

Formation React

3 jours (21 heures)

Présentation

React est une bibliothèque open-source JavaScript développée par Facebook depuis 2013 en partenariat avec Instagram et soutenue par une grande communauté de développeurs. Le but principal de cette librairie est de faciliter la création d'application Single-Page via des composants dépendant d'un état et générant du HTML à chacun de ses changements. Grâce à ses performances élevées, elle a tout de suite conquis les entreprises, dont Airbnb, Sony, Netflix !

Avec cette formation vous allez découvrir comment exploiter pleinement cette technologie web avancée, sur l'une des librairies les plus utilisées notamment dans les nouveaux projets de Web Apps.

Comme tous nos programmes, notre formation porte sur la toute dernière version en date de cette librairie ([React 16.9.0](#) version sortie en août 2019 à la date de l'article).

Objectifs

- Développer et Maîtriser les fonctionnalités dans la dernière version 16 de React la technologie JavaScript de Facebook
- Créer des composants web complexes rapidement
- Application SinglePage avec React.JS, Flux et isomorphisme
- Mettre en place une application web performante
- Appréhender React Native

Public visé

Développeurs web

Pré-requis

Connaissances en JavaScript

Pour aller plus loin

Afin de compléter cette formation, nous vous proposons 3 formations exclusives :

- sur les Test Unitaires et l'optimisation avec [React Avancé](#)
- sur la [Programmation reactive fonctionnelle avec Redux et RxJS](#)
- sur la version [full mobile avec React Native](#)

Programme de notre formation React v16

Introduction

- Les fondamentaux. HTML5, CSS3, JavaScript, DOM et ShadowDom
- Design patterns applicatifs classiques. Limitations des applications JavaScript
- Ecosystème des frameworks web JavaScript
- Principes de Data-Binding : dirty-checking, observable, virtual-dom, immutabilité
- La nouvelle vision de ReactJS
- Transpilés, JSX
- Environnement de développement: IDE, outils et plug-ins

Présentation de ReactJS

- Le "V" de MVC ? MVC et Virtual Dom, un choix de performance
- Composants à état, fonction de rendu simple, performance
- JavaScript ou JSX
- Comprendre JSX
- Méthodes principales de l'API
- Création de composants de vues. Cycle de vie
- Initialisation de propriétés

ES6, ES2015 et TypeScript

- Compilateurs : babel, traceur et typescript
- Nouveautés ES6 : portée, bloc, template literals, arrow functions...
- TypeScript: Typage fort et Flow
- CommonJS & browserify/webpack
- ES6 & systemjs

Composants

- Événements, autobinding et délégation
- Design Pattern : les composants à état
- Composer par ensembles

- Classes, constructeur et héritage
- Render, Binding
- Props VS state
- Component Data Flow : propriétaire, enfants et création dynamique
- Composants réutilisables : contrôle et transfert de propriétés
- Contrôle des composants de formulaire
- Manipulation du DOM

Single Page Application

- Présentation des Flux
- Propagation de données
- Création de vues et contrôleurs
- Rôle du Dispatcher pour les actions
- Stores, gestionnaire d'états logique
- Extension pour ReactJS : animation
- Data-binding bidirectionnel

MobX vs Redux vs Flux

MobX

- Les actions et les propriétés calculées
- Utilisation des providers et de l'injection des composants pour organiser un Store MobX
- Exemple de TODO list avec Mobx

Isomorphisme

- Principe et bénéfices du développement isomorphe: Application universelle
- Internationalisation
- Gestion des données serveur: Relay
- Écosystème du JavaScript côté serveur
- Initiation à Node.js
- Configuration de Node.js pour ReactJS
- Configuration et organisation des routes

Performance et Architecture

- Introduction et utilisation de Flux
- Introduction et utilisation de Redux
- Lifecycle hooks
- Immutabilité
- React Router
- Outils de développement & tests
- Server-side rendering

Module complémentaire React Avance & Redux - 2 jours

Design patterns avancés

- HOC : Higher Order Components
- Portals : Eléments DOM distants
- IOC : Injection de dépendances avec les Context
- Fragments & render props

Redux Avancé : Advanced

- Immutabilité avancée
 - Immutable.js
- Avantages & inconvénients
- Normalisation de la forme d'état
 - Création d'un état par défaut normalisé
- Création Middleware Redux
 - Redux Middleware
 - React-redux
 - Redux Thunk
 - Redux Saga
 - Redux DevTools

Internationalisation & optimisation

- Internationalisation VS localisation
 - Bonnes pratiques
 - Différences et scénarios d'utilisation
 - Présentation des principales bibliothèques d'internationalisation
- Optimisation
 - Cycles de vie
 - Tuning

React Native (En option)

- Introduction à ReactNative
- Développement natif mobile et tablette pour iOS, Android et Windows
- Native Components
- Asynchronous Execution
- Touch Handling

- Flexbox and Styling
- Polyfills
- Extensibility
- Views
- OpenGL Application

Sociétés concernées

Cette formation s'adresse aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiple permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.