

第21期-数据库模型迁移工具

2019年4月28日 22:58

EF Core 提供两种主要方法来保持 EF Core 模型和数据库架构同步。至于我们应该选用哪个方法，请确定你是希望以 EF Core 模型为准还是以数据库为准。

如果希望以 EF Core 模型为准，请使用迁移。对 EF Core 模型进行更改时，此方法会以增量方式将相应架构更改应用到数据库，以使数据库保持与 EF Core 模型兼容。

如果希望以数据库架构为准，请使用反向工程。使用此方法，可通过将数据库架构反向工程到 EF Core 模型来生成相应的 DbContext 和实体类型。

模型迁移

开发期间，数据模型将发生更改并与数据库不同步。可以删除该数据库，让 EF 创建一个新的数据库来匹配该模型，但此过程会导致数据丢失。EF Core 中的迁移功能能够以递增方式更新数据库架构，使其与应用程序的数据模型保持同步，同时保留数据库中的现有数据。

迁移工具

PMC: Microsoft.EntityFrameworkCore.Tools

CLI: Microsoft.EntityFrameworkCore.Design

创建迁移

```
Add-Migration InitialCreate
dotnet ef migrations add InitialCreate
```

向 Migrations 目录下添加以下三个文件：

- (00000000000000000000_InitialCreate.cs--主迁移文件。包含应用迁移所需的操作 Up() 和还原迁移所需的操作Down() 。
- (00000000000000000000_InitialCreate.Designer.cs--迁移元数据文件。包含 EF 所用的信息,如给实体类型构建属性、主键、外键、索引、映射到数据表，主体和依赖关系等等。
- (BlogggingContextModelSnapshot.cs - 当前模型的快照。用于确定添加下一次迁移时更改的内容(用于递增更新)。

更新数据库

```
Update-Database InitialCreate
dotnet ef database update InitialCreate
```

当数据库模型未更改时，可能出现空迁移。

删除迁移

```
Remove-Migration
dotnet ef migrations remove
```

```
Update-Database -Migration:0
```

还原迁移

如果已对数据库应用一个或多个迁移，但需将其复原，可指定回退时的目标迁移名称。

```
Update-Database LastGoodMigration  
dotnet ef database update LastGoodMigration
```

生成 SQL 脚本

```
Script-Migration  
dotnet ef migrations script  
  
Script-Migration -From 0 -To InitialCreate -Output D:\InitialCreate.sql
```

删除数据库

Drop-Database

在运行时应用迁移

启动或首次运行期间，一些应用可能需要在运行时应用迁移。

```
myDbContext.Database.Migrate();  
  
dbContext.GetService<IMigrator>()
```

此方法构建于 IMigrator 服务之上，该服务可用于更多高级方案。