**Learning Migrations**

* قبل از ارائه اولین نسخه محصول، اصلا و ابدا نیازی به عملیات Migration نمی‌باشد!
* **به هیچ عنوان**، تا قبل از اولین مرحله Migration (InitialCreate)، نباید پروژه را به مشتری تحویل دهیم! و یا پروژه (سایت/سامانه) را Launch نماییم!
* عملیات Migration، کاری بسیار مهم، دقیق و احتمالاً پر خطا می‌باشد! و باید در یک تیم، معمولا یک نفر، که معمولا با تجربه‌ترین و بادقت‌ترین فرد می‌باشد، این عمل را انجام دهد!
* قبل از عملیات Migration، همیشه از بانک اطلاعاتی یک Backup می‌گیریم!

در صورتی که هنوز پروژه اجرایی نداریم، برای عملیات Migration، باید در کلاس DatabaseContext، یک Default Constructor نوشته و تابع OnConfiguring را Override نماییم!

ولی در صورتی که پروژه ASP.NET Core داشته باشیم، می‌توانیم به صورت Routine، از Constructor ای که ورودی Options دارد استفاده کرده و نیازی به Override کردن تابع OnConfiguring وجود ندارد.

Tools → NuGet Package Manager → NuGet Package Console:

Note: **Very Important!** Select [[Persistence]] Project:

Add-Migration InitialCreate → **Error!**

Add-Migration InitialCreate -Context DatabaseContextForMigration

Remove-Migration → **Error!**

Remove-Migration -Context DatabaseContextForMigration

مجددا:

Add-Migration InitialCreate -Context DatabaseContextForMigration

Update-Database → **Error!**

Update-Database -Context DatabaseContextForMigration

Check Database!

Remove-Migration → **Error!**

Remove-Migration -Context DatabaseContextForMigration → **Error!**

Update-Database 0 → **Error!**

Update-Database 0 -Context DatabaseContextForMigration

Check Database!

Remove-Migration -Context DatabaseContextForMigration

مجددا:

Add-Migration InitialCreate -Context DatabaseContextForMigration

Update-Database -Context DatabaseContextForMigration

تغییراتی را در بانک اطلاعاتی اعمال می‌کنیم که اطمینان حاصل کنیم که عملیات Migration، به درستی عمل می‌کند! یعنی بعد از عملیات Migration جدید، اطلاعات موجود در بانک اطلاعاتی حذف نشده و مشکلی ایجاد نمی‌کند!

حال مثلاً در جدول User، یک Property به نام AdminDescription ایجاد می‌کنیم!

**نکته مهم:** یا فیلد‌های Nullable ایجاد می‌کنیم! و یا اگر یک فیلد غیر Nullable ایجاد می‌کنیم، باید به آن‌ها Default Value بدهیم!

Add-Migration Version\_001 -Context DatabaseContextForMigration

فایل Migration را بررسی کرده و کدهایی که می‌خواهد اطلاعات پایه و یا تستی را حذف نماید را پاک می‌کنیم!

Update-Database -Context DatabaseContextForMigration

برگرداندن به حالت قبل از آخرین Migration:

~~Update-Database 0 -Context DatabaseContextForMigration~~

~~Remove-Migration -Context DatabaseContextForMigration~~

~~Remove-Migration -Context DatabaseContextForMigration~~

Update-Database InitialCreate -Context DatabaseContextForMigration

Remove-Migration -Context DatabaseContextForMigration

مجددا

Add-Migration Version\_001 -Context DatabaseContextForMigration

Update-Database -Context DatabaseContextForMigration

حال Property قبل، یعنی AdminDescription را به AdministratorDescription تغییر نام می‌دهیم:

Add-Migration **Version\_002** -Context DatabaseContextForMigration

Update-Database -Context DatabaseContextForMigration

migrationBuilder.DropColumn(

name: "AdminDescription",

table: "Users");

migrationBuilder.AddColumn<string>(

name: "AdministratorDescription",

table: "Users",

type: "nvarchar(max)",

nullable: true);

→

migrationBuilder.RenameColumn(

name: "AdminDescription",

table: "Users",

newName: "AdministratorDescription");