

ADO.NET Entity Framework - Développer une solution d'accès aux données avec Visual Studio

Référence : ADON

Niveau : Intermediaire

Durée : 5 jours (35h.)

Tarif: Nous Contacter

Date: Avril, Mai, Juin

Contact: +225 22469017 / 74622582

Objectifs

majorité des applications d'entreprises actuelles doit pouvoir accéder à un volume important de données de nature hétérogène. ADO.NET Entity Framework permet aux programmes de ne pas être liés à un schéma de stockage particulier (relationnel par exemple). Concrètement, il suffit pour le développeur de définir un modèle conceptuel centré sur l'application ainsi qu'un mappage pour faire le lien entre ce modèle et le schéma spécifique du stockage. En cas d'évolution (par exemple format de stockage XML au lieu de relationnel), il n'y aura pas à intervenir sur le code de l'application mais uniquement au niveau du mappage. Autrement dit, le développeur s'éloigne des contraintes de stockage et rend son application plus facile à lire, à écrire et à maintenir en manipulant des objets proches de son application et non des « objets de stockage ». L'aspect relationnel, et dans son sillage le SQL, n'étant donc plus « obligatoirement » présents, d'autres concepts seront utilisés pour accéder aux données comme LINQ, WCF, Web Services, etc. Cette formation ADO.NET Entity Framework vous apprendra à définir une architecture complète d'accès aux données dans le respect des bonnes pratiques professionnelles et des performances. Concrètement, ce cours vous permettra de :

- Savoir utiliser les datasets proposés par ADO.NET
- Comprendre la philosophie et l'architecture de ADO.NET Entity Framework
- Apprendre à exécuter des requêtes via un objet de commande
- Construire des modèles de données d'entité (Entity Data Model)
- Manipuler les données à travers ces modèles (création, mise à jour, suppression)
- Gérer la concurrence et les transactions dans un contexte multi-utilisateurs
- Récupérer les données à travers des services Web (REST) ou WCF
- Être capable d'écrire des requêtes LINQ (Language Integrated Query)
- Comprendre les nouveautés et la philosophie de Entity Framework Core ainsi que les différences et points communs avec EF67

Public

Cette formation s'adresse aux développeurs et chefs de projets

Pré-requis

Pour suivre ce cours dans de bonnes conditions, il vous faut avoir certaines connaissances sur le langage C#. Ces connaissances s'acquièrent en suivant la formation « C#7, Développer en .NET 4.7 avec Visual Studio 2017 » (SLVS7).

Contenu du cours

Architecture et technologies d'accès aux données

Les technologies d'accès aux données (API diverses, DataSet, Entity)
Avantages et inconvénients des différentes architectures
Problématique des transactions pour le développeur
Performances et architectures
Limites de ADO.NET et apports de ADO.NET Entity Framework
Présentation de Entity Framework Core (.NET Core)

Le modèle historique ADO.NET

Récupération et modification de données en utilisant les commandes ADO.NET
Récupération et modification de données en utilisant les DataSets
Gestion de transactions et gestion de la concurrence dans des scénarios multi-utilisateurs

Travaux Pratiques :

Exploitation des données à partir d'une source ADO.NET

LINQ et ADO.NET

Philosophie et champs d'application de LINQ
Interrogation riche et optimisée du dataset avec LINQ to DataSet
LINQ to Objects
Interroger un Entity Data Model avec LINQ to Entities
LINQ to XML

Travaux Pratiques :

Requêtage sur un fichier XML avec LINQ to XML

Entity Data Models (conception)

Ne pas se soucier du format des sources (SGBDR, XML, tableur, texte, etc.)
Savoir concevoir un modèle de données d'entité
Le mapping entre les données et le modèle (fichier .edmx)
Les outils de Visual Studio 2017 (Entity Designer, Wizard, etc.)
Synthèse sur les différentes approches de conception (DataBase First, Model First, Code First)

Travaux Pratiques :

Utilisation des outils de Visual Studio 2017 pour concevoir et générer une base de données à partir d'un modèle
Génération d'une base de données à partir de classes (Code First)

Entity Data Models (interrogation)

Récupérer des données en utilisant LINQ to Entities
Récupérer des données en utilisant Entity SQL
Récupérer des données en utilisant le fournisseur EntityClient
Récupérer des données en utilisant des procédures stockées
Test unitaire sur le code d'accès aux données

Travaux Pratiques :

Connexion à différentes sources de données à l'aide des modèles de données d'entité

Création, mise à jour et suppression de données d'entité

Comprendre le suivi des modifications dans Entity Framework
Modifier des données dans un modèle de données d'entité

Travaux Pratiques :

Mettre à jour des données à partir d'un modèle de données d'entité

Gestion de scénarios multi-utilisateurs en utilisant les services d'objet

Gestion de la concurrence dans Entity Framework
Support des transactions dans Entity Framework

Travaux Pratiques :

Implémentation de la gestion des transactions à l'aide d'Entity Framework

Construire des solutions optimisées en utilisant les services d'objet

Les étapes d'exécution de requête
Suivi des modifications et matérialisation d'objet
Utilisation de requêtes compilées
Utilisation des vues Entity Framework générées à la conception
Surveillance des performances
Exécuter des modifications de données asynchrones

Travaux Pratiques :

Mise en place des optimisations à l'aide d'Entity Framework

Exemple de supervision des performances

Personnalisation d'entités et construction de classes d'entité personnalisées

Surcharge des classes générées
Utilisation de modèles pour personnaliser des entités
Création et utilisation de classes d'entité personnalisées

Travaux Pratiques :

Création et utilisation de classes d'entité personnalisées

Utilisation des classes POCO ("plain old" CLR Objects) avec Entity Framework

Conditions pour l'utilisation des classes POOCO
Classes POOCO et chargement tardif
Classes POOCO et suivi des modifications
Extensions des types entité

Travaux Pratiques :

Création et utilisation des classes POOCO et analyse des impacts

Construire une solution n-tiers en utilisant Entity Framework

Conception d'une solution n-tiers
Définition d'opérations et implémentation des structures de transport de données
Protéger les données et les opérations

Gestion des mises à jour dans une solution n-tiers en utilisant Entity Framework

Suivi des entités et persistance des modifications
Gestion d'exceptions dans une solution n-tiers

Travaux Pratiques :

Création d'une application utilisant une architecture n-tiers et implémentation de la couche d'accès aux données

Construire des solutions occasionnellement connectées

Mise en cache de données hors-ligne en utilisant XML

Services de données WCF

Introduction aux services de données WCF
Création d'un service de données WCF
Consommation d'un service de données WCF
Création, mise à jour et suppression de données dans un service de données WCF
Utilisation des services de données WCF avec des données non relationnelles
Protection des données et des opérations dans un service de données WCF
Prévenir les mises à jour non autorisées et améliorer les performances

Travaux Pratiques :

*Création d'un service de données WCF permettant de récupérer toutes les publications d'un auteur.
Récupération des données dans notre application.*

Web Service et données

Qu'est-ce qu'une application ASP.NET ?
Principes des Web Services
Publication d'un Web Service
Exemples de Web Services avec C# 6/7, ASP.NET, et IIS

Travaux Pratiques :

Création de Web Service sous Visual Studio 2017, utilisation pour alimenter une application WinForms

Envoi de mails par notre application via un Web Service

Présentation de Entity Framework Core

Philosophie d'EF Core

Que change la dimension Open Source concrètement pour le développeur ?

Classes et concepts communs entre Entity Framework 6 (EF6) et Entity Framework Core

Le cœur et les services optionnels

Les performances par rapport à EF6

Gestion du stockage des providers de logs simplifiée

Tests et provider in memory

La nouvelle DOTNET CLI (base, DbContext, migration)

L'avenir de EF Core