

Creating a Data Model and Database with EF Core



Julie Lerman

MOST TRUSTED AUTHORITY ON ENTITY FRAMEWORK

@julielerman thedatafarm.com



Overview



Preparing the .NET Framework solution

Adding in Entity Framework Core

Building a data model from your classes

Understanding mappings and relationships

Creating a database using migrations

Highlight similarities, differences from EF6

Re-creating the solution in .NET Core



My Development Environment

Windows 10

Visual Studio
2015
Update 3

SQL Server
2016
LocalDb

.NET Core 1.1

Everything updated to latest releases



Setting up the Solution





Seven Samurai

Akira Kurosawa | 1954

By Copyright © 1954 by en:Toho Studios. [Public domain], via Wikimedia Commons



Adding in the Domain Classes





Tokugawa Ieyasu with help from the Jodo monks of the Daijuji temple in Okizaki, defeats the Ikkō-ikki at the battle of Azukizaka, 1564
https://commons.wikimedia.org/wiki/File:Azukizaka_1564.JPG

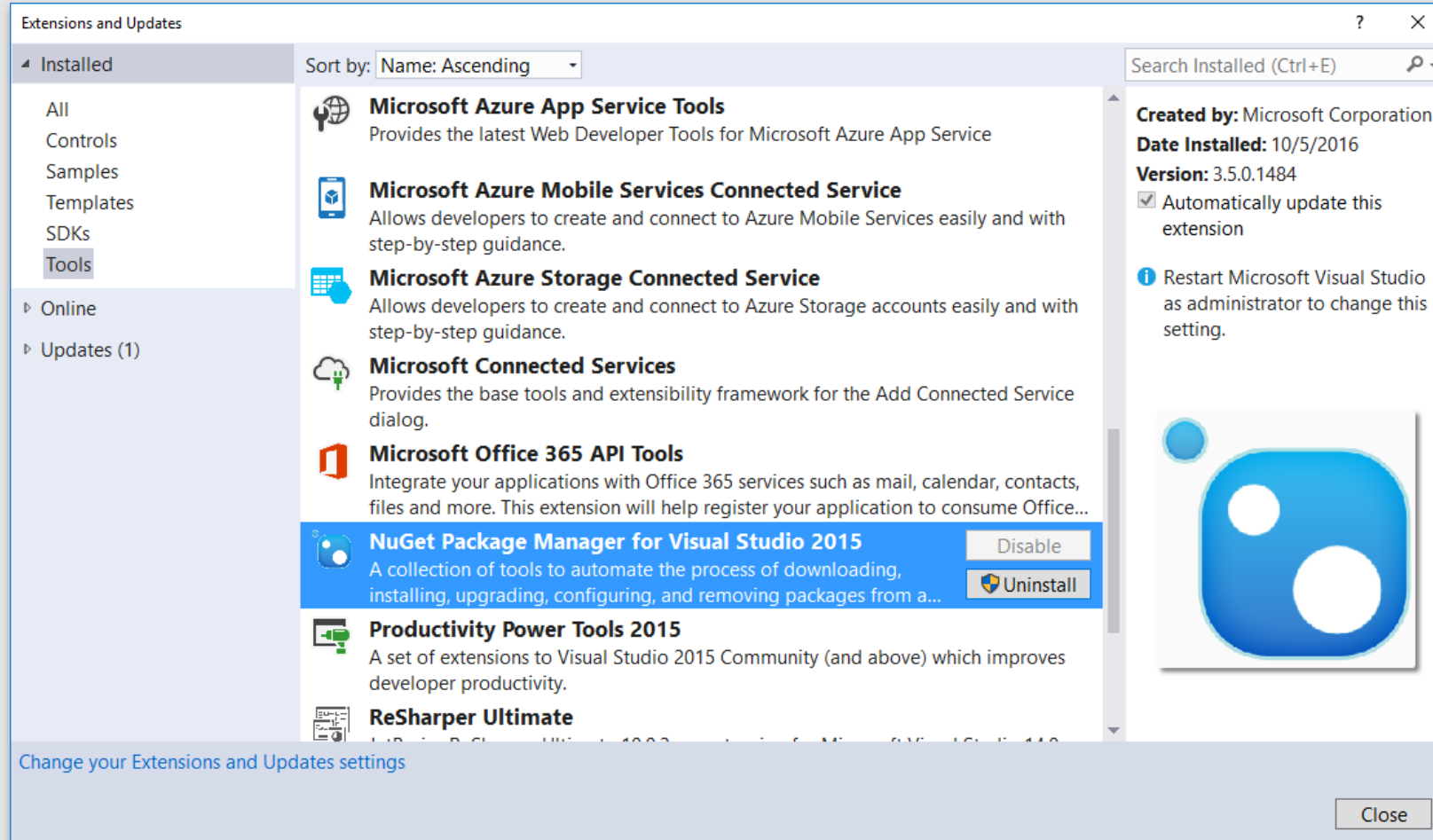


Adding EF Core with the NuGet Package Manager



NuGet Package Manager

[Visual Studio Extension]



Adding EF Core via PowerShell Commands



Creating the Data Model



DbContext Definition Affects ...

How you interact
with EF in code

How EF handles
relationships

How EF infers the
database schema



Specifying the Data Provider and Connection String



EF 4.1 - 6 Code First Magic

**Infer database provider
if missing**

**Infer connection string
if missing**

Create database if missing

(CreateDatabaseIfNotExists IDatabaseInitializer)

Other IDatabaseInitializers

DropCreateDatabaseIfModelChanges
DropCreateDatabaseAlways
MigrateDatabaseToLatestVersion



No More Database Magic



Specify Database Provider and Connection

Configure in DbContext

**Configure in
ASP.NET Core Startup.cs**

Details can be stored in .NET app/web.config



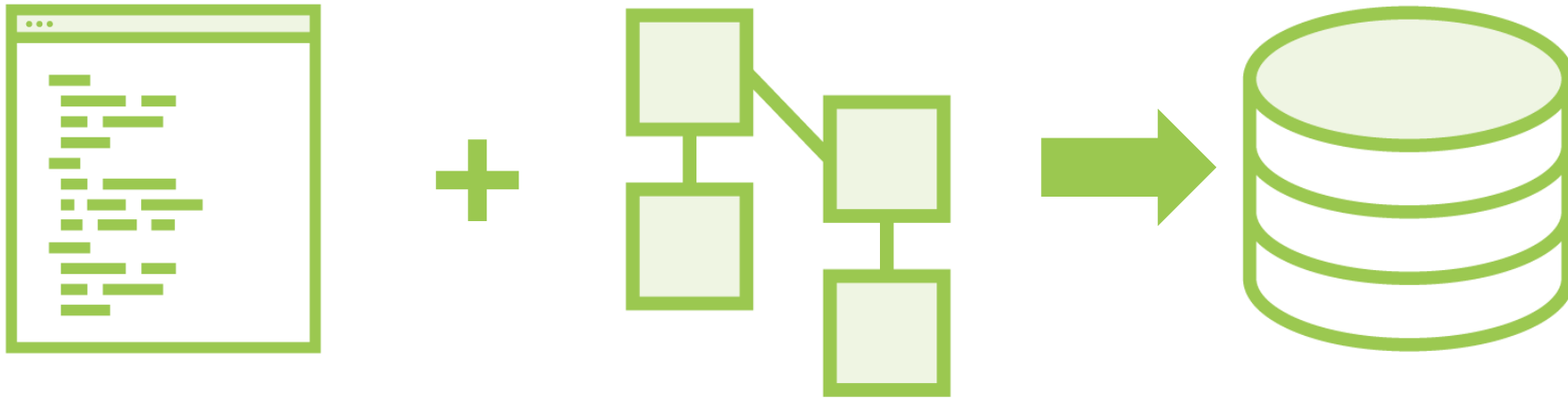
Understanding EF Core Migrations



EF Core Basic Migrations Workflow



Mapping Your Data Model to the Database



Adding Your First Migration



Migrations are now
source-control friendly!



Scripting Database Creation Script from the Migration



EF6 Migrations

Available Commands

Cmdlet	Description
Enable-Migrations	Enables Code First Migrations in a project.
Add-Migration	Scaffolds a migration script for any pending model changes.
Update-Database	Applies any pending migrations to the database.
Get-Migrations	Displays the migrations that have been applied to the target database.

Update-Database

SYNTAX

```
Update-Database [-SourceMigration <String>] [-TargetMigration <String>] [-Script] [-Force]
[-ProjectName <String>] [-StartupProjectName <String>]
[-ConfigurationTypeName <String>] [-ConnectionStringName <String>] [-AppDomainBaseDirectory
<String>] [<CommonParameters>]

Update-Database [-SourceMigration <String>] [-TargetMigration <String>] [-Script] [-Force]
[-ProjectName <String>] [-StartupProjectName <String>]
[-ConfigurationTypeName <String>] -ConnectionString <String> -ConnectionProviderName
<String> [-AppDomainBaseDirectory <String>] [<CommonParameters>]
```



Script Migration Has Changed

EF4.3 - EF6

Update-Database -script

Scripts the latest migration (default)

EF Core

Script-Migration

Scripts all migrations (default)

Script-Migration -Idempotent

Scripts all migrations with If/Then logic

Script-Migration -From [migration] -To [migration]

Controls which range of migrations are scripted



Script-Migration vs. Update-Database

Script-Migration

More control
Production databases
Apply needed tweaks

Update-Database

Best for local development database



Creating the Database Directly from the Migration



Preparing Your System for .NET Core



Recreating the Model in .NET Core



.NET Core 1.1
and it's current SDK

Visual Studio Tools



Visual Studio 2017

has updated tools for .NET Core



.NET Core Projects are different in **Visual Studio 2017**



Microsoft.EntityFrameworkCore.Tools 1.1.0
(the PowerShell tools)


**is not compatible with
Visual Studio 2015**



With 1.0.0 tools,
dotnet ef will now work in
this scenario,
but sub-commands still
need the additional info
explained here



CLI dotnet Commands Are Not PowerShell Commands

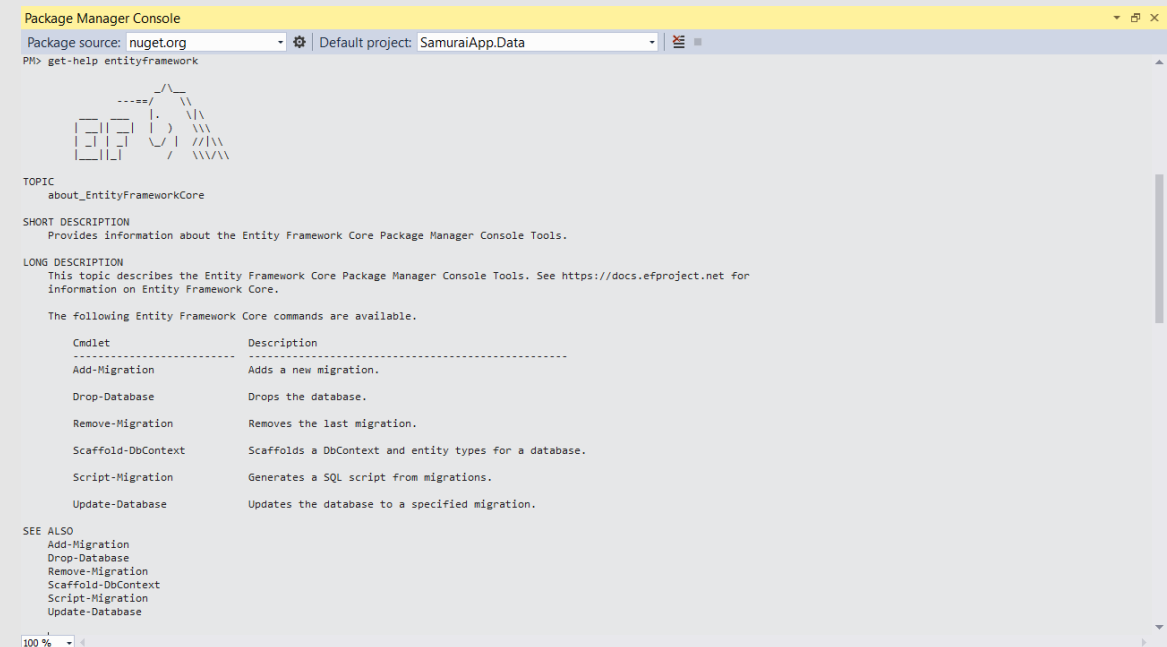


```
Entity Framework Core Console Commands 1.1.0-preview4-22752
Usage: ef.dll [options] [command]


Options:
  -h|--help                Show help information
  --version                Show version information
  --verbose                Show verbose output
  --no-color                Do not use color in console output
  --assembly <assembly>   The assembly file to load.
  --startup-assembly <assembly> The assembly file containing the startup class.
  --data-dir <dir>         The folder used as the data directory (defaults to current directory)
  --project-dir <dir>      The folder used as the project directory (defaults to current directory)
  --content-root-path <dir> The folder used as the content root path for the application
  --root-namespace <namespace> The root namespace of the target project (defaults to the project namespace)
  -e|--environment <environment> The environment to use. If omitted, "Development" is used.

Commands:
  database    Commands to manage your database
  dbcontext   Commands to manage your DbContext types
  migrations  Commands to manage your migrations

Use "ef.dll [command] --help" for more information about a command.
```




EF Core CLI vs PowerShell Commands



```
Entity Framework Core Console Commands 1.1.0-preview4-22752
Usage: ef.dll [options] [command]
Options:
  -h|--help                Show help information
  --version                Show version information
  --verbose                Show verbose output
  --no-color                Do not use color in console output
  --assembly <assembly>   The assembly file to load.
  --startup-assembly <assembly> The assembly file containing the startup class.
  --data-dir <dir>         The folder used as the data directory (defaults to curr
  --project-dir <dir>      The folder used as the project directory (defaults to c
  --content-root-path <dir> The folder used as the content root path for the applic
  --root-namespace <namespace> The root namespace of the target project (defaults to t
  -e|--environment <environment> The environment to use. If omitted, "Development" is us
Commands:
  database    Commands to manage your database
  dbcontext   Commands to manage your DbContext types
  migrations  Commands to manage your migrations
Use "ef.dll [command] --help" for more information about a command.
```

EF Core CLI commands



Package Manager Console

Package source: nuget.org | Default project: SamuraiApp.Data

PM> get-help entityframework

TOPIC
about_EntityFrameworkCore

SHORT DESCRIPTION
Provides information about the Entity Framework Core Package Manager Console Tools.

LONG DESCRIPTION
This topic describes the Entity Framework Core Package Manager Console Tools. See <https://docs.efproject.net> for information on Entity Framework Core.

The following Entity Framework Core commands are available.

Cmdlet	Description
Add-Migration	Adds a new migration.
Drop-Database	Drops the database.
Remove-Migration	Removes the last migration.
Scaffold-DbContext	Scaffolds a DbContext and entity types for a database.
Script-Migration	Generates a SQL script from migrations.
Update-Database	Updates the database to a specified migration.

SEE ALSO
Add-Migration
Drop-Database
Remove-Migration
Scaffold-DbContext
Script-Migration
Update-Database

EF Core PowerShell commands



EF [Preview] Commands Require an Exe

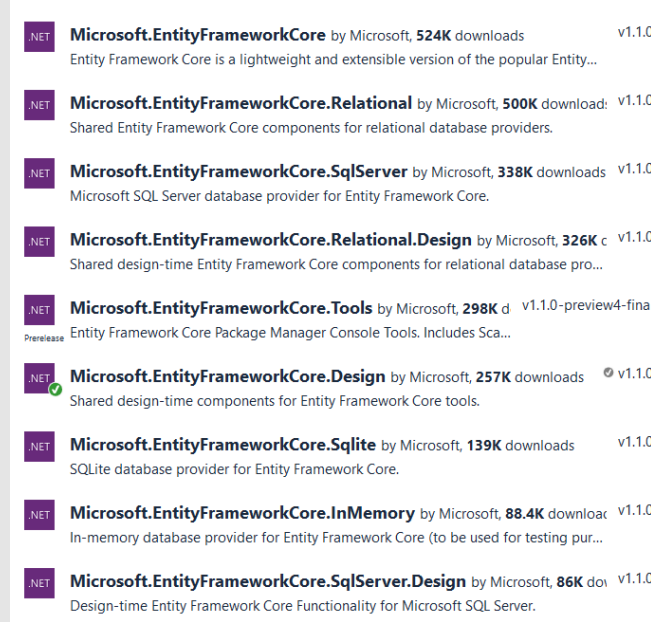
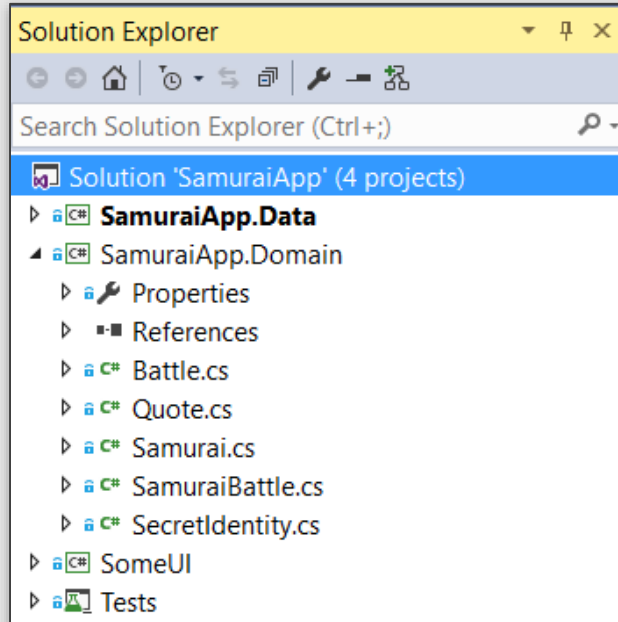
```
dotnet ef --startup-project [pathToExe] [ef commands]
```



Using EF Core Migrations in .NET Core CLI



Review



```
public class SamuraiContext : DbContext
{
    public DbSet<Samurai> Samurais { get; set; }
    public DbSet<Battle> Battles { get; set; }
    public DbSet<Quote> Quotes { get; set; }

    protected override void OnConfiguring
        (DbContextOptionsBuilder optionsBuilder)
    {
        optionsBuilder.UseSqlServer(
            "Server = (localdb)\\mssqllocaldb; Database = SamuraiData; Trusted_Connect..."
    )
}
```

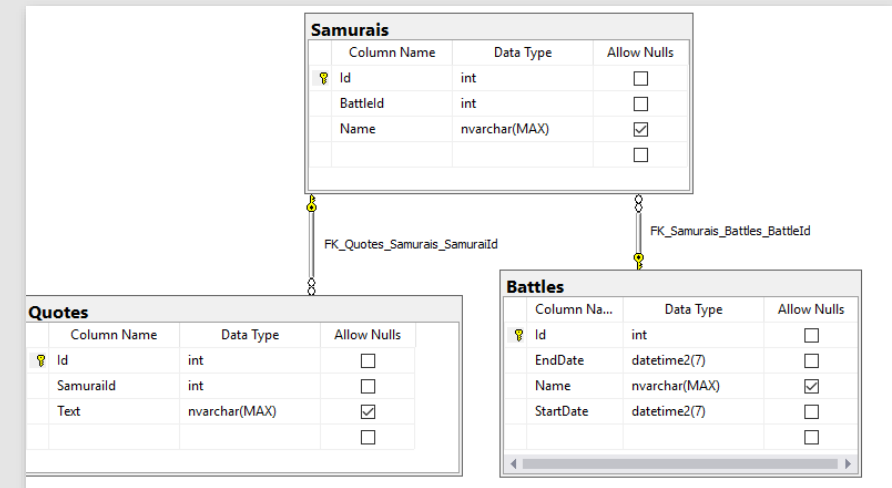
Script-Migration vs. Update-Database

Script-Migration

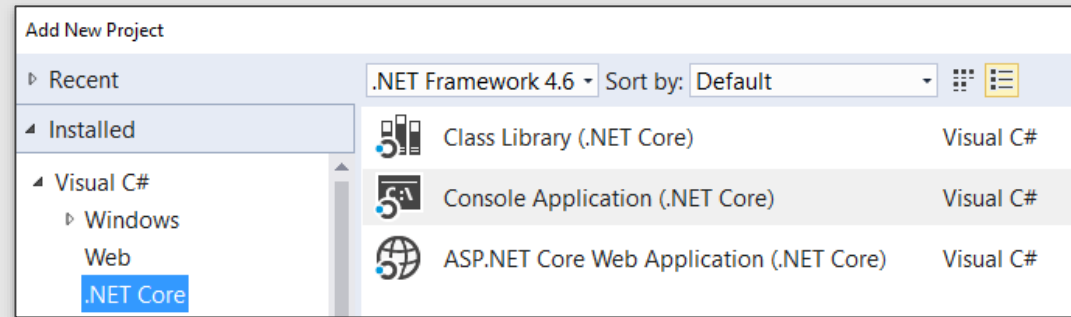
More control
Production databases
Apply needed tweaks

Update-Database

Best for local development database



Review



```
Entity Framework Core Console Commands 1.1.0-preview4-22752
Usage: ef.dll [options] [command]
Options:
  -h|--help                Show help information
  --version                Show version information
  --verbose                Show verbose output
  --no-color                Do not use color in console output
  --assembly <assembly>   The assembly file to load.
  --startup-assembly <assembly> The assembly file containing the startup class.
  --data-dir <dir>         The folder used as the data directory (defaults to current working directory).
  --project-dir <dir>      The folder used as the project directory (defaults to current working directory).
  --content-root-path <dir> The folder used as the content root path for the application (defaults to application base directory).
  --root-namespace <namespace> The root namespace of the target project (defaults to the project assembly name).
  -e|--environment <environment> The environment to use. If omitted, "Development" is used.

Commands:
  database    Commands to manage your database
  dbcontext   Commands to manage your DbContext types
  migrations  Commands to manage your migrations

Use "ef.dll [command] --help" for more information about a command.
```



Resources

Entity Framework Core on GitHub github.com/aspnet/entityframework

EF Core Roadmap bit.ly/efcoreroadmap

EF Core Documentation docs.efproject.net

Info about MSBuild:

blogs.msdn.microsoft.com/dotnet/2016/11/16/announcing-net-core-tools-msbuild-alpha/

.NET Core Installation page www.microsoft.com/net/download/core

Troubleshooting the dotnet ef command for EF Core Migrations bit.ly/efcoretips

Run EF Core on Both .NET Framework and .NET Core bit.ly/2dw4Vab

MSDN Magazine, Data Points Oct 2016, Julie Lerman

CodeMaid Visual Studio Extension codemaid.net

Add New File Visual Studio Extension bit.ly/2egY9I8

Open Command Line Visual Studio Extension bit.ly/2jJ2qp1

