Entity Framework 6.1, Part 6 of 6: Code-First Dev

Meet the expert: Don Kiely is a featured instructor on many of our SQL Server and Visual Studio courses. He is a nationally recognized author, instructor, and consultant specializing in Microsoft technologies. Don has many years of teaching experience, is the author or co-author of several programming books, and has spoken at many industry conferences and user groups. In addition, Don is a consultant for a variety of companies that develop distributed applications for public and private organizations.

Prerequisites: This course assumes that you are experienced with relational database design and programming concepts. Entity Framework is all about data access, and the course doesn't spend any time on data fundamentals.
You'll also need a good understanding of object-oriented programming in C#, so that you know how to use the entity data objects that Entity Framework generates from your data model. You'll also need to have a working knowledge of Language INtegrated Queries (LINQ) in C#.
The course uses Entity Framework with various SQL Server databases, so you should know enough about SQL Server to know how to perform various tasks and be able to connect to a database. But you don't need deep knowledge of SQL Server to use Entity Framework productively. You should have already viewed the Entity Framework 6.1: SQL Server Features course before taking this course.

Runtime: 03:23:04

Course description: This course will start by covering how code-first works by default, which will probably work for most applications early in their development cycle. But when you're ready to deploy the application to a production server, or need more flexibility in how Entity Framework creates a database. You'll see how to create a code-first model and create a database from it, and see an application that makes use of it to maintain data in the database. You'll also learn how to customize the database using both data annotations and the DBModelBuilder API, which lets you write code instead of use data annotations. Last you'll see how code migrations is a newer feature of code first that goes beyond just deleting and recreating the database when the model changes.

Course outline:

Creating A Model
- Introduction
- Domain-Centric Modeling
- Using Code-First
- Creating and Using a Model
- Demo: Ordering Model
- Demo: Address & Order
- Demo: Navigation Properties
- Demo: Ordering Context
- Demo: Connection
- Summary

Model in ASP.NET MVC App
- Introduction
- Demo: Scaffold
- Demo: Add View
- Demo: Create
- Demo: Add Customer
- Summary

Run App and Explore Data
- Introduction
- Demo: Run App
- Demo: Create Database
- Demo: Explore Data
- Summary

Update and Delete Customers
- Introduction
- Demo: Details
- Demo: Update Customer
- Demo: Add View
- Demo: Delete User
- Demo: Run Application
- Summary

Changing the Model
- Introduction
- Demo: Change the Model
- Changing the Model
- Demo: Database Re-creation
- Summary

Seeding the Database with Data
- Introduction
- Demo: Ordering Initializer
- Demo: Ordering Initializer Cont
- Demo: Change the Model
- Summary

Customizing the Database
- Introduction
- Customizing the Database
- Conventions
- Data Annotations
- Demo: Primary Keys
- Demo: Annotations
- Demo: Override Names
- Demo: Using Annotations
- Summary

Db Model Builder API
- Introduction
- Demo: Remove Convention
- Config Properties & Entities
- Demo: Property Method
- Demo: Change Column Name
- Summary

Code First Migrations
- Introduction
- Code-First Migrations
- Migration Implementation
- Demo: Package Manager
- Demo: Migration Object
- Demo: Automatic Migration
- Demo: Change Initialization
- Demo: Database Schema
- Demo: Data Loss
- Demo: Accept Data Loss
- Demo: Add Model Class
- Summary