

Événements de cycle de vie de React

Découvrez les événements React Lifecycle et comment vous pouvez les utiliser

Les composants de la classe React peuvent avoir des Hooks pour plusieurs événements du cycle de vie.

Les Hooks permettent également aux composants de fonction d'y accéder, d'une manière différente.

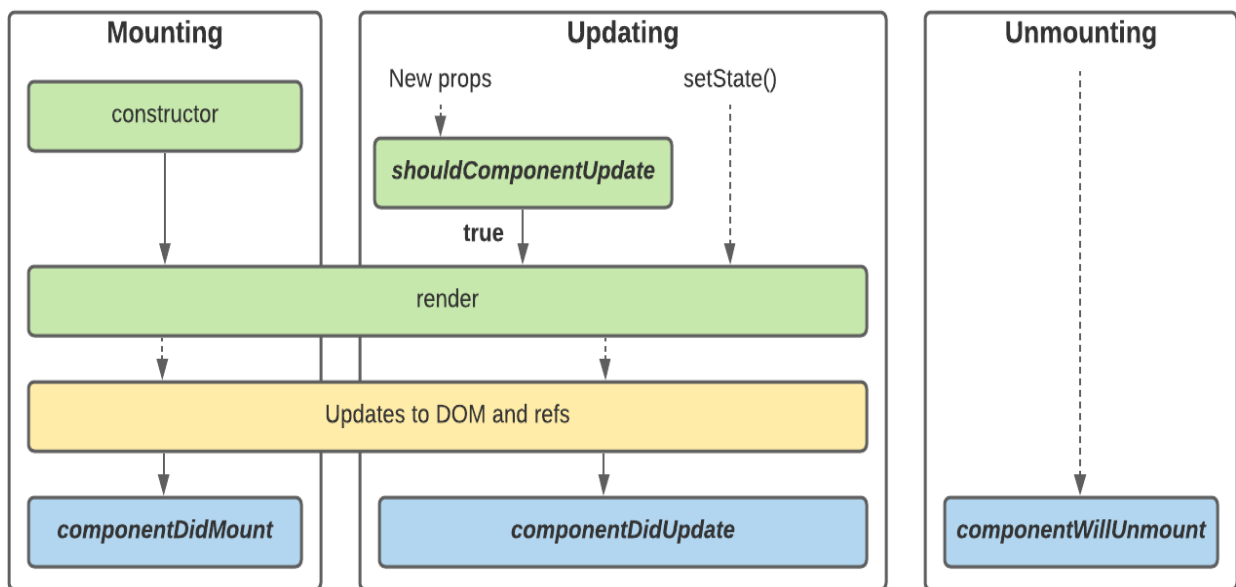
Pendant la durée de vie d'un composant, une série d'événements est appelée, et à chaque événement, vous pouvez accrocher et fournir des fonctionnalités personnalisées.

Quel crochet est le mieux pour quelle fonctionnalité est quelque chose que nous allons voir ici.

Tout d'abord, il y a 3 phases dans le cycle de vie d'un composant React:

- Montage
- Mise à jour
- Démontage

Voyons ces 3 phases en détail et les méthodes qui sont appelées pour chacune.



I. Montage

Lors du montage, vous disposez de 3 méthodes de cycle de vie avant le montage du composant dans le DOM : le **constructor**, **render** et **componentDidMount**.

Constructeur

Le constructeur est la première méthode appelée lors du montage d'un composant.

Vous utilisez généralement le constructeur pour configurer l'état initial en utilisant **this.state =**

render()

À partir de la méthode `render ()`, vous retournez le JSX qui construit l'interface du composant.

C'est une méthode pure, elle ne devrait donc pas causer d'effets secondaires et devrait renvoyer la même sortie lorsqu'elle est appelée plusieurs fois avec la même entrée.

componentDidMount ()

Cette méthode est celle que vous utiliserez pour effectuer des appels d'API ou traiter des opérations sur le DOM.

II. Mise à jour

Lors de la mise à jour, vous disposez de 5 méthodes de cycle de vie avant que le composant ne soit monté dans le DOM : le **shouldComponentUpdate**, **render** et **componentDidUpdate**.

shouldComponentUpdate (nextProps,nextState)

Cette méthode renvoie un booléen, **true** ou **false**. Vous utilisez cette méthode pour dire à React s'il doit continuer avec le rendu, et la valeur par défaut est **true**. Vous reviendrez **false** lorsque le rendu coûte cher et que vous voulez avoir plus de contrôle sur le moment où cela se produit.

render()

Voir la description ci-dessus pour cette méthode.

componentDidUpdate (prevProp,PrevState)

Cette méthode est appelée lorsque le composant a été mis à jour dans le DOM. Utilisez cette option pour exécuter toute API DOM tierce ou appeler des API qui doivent être mises à jour lorsque le DOM change.

III. Démontage

Dans cette phase, nous n'avons qu'une seule méthode, **componentWillUnmount**.

composantWillUnmount ()

La méthode est appelée lorsque le composant est supprimé du DOM. Utilisez ceci pour effectuer tout type de nettoyage que vous devez effectuer.