

React

La mise à jour de l'état du state ou des props peut être asynchrone. Par exemple si on a plusieurs appels de la méthode `setState`, React peut regrouper ces `setState` et faire une seule mise à jour.

Une conséquence de cette mise à jour est que la valeur du state peut ne pas correspondre à ce que vous souhaitez faire.

Créez une classe `App` avec le code suivant, nous utiliserons la méthode `componentDidMount` pour mettre à jour la valeur du state "taxe" :

```
// Dans le composant App

componentDidMount(){
  // 30 % d'augmentation
  this.setState({percentage : 0.2});
  this.setState({percentage : this.state.percentage + 0.05 });
  this.setState({percentage : this.state.percentage + 0.05 });
}
```

Passez maintenant la valeur `taxe` en props à un composant `Price` comme suit :

```
// Dans la classe App
render(){
  return (
    <React.Fragment>
      <p>Tax : {this.state.percentage} </p>
      { /* Augmentation du prix */ }
      <Price percentage={ 1 + this.state.percentage } />
    </React.Fragment>
  );
}
```

Dans le composant `Price` utilisez également la méthode `componentDidMount` pour mettre à jour le state `total`; ce dernier dépend des props du composant `App` :

```
componentDidMount(){

  this.setState({ total : 100 * this.props.percentage });
  this.setState({ total : (this.state.total + 200) * this.props.percentage });
  this.setState({ total : (this.state.total + 50) * this.props.percentage });

}
```

Affichez maintenant le résultat, pour information vous devriez trouver la valeur

suivante, le 1.3 correspond à l'augmentation de 30 % du prix :

```
( (100 * 1.3 + 200) * 1.3 + 50 ) * 1.3
```

622.7

Vous constaterez qu'en réalité vous n'obtenez pas cette valeur, React fait une mise à jour des props & state asynchrone pour des raisons d'optimisation. Pour pouvoir corriger cet effet de bord, vous devez passer à la fonction `setState` une fonction anonyme pour forcer la mise à jour synchrone du state.

```
// Dans le composant Price
  this.setState((state, props) => ({total : 100 * props.percentage }) );
// Dans le composant App
  this.setState((state, props) => ({percentage : 0.2 }) );
```

Ce qu'il faudra retenir c'est que si vous avez plusieurs `setState` à faire l'un à la suite des autres il faudra sans doute utiliser la méthode de la fonction fléchée que l'on vient de voir.