

Appunti gestione GWMS DB

Per la gestione dell'accesso al DB si opera con EFCore --> app blazor server

Versione MsSql iniziale

Versione iniziale basata su SqlServer

Scaffolding

Per generare le classi da un DB esistente con cui operare EFCore CodeFirst usare lo scaffolding coi seguenti comandi. Attenzione: la classe DbContext viene creata INSIEME alle viste nella folder DatabaseModel (nell'esempio seguente...)

DB iniziale

```
Scaffold-DbContext "Server=SQL2016DEV;Database=GWMS;Trusted_Connection=True;"  
Microsoft.EntityFrameworkCore.SqlServer -OutputDir DatabaseModels
```

SOLO di tabelle/viste selezionate (con force update)

```
Scaffold-DbContext "Server=SQL2016DEV;Database=GWMS;Trusted_Connection=True;"  
Microsoft.EntityFrameworkCore.SqlServer -OutputDir DatabaseModels -Tables nome_tabella, nome_vista
```

Reset DB MsSql

Per resettare un db MsSql (cancellando tutto x poter ridare update) eseguire il seguente codice

```
-- drop constraints  
DECLARE @DropConstraints NVARCHAR(max) = ''  
SELECT @DropConstraints += 'ALTER TABLE ' +  
QUOTENAME(OBJECT_SCHEMA_NAME(parent_object_id)) + '.'  
+ QUOTENAME(OBJECT_NAME(parent_object_id)) + ' ' + 'DROP  
CONSTRAINT' + QUOTENAME(name)  
FROM sys.foreign_keys  
EXECUTE sp_executesql @DropConstraints;  
GO  
  
-- drop tables  
DECLARE @DropTables NVARCHAR(max) = ''  
SELECT @DropTables += 'DROP TABLE ' + QUOTENAME(TABLE_SCHEMA) + '.' +  
QUOTENAME(TABLE_NAME)  
FROM INFORMATION_SCHEMA.TABLES  
EXECUTE sp_executesql @DropTables;  
GO
```

Il DB viene poi rigenerato dal Package Maganer console con

```
Update-Database
```

Versione MySql in produzione

Messa variabile boolean x imporre versione MySql (si potrebbe fare una cosa più "fine-tuned" x selezione aruntime del provider)

Pensata per impiego con MariaDB /MySql in caso linux dotnet 5

Impiegato provider dotnetcore **Pomelo**:

- <https://github.com/PomeloFoundation/Pomelo.EntityFrameworkCore.MySql>
- <https://www.tektutorialshub.com/entity-framework-core/using-mysql-mariadb-in-entity-framework-core/>

Generazione script

Sono stati ricreati gli script di generazione x MySql cancellando TUTTE le migrations + stato db (ogni file in folder Migrations) + comandi

```
Add-Migration InitDb  
Update-Database
```

Generazione DB + utente DB MySql

All'avvio applicazione, prima del seed di dati, viene eseguito un check dell'utente corrente dell'installazione x verifica se esista DB + Utente dedicati (come x applicazioni EgtBW ad esempio)

...da completare codice + descrizione...

Impiego multi provider

In caso di scelta per impiego di più providers (MsSql, MariaDB, SqlLite, ...) è necessario gestire in modo diverso le classi di inizializzazione del DbContentxt. Non ancora verificato, ma da approfondire secondo i seguenti links:

- Multi provider: <https://logu.co/efcore-multiple-providers.html>
- Migrations con + providers: <https://docs.microsoft.com/en-us/ef/core/managing-schemas/migrations/providers?tabs=dotnet-core-cli>
- Auto update all'avvio progetti (se le migrations sono state generate):
<https://jasonwatmore.com/post/2019/12/27/aspnet-core-automatic-ef-core-migrations-to-sql-database-on-startup>

Ottimizzazioni e spunti

Alcuni spunti da approfondire:

- Lifetime x DbContext secondo casi: <https://docs.microsoft.com/en-us/ef/core/dbcontext-configuration/>

Migrations

Approfondimenti:

- <https://docs.microsoft.com/en-us/ef/core/managing-schemas/migrations/?tabs=dotnet-core-cli>

Approfondimenti

Qualche link di approfondimento:

- <https://docs.microsoft.com/en-us/ef/core/>
- <https://docs.microsoft.com/en-us/ef/core/extensions/>
- <https://www.entityframeworktutorial.net/efcore/create-model-for-existing-database-in-ef-core.aspx>
- <https://entityframework.net/ef-code-first>