

NKC

Progetto per NKC Sauder con Egaltech

Architettura

L'architettura prevede

Db Scaffolding EFCore

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

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=Sauder_NKC;Trusted_Connection=True;"  
Microsoft.EntityFrameworkCore.SqlServer -OutputDir DbModels
```

SOLO di tabelle/viste selezionate (con force update)

```
Scaffold-DbContext "Server=SQL2016DEV;Database=Sauder_NKC;Trusted_Connection=True;"  
Microsoft.EntityFrameworkCore.SqlServer -OutputDir DbModels -Tables Models
```

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>

Note setup

Va installato

- powershell
- salt
- IIS
- SqlServer
- Redis
- tools vari x gestione dei 3

va inoltre configurato IIS per gestire le chiamate PUT/DELETE di WebAPI: in particolare per **abilitare i**

METODI PUT

Task	Come
Conf IIS	modificare il file applicationHost.config che si trova in <i>C:\Windows\System32\inetsrv\config</i>
WebDAV	disinstallare webDav oppure commentare le righe dell' <i>esempio (1)</i> sottostante
Conf IIS	aggiungere PUT/DELETE a handler come nell' <i>esempio (2)</i> sottostante

esempio (1)

```
<!-- <add name="WebDAVModule" /> -->
<!-- <add name="WebDAVModule" image="%windir%\System32\inetsrv\webdav.dll" /> -->
<!-- <add name="WebDAV" path="*"
verb="PROPFIND,PROPPATCH,MKCOL,PUT,COPY,DELETE,MOVE,LOCK,UNLOCK"
modules="WebDAVModule" resourceType="Unspecified" requireAccess="None" /> -->
```

esempio (2):

```
<add name="ExtensionlessUrlHandler-ISAPI-4.0_64bit" path="*."
verb="GET,HEAD,POST,DEBUG,PUT,DELETE" modules="IsapiModule"
scriptProcessor="%windir%\Microsoft.NET\Framework64\v4.0.30319\aspnet_isapi.dll"
preCondition="classicMode, runtimeVersionv4.0, bitness64" responseBufferLimit="0" />
<add name="ExtensionlessUrlHandler-Integrated-4.0" path="*."
verb="GET,HEAD,POST,DEBUG,PUT,DELETE"
type="System.Web.Handlers.TransferRequestHandler"
preCondition="integratedMode, runtimeVersionv4.0" responseBufferLimit="0" />
```

Specifica codici DataMatrix

Abbiamo convenuto di tenere i codici datamatrix piùcporti possibili. La lunghezza target è 10char. Le prime due cifre servono a distinguere COSA stiamo tracciando Le restanti 8 ci permettono di codificare in HEX (escludenxo 0x iniziale) cifre sino a FFFF FFFF = 4 miliardi di codici (INT di SQL è la emtà essendo con segno).

Ogni tipo di oggetto tracciato avrà quindi una codifica parlante in cui la prima cifra indicherà la sua natura secondo il seguente schema:

Cifra iniziale	Oggetto	Esempio	Note
BN	Bin	BN00041F120	Ceste, usiamo la conversione HEX del contatore BinId
CR	Cart	CR004B0F120	Carrelli, usiamo la conversione HEX del contatore CartId
IT	Item	IT000F1234	Usiamo la conversione HEX del contatore ItemId
IG	Item GENERIC	IG000F2314	Viene staccato ogni giorno un nuovo numero x indicare gli items uguali della giornata, con conversione Hex

Cifra iniziale	Oggetto	Esempio	Note
MT	Material	MT00006120	Viene usato il codice del materiale originale con padding di zero (no conv HEX)
OR	Order	OR000B0F120	Usiamo la conversione HEX del contatore OrderId
ST	Stack	ST000FB120	Usiamo la conversione HEX del contatore StackId

Postazioni

Postazione OrderManager

Questa prima postazione serve alla gestione preliminare degli ordini e quindi al calcolo del tempo stimato dal **NESTING + CNC** delle lavorazioni connesse.