
Corrigé Atelier N°2 States avec les classes

Exercice 1 :

Dans cet exercice, nous allons définir l'état du composant Client et Fournisseur. Chaque client est caractérisé par un nom et un numéro de téléphone alors qu'un fournisseur est caractérisé par l'article qu'il vend et son prix. Les classes client et fournisseur sont élaborés dans des fichiers externes. Puis on fera leurs imports dans App.js.

Solution :

Classe Fournisseur :

```
import { Component } from 'react';
class Fournisseur extends Component{
  constructor(){
    super();
    this.state={
      "article":"41",
      "prix":"2200"
    }
  }
  render(){
    return(<h1>le fournisseur fournie l'article {this.state.article} avec le prix {this.state.prix}</h1>)
  }
}
export default Fournisseur;
```

Classe Client :

```
import { Component } from 'react';
class Client extends Component {
  constructor(){
    super();
    this.state={
      "nom":"Mohamed",
      "telephone" : "89854587"
    }
  }
  render() {
    return (<h1>le client {this.state.nom} porte le numéro {this.state.telephone} </h1>)
  }
}
export default Client;
```

Fichier App.js :

```
import { Component } from 'react';
import Client from "../Client";
import Fournisseur from "../Fournisseur";
import './App.css';
function App() {
  return (
    <div className="App">
      <Client />
      <Fournisseur />
    </div>
  );
}
export default App;
```

Résultat obtenu :

Données définies dans la classe Client → **le client Mohamed porte le numéro 89854587**

le fournisseur fournit l'article 41 avec le prix 2200 ← Données définies dans la classe Fournisseur

Exercice 2 :

Considérons la classe personnes ayant le state l'objet data caractérisé par un id, un nom et un âge.

1. Créer la classe Personnes qui permet d'initialiser l'état de l'objet data puis afficher les informations relatives à cet objet, en utilisant la méthode map. Les données du tableau d'objets « data » sont les suivantes :

```
data:[
  {
    "id":"100",
    "nom":"Mohamed Ali",
    "age":"33",
  },
  {
    "id":"101",
    "nom":"Abbes",
    "age":"42"
  },
]
```

```

    {
      "id": "102",
      "nom": "Souhaila",
      "age": "29"
    },
    {
      "id": "103",
      "nom": "Asma",
      "age": "18"
    }
  ]

```

2. Faire appel à la classe Personnes dans App.js.

Solution :

Classe Personnes :

```

import React, { Component } from "react";
class Personnes extends Component{
  constructor(){
    super();
    this.state=
    {
      data:[
        {
          "id": "100",
          "nom": "Mohamed Ali",
          "age": "33",
        },
        {
          "id": "101",
          "nom": "Abbes",
          "age": "42"
        },
        {
          "id": "102",
          "nom": "Souhaila",
          "age": "29"
        },
        {
          "id": "103",
          "nom": "Asma",
          "age": "18"
        }
      ]
    }
  }
}

```

```

    ]
  }
}
render()
{
  return(
    <div>
      {this.state.data.map((person ,i)=>
        <h1>id:{person.id} Nom:{person.nom} Age:{person.age}</h1>)}
    </div>
  )
}
}
export default Personnes;

```

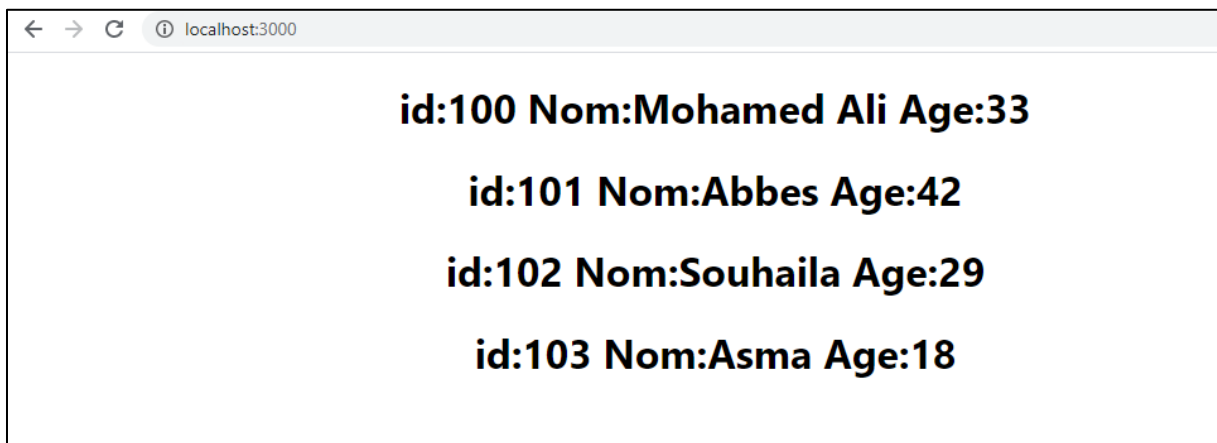
Fichier App.js :

```

import Personnes from "./Personnes"
import './App.css';
function App() {
  return (
    <div className="App">
      <Personnes />
    </div>
  );
}
export default App;

```

Résultat obtenu :



Changement de State

Exemple 1

```
import React, { Component } from 'react'

export default class Personne extends Component {
  constructor(){
    super()
    this.state = {
      name: "Samia",
    };
  }
  render() {

    return (
      <div>
        <h1>{this.state.name}</h1>
        <button onClick={() => this.setState({ name: "Ahmed" })}>
          Change Name
        </button>

      </div>
    )
  }
}
```

State tableau:

```
import React, { Component } from 'react'

export default class Departement extends Component {
  constructor(){
    super()
    this.state={
      name:"ISET",

      contact: { nom: "mohamed", email: "iset@gmail.com", logo: "images/iset.png"
    },
    departs: [
      { id: "1", nom: "informatique" },
      { id: "2", nom: "Gestion" },
      { id: "3", nom: "Mecanique" },
      { id: "4", nom: "Civil" },
      { id: "5", nom: "electrique" }
    ]
  }
}
```

```

    }

    }
    render() {
      return (
        <div>
          <h1>{this.state.name}</h1>
          <button onClick={() => this.setState({ name: "ISET Sfax" })}> Change Name
        </button>
          <h2> Départements </h2>
          <table border="1">
            {this.state.departs.map((value,index)=>{
              return(<tr><td key = {index}>{value.id}</td><td> {value.nom}</td>
            </tr> )
            }) }

          </table>
        </div>
      )
    }
  }
}

```

Changer le state d'un contact

```

changerContact=()=>{

  /* première méthode avec prevState
  this.setState(prevState => ({
    contact: {
      ...prevState.contact,
      nom: "Hassan",
      email: "hsan.hachicha@gmail.com"
    }
  }));

  deuxième méthode avec this.state*/
  this.setState(() => ({
    contact: {
      ...this.state.contact,
      nom: "Hassan",
      email: "hsan.hachicha@gmail.com"
    }
  }));
}

```

```

import React, { Component } from 'react'

export default class Departement extends Component {
  constructor(){
    super()
    this.state={
      name:"ISET",

      contact: { nom: "mohamed", email: "iset@gmail.com", logo: "images/iset.png"
    },

    departs: [
      { id: "1", nom: "informatique" },
      { id: "2", nom: "Gestion" },
      { id: "3", nom: "Mecanique" },
      { id: "4", nom: "Civil" },
      { id: "5", nom: "electrique" }
    ]
  }
}

AjoutDep={()=>{
  this.setState({ departs: [{ id: "6", nom: "procédés" } ]})
}

render() {
  return (
    <div>
      <h1>{this.state.name}</h1>
      <button onClick={() => this.setState({ name: "Infosoft" })}> Change Name
    </button>
      <h2> Départements </h2>
      <table border="1">
        {this.state.departs.map((value,index)=>{
          return(<tr><td key = {index}>{value.id}</td><td> {value.nom}</td>
        </tr> )
        }) }

      </table>

      <button onClick={() => this.AjoutDep()}> Ajouter depart</button>
    </div>
  )
}
}

```

Exécution:

Avant appui sur le bouton Ajouter depart

Départements

1	informatique
2	Gestion
3	Mecanique
4	Civil
5	electrique

Changer depart

Après appui sur le bouton Changer depart

Départements

6	procédés
---	----------

Changer depart

Constatations :

On remarque que les valeurs du state initiale (`departs[]`) seront tous remplacés par les nouvelles valeurs du state (6, procédés)

La solution étant de conserver le state précédent et de lui ajouter les nouvelles valeurs envoyés

Solution 1

```
AjoutDep=()=>{  
    this.setState({ departs: [...this.state.departs,{ id: "6", nom:  
"procédés",directeur: "slim ben hlima" } ]})  
}
```

Départements

1	informatique
2	Gestion
3	Mecanique
4	Civil
5	electrique
6	procédés

Changer depart

Solution 2

```
changeDep={()=>{  
  this.setState(prevState => ({  
    departs : [...prevState.departs,  
      {id: "6", nom: "procédés"} ]  
  }))  
})
```