

# Formation Play ! Framework 2

## Objectifs de la formation play framework

Play 2 ! est un framework complet dédié au Web pour créer rapidement des applications web en Java ou en Scala. Prenant sa distance par rapport à l'écosystème JEE, Play 2 ! a été pensé pour rendre le développement plus productif et plus fun. Simple, puissant, moderne, adaptable ; ses possibilités font de lui un framework résolument tourné l'avenir.

Cette formation Play framework vous permettra de :

- Situer le framework Play ! dans l'écosystème Java
- Concevoir, développer et industrialiser une application Web Play !
- Acquérir quelques bonnes pratiques de développement dans cet environnement

## À qui s'adresse cette formation ?

### Public :

Cette formation Play framework s'adresse aux développeurs, aux ingénieurs et aux chefs de projet

### Prérequis :

Connaissance intermédiaire des bases du langage Java

## Contenu du cours play framework

### Play ! Dans l'écosystème Java

Rappels sur l'architecture d'application Web. Comprendre le besoin. Solutions Java concurrentes. Avantages du Framework Play ! Concepts mis en œuvre Variables Opérateurs Structures de contrôle Commentaires Tableaux

## **Fondements du langage**

Simplicité et productivité API Restful Full stack : mise à disposition des composants natifs pour le développement Web.Stateless Montée en charge

## **Première application**

Installation Console Play ! Commandes de base Structure de l'application. Développement avec un IDE Débogage Configuration de l'application en environnements d'entreprise Déclaration des URLs (HTTP Routing).

## **Conception de l'application Web MVC**

Design Pattern MVC et mise en œuvre avec Play ! Le Modèle intégration de JPA / ORM.Le Contrôleur Les "Actions" Fourniture de la réponse. La Vue Templating (Scala, Helpers)

## **Boîte à outils Play !**

Le ressources (css, javascript, images). Intégration aux autres framework Web ! (Google closure, require.js, css bootstrap). Cache et gestion de la problématique du stateless. Web Socket avec socket.io.

## **Industrialisation**

Tests Unitaires Tests fonctionnels Tests des templates, des routes, des controllers.Tests D'intégration Intégration continue. Déploiement Dépendances et versioning Configuration Avancée Packaging de l'application Sécurisation Cloud computing

Traitement des formulaires de saisie Scopes Web Programmation asynchrone