React : Introduction

React que l’on appelle aussi React.js ou ReactJS) ;

C’est une bibliothèque JS développée par FB depuis 2013 ;

Objectif : faciliter la création d’application web monopage (c’est-à-dire, une application web accessible via une page web uniquement, qui permet d’éviter le chargement d’une nouvelle page à chaque action demandée), via la création de composants dépendant d’un état générant une page HTML à chaque changement d’état.

React c’est une bibliothèque qui ne gère que l’interface de l’application, elle peut aussi être utilisée avec une autre bibliothèque ou un Framework MVC (traduction) comme Angular.JS.

La bibliothèque se démarque de ses concurrents par sa flexibilité, ses performances, en travaillant un dom virtuel et en ne mettant à jour le rendu dans le navigateur qu’en cas de nécessité.

React est utilisée par Netflix, yahoo, airbnb, sony, facebook, instagram, whatsApp.

En 2015, wordpress est développée en JS avec node.js et react

Histoire

React est créé par Jordan Walke, un ingénieur au sein de la société fb à la fin de l’année 2011. Peter Hunt ingénieur travaillant pour instagram est intéressée par la bibliothèque. React est publié sous licence Apache 2.0, le 29 mai 2013.

Fonctionnalités :

React a été conçu comme étant une bibliothèque et non un framework MVC

MVC est un motif d’architecture logicielle destiné aux interfaces graphiques. Il est composées de trois types de modules :

Un modèle contient les données à afficher (Model)

Une vue (view) contient la présentation de l’interface graphique.

Un controleur (controller) contient la logique concernant les actions effectuées par l’utilisateur.

React encourage la création de composants réutilisables, avec en entrée des données, pouvant changer au cours du temps.

Par ailleurs, react n’utilise pas de système de templates et ne fonctionne qu’avec du JS, permettant une insertion complète du composant au sein d’une unique classe. Pour faciliter l’écriture de ala vue initiale chez FB a développé un langage JSX, qui permet de générer des objects JS avec une notation similaire à HTML

DOM virtuel est une représentation du DOM en JS, au lieu de générer le DOM lui-même comme avec un langage de templating, cad au lieu de dialoguer avec les api du navigateur pour construire le dom, on ne génère qu’un arborescence d’objects JS en mémoire.

Pour plus de précision le DOM est une interface de programmation, qui permet à des scripts d’examiner et de modifier le contenu du navigateur web. par le dom, la composition d’un docuement HTML ou XML est représentée sous forme d’un jeu d’objets, lesquels peuvent représenter une feneêtre, une phrase, ou un style, par exemple. Relié selon une structure en arbre

Quand au dom virtuel :

Le DOM virtuel (VDOM) est un concept de programmation dans lequel une représentation idéale, ou « virtuelle », d’une interface utilisateur (UI) est conservée en mémoire et synchronisée avec le DOM « réel » par une bibliothèque telle que ReactDOM. Ce processus s’appelle [réconciliation](https://fr.reactjs.org/docs/reconciliation.html).

Cette approche rend possible l’API déclarative de React : vous indiquez à React dans quel état vous souhaitez que l’UI se trouve, et il s’assure que le DOM correspond à cet état. Ça permet de faire abstraction de la manipulation des attributs, de la gestion des événements et de la mise à jour manuelle du DOM que vous auriez normalement dû faire vous-mêmes pour créer votre application.

Puisque le « DOM virtuel » est plus un modèle qu’une technologie spécifique, on l’emploie parfois pour désigner différentes choses. Dans le monde React, le terme « DOM virtuel » est généralement associé aux [éléments React](https://fr.reactjs.org/docs/rendering-elements.html), car il s’agit des objets représentant l’interface utilisateur. Cependant, React utilise également des objets internes appelés « fibres » (fibers, NdT) pour conserver des informations supplémentaires sur l’arbre des composants. Ils peuvent également être considérés comme faisant partie de l’implémentation du « DOM virtuel » dans React.

En 2015, react native fait son apparition, ce framework est basé sur react et permet de créer toujours en JS des applications multi-plateformes android et ios

Déclaratif

Grâce à React, il est facile de créer des interfaces utilisateurs interactives. Définissez des vues simples pour chaque état de votre application, et lorsque vos données changeront, React mettra à jour, de façon optimale, juste les composants qui en auront besoin.

Des vues déclaratives rendent votre code plus prévisible et plus facile à déboguer.

À base de composants

Créez des composants autonomes qui maintiennent leur propre état, puis assemblez-les pour créer des interfaces utilisateurs complexes.

Dans la mesure où les composants sont écrits en JavaScript plutôt que sous la forme de gabarits, vous pouvez facilement utiliser des données complexes dans vos applications et garder l’état hors du DOM.

Utilisable partout

Comme nous ne présumons rien sur les autres technologies que vous utilisez, vous pouvez développer de nouvelles fonctionnalités avec React sans avoir à réécrire votre code existant.

React peut aussi être utilisé côté serveur avec Node, ou pour créer des applications mobiles grâce à [React Native](https://react-native.dev/" \t "_blank).

const element = <h1>Bonjour, monde !</h1>;

Cette drôle de syntaxe n’est ni une chaîne de caractères ni du HTML.

Ça s’appelle du JSX, et c’est une extension syntaxique de JavaScript. Nous recommandons de l’utiliser avec React afin de décrire à quoi devrait ressembler l’interface utilisateur (UI). JSX vous fait sûrement penser à un langage de balisage, mais il recèle toute la puissance de JavaScript.

useState()

Props

Components

Hook

useEffect etc