

Mis à jour le 21/08/2026

S'inscrire

Formation Optimisation des applications .NET avec C#

3 jours (21 heures)

Présentation

Optimiser une application .NET avec C# consiste à maîtriser les performances CPU, la mémoire, les accès aux données et la scalabilité pour offrir une expérience fiable aux utilisateurs.

Notre formation Optimisation des applications .NET avec C# vous permettra d'identifier les causes réelles de lenteur et de mettre en oeuvre une démarche structurée de mesure, d'analyse et d'amélioration. Vous apprendrez à distinguer les problèmes liés au code, aux entrées sorties, aux requêtes de données, à la concurrence et aux ressources système.

Vous utiliserez les outils de profilage et de diagnostic de l'écosystème .NET pour analyser la consommation processeur, les allocations mémoire, les appels coûteux, les blocages et les temps de réponse. La formation aborde également les mécanismes du Garbage Collector, les tâches asynchrones et les bonnes pratiques de gestion des collections et des chaînes de caractères.

Vous apprendrez à optimiser les accès aux bases de données avec Entity Framework Core, à réduire les échanges inutiles, à appliquer des stratégies de cache adaptées et à améliorer les performances des API ASP.NET Core. Les choix de configuration, de compression, de limitation de débit et de traitement des requêtes seront étudiés selon les contraintes de production.

À l'issue de la formation, vous serez en mesure de définir des indicateurs pertinents, de reproduire un incident de performance, de proposer des corrections mesurables et de valider leurs effets par des tests de charge. Vous disposerez d'une méthode pragmatique pour fiabiliser les applications .NET en environnement réel.

Comme toutes nos formations, celle-ci vous présentera **la dernière version stable** de la technologie et ses nouveautés.

Objectifs

- Mettre en place une démarche de mesure et d'analyse des performances d'une application .NET
- Identifier les goulots d'étranglement liés au CPU, à la mémoire, aux entrées sorties et à la concurrence
- Utiliser les outils de profilage et de diagnostic de l'écosystème .NET
- Optimiser le code C#, les accès aux données et les traitements asynchrones
- Améliorer les performances et la scalabilité des applications ASP.NET Core
- Valider les optimisations par des indicateurs, des tests de charge et une analyse comparative

Public visé

- Développeur .NET
- Développeur C#
- Développeur ASP.NET Core
- Architecte logiciel
- Tech Lead .NET

Pré-requis

- Maîtrise de C# et de la programmation orientée objet
- Expérience du développement d'applications avec .NET
- Connaissances des mécanismes async, await et LINQ
- Notions d'accès aux données avec SQL ou Entity Framework Core

Pré-requis techniques

- Poste sous Windows 10, Windows 11, macOS ou Linux
- .NET SDK 8.0 ou version ultérieure installé
- Visual Studio 2022 ou Visual Studio Code avec les extensions C#
- Ordinateur doté de 16 Go de RAM recommandés et d'un processeur multicore
- Connexion Internet stable autorisant l'installation des outils et packages NuGet

Programme de notre formation Optimisation des applications .NET avec C#

[Jour 1 - Matin]

Mesurer avant d'optimiser

- Les enjeux de performance, de latence, de débit et de consommation de ressources

- La démarche d'optimisation : établir une référence, formuler une hypothèse, mesurer et valider
- Les indicateurs essentiels : temps de réponse, CPU, mémoire, allocations, exceptions et activité des threads
- Les pièges des mesures non reproductibles et l'influence des environnements de développement et de production
- Présentation des outils Visual Studio Profiler, dotnet-counters et dotnet-trace
- Atelier pratique : établir le profil de référence d'une application .NET présentant des lenteurs contrôlées

[Jour 1 - Après-midi]

Optimiser le code C# et le CPU

- Identifier les points chauds à partir des appels, piles d'exécution et échantillons processeur
- Réduire le coût des boucles, conditions, conversions, concaténations et traitements répétitifs
- Choisir les collections adaptées et exploiter efficacement Span<T>, Memory<T> et les structures de données
- Maîtriser les compromis entre lisibilité, allocations, copies de données et micro-optimisations
- Éviter les blocages liés à Task.Result, Wait et aux appels synchrones coûteux
- Atelier pratique : analyser un point chaud CPU et refactorer le code C# pour réduire son temps d'exécution

[Jour 2 - Matin]

Maîtriser la mémoire et le Garbage Collector

- Comprendre le tas managé, les générations, le Garbage Collector et les objets volumineux
- Repérer les allocations excessives, la rétention mémoire et les symptômes de pression mémoire
- Comparer les modes Workstation GC et Server GC selon le type de charge applicative
- Limiter les allocations avec la réutilisation d'objets, les pools, les buffers et les API adaptées
- Diagnostiquer les fuites logiques, les abonnements persistants et les objets conservés inutilement
- Atelier pratique : investiguer une surconsommation mémoire puis réduire les allocations d'un traitement .NET

[Jour 2 - Après-midi]

Accélérer les entrées sorties et l'accès aux données

- Identifier le coût des entrées sorties, des accès réseau et des appels à des services distants
- Structurer les traitements asynchrones avec async, await, annulation et gestion des délais
- Optimiser les requêtes Entity Framework Core : projection, filtrage, pagination et requêtes sans suivi
- Réduire les allers retours, prévenir les chargements excessifs et maîtriser les requêtes générées
- Mettre en place des stratégies de cache mémoire et de cache distribué selon la cohérence attendue

- Atelier pratique : optimiser un endpoint accédant à une base de données et comparer les métriques avant et après correction

[Jour 3 - Matin]

Améliorer les performances des API ASP.NET Core

- Analyser le cycle de traitement d'une requête ASP.NET Core et l'ordre des middlewares
- Optimiser la sérialisation, la taille des réponses, la pagination et le traitement des grands volumes
- Différencier response caching, output caching et cache applicatif
- Configurer la compression des réponses et évaluer ses compromis de performance et de sécurité
- Protéger les ressources avec la limitation de débit, les délais d'exécution et la régulation de charge
- Atelier pratique : améliorer les performances d'une API ASP.NET Core avec cache, pagination et compression

[Jour 3 - Après-midi]

Diagnostiquer et valider en conditions réalistes

- Construire un scénario de test de charge représentatif des usages et des volumes réels
- Interpréter les métriques de débit, latence, saturation CPU, files d'attente et erreurs applicatives
- Collecter et exploiter des traces avec dotnet-trace et des compteurs avec dotnet-counters
- Mettre en place une instrumentation utile : métriques métier, journaux structurés, traces et alertes
- Établir un plan d'optimisation priorisé et sécuriser les régressions par des tests automatisés
- Atelier pratique : réaliser un diagnostic complet sous charge, appliquer des corrections et présenter les gains mesurés

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format

numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.