# EntityFramework (EF)

## Package Manager Console

### Enable-Migrations

### Add-Migration InitialModel

Scaffolding migration 'InitialModel'.

The Designer Code for this migration file includes a snapshot of your current Code First model. This snapshot is used to calculate the changes to your model when you scaffold the next migration. If you make additional changes to your model that you want to include in this migration, then you can re-scaffold it by running 'Add-Migration InitialModel' again.

### Update-Database

Specify the '-Verbose' flag to view the SQL statements being applied to the target database.

Applying explicit migrations: [201610051435163\_InitialModel].

Applying explicit migration: 201610051435163\_InitialModel.

Running Seed method.

### Add-Migration CreateGigTable

***Model must derive from DbContext before EntityFramework can build scaffolding***

### Add-Migration CreateGigTable

Only the Designer Code for migration 'CreateGigTable' was re-scaffolded. To re-scaffold the entire migration, use the -Force parameter.

### Add-Migration CreateGigTable -Force

### Update-Database

***EntityFramework uses Convention over Configuration. There are two ways to overriding Conventions:***

**Data Annotations** - Easier but with limitation

**Fluent API** - More powerful but a bit more complex

### Add-Migration OverrideConventionsForGigsAndGenres

### Update-Database

When we execute add-migration. We have a class named in datetime\_migrationName. This class has two methods. Up() and Down(). Up is to tell how to upgrade the database. Thus Down is to tell it how to downgrade. This could be required if we required to revert to a previous version. For example, I wish to go back to the state when we migrated to CreateGigTable. To do this I will need to execute a command and that will run the Down() method of that migration plan. The command is: ***Update-Database -TargetMigration 201610051514436\_CreateGigTable***

### Add-Migration PopulateGenresTable

### Update-Database

## Errors & Tips

To resolve the following error:

**System.Data.SqlClient.SqlException**: Cannot attach the file '.\\*.mdf' as database '\*'

Ensure no other process is hooked into the \*.mdf file like SQL Server (Check using Manager) Drop using SQL and then Delete the file.

User.Identity.GetUserId() ==> is an expression thus it can not be converted to SQL

var artist = \_context.Users.Single(x => x.Id == User.Identity.GetUserId());

Storing the expression into a variable will force it to run. This will generate a primitive type, which of course can be converted to SQL

string artistId = User.Identity.GetUserId();

ApplicationUser artist = \_context.Users.Single(x => x.Id == artistId);