



# React

Durée 3 jour(s) (REACT-03)

Développement d'applications dynamiques avec React

## Description

React est une librairie née des problématiques de performance, productivité et maintenabilité rencontrées par les équipes de développement de Facebook et Instagram. Il en résulte une librairie puissante et agréable à utiliser. Elle nécessite toutefois d'être utilisée dans un environnement technique adéquat et a pour ambition de "réécrire" les bonnes pratiques. A nous de vous donner toutes les bases nécessaires pour tirer la quintessence de React.

## Objectifs

- Comprendre la philosophie de React
- Appréhender le modèle Redux, Flux & co
- Construire une Single-Page Application basée sur React

### Public

- Développeur web

### Prérequis

- Maîtrise du langage Javascript
- Connaissance des technologies web

### Répartition

40% Théorie, 60% Pratique

## Programme

## Introduction

- Paysage des frameworks JS
- Fondamentaux de NodeJS et NPM
- Historique de React

## React

- La philosophie
- Le fonctionnement interne : Virtual DOM, reconciliation
- Le package : Addons, react-tools, JSXTransformer, React Developer Tools

## Rendu des composants React

- Notre premier composant
- API React
- Rendu DOM
- JSX
- TP : Création guidée d'une application en composants React

## États des composants React

- État interne d'un composant (State React)
- Propriétés d'un composant (Props React)
- Validation des propriétés
- TP : Animation de l'application fil rouge part les props et state react

## Cycle de vie des composants React

- Présentation des Hook React de montage d'un composant
- Présentation des autres Hook React
- Initiation aux performance d'un composant React

## Environnement CommonJS

- CommonJS
- ESModules
- Création du bundle client
- TP: Découpe de l'application en modules et création d'un bundle avec Webpack

## Tests

- Lanceur de tests
- Test Driven Development
- Test d'une fonction

- Mocking
- Test d'un composant React
- TP: Création guidée des tests de l'application

## Architecture REST

- Présentation de l'architecture
- Interactions entre une application React et un Serveur REST
- TP : Récupération de données sur un serveur NodeJS

## Redux

- Le workflow unidirectionnel
- Flux
- Les principales composantes de Redux
- Redux dans le contexte React
- TP: Mise en place de Redux + Tests

## Routage

- Le projet react-router
- TP: Passage de l'application en Single-Page Application avec React-router

## Formulaire

- Les formulaires
- Validation d'un formulaire React
- TP: Création et validation d'un formulaire avec React + Tests

## Performances

- Diminution des reconciliations avec shouldComponentUpdate
- Utilisation de PureRenderMixin
- Addon de mesure de performances
- TP: Analyse des performances et tuning de l'application

## Isomorphisme

- Présentation du concept
- Isomorphisme et les données
- Isomorphisme et redux
- TP: Transformation du TP fil-rouge en application isomorphique

## Aller plus loin