

## FILIERE DEVELOPPEUR JAVA

### METIER VISE

---

L'objectif du développeur Java est de concevoir des applications en Java SE et en Java EE (Standard Edition et Enterprise Edition).

Il développe des composants, des services ou des sites internet en se basant sur un cahier des charges établi par son chef de projet.

Pour cela, il analyse les besoins du client puis choisit la solution technique. Il développe ensuite toutes les fonctionnalités techniques de l'application. Il finit par tester, intégrer et valider les fonctionnalités développées.

Rattaché au chef de projet technique, le concepteur bénéficie d'une liberté de création. Il doit cependant tenir compte des délais souvent serrés. La rigueur est son mot d'ordre. Il doit enfin être à l'écoute de ses clients et se tenir régulièrement informé sur les développements technologiques.

### PUBLIC

---

- Bac+2 informatique
- Bac + 5 informatique, ou scientifique avec une expérience dans le développement informatique

### PRE REQUIS

---

- Connaissance de l'anglais technique
- Connaissance de l'outil informatique
- Des notions d'algorithmique
- Des notions en langage (langage de présentation HTML, langage basique)

### DUREE DU CURSUS

---

- 57 jours / 399 heures

### OBJECTIFS

---

- Acquérir les bases de toute conception de logiciel (concept Objet, UML)
- Concevoir et développer des applications en Java Standard Edition (Java SE)
- Concevoir et développer des interfaces graphiques riches (en HTML, CSS, JavaScript)
- Développer des applications en Java Enterprise Edition (Java EE : Servlet, JSP)
- Construire des socles applicatifs intégrés autour de Spring
- Développer et intégrer des services Web

## PROGRAMME

---

### Intégration et travail en équipe – 1 jour

### La place du développement dans le projet informatique – 2 jours

- Les éléments essentiels à la vie d'un projet
- Le rôle du concepteur développeur dans un projet

### Algorithmique – 2 jours

- Structures logiques
- Démarche de construction

### La conception objet – 2 jours

- Les bases de la conception objet
- Les bases des outils de développement
- La notation UML

### Conception de bases de données relationnelles – langage SQL – 3 jours

- La conception des bases de données relationnelles
- La manipulation du langage SQL

### Les fondamentaux Java – 5 jours

- Plateformes Java SE et Java EE, la machine virtuelle JVM, les outils JDK, eclipse
- Le langage Java : syntaxe, types de données, structure
- La gestion des erreurs avec les exceptions
- La manipulation des données
- Présentation des interfaces graphiques
- Les tests unitaires
- Création et déploiement d'une application Java SE

### JAVA WEB : Servlet, JSP, développement MVC – 5 jours

- Architecture MVC2
- Développement de composants Java EE : JSP, Servlet, Taglib
- Manipulation et création de bibliothèques de balises.
- Mise en œuvre sous un serveur Tomcat
- Développement et déploiement des applications Web Java EE sur les serveurs d'applications

### Java et la persistance des données – 5 jours

- Architectures d'accès aux données
- Modèle relationnel/modèle Objet
- Le standard JPA (Java Persistence API)
- Le standard Hibernate
- Développement et déploiement des applications sur les serveurs d'applications

### Java Spring – 4 jours

- L'accès aux services et ressources JNDI
- L'intégration de Spring avec Struts et Hibernate
- Le module MVC de Spring
- Les transactions
- Les connecteurs

Développer des services Java – Architecture SOA, REST – 5 jours

- Architecture SOA (Service Oriented Architecture)
- XML et Java
- Création d'applications clientes
- Fourniture de services
- Développement, déploiement et administration de services
- Sécurisation

HTML 5, CSS 3 et JavaScript avec Angular – 8 jours

- Développement d'un site web avec HTML 5
- Le graphisme avec CSS 3
- Développement avec JavaScript avec Angular

Travaux Pratiques – 15 jours

Les travaux pratiques sont basés sur la construction de 2 applications :

- Une application back-end full Java Springs, fournissant des services Rest / Json
- Une application Front-end HTML/CCS/Angular Single App, consommant les services du back-end

## Intégration et travail en équipe

Durée : 1 jour

---

### Objectifs

Ce cours a pour objectif :

- Une bonne intégration des participants au groupe en mettant en valeur l'importance de réussir cette phase dans toute prise de fonction
- Une prise de conscience de l'importance de la dimension relationnelle et comportementale dans l'atteinte des objectifs pour les amener à accorder de l'attention dans la relation à des aspects comme l'écoute, la disponibilité, la confiance, l'esprit de coopération, etc.
- Une valorisation de l'impact de la méthodologie choisie sur l'atteinte des objectifs pour amener les participants à développer une approche structurée de leur travail.
- Un accompagnement à l'ancrage d'une projection individuelle et collective positive à l'issue du parcours.

---

### Contenu

#### Accueil et Présentation du Parcours

- Présentation du Groupe Global Knowledge
- Présentation du programme du Parcours et des moments clés
- Présentation des règles générales d'organisation

#### Jeux pédagogiques et ludiques

Dans un cadre décontracté et une ambiance de jeu, les participants sont accompagnés tout au long de la journée à travers des jeux pédagogiques et ludiques pour les amener à s'intégrer et à prendre conscience de l'importance des dimensions non techniques et de leur impact sur l'atteinte de leurs objectifs et sur leur réussite dans leur rôle.

- Tirage au sort et organisation en équipes
- Jeu 1 : Plus tu connais plus tu gagnes
  - ✚ Principe : présentation croisée des participants
  - ✚ Objectifs pédagogiques : Se connaître les uns les autres en mettant en relief la richesse du groupe
- Jeu 2 : Les 5 objets
  - ✚ Principe : Transcription d'un schéma d'objets par commande distante
  - ✚ Objectifs :
    - Mettre en évidence l'importance de la phase de stratégie et planification dans tout projet
    - Analyser les causes possibles des écarts entre le message transmis et le message reçu
    - Valorisation de sens du détail et de la gestion optimisée du temps
- Jeu 3 : Le cercle des objets
  - ✚ Principe : Transmission de multiples objets au sein du groupe avec diverses contraintes
  - ✚ Objectifs pédagogiques :
    - Développement de la capacité à gérer la pression et le stress lié à la multiplicité des objets et à leur spécificité
    - Respect du récepteur dans l'acte de transmission et intégration de ses contraintes
    - Lien avec l'équilibre vie professionnelle/vie professionnelle d'une part et avec le renvoi des balles dans l'entreprise
- Jeu 4 : Le mur de la confiance
  - ✚ Principe : Aller vers un « mur de collègues »
  - ✚ Objectifs pédagogiques :
    - Prise de conscience des participants de leur distance spontanée de la notion de confiance quand le vis-à-vis en est incontestablement digne
    - Valoriser la dimension stratégie de l'équipe et son impact sur la performance individuelle et collective
- Jeu 5 : La corde au carré
  - ✚ Principe : Construction d'un carré parfait
  - ✚ Objectifs pédagogiques :
    - Renforcement de l'importance de l'organisation, de la planification et de la stratégie de groupe dans l'atteinte des objectifs
    - Mise en évidence des aptitudes du groupe à l'application minutieuse d'un plan de déploiement préétabli
    - Valorisation de la capacité du groupe à rebondir en cas d'imprévu
- Jeu 6 : La tour de papier
  - ✚ Principe : Construction de la plus haute tour
  - ✚ Objectifs pédagogiques :
    - Développement de la stratégie et de la différenciation
    - Accompagnement des participants à la valorisation du sens du détail et de la rigueur ainsi que leur

- impact sur la performance
- Renforcement du sens d'équipe et de l'importance de la planification et de l'organisation et la répartition des rôles
- Jeu 7 : Hymne du Groupe
  - ✚ Principe : Construire une chorégraphie d'équipe avec une chanson autour du métier
  - ✚ Objectifs pédagogiques :
    - Construction d'équipe
    - Activation des compétences de créativité des participants
    - Accompagnement des participants à la mise en avant des dimensions de leur futur métier
    - Valorisation du métier et projection positive des participants
    - Ancrage de l'émotion positive à travers la musique et la chorégraphie

#### Visualisation et débriefing

- La visualisation des différents exercices permet aux participants de prendre du recul face à leurs attitudes et comportement.
- Accompagnement vers une synthèse des enseignements de la journée autant à l'échelle individuelle que collective.
- Mise en avant de l'importance des notions suivantes :
  - ✚ Intégration au sein d'un nouvel environnement
  - ✚ Cohésion d'équipe
  - ✚ Travail en équipe
  - ✚ Stratégie et organisation du travail
  - ✚ Sens du détail et rigueur dans la mise en œuvre d'un plan d'action
  - ✚ Le récepteur est central dans l'acte de communication
  - ✚ Plaisir et travail

---

### Le plus de cette formation

Dans un cadre convivial, les participants sont accompagnés à passer d'un ensemble d'individus à une équipe qui apprend dans le plaisir et qui se projette positivement.

## La place du développement dans le projet informatique

Durée : 2 jours

Réf du cours : GKDEVPROJ

---

### Résumé

Cette formation permet de prendre connaissance des éléments essentiels à vie d'un projet informatique. Elle positionne le rôle du concepteur développeur dans le projet et les connaissances nécessaires à avoir pour mener à bien des développements logiciels.

---

### Public visé

Développeurs, Chefs de projets, Architectes techniques.

---

### Objectifs

- Identifier les composants d'une application informatique, les architectures possibles d'exécution des applications et l'apport des technologies objet dans le cycle de vie d'un projet informatique.
- Positionner la conception, la réalisation et les tests du développement applicatif pour mieux comprendre leurs intérêts et leurs évolutions.
- Appréhender le vocabulaire, les modèles, les technologies, les outils et Framework utiles à tout concepteur développeur.
- Se rendre compte de l'intérêt de l'industrialisation pour l'amélioration continue de la qualité des applications.

---

### Prérequis

Les participants doivent avoir des connaissances générales en informatique.

---

### Contenu

#### Introduction au Système d'Information

- Importance du système d'information dans l'entreprise.
- Les différentes composantes d'un système d'information.
- L'informatique, la DSI (direction des systèmes d'information) et ses services.

#### Les architectures

- Historique des architectures.
- Les différentes architectures : Client/serveur, Web, n- tiers,
- Les architectures orientées services (SOA, REST, Cloud).
- Les contraintes et les évolutions techniques (matériel/logiciel).
- La virtualisation du poste de travail et des applications.

#### La gestion des applications

- Les composants d'une application et le modèle en couche.
- Les serveurs de données : des bases de données relationnelles au Big Data.
- Les serveurs d'applications.
- La présentation aux utilisateurs : les différentes interfaces homme/machine (IHM), du client lourd, client léger au client riche vers la mobilité.
- Les applications cloud, mode SaaS (Software as a Service).

#### Le projet informatique

- Définitions et objectif.
- Méthodologie et méthode de projet informatique.
- Nécessité de modéliser.
- Maîtrise d'œuvre / Maîtrise d'ouvrage, MOA/MOE.

#### Les phases du projet informatique.

- Présentation des différentes phases d'un projet informatique.
- Les acteurs du projet et leurs rôles.
- L'évolution des démarches : du linéaire à l'itératif (cycle en V) et incrémental (spirale de Boehm) vers l'Agile (Scrum).
- L'importance de la communication, de l'échange, le travail en équipe.

#### Conception, réalisation et tests.

- La place du développement et le rôle des acteurs (concepteur, architecte, développeur).
- La conception avec UML, historique et évolutions.
- Les Design patterns les plus utilisés (MVC, DAO...).
- Les langages du Web : HTML, CSS, JavaScript, XML, Ajax, JQuery.
- Les langages objet, les plates-formes Java EE et .Net.
- L'importance des tests, les environnements de tests, xUnit.
- Les outils et Framework nécessaires à la mise en œuvre d'applications.

#### **L'industrialisation du développement logiciel**

- Le logiciel développé, ses dépendances, l'inversion de contrôle (IoC) pour limiter les dépendances, ses modules externes ou internes, l'enchaînement des tâches.
- La gestion et l'automatisation de la production de logiciel (Ant, Maven).
- L'intégration continue sur des plates-formes d'intégration (Hudson, Jenkins).
- La qualité du logiciel produit.

#### **Les référentiels de qualité et les certifications.**

- Présentation des référentiels : CMMI, ITIL, PMP, Lean, Prince 2...
- Les champs d'applications.
- Les certifications.

## Algorithmique

Durée : 2 jours

---

### Résumé

Cette formation permet de s'approprier les structures logiques de base ainsi qu'une démarche de construction de solutions à partir d'un problème.

---

### Public visé

Développeurs

---

### Prérequis

Aucune connaissance informatique n'est nécessaire pour ce stage.

---

### Contenu

#### L'algorithme dans le contexte projet

- De la demande utilisateur à la réalisation du logiciel.
- Qu'est-ce qu'un algorithme ?
- En quoi est-il indispensable en programmation ?
- Qu'entend-on par programmation procédurale ?
- Qu'en est-il de la programmation événementielle, de la programmation objet ?

#### L'algorithmique de base

- Intérêt de l'utilisation d'un langage symbolique.
- Les séquences alternatives simples et itérations simples.
- Les opérateurs logiques ET, OU et NON : Algèbre de Boole et loi de Morgan.
- Analyse : la démarche descendante par étapes.
- Les tableaux à une dimension.
- Les alternatives et débranchements complexes.
- Les différents types de données.

#### L'analyse d'un programme

- Intérêt de la structuration d'un programme.
- La modularisation d'un programme.
- La structuration d'un programme.
- Intervention en maintenance dans un programme structuré.
- Modules externes, procédures et fonctions.
- Différence entre procédure et fonction ?
- Transmission de paramètre.

#### Les tests

- Intérêt et mise en œuvre.

#### Les fichiers et les bases de données

- Les fichiers : notions d'enregistrements, de rubriques.
- Les différentes organisations de fichiers.
- Etude détaillée des fichiers séquentiels et séquentiels indexés.
- Les bases de données relationnelles : organisation et manipulation des données



## La conception objet

Durée : 2 jours

Réf de cours : GKOBJ\_2J

---

### Résumé

Cette formation permet d'acquérir les connaissances de base solides sur la conception objet et celle basée sur les composants, en manipulant un langage objet comme Java, C#, C++.

Les participants seront capables de comprendre et de raisonner en classes et objets, d'appréhender la notation UML et la nécessité de la modélisation. Il s'agit d'assimiler les architectures et technologies objets avant d'entreprendre l'étude d'un langage objet ou des développements "orientés objets".

---

### Objectifs

- Connaître les concepts essentiels de l'Objet pour faciliter l'approche de tout langage de programmation Orienté Objet.
- Connaître les manipulations de base d'un outil de développement (Eclipse, Visual Studio...).
- Appréhender la modélisation UML pour l'analyse fonctionnelle et technique.

---

### Public visé

Développeurs, Concepteurs, Chefs de projets, Architectes techniques.

---

### Prérequis

Les participants doivent avoir des connaissances générales en informatique, notamment en matière de conception d'applications.

---

### Contenu

#### Objectifs de la conception Objet

- Historique de l'objet.
- L'objet par rapport aux concepts traditionnels.
- Les principes fondamentaux de la programmation orientée "objet".
- La modélisation Objet avec la notation UML.

#### Développement d'applications objets, l'industrie de l'objet

- Les langages Objet (Java, C++, C#, VB net, PHP...), les plateformes (Java EE, .Net).
- L'objet et les bases de données relationnelles.
- Le modèle en couches dans les architectures.
- Les objets métier, les composants.
- Le principe de la réutilisation des objets dans différentes applications (client lourd, client léger, client riche).
- Les outils de développement, les IDE (Integrated Development Environment) : Eclipse, NetBeans, Visual studio...

#### Classes - Encapsulation - Instanciation

- Les packages et les espaces de noms, les bibliothèques.
- Les classes, les attributs, les méthodes (constructeurs, destructeurs et accesseurs).
- L'encapsulation.
- L'instanciation, les objets.
- La modélisation UML, les diagrammes de classes et d'objets.
- Traduction des concepts Objet en langage, mise en œuvre sous l'outil de développement.

#### Liens entre les classes : Composition, agrégation

- Définition et utilité des liens de composition et d'agrégation.
- La représentation UML des liens dans le diagramme de classe.
- Des concepts au langage, mise en œuvre sous l'outil de développement de liens de composition.

#### Liens entre les classes : Héritage simple et multiple

- Définition et utilité des liens de généralisation/spécialisation.
- Le polymorphisme, la surcharge.
- La représentation UML des liens d'héritage.
- Les classes abstraites.
- Les références à l'objet.
- Des concepts au langage, mise en œuvre sous l'outil de développement.
- L'héritage multiple et les limites dans les langages Objet.

#### Les interfaces

- Définition et intérêts des interfaces.
- L'héritage multiple avec les interfaces.
- Les interfaces dans les architectures distribuées.
- Des concepts au langage, mise en œuvre sous l'outil de développement.

#### Les tests dans le développement d'applications objets

- Les tests unitaires avec l'outil de développement.
- Manipulation de l'éditeur de source, du debug.
- Exécution de l'application pas à pas, affichage des valeurs des variables, des objets

### Autre(s) Information(s) :

Un support de cours en français sera remis aux stagiaires.

Cette formation permet aux participants une mise en pratique progressive de tous les concepts objets, via un langage objet dans un outil de développement. Une application simple sera mise en œuvre afin d'aborder la conception et la modélisation objet et d'en comprendre les tenants et les aboutissants. Le participant sera ainsi opérationnel pour débiter dans le développement d'application et ce quel que soit le langage objet choisi, l'outil de développement et le type d'application à mettre en œuvre (Web, mobile,...).

## Les bases du SQL

Durée : 3 jours

Réf du cours : GKSQ

### Résumé

SQL est le langage universel des bases de données relationnelles, quel que soit le système de gestion de la base (Oracle, DB2 d'IBM, Access et SQL Server de Microsoft, MySQL, MariaDB, PostgreSQL de l'open source...). Langage incontournable dans les développements d'application, le SQL est aussi utilisé dans les outils d'analyse de données. Cette formation permet aux participants de comprendre la conception des bases de données relationnelles et d'acquérir les compétences et connaissances nécessaires pour manipuler le langage SQL.

### Public visé

Cette formation s'adresse aux développeurs, administrateurs, architectes de base de données et utilisateurs de la Business Intelligence

### Objectifs

- Comprendre la structure des tables, les clés primaires et étrangères du modèle relationnel.
- Etre capable de faire des consultations, sur une ou plusieurs tables d'une base de données relationnelle, à l'aide du langage SQL.
- Effectuer des mises à jour de données (création, modification et suppression de lignes), des créations de tables et de vues

### Prérequis

Pour suivre cette formation, il faut avoir des connaissances générales en informatique.

### Contenu

#### Principes et concepts du modèle relationnel

- Les domaines, les relations, les tables, les lignes et colonnes.
- Les dépendances fonctionnelles, les clés (candidates, primaires).
- Les intégrités d'entité, les intégrités référentielles (clés étrangères), les valeurs NULL.
- Normalisation des données : les formes normales.
- Utilité de la normalisation pour la base et conséquences lors des accès.
- La modélisation des données (MCD Modèle Conceptuel de Données, MLD (Modèle Logique de Données)).

#### Le modèle relationnel dans la technologie Objet

- La modélisation avec les diagrammes de données d'UML (Unified Modeling Language).
- La correspondance entre le relationnel et l'objet (Mapping O/R)
- La persistance des données.

#### Les requêtes avec le langage SQL

- Structure générale de l'ordre SELECT.
- Sélectionner la table, les colonnes.
- Sélectionner le résultat : restrictions avec la clause WHERE. Les opérateurs de restrictions : LIKE, BETWEEN, IN.
- Trier les résultats avec la clause ORDER BY.
- Accéder à plusieurs tables, les jointures internes : INNER JOIN. Les sous-requêtes.

- Effectuer des statistiques avec les fonctions d'agrégation : COUNT, MAX, MIN, AVG, SUM.
- Partitionner les résultats statistiques : GROUP BY et HAVING.
- Les autres opérateurs ensemblistes (UNION, EXCEPT et INTERSECT).
- Accéder à plusieurs tables : les jointures externes : OUTER JOIN.
- Les fonctions SQL (manipulations de chaînes de caractères, manipulations de dates...).

#### La mise à jour de données avec le langage SQL

- Valider et invalider des mises à jour (Commit/Rollback)
- Créer des lignes dans une table : INSERT.
- Modifier des lignes : UPDATE.
- Supprimer des lignes : DELETE.
- Conséquences des intégrités référentielles.

#### Travailler avec des tables et avec des vues

- Présentation des vues.
- Intérêt d'une vue.
- Création et manipulation d'une vue avec SQL.

#### Evolutions des données d'entreprises

- Du SQL au décisionnel, l'intelligence décisionnelle ou la Business Intelligence.
- Du décisionnel au Master Data Management (MDM).
- Le Big data.

## JAVA Programmation : Maîtriser les fondamentaux de la programmation Java

Durée : 5 jours

Réf du cours : GKJAVA

### Résumé

Cette formation permet de prendre en main l'environnement JAVA et d'en utiliser les outils de développement. La plate-forme Java est une solution de référence dans le développement d'applications. Elle offre un langage orienté objet, de nombreuses API de haut niveau et la portabilité en exécution fournie par la Machine Virtuelle Java. La richesse des API Java et l'orientation Objet induisent la nécessité d'une démarche structurée en termes de Conception et d'Analyse Objet. Une modélisation avec UML des classes et Entités de l'application va de pair avec la maîtrise des API et de la Syntaxe Java.

### Public visé

Cette formation s'adresse aux développeurs d'applications, concepteurs ou architectes techniques.

### Objectifs

- Prendre en main l'environnement et les outils de développement Java
- Etre capable de décrire les différentes technologies de Java (Java SE, Java EE, Java embarqué, ...)
- Maîtriser les types de données et la syntaxe du langage Java
- Etre en mesure de créer et déployer une application Java
- Mettre en œuvre la gestion des erreurs, les exceptions
- Maîtriser les concepts objet et l'architecture logicielle en couches
- Appréhender la modélisation UML pour l'analyse fonctionnelle et technique
- Manipuler les données avec JDBC et connaître les problématiques de performance et de cohérence des données
- Encapsuler proprement les exceptions techniques dans un Framework d'exceptions métier
- Utiliser les bonnes pratiques de qualité avec les tests unitaires et la documentation
- Appréhender les applications client riche avec Swing
- Appréhender les interfaces graphiques

### Prérequis

Il est nécessaire d'avoir suivi les formations GKSQL « Les bases du SQL » et GKCOBJ « Les concepts : de la modélisation au développement objet » ou posséder les connaissances équivalentes.

### Contenu

#### Rappel de la programmation Orientée Objet

- Les classes, les attributs, les méthodes
- L'instanciation, les objets
- L'encapsulation
- L'héritage, le polymorphisme

#### Introduction Java, Java SE

- Les origines de Java, son historique
- Le JDK (Java Development Kit), ses versions
- L'outil de développement : Eclipse
- La plate-forme Java SE 8, Standard Edition
- Les principaux apports des versions Java 5 à Java 8
- Le fonctionnement de Java, la machine virtuelle (JVM), les exécutable, la gestion de la mémoire (garbage collector)
- La documentation, Javadoc

#### Les architectures logicielles et les techniques Java

- Architecture logicielle et technique en couches
- La plate-forme Java EE Enterprise Edition
- Les principaux composants Java EE (EJB, Servlet, JSP)
- Java embarqué, Android

#### Le langage de programmation Objet de Java

- Les packages, les classes, les méthodes
- Les objets, les variables, les composants
- La création du projet Java sous Eclipse

#### Les bases du langage

- Les commentaires pour la Javadoc
- Généralités, littéraux, opérateurs, annotations
- Les structures de contrôle

#### Création d'application

- Structure générale d'un traitement
- La construction d'un composant métier
- La séparation Conceptuel/Visuel
- Tests associés au composant
- JUnit : outil de tests unitaires

#### Les chaînes de caractères

- La classe String
- Les constructeurs, les méthodes
- Les conversions de types

### Les structures de données

- Les tableaux
- Les classes de données
- Les Collections, les Maps
- Les énumérations, les itérateurs
- Les Génériques

### Les exceptions

- La gestion des erreurs
- La syntaxe Java pour gérer les exceptions
- Définir ses propres exceptions
- Exécution de l'application avec et sans la gestion des erreurs

### La communication d'objets Java Bean/Thread

- Présentation des Beans
- Cycle de vie de l'objet géré par la JVM
- Thread : processus et multi-tâches

### Classes abstraites et interfaces

- Intérêt des classes abstraites et des interfaces
- Mise en œuvre des interfaces
- Les interfaces dans les architectures distribuées

### Présentation d'interface homme machine IHM

- Introduction aux interfaces graphiques (AWT, Swing, SWT)
- Réalisation d'interfaces avec Visual Editor
- Gestion des événements utilisateur avec les Listeners
- Les classes internes et anonymes

### Accès aux bases de données relationnelles

- Accès aux données avec l'API JDBC et les DAO
- Les composants Java mis en œuvre : les drivers, les requêtes, les connexions, les ensembles de résultats
- Gestion et encapsulation des exceptions
- Gestion des transactions
- Création, récupération, mise à jour et suppression de données dans la base (CRUD)
- La persistance des objets dans une base relationnelle
- Le mapping relationnel/objet
- Les solutions avec EJB, Hibernate, JPA

### Entrées sorties

- Accès au système de fichiers
- Les fichiers bruts, les fichiers textes et les fichiers de propriétés
- Récupération de propriétés
- Les objets et la sérialisation
- La persistance des objets dans un fichier sauvegarde et restauration d'objet

### Le projet Java réalisé sous Eclipse

- Création d'une archive java : .jar
- Exportation et importation de projet
- Génération de la documentation de projet

### Présentation de quelques API Java 8

- API Date and Time : concepts de date, d'heure, d'instant, de durée et de période temps. Classes immutables, conception liée au domaine, séparation des chronologies. Formatage et conversion de date
- Les expressions Lambdas
- Les Streams

## JSP Servlets : Développer des applications Web MVC2 avec Java EE

Durée : 5 jours

Réf du cours : GKJSPWEB

### Résumé

La plupart des applications Web d'entreprise sont basées sur les standards Java EE : Servlet, JSP, JSTL, TagLib et Bean. Au-delà de la maîtrise de ces standards, les concepteurs développeurs doivent être capables de proposer, pour ces applications, des architectures pérennes de type MVC 2 : en séparant bien les points de vue présentation et métier. Cette formation apporte les compétences et connaissances nécessaires au développement d'application Java Web. Mise en œuvre sous le serveur d'application Tomcat, le participant sera capable de déployer l'application sous tous types de serveurs (Weblogic d'Oracle, WebSphere Application Server d'IBM, ou pour l'open source GlassFish, JBoss, JOnAs...).

### Public visé

Cette formation s'adresse aux développeurs, concepteurs, chefs de projet ou architectes techniques.

### Objectifs

- Maîtriser le développement d'application Java Web à base de Servlets et JSP (Java Server Page)
- Aborder les principes d'administration des serveurs pour l'exécution des applications Web et faciliter la collaboration des équipes de développement, de design et d'opérations
- Répondre aux problématiques de localisation des applications
- Comprendre et mettre en œuvre le pattern d'architecture MVC2 (Modèle Vue Contrôleur)
- Développer des balises personnalisées (Taglibs) et les intégrer dans des JSPs
- Manipuler les bibliothèques de balises JSTL (Java Standard Tag Library)
- Internationaliser l'application
- Appréhender et comparer les Framework Web de référence : Struts et JSF (Java Server Faces)

### Prérequis

Pour suivre cette formation il faut avoir suivi les formations « Java - Maîtriser les fondamentaux de la programmation » et « HTML 5, CSS3, JavaScript : développement client » ou posséder les connaissances équivalentes.

### Contenu

#### L'architecture Serveur d'application

- Le modèle en couche (Présentation, Application, Données)
- Les serveurs d'applications Java EE : utilité, intérêts et fonctionnement
- La plate-forme Java EE et ses évolutions
- L'architecture Web (protocole et serveur http)

#### Les servlets, composants de la plateforme Java EE

- Les principes et le fonctionnement des servlets
- La structure de base des servlets
- L'exécution sous le serveur d'applications
- La synchronisation des servlets
- Les redirections d'URL : permanente et temporaire
- La gestion conversationnelle : cookies, sessions
- Le chaînage de servlets : inclusion, délégation

#### Déploiement d'applications

- Structure d'une application Web
- Fichiers d'archives
- Descripteurs de déploiement
- Les nouveautés de la plateforme Java EE, l'impact sur le développement et le déploiement

#### Conception

- Le design pattern MVC2 (Modèle Vue Contrôleur)
- Les composants Java MVC : le Modèle = Beans, EJB la Vue = JSP, le Contrôleur = Servlet
- Les Framework MVC2 : Struts, JSF....
- Les applications internet riches RIA (Rich Internet Application)

#### La technologie Java Server Page (JSP)

- Présentation et fonctionnement des JSPs
- Les JSPs et les servlets
- Interactions entre les composants

#### Le développement de JSPs

- Les Scripts (expression, scriptlets, déclaration)
- Les directives (page, include)
- Les actions (UseBean, Include, Forward)

#### Le langage EL

- Présentation du langage EL (Expression Language)
- Manipulation dans les JSP

#### **Les balises personnalisées ou Taglib**

- Concepts des balises personnalisées (Taglibs)
- Utilité et utilisation
- Développement de balises personnalisées : gestionnaire de balises, interfaces, descripteur de bibliothèque de balises
- Déploiement d'une bibliothèque de balises
- Balises simples avec attributs
- Balises présentant des variables de scripts
- Corps de balises (itération, filtrage de contenu, imbrication de balises)
- Internationalisation des applications

#### **Les balises standards : JSTL**

- Introduction à Java Standard Tag Library
- Les objectifs
- Principales bibliothèques et balises
- Manipulation et intégration des balises dans les JSP

## HTML5, CSS3, JavaScript – Développement Web

Durée : 5 jours

Réf du cours : GKHTML

---

### Résumé

Les applications Web d'entreprise sont basées sur des standards tels que HTML (Hyper Text Markup Language) et CSS (Cascading Style Sheet). Avec la normalisation fin 2014 de la norme HTML5, les possibilités offertes aux designers par CSS3 et JavaScript, le développement Web côté client est entré dans une nouvelle ère. Cette formation permet aux participants d'acquérir les compétences et connaissances nécessaires pour développer l'aspect graphique de toutes interfaces utilisateurs du monde du Web.

---

### Public visé

Développeurs, Concepteurs, Chefs de projet, Architectes techniques.

---

### Objectifs

- Etre capable de développer un site Web mettant en œuvre les concepts et les balises HTML5.
- Gérer l'aspect graphique du site à l'aide de CSS3.
- Développer avec JavaScript

---

### Prérequis

Notions Web de base, algorithmique. Avoir suivi la formation « Conception objet et notation UML » ou posséder les connaissances équivalentes.

---

### Contenu

#### Rappels des architectures Web

- Internet et ses protocoles.
- Les composants architecturaux d'une application Web: navigateur, serveur, ressource.
- Caractéristiques de http (hyper text transfer protocol).
- Du web statique au web dynamique.

#### De HTML à HTML5

- Les différentes évolutions de HTML (Hyper text markup language).
- Notion de Web sémantique.

#### Structure d'un document HTML

- Doctype.
- Sections, entêtes et pied
- Articles.

#### Les formulaires

- Nouveaux champs de saisie.
- Techniques de validation.

#### Le multimédia

- L'élément vidéo, l'élément audio,
- L'élément canvas.

#### CSS3

- Comment gérer la compatibilité des navigateurs.
- Couleurs et effets : transparence, dégradés, coins arrondis, ombres portées.
- Caractères : web fonts, ombrage.
- Transformations et animations : rotation, redimensionnement, déformation, transitions, animations.
- Mise en page : multicolonnage, mise en page flexible.

#### Le langage Javascript

- L'objectif de JavaScript.
- L'intégration aux pages HTML.

#### Éléments syntaxiques

- La structure générale d'un programme de scripts.
- Les données, les opérateurs, les instructions de contrôle.
- Les fonctions et l'appel de scripts.

#### Les objets

- Les différentes catégories d'objets du navigateur.
- Les objets Windows, Frame et Document.
- Les objets Link, Anchor et Location.
- Les objets spécifiques à JavaScript.

#### Le contrôle des formulaires

- Les objets Button, Text, Select, Option, Checkbox et Radio.



# Développement Web avec AngularJS

Durée : 3 jours

Réf du cours : GKANG

---

## Résumé

Ce cours vous permet d'acquérir les fondamentaux du Framework AngularJS

---

## Public visé

Développeurs Web

---

## Prérequis

Avoir des connaissances HTML et JavaScript.

---

## Contenu

### Présentation AngularJS

- Principe des SPA (Single Page Web App)
- Structure d'une première application AngularJS
- Rôles contrôleur & modèle
- Les scopes, arborescence et héritage

### Template, contrôleurs

- Notion de View Template, l'approche déclarative
- Présentation de quelques directives standards
- Organiser le code avec des modules
- Principe du data binding

### Créer un service

- Notion de service AngularJS
- Les différents types de service
- Présentation de quelques services prédéfinis
- Injection des dépendances
- Organiser le code entre contrôleurs et services

### Communiquer avec un serveur, invoquer un service REST

- Les techniques d'invocation
- Le service \$http (utilisation dans un contrôleur, puis dans un service)
- Invoquer un service REST avec \$resource

### Afficher et formater les données

- Les directives ng-bind, ng-bind-html
- Les directives conditionnelles ng-if, ng-show, ng-hide, ng-switch
- Afficher des collections avec ng-repeat
- Personnaliser l'affichage avec ng-class
- Les filtres standards AngularJS
- Traitement des formulaires, mécanismes de validation

### La navigation entre différentes pages

- Principe du routage dans une application SPA
- Le routage avec AngularJS
- Utiliser le service \$route pour le routage

## Développer des services Java – Architecture SOA, REST

Durée : 5 jours

Réf du cours : GKJAVSERV

### Résumé

Cette formation permet aux participants d'acquérir les compétences et connaissances nécessaires pour développer des applications Java distribuées d'une nouvelle génération basée sur Internet en collaboration avec les concepts XML. Des applications dispersées sur le réseau peuvent alors être intégrées comme fonction d'un système plus large. Basés principalement sur XML et des protocoles d'échanges, les services web sont naturellement très adaptés au langage Java et la plate-forme Java EE. Les participants pourront alors grâce à de nombreux ateliers pratiques, développer, publier, utiliser leurs services Java.

### Public visé

Développeurs, Concepteurs, Chefs de projet, Architectes techniques.

### Objectifs

- Découvrir XML et les domaines d'applications liés. Utiliser XML dans les applications.
- Générer et manipuler des données XML.
- Spécifier, Développer, déployer et administrer des services Web JAX-WS et SAAJ.
- Créer des applications clientes pour les services avec JAX-WS
- Créer et déployer des services à partir de WSDL.
- Développer des clients synchrones et asynchrones pour les services Web dans Java.
- Fournir des services Web de type RESTful pour de l'AJAX. Sécuriser de façon optimale les services Web.

### Prérequis

Avoir suivi la formation « Maîtriser les API avancées de la plateforme Java SE » ou posséder les connaissances équivalentes.

### Contenu

#### Présentation du XML

- Liens avec HTML et SGML
- Les domaines d'applications du XML
- Recommandation du W3C
- Apport du XML aux applications
- Outils XML : navigateurs, parsers, éditeurs

#### Mise en œuvre d'XML

- Gestion de document
- B2B
- Utilisation d'XML dans les applications
- Echange de données entre applications
- Génération de données XML
- Commerce électronique

#### Le langage XML

- Règles de Définition
- Création des balises
- Eléments du méta-langage
- Documents valides ou Documents formés
- Syntaxe et Structure

#### DTD - Schéma XML - XSD

- Définition, Namespace
- Règles de validation
- Règles d'écriture, contraintes, occurrences, ...
- Construction
- Les éléments, attributs, entités
- Déclaration dans un document XML

#### DOM et SAX

- Définition
- SAX : API de traitement des événements dans les documents
- DOM : Modèle Objet
- Manipulation des données XML Client/serveur

#### L'Architecture SOA

- Les 5 caractéristiques de SOA Service Oriented Architecture
- Les 7 briques d'une architecture SOA
- Relation entre SOA et Web Services
- Technologies principales des Web Services-: HTTP, XML, SOAP, WSDL, UDDI

## XML dans Java

- Syntaxe et espaces de nom XML
- Description de XML avec les schémas
- Conversion et « reconversion » des paramètres avec JAXB
- Personnalisation des liaisons XML-Java
- Manipulation XML avec les APIs SAX, DOM et Sax

## Messages SOAP

- Rôle de SOAP dans les services Web
- Structure des messages SOAP
- SOAP-ENV, Header, Body, Attachement
- Opération, messages et erreurs

## Interface avec WSDL

- Rôle de WSDL dans les Web Services
- 5 sections clés d'un document WSDL
- Description du contrat / interfaces / opérations dans section port Type
- Comprendre les sections 'abstraites': messages, types
- Comprendre les sections 'concrètes': building, services
- Spécification de l'implémentation du service (service)
- Spécification du protocole d'échange de messages (Baining)

## Solutions de gouvernance des web services

- Gouvernance: Respect de l'architecture SOA
- Conformité avec les normes WSI (Web Service Interopérabilité)
- Intégration de proxies et d'adaptateurs
- Mise en œuvre d'une terminaison de service Web avec JAX-WS
- Personnalisation de services Web JAX-WS
- Déploiement d'un fichier WAR de service Web
- Interception de trafic entre les services Web et les clients
- Optimisation de la transmission de messages

## Développement Web Services SOAP en Java

- Créer WS à partir de WSDL (WSDL-first) ou à partir du code (code-first)?
- Utiliser l'API de 'haut niveau' JAX-WS
- Utiliser l'API de 'bas niveau' SAAJ
- Valider la conformité de Web Service avec les normes WS-I
- WS-I: outils: Analyzer, Monitor
- Création d'intercepteurs (Handler) coté web service et coté client
- Optimisation de la transmission de messages avec MTOM

## Développement Web Services SOAP avec Framework /IDE

- Valeur ajoutée des Framework Web Services
- Panorama des Framework: CXF, AXIS, METRO, Boss WS
- Valider la conformité de son WS avec WS-I
- Création de livrable Web Services WAR
- Création livrable Web Services type EJB Session
- Création d'intercepteurs (Handler) coté web service et coté client
- IZARD pour créer et tester des Web Services dans Eclipse et NetBeans

## Clients de services Web

- Accès à des services Web via WSDL
- Création de fichiers sources clients à partir de WSDL
- Personnalisation des fichiers sources générés par le biais de JAX-WS
- Création de services et de clients de notification (one-wax)
- Écriture de clients en multithreading
- Interception et modification de messages SOAP

## Approche REST

- Développer de services Web RESTful avec JAX-WS
- Caractéristiques de POJO annotés, HTTP, XML / JSON
- Créer de l'interactivité du côté client
- Invoquer des services Web par le biais de l'API Dispatché
- Traitement des messages XML reçus
- Sécurité déléguée au transport
- Développement REST avec le Framework Jersey

## Comparaison SOAP / REST

- Interopérabilité
- Spécifications
- Sécurité
- Transport
- Gouvernance

## La persistance des données

Durée : 5 jours

Réf du cours : GKJPA5J

---

### Résumé

Cette formation permet d'acquérir les connaissances et compétences nécessaires pour maîtriser la mise en place d'un socle JPA (Java Persistence API) et des outils appropriés.

JPA apporte une couche d'abstraction normalisée pour les couches de persistance des applications Java EE.

Capitalisant sur la brique Hibernate, JPA généralise le modèle POJO de persistance des objets avec un ensemble d'interfaces et de factories indépendantes du provider sous-jacent selon le pattern Abstract Factory. On peut ainsi choisir un fournisseur Hibernate ou basculer sur d'autres providers sans impacter le code de la couche de persistance.

JPA normalise tous les besoins sur la couche de persistance : mappings simples ou complexes, langage JPQL pour les requêtes objets.

---

### Public visé

Cette formation s'adresse aux Développeurs Java avec une expérience Projets.

---

### Objectifs

- Maîtriser la mise en place d'un socle JPA et des outils appropriés
- Maîtriser les différentes stratégies de mappings : relations, collections et héritage
- Manipuler les données avec le langage de requêtage JPQL
- Comprendre le mécanisme transactionnel et inscrire la persistance JPA dans une transaction distribuée

---

### Prérequis

Avoir une bonne maîtrise de Java CORE, JDBC et SQL.

---

### Contenu

#### Architecture Technique

- Architecture logicielle
- Architecture Technique JPA
- Mise en place de l'environnement

#### Fondamentaux du Mapping des Objets Persistants

- Notion de classe persistante
- Mappings sur des relations unidirectionnelles
- Mappings sur des relations bidirectionnelles

#### Mapping des objets du domaine

- Mappings de relations d'agrégation
- Mappings d'une relation d'héritage
- Mapping des Collections et Types Objets Complexes

#### Effectuer ses requêtes avec JPQL

- Le langage JPQL
- Associations et jointures
- Préparation des requêtes

#### Gestion des transactions

- Introduction aux différents types de transactions
- Approfondir le mécanisme des transactions distribuées

#### La persistance des données avec Hibernate

## Spring : Construire des socles applicatifs intégrés

Durée : 4 jours

Réf du cours : GKJSPG

### Résumé

Cette formation permet aux participants d'acquérir les compétences et connaissances nécessaires pour concevoir des architectures applicatives modulaires et faiblement couplées.

Le principe d'inversion de contrôle IOC ou injection de dépendances permet ainsi de réaliser un couplage faible entre les modules applicatifs : composants, couches logicielles, Services Métiers et DAO CRUD. Sous l'angle technique, Spring permet de fédérer les Framework en un seul socle homogène : il permet ainsi une intégration avec Struts, Hibernate, JMS et les Web Services en une seule architecture technique.

Spring s'impose ainsi aujourd'hui comme un élément central des architectures techniques J2EE et comme un ingrédient essentiel pour structurer les architectures applicatives en modules et services réutilisables.

### Public visé

Cette formation s'adresse aux Développeurs d'Applications J2EE.

### Objectifs

- Comprendre le mécanisme d'injection de dépendances pour faciliter le couplage faible entre composants et couches logicielles d'une Application
- Utiliser Spring pour faciliter l'accès aux Services et Ressources JNDI
- Intégrer Spring avec les Framework de référence
- Struts et Hibernate
- Aborder le module MVC de Spring
- Comprendre les transactions avec Spring
- Comprendre les connecteurs EAI Web Services, Remoting et JMS

### Prérequis

Avoir une bonne maîtrise du développement Web J2EE. Des notions sur Struts et Hibernate sont un plus.

### Contenu

#### Introduction

- Les différents modules constituant le Framework
- Contextes Spring et cycle de vie des beans

#### IOC : l'injection des dépendances

- Présentation de l'application de référence
- Présentation de l'IOC et application aux couplages faibles entre couches logicielles
- Liaison de beans et injection des services
- Paramètres, Initialisation et Destruction

#### La gestion des pools et des transactions

- Data Sources et accès au pool de connexions via JNDI
- Transactions applicatives et déclaratives : démarcation et propriétés ACID

#### Intégration avec le Framework Struts

- Accès au contexte par héritage de Action Support
- Accès aux actions par délégation aux proxys Spring

#### Intégration avec le Framework Hibernate

- Prise en charge de la configuration d'Hibernate
- Support Hibernate dans les DAO avec le Helper HibernateTemplate

#### Spring MVC

- Architecture générale: contrôleurs, Dispatchers et Handlers
- Processus de traitement d'une requête: handler Mappings et Interceptors
- Implémentation des vues sous format PDF, HTML ou Excel
- Intégration handler / beans dans Spring
- Data Binding et Validation dans Spring

#### Spring Notions Avancées

- Les connecteurs Web Services et Remoting
- Les connecteurs JMS
- Les aspects AOP

