

- en cas d'erreur : `{status:"error",texterror:response.data["description"]}`
- sinon : `{status:""}`.

Q 2.2 Même question pour le bouton de déconnexion.