EF Core Tutorial

Using Singular Table Names with EF Core 2

`c#` `ef-code-first` `entity-framework-core`

English (en) ▼

Question

I want my domain class name to match my db table name (no pluralisation).

In EF Core 1.1, I used this code to do that:

```
public static void
RemovePluralisingTableNameConvention(this ModelBuilder
modelBuilder)
{
    foreach (IMutableEntityType entityType in
modelBuilder.Model.GetEntityTypes())
    {
        entityType.Relational().TableName =
entityType.DisplayName();
    }
}
```

In EF Core 2.0, this code doesn't compile as `Relational()` is not a method on IMutableEntityType. Anyway, in EF Core 2.0, they have added IPluralizer, documented here:

https://github.com/aspnet/EntityFramework.Docs/blob/master/entity-framework/core/what-is-new/index.md#pluralization-hook-for-dbcontext-scaffolding

There aren't many examples to show how to achieve the same behaviour that I had before. Any clue of how to remove pluralisation in EF Core 2?

Adam

Accepted Answer

You can use exactly the same code. `Relational()` is *extension* method defined in the `RelationalMetadataExtensions` class inside `Microsoft.EntityFrameworkCore.Relational.dll` *assembly*, so make sure you are referencing it.

What about `IPluralizer`, as you can see from the link it's just a *Pluralization hook for DbContext Scaffolding*, i.e. entity class generation from database, *used to singularize entity type names and pluralize DbSet names*. It's not used for table name generation. The default table name convention is explained in Table Mapping section of the documentation:

> By convention, each entity will be setup to map to a table with the same name as the `DbSet<TEntity>` property that exposes the entity on the derived context. If no `DbSet<TEntity>` is included for the given entity, the class name is used.

Ivan Stoev

Popular Answer

You can do it this way without using internal EF API calls by using the `ClrType.Name`

```
protected override void OnModelCreating(ModelBuilder
modelBuilder)
{
    foreach (var entityType in
modelBuilder.Model.GetEntityTypes())
    {
        // Use the entity name instead of the
Context.DbSet<T> name
        // refs https://docs.microsoft.com/en-
us/ef/core/modeling/relational/tables#conventions

        modelBuilder.Entity(entityType.ClrType).ToTable(entityType

    }
}
```

Nathan Robb

≝ View more on Stack Overflow

≝ View more on Stack Overflow