

Formation Redux et RXJS

3 jours (21 heures)

Présentation

La programmation réactive fonctionnelle (FRP) vous permet de mieux structurer votre code métier et de gérer précisément les événements provenant de l'interface utilisateur. Avec ce système, tout devient un flux d'information contrôlable. Fini le code spaghetti et les interdépendances ! De cette manière il vous sera extrêmement rapide de traiter des données venant de nombreuses sources différentes et de les lier très simplement à votre interface d'application riche. Dans cette formation, nous aborderons les technologies les plus avancées dans le domaine du Web à savoir : [RxJS](#) ([ReactiveX](#) / Reactive Extensions) et [Redux](#).

Comme dans toutes nos formations, celle-ci vous présentera la toute dernière version de Redux (à la date de rédaction de l'article : [Redux 4.0](#)).

Objectifs

- Comprendre les enjeux de la programmation réactive fonctionnelle (FRP)
- Appréhender les concepts d'Actions, de Reducers, Store et de Data Flow
- Mettre en place une application en utilisant Redux et RxJS
- Synchroniser ses données avec le serveur
- Maîtriser les concepts avancés et les bonnes pratiques des technologies ReactiveX, RxJS et Redux

Public visé

Développeurs, Architectes

Pré-requis

Connaissances dans un langage orienté objet. Nous vous conseillons d'avoir suivi la formation [React](#) ou [Angular](#).

Programme de la formation Redux et RxJS

Introduction : Qu'est-ce que la programmation réactive ?

- Influences (Elm, ES6, Flux...)
- Data Flow : Unidirectionnel et Two way Data Binding
- Les 3 principes (Single source, State Read Only, Changes pure functions)
- Écosystème
- Travaux Pratiques : Exemple simple de mise en œuvre

Les bases de RxJS

- Qu'est-ce qu'un Observable
- Rx Pattern & Observable
- Hot & Cold Observables
- Travaux Pratiques : Mise en place et création d'un Observable

Core Concepts : Les bases de Redux

- Actions
- Reducers
- Store
- Data Flow
- Utilisation sous React
- Travaux Pratiques : Création d'une liste dynamique de chose à faire

Les concepts avancés de la programmation asynchrone

- Async Actions
- Async Flow
- Middleware
- Utilisation de React Router
- Travaux Pratiques : Mise en place d'une API Reddit-like

Redux Patterns

- Améliorez vos fonctions : les réduire et extraire les paramètres
- Travaux Pratiques : Reducer Patterns
 - Ajout & Suppression d'éléments dans une liste
 - Ajout / Suppression et Changement de propriété sur un objet

Construire son application

- Construire son application
- Actions / Views / Storage
- Méthodes et environnement (Babel, Webpack...)
- Travaux Pratiques : Implémentation
 - Action Creators
 - Reducers
 - Modification d'état sans librairie tiers
 - Selectors
 - Connecter son App à votre interface UI

Synchronisez vos données avec le serveur

- Actions & Server
- API module
- Action creators
- Updating reducers
- Sauvegarde et récupération des données
- Alternatives à redux-thunk

Persistance d'état coté client

- Persistance : Automatique VS Manuelle
- Manuelle : persistance et restauration d'état
- Automatique : persistance et refactorisation

Analytique du Middleware : Suivre les interactions des utilisateurs

- Pourquoi est-ce utile ?
- Spécification et création du middleware
- Enrichir les actions grâce aux méta-données
- Accéder à l'ensemble de vos états au sein de vos track function
- Travaux Pratiques : Implémentation finale

Structurez vos Reducers

- Structurer vos Reducers
- Concept
- Structure basique d'un Reducers
- Séparer les logiques (Reducer Logic)
- Exemple de refactorisation
- Usage de combineReducers : mettre à jour vos états
- Normaliser vos données relationnelles
- Mettre à jour vos données
- Factoriser et réutiliser vos Reducers Logic
- Pattern : Immutable Update
- Initialisation d'état

Les bonnes pratiques

- Action : Éviter les ambiguïtés
- Reducer Initial state
- Quelques astuces sur les Selectors, Middleware et Store Enhancers
- Bien Structurer de vos projets (Concept Type, Organisation par domaines et fonctionnalités)
- Outils de développement, debug & astuces de test

Maîtrisez de façon avancée Redux et bien l'appréhender

- Bonnes pratiques pour migrer sur Redux
- Usage de l'Object Spread Operator
- Action/Reducer : Boilerplate pattern
- Gestion du côté serveur : Server Rendering
- Tester son code
- Isolation des Sub-Apps
- Manipuler vos données grâce à la library Reselect
- Travaux Pratiques : Créer une gestion d'historique autorisant l'annulation

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.