

Guida all'Uso del Generatore Classi DAL (Data Access Layer)

di Giulio Ferri

Introduzione

Questo strumento permette di generare automaticamente classi VB.NET (Data Access Layer) a partire da tabelle e viste presenti in un database SQL Server. Le classi generate possono essere utilizzate per interagire con il database in modo semplice e tipizzato.

Le classi vengono create all'interno di file `.vb` contenenti:

- Le proprietà corrispondenti alle colonne della tabella/vista
- Metodi per: recuperare tutti i record (`GetAll`), recuperarne uno per chiave primaria (`GetById`), ordinare (`GetAllOrderBy`), recuperare record secondo condizioni custom (`GetWhere`), recuperare solo il primo record che soddisfa la condizione specificata (`GetSingleWhere`), inserire (`Insert`), aggiornare (`Update`) ed eliminare (`Delete`)
- Un `Enum` per l'ordinamento sicuro dei campi

Gestione degli errori

Tutte le classi generate includono una gestione automatica degli errori tramite queste proprietà statiche:

- **LastError**: contiene l'eccezione dell'ultimo errore riscontrato nel DAL.
- **HasError**: restituisce **True** se si è verificato un errore nell'ultima operazione.
- **ResetError()**: azzera lo stato di errore.

Come funziona

- Ogni metodo (GetAll, GetById, GetWhere, Insert, Update, Delete, ecc.) esegue in automatico il reset e la gestione try-catch. Se c'è un errore viene popolato **LastError**.
- Se il metodo ha successo, **HasError** sarà **False** e puoi usare il risultato.

Esempio di utilizzo:

Unset

```
Dim utenti = UtenteDAL.GetAll()
```

```
If UtenteDAL.HasError Then
```

```
    Response.Write("Errore: " &  
    UtenteDAL.LastError.Message)
```

```
Else
```

```
    ' Usa la lista utenti...
```

```
End If
```

```
' Su Insert
```

```
Dim nuovo As New UtenteDAL()
```

```
nuovo.Nome = "Mario"
```

```
nuovo.Insert()
```

```
If UtenteDAL.HasError Then
```

```
    Response.Write("Errore inserimento: " &  
    UtenteDAL.LastError.Message)
```

```
End If
```

Questa gestione è centralizzata, non serve usare try-catch ovunque nei form: basta verificare la proprietà `HasError` subito dopo ogni operazione!

Struttura della Classe Generata

Esempio per una tabella `Email`:

Unset

```
Public Class EmailDAL

    Public Property IDEmail As Integer

    Public Property Nome As String

    Public Property Email As String

    Public Property DataIns As DateTime

    ' ...altre colonne

Public Enum Campi

    IDEmail

    Nome

    Email

    DataIns

    ' ...

End Enum
```

Metodo `SetConnectionString`

Imposta la stringa di connessione condivisa per l'accesso ai dati:

Unset

```
EmailDAL.SetConnectionString("Data Source=...;Initial  
Catalog=...;Integrated Security=True")
```

Metodo GetAll

Recupera tutti i record della tabella:

```
Unset
Dim lista As List(Of EmailDAL) = EmailDAL.GetAll()

For Each email In lista
    Response.Write(email.Email & "<br/>")
Next
```

Metodo GetSingleWhere

Restituisce **solo il primo record** che soddisfa la condizione specificata, oppure **Nothing** se nessun record è trovato. È utile per ricerche dove ti aspetti un solo risultato (ad esempio per Email, Username, ecc.).

```
Unset
Dim utente = UtenteDAL.GetSingleWhere("Email =
'mario@pippo.it'")

If utente IsNot Nothing AndAlso utente.Attivo Then
    Response.Write("Utente attivo!")
End If
```

Gestisce gli errori in automatico tramite le proprietà `HasError` e `LastError`.

Metodo `GetById`

Recupera un record specifico tramite ID (la chiave primaria, presunta prima colonna):

```
Unset
Dim email As EmailDAL = EmailDAL.GetById(5)

If email IsNot Nothing Then
    Response.Write("Trovata: " & email.Nome)
End If
```

Metodo `GetAllOrderBy`

Restituisce i record ordinati per campo specifico:

```
Unset
Dim ordinati =
EmailDAL.GetAllOrderBy(EmailDAL.Campi.DataIns, False) '
DESC
```

Metodo **GetWhere**

Permette di recuperare le righe secondo una condizione personalizzata (qualsiasi condizione SQL valida dopo WHERE).

```
Unset
```

```
' Una sola condizione
```

```
Dim attive = EmailDAL.GetWhere("Stato=1")
```

```
' Due condizioni
```

```
Dim lista = EmailDAL.GetWhere("Stato=1 AND Nome LIKE  
'Mario%'")
```

```
' Condizione complessa
```

```
Dim lista = EmailDAL.GetWhere("Stato=1 AND (Citta='Roma'  
OR Citta='Milano')")
```

⚠ **Attenzione:** la condizione è inserita direttamente nella query SQL. Non inserire input non validato proveniente da utenti esterni per evitare rischi di SQL injection.

Metodo **Insert**

Inserisce un nuovo record nel database:

```
Unset
Dim nuova As New EmailDAL()
nuova.Nome = "Mario"
nuova.Email = "mario@example.com"
nuova.Insert()
Response.Write("Inserito con ID: " & nuova.IDEmail)
```

Metodo **Update**

Aggiorna un record esistente:

```
Unset
Dim email = EmailDAL.GetById(5)
If email IsNot Nothing Then
    email.Nome = "Mario Rossi"
    email.Update()
End If
```

Metodo Delete

Elimina un record dal database:

```
Unset
Dim email = EmailDAL.GetById(10)
If email IsNot Nothing Then
    email.Delete()
End If
```

Gestione delle Viste

Se viene generata una classe da una **vista**, i metodi **Insert**, **Update**, **Delete**, **GetById**, **GetAllOrderBy** **non vengono creati** (sono supportati solo per le tabelle). In quel caso, è disponibile solo **GetAll**.

Note Finali

- Le classi sono generate automaticamente nella cartella `/public/DAL/`
- Al termine della generazione, è possibile scaricare i file in formato `.ZIP` oppure uno singolo se è stata selezionata solo una tabella/vista
- I file vengono eliminati automaticamente dopo il download

Contatti

Per assistenza, segnalazioni o suggerimenti puoi contattare il team tecnico oppure migliorare questo script in base alle tue esigenze.