

Formation

Architecture Full Stack – Concevoir des applications modernes

Formation disponible en présentiel et en ligne à la demande. Apprenez à concevoir et piloter une architecture Full Stack moderne couvrant le front-end, le back-end, les API, les données, la sécurité, le cloud et le déploiement continu.

Formation 100% pratique

Exercices & cas d'usage

Formateurs expérimentés

4 jours

24-26 h

Format

Ateliers &
labs guidés

Modalité

Présentiel
ou en ligne

Niveau

Initiation

INSCRIPTION / RÉSERVATION



Je m'inscris
maintenant



OBJECTIFS PÉDAGOGIQUES

- Comprendre le rôle et les principes de l'architecture Full Stack dans un SI moderne.
- Concevoir une architecture front-end et back-end cohérente, évolutive et maintenable.
- Structurer les échanges grâce à des APIs robustes (REST, événements, messages).
- Intégrer les enjeux de données, sécurité, performance, déploiement et observabilité.
- Proposer une architecture cible et une trajectoire d'évolution réaliste.

PUBLIC CIBLE

- Développeurs Full Stack ou back-end souhaitant renforcer leurs compétences en architecture.
- Leads techniques, Tech Leads et Scrum Masters à dominante technique.
- Architectes débutants ou récemment en poste.
- Chefs de projet IT souhaitant améliorer leur communication avec les équipes techniques..

PRÉREQUIS

- Expérience pratique en développement web (front-end et/ou back-end).
- Pratique Windows 10/11.
- Bonne compréhension des bases de données (SQL et/ou NoSQL).
- Notions de base sur les concepts du web (HTTP, APIs, JSON, etc.).



PROGRAMME DE LA FORMATION – DÉTAILLÉ

Introduction à l'architecture Full Stack

Vision globale & enjeux

- Définir l'architecture Full Stack : composants, responsabilités, frontières.
- Monolithes, microservices, micro-frontends, architectures modulaires.
- Enjeux business : time-to-market, scalabilité, maintenabilité, coût.
- Rôles de l'architecte, du Tech Lead et des équipes produit.

Principes d'architecture logicielle

- Responsabilités, séparation des couches, dépendances maîtrisées.
- Principes SOLID, DRY, KISS, YAGNI.
- Patrons d'architecture : couche de présentation, services, domaine, données.

Architecture front-end moderne

Structurer le front-end

- SPA, SSR, CSR, ISR : choix d'architecture et impacts.
- Architecture par composants, design system, réutilisabilité.
- Gestion de l'état (local, global, serveur) et performance.

Communication avec le back-end

- Consommation d'APIs REST : versioning, gestion des erreurs, pagination.
- Authentification, autorisation, gestion des tokens côté front.
- Gestion du cache, optimisations réseau (lazy loading, bundling, CDN).

Qualité & UX

- Accessibilité, responsivité, performance perçue.
- Tests unitaires et end-to-end sur le front-end (concepts).

Architecture back-end & APIs

Structurer le back-end

- Architecture en couches : API, services, domaine, infrastructure.
- Introduction à l'architecture hexagonale / Clean Architecture.
- Gestion des dépendances, inversion de contrôle, injection de dépendances.



PROGRAMME DE LA FORMATION – DÉTAILLÉ

Conception d'APIs

- Principes REST : ressources, verbes, statuts HTTP, HATEOAS (survol).
- Spécification d'API : OpenAPI / Swagger, contrat d'interface, documentation.
- Gestion des erreurs, validation des données, idempotence.

Monolithes modulaires & microservices

- Quand choisir un monolithe ? Quand envisager les microservices ?
- Découpage fonctionnel, Bounded Contexts (introduction DDD).
- Communication synchrone (HTTP) vs asynchrone (file de messages, events).

Données, persistance & intégration

Conception des données

- Modélisation logique et physique, normalisation / dénormalisation.
- Gestion des identifiants, intégrité référentielle, contraintes.

SQL, NoSQL & choix d'architecture de données

- Bases relationnelles vs documentaires, clé-valeur, colonnes, graphes (survol).
- Critères de choix : cohérence, volume, performance, flexibilité.
- Patrons : CQRS (survol), Event Sourcing (notions).

Intégration & interopérabilité

- Intégration via messages, événements, ETL/ELT, webhooks.
- Gestion des flux entre systèmes internes et externes (SI, partenaires, tiers).

Sécurité, performance & observabilité

Sécurité d'une architecture Full Stack

- Principes de base : défense en profondeur, Zero Trust, principe du moindre privilège.
- Gestion des identités, authentification (OAuth2/OIDC, SSO), autorisation (RBAC).
- Protection des données en transit et au repos, secrets management.

Performance & résilience

- Mise en cache (front, back, base), CDN, optimisation des requêtes.
- Timeouts, retries, circuit breaker, backpressure (notions).
- Tests de charge, profilage, gestion des goulets d'étranglement.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Observabilité & supervision

- Logs structurés, métriques, traces distribuées.
- Tableaux de bord, alertes, indicateurs clés (SLA, SLO, SLI).
- Patterns cloud : autoscaling, blue/green, canary releases.

Architecture cloud, déploiement & industrialisation

Architecture Full Stack dans le cloud

- Déploiement sur les principaux clouds (Azure, AWS, GCP – concepts).
- Conteneurs & orchestrateurs (Docker, Kubernetes – survol architectural).

CI/CD & pipelines de déploiement

- Chaîne de valeur : build, tests, packaging, déploiement, rollback.
- Intégration des tests automatiques et de l'analyse de code dans le pipeline.
- Gestion des environnements (dev, test, pré-prod, prod).

Étude de cas : définir une architecture cible

- Mise en situation : à partir d'un besoin métier, proposer une architecture Full Stack.
- Choix technologiques, urbanisation, schémas d'architecture.
- Plan de migration ou de mise en œuvre par étapes.

