

## EmberJS

### Développer une application internet riche (RIA) avec EmberJS

EmberJS est un framework permettant de créer des applications ambitieuses. Il vous permet d'écrire moins de code avec une gestion de template intégrée « ember handlebars ». Ce framework permet également d'utiliser des composants spécifiques qui étendent les tags HTML pour des comportements personnalisés. Il facilite également l'accès aux données serveur avec une API REST native. Ce stage de 2 jours vous permettra de créer des interfaces riches et réactives pour vos applications web.

#### Détails

- **Code** : DW-EMB
- **Durée** : 2 jours ( 14 heures )

#### Public

- Chefs de projets
- Développeurs
- Web Designers

#### Pré-requis

- Avoir suivi le stage JavaScript ou posséder les connaissances équivalentes

#### Objectifs

- Comprendre les principes de Ember.js
- Mettre en oeuvre les bonnes pratiques de développement

#### Programme

##### Introduction

- Développement Javascript objet, prototypage et bonnes pratiques
- Origines et installation d'Ember.js

##### Environnement de développement

- Sources et dépôt Git
- Yeoman et générateur Ember.js
- Gestion de build et dépendances avec Bower et Grunt
- Debug avec Ember Inspector pour Chrome et Firefox
- Création d'une première application

##### Les bases Ember.js

- Exemple simple Hello World
- Concepts : templates, router, composants, models, route, controllers
- Conventions de nommage : application et route
- Modèle objet : classes et instances, propriétés dynamiques
- Événements : observers et bindings
- Utilisation des mixins

##### Routage et gestion d'état

- Création d'un router d'application
- Mapping d'URL avec les routes
- Cycle de changement d'état
- Configurer le router, définir des chemins
- Gestionnaire de ressource et model d'une route
- Routage asynchrone
- Configurer un controller de route et un template

##### Les templates d'application

- Présentation Handlebars.js
- Déclarer, insérer et écrire les templates
- Les variables Ember.js
- Les helpers link, input et each

- Les conditions et les interactions utilisateur
- Bound et unbound expressions
- Étendre les templates, créer ses propres helpers

##### Les controllers

- Déclarer, attacher aux models
- Afficher des données dynamiques des controllers
- Les propriétés et propriétés calculées
- Événements, utilisation des Promises, types de controllers
- Méthodes des controllers
- Gestion des vues

##### Les vues et la gestion d'événement

- Déclarer les vues
- Associer : le controller, le template, les tags d'élément
- Mettre à jour les attributs
- Insertion : dans le DOM, dans les templates
- Les layouts
- Gestion des événements
- Composants Ember.js : champs de saisie, containers

##### Composants

- Description et déclaration
- Différence entre vue et component
- Les propriétés
- Définir les types de tag, class et attributs
- Gérer les événements, définir des actions
- Les layouts

##### Persistance de données avec REST

- Les requêtes Ajax avec Ember.js
- Présentation de Ember-data
- Créer un data store
- Models, relations, accès aux données
- Abstraction : Store, Serializer et Adapters

## Logs, debug et tests

- Logs et debug : Objets, routers et routes, templates, controllers et views
- Utiliser Ember.js Inspector
- Gestion d'erreur
- Écrire des tests
- Helpers de test : synchrones et asynchrones
- Les tests unitaires : computed properties, appels de méthodes et observers
- Tests d'intégration
- Build de l'application

## Modalités

- **Type d'action** :Acquisition des connaissances
- **Moyens de la formation** :Formation présentielle – 1 poste par stagiaire – 1 vidéo projecteur – Support de cours fourni à chaque stagiaire
- **Modalités pédagogiques** :Exposés – Cas pratiques – Synthèse
- **Validation** :Exercices de validation – Attestation de stages