

Formation RxJS

RxJS est une librairie JavaScript pour la programmation réactive fonctionnelle. Paradigme essentiel notamment à la conception d'interfaces graphiques, la programmation réactive permet d'écrire du code facile à tester et à maintenir. Implémentation JavaScript de ReactiveX, et bibliothèque la plus populaire du marché, RxJS est souvent utilisée de concert avec Angular. A l'issue de cette formation RxJS, vous maîtriserez les concepts de programmation réactive fonctionnelle, les bases de RxJS ainsi que des concepts avancés de la librairie. Vous saurez manier ses opérateurs, ses observables et l'intégrer à une application Angular.

Durée

2 jours

Objectifs pédagogiques

- Se familiariser avec la programmation réactive et la bibliothèque RxJS
- Utiliser les opérateurs de base pour manipuler des flux de données
- Gérer les erreurs
- Maîtriser l'utilisation des Subjects et le multicasting
- Intégrer RxJS dans une application Angular
- Mettre en œuvre les meilleures pratiques de développement RxJS

Public

Développeurs, architectes, chefs de projet

Prérequis

Maîtrise de JavaScript. Pour l'intégration de RxJS dans une application Angular, des connaissances de base sur Angular sont également conseillées.

Programme de formation

Introduction à la formation RxJS
Programmation fonctionnelle et programmation réactive
JavaScript
Les concepts de base dans RxJS : observable, observer, subscription et subject

Les opérateurs

Manipuler les flux de données avec map, filter, mergeMap, switchMap et concatMap
Utiliser scan, startWith, delay et debounceTime
distinctUntilChanged, takeUntil, takeWhile

Observables et opérateurs de fusion

merge, concat, zip : combiner plusieurs flux de données en un seul flux
Utiliser forkJoin et combineLatest
Combiner les dernières valeurs de plusieurs flux de données en un seul avec fluxwithLatestFrom

Erreurs et retours vides

Eviter que les erreurs ne se propagent avec catchError
retry pour relancer un flux de données

Gérer les données non disponibles avec defaultIfEmpty
Créer un flux de données avec of
Générer une erreur avec throwError

Les Subjects et le multicasting
Les différents types de Subject : BehaviorSubject, ReplaySubject, AsyncSubject
Utiliser multicast avec ConnectableObservable

Marble et les tests unitaires

Présentation du Marble Testing
Créer un test stream, utiliser des opérateurs...
Les bonnes pratiques pour écrire des tests avec Marble

Meilleures pratiques, trucs et astuces
Gestion de la mémoire et performance
Modèles de programmation réactive avancés
Débuguer et développer avec RxJS

Moyens et méthodes pédagogiques

- La formation alterne entre présentations des concepts théoriques et mises en application à travers d'ateliers et exercices pratiques.
- Les participants bénéficient des retours d'expérience terrains du formateur ou de la formatrice
- Un support de cours numérique est fourni aux stagiaires

Modalités d'évaluation

- **En amont de la session de formation**, un questionnaire d'auto-positionnement est remis aux participants, afin qu'ils situent leurs connaissances et compétences déjà acquises par rapport au thème de la formation.
- **En cours de formation**, l'évaluation se fait sous forme d'ateliers, exercices et travaux pratiques de validation, de retour d'observation et/ou de partage d'expérience.
- **En fin de session**, le formateur évalue les compétences et connaissances acquises par les apprenants grâce à un questionnaire reprenant les mêmes éléments que l'auto-positionnement, permettant ainsi une analyse détaillée de leur progression.