
Programme de Formation

ReactJS : Créer des interfaces web modernes et dynamiques

Description

Approfondissement de ReactJS pour maîtriser la création de SPA, les composants, la gestion d'état, JSX, les interactions API et l'optimisation des performances.

Points Forts

Apprenez à créer des applications web riches et performantes avec ReactJS : composants, hooks, gestion d'état (Flux/Redux), JSX et optimisation des performances, à travers des ateliers pratiques inspirés de cas concrets.



Public visé

- Développeurs web front-end
- Développeurs full-stack souhaitant maîtriser React
- Architectes applicatifs
- Chefs de projet techniques



Objectifs pédagogiques

- Comprendre les concepts clés de ReactJS et ses spécificités
- Créer des composants fonctionnels
- Maîtriser la syntaxe JSX
- Utiliser Flux et gérer les bugs
- Tester son développement

Organisation

Durée : 21 heures

Mode d'organisation : Mixte

Contenu pédagogique



Description

Jour 1 :

Comprendre les concepts clés de ReactJS et ses spécificités

- Les Single Page Applications
- Les limites de Javascript et de jQuery
- Lenteurs du DOM
- Positionnement de React JS
- Création une SPA « from scratch en JS »
- Exemples de travaux pratiques : le formateur réalise une maquette simple d'une page web. Il réalise un diagramme de classe UML pour représenter la modélisation et le découpage de la page sous forme de composants web. A partir du diagramme de classe, il implémente une partie du code pour montrer le fonctionnement d'une SPA en "from scratch" (en partant de zéro, sans Framework) et mettre en application les syntaxes ES6. Les participants reproduisent et exécutent le code sur leur poste de travail.



Maîtriser la syntaxe JSX

- Ecrire un premier composant
- Les props, et les states
- La fonction render et le langage JSX
- Les bonnes pratiques ou pièges à éviter (boucles, rendu conditionnel, subsets JSX,)
- Découpage d'une application ou d'une page en composants
- L'intérêt de Typescript pour l'écriture d'applications REACT
- Exemples de travaux pratiques : les participants initialisent et créent un projet REACT avec l'outil de développement VITE. Ils créent un composant qui reçoit des données avec des props. L'objectif est de rendre un composant interactif (le composant change en fonctions d'événements qui altèrent son état)

Jour 2 :

Créer des composants fonctionnels

- Composition de multiples composants
- Cycle de vie d'un Composant
- Mise en place d'une navigation (mécanismes de routing)
- Chargement différé et optimisation du build (code splitting) avec le lazy loading
- Utilisation des Hooks (useEffect, useMemo...)
- Gestion des formulaires (état et validation des données)
- Consommer une API REST avec Axios ou l'API native fetch de JS
- Créer un formulaire et traiter ses données et traiter ses données de A à Z (validation, envoi...)
- Exemples de travaux pratiques : les participants suivent un énoncé qui permet de manipuler chaque cas de façon indépendante. L'objectif étant de construire un formulaire permettant aux personnes de renseigner des informations (par exemple saisir un nouveau produit, se connecter)

Utiliser Flux et gérer les bugs

- Limites de l'approche MVC
- Présentation de l'API Context de REACT
- Création et utilisation d'un contexte REACT, avec son système de
- Présentation de Flux et Redux
- Création un store (à la mode Redux avec Store, Action, Reducers)
 - Initialiser, consommer et modifier le store depuis les composants
 - Restaurer un état depuis le store
- Exemples de travaux pratiques : les participants suivent un énoncé qui permet de manipuler chaque cas de façon indépendante. Le scénario proposer permet de comparer les 2 approches : Redux Stores ou React Context Providers. Par exemple avec la gestion d'un panier de produits (ajout au panier, supprimer du panier, calcul du total...)

Jour 3 :

Tester son développement

- Compiler et déployer une application ReactJS
- Mesurer la performance côté client
- Aperçu des outils de Build
- Mesurer la performance avec React DevTools
- Exemples de travaux pratiques : le formateur réalise différentes manipulations pour les configurations et l'exécution des builds. Les participants reproduisent les commandes et les manipulations.

Identifier les impacts des architectures RIA

- Principe et bénéfices du développement isomorphe (fullstack JS)
- Brainstorming : le formateur interroge les participants concernant les avantages ou inconvénients des RIA comme React ou des architectures fullstack. Chacun peut faire part de son retour d'expérience, ou commenter.

★ **Prérequis**

- Connaissances pratique du développement Web
- Maîtrise et pratique de JavaScript (ES5 minimum)
- Un questionnaire de positionnement sera réalisé au préalable pour adapter la formation à vos besoins



Modalités pédagogiques

- Exercices concrets
- Cas pratiques
- Quiz d'évaluation des connaissances



Moyens et supports pédagogiques

- Mise à disposition de postes de travail
- Vidéoprojecteur, tableau blanc, supports numériques
- Documentation pédagogique numérique



Modalités d'évaluation et de suivi

- Positionnement en amont de la formation :
 - Un quiz de consolidation des pré-requis sera administré en amont de la formation
- Suivi « pendant » :
 - Feuilles de présence
 - Exercices pratiques
 - Évaluation « fin de formation »
 - Évaluation des acquis en fin de formation
 - Formulaire d'évaluation de la formation
- Évaluation à froid :
 - Suivi post-formation : Questionnaire de satisfaction à j+30