# Aggiornamento del capitolo
# 29. Uso di EntityFramework 6 con SQLite

La libreria System.Data.SQLite che è stata utilizzata con Visual Studio 2019 e il .Net Framework 4.7 può essere utilizzata anche con Visual Studio 2022 e .NET 7.
In particolare il problema è che con Visual Studio 2022 non c'è più il file di configurazione App.config in formato XML ma c'è il file appsettings.json.
Nel file App.config c'erano le istruzioni per la configurazione dell'EntityFramework e la stringa di connessione al database di SQLite.

La prima possibilità è quella di evitare di usare un file di configurazione e programmare interamente l'EntityFramework con codice C#.
Si tratta di creare la classe SQLiteConfiguration per impostare SQLite come ProviderFactory:

```
using System;
using System.Data.Entity;
using System.Data.Entity.Core.Common;
using System.Data.SQLite.EF6;
using System.Data.SQLite;


namespace ProvaDatabase
{
   public class SQLiteConfiguration: DbConfiguration
   {
      public SQLiteConfiguration()
      {
         SetProviderFactory("System.Data.SQLite", SQLiteFactory.Instance);
         SetProviderFactory("System.Data.SQLite.EF6",
                SQLiteProviderFactory.Instance);
         SetProviderServices("System.Data.SQLite",
                (DbProviderServices)SQLiteProviderFactory.Instance
                .GetService(typeof(DbProviderServices)));
      }
   }
}
```

La classe ScuolaContext utilizza il costruttore della classe base passandogli un oggetto di tipo SQLiteConnection:

```
using System;
using System.Data.SQLite;
using System.Data.Entity.ModelConfiguration.Conventions;
using System.Data.Entity;

namespace ProvaDatabase
{
   public class ScuolaContext: DbContext
   {
      // definisco i DbSet delle tabelle del database
```

```
      public DbSet<Studente> Studenti { get; set; }
      public DbSet<Classe> Classi { get; set; }

      public ScuolaContext() : base(new SQLiteConnection(
                  "Data Source=scuola2023.db;Foreign Keys=On"), true)
      {
         //  true significa che la connessione viene eliminata quando
         //  il contesto viene eliminato
      }


      // per evitare la convenzione che i nomi delle tabelle abbiano
      // la s finale, ovvero siano  Studentes  e  Classes
      protected override void OnModelCreating(DbModelBuilder modelBuilder)
      {
         modelBuilder.Entity<Studente>().ToTable("Studenti");
         modelBuilder.Entity<Classe>().ToTable("Classi");
      }
   }
}
```

Le classi Studente e Classe sono le solite:

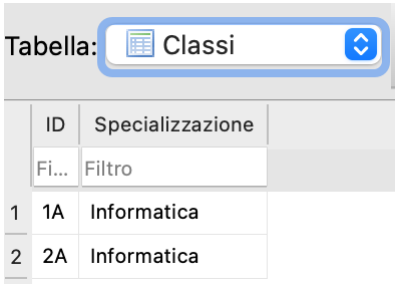
```
      public class Studente
      {
            public int ID { get; set; }
            public string Nome { get; set; }
            public string ClasseID { get; set; }
            // navigation property
            public virtual Classe Classe { get; set; }
      }

      public class Classe
      {
            public string ID { get; set; }
            public string Specializzazione { get; set; }
            // navigation property
            public virtual ICollection<Studente> Studenti { get; set; }
      }
```

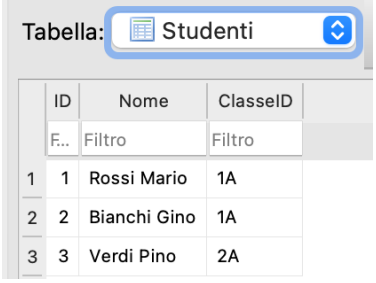
La classe Program consente di testare il funzionamento

```
using ProvaDatabase;

// aggiungere il pacchetto System.Data.SQLite

// See https://aka.ms/new-console-template for more information

// creo il contesto
ScuolaContext scuola = new ScuolaContext();
// gli studenti della classe 1A in ordine di nome
List<Studente> listaStudenti = scuola.Studenti
      .Where(x => x.ClasseID == "1A")
      .OrderBy(x=> x.Nome).ToList();
```

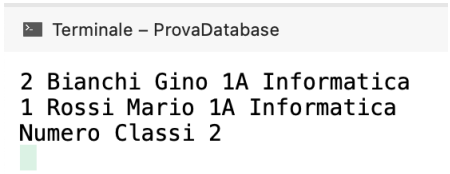
```
// stampa su console
foreach (Studente s in listaStudenti)
{
   Console.WriteLine(s.ID + " " + s.Nome + " " + s.ClasseID
                     + " " + s.Classe.Specializzazione);
}

// quante sono le classi di informatica?
int n = scuola.Classi.Count(x => x.Specializzazione == "Informatica");
Console.WriteLine("Numero Classi " + n);
```

Con i seguenti dati nel database:

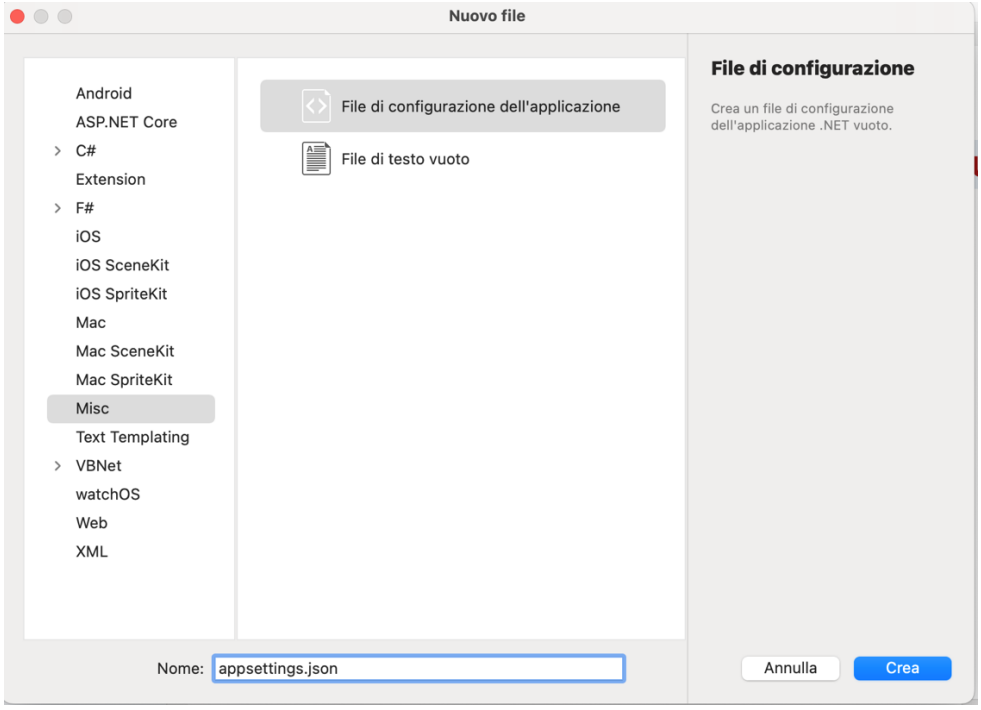



Si ottengono questi risultati



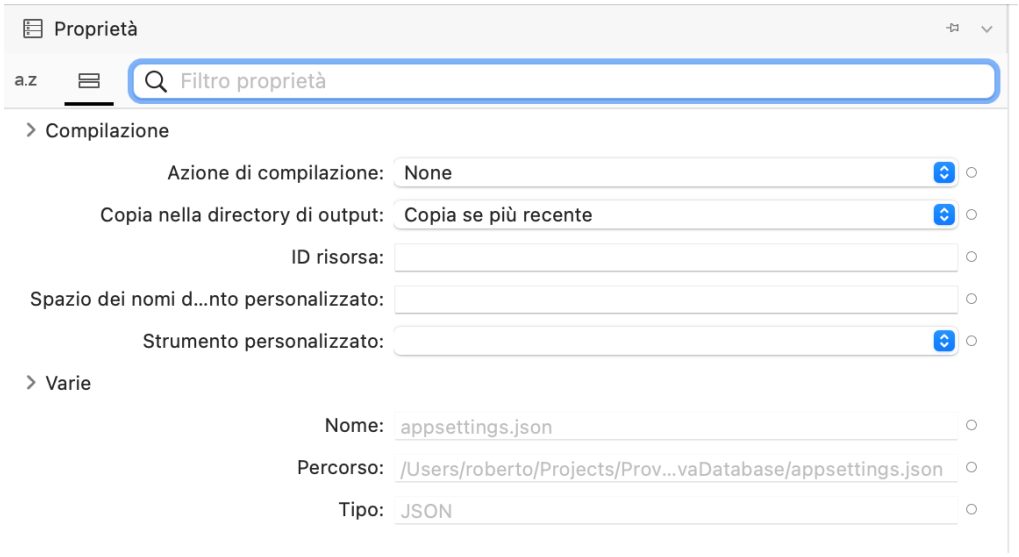

Qualora si preferisse utilizzare il file di configurazione appsettings.json per contenere la stringa di connessione, si deve innanzitutto aggiungere al progetto tale file:



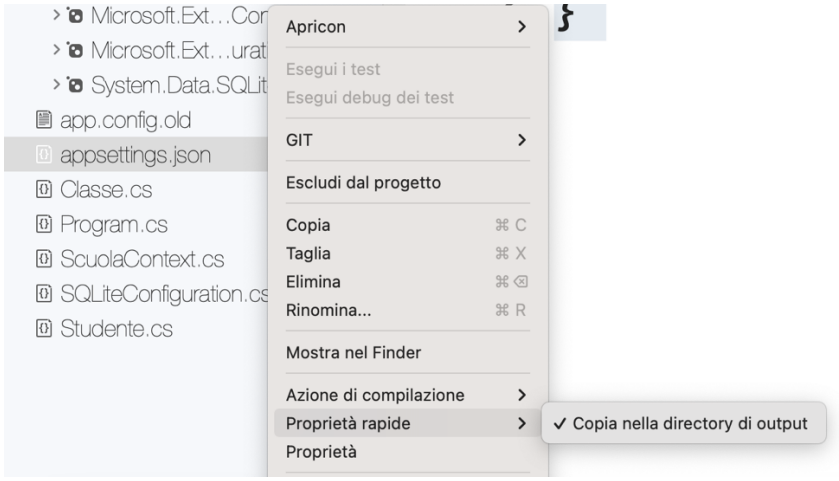
Nel file appsettings.json si aggiunge la chiave “ConnectionStrings” con valore un oggetto contenente il nome della stringa “Scuola” e la stringa di connessione al database:

```
{
  "ConnectionStrings": {
    "Scuola": "Data Source=scuola2023.db;Foreign Keys=On"
  }
}
```

Questo file deve essere copiato anche nella cartella di output dell’applicazione \bin\debug\net7.0 e per ottenere questo in modo automatico si clicca con il destro sul file appsettings.json, si aprono le sue proprietà e si applica la “Copia nella directory di output”:




Tale proprietà è anche accessibile come proprietà rapida:




Se questo procedimento non dovesse andare a buon fine, si può modificare direttamente il file .csproj che si trova nella cartella del progetto (nel nostro caso il file si chiama ProvaDatabase.csproj) e aggiungervi dentro la sezione <ItemGroup> le seguenti specifiche:

```
<ItemGroup>
   <None Update = "appsettings.json">
```

```
   <CopyToOutputDirectory> PreserveNewest </CopyToOutputDirectory>
   </None>
</ItemGroup>
```

Si devono importare i pacchetti Microsoft.Extensions.Configuration e Microsoft.Extensions.Configuration.Json per poter leggere il file appsettings.json.

Nella classe ScuolaContext si deve:

1) aggiungere la direttiva

using Microsoft.Extensions.Configuration;

2) aggiungere il metodo static GetConnection()

```
public static SQLiteConnection GetConnection()
{
   // è di tipo IConfigurationRoot
   var config = new ConfigurationBuilder()
                .AddJsonFile("appsettings.json").Build();
   string connectionstring = config["ConnectionStrings:Scuola"];
   return new SQLiteConnection(connectionstring);
}
```

3) modificare il costruttore in modo che utilizzi il metodo GetConnection() per costruire la connessione con la stringa letta dal file di configurazione:

```
public ScuolaContext() : base(GetConnection(), true)
{  }
// true significa che la connessione viene eliminata quando il contesto
// viene eliminato
```